

КОМПЬЮТЕРНОМУ УЧЕНИКУ

Информатизация и UNIX.

Кузнецов И.А., 2022.

Автор разрешает свободное распространение книги без изменения текста, и его разрешения и уведомления, с возможностью аннотирования и смены оформления.

Посвящается всем кто сделал программы и руководства к ним бесплатными.
Спасибо вам.

1. Оглавление

Глава 1. Предисловие.....	10
Если вы недовольны этой книгой.....	11
Если у вас что-то не получается.....	11
Вы можете печатать эту книгу, или её части. Как и другие публикации.....	12
Глава 2. Проблема, касающаяся всех пользователей компьютерной техники.....	12
Глава 3. Пользователь, специалисты и информационные системы.....	13
История инфосистем и компьютеров.....	14
Аппаратура.....	14
Программная среда.....	14
Глава 4. А вдруг я что-нибудь нажму?.....	15
Глава 5. Я буду просто нажимать кнопки!.....	16
Глава 6. Программы.....	16
Программный фронтэнд, и бэкенд.....	18
Примеры программ.....	18
Первый пример.....	18
Второй пример.....	19
Третий пример.....	19
Четвёртый пример.....	19
Пятый пример.....	19
Шестой пример.....	20
Программы и команды.....	20
Глава 7. Что я смогу получить от этих информационных систем?.....	20
Платная работа.....	21
Глава 8. Проблемы с информационными системами. Медицинские нормы использования компьютеров.....	21
Медицинские нормы для компьютерной работы.....	23
Глава 9. Чего мне будут стоить все эти услуги?.....	23
Глава 10. А вдруг я что-то сломаю?.....	24
Глава 11. Мне нужен администратор! Вам нужен Шерлок Холмс.....	24
Глава 12. Что такое эти операционные системы, где они работают и почему я должен знать этот UNIX?.....	25
Что такое ОС с точки зрения пользователя.....	25
Юридический аспект использования и свойств операционных систем и программ.....	26
Наличные операционные системы.....	27
Версии операционной системы и обновление операционной системы.....	29
Оценка ОС, и опять о свойствах ОС.....	30
Несколько мифов о UNIX.....	31
Правда о UNIX.....	32
Где в первую очередь стоит применить UNIX.....	32
Где UNIX может вызвать проблемы.....	32
Глава 13. «Это для меня слишком сложно».....	32
Глава 14. Самые главные вопросы.....	33
Опасности.....	33
Требования техники.....	34
Архитектура компьютерного устройства.....	34
Требования носителей.....	35
Требования сетей.....	35
Требования программ и операционных систем.....	36
Глава 15. Отличия между операционными системами Windows и UNIX.....	36
«Мудрость» UNIX.....	37
Глава 16. Техника и покупка оборудования. Сеанс работы компьютера.....	37
Компьютерное устройство.....	38
Подключаемые устройства.....	42
Принтер.....	42

Сканер.....	43
Монитор.....	44
Камера.....	44
Устройство для электропитания.....	44
Звуковые устройства.....	44
Мышь.....	44
Клавиатура.....	45
Порты компьютера.....	46
Детали компьютерного устройства.....	47
Сетевые устройства.....	48
Покупка оборудования.....	52
Сеанс работы компьютерного устройства. Хранитель экрана (screensaver). Режимы энергопотребления. BIOS и EFI.....	53
Вход и выход из обслуживания операционной системой.....	55
Энергопотребление и «пробуждение» компьютера.....	55
EFI.....	56
Глава 17. Взаимодействие пользователя и компьютерного устройства.....	56
Программная среда и техника.....	57
Проблема интерпретации приёмов работы. Ведение тетради.....	57
Проявления программной среды.....	59
DE и оконная среда.....	59
Окна. Рабочий стол.....	60
Консоль и оконная среда. Простейшие действия с текстом.....	62
Элементы управления.....	63
Простейшие операции с текстом.....	66
Глава 18. Как получить помощь.....	66
map-страницы.....	67
Читаемые с экрана документы.....	69
Удалённый доступ.....	70
Глава 19. Трудности, ошибки и освоение программ. Неожиданности.....	72
Глава 20. Основы.....	73
Вопросы хранения данных и программ.....	73
Носители.....	74
Свойства и роль файловой системы.....	75
Свойства файлов и каталогов.....	76
Операции с файлами и каталогами.....	76
Автосохранение редактируемого файла.....	77
Live-диск, live-карта, или live-брелок.....	77
Вопросы работы программ, и программных процессов.....	77
Вопросы передачи данных и программ.....	78
Глава 21. Командное управление и автоматизация.....	78
Командное управление.....	78
Пользование масок файлов и каталогов при операциях с ними.....	79
Автоматизация.....	79
Основные команды.....	81
Стандартный ввод, выход и поток ошибок. Передача данных в программу и из неё с помощью пайпа (pipe). Перенаправление в файл и из файла.....	82
Перенаправление между программами.....	82
Перенаправление в файл.....	83
Если не хватает потоков вывода.....	83
А теперь всё вместе.....	83
Замечание про автоматическую обработку текста.....	83
Командный интерпретатор и программирование.....	84
Что такое язык программирования.....	84
Немного о BASH.....	85
Выполнение скрипта.....	85
Выдача строк на экран.....	85

Чтение строки в переменную.....	85
Выдача переменной на экран.....	86
Создание переменных в скрипте.....	86
Операторы в BASH.....	86
Условие.....	87
Повторение действия.....	87
Стандартный интерпретатор и его конкуренты.....	87
Особые случаи и предостережения.....	88
Команды и другие средства программирования в UNIX.....	89
Проблемы программирования.....	90
Глава 22. Базовые операции с файлами и каталогами.....	91
Скорость копирования и перемещения файлов и каталогов.....	92
Символические и жёсткие ссылки файловой системы.....	93
Глава 23. Поиск файлов, каталогов и текста.....	93
Recoll.....	93
Grep.....	95
Какой программой искать имена файлов и каталогов.....	96
Если у вас медленный компьютер.....	96
Определение размера содержимого каталогов.....	96
Пример поиска по свойствам файла.....	96
Поиск по MIME.....	96
Отбор некоторых файлов и каталогов.....	97
А также.....	97
Глава 24. Сравнение каталогов и текстов.....	98
Глава 25. Перевод англоязычных текстов.....	98
Перевод в браузере.....	99
Как установить расширение для перевода.....	100
Перевод в консоли или терминальной программе.....	102
Глава 26. Обработка текста. Вывод публикаций на бумагу.....	103
Понятие текстового файла.....	104
Базовые приёмы работы с текстом.....	105
Текстовый редактор nano.....	105
Основы редакции текста в nano.....	106
Текстовый редактор Geany.....	107
Gedit.....	108
Пара слов о Latex.....	108
Легковесные языки разметки.....	110
Распознавание текста.....	111
Печать.....	111
Алфавитный указатель.....	111
Далее про печать.....	112
Подготовка к печати.....	113
Пример устройства с дешёвой печатью – принтер Epson L120.....	114
Печать со смартфона.....	114
Печать в файл.....	115
Глава 27. Передача данных между программами. Буфер обмена.....	115
Глава 28. Электронные таблицы и базы данных.....	116
Глава 29. Звук и видео (мультимедиа).....	117
Оцифровка.....	119
Настройка звука в Linux.....	120
Глава 30. Изображения и анимация.....	120
Глава 31. Попутные работы.....	122
Архивы.....	122
Чтение и запись на носители.....	123
Взаимодействие с другими устройствами.....	125
Глава 32. Интернет и локальная сеть.....	126
Зачем нужен выход в интернет.....	126

Какой нам нужен выход в интернет.....	126
Интернет и разные типы сетей.....	127
Условия выхода в интернет.....	128
Протоколы связи.....	128
Акт подключения.....	128
Руководство по подключению к интернет с иллюстрациями.....	131
Связанные с интернетом программы.....	134
Оплата.....	134
Доступность и защита сети Wi-Fi.....	135
Проблемы с сигналом мобильной сети.....	135
Что я получу в интернете?.....	136
Браузер.....	138
Личный кабинет, личный режим и регистрации.....	142
Платное получение файлов и книг.....	144
Поиск в интернете.....	144
Пример сервиса – Overleaf.....	146
Пример сервиса – Wolfram Alpha.....	147
Покупки в интернет. Использование банковских карт.....	147
Госуслуги.....	151
Как пользоваться форумом (на примере rkprouincia.ucoz.ru).....	152
Документация и описание браузеров.....	155
Конкретнее об использовании консольного браузера.....	155
Браузер в телефоне.....	156
Оконные браузеры.....	157
Links.....	157
Firefox. Установка браузера с запросом установочного файла с помощью Links... ..	162
Midori.....	165
Opera.....	167
Vivaldi.....	167
Chrome.....	168
Yandex Browser (Яндекс браузер).....	169
Общее описание окна браузера.....	171
Браузеры в мобильных телефонах.....	174
Yandex Lite.....	174
Электронная почта (e-mail).....	180
Работа с почтой в браузере.....	181
Как удалить письмо.....	183
Торренты.....	183
Yandex messenger.....	185
Agia2: копируем файлы из интернета, в том числе через торрент.....	185
Регистрация Яндекс (Yandex).....	186
Парсинг сайтов и автоматизация работы браузера.....	187
Экономия и стоимость.....	187
Выделенные линии.....	187
Мобильные данные.....	188
Специальное предложение для загородного дома.....	188
Экономия при пользовании интернетом. Телефоны для консультаций.....	188
Как составить список файлов для запроса с другого компьютера или пересылки почтой.....	189
Запрос в репозиторий.....	190
Как узнать размер суммы файлов или одного файла.....	190
Запрос в файловое хранилище.....	190
Спутниковый интернет.....	191
Глава 33. Телевизионный плеер, интернет-радио, ленты новостей. Чтение публикаций.....	192
Мобильный телефон. Телеприставки. Инфосистемы вне городов.....	192
Библиотеки.....	193
Инфосистемы на селе.....	196

Сервис VPS/VDS.....	197
Ещё о локальной сети.....	198
Контейнеры.....	199
Голосовое общение через Wi-Fi локальную сеть. WhatsApp, Viber, ShareIt на селе.	199
Viber.....	201
Теперь рассмотрим ShareIt.....	203
Сообщения и звонки без интернета.....	203
Другие способы связи.....	206
Поселковый форум.....	207
Доступ к командной строке UNIX.....	207
Начальные сведения о веб-сервере.....	208
Быстрый и простой ввод в эксплуатацию уже настроенных поселковых сервисов.	208
Автоответчик, голосовое меню и SMS в качестве услуг в локальных сетях.....	208
Мобильный телефон.....	209
Телевизионный плеер.....	212
Интернет-радио.....	212
Чтение публикаций.....	213
Ленты новостей (RSS).....	213
Телеприставки (TV Box).....	213
Инфоматы, голосовые ассистенты.....	214
Микрокомпьютер.....	215
Подключение внешнего читателя лазерных дисков.....	216
Глава 34. «Электронная книга».....	216
Глава 35. Некоторые интересные программы для Android.....	217
Глава 36. Элементы автоматизации.....	218
AWK.....	219
Python.....	220
Программа paste.....	221
Регулярные выражения.....	222
xargs.....	222
Как получить имена всех файлов каталога с путями к ним.....	222
Как найти 10 самых больших файлов в файловой системе.....	223
Языки программирования.....	223
Глава 37. Обзор административных работ и вопросов защиты программной среды и данных.	223
Установка и удаление программ – классический путь.....	224
Особые случаи установки программ.....	228
Портабельные программы.....	229
Кросс-системные пакеты.....	230
Создание портабельных программ с помощью CDE.....	230
Безопасность и защита.....	230
Пароли. Двухфакторная аутентификация.....	231
Способы контроля пользовательской активности и защиты.....	232
Обновления.....	232
Обновление расширений браузера.....	232
Обновление браузера.....	233
Сведения браузера.....	233
Беспроводные сети опасны.....	233
Обновление антивирусных баз.....	233
Обновление программ роутера или точки доступа.....	233
Брандмауэр (firewall) и роутер или точка доступа.....	233
Временные файлы.....	233
Антивирусная проверка.....	233
Запрет на установку и использование программ, доступ к устройствам и сведениям о местоположении.....	233
Сайт организации.....	234

Резервирование данных.....	234
Перенос части инфосистемы на специализированный сервис.....	234
Контроль за работниками.....	234
Напутствие директорам.....	234
Права доступа, владение и UMASK.....	235
Учётные записи и регистрация пользователей.....	235
Выполнение заданий.....	236
Настройка сети и роутер или точка доступа.....	237
Настойка соединения.....	239
Проблемы соединения.....	240
Скорость передачи данных через интернет.....	241
Дисковые квоты.....	241
Подготовка носителей.....	241
Что такое форматирование.....	242
Лазерные диски пакетной записи.....	242
Подготовка жёсткого диска.....	242
Резервирование.....	242
Интересные программы.....	244
Отслеживание процессов и управление ими.....	244
Cpulimit.....	246
Переключение между программами.....	246
Демоны.....	246
Журналы (логи).....	246
Фаервол (брандмауэр).....	247
Об SSH и доступе к удалённому компьютеру.....	247
Производительность.....	248
Профилактика.....	248
Температура.....	248
Вода и моющие средства.....	249
Пыль и грязь.....	249
Программные процессы.....	249
Электропитание и ACPI.....	249
Управление контейнерами docker.....	250
Rclone и экономия при получении файлов из интернета.....	250
Сбор сведений об оборудовании.....	251
Использование образов дисков.....	251
Образы лазерных дисков.....	252
Образы флэш-памяти.....	252
Образы жёстких дисков для виртуализации.....	252
Организация фрэймбуфера.....	252
Глава 38. Идентификация, индексация и каталогизация.....	253
Глава 39. Искусственный интеллект.....	254
Глава 40. Как настроить шрифты в UNIX.....	254
Настройка шрифтов в консоли Debian.....	254
Настройка шрифтов в терминальной программе Mlterm.....	255
Шрифты в оконной среде.....	255
Некоторые рекомендации.....	255
Глава 41. Программирование в интересах регионов.....	255
Глава 42. Как подключить читатель лазерных дисков, мышь и клавиатуру к смартфону.....	256
Глава 43. Лазерные диски, дискеты. Каталогизация данных носителей.....	257
Лазерные диски.....	257
Дискеты.....	257
Каталогизация.....	257
Глава 44. Как попробовать другую операционную систему. Литература.....	258
Проба другой ОС.....	258
Ubuntu.....	258
FreeBSD.....	258

Debian.....	259
OpenSUSE.....	260
Fedora.....	260
Oracle Linux.....	260
NixOS.....	261
Android.....	261
ULBSD.....	262
Haiku.....	262
Hyperbola.....	262
Литература и сайты.....	262
Как попробовать другую систему.....	264
Чтение файлов другой операционной системы.....	267
Установка Linux на смартфон или планшет.....	267
Проба ОС на VPS/VDS.....	268
Live-диск или live-флеш брелок.....	268
Подробнее о Virtualbox.....	268
Другие эмуляторы.....	269
Глава 45. Вопросы качества и производительности работы.....	269
Человеческий фактор.....	269
Технический аспект.....	270
Программный аспект.....	270
Контроль рабочего процесса и журналирование.....	270
Глава 46. Что-то не работает или работает не так?.....	271
Что может выглядеть проблемным?.....	272
Глава 47. Будущее инфосистем.....	273
Alt Linux.....	273
Глава 48. Блокировка рекламы.....	274
Глава 49. Общие соображения.....	276
Глава 50. Словарь основных компьютерных и смежных терминов.....	276

Глава 1. Предисловие.

В России в 2010-е годы сложилась следующая ситуация. Можно сказать, что произошло наполнение рабочих мест и квартир компьютерами и подобными устройствами. Они стали не только привычными, но и часто необходимыми в быту и на работе. В обозримом будущем это так и будет.

При этом стало отчётливо видно что старшее поколение в плане своей квалификации совершенно отстало от молодёжи.

К тому же в силу причин политического и экономического свойства в стране распространяется новая для многих рабочая среда – тем более что она является естественной и основной на развивающейся отечественной электронике. Автор имеет в виду операционную систему Linux. Поэтому, несмотря на прошлый опыт компьютерной работы, нужно учиться. Это тем более верно в случаях поверхностной подготовки в уже привычной среде операционных систем семейства Windows.

Эта книга имеет в качестве задачи-минимум помощь людям, которые сталкиваются с агитацией или необходимостью смены операционной системы на Linux. В качестве более масштабной задачи выступает непрофессиональная подготовка людей, которые не имели базовой или полноценной подготовки даже в среде Windows. Тут я постараюсь помочь в том числе жителям провинциальных районов – совершенно неграмотным в области информатики. Эта книга будет полезна и тем кто не пользовался компьютером. Её содержание может помочь при освоении планшетов. Тут под компьютером понимается и микрокомпьютер. Эта книга направлена на создание картины, достаточной для понимания того с чем вам придётся иметь дело, что такое все эти непонятные «компьютеры» и «ентернеты». Так как ставится задача образования провинциалов, то написание этой книги сталкивается с дополнительными трудностями. Прошлый опыт может затруднить получение нового знания, и новых навыков.

Одна только терминология создаёт много трудностей.

Безграмотность во многом создаётся наблюдением успешной чужой работы за компьютером без чтения книг. Но не надо забывать, что обычно эти люди либо прошли курс информатики, либо – на самом деле – плохо знают этот предмет, и быстро работают лишь в узкой, привычной области.

Вспоминается ироничное замечание из классической литературы: *«Нет ничего удивительного, что человек такой поразительной наружности и такого странного поведения доставлял жителям Айпинга обильную пищу для разговоров. Относительно его занятий мнения расходились. Миссис Холл в этом деле была весьма щепетильна. На вопрос, что он делает, она обыкновенно отвечала с большой торжественностью, что он занимается «экспериментальными исследованиями», — эти слова она произносила очень медленно и осторожно, точно боясь оступиться. Когда же её спрашивали, что это означает, она говорила с оттенком некоторого превосходства, что это известно всякому образованному человеку, и поясняла: «Он делает разные открытия».* Или вот ещё шедевр из советского документооборота: вместо неприличных выражений «упал оголённый проводник» (то есть провод без оболочки) и «дом был разрушен применением бабы» (тяжёлое орудие вроде гири) было написано «упал одетый мужчина», и «дом был разрушен женщиной».

Разумеется, что при наличии компьютерных устройств и часто неплохих возможностях выхода в интернет роль бумажных книг всё более сомнительна. Тем не менее автор считает что это издание найдёт своего читателя. Это люди, которые либо не могут, либо затрудняются читать с экрана – по любым причинам. Автор вполне понимает нежелание и трудность использования компьютеров и подобных устройств практически *до их освоения*.

Компьютерная литература не похожа на привычные виды литературы. Это не нечто вроде детектива для чтения в поезде, уснув над которым вы ничего не потеряете. Это, можно сказать, учебная литература. Но при этом она в силу обстоятельств стала нужна и полезна огромному количеству людей. Читатели, разумеется, обычно не являются научными работниками. Поэтому технические и наукоподобные тексты могут быть не только не понятны, но и непопулярны. Читателей можно понять. Потребовалось много лет чтобы уйти от буквального, трудного для понимания стиля книг к более поверхностной, более пригодной литературе. Но при этом надо понимать, что читателю потребуются и разбираться с конкретной, требующей подготовки информацией, но позже. Это особенно верно про

классические компьютеры, появившиеся раньше планшетов и смартфонов, и микрокомпьютеров.

Этот труд не претендует на глубокую подготовку читателя. Есть понятие компьютерной науки, и она очень велика. Знать её в полном объёме практически невозможно. И не нужно. Нужно знать достаточно – а сколько это – в каждом случае решается отдельно. Тут мы, уважаемый читатель, подходим к важной проблеме, описанной немного ниже.

Эта книга может обходить конкретные вопросы. Это так не из-за лени автора, а потому что если вы хотите иметь хотя бы базовую подготовку, и знать важные новости, то вам всё равно надо уметь пользоваться браузером, и искать в интернете. Там вы найдёте свежую, и конкретную информацию. Использование браузера и поиск информации будут подробно описаны.

Наконец о том как была создана книга. Она была бы совсем другой если бы я не опирался на многочисленные статьи, и документацию программ. Надо высказать слово благодарности всем кто пишет полезные тексты по тематике книги, всем авторам описаний программ. Книга создана в LibreOffice Writer. Это качественная программа, которой я давно пользуюсь.

Если вы недовольны этой книгой.

Эта книга посвящена очень сложной и большой науке, которую, к сожалению, иногда просто необходимо знать – хотя бы отчасти. Эта книга определённо не достаточна для создания у вас полной квалификации – как и масса других книг. Также тут могут быть неизвестные вам термины. Они могут быть объяснены ниже по тексту, или в словарях. Если вас всё это раздражает, предлагаю посмотреть, например, раздел статистики в крупной библиотеке. Разве там одна книга? Попытки создать достаточный «компьютерный» учебник всегда заканчивались книгами чудовищного размера, и они были недостаточными.

Писать компьютерную литературу особенно трудно, и самое трудное – писать её для неподготовленного читателя. Это так потому что такая литература сильно напоминает попытку объяснить теорию относительности. Компьютеры уже стали таким же обычным устройством как телевизор. Но освоить их всё ещё может быть сложно, и не следует спешить. Люди рационального склада ума будут недовольны апелляцией к интуиции ученика, а люди интуитивного типа могут быть недовольны присутствующим рациональным подходом, как и невозможностью для многих обойтись без достаточно долгого этапа образования. Особенно трудно принять необходимость не просто действовать, а думать, и принимать решения самостоятельно – без знания того как работает компьютерное устройство. Заметим, что эта книга, как и масса подобных ей, не ставит задачу дать глубокое знание вопросов аппаратуры.

Есть такое мнение что я пишу слишком сложную литературу. Так же как то, что я пишу обзорную литературу. У меня лишь одно возражение: это не я пишу сложно. Я пишу о том что *есть сложно*. Попытки обойти сложные вопросы – это лёгкий путь для автора, но трудный путь для читателя. Что касается поверхностности моих текстов – да, я не всегда уплываю на глубину. Это правда. В наше время каждый ученик должен не просто уметь, а реально добывать знание – в том числе читать англоязычную литературу и документы с экрана по мере надобности; суммарная масса таких текстов очень велика.

Если у вас что-то не получается.

Не следует думать, что именно у вас – благодаря вашему уму и усидчивости – всегда всё будет получаться. Я не представляю как такое может быть. Такие случаи не есть повод останавливаться на пути. Наоборот, потребуются думать и читать чтобы двигаться вперёд. Освоение компьютерной науки никогда не было другим. А если вам говорят обратное, то это значит что, скорее всего, человек не много умеет, и легко учится. Скажу нечто более грустное: я практически гарантирую вам большее или меньшее раздражение на этом пути. Это – не подарок под новогодней ёлкой, но факт. Вот вы уже решили, что знаете достаточно, сделали всё правильно, и теперь, конечно, у вас всё получится. Но нет! Или даже вы уже сделали что-то. Но проверьте результат. Получилось? Нет? О чём это говорит? Вы мало знаете, и/или чего-то не понимаете. Обычно такие проблемы решаются чтением. Но в эмоциональном плане они обходятся зачастую очень дорого – гипертоники, пенсионеры – берегите себя. Это вам не вторая школа, это второй ВУЗ.

Вы можете печатать эту книгу, или её части. Как и другие публикации.

Эта глава набрана в предположении что эта книга распространяется как файл через интернет. Невежественные ученики, или просто имеющие трудности с чтением текстов с экрана могут отказаться читать с экрана. Это нормально. Библиотекари могут печатать любые готовые к печати документы – например эту книгу – следя за тем чтобы это не противоречило закону и расходы для печати не были чрезмерными. Нужно обсудить этот вопрос с директором библиотеки. Также частный читатель может – с указанными оговорками – печатать такие документы. В интернете есть множество полезных публикаций разной тематики, но они не всегда на русском, и не всегда уже готовы к печати. Поэтому надо иметь словари; для изучения компьютерных учебников нужны толковые словари по этой теме. Читатель заинтересован справиться с компьютерным устройством чтобы получить доступ к массе литературы и статей, читаемых с экрана.

Библиотека с возможностью дешёвой печати способна создать множество книг и статей без необходимости их типографского издания. Также городские власти вполне могут купить технику для дешевой печати, которая будет обслуживать городские библиотеки, или они могут заключить договор с типографией. Например, используя принтер Рихта 2411 можно с одной заправки чернил напечатать около 12000 чёрно-белых страниц обычного текстового заполнения. При этом сам принтер стоит примерно 9000 рублей, а чернила (альтернативные) можно купить примерно за 1500 рублей.

И печать, и подготовка материала для печати могут поначалу нервировать. Подготовка публикаций – более многогранная, сложная тема чем печать. В этой книге есть глава о печати.

Для выдачи на бумагу этой книги цветная печать не обязательна.

Глава 2. Проблема, касающаяся всех пользователей компьютерной техники.

Люди слишком невежественны и немотивированны чтобы достаточно знать. Или вообще хотя бы иметь какое-то понятие о компьютерной практике. Это является как социальной, так и технической проблемой. Например – и это очень важно – часто люди не просто не обеспечивают себя средствами защиты, но и не знают о том что это нужно.

Они же испытывает закономерные трудности при чтении специализированных руководств, насыщенных терминами, рассчитанных на некую – в данном случае не существующую – подготовку. В результате читатель руководств может просто забросить в дальний угол эту литературу. Но гораздо менее вероятно, что он так сделает с устройствами, про которые он пытался читать. Потратив на них деньги, пользователь берётся их применить. Так как он иногда неверно понимает устройства, то ему особенно трудно правильно с ними обращаться, и часто он делает неверные выводы о работе устройств.

Авторы инструкций имеют веские причины писать непонятные инструкции. Они заинтересованы писать как можно меньше чтобы минимизировать время на правку текста в условиях, когда время в буквальном смысле означает деньги. Поэтому даже такая крупная американская фирма как Oracle не пишет исчерпывающего руководства по своим продуктам. Она зарабатывает на консультациях, и отказывается от несения ответственности за проблемы со своей продукцией. В этой ситуации сам пользователь должен заботиться о своей подготовке.

Пользователь обращается к книгам, к статьям и журналам. Они также часто пишутся не очень понятно для невежд. Поэтому очень важно выработать правильное отношение к литературе. В качестве основы выступает так называемая грамотность. На сегодня учебники – в том числе школьной информатики – в библиотеке могут выглядеть совершенно по-разному. Не всегда это полезное чтение. Во-вторых, существует терминология. С ней лучше справляться с помощью словарей. Надо быть готовым к чтению нескольких книг – хотя бы частично.

Важно учитывать, что мнение людей не равно статье энциклопедии. Оно часто просто отражает их опыт и догадку. Автор этой книги не является специалистом и не имеет опыта платной работы как компьютерный специалист.

Ум, память, способность быстро работать не означают квалификации. Умение

пользоваться компьютером или чем-то подобным может вводить в заблуждение. Это не техника для гениев, и – в сегодняшних условиях – не всегда техника для специалистов. Наверное, многие пожилые ученики считают своих внуков гениями, и убеждены что они всё знают. Я провёл за компьютером более 20 лет. И я всё ещё учусь, и буду учиться. Сами пенсионеры переносят на компьютер свои представления о старой технике, и не волнуются о тех опасностях, которые специфичны для компьютерных устройств.

Сайт pkprovincia.ucoz.ru в большой степени посвящён инфосистемам вне городов. Он также пытается научить вас полезным сведениям, позволяющим получать современное обслуживание и услуги в области обработки информации. Однако этот сайт предполагает что вы уже имеете компьютерную грамотность. Для этого лучше всего найти соответствующий учебник в библиотеке. Также создан сайт pkp3.home.blog как продолжение тематики указанного сайта.

Глава 3. Пользователь, специалисты и информационные системы.

Информационная система – это, говоря простым языком – всякая система для обработки информации. В советскую эпоху такой системой являлись бумажные журналы, калькуляторы, печати, дыроколы и прочие подобные вещи. Компьютеры не использовались почти нигде. На современном этапе информационная система – это прежде всего или только компьютерная система, где основным средством обработки информации является как минимум одно компьютерное – в широком смысле слова – устройство. Оно в своих возможностях далеко превзошло калькулятор.

Информационная система может пониматься как система, обеспечивающая людей услугами. Но для того, чтобы она существовала и была полезной требуется *компьютерный администратор*. Это часто штатный сотрудник тех организаций, которые имеют информационную систему.

Кроме оборудования, такая система включает в себя нужные программы, которые ещё иногда приходится создавать или править, или настраивать. Этим обычно занимается программист – если говорить о больших программах. Он может быть временно нанятым специалистом.

Это – основа. Есть организации, где штат гораздо больше. В этой схеме пользователь – это не администратор, и не программист, а именно рядовой работник, занятый полезным использованием уже готовой к эксплуатации компьютерной системы. Это не заказчик или партнёр. Так, например, кассир в магазине является пользователем торговой информационной системы. Она должна обеспечивать ценники, выписки и выборки на бумаге, правильную работу касс, и так далее.

Но реально при использовании компьютерного устройства дома пользователь может быть вынужден выполнять роль администратора, и, иногда – в широком смысле – программиста. Это перспектива не радует. Но она не должна пугать. Администратор не является специалистом в технике. Тем более он не является инженером-электронщиком. То же про программиста. Последний не обязан разбираться в прикладной математике и других подобных вопросах. Пользователь может быть заинтересован в автоматизации, которая может требовать знать программирование на уровне дилетанта, в прикладной плоскости.

Есть понятие компьютерной науки. Она сложна и велика. Масса домашних пользователей во всём мире ограничивают своё образование не очень большой подготовкой. Но она обычно не является только результатом чтения, но и мышления, и опыта.

При трудностях домашние начинающие пользователи часто вызывают настройщика. Он не является специалистом узкой специализации. Он также в точном смысле слова не специалист по ремонту. Попутно можно заметить, что мастер, или настройщик – это часто чуть ли не мошенническая профессия. Пользователь должен понимать что он обычно способен и заинтересован сам решать почти все свои проблемы, хотя для этого ему потребуется самому учиться, думать, и действовать. Если вы вызываете мастера, то всегда старайтесь заранее выяснить стоимость работ.

Глубокое вмешательство в компьютерную систему может быть сделано лишь с пониманием того что именно делается, и что происходит. Оно часто и не должно выполняться

безответственными или невежественными людьми. Пользователю остаётся лишь поверхностно управлять некоторыми процессами. Он не может управлять всем, и часто не очень хорошо понимает что делает, а если это пожилой человек, что имеется склонность отталкивать от себя это знание. В таких условиях приходится работать. Невозможность сделать желаемое и невежество являются источником раздражения и огорчения.

В организациях должна быть *техническая поддержка* оборудования и программ. Иногда она существует как отдельная коммерческая фирма, отвечающая на звонки и способная прислать своего мастера. Техподдержка — в обычном смысле — как правило обеспечивается для легального владельца платной программы. Это не поддержка по вопросам оборудования или других программ.

История инфосистем и компьютеров.

Современное состояние инфосистем – не последнее, но ему предшествовал период долгого развития мысли и техники для счёта.

Аппаратура.

Вначале счётные машины управлялись вручную. Старшее поколение помнит что в СССР несколько десятилетий назад были в ходу ручные арифмометры «Феликс». Несомненно, это был большой шаг вперёд по сравнению со счётами. В 17-м веке они получили себе конкурента в виде логарифмической линейки. Но линейки тоже были в ходу, например, у инженеров. В 19-м веке англичанин Чарлз Бэббидж проектировал счётную машину, но при его жизни она не была собрана. Благодаря его идеям сформировались нужные представления. В частности, это было представление о программе работы счётной машины.

Первые компьютеры были электронными счётными машинами – ЭВМ. Другие функции добавились позже. Тут пора назвать электромеханический компьютер «Марк 1», созданный в США в 1943 году. Вторая Мировая война заставила английскую разведку использовать компьютеры для дешифровки немецких военных распоряжений.

В СССР, как и на Западе, первоначально компьютеры были доступны лишь учреждениям или фирмам. СССР даже однажды вырвался вперёд в мировой гонке компьютеростроения, обеспечив скорость вычислений 1 миллион операций в секунду (1967 год). *Персональные компьютеры* начали появляться на Западе в 70-е годы 20-го века. Ещё в начале 90-х годов в России персональный компьютер дома был редкостью. Технически он основан на микросхемах, вмещающих множество микроскопических «радиодеталей». На сегодня лишь несколько стран способны их производить.

Современные компьютеры в магазинах могут быть российской *сборки*, но не производства. Сейчас понятно, что российская техника уже существует, будет выпускаться на территории России, и быстро догоняет по характеристикам иностранные модели, но недостаточные тиражи делают её дорогой. Выход на мировые рынки обеспечит снижение цены, но это означает войти в конкуренцию уже существующим производителям, что может быть трудно. Финансовые вливания в компьютерное производство могут быть трудны для России. Тем не менее нужна собственная производственная база так как – ВАЖНО – импорт может подвести отечественные инфосистемы.

Старые инфосистемы были медленными, и включали в себя ручной труд – например, пробивку перфокарт. Перфокарта – это картонная пластина, отверстия которой обозначают данные или программы.

Программная среда.

Затронем и развитие программ и операционных систем. Операционные системы семейства Windows появились в 1985 году, но по-настоящему стали популярными в России в течение 90-х годов параллельно с увеличением парка техники. Операционные системы семейства UNIX – которым во многом посвящена эта книга – появились в конце 60-х годов. Сейчас в России и других странах есть ощутимое количество UNIX-подобных систем, используемых на персональной технике; это, прежде всего, Linux. Огромный процент техники продаётся с уже установленной Windows, и покупатель оплачивает её использование при покупке компьютера. На «больших» компьютерах – обслуживающих научные институты и прочие подобные организации – UNIX представляется более адекватной системой чем Windows, которая по сути была рассчитана на персональное использование и не имеет

достаточной устойчивости в работе. Третья система, которую нельзя обойти вниманием – Android – вышла на рынок носимых устройств в 2009 году. Будучи удобной для смартфонов и планшетов, она пока не достигла уровня «серьёзности» других рассматриваемых тут систем. Как система для устройств с малым экраном и управлением касаниями, Android может так никогда не перерасти своё «детское», второплановое состояние. Для компьютеров производства Apple была создана система OS X, которая не работает на другой технике. Также этой фирмой была создана операционная система iOS. Компьютеры Apple тут не рассматриваются не только из-за дороговизны, но и потенциальных опасностей импортной техники для России вообще. Также стоит обратить внимание на семейство систем BSD. Автор считает их интересными для российского применения. Для всех рассмотренных систем есть много полезных программ. Непонятные слова будут объяснены ниже.

Глава 4. А вдруг я что-нибудь нажму?

Есть такой страх. Он естествен. Люди, привыкшие к тяжёлой работе, или просто не привыкшие к мелкой работе руками, могут случайно нажать что-то, и даже не заметить этого. Обычно последствия случайных нажатий не очень вредны. Тем не менее следите за своими руками. Это особенно верно там где чувствительный экран. Наверное, лучшее средство от страха и плохих последствий – образование. Займитесь своими руками, помните пластилин, помассируйте руки, дайте им отдохнуть. Не беритесь сразу за ответственные, а тем более сложные дела.

Вам может помочь виртуальная машина (об этом ниже), или компьютер без ценных данных. Многие семьи имеют всего одно компьютерное устройство. Возможно, у ваших соседей есть старый, неиспользуемый компьютер. Некоторые организации достаточно богаты чтобы списывать старую технику.

Как быть если есть всего один компьютер, и он используется? Например так: отключить носители с ценными данными, и ценной программной средой. Вместо этого можно использовать live-диск, или флэш-брелок с готовой к использованию рабочей средой – где нечем рисковать. Выход в интернет (или локальную сеть), или действия там не являются причиной для наказания – пока вы явно не нарушаете закон, или ломаете что-то. Ученик не может случайно сделать это.

В классическом компьютере обычно единственным носителем ценных данных является жёсткий диск. Его не обязательно отключать физически. Достаточно настроить BIOS. Об этой программе написано ниже. Там же можно настроить – если надо – работу другого жёсткого диска – если он есть. Флэш-брелки и карты памяти проще всего отключить при выключенном компьютере.

Создание носителя для ваших опытов более сложно чем хотелось бы. Брелок имеет преимущество – он позволяет вам хранить данные на себе между сеансами работы компьютера. Вначале это может быть не важно. В интернет-кафе и библиотеках может не быть возможности создать лазерный диск – в данном случае live-диск. Скорее там возможно создать брелок для тех же целей. Всё что уже записано на него будет стёрто. Не делайте этого самовольно, а спросите можно ли. Брелок с Windows будет нелегальным. Напротив, брелок со свободно-распространяемой операционной системой будет легальным, и может быть более достигаемым так как если эту систему брать из интернета то в последнем случае есть шанс брать мало. Вы должны быть готовы к тому что в библиотеке вам откажут. Это может быть от того что библиотекари не понимают что вы хотите. Их не учили.

Также на один жёсткий диск можно поставить более одной операционной системы, и вам будет доступна часть диска.

Ещё можно практиковаться с системой, доступной через интернет или локальную сеть.

В городах может быть достаточно легко найти старый компьютер. Вы можете дать объявление в газету.

Вместо брелка можно использовать карту памяти, но к ней может потребоваться картридер.

Чтобы ваш носитель был полезен может потребоваться настройка BIOS.

Также можно практиковаться с отдельной программой, доступной через интернет.

Эта глава будет более понятна благодаря дальнейшему чтению.

Глава 5. Я буду просто нажимать кнопки!

Не получится. Типичное заблуждение пользователя состоит в том, что ему дадут не вызывающую разночтений инструкцию, и ему достаточно будет её выполнять. Это особенно ошибочно при домашней практике. От пользователя требуется не только и не столько знание, сколько способность ориентироваться и думать. При этом важно не пытаться зубрить тексты. Вместо этого надо собирать своё внимание, и обеспечивать организованную память. Поэтому нельзя между делом или в условиях посторонних вмешательств рассчитывать на продуктивную или безопасную работу.

Рассеянность, путаница в голове, незнание терминологии могут сделать компьютерные работы бесполезными, и даже опасными.

Также компьютерная квалификация не является просто набором конкретных, верных всегда приёмов работы. Она должна включать понимание роли, или функции действий. Надо уделять функции действий не меньше внимания, чем самому действию. Это позволит переносить одни и те же приёмы на разные рабочие среды, и использовать их по мере реальной надобности, а не механически. Также важно понимать роль логики. Если некая возможность должна быть, то она есть. И наоборот, если нечто не должно быть, то этого может не быть.

Рабочая среда может быть самой разной, и она не обязана быть способной делать всё что вы хотите. Она также не обязана быть похожей на другую рабочую среду.

Глава 6. Программы.

Понятие программы неизвестно совсем невежественным людям. Компьютерные устройства – в том числе мобильные телефоны – являются программно-управляемой техникой. Программы – это, по сути, инструкции работы других программ или техники. Конкретика программного управления техникой определяется особенностями этой техники. Тем не менее можно констатировать что ручное управление программами в большой степени унифицировано. Пользователю не надо знать подробности, ему не требуется много знать о технике. Эта ситуация тесно связана с устоявшимся набором условностей и приёмов.

На заре развития компьютеров программы могли сами управлять техникой без каких-либо окружающих сред, но и без ручного управления программой. Они являлись именно набором команд для оборудования. Позже ситуация изменилась не только из-за развития техники, но и для увеличения гибкости работы программы.

Самые первые программы были заняты только счётом. Они должны были принять данные, выполнить счётные операции, и обеспечить получение человеко-читаемых результатов работы. Но если вы хотите, например, как сейчас я, набирать и тут же видеть текст, то вам потребуется программа, способная не только работать, но и управляться вами одновременно. Управление должно обеспечить вывод на экран текста по мере его набора, и так далее.

Без организованного управления массой программ возникнут противоречия. Часть своих функций программа должна выполнять не самовольно, а при разрешении рабочей среды. Таким образом нужен не набор программ, а система программ. Поэтому существуют *операционные системы*.

Всё что вы будете делать за компьютером является не эксплуатацией или функцией компьютерного устройства, а функцией программ. Программы опираются на функции и возможности аппаратуры, но сами функции аппаратуры далеки от конкретных действий в программах, и от их назначения.

Если попытаться отобразить это на схеме, то получится, что ниже всего, в основе, лежит оборудование, затем программа BIOS, затем драйверы к оборудованию – это тоже программы, затем операционная система, затем прикладные программы – *приложения*, и утилиты. Определение утилит не очень конкретно. Оно близко к понятию служебной программы, то есть не имеющей никакого или много самостоятельного значения. Служебные программы сами не нужны для ваших целей, но нужны для поддержания аппаратно-программного комплекса в нормальном состоянии.

Программы часто – просто инструментарий для ручной работы. Так, например, если вы хотите рисовать за компьютером, то нужная программа будет чем-то вроде альбома с имитацией карандаша, кисти, ластика, и так далее. С другой стороны, есть программы,

которые сами выполняют работу, и присутствие человека не нужно. Но пока почти никакие программы не обладают интеллектом. То есть все решения принимает человек. Программа может выполнить рутинную работу. Программы можно поделить в зависимости от их разных свойств, а именно:

- количества ожидаемой ручной работы
- наличия полноценного описания программы
- требований к операционной системе, аппаратуре и выходу в интернет
- наличия хотя бы одного программиста для отслеживания современных требований, для совершенствования программы, и исправления её возможных дефектов
- адаптации к принятым в России условиям, и русскому пользователю
- скорости работы программы
- размера файла или файлов для установки программы
- необходимости предварительной подготовки перед использованием программы

Большинство программ для своей работы нуждаются в однократной установке. Установка может быть разной – в том числе при установке операционной системы. Обычно программа и до установки, и после неё представляет собой хотя бы один файл. О файлах написано ниже. Файл можно сравнить с книгой. Установку рассмотрим ниже более подробно. После установки программа обычно существует как множество файлов, среди которых есть хотя бы один, называемый программным, или исполнимым. Внешне такой файл может ничем не отличаться от других. Именно относительно него выполняется запуск программы, и благодаря этому программа работает. Установка выполняется однократно, хотя ничто обычно не мешает ставить программу более одного раза. Также программу можно удалить, и обычно это не является просто стиранием её файлов.

Период работы программы в принципе ничем не ограничен. Программа сама завершит работу, когда выполнит свою задачу – обычно определяемую человеком. С другой стороны, есть программы, задача которых не ограничена – например, обрабатываемыми данными, хотя, конечно, ограничена тематически. В качестве примера можно привести программы, обеспечивающие выдачу прогноза погоды или расписаний работы транспорта. По идее такие программы должны работать непрерывно – поставляя свежие данные по мере надобности в них. Известны случаи, когда компьютеры не отключают по много месяцев, и программы могут работать не меньше.

Программы обычно обрабатывают некие данные. Для этого они изначально и создавались. При этом они выполняют работу. Точнее, они работают, используя аппаратные возможности – прежде всего *вычислительную мощность*, и занимая *оперативную память* – вернее, часть её. То есть аппаратные ресурсы. В отдельных случаях такое использование сильно замедляет работу других программ. В таком случае может быть надо *оборвать* работу программы. Но это может оставить необработанными, и сделать непригодными данные, с которыми она работала. Особо требовательные программы иногда не нуждаются в ручном управлении, и их можно оставлять работать без большого количества конкурирующих программ – например, на ночь. Для деятельности массового потребителя такая ситуация не характерна.

В популярной практике компьютерное устройство используется для не очень долгих сеансов работы программ. Устройство поэтому обычно можно отключить почти в любой момент – но после завершения работы программ. Также большинство программ не работают без работы операционной системы.

Назначение программ достаточно трудно очертить, и с каждым годом сделать это всё труднее. Наиболее популярны программы для получения сведений из интернета, просмотра фильмов, слушания музыки, и офисных работ, а также инженерной и научной работы, как и учётной. Также очень распространено создание публикаций, а оно целиком делается в программах – хотя черновик можно написать на бумаге.

Каждая программа имеет название, и сферу своей работы. Сфера работы должна быть понятна потребителю, но сам потребитель разный. Если программа профессиональная, то постороннему человеку она может быть совершенно непонятна. Подобная проблема существует в том смысле что в документации к программе используется неизвестная ученику терминология. Это ничего не говорит о качестве программы. Документация к программе может быть как очень короткой, так и огромной. Это в основном определяется назначением программы. Также размер документации косвенно связан с тем насколько программа популярна и давно создана.

Возможно, некоторые авторы программ считают, что пользователю достаточно статей о программе, доступных в интернете или иначе. Насколько автор может судить, программисты не всегда уважают пользователя, и не всегда придают значение качественной документации.

Тем не менее мы видим много примеров хорошей документации, но обычно она на английском языке. Также стоит поискать книги на русском. Так, например, программа **AWK** создана много лет назад, с тех пор многократно усовершенствована, и на сегодня снабжена современной документацией размером в сотни страниц. Но эта документация на английском языке. Эта книга в большой степени описывает программные среды, о которых на русском мало книг. Вопрос об автоматическом переводе англоязычных текстов рассмотрен в этой книге.

Каждая программа должна быть совместима хотя бы с одной операционной системой. Несовместимая программа не будет работать. Также она должна быть совместима с оборудованием.

Программы достаточно часто являются товаром, и их использование имеет ограничения, накладываемые *лицензией*. Это обстоятельство много лет игнорировалось в нашей стране, и органы правопорядка допускали открытую торговлю программами с нарушением лицензий. Однако это было так когда у населения не было денег на оплату программ. Ситуация меняется. Старый опыт не пригоден, и беспечность в вопросе оплаты и соблюдения лицензии стала опасной. В частности, лицензия может запретить вам делиться программой с кем-либо.

Понятие версии программы обозначает выпуск программы, и её особенности. Версии, обозначенные как альфа, бета, и с номером меньше единицы являются рискованными так как не доведены до готовности к использованию. Авторы программы так показывают что в программе могут быть ошибки – баги (bug). Ошибки не исключены и в готовых к использованию программах, но их официальный статус уже другой. Ими можно пользоваться без опасений. Вы часто или всегда можете обратиться к авторам с сообщением об ошибке или вопросом. Хотя иногда авторы бросают свои произведения. Программа, официально созданная для одного выпуска операционной системы может не быть таковой для более позднего выпуска. С другой стороны программы могут появиться в наборе для более нового выпуска.

Программный фронтэнд, и бэкенд.

Это английские слова frontend, backend. Они могут сбить с толку пока вы не понимаете их смысла. Объясню их смысл на примере. В автомобиле водителю даны руль, педали и прочее как тому кто должен вести машину. По отношению к содержимому капота это – фронтэнд. А двигатель, трансмиссия, тормоза по отношению к содержимому салона – бэкенд. Иногда одна и та же программа может работать, и управляться и из консоли, и из окна, и работать без оконных фронтендов. Это особенность UNIX (об этом ниже рассказано).

Примеры программ.

Эта книга – по достаточным причинам – в основном посвящена UNIX, где много консольных программ, которые достаточно легко понять так как они управляются текстовыми командами, часто обрабатывают текст, и часто выдают текст – в том числе как комментарий к своей работе. О консоли и командах ниже будет ещё написано. Пока приведу примеры простых программ.

Первый пример.

Программа **wc** настолько проста, что для её понимания не нужно даже знать компьютерную грамоту. Программа показывает сколько строк, слов и байтов содержит файл. Основное назначение программы можно объяснить не прибегая к компьютерной науке. Файл подобен книге, а количество байтов обычно никого не интересует. Можно сказать что это количество букв, точнее символов. Программа запускается командой где надо набрать её имя, и название файла. Вот так

```
wc rufontshow
```

У меня на компьютере есть файл **rufontshow**, а у вас он не обязан быть. После набора показанной команды надо нажать на клавиатуре **Enter** чтобы стартовать работу программы. Программа выдаст результат работы и выключится. Всё это вместе будет выглядеть так.

```
wc rufontshow
1386 5309 199295 rufontshow
```

Таким образом, мы видим, что в файле 1386 строк. Это может быть практически важно.

Второй пример.

В свободно-распространяемых операционных системах обычно работает интерпретатор команд **BASH**. Сейчас мы не будем его разбирать подробно. Просто упомянем, что благодаря ему можно самостоятельно написать простейшие, или более сложные программы, и использовать их. В данном случае эти программы являются просто текстовыми файлами, но их надо правильно писать. Иначе их работа как программ будет неправильной. Напишем простейшую программу – только для того чтобы она показала что она работает. Как это сделать? Можно было бы написать некий текст в текстовый файл. Но можно поступить и иначе – подать выражение, являющееся программой, при вызове **BASH**. А именно так.

```
bash -c 'echo text'
```

Нажав **Enter**, мы заставим программу работать, и она выдаст:

```
text
```

Третий пример.

В среде UNIX существует команда и программа **date**. Используем её. И получим это.

```
date
Вт июл 9 18:26:04 MSK 2019
```

Программа обращается к аппаратным часам компьютера, и показывает дату, время, и день недели. Также она покажет город, точнее часовой пояс – MSK (часовой пояс Москвы).

Четвёртый пример.

Часто в сборку операционной системы входит **Python**. Это – программа, позволяющая вам писать программы любого размера для выполнения этим «питоном». То есть **Python** является таким же интерпретатором как **BASH** – с той разницей, что последний не предназначен для профессионального программирования. Если вы подадите команду

```
python
```

то получите что-то вроде следующего

```
Python 2.7.13 (default, Sep 26 2018, 18:42:22)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Это нормально. **Python** выдал несколько строк и ждёт что вы либо что-то введёте, либо нажмёте одновременно **Ctrl** и **D** чтобы выключить **Python**. Если вы что-то вводите, то **Python** либо посчитает это корректным выражением, и выполнит его, либо сообщит об ошибке.

Если вы рядом с «>>>» набьёте

```
1+1
```

и нажмёте **Enter**, то **Python** посчитает и выдаст ответ:

```
2
```

Затем вы можете нажать описанные клавиши для завершения работы интерпретатора **Python**.

Пятый пример.

Используйте теперь программу **SED** для изменения неких данных. Данные можно было бы прочесть из файла, но мы поступим иначе – подадим их прямо в командной строке.

```
echo 1|sed 's/1/2/'
```

Мы подали единицу как данные для обработки программой **SED**. Она работает, выполняя текстовые выражения – которые являются с её точки зрения программами. В данном случае это выражение заменяет лишь первую единицу на двойку. Результат будет показан как двойка. Между единицей и именем программы – не буква или цифра, а «палка» – вертикальная черта, позволяющая передачу данных между программами благодаря **BASH**.

Шестой пример.

Во многих операционных системах семейства UNIX установлена, но не работает оконная программа **clock**. Запустим её. Если работает оконная среда, то вы получите окно с часами. Или можно стартовать программу **oclock**. В обоих случаях пока программа работает вы не можете вводить команды там же где её запустили. Чтобы получить эту возможность надо оборвать работу программы, нажав одновременно **Ctrl** и **C** пока окно терминальной программы активно. Непонятные термины станут понятны по мере чтения.

Программы и команды.

Часто вы прочтёте или услышите про команды в среде UNIX или другой среде. Так же как про программы. Так что же работает, и какая разница между ними? Ответ не прост. Тут важно понимать терминологию. Команда – это всегда строка, и она подаётся в интерпретатор – не обязательно **BASH** – для её оценки и исполнения. Даже простое нажатие **Enter** на клавиатуре может быть обработано с получением изменения на экране. Программа в классическом виде – это то что способно реально работать – в частности по указанию команд, при управлении с помощью команд. Так, например, подавая команду

```
date
```

Вы заставляете интерпретатор искать и запускать программу **date**.

Однако при этом можно подавать команды подобной программе. Например, если я подам команду

```
wc -l rufontshow
```

то «-l» будет командой (вернее, ключом) для программы **wc**.

С другой стороны, команды могут содержать не просто название программы, но и то, что считается текстовой программой, которая *выполняется* интерпретатором или другой способной на это программой. Так, если я подам команду

```
python -c 'print("abc")'
```

то я заставлю интерпретатор искать и запускать **Python**, который должен будет выполнять программу **print("abc")**. При этом **Python** сам должен оценить пригодность этого выражения для выполнения. Если оно не может быть выполнено то он выдаст сообщение об ошибке в выражении.

Глава 7. Что я смогу получить от этих информационных систем?

Просто говоря – новые средства массовой информации, и не массовой тоже. Разве не достаточно телевидения, радио, газет, книг, телефонии? Автор понимает это мнение. Однако не согласен с ним. Читатель, не хочешь ли ты выбросить телевизор? Нет? А почему? Разве мало того что останется? Значит телевидение нужно? Да. Но это – не всё. Инфосистемы – это качественный шаг вперёд. Это не просто системы для подачи вам информации. Они могут собирать информацию со стороны потребителя, и использовать её для выдачи информации ему же. Конечно, если вы разговариваете с телефонным диспетчером, то он тоже выдаёт информацию по вашему запросу, как и справочная служба. Но он дорог, и он может генерировать 10000 разных ответов в секунду? А инфосистемы могут. Но и это ещё не всё. Инфосистема умеет обрабатывать данные. Например, вам нужны сведения по урожаю яблок в Ростовской области за 2017 год – не по всей России, не за все годы. Инфосистема умеет отобразить эти данные и представить в удобном к прочтению и печати виде. Простейший пример обработки данных – счёт. Это, конечно, не единственный вид обработки. Но самый понятный.

Инфосистема позволяет вам создавать, изменять, хранить и распространять информацию, а также получать её. Пример чему – эта книга – набранная и свёрстанная, конечно, в компьютере.

Далее. Инфосистема может обеспечить вам частный характер деятельности и обслуживания. Секретность информации, и частное хранение данных. Вы читаете ту же газету что и другие. Это один и тот же тираж. То же с телевидением и радио, с книгами.

А что там за информация? Наверное, она мне не нужна – подумают некоторые. И

ошибутся. Конечно, вся информация вам не нужна. Но часть нужна. Что можно найти в интернете? Просто говоря – всё. Да, но, может быть, этого всего понемногу? Когда как. Я не жду что даже крупнейшая библиотека сможет сравниться с запасами интернета. Вспомните советский быт. Что у вас было? Справочная «09», и справочники, карты. Например, в моей семье был справочник городского транспорта. А у вас был? А что если меняется маршрут? Делать новое издание? Интернет не боится этой проблемы. Вы получаете всегда свежую информацию – настолько, насколько свеж её источник.

Платная работа.

Инфосистема – реальный способ заработка. Часто люди работают с информацией. Так, например, конструирование – это форма работы с информацией о параметрах, размерах, свойствах, технологии, и так далее.

Это особенно интересно людям, которые не могут работать физически, а также ездить на работу. Во многих случаях не важно где вы работаете. Важно уметь работать. Родители с малыми детьми, нездоровые, имеющие избыточный вес могут попытаться брать работу на дом. Ради этого стоит учиться и купить компьютер не стесняясь в средствах. Им потребуется стабильный и достаточно быстрый выход в интернет. Вблизи городов это не исключено. Даже оплата работы может не требовать приезда – деньги попадают на банковскую карту.

Глава 8. Проблемы с информационными системами. Медицинские нормы использования компьютеров.

Ситуация в России заметно отличается от ситуации в США и Европе. Замечены следующие проблемы:

- заблуждение, что компьютерные устройства и инфосистемы можно игнорировать. Это уже не так. Инфосистемы делаются без уступок необразованным людям, но стали массовым, необходимым явлением. Средства массовой информации не помогают даже понять значение инфосистем. В результате ремонтные конторы многократно увеличили необоснованные доходы на невежественных людях
- пассивность населения – прежде всего старшего поколения – в вопросах образования по инфосистемам. Это отношение, по-видимому, имеет корни в техническом, статическом, ошибочном понимании инфосистем. Примитивные представления об оборудовании и программах из-за этого, непонимание важности совместимости частей инфосистемы. Непонимание временного эффекта вызова мастеров, непонимание потребности самостоятельно настраивать программы. В результате настройка делается подростками, которые могут быть не очень знающими, а реальные пользователи так и остаются невеждами. Когда же помогать некому, то они остаются без нужных им услуг
- не всегда адекватные школьные курсы, и не всегда качественные очные курсы
- большая трудность достаточного образования по инфосистемам. Это так в том числе из-за наследия 60-х годов XX века, когда инфосистемы управлялись специалистами. С другой стороны, распространённые рабочие среды создают заблуждение что *и не надо много или глубоко знать*. Компьютеры и программы являются самым сложным предметом среди популярных. Это огромная проблема, и она не будет решена быстро. Ученик должен понимать что он не сможет быстро и в то же время хорошо изучить инфосистемы
- заблуждение, что работа за компьютером – это просто нажатие известных кнопок, без понимания и мышления. Достаточно часто авторы, избегая трудного для них пути сущностного объяснения, просто указывают нужные действия. Они же избегают рассказывать об основах, что особенно трудно для них. Они же часто не контролируют себя, и дают бессмысленную, бесполезную информацию. Читатели идут на поводу у такой литературы, и, в результате, мы имеем массу людей, выполняющих «ритуальные действия», неспособных справляться даже с небольшими трудностями. Старшее поколение понятия не имеет что надо думать. Оно избегает даже знания, и авторы не объясняют зачем нужно знание и мышление
- наплевательское отношение даже к вопросам защиты собственной информации, недостаток дисциплины

- отсутствие потребительского качества выхода в интернет для многих районов страны – обрывы связи, низкая скорость передачи данных, отсутствие связи вообще. Трудность или невозможность достаточной защиты из-за этого, или слишком позднее обеспечение защиты инфосистемы
- сравнительная дороговизна оборудования и расходных материалов для многих покупателей
- часто доверчивость к не очень подготовленному мнению, и самонадеянность у не очень подготовленных людей. Не всегда качественная литература, недостаток литературы, её высокая стоимость. Непонимание потребности в свежей литературе в библиотеках. Низкая доступность или отсутствие словарей — особенно качественных и специальных
- непонимание доступности информации о незаконном вторжении в инфосистему, отсутствие контроля деятельности подростков и понимания значения взлома инфосистем¹. Доступность вредной информации, дезинформация
- непонимание многих статей интернета, их неготовность к чтению учеником. Игнорирование невежества старшего поколения. Недостаток или отсутствие хороших пособий. Многие из них либо слишком поверхностны, либо непонятны среднему пользователю – особенно старшим пользователям.
- обывательские представления о защите информации, недостаток популярной литературы по защите²
- нервирование, и последствия стресса из-за неудач или избыточной нервной нагрузки. По мнению автора, именно компьютеризация является наиболее важной причиной ухудшения здоровья — замеченного в России. Будучи по сути психологической проблемой, стресс является угрозой производительности труда, здоровья детей, и так далее. Стресс в большой степени связан с попыткой сделать работу слишком быстро, он также связан с недостатком знания и понимания
- отсутствие нормального электропитания
- непонимание роли программ, незнание о них
- незнание о возможности автоматизации действий. Источником проблем является убеждение что всё известно, и что неизвестно того нет. Гораздо полезнее думать что известно лишь немного из всего что можно знать
- трудность введения в эксплуатацию инфосистем, неправильное отношение к ним, неграмотность – в том числе руководителей организаций где вводятся или используются инфосистемы. Недостаток квалификации, недостаток дисциплины и предвидения, недостаток системного мышления. Непонимание потребности в специальной квалификации людей, управляющих инфосистемами. Профанация. Неготовность пользователей к применению инфосистем, отсутствие качественных учебных материалов и курсов в любом виде. Руководители могут не понимать что инфосистема может требовать быстрого обслуживания настройщиком, или специалистом по ремонту или восстановлению данных, в противном случае нарушается или останавливается вся работа. Они же могут не понимать что инфосистема по своей природе – не то же, что их традиционный объект работы, и, являясь сложной компьютерной системой, нуждается в долгом и тщательном *планировании* и постоянном внимании. Вообще говоря, всякая система может быть сложна, и требовать много внимания и времени. Информатизация намного увеличивает количество работы руководителя, и повышает требования к руководству. Нужно добиться чтобы инфосистема эффективно служила организации, а в конечном счёте – людям, а не считалась просто очередной «модой», или чем-то посторонним; инфосистема вместе с традиционной системой организации работы должны входить в новую, более производительную систему. Начальству приходится решать многие вопросы вместе с компьютерным администратором – в том числе в русле квалификации последнего, в том числе конкретные. Пожилые руководители могут быть не готовы к такой работе. Им самим нужно ориентироваться в вопросах

1 Уголовно наказуемое действие.

2 Всё ещё пароль «123456» является самым популярным. Люди думают что пароль будут подбирать люди, а не компьютеры.

информатизации. Автор не считает правильным просто поручить компьютерному администратору управление и создание инфосистемы. Он не всегда разбирается в специфике организации, и может создать непригодную инфосистему (а начальство не знает о том что надо её проверять). Инфосистема должна строиться в связи с особенностями работы организации, с учётом обстоятельств, человеческого фактора, закона, и форс-мажорных ситуаций. Также надо понимать что инфосистема будет меняться, что может вызвать перерывы в её работе.

- компьютерозависимость (и другие виды психологической зависимости)
- туннельный синдром (болезнь рук)
- болезни зрения — предположительно от перенапряжения

Медицинские нормы для компьютерной работы.

Несоблюдение требований безопасности (людей) приводит к тому, что при работе за компьютером сотрудник может ощущать дискомфорт: возникают головные боли и резь в глазах, появляются усталость и раздражительность. У некоторых людей нарушается сон, аппетит, ухудшается зрение, начинают болеть руки, шея, поясница и тому подобное. При ненормированной работе возможно нервное истощение (!).

- не рекомендуется работать за компьютером более 6 часов за смену;
- рекомендуется делать перерывы в работе за ПК продолжительностью 10 минут через каждые 50 минут работы;
- продолжительность непрерывной работы за компьютером без регламентированного перерыва не должна превышать 2 часов;
- во время регламентированных перерывов целесообразно выполнять комплексы упражнений;
- при нерегламентированной работе повышенной интенсивности возможны головные боли, нервные срывы (!) и др.

В помещениях должна проводиться ежедневная влажная уборка и систематическое проветривание после каждого часа работы. Шумящее оборудование (печатающие устройства, сканеры, серверы и тому подобные), уровни шума которого превышают нормативные, должно размещаться вне рабочих мест сотрудников. Вы можете купить противошумные наушники, дверь в комнату можно обшить доступными в продаже шумоподавляющими материалами. Вместо открытого окна можно использовать кондиционер.

Важно, чтобы офисный работник сидя за компьютером находился за хорошо освещённым рабочим столом. Чаще всего именно плохое освещение рабочего места оказывает более пагубное для зрения влияние, чем сам факт нахождения за компьютером.

(по текстам Википедии)

Также отметим важность вредного влияния бликов на экране.

Напомню одно из правил Дейла Карнеги – отдыхайте прежде чем устанете.

В мае 2019 года ВОЗ описала признаки синдрома эмоционального выгорания. Его определяют как «чувство истощения энергии, увеличение умственной дистанции от работы, циничное или негативное отношение к работе, а также падение профессиональной производительности». Синдром определяют как феномен, вызванный длительным стрессом.

Сторонники существования синдрома считают, что современный мир неизбежно создаёт условия для выгорания. Офисные работники проводят за экраном по 8-10 часов в день, люди меньше общаются лично, пользуясь для этого соцсетями и мессенджерами, поток поступающей информации постоянно растёт. Человеку трудно справляться с таким давлением. Выгорание — естественное последствие этого, говорят эксперты.

(по текстам РБК)

Заметим что страх любого рода будет портить самочувствие и создавать угрозу здоровью.

Аппаратно-программный комплекс, которому в основном посвящена эта книга, появился исторически как рабочее место специалиста, человека с организованным мышлением и способного терпеливо работать.

Глава 9. Чего мне будут стоить все эти услуги?

Этот вопрос лучше прояснить заранее. Отвечаю.

Для огромного большинства населения хотя бы приобретение соответствующего телефона

или планшета – оправданное дело. Оборудование обойдётся вам дешевле если оно подержанное, и намного. За всю мою 20-летнюю компьютерную практику не изменилось правило «дешёвый компьютер стоит 300\$». Но можно и дешевле. Наверное, самое нужное дополнительное устройство – принтер. Он стоит начиная от 1500 рублей – но тогда дороже чернила.

Выход в интернет часто можно организовать с помощью мобильного телефона. Вы оплачиваете ваше оборудование для этого – в данном случае телефон, и обычно платите абонентскую плату. Часто в городах дешёвый проводной канал интернета стоит примерно от 200 рублей в месяц. Если вы пользуетесь мобильным телефоном, то лучше всего смотреть тарифы фирмы, предоставляющей вам мобильную связь. Всё ещё часто у вас есть месячный лимит на объём переданных данных. Его часто можно превышать, но за дополнительную плату. Также стоит помнить про связь через модем – например, если телефон не годится.

Если у вас плохой сигнал беспроводного интернета, то стоит купить антенну, или настроить роутер. Подробности проясните потом.

Если вы живёте вне действия наземных сетей, то стоит поискать дешёвый выход в интернет через спутник. Оборудование дорого, но его можно покупать коллективно.

Во многих общественных местах, и иногда на транспорте есть бесплатный выход в интернет.

Вам потребуется учиться. Если бы вы учились по одной книге то её объём был бы сотни страниц. Может потребоваться знать английский – на уровне школы.

Вам потребуется терпение и готовность к неудачам. Так же как при изучении математики в школе. Не доводите себя до мигрени. Оно того не стоит.

Многие программы и операционные системы бесплатны, но – обратите внимание – тут могут быть условия. Обычно они поставляются с лицензиями на использование. Лицензия, скорее всего, не существует на бумаге.

Мышление за компьютером, или подобным устройством, может быть достаточно сложным. Нужна гибкость, и готовность принимать решения. Нужно предвидение, и понимание последствий. Не всегда у вас будет инструкция. Вам потребуется принимать достаточно важные решения, иногда с учётом риска потери данных.

Глава 10. А вдруг я что-то сломаю?

Наверное, лучше всё же не задавать этот вопрос после покупки техники, а задавать до неё. В обиходе компьютерные работы называются «я работаю с компьютером». На самом деле вы работаете почти всегда с программами, а с оборудованием работаете не вы, а операционная система. Эта прослойка содержит так называемую «защиту от дурака», она защищает технику и программы от опасных или тупиковых ситуаций. Тем не менее нельзя плясать на кнопках, и пускать на стол кошек. Это не сможет сломать технику. Но сможет повредить программы, и данные.

Глава 11. Мне нужен администратор! Вам нужен Шерлок Холмс.

Я пока не видел чтобы кто-то предлагал услуги именно в таком виде *частному* пользователю. Это было бы очень дорого. Кто же вам поможет? Отвечаю – часто вы сами. Скорее всего после соответствующего чтения, и мышления. Где взять информацию:

- на бумаге. Может быть дорого, и не всегда понятно. Начинать лучше всего с учебников для школы
- в интернете. Легко достигаемые тексты, но часто написанные в расчёте на уже существующую подготовку
- на лазерных дисках. Потребуется их купить. Как и с книгами, информация может устареть

Вызов компьютерного мастера может не только дорого обойтись, но и привести к обману. Вы не можете проверить исправность, а мастер может вас обмануть. Мастер не может просто унести дефектный носитель. Он должен скопировать с него на ваш носитель всю ценную информацию, и, что тоже важно, нельзя ему отдавать носители с вашими паролями, или вам

надо сменить пароли.

В качестве примера расскажу реальную историю. Мой компьютер, исправно работавший много лет, неожиданно перестал начинать работу. Примерно говоря, он перестал включаться. Это меня огорчило. Но не испугало. Всегда надо понимать, что нормальная работа и сохранность данных гораздо важнее поломок аппаратуры. В данном случае на компьютере не было ничего что было бы опасно терять. Более того – и не могло быть. Я понимал, что это аппаратная, или близкая к аппаратной, проблема. Я не мог быть уверенным что это именно поломка. Так как я уже давно вышел из подросткового возраста, то не стал рвать волосы, и спешить. Я не побежал звонить мастерам. Я мог долго просто думать, и состояние техники от проходящего времени не могло испортиться. Надо рассуждать логически. Перед проблемой обычно исторически расположена причина. Это сразу мне ничего не дало. Я стал ставить операционные системы. Ни одна не установилась. Я задал вопрос на форуме в интернете. Мне ответили то же что я сам понимал – раз много систем не ставятся, то дело в технике. «Глючный компьютер» – написали мне. Лишь операционная система FreeBSD установилась. К моему везению, я достаточно быстро читаю с экрана, и улавливаю смысл. Я вспомнил что после добавления планки оперативной памяти по экрану промелькнуло сообщение о «неоптимальном распределении памяти». Это такое описание для предупреждения о возможных проблемах. Причина оказалась совсем проста. Большинство используемых мной операционных систем *не сумели* работать с неоптимально поставленными планками оперативной памяти. Само это положение не вызвало поломок. И не могло. Так как вы физически *можете* неправильно поставить планки. Что это значит? Чтобы при этом не ставить нечто ценное под угрозу надо чтобы это либо позволило безопасно работать, либо чтобы вообще ничего не работало – и поэтому не могло испортиться. Как вы думаете это проблема аппаратуры или программ? В чистом виде – ни то, ни другое. Можно было вызвать мастера. Он переставил бы планки, и взял с меня много денег. Он помог бы мне? Да. А стоило бы его вызывать? Нет.

Глава 12. Что такое эти операционные системы, где они работают и почему я должен знать этот UNIX?

Существует точное определение операционной системы. Это «комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем» (Википедия). Определение верно, но не всем понятно, и не очень конкретно. С точки зрения пользователя операционная система (ОС) – это всего лишь рабочая среда. С точки зрения программиста – это система программ. С точки зрения администратора – средство обеспечения работы инфосистемы. Тем не менее пользователю надо знать что ОС – не просто место его работы.

Что такое ОС с точки зрения пользователя.

Это набор программ, удовлетворяющий следующим требованиям:

- возможность управления оборудованием – хотя бы основным, возможность собирать сведения о нём
- возможность записи, чтения, редакции, вывода на экран хотя бы текстовой информации
- организованное хранение данных и получение хотя бы базовых сведений о хранимых данных
- возможность получения и отправки данных через компьютерную сеть
- организованное управление набором установленных программ
- возможность обрыва работы программ при надобности
- средства защиты данных и секретности

Фактически, кроме этого от современной и массовой ОС ожидается следующее:

- возможность одновременной и бесконфликтной работы нескольких человек, обеспечение недоступности режима администратора для обычного пользователя
- возможность индивидуальной настройки рабочей среды для каждого пользователя
- защита данных одного пользователя от изменения и/или чтения другим пользователем
- не требующая подготовки специалиста система для управления нужными процессами и

- состояниями
- достаточная и не слишком трудная в понимании документация об ОС
- возможность обработки данных уже сложившегося типа (то есть формата) – иначе ОС не будет популярной
- умение ОС управлять популярным оборудованием – обычно через драйверы
- достаточность и полезность набора программ
- возможность автоматизации работы
- поддержка популярных сетей, протоколов связи
- обеспечение быстрого и полного исправления ошибок в программах
- возможность иметь на одном носителе более одной ОС
- наличие средств защиты данных и программ от вмешательства посторонних программ и пользователей
- набор средств именно для использования ОС в сложных и ответственных инфосистемах, набор средств для организации сервера
- поддержка особенностей, присутствующих в разных странах – письменность и прочее
- возможность одновременной работы нескольких (произвольных, разных) программ; часто есть возможность иметь несколько работающих экземпляров одной программы
- возможность оценки изношенности жёсткого диска (или другого носителя данных), наличие средств восстановления данных
- возможность общения с разработчиком ОС, непрерывность существования команды разработки, гарантия техподдержки в любом виде на обозримое будущее
- соответствие ОС запросам современных технологий

Этот список не надо зубрить. Он постепенно станет очевидными и понятными свойствами ОС.

Юридический аспект использования и свойств операционных систем и программ.

Обычно операционная система или программа поставляются с лицензией, хотя обычно это не печатный документ, и лицензия может быть лишь указана – иногда значком, а её текст надо искать, и она не предоставляется вместе с программой или операционной системой. Например так.



Википедия говорит, что лицензия на программное обеспечение — это правовой инструмент, определяющий использование и распространение программного обеспечения, защищённого авторским правом. Также она говорит, что бесплатное программное обеспечение, в отличие от свободного программного обеспечения, может распространяться без исходных текстов и может содержать ограничения на коммерческое использование, модификацию и т.д. Свободное программное обеспечение предоставляет каждому, помимо права на использование программного обеспечения, право модификации и ряд других прав.

В связи с отличиями между этими типами программ легко может возникнуть путаница между тем что свободно от уплаты (free of charge) и распространяется свободно (СПО, англ. free software, также software libre или libre software). Википедия говорит, что свободный софт (то есть программы) — программное обеспечение, пользователи которого имеют права («свободы») на его неограниченную установку, запуск, свободное использование, изучение, распространение и изменение (совершенствование), а также распространение копий и результатов изменения.

Пользователь также должен знать что такое условно-бесплатные программы. Условно-бесплатное программное обеспечение (англ. Shareware, от англ. share — делиться) — программное обеспечение с безвозмездным (или возмездным при определённых условиях) использованием. Свободное распространение не подразумевает его принадлежности к свободному программному обеспечению, равно как возмездность этого не исключает.

Также надо знать что такое коммерческое программное обеспечение. Коммерческое

программное обеспечение (англ. commercial software) — программное обеспечение, созданное с целью получения прибыли от его использования другими, например, путём продажи экземпляров. Лицензия всегда защищает материальные выгоды коммерческого использования программ.

Основной характеристикой проприетарных лицензий является то, что издатель программы в лицензии даёт разрешение её получателю использовать одну или несколько копий программы, но при этом сам остаётся правообладателем всех этих копий. Одно из следствий такого подхода заключается в том, что практически все права на программы остаются за издателем, а пользователю передаётся лишь очень ограниченный набор строго очерченных прав. Для проприетарных лицензий типично перечисление большого количества условий, запрещающих определённые варианты использования программы, даже тех, которые без этого запрета были бы разрешены законом об авторском праве.

В отличие от проприетарных, свободные и открытые лицензии не оставляют права на конкретную копию программы её издателю, а передают самые важные из них (составляющие 4 свободы по определению Фонда СПО, или попадающие под другое определение свободного или открытых программ) конечному пользователю, который и становится владельцем. В результате пользователь по умолчанию получает важные права, которые закон об авторском праве по умолчанию даёт только владельцу копии, однако все авторские права на программы по-прежнему остаются у издателя.

Пользовательское соглашение (англ. end-user license agreement — «лицензионное соглашение с конечным пользователем»; сокращается до EULA) — договор между владельцем компьютерной программы и пользователем её копии. Обычно используется вместе с проприетарным программным обеспечением (например, продуктами корпораций Microsoft, Apple), а также дистрибутивами свободного программного обеспечения с несвободными элементами (например, Mozilla Firefox до версии 3.0.4 включительно).

Кроме передачи прав и определения ограничений на использование программного обеспечения, лицензии обычно содержат условия, определяющие обязательства и ответственность для сторон лицензионного договора.

(по текстам Википедии)

Наличные операционные системы.

Теперь рассмотрим подробнее не свойства ОС, а реальные ОС. Наиболее популярной ОС на сегодня является что-то из семейства Windows. Это так с середины 90-х годов. Но все системы этого семейства — платные. Даже в богатых странах это важно. Windows версий 8 и 10 достаточно сильно нагружают технику. Владельцам старых компьютеров они могут не подойти. По описанным причинам стоит обратить внимание на другие — прежде всего бесплатные системы (заметьте: обычно на самом деле это системы лишь с возможностью бесплатного использования, убедитесь что ваша система такова, либо оплатите платную). Какие? Системы из семейств BSD и Linux (то есть GNU) — часто относимые к группе UNIX. Точнее, они являются похожими на UNIX. Эти системы существуют в виде разных разработок — более или менее похожих между собой. Не все системы удовлетворяют всем требованиям в полной мере. Важно понимать, что это часто проекты, которые никто не обязан поддерживать и продолжать. Тем не менее, даже фирмы ставят свою коммерческую деятельность в зависимость от этих систем. Есть такое мнение, что бесплатным может быть либо нечто некачественное, либо бонус. В данном случае это неверно. Бесплатный статус на самом деле происходит от исторических причин, и лицензирования. Большинство систем семейства Linux — бесплатны. В обиходе эти системы называют Linux'ами.

Являются ли эти бесплатные системы конкурентами Windows, достаточно ли они понятны, развиты и просты? Вопрос без общего ответа. Так как довольно рискованно говорить вообще, и это зависит от пользователя. Не все захотят вникать в особенности бесплатных систем. Это даже вопрос доступности программ. Тем не менее бесплатные системы, и, прежде всего, Linux, показывают хороший рост вопреки сопротивлению других систем. Есть такое мнение, что не стоит обращать внимание на ОС, которая не захватила более 2-3% парка техники. На самом деле точно этот процент трудно знать. Бесплатные системы бурно развивались, и развиваются, и часто предлагают не менее простую рабочую среду, чем Windows. Вы поступите мудро если попробуете UNIX — это вам не будет ничего стоить. Затем — если вы не сможете им пользоваться — вы купите Windows. Ей вы тоже, возможно, не сможете

пользоваться. Это зависит от вас. Часто Windows уже есть на классическом компьютере. Я не призываю её заменять. Тем не менее эта книга – о UNIX, который не следит за вами, работает на старом оборудовании, и более устойчив к вирусным программам. Также на работе UNIX более перспективен по причине больших возможностей автоматизации работ. Это не значит что он лучше везде и во всём. Windows знает гораздо больше людей.

Некоторым будет трудно именно работать с UNIX. Эти системы можно сравнить с потёмкинской деревней, где за достаточно современным фасадом иногда кроется «седая древность» в виде старых программ. Эти программы не буквально старые, они просто мало изменились со времени своего создания. Почему это так? Авторы UNIX не видят надобности что-то менять. Эти программы не плохи, они просто не подходят для тех кто не имеет времени или способности учиться. Выросло целое поколение молодёжи, привыкшее к примитивной деятельности в среде Android. Им может быть психологически трудно принять концепцию цепочки работающих на одну задачу программ. Это для них слишком сложно.

Также вы можете попробовать ОС Android на компьютере. Эта система намеренно сделана простой, но она менее гибка и информативна, чем «классические» системы — описанные выше. Она бесплатна. Также стоит обратить внимание на интернет-киоски — системы, требующие выхода в интернет, и намеренно упрощённые. Они полагаются на доступные в интернете полезные службы и просто чтение текстов оттуда, и пр.

Одна из причин движения в сторону альтернативных операционных систем – критика привычных, контролируемых бизнесом систем. Сайт gnu.org прямо пишет о вредности программ и операционных систем. В частности там сказано: «Microsoft случайно оставила пользователям способ установить GNU/Linux на планшетах RT с Windows, но теперь они “исправили” “ошибку”. Они имеют наглость называть это “защитой” пользователей. В статье говорится об установке “Linux”, но из контекста ясно, что на самом деле пользователи устанавливают GNU/Linux.», а также «Несвободные программы очень часто вредоносны (спроектированы несправедливо по отношению к пользователю). Несвободные программы контролируются своими разработчиками и производителями, что даёт им власть над пользователями; это в корне несправедливо. Разработчики часто пользуются этой властью в ущерб пользователям, которых они должны обслуживать.». Более того – журнал BYTE (?) опубликовал разгромную статью о Windows. Указать точные сведения автор не может, но статью читал, и журнал уважаемый и известный. Это было много лет назад.

Если это так – спросите вы – то почему Windows и другие подобные системы так широко используются. Отвечаю. Люди покупают компьютерное устройство, оплачивая уже установленную там операционную систему. Многие люди не считают важными недостатки своей операционной системы. Некоторые просто боятся менять то что есть на нечто другое. Некоторые не знают что альтернативы существуют, или считают их непригодными для своих задач. Многие люди заняты, и не занимаются ничем кроме простого использования операционной системы – не интересуясь её свойствами. Они могут думать что такая свежая система как у них, да ещё популярная – не может быть слишком плоха.

Но есть достаточно причин для замены системы. Наверное, самая суть этого в том что системы с открытым исходным кодом поддаются контролю пользователей. Никто не защитит их лучше чем они сами. Похоже что государства имеют склонность декларировать право на частную информацию и секретность, но решать свои задачи за счёт этого права. Они найдут множество причин для этого. Но пользователи не хотят чтобы приватное общение или секреты коммерческой работы, или тексты о достижениях, авторство которых может быть надо доказывать – стали доступны посторонним. То же верно про тайну финансовых операций, и вообще всё что раньше называлось «для служебного пользования». Ваш смартфон может быть использован для получения информации которую вы не хотите передавать посторонним.

Все указанные выше системы работают на обычных компьютерах типа PC (то есть персональный компьютер). Это – группа компьютеров, соответствующая спецификациям фирмы IBM. Обычно ими и пользуются. Тем не менее стоит рассмотреть и другую технику. Что это за техника?

- Планшеты и «умные» телевизоры, где обычно работает ОС Android.
- Мобильные телефоны, где часто работает она же.
- Другие мобильные телефоны.
- Телевизионные приставки, где часто также работает Android.

- Роутеры и точки доступа, где часто работает или может работать Linux или BSD, хотя изначально – после покупки – там работают свои системы неизвестного автору семейства.

Какое значение имеют эти устройства? Планшеты, телефоны, телеприставки, роутеры, точки доступа – это «младшие братья» компьютеров. Они оказываются более доступными по цене, и более приспособленными к переноске в кармане или сумке. В отдельных случаях – например, в командировке – это важно. Также отметим, что на планшетах с Android можно нетипичным образом поставить и использовать некоторые системы семейства Linux. Также может быть важно что часто это аккумуляторные устройства, независимые от постоянного электропитания.

Однако нужно понимать, что дешевизна и аккумуляторное питание не означают сами пригодность устройства. Другие качества могут сделать оборудование бессмысленной покупкой, или непригодной в планируемой инфосистеме.

ОС, родственные Solaris, на сегодня уже доступны для широкого пользователя персонального компьютера, однако иногда плохо документированы и совместимы с небольшим набором оборудования. Тем не менее стоит иметь в виду и их. Так как именно коммерческая ОС Solaris обладает большой документацией, и эта документация пишется и редактируется крупной фирмой – Oracle.

В качестве примера поддержанной коммерческой фирмой системы можно привести бесплатный для домашнего пользователя Oracle Linux. Лично я считаю эту систему не очень подходящей для старой техники, и русского пользователя. Тем не менее она является хорошим кандидатом на эксплуатацию у вас дома из-за большой англоязычной документации, и довольно большого количества статей о родственной ей системе Red Hat Linux. Также фирма Novell поддерживает бесплатную операционную систему OpenSUSE, которая всегда считалась дружественной и несложной в использовании, но её свежие выпуски могут слишком медленно работать на старых компьютерах.

На самом деле выбор гораздо шире. В России развиваются ОС Calculate Linux, Alt Linux, ROSA, Runtu. Simply Linux также является простой и достаточно быстрой системой, развиваемой в России. Многие ОС семейства Linux и BSD развивают интернациональные команды. Там никто не смотрит на национальность. Операционная система Ubuntu Linux поддерживается и развивается фирмой Canonical, и считается подходящей для учеников.

Начиная с 2012 года в быт стали входить микрокомпьютеры. Часто на них может работать Linux, и иногда – FreeBSD, NetBSD – ОС семейства BSD. Микрокомпьютеры очень дешёвы, и могут быть хорошим выбором для провинциального покупателя – не из-за удобства, а из-за дешевизны. Они не обладают большой вычислительной мощностью³.

Одним из доказательств устойчивой работы Linux является её применение в научной работе. Например, она, а не Windows, используется в CERN⁴.

Пользователю полезно знать, что все ОС семейства Linux можно поделить на несколько групп:

- родственные Red Hat
- родственные Debian
- родственные Arch
- родственные Slackware
- родственные Gentoo
- другие

Какое это имеет значение? Вы можете использовать документацию другой ОС из той же группы. Может быть полезна литература по UNIX.

Также надо понимать, что отдельные ОС рассчитаны на компиляцию программ, в других это – лишь возможность. Рассмотрим этот вопрос позже.

Версии операционной системы и обновление операционной системы.

Сейчас я буду писать лишь о UNIX.

Часто ОС имеет современный, сырой и устаревшие выпуски. Ваш интерес в том чтобы

³ Количество элементарных операций в секунду.

⁴ Крупнейшая европейская физическая лаборатория.

всегда использовать лишь современный выпуск. Это версия ОС, которая обычно обозначается как стабильная (stable) или текущая (current). Также версия, обозначаемая как release candidate не пригодна для работы когда нужна полноценная и несомненная надёжность. Для каждого выпуска наступает момент окончания обслуживания. После этого любые проблемы с ОС могут игнорироваться её производителем.

Обновление самой ОС иногда вызывает жалобы при обновлении без установки. Такой способ называется rolling. Он позволяет сохранить нужные параметры программной среды, но не считается лучшим в плане работы ОС после обновления. Автор не рекомендует этот способ. Он может быть оправдан лишь в крайнем случае. Автор не считает что достаточной причиной является невежество ученика в вопросах установки или настройки ОС.

Лицензия ОС – как я понимаю – определяет возможные лицензии используемых программ. Существует понятие либеральной лицензии. Что это такое? Просто говоря – это лицензия с относительно большими возможностями. На практике это значит что у пользователя, администратора и программиста нет чрезмерных препятствий к использованию ОС или конкретной программы. Часто это лицензия GPL. Она позволяет пользователю не нарушая лицензию просто скопировать ОС и программы, и использовать без оплаты. Поэтому важно быть грамотным, и не платить когда вам продают практически бесплатный товар. Причём эта торговля законна, но бессмысленна во многих случаях для покупателя. Ещё более либеральная лицензия – BSD. Я считаю её в конечном счёте опасной для пользователя так как она открывает бизнесу возможность вредного поведения даже без трудов по созданию программ, которые можно просто взять без нарушения закона.

Вы можете обратиться к своим знакомым, или связаться с кем-то через интернет. Вам даже обычно не обязательно чтобы этот человек тратил время и использовал лазерные диски. Любой носитель годится если он достаточно ёмок. Как минимум вы получите набор пакетов. Вам потребуется образ первого, установочного диска. Это может быть образ небольшого размера. Важно лишь что он позволит поставить ОС. Всегда будьте внимательны к тому что конкретно получаете. Если рассматривать Debian, то его просто передавать как набор файлов-образов дисков. Диски нумерованы. Чем меньше номер тем нужнее диск. Вы можете взять не все вообще доступные диски. У некоторых людей может сформироваться набор пакетов или образов полученных от разных людей или из интернет. Это нормально, и все их можно использовать при совместимости. Источник сам по себе не важен. Так, например, ваш сосед по району может без трудностей «залить» на вашу карту памяти довольно большое количество программ. Если для этого используется носитель, на который может быть поставлена – или уже стоит – некая ОС, то копировать пакеты или образы может быть лучше в отдельный раздел носителя. Это позволит не возиться с лазерными дисками, и сменять ОС не стирая полученные пакеты и образы. Если вы доверяете вашему помощнику можно просто переслать носитель по обычной почте. Ищите дешёвые и ёмкие карты памяти. Даже исправление файлов искажённых из-за низкого качества носителя – не очень проблематично. Непонятные слова объяснены ниже.

Оценка ОС, и опять о свойствах ОС.

Даже в среде Windows вопрос о выборе и оценке операционной системы важен. Он тем более важен при выборе ОС когда надо выбрать UNIX. Обозначу критерии выбора.

- Достаточность и характер программ-приложений, которые либо уже поставлены поверх ОС, либо должны быть поставлены по мере надобности. Для популярных систем семейства Linux и для FreeBSD достаточно программ – в том числе для платной работы, и всех популярных занятий – в том числе прослушивания музыки и просмотра видео. На настольном компьютере – например, дома, в таком случае оконные программы в плане своих свойств, и ожидаемого мышления и действия пользователя мало отличаются от таковых в случае Windows. Но при этом много и других программ, рассчитанных на большее знание, и более требовательных в плане обучения перед их использованием. В основном это консольные программы.
- Стоимость. Правильно говорить о стоимости услуг или рабочего места. В более узком смысле бесплатные системы выигрывают сравнение с платными так как их приложения обычно тоже бесплатны. Так, например, Debian обеспечен примерно 33000 пакетов – не считая пакетов-библиотек. Это не всегда самостоятельные программы. Даже если мы посчитаем, что лишь каждый 50-й пакет является самостоятельной,

полезной программой, то мы получим 660 программ. Все они, или почти все – бесплатны. На самом деле это заниженная оценка.

- Возможность работы оборудования – в том числе отдельных деталей. Для бесплатных систем этот вопрос в большой степени решён. Тем не менее надо понимать, что свежее оборудование – особенно если оно редко нужно – требует проверки. Не надо его покупать для этого. Достаточно почитать тексты интернета. Во-вторых, набор функций устройства и характер работы могут быть отличными от такового в случае Windows. В частности важно что ускорители видеокарт работают лишь в некоторых системах. Это важно так как ускоритель может кратно поднять скорость обработки данных.
- Возможность использовать маломощный компьютер, с небольшим объёмом оперативной памяти. Тут бесплатные системы выигрывают – в том числе из-за возможности выбора оконной среды или DE.
- Адаптация к требованиям, принятым в нашей стране. Это, прежде всего, язык. Ситуация достаточно хорошая – когда вы можете выбрать систему. Некоторые из бесплатных систем не работают с русским языком. Иногда это можно исправить без больших сложностей.
- Устойчивость работы, надёжность. По крайней мере старые системы семейства Windows считались менее устойчивыми, чем Linux и BSD. Надо различать устойчивость системы, и устойчивость отдельной программы.
- Документированность. Описания прикладных программ для бесплатных систем обычно есть, но на английском. Общие описания и учебники по бесплатным системам на русском языке пока изданы лишь в количестве не более одного книжного шкафа – если не полки. Среди этих книг я считаю хорошей, например, книгу Юрия Магды «UNIX для студента». Также надо быть готовым к тому что описание программы – на сайте. Если учебники по Windows могут грешить дилетантизмом, то книги и встроенная документация в UNIX имеет склонность в обратную сторону – что тоже плохо.
- Распространение. Это не очень важно. В интернете вы обычно можете найти сведения по любому вопросу. Но настройщика и администратора для бесплатной системы может быть трудно найти, и они могут быть дороже.
- Техподдержка вашей системы провайдером, то есть поставщиком интернет-канала. Это обычно не очень важно – по крайней мере для частного пользователя.
- Сложность и непонятность установки. Наиболее дружественные системы делают установку достаточно простой.
- Сложность эксплуатации. Наиболее дружественные системы и DE довольно наглядны и понятны, но нельзя исключать потребность в чтении и работу в терминале.
- Наличие дома и в продаже носителей с программами. Пока для бесплатных систем нет широкого предложения программ, и их самих, нет запасов программ дома. Но есть заказ файлов по почте. Ситуация не безнадёжна.

Несколько мифов о UNIX.

Как о всякой не очень известной вещи, о UNIX есть свои мифы.

- Это что-то очень устаревшее.
- Это для старой техники.
- Это годится лишь для англоговорящих стран.
- Это слишком сложно.
- Это для специальной техники.
- Это для расчётов и науки.
- Там нет игр.
- Очень трудно найти литературу.
- Там очень мало программ.
- Там всё делается только командами.
- Тем надо очень долго компилировать все программы.
- Там надо быть программистом.
- Это не будет работать с интернетом.
- Слишком многое там не будет работать.
- Почти невозможно ставить программы.
- Это как-то связано с нечистой силой (это из-за эмблемы FreeBSD).

- Так как программы часто сделаны любителями то их качество никудашное.
- UNIX это лишь программный мусор. Иначе он бы стоил денег.

Правда о UNIX.

- Отчасти то что является некоторыми мифами.
- Это во многом результат труда фирм и академических центров.
- Эмблема, как утверждается, безвредна, и обозначает совсем другое. Но выбирая эмблему надо всё-таки думать.
- UNIX действительно консервативен.
- UNIX действительно не всегда годится невежественным или нетерпеливым людям.
- Ваши возможности будут примерно такими же как в случае с Windows.
- Обычно не надо долго или сложно ставить программы.
- Популярные форматы, используемые в среде Windows, обрабатываются.
- Действительно в UNIX нет Photoshop. Так же как нет Word. Но там есть не буквальные аналоги этих программ.
- UNIX – это полноценная рабочая среда, со всеми нужными свойствами. Обычно UNIX распространяется свободно, и может продаваться.

В одной из книг про Linux было написано что-то вроде того что «вы в лабиринте одной из самых запутанных систем» – что-то такое, этот текст я не смог найти сейчас. К счастью, я прочёл это намного позже, *после* того как убедился что с этой системой можно «жить». Только не надо думать что я был 10-летним мальчишкой, и мои свежие мозги позволили мне примириться с этим монстром. Я помню как заказал диск с Linux где-то в районе 2008 года, а по-настоящему перешёл на него когда мне было около 40 лет. Мнение что перед вами сложная, требующая терпения система – полезное, но мне сейчас не кажется что Linux или UNIX так ужасны. Проблемно выглядят блуждания по каталогам если они связаны ссылками, и убогая консольная среда. Но и то, и другое я не считаю настолько крупной проблемой. Обстоятельства заставляют нашу страну двигаться в сторону Linux, и возражения против этого могут быть весомыми. Но они ничего не изменяют в принципе.

Где в первую очередь стоит применить UNIX.

- Где нет денег или нового оборудования.
- Где требуется повышенная надёжность и хранение тайны.
- Где нужна повышенная производительность или автоматизация.
- Где есть проводные каналы в интернет.
- Где нужно организовать цифровую АТС.
- Где нужна виртуализация.

Где UNIX может вызвать проблемы.

- Где персонал не может или не хочет учиться.
- Где используется экзотическое оборудование.
- Где нужна профессиональная работа со звуком.
- Где используется модное подключение к интернет.
- Где люди не знают английского языка.
- При очень медленных или дорогих каналах в интернет.
- Где нет денег на дорогого администратора.
- Где нужно использовать вычислительные возможности видеоплат.
- Где нужно использовать платы или адаптеры для Wi-Fi. Их надо заменить на точки доступа или роутеры.
- Где нужно точное соответствие работы периферийного оборудования их инструкции.

Глава 13. «Это для меня слишком сложно»

Так как эта книга в большой степени направлена на образование жителей российских регионов, я не могу обойти столь популярное мнение. Старшее поколение – выросшее в СССР – действительно не привыкло к сложностям в жизни, привыкло к трудностям. И всё же мне кажется что часто за этими словами – не неспособность понять сложные вещи, а нежелание ими заниматься. Что закономерно возникает пока не понятно зачем всё это. Это так в

условиях когда курсы могут быть не очень понятны, как и литература. Телевидение не помогает ориентироваться в вопросах информатики, как и радио.

И всё же – может быть это действительно слишком сложно? Думаю, что для отдельных людей это может быть так. Тем не менее это посильная наука для многих – особенно если есть время и внимательность. Такой ученик готов учиться, но его могут разозлить или остановить длинные полки специальной литературы. Всю её не надо читать. Её читают по мере надобности. Впечатление сложности может возникнуть при неорганизованном понимании материала. Также надо быть готовым к тому что термины не объяснены, и вообще не очень верны так как они являются переводом с английского.

А в общем вся эта наука, конечно, сложнее чем многие разделы профессионального знания в СССР – в том виде, в каком была реальная квалификация. Особенную проблему представляет то что надо не только знать, но и думать, и решать что делать. Если ученик раньше работал, например, инженером, то ему легче.

Глава 14. Самые главные вопросы.

Пора назвать самое главное, без чего нельзя считать себя подготовленным.

Опасности.

Опасности надо знать, но не надо их пугаться. Точно так же как не надо отказываться готовить еду чтобы не обжечься.

- Главная опасность – в потере ценной информации. Мы уже живём в информатизированной стране, и потеря данных означает в бизнесе остановку работы или крах, или большие проблемы. На уровне домашнего пользователя это всегда очень горько – даже если данные не очень нужны, или вы не помните что потеряли. Не следует думать что за компьютером можно только играть. Потеря набираемой студенческой работы, например, это проблема. Потеря информации вероятно прежде всего из-за отказа носителя, конкретно – флэш-памяти или жёсткого диска. Кроме того, потеря может быть из-за затирания старых данных новыми – прежде всего при неправильном копировании или переносе данных. В частности, очень опасно пользоваться непонятными программами, которые это делают. В крайнем случае вы будете вынуждены идти в сервис восстановления, но восстановление не гарантировано. Флэш-память имеет ограничения на количество актов записи данных.
- Вторая опасность – потеря нормальной работы, потеря качества рабочей среды. Это может происходить от неправильной деятельности пользователя. Тут вам поможет логика. Она, несомненно, вам нужна. Опять же – очень важно не только понимать наличные программы – в основном скрипты, но и убедиться в их безвредности. В интернете вы можете видеть массу примеров таких программ. То что они умещаются на двух строках ничего не говорит об их безобидности. Потерю нормальной работы надо отличать от ненастроенности чего-либо пока вы этого не настроили, пока у вас не готова рабочая среда.
- Проблемы с сетями, с передачей данных или программ. Надо различать плохие условия выхода в интернет, плохую связь в локальной сети, и проблемы от ненастроенности ОС или отдельных программ. Также оборудование обладает разной чувствительностью, и изменение скорости передачи нормально. Практически многим приходится тратить время и нервы на настройку. В худшем случае человек пользуется непонятным для него оборудованием, и думает что всё хорошо, но защиты нет.
- Проблемы из-за отсутствия защиты данных, получения вирусов и вторжений извне. В худшем случае они приводят к краже или стиранию информации, и потере управляемости.
- Потеря конфиденциальности и секретности. Беспечному отношению к сведениям для служебного пользования нет места. Это, в частности, значит, что нельзя дать место для шпионажа. Если вы берёте работу на дом то об этом должен знать начальник, не скрывайте этого. Ваши родственники ничего не должны знать о вашей работе. Любая передача конфиденциальных данных выполняется через зашифрованный канал.
- Кажущиеся проблемы – прежде всего от непонимания ситуации.
- Неожиданные, и слишком большие расходы, обрыв работы из-за истощения денег на

счёте.

Требования техники.

Прежде чем покупать технику, убедитесь, что вы обеспечите следующее:

- отсутствие резкой смены температуры, температура около +20 градусов
- неподвижность основания – качающиеся столы и гибкий пол не годятся
- отсутствие пыли, табачного дыма, отсутствие избыточной влажности
- для радиопередающих и радиопринимающих устройств – отсутствие сильно экранирующих стен или крыш
- постоянство и правильность электропитания – без выбросов напряжения

Для обеспечения чистоты требуется очистка техники. Это особенно актуально для любителей прополки с телефоном в руках, просмотра фильмов на сеновале, в пустынях Средней Азии и так далее. Если пыльную юбку цыганки Маши можно просто встряхнуть, то тут дело несколько сложнее. Я пока не знаю как чистить телефоны и планшеты. Проще всего очистить стационарный компьютер. ВАЖНО – это нельзя делать пылесосом так как он несёт заряд статического электричества, которое может «убить» микросхемы. В технике накапливается мелкая, летучая пыль, и вы заинтересованы чтобы она улетела с ветром, а не попала вам в органы дыхания.

Заметим, что бытовые насекомые не видят в компьютерных устройствах места для жизни – может быть из-за их работы.

Особенно уязвимы микрокомпьютеры. Типичный микрокомпьютер продаётся без корпуса. Один рассеянный зевок в его направлении – и вам обеспечена долгая возня, и даже возможна потеря данных.

Для пригодного электропитания иногда просто необходимы дополнительные траты. Это может быть покупка более ёмкого аккумулятора, устройства бесперебойного питания (УБП), и сетевого фильтра. Также для телефона и планшета можно купить отдельный аккумулятор на случай потери тока питания на несколько часов. УБП может быть тяжёлым и дорогим устройством, но важным. Оно обеспечивает технику питанием около 3-5 минут – смотря по потреблению техникой электричества. За это время программы или человек успевают сохранить редактируемые данные, и правильно выключить операционную систему. Эти программы работают благодаря сигналу о потере питания с УБП.

Архитектура компьютерного устройства.

Википедия выражается по поводу этого понятия довольно туманно. Архитектура рассматривается как технический тип, которому соответствуют – и это важно – операционные системы, и, добавим от себя, даже конкретная версия ОС в случае планшета или смартфона. Компьютерное устройство – более широкое понятие чем компьютер в обиходном смысле. Именно компьютер в обычном значении слова в России почти всегда имеет архитектуру AMD64 или 86. Последняя представлена техникой на основе 32-битных микропроцессоров, одним из последних тут был Intel Pentium4, AMD Duron, AMD Athlon. 32-битная архитектура устарела, но сами устройства могут работать. Они даже могут быть полезными. Микрокомпьютеры часто имеют на сегодня архитектуру ARM версии 8, но микрокомпьютеры не всегда стремятся догнать по техническим свойствам обычные компьютеры. Там часто используется адаптированная версия Debian, также стоит обратить внимание на Android.

Смартфоны и планшеты управляются обычно ОС Android, которая адаптирована как к носимой технике, так и стационарным компьютерам, но это разные выпуски ОС. Сами носимые устройства могут иметь разную техническую начинку – как 32-битную, так и 64-битную. Последняя всегда требует больше оперативной памяти для работы программ.

Компьютеры с архитектурой 86 в основном использовали ОС Windows95 – WindowsXP. Эта техника всё ещё обеспечена выпусками таких ОС как Linux и FreeBSD, а также NetBSD. Заметим, что там работает ОС DOS, но это не имеет существенного практического значения. Если ученик имеет технику экзотической архитектуры, то ему, в первую очередь, следует понять где он сможет купить к ней совместимые устройства. NetBSD предоставляет выпуски, предназначенные для множества архитектур.

Техника производства Apple в этой книге не рассматривается так как используется редко, и слишком дорога.

Требования носителей.

Тряска не страшна телефонам и планшетами – типичный планшет не имеет жёсткого диска. Напротив, компьютеры с жёстким диском (HDD) нельзя двигать и трясти во время их работы. Флэш-носители – в том числе диски SSD не боятся сотрясений.

Лазерные диски должны храниться без трения о твёрдые поверхности, вне солнечного освещения и повышенной влажности. На диски нельзя приклеивать наклейки с рабочей стороны. Также наклейки для противоположной стороне могут мешать в отдельных случаях – например, увеличивать толщину стопки дисков. Стопка дисков в современном мире – это не анекдот, а, скорее, норма. Хотя в конторах дешевле пользоваться стримером. Всегда убедитесь, что вставляемый диск чист, и сотрите отпечатки пальцев чистой, неабразивной материей. Писать на лазерном диске можно лишь специальным фломастером. Не допускайте падений дисков. Они могут перестать читаться.

Все носители реально имеют срок работы и износ. Это – важная причина резервирования (см ниже). Для лазерных дисков срок хранения данных – в том числе программ и операционных систем – будет разным в зависимости от их качества. В течение 2000-х годов качество улучшалось, но до этого оно иногда было явно низким. Поэтому старые данные нужно переписать на новые носители. Никогда не используйте немаркированные или неизвестно чьего производства диски. Наоборот, покупайте по возможности диски Verbatim или TDK – особенно для ценных данных или долгого хранения.

Флэш-носители имеют ограниченный ресурс актов записи, и есть такое мнение что современные SSD-диски чаще неожиданно выходят из строя. Возможно, эта проблема решится с прогрессом техники. Получается, что лучшее место этих дисков – в системах, где надёжность хранения не очень важна, а скорость очень желательна. Первая мысль об этом – про «карман», кэш – диск для копий часто используемых данных. Такой диск более вероятен на работе чем дома. Заметьте – это не означает что всякие данные не важны. Это, определённо, не так. Для увеличения до приемлемого уровня надёжности на работе требуется RAID или что-то вроде него. Сервисы VPS\VDS часто рекламируют себя как использующие «быстрые SSD диски», но важно помнить, что это в том числе борьба за клиентов, и, во-вторых, вы сами управляете ОС и программами. Я не уверен что ваш сервис сам делает копии ценных данных, или делает их хорошо.

Называемые в обиходе «флэшками», флэш-брелки в опыте автора выглядят плохо – несколько раз выходили из строя, а это означает потерю как минимум *читаемости* данных. Особенность этих носителей в том что они содержат контроллер, который сам может ломаться. Возможно, лучше заменить их на карты памяти и картридер. Покупайте только брелки популярных фирм.

Наиболее популярным средством оценки устаревания жёсткого диска являются такие программы как **Victoria**, **MHDD**, **smartmontools** и другие. Есть программы для восстановления данных для других носителей.

Требования сетей.

Наверное, распространённой ошибкой при создании домашней сети является неправильное расположение устройств. Даже тонкая стена может сильно испортить беспроводной сигнал если он идёт в ней по диагонали. Это обстоятельство означает что нельзя переставлять роутер или точку доступа.

Наиболее популярными сетями на сегодня являются основанные на протоколе IP. Он многократно описан с разной степенью понятности и практичности текста. Все вопросы распределения данных, и вопросы фильтрации данных с помощью брандмауэра должны быть корректно разрешены. Иначе даже безупречная в *техническом* смысле сеть не будет работать.

Но все эти трудности могут выглядеть чуть ли не мечтой по сравнению с мучениями безграмотного ученика когда он настраивает свою технику и программную среду для первого выхода в интернет. Это особенно верно для негородской среды.

При всех проблемах с компьютерами всех видов вам на помощь приходит даже дешёвый телефон. С его помощью – возможно, после установки на него браузера – вы часто сможете найти массу текстов любой тематики.

Требования программ и операционных систем.

Это – не самая страшная тема. Требования (requirements) ОС обычно чётко написаны либо на коробке с диском, либо на сайте ОС. То же верно с программами. Но в среде UNIX всё чуть сложнее. Так, например, вы легко найдёте в интернете требования к аппаратуре Debian, но не найдёте требований программ к нему (приложений). Сами приложения различаются в своих требованиях.

В частности, надо понимать, что использование технологии CUDA, используемой во многих дорогих видеокартах, предъявляет свои требования к аппаратуре. Программы с таких хранилищ как **github** вообще могут предоставляться без описания требований к аппаратуре. Но в наше время, когда старый компьютер – всё ещё пригодный для многих работ – стоит 1000-2000 рублей, это не очень важно. Вычислительная мощность, и объём памяти обычно достаточно велики – у классических компьютеров. К микрокомпьютерам, неттопам, планшетам это не всегда относится.

А в общем дела задают программы, они задают ОС, и они вместе – аппаратуру.

Программа требует доступности данных. Это, прежде всего, данные, которые явно обрабатываются программой. Кроме того, это могут быть некие файлы с параметрами работы и так далее. Конкретика определяется программой. Это – ВАЖНО – означает, что данные, которые находятся на съёмном носителе, или в интернете, или в локальной сети – тоже должны быть доступны. Подключённый носитель нельзя отключить пока используются его данные, выход в локальную сеть и интернет нельзя оборвать в любое время.

Программы требуют достаточно места там куда они пишут данные. Обычно это не проблема.

Программы требуют возможности записи и чтения данных, которые регулируются правами на операции с файлами и каталогами.

Глава 15. Отличия между операционными системами Windows и UNIX

Можно было бы рассматривать больше семейств операционных систем, но именно эти практически важны если рассматривать классические, важные работы, могущие быть формой платной деятельности. Эти два семейства систем сейчас конкурируют между собой, и эта конкуренция будет обостряться так как Россия уходит от использования Windows по мере возможности. Её место займут потомки UNIX, а именно, прежде всего, Linux. И те, и другие работают в основном на настольных компьютерах и ноутбуках, и создаются для них. И те, и другие прекрасно приспособлены к работе с интернетом и локальной сетью, оба семейства неплохо (или отлично) обеспечены оборудованием самого разного назначения.

Но при этом есть и большие отличия.

- Первое, что приходит в голову, это отличия в ожидаемых личных свойствах пользователя. Если Windows сделана для тех, кому надо чтобы всё было быстро и не требовало чтения руководств (хотя совсем не читать не получится), то потомки UNIX -- для людей с запасом времени, и конструктивным мышлением. И терпением, готовностью к образованию. Что же мы получаем за эти труды? Важное качество, но нужное не всегда -- гибкость. И ещё огромную возможность автоматизации без необходимости работать программистом на глубоком уровне. На поверхностном уровне достаточно, хотя это и правда написание текстовых программ -- как правило, размером не более полстраницы, а часто гораздо короче.
- Отличие второе -- потребность в настройке. Лично я не очень понимаю авторов таких систем как Debian, а тем более FreeBSD, которые могли бы дать документацию *прежде* установки системы или работы с ней, но до сих пор не сделали этого. Вы можете зайти на соответствующий сайт, и прочесть руководство. Но давно пора положить его прямо у вас перед носом -- на установочный диск, и автоматически выводить на экран прежде, чем вы что-то сделаете.
- Отличие третье -- организованное хранение приложений. Если с Windows вам надо часто искать программу, то с UNIX вы подключаете хранилища, и обычно они содержат огромный процент программ, вообще работающих с вашей системой. Отсюда простота получения всех этих программ почти одной командой. Но тут же и трудность

получения программ так как нужен выход в интернет. Пока, вроде бы, в России нельзя легко купить диск с этими программами, но это может появиться очень скоро.

- Отличие четвёртое -- в UNIX на каждом шагу одни программы требуют других, и их установка выполняется не вручную, а с обращением к базе данных пакетов.
- Отличие пятое -- UNIX иногда поначалу выглядит очень старомодно и отталкивает любопытного человека, уже готового на него посмотреть и попользоваться. Это -- существенный, и необоснованный недостаток UNIX.
- Отличие шестое -- в России по UNIX мало литературы в любом виде. Большинство документов -- на английском языке. Я призываю людей писать как можно больше, и обозначать дату так как информация на вес золота, и ситуация меняется.
- Отличие седьмое -- пока я не видел провайдера, официально поддерживающего UNIX. Реально это не большая проблема, но факт.
- Отличие восьмое -- если Windows требует сравнительно нового компьютера, то UNIX существует в том числе для очень старой техники.
- Отличие девятое -- большинство ОС UNIX бесплатны. Это особенно важно для бедных людей.
- Отличие десятое -- в количестве вирусных программ и качестве защиты самой операционной системы от них. Тут UNIX пока выглядит гораздо лучше Windows.

«Мудрость» UNIX

Один из авторов известного руководства употребил это понятие без кавычек. Это важная особенность. Что, программы мудры? Нет. И не пытаются такими быть. Суть не в этом. Авторы ранних версий этих систем мудро создали программы, разделили между ними функции, и создали возможности. Первоначально эти возможности могут показаться лишними. Эти качества программ сохранились и продолжатся в будущем, хотя эти программы сейчас как бы заслонены более свежими разработками, которые – как я думаю – делались иначе. В UNIX часто вам нужна не одна программа, а несколько программ. Для любителей всё делать одним нажатием это плохо. Но хорошо то что ученик получает гибкость, нечто вроде конструктора. Конкретно я имею в виду командную оболочку, и многие консольные программы. Всё это рассчитано на «инженерный» подход к работе, когда вы поступаете рационально, и понимаете детально нужные изменения данных.

Глава 16. Техника и покупка оборудования. Сеанс работы компьютера.

Предполагаю, что некоторые читатели считают, что «надо просто щёлкать», или что-то нажимать. Это мнение может происходить от того что есть люди, которые за них разобрались со всеми предыдущими вопросами – техники, установки программ и так далее. Если вы используете компьютерное устройство дома, или, тем более, хотите его купить, а также изменить технически, то вопросы техники вам надо знать. Также вам надо знать вопросы совместимости если вы хотите добавить некое другое устройство – непосредственно или опосредованно.

С другой стороны, некоторые, наверное, считают, что надо знать как работает техника. Я – не инженер-электронщик, и мне, как и вам, не надо им быть. Возможно, вас раньше учили тому как работает техника. Это считалось нужным для управления ей. В случае с компьютером это в общем не так.

Оба мнения ошибочны.

Технику надо, скорее, не знать, а понимать. В науке есть понятие чёрного ящика. Это устройство или всё что угодно, что обрабатывает, изменяет то что в него входит, и выдаёт его в изменённом виде. Это – примерно то понимание, которое вам нужно. Содержимое чёрного ящика не важно, и его не надо знать. Оборудование обладает важными для вас свойствами – при том, что вы можете не понимать его работы. Компьютерное и сетевое оборудование созданы не для специалистов, а для массового пользователя. Поэтому обычно вы сталкиваетесь, например, с инструкцией к принтеру, которая написана достаточно понятно если вы обладаете компьютерной грамотностью. Заметим, что сетевые устройства скорее относятся к администратору, и не надо ожидать что их инструкции будет легко понять.

Но тут надо пояснить, что даже проведение границы между сетевыми и персональными устройствами может быть трудно. Так, например, сетевой принтер по сути есть персональное устройство. Но чтобы работать именно как *сетевой* принтер, оно должно быть частью локальной сети. Тогда задание печати можно посылать на принтер с нескольких компьютеров.

В наше время распространены многофункциональные устройства. Обычно такое устройство объединяет в себе принтер, сканер и копир (то есть «ксерокс»).

Каждый покупатель должен знать что нельзя просто купить наиболее дешёвое устройство, так же как наиболее дорогое, или наиболее популярное. Ни одна книга не рассматривает всё множество устройств. Всегда надо тщательно разбираться с каждой единицей техники. Может потребоваться литература. Инструкции часто есть в интернете. Они бесплатны.

Заметим, что сводка характеристик техники может быть неполной и грешить формализмом. Легко упустить такое свойство работы оборудования, как время или сложность освоения, и скорость работы. Медленно работающая техника означает низкую производительность труда.

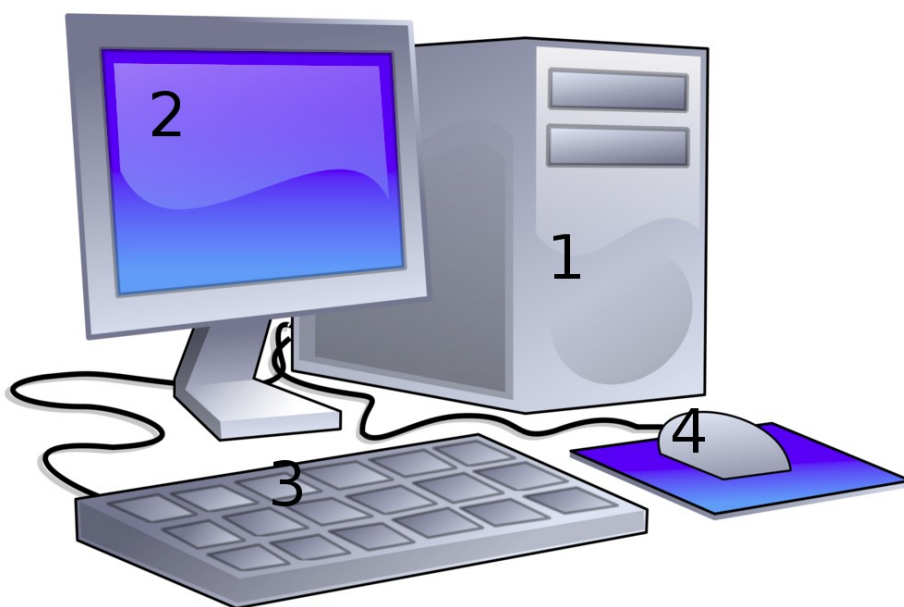
Техника часто имеет встроенную программную среду. Иногда её можно обновить, или даже сменить. Также типично, что для устройства нужен драйвер, то есть программный адаптер под конкретную ОС. Драйвер также можно сменить или обновить. Всегда убедитесь, что устройство может работать с вашей ОС. О работе с UNIX может быть не написано в документации. Но написано в интернете. То есть бесплатная ОС может не быть официально поддерживающей устройство, и техподдержка может не обсуждать вопросы работы с ней.

Рассмотрим более подробно технику.

Компьютерное устройство.

В этой книге я пытаюсь обозначить разные устройства – обладая опытом работы почти исключительно с «классическим» компьютером. Это делается намеренно – чтобы ученик имел выбор, и мог понимать что может входить в локальную сеть. Тут под компьютерным устройством понимается компьютер, планшет, и, можно сказать, развитый мобильный телефон, а также – хотя, возможно, не всегда – роутер или точка доступа с полноценной, не штатной операционной системой, и микрокомпьютер. Такое определение существует для того чтобы подчеркнуть вычислительную, полезную роль этих устройств для инфосистем, и населения.

Наиболее описанным, наиболее рано возникшим устройством является личный компьютер – в виде стационарного оборудования, или ноутбука, или неттопа. Сам этот тип устройств возник намного позже компьютеров вообще. Рассмотрим стационарный компьютер. Он обычно состоит из системного блока (1), монитора (2), клавиатуры (3) и «мыши» (4).



Всё остальное что может быть к нему подключено, может быть не обязательно. При этом типично, что офисный компьютер, или компьютер для использования на работе, не имеет встроенного читателя лазерных дисков. В таком случае многим потребуется подключить его как встраиваемый или переносной узел. На работе делать этого не рекомендуется так как это может вызвать раздражение начальства. Лазерные диски нарушают выстроенную аппаратную систему, обеспечивающую достаточную независимость и надёжность без вмешательства рядового работника в сферу работы администратора. Напротив, в домашней эксплуатации читатель лазерных дисков очень полезен – хотя несёт ту же опасность – в основном из-за подростков. Старые компьютеры оснащены читателем дискет. Его не требуется убирать из-за смены программной среды или деталей. Он на сегодня обычно потерял свою полезность так как его заменили флэш-брелки, и, можно сказать, локальная сеть. Это так потому что локальная сеть стала не всегда используемым, но реальным источником программ – особенно на работе. Некоторые компьютеры обладают читателем карт памяти. Они в наше время в своей роли сходны с лазерными дисками. Стационарный компьютер легче и дешевле всего изменять, заменяя детали. Также его достаточно легко физически очищать.

Ноутбук в своём применении мало отличается от стационарного компьютера – исходя из самых общих характеристик. Но конкретная модель может обладать лишь базовыми возможностями из-за малой вычислительной мощности. То есть типичный стационарный компьютер в общем мощнее чем ноутбук. С прикладной стороны у ноутбука есть важные отличия – он переносим, и не боится отключения тока питания – имея аккумулятор. Он является полноценным рабочим местом, обладая встроенной клавиатурой, и встроенным монитором.



Обычно ноутбук имеет размер экрана более площади стандартного листа бумаги. Такой ноутбук можно назвать наиболее универсальным. Но обычно он довольно тяжёл, и неудобен в переноске. Более лёгкие модели могут быть намного более дорогими из-за этого. Мелкие модели называют нетбуками так как они направлены в основном на действия с интернетом, без запроса на большую вычислительную мощность. Ноутбуки любого типа достаточно трудно менять аппаратно, и трудно чистить. Продолжительность работы от аккумулятора зависит, в том числе, от занятости компьютера. ВАЖНО – захлопывание ноутбука может вызвать обрыв или перерыв в его работе.

Неттопы – это попытка дать пользователю уменьшенный компьютер без всех подключаемых к нему устройств.



Часто монитор не требуется менять. Неттоп получается недорогим, но сам по себе он бесполезен. Если вы используете телевизор с функцией монитора, то неттоп становится осмысленной покупкой. Его легко переносить.

Микрокомпьютер – это попытка дать вам дешёвый компьютер без подключаемых устройств. Часто он продаётся как отдельная плата.



В таком виде, сам по себе, он бесполезен. Его вычислительная мощность в случае старых моделей, и вообще некоторых, невелика. Это – интересное устройство для бедного покупателя, но если он согласен на ограничения конструкции, и особенности этого устройства. Применение такого компьютера – тема обширных размышлений так как даже заурядные дела могут происходить иначе, и с трудом, медленно. Таких компьютеров много. И всё же я считаю, что микрокомпьютер может реально улучшить качество жизни на селе. Он имеет преимущества перед планшетом и телефоном из-за своего сходства с «классическим» компьютером. Это так в том числе из-за его ОС. ОС для таких компьютеров много. В частности это легковесная DietPi, которая может быть опробована в виртуальной машине на обычном компьютере.

Планшетные компьютеры, то есть планшеты, совершили революцию в оборудовании, предложив себя как доступное и понятное устройство, которое обычно сильно отличается от всего что рассмотрено выше.



Простое определение такого планшета – это ваш помощник, но не всегда средство серьёзной работы. Типично, что планшет умеет работать с сетью Wi-Fi, но не имеет многих гнёзд, которые есть в стационарном компьютере. Объём встроенной постоянной памяти обычно невелик, что может привести к затруднениям. Есть аккумулятор. Основная особенность планшета – в управлении касаниями, и ОС. Планшет может показаться хорошим устройством для ученика. Но это не гарантия того, что он с ним справится. Точнее – часто планшет не может конкурировать с компьютером из-за того что он так же отличается от него как высшая математика от арифметики. Такая оценка планшета сильно зависит от установленных там программ, и они определяют возможность выполнения работ. Планшет обычно недорог.

Мобильные телефоны могут быть самыми разными. Основной интерес представляют модели, очень похожие на планшет.

Точки доступа и роутеры по идее вообще не должны обсуждаться как компьютерные устройства. Тем не менее о них надо помнить. Их использование требует большого терпения и догадливости. Без опытного помощника может быть трудно. Возможности этих устройств могут быть невелики, и скорее нужны администраторам.

Подключаемые устройства.

Я не смогу хорошо объяснить к чему они подключаются. Так как это достаточно сложная тема. Можно сказать – к гнёздам, которые правильно называют портами. Об этом нужна отдельная глава.

Подключаемые устройства не есть носители. Это также не сетевые линии связи или устройства.

Принтер.

Основным подключаемым устройством всё же является принтер. Его основное назначение – печать на бумагу. Принтер – интересная покупка в случае невежественных учеников компьютерной науки. С бумаги гораздо привычнее читать. Но качество печати должно быть не самым плохим. Так можно создать целую библиотеку. Но печать должна быть не слишком дорогой. Это означает не только дешёвые чернила или порошок, но и дешёвый барабан – он изнашивается. Каждый принтер имеет максимальный размер листа или ширину ленты, и

скорость выдачи одного листа – которая может зависеть от цветности печати. Принтер не занимается делением публикации на листы или другими интеллектуальными функциями. Лучший способ проверить публикацию перед печатью – посмотреть её в формате PDF. Всегда надо прежде убедиться что печать будет правильной. Также каждый пользователь должен уметь выполнять дополнительные операции – очистку головок принтера и так далее. См рисунок.



Сканер.

Сканер – это устройство для получения «снимков» с прозрачных и\или непрозрачных плоских оригиналов. Это не фотоаппарат. Подвижная читающая головка типичного сканера резко «видит» лишь то что прижато к стеклу над ней. Обычно это лист бумаги. Сканер – это как бы «принтер наоборот». Он позволяет вводить в память компьютера текст и иллюстрации. Но чтобы текст был редактируемым, требуется *распознавание* текста. Оно не делается сканером, а выполняется программой, обрабатывающей создаваемый графический файл. Большинство сканеров – если не все – могут делать снимок с разной скоростью. Чем выше разрешение снимка тем ниже скорость. Пока вам надо делать по два снимка в день это не важно. Но когда потребуется делать сотни их, то годится лишь быстрый сканер. Он намного дороже. Конкурентом сканера является фотоаппарат. См рисунок.



Монитор.

Монитор, то есть устройство для отображения объектов вашей работы, и средств этой работы, и многого другого – достаточно сложное устройство с точки зрения покупателя – не инженера. Обычно пользователю не нужен монитор экстра-класса, используемый для точной цветопередачи и особо аккуратного отображения частей экранной картинке. В качестве альтернативы выступает телевизор с функцией монитора.

Камера.

Вэб-камера. Такая камера не даст хорошего изображения, и обычно используется для видеосвязи.

Устройство для электропитания.

Устройства для обеспечения питания. Это либо устройство бесперебойного питания, либо сетевой фильтр. Первое часто включает второе, и обеспечивает работу техники после потери питания в течение нескольких минут. При потере питания оно подаёт звуковой сигнал. Всегда желательно обеспечить автоматическое, правильное выключение техники при потере питания.

Звуковые устройства.

Наушники или колонки, микрофон.

Мышь.

Мышь является и указателем на элемент экранной картинке, и средством управления программами. Мышь на сегодня нужна почти всегда. Редкое исключение – текстовые среды – в первую очередь консоль серверного компьютера или используемая для управления контейнером. При покупке компьютера – даже если он обладает площадкой для имитации мыши – мышь нужна. Дешёвые модели мышей работают год-два. На сегодня обычно мышь подключается к компьютеру через Bluetooth-адаптер к порту USB, а мышь не связана с компьютером проводом. В случае планшета или смартфона можно использовать провод-переходник для подключения через OTG, но не все такие устройства это умеют. Микрокомпьютер имеет один-два или больше портов USB, и может потребоваться

разветвитель так как устройств больше чем гнезд. Вид мыши.



Старые модели мышей не имели колеса между кнопками. Колесо позволяет не двигая мышью листать документы. Также колесо является третьей кнопкой. Показанная мышь – обычная, трёхкнопочная. Таких мышей – большинство. Обычному пользователю не нужна игровая или специальная мышь. Экзотическая мышь может не работать в среде свободно распространяемой ОС. Беспроводные модели требуют батареек.

Практика использования мыши – это обычно действия в оконной среде. Мышь нужно брать так чтобы кнопки были под пальцами, «смотрели» от вас. Мышь гоняет свой курсор – не курсор текста – по экрану. Курсор мыши обычно имеет вид стрелки, и не может выйти за пределы экранного изображения. Для движения курсора мышь гоняют по столу. Мышь должна быть на столе, не отрываться от него. Следите за руками, не делайте лишних нажатий на кнопки.

Наиболее частое нажатие – однократное на левую кнопку. Другие нажатия могут явно обозначаться в текстах. Это двукратное на левой кнопке, однократное на правой и на колесе. Колесо при этом не должно вращаться. Эффект от нажатий может быть разным смотря по среде где используется мышь. Вы можете экспериментировать, но помня что это не игра, и нажатия дают свой эффект.

При обработке текста курсор меняет вид, и работают некоторые способы которые могут существенно ускорить и облегчить работу. Например, нажим с задержкой левой кнопки и протягивание курсора по тексту выделяет текст. Таким образом вы можете указать программе на блок текста с которым вы хотите что-то сделать.

Клавиатура.

Клавиатура также необходима, причём надо стараться покупать клавиатуру с крупными кнопками и буквами на кнопках. Компьютеры без указания букв русского алфавита потребуют приклеивания наклеек с буквами. Общая рекомендация – покупать не специальную, а обычную клавиатуру с классическим положением и набором кнопок. Вид клавиатуры.



Клавиатура может использоваться для управления программами. Она теснейшим образом связана с набором символов, или языками, поддерживаемыми ОС. Свободно распространяемые ОС обычно интернациональны, и язык определяется настройкой, а не клавиатурой.

Подключение к компьютеру обычно делается через порт USB. Беспроводные модели требуют батареек.

Клавиатура позволяет переключаться между языками вводимого текста. Часто для этого назначены клавиши левый **Alt** и левый **Shift**, которые надо нажимать одновременно. В оконной среде, о которой написано ниже, легко узнать какие клавиши нажимать; там же легко сменить эти клавиши. В консоли назначение клавиш сменить труднее, так же как труднее добавить язык, поэтому важно сделать это при установке операционной системы.

Настройка языка ввода в консоли Debian может быть достаточно простой. Вы можете отредактировать файл `/etc/default/keyboard` вручную, например для такого вида:

```
XKBMODEL="pc105"  
XKBLayout="us,ru"  
XKBVARIANT=""  
XKBOPTIONS="grp:alt_shift_toggle"  
BACKSPACE="guess"
```

Переменная `XKBMODEL` содержит модель клавиатуры;

Переменная `XKBLayout` содержит список используемых раскладок;

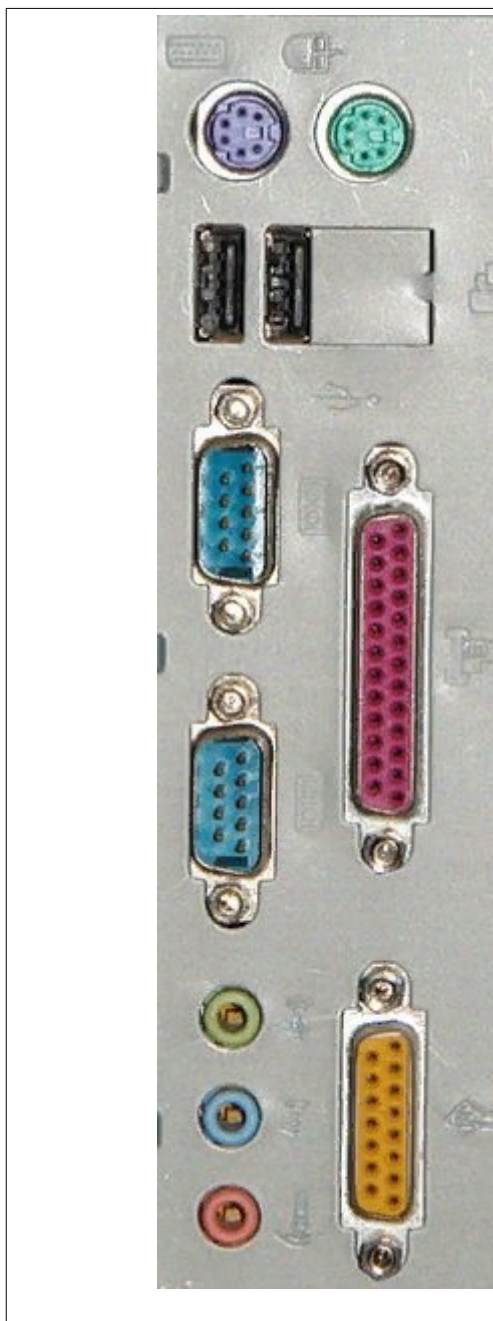
"`grp:alt_shift_toggle`" задает переключение раскладок комбинацией клавиш (`<Alt>+<Shift>`).

Простейший способ заставить работать новые параметры – перезагрузить компьютер.

(по тексту документации Debian)

Порты компьютера.

В данном случае это не порты программ, например 631 порт для CUPS.



Аппаратные порты компьютера (сверху вниз и слева направо)

PS/2-порты клавиатуры и мыши, 2 USB-порта, 2 COM-порта, LPT-порт, 3 звуковые порта джек 3,5 мм, игровой порт

(по материалам Википедии)

Детали компьютерного устройства.

Эта тема – достаточно обманчивая по той причине, что техника достаточно быстро меняется. Кроме того, моделей деталей очень много. Рассматривать можно не модели, а виды деталей.

Простейшим устройством является микрокомпьютер. Он обычно очень прост на вид. Это всего одна плата.

В планшетах носителем постоянной памяти является флэш-память, микропроцессор часто имеет архитектуру ARM, а сетевые функции обычно реализованы с помощью сети Wi-Fi.

То же верно про многие развитые телефоны.

Ноутбуки во многом повторяют стационарные компьютеры – с той разницей, что там требуются более мелкие детали.

Роутеры и точки доступа характерны очень скромными техническими возможностями. Они не рекомендуются к использованию учениками в качестве замены других, более традиционных компьютерных устройств.

Проще всего рассказывать про детали стационарного компьютера. Что это за детали?

- Корпус. Корпус часто продаётся вместе с блоком питания. Корпус должен соответствовать прежде всего материнской плате. Иначе они будут несовместимы.
- Блок питания. Блок питания преобразует входящее в него напряжение в пригодное для низковольтных частей компьютера. Блок питания имеет мощность, которая должна быть больше чем требуется для самой большой вычислительной работы.
- Материнская плата. Это основная, обычно самая большая плата, с которой связаны главные функции компьютера – обработка, передача, и хранение информации.
- Микропроцессор. Это основная микросхема. То есть деталь, в которой «спрятаны» многие аналоги обычных радиодеталей. В микропроцессоре происходит обработка информации.
- Кулер, то есть устройство для отвода лишнего тепла от микропроцессора.
- Оперативная память. Место для временного хранения данных и программ – пока они используются.
- Жёсткий диск – всё ещё основной носитель данных и программ. Устройство постоянной памяти.
- Видеоплата. Устройство, формирующее экранное изображение, и, иногда, обладающее неким подобием микропроцессора. Обычно видеоплата с ускорителем (GPU) не требуется. Она намного дороже обычной.
- Читатель лазерных дисков. Он же, обычно, умеет писать на них.
- Сетевая плата. Обычно под этим понимается плата для локальной сети или проводного соединения с неким сетевым устройством. Обычно такая плата имеет разъём для провода Ethernet.
- Звуковая плата. Занята выдачей, получением, и генерацией звука.
- Плата для Wi-Fi. Может иметь небольшую антенну.

Заметим, что некоторые из перечисленных плат могут быть встроены в материнскую плату.

Сетевые устройства.

Надо внести ясность в понятие сетевого устройства. Устройство, доступное по сети само по себе не обязательно считается сетевым. В более точном, используемом тут смысле сетевое устройство есть устройство, которое по сути, по своей функции сетевое – бессмысленное без сети. Заметим, что в качестве такового иногда может использоваться компьютер, который сам не является сетевым устройством так как прекрасно работает без всяких сетей, но возможности пользователя при этом сокращены.

Какое для нас имеют значение сетевые устройства? Большое. Благодаря им мы пользуемся сетями, то есть, просто говоря, имеем доступ к данным, программам, и управлению техникой, доступной по сети. Что это за сетевые устройства?

- Встроенный аппаратный модуль для работы с сетью Wi-Fi в мобильном телефоне или планшете. Никакой заботой для пользователя не является. Всегда работает при штатной ОС и своей исправности. Подключение к сети делается просто – часто касанием соответствующего значка на экране. При этом надо понимать что *закрытая* сеть требует ввода пароля и своего имени для подключения – даже если аппаратура позволяет работать с сетью.
- Встроенный модем для выхода в интернет там же. Тут ситуация немного сложнее. Передача данных – в том числе программ – через такой модем называется передачей мобильных данных. Провайдер телефонной связи обычно предоставляет такую услугу, но гарантировать это нельзя. Стоимость услуги надо отличать от стоимости объёма переданных данных. Идентификация устройства, насколько я могу судить, происходит по номеру телефона. Скорость передачи данных часто ниже чем в случае с Wi-Fi. Хотя фактически данные идут через беспроводную сеть, но обычно её так не называют, а скорее называют мобильной сетью. Работа встроенного модема не влияет на возможность делать звонки и получать SMS. Последнее исторически было раньше появления телефонов со встроенным модемом. Заявленная скорость передачи данных, обозначаемая через поколение сетевого оборудования – в наше время часто 3G или 4G – не реальная скорость, а расчётный максимум. Управление встроенным модемом определяется моделью телефона. Смотрите его инструкцию. Важно, что часто

встроенный модем телефона можно использовать как отдельный модем для доступа в интернет с *компьютера*. Скорее всего телефонный модем будет рассматриваться ОС компьютера как заурядный отдельный модем, подключаемый к компьютеру проводом. Фактически вы подключаете телефон проводом. Обычно настройка такого подключения к интернету либо не требуется, либо проста – по крайней мере при дружественной оконной среде.

- Отдельный модем для компьютера, работающий посредством SIM-карты. Ситуация похожа на описанную выше. Модем часто имеет гнезда для подключения антенны, а антенна может увеличить скорость передачи данных. Такой модем теоретически может работать с планшетом, и его надо отличать от модема для коммутируемого доступа через телефонную сеть. Последний вид модема устарел, и обеспечивает лишь очень низкую скорость. Убедитесь, что модем будет работать с вашей ОС. См рисунок.



- DSL-модем для подключения компьютера к интернету через городскую телефонную сеть с цифровыми АТС. Такое соединение с интернетом – достаточно быстрое, и скорость не пропадает, хотя зависит от техники, с которой вы взаимодействуете. Подробности – у провайдера. Городской способ выхода в интернет. Управлять самим модемом, скорее всего, не требуется. Нужно совершать акт выхода в интернет, и отключения от него. Такой модем надо отличать от модема для коммутируемого доступа через телефонную сеть. Последний вид модема устарел, и обеспечивает лишь очень низкую скорость. Убедитесь что модем будет работать с вашей ОС. См рисунок.



- Адаптер Wi-Fi для компьютера. Может потребоваться настройка. Интересен для домашних пользователей семейной сети Wi-Fi – то есть квартирной, а в деревне – домовой сети. Прежде чем покупать адаптер, убедитесь, что он будет работать на вашей ОС. То же верно про Wi-Fi-плату. Адаптер является устройством для подключения к сети Wi-Fi. Благодаря ему вы можете пользоваться выходом в интернет – если сама сеть это позволяет. См рисунок.



- Сетевая плата – встроенная в материнскую плату или отдельная. Отдельная плата дешёва. Самое простое средство получения быстрого канала в интернет если есть выделенная – ваша личная – линия связи. Городской тип связи и выхода в интернет, или способ соединения с локальной сетью.
- Устройство для передачи данных через сеть питания. Я не уверен что оно само позволяет выход в интернет. Убедитесь что оно работает с вашей ОС. См рисунок.



- Точки доступа и роутеры (маршрутизаторы). В быту часто используются для выхода в интернет через сеть Wi-Fi, как и через провод и сетевую плату. Иногда имеют гнездо, позволяющее выходить в интернет через модем. См рисунок.



В данном случае модель с наружными антеннами.

- Устройства для усиления сигнала. Интересны для создания увеличенной беспроводной сети, или ускорения связи модема.
- Спутниковый приёмник данных интернета, или приёмопередатчик. Имеет смысл лишь в глухих местах – где нет альтернативы. Дорогое оборудование при поставке фирмой-провайдером. Подробности – у фирм-продавцов. Стоит поискать сведения о более дешёвых способах выхода в интернет через спутник.

Популярное, пользовательское оборудование надо отличать от техники для фирм связи.

Часто роутер и точка доступа является достаточно непонятным и сложным устройством. Тем не менее надо в нём разобраться. Надо понимать, что программная среда этих устройств также нуждается в обновлении.

Подробности о настройке ОС – в инструкциях к оборудованию, и в интернете. Типично, что в случае с Windows прилагается простая программа для действий с отдельным модемом. В случае с другими ОС может быть надо больше знания и времени. Хотя масса людей выходят в интернет с бесплатных, альтернативных ОС.

Использование спутникового оборудования имеет ту трудность что невежественный человек может подумать, что его достаточно подключить.

Покупка оборудования.

Не рекомендуется покупать у частного продавца, которому нет полного доверия. Также обычно не требуется покупать в дорогом магазине. Даже купленный там товар не застрахован от поломок. Всегда не просто покупайте, а выбирайте товар. Помните, что задача магазина – продать, а ваша задача – правильно выбрать. Нет уверенности что вы сможете вернуть купленное. Не спешите. Всегда рассматривайте покупку в свете своих целей, а не чужого мнения, или моды. Оборудование задаётся обстоятельствами, в первую очередь его возможностями, но в связи с работой программ. Ваши задачи определяют программы, программы имеют требования к технике. Поэтому очень рискованно покупать без знания программ, и уверенности, что ваша ОС способна работать с техникой. Это особенно верно про бесплатные ОС. Обычно ОС продаётся в компьютере, планшете и телефоне. В компьютере вы можете её заменить, но вам никто не вернёт деньги за прошлую ОС. У вас есть экономический интерес купить компьютер без ОС, но его трудно найти, легче заказать сборку. Но сами работы по сборке платные. Вы можете сами собрать компьютер, но не ноутбук или неттоп, а стационарный. Также вы можете менять его состав. В любом случае нужно иметь совместимые

детали. Всегда берите гарантийный талон.

ВАЖНО – техника, принесённая с улицы в тёплое помещение может покрыться конденсатом. В таком виде её нельзя включать. Дайте ей просохнуть и прогреться.

Компьютерная мебель может быть бессмысленной покупкой. Вполне подходит обычный устойчивый стол. Не рекомендуется ставить на одну мебель компьютер и трясущийся принтер.

Обычно техника рассчитана на евrorозетку. Не используйте ненадёжные переходники и удлинители, не экономьте на них. Если можете – вызовите электрика и обеспечьте реальное заземление в розетках – в случае платной работы это того стоит. При операциях внутри оборудования не носите синтетику – она опасна для микросхем. Иногда может иметь смысл нечто вроде электрогенератора (особенно на селе).

Обычно товар обозначен либо как RETAIL (ритэйл), либо как OEM (оем). В первом случае товар готов к использованию конечным потребителем, во втором – он предназначен для сборки из него технических систем, и обходится дешевле. Это может быть важно хотя бы потому что в первом случае вы получаете программы для техники.

При покупке важно не стесняться своего незнания. Но при этом так же важно не показывать его чтобы вас не ввели в заблуждение. Поэтому нужно читать. И покупать лишь в магазинах, желательно – крупных. Покупка может оказаться более дорогой, но менее рискованной. По мере образования возможность экономии будет расти. Важно понимать что не всегда требуется сменять устройство целиком. Иногда надо заменить деталь. Иногда надо менять не оборудование, а программы.

Сеанс работы компьютерного устройства. Хранитель экрана (screensaver). Режимы энергопотребления. BIOS и EFI.

Сеанс работы компьютерного устройства зависит от самого устройства. Технически это – не более чем обнаружение и работа оборудования. Компьютер не «знает» что им управляет человек. Человека можно идентифицировать косвенно. С точки зрения пользователя сеанс работы состоит из следующего:

- включение устройства
- самопроверка устройства, не требующая наличия ОС
- поиск, и запуск ОС
- вход в среду ОС – для управляемого человеком устройства
- возможность действий в среде ОС
- выключение устройства средствами ОС, или выход и вход другого пользователя, или выключение и последующее включение устройства – *перезагрузка* его

Очень вероятно что благодаря BIOS можно заставить компьютер включаться в определённое время.

Также типично, что есть возможность заставить устройство перезагрузиться независимо от его состояния. При обычном выключении ОС работает, следя за тем чтобы данные не разрушались и не терялись. Перед выключением устройства – ВАЖНО – сохраните все явно открытые вами файлы, убедитесь, что запущенные вами программы отработали. В случае когда пользователь перезагружает устройство независимо от этих явлений, грубо, он может повредить не только данные и работу других пользователей, но и ОС. Поэтому так делать можно лишь в крайнем случае. На стационарных компьютерах есть кнопка **RESET**, которая может быть не надписана, и незаметна. На планшетах и телефонах это также нажатие кнопок. Всегда надо знать как выполнить грубую перезагрузку – не потому что ОС ненадёжна, а на случай её «зависания», то есть отсутствия возможности вашего управления. Зависание надо отличать от замедления из-за большой вычислительной работы или медленной работы техники. Зависнуть может и отдельная программа, что не означает зависание ОС.

Поиск ОС важен потому что может быть запущена не та ОС которая нужна. Вообще говоря, ОС может быть запущена либо с носителя, либо по сети. Второй вариант – редкий, и означает, что некие программы будут запрошены из локальной сети или интернета. Этот способ может быть полезен в отдельных случаях, но его недостаток в том что часто нет хорошего или дешёвого канала в интернет. Получить даже 100 мб за городом проблематично, а обычная ОС больше. Загрузка ОС по сети не равна работе тонкого клиента или сетевой ОС, называемой иногда веб-ориентированной. Загрузка ОС по сети регулируется в программе BIOS, как и поиск носителей с ОС.

Чтобы можно было стартовать ОС с носителя он должен быть загрузочным. В качестве такового могут выступать жёсткие диски, дискеты, лазерные диски, флэш-брелки, карты памяти (которые тоже являются флэш-памятью). Сам носитель не обязан быть загрузочным. Он им может быть. На практике такой носитель может быть нужен в быту, и часто его приходится делать пользователю. Каждый пользователь должен иметь его на случай проблем, хотя не всякий поймёт что именно делать при проблемах. У автора есть опыт старта ОС в основном с лазерного диска. Есть специальные диски для решения проблем. Они обычно не продаются, а создаются самостоятельно из файла-образа (image).

BIOS является программой, записанной в микросхеме на материнской плате. Её работа, и вид определяются не ОС, а инженерами материнской платы. Обычно BIOS доступна при нажатии обозначенной на экране клавиши после включения компьютера. Нажимать надо пока это сообщение видно. Обычно эта программа выглядит просто и загадочно для ученика. Лучше всего иметь под рукой документацию по материнской плате. BIOS, видимо, есть не только на классических компьютерах, но там она может быть недоступна. См рисунок.



Надо понимать что записанная на носителе ОС может быть не адаптирована к наличному оборудованию. Просто взять у знакомых носитель, и с него загрузить компьютер – может быть слишком рискованно. Он может не сработать. При установке ОС она адаптируется к оборудованию.

Для UNIX и современных версий Windows типично что ОС может быть использована более чем одним человеком. UNIX изначально позволял делать это даже одновременно, но тут надо понимать что возможности оборудования не бесконечны. Важно что перезагрузка и выключение компьютера или ОС угрожают работе других людей. Если надо одновременно работать многим людям, то надо сделать их рабочую среду достаточно лёгкой для техники. Таким образом она – среда – станет более быстрой. Самой быстрой средой является консоль.

Надо различать сеанс работы техники, и сеанс работы пользователя в среде ОС. Второе можно сделать без перезапуска ОС. Пользователь может выйти из среды ОС, и зайти в неё затем. Обычно это требует ввода имени пользователя и его пароля.

Если компьютер используется как сервер, то вмешательство в его работу обычно не нужно. Сервер важен, и он не может управляться случайными людьми, или кем угодно. В данном случае имеется в виду не просто некий компьютер предоставляющий сервис, а компьютер,

который делает это сам, программы которого стартуют и работают без ручного запуска и управления. Такой сервер часто не годится для ручной работы, и часто не имеет оконной среды. Для него лучше иметь отдельное закрытое помещение, и – ВАЖНО – систему вентиляции с очисткой воздуха.

Screensaver (скринсейвер) – это программа, и часть оконной среды, которая может блокировать экран, и часто выдаёт на экран бессмысленные изображения. Смысл этой программы в том что она сохраняет экран от лишней нагрузки, и не позволяет посторонним знать что происходит, и вмешиваться в вашу рабочую среду. Скринсейвер регулируется неадминистратором, и может отсутствовать на конкретном компьютере. Для его выключения может быть надо ввести логин и пароль. На планшетах и смартфонах буквально такой программы нет. Это личные устройства, не многопользовательские. Скринсейвер начинает работу при отсутствии действий пользователя. Для убирания его изображений с экрана достаточно подвигать мышь или нажать любую клавишу.

Работу этой программы надо отличать от отключения экрана, а отключение – от затемнения экрана, которое может выполняться скринсейвером, или по прошествии некоторого времени пока пользователь не работает.

Вход и выход из обслуживания операционной системой.

То что вынесено в название главы в обиходе называется входом и выходом из сеанса использования ОС. На серверах вход может делаться редко так как эти компьютеры (или виртуальные машины) выполняют свои функции самостоятельно. Напротив, в ситуации настольной машины с ОС классического типа (не Android) ОС часто настроена так что должна пускать лишь некоторых людей; до входа нельзя управлять программной средой. ОС обычно запрашивает имя пользователя, то есть логин (login, user name) и *соответствующий* пароль. Разумеется, в разных инфосистемах один пользователь может (и должен) иметь разные пароли, и, желательно, разные имена. Всё что тут описано не является способом убрать хранитель экрана. На компьютере даже без оконной среды также потребуется вводить логин и пароль.

Сеанс ручной работы в среде ОС завершается при её выключении, а не только при выходе из сеанса её использования человеком.

Также типично что нужно вводить пароль и логин при выходе компьютера из режима сна (см следующую главу).

Пароли, логины, и группы пользователей формируются администратором. Названия групп, логины и пароли – это строки, которые хранятся ОС так что неадминистратор не может ни изменить, ни узнать чужие логины и пароли.

Энергопотребление и «пробуждение» компьютера.

Классический компьютер может находиться в разных режимах, или состояниях. Лучше всего о них узнавать из описания техники – тем более что она меняется, и бывает разной. Опираясь на свой опыт, обозначу несколько режимов.

- Режим при «разгоне» микропроцессора. Микропроцессор работает на пределе своей производительности. Но это именно его производительность. Она не отменяет влияния программ, которые могут быть не быстрыми. Обычно пользователю не надо использовать этот режим. Выполнить большую работу быстро нереально. Можно попытаться во время этой работы что-либо делать. Но часто эта второстепенная деятельность сильно замедлена более требовательным процессом. Думаю, что есть способы всё же достаточно быстро работать. Так, например, аппаратные запросы виртуальной машины регулируются, и можно даже просто задать приоритет процесса. Заметим, что иногда не надо ждать результата, а оставить компьютер работающим на ночь. Требуется повышенное охлаждение техники.
- Обычный режим. Именно он описан в наиболее доступной документации и литературе. Охлаждающая часть компьютера рассчитана на этот режим при сборке компьютера. Опять же – один процесс может «давить» другой.
- Режим пониженного энергопотребления – возможно, с ограничениями вроде пониженной производительности и так далее.
- Ждущий режим. Электроснабжение необходимо. Работа программ прерывается. ВАЖНО – убедитесь что это не нанесёт вреда ценным данным и не оборвёт нужный

- программный процесс.
- Спящий режим. Электроснабжение не обязательно. Работа программ прерывается. ВАЖНО – убедитесь что это не нанесёт вреда ценным данным и не оборвёт нужный программный процесс.
- Режим когда компьютер способен лишь вернуться в обычный режим. Именно этот режим стоит использовать когда вам может быть нужен удалённый компьютер, но его постоянная работа не нужна. Сетевая плата часто способна принять специальный «магический пакет данных» для «пробуждения» компьютера. Но для этого как минимум компьютер должен иметь публичный (то есть «белый») адрес при доступе к устройству через интернет. Многие части компьютера могут бездействовать.

EFI

Это программная система – не являющаяся ОС – которая работает до селектора режимов или выбора ОС. Она не меняется при смене ОС. При выборе установочного или live-диска старайтесь выбрать приспособленный к наличию этой системы. Тем не менее в моём опыте с Debian мне удалось поставить Debian 10 на компьютер с EFI без использования последней. Для этого необходимо было в BIOS активизировать режим **Legacy**. В перспективе EFI рассматривается как замена BIOS. Так как EFI является набором программ то теоретически её можно заменить, но это может привести к разным казусам с ОС, которая поставлена ранее. Для пользователя наличие или использование EFI может быть незаметно пока он просто использует некую ОС. Сама EFI умеет выполнять примитивные операции, и может работать без работы любой ОС. Это позволяет не тратить электричество и аппаратные ресурсы, например, в условиях работы от батареи. В таком случае потребуется уметь работать с EFI. Знание этой системы не ставится задачей в обычном курсе информатики. Заметьте, что вам нужно знать свойства конкретной EFI на вашей технике. ВАЖНО – режим **Legacy** или подобный может отсутствовать на новой технике. В таком случае стоит поискать в интернете побольше информации. Производитель техники может не отвечать на ваши вопросы об альтернативных ОС. Изменения в BIOS могут всё же позволить поставить и использовать альтернативную ОС. А в общем лучше искать информацию о том где может работать Linux (или что у вас там) до покупки оборудования.

Глава 17. Взаимодействие пользователя и компьютерного устройства.

“Мне кажется, чрезвычайно важно, чтобы мы, занимаясь информатикой, получали радость от общения с компьютером. С самого начала это было громадным удовольствием. Конечно, время от времени встречали заказчики, и через какое-то время мы стали серьёзно относиться к их жалобам. Нам стало казаться, что мы вправду отвечаем за то, чтобы эти машины использовались успешно и безошибочно. Я не думаю, что это так. Я считаю, что мы отвечаем за то, чтобы их тренировать, указывать им новые направления и поддерживать уют в доме. Я надеюсь, что информатика никогда не перестанет быть радостью. Я надеюсь, что мы не превратимся в миссионеров. Не надо чувствовать себя продавцом Библий. Таких в мире и так достаточно. То, что Вы знаете о программировании, могут выучить и другие. Не думайте, что в ваших руках ключ к успешной работе с компьютерами. Что у вас, как я думаю и надеюсь, есть — это разум: способность увидеть в машине больше, чем Вы видели, когда вас впервые к ней подвели, увидеть, что Вы способны сделать её большим.”

Алан Дж. Перлис (1 апреля 1922 – 7 февраля 1990)

Эта тема может считаться самой главной. Ошибочно то мнение, что компьютерное устройство – это некое подобие обычной техники – не требующей вмешательства человека. Если у вас на устройстве обеспечена возможность автоматического течения нужных процессов, и установка нужных свойств и параметров, то – да, ваше оборудование может

работать автономно, но первоначальная его настройка делается человеком. Можно сказать, что именно компьютеры наиболее способны на многомесячную автономную работу – если не сказать что только они и годятся для неё. Заметим, что автономная работа требует решения вопроса о бесперебойном электропитании – и это всего один из многих вопросов. Мы сейчас практически описали работу сервера – служебного компьютера, который *в решаемых им задачах* не равен типичному настольному компьютеру, планшету, телефону. Это так потому что типичная работа – за типичным компьютером, например, это непредсказуемая, не формализованная, не всегда автоматизированная работа, где человек использует в качестве инструмента работы в основном программы. Программы предлагают набор возможностей, но сами обычно ничего не делают. Ими надо управлять. Поэтому взаимодействие пользователя (и администратора) с компьютерным устройством – это во многом управление программами.

Программная среда и техника.

Трудность этой темы – в том что мы имеем множество разных устройств, и разные среды работы. Наиболее описанное устройство – компьютер типа PC, а наиболее описанные среды работы – это наиболее популярные ОС, и работающие поверх них *окружения рабочего стола* – обозначаемые как DE⁵, и консоль.

Рассмотрим всё это более внимательно. В качестве оборудования выступают

- примитивные телефоны – малоинтересные нам устройства из-за небольших возможностей. Не описаны тут.
- более развитые телефоны с ОС, отличной от Android, iOS – очень трудный предмет из-за разнообразия их ОС. Не будут рассмотрены.
- Телефоны (смартфоны) и планшеты с ОС Android – более-менее унифицированные устройства, и интересные в основном как ассистирующие, уступающие компьютерам в возможностях полезного использования
- планшеты и смартфоны с ОС iOS – дорогие и неизвестные автору устройства. Не описаны тут.
- планшеты с ОС Windows – потенциально интересные, но неизвестные автору устройства. Не описаны тут.
- обычные компьютеры типа PC – наиболее понятные, и часто используемые устройства. Описаны, но, в основном, без разбирательства вопросов их технического содержания.
- микрокомпьютеры – дешёвые, во многом похожие на PC компьютеры. Затронуты в этой книге как интересные бедному покупателю.
- компьютеры других типов – редко применяемые в России. Не будут описаны тут.
- более экзотические устройства – мало известные автору. Не описаны тут.

Таким образом, в этой книге будут в первую очередь рассмотрены компьютеры типа PC, а также устройства с Android. Достаточно ли этого? Для полного описания – нет. Но *понимание* работы с ОС для этих устройств само во многом поможет разобраться с остальными устройствами и ОС.

Каждый пользователь должен чётко понимать деление компьютерной науки на образование по двум предметам: оборудованию, и программной среде. Также он должен понимать *взаимосвязь* техники и программной среды. ОС должна быть совместима с техникой, иначе она будет неработоспособна. Программная среда – это совокупность всех программ, поставленных на единицу техники. Техника нас интересует как обладающая возможностями – предоставляемыми, например, её вычислительной мощностью, объёмом оперативной памяти, разъёмами, и так далее. При всей важности техники – основное внимание уделяется программной среде. Это так потому что именно она должна быть понята для обретения способности управлять обработкой, передачей, и хранением данных, а также изменением программной среды.

Проблема интерпретации приёмов работы. Ведение тетради.

В этой области лежит труднообъяснимая проблема интерпретации ситуации в программной среде. Это вторая трудность для автора, и читателя. Почему это так? Потому что ситуация, в которой потребуется работать, будет меняться, и неверно, что надо просто выполнять инструкцию. Самая лучшая документация иногда ничего не говорит о том *что*

делать. Она описывает возможности, а не действия, или действия «птичьим языком» для уже грамотных людей. Каждый ученик должен это понимать. Документация не является инструкцией. Она предполагает что вы сами решите что делать. Поэтому даже документация – не всегда то, что ожидается, и не позволяет вам работать бездумно. Тем более меняющаяся ситуация – затрагивающая разные вопросы – требует не столько знания, сколько мышления.

Насколько это сложно и возможно? Это уже делается – например, в работе водителя. Нет инструкции по вращению руля – есть карта, и мышление. Одно и то же вращение руля ведёт к разным пунктам назначения – в Омск, и в Магадан попадают, вращая руль – но только смотря на карту. Насколько это опасно? Опасность есть. В большой степени опасность состоит не в свободе интерпретации, а в бессистемной, механической работе. Не надо к ней стремиться. Инструкции могут быть непригодны, они могут устаревать.

Что такого трудного в этой интерпретации? Надо разделить *приёмы* работы, и их функцию в конкретной программе. Попытаюсь объяснить. Если вы разобьёте на сковороду яйца, то получите яичницу. Если вы положите мясо, то получите шницель. В обоих случаях вы включаете газ, льёте масло. Но результат получается разный. В данном случае функцией является либо приготовление одного блюда, либо другого. Но действия сами по себе никакого отношения к функции не имеют. Когда вы будете жарить картошку, то вы опять включите газ. Почему? Потому что вам надо *жарить*. Оказывается, что прошлый опыт в кулинарии может помочь готовить новые блюда. Я подозреваю, что многие ученики старшего поколения напоминают поваров, не способных учиться на уже выполненной работе.

То же с компьютером. У вас обычно не будет инструкции по жарке яичницы. У вас будут *способы* нагрева, равномерной обжарки (вращение и помешивание), избегания обугливания (масло), и так далее. Именно то что приёмов немного позволяет освоить управление программами и данными, и вообще освоить компьютерное устройство.

Так всё же – что делать? Особенность инфосистем в нынешнем виде в том что вам часто не нужен точный рецепт. Не потому что можно делать что угодно. А потому что способы и знание покажут что делать. Ваше внимание должно с действий во многом переместиться в плоскость сопоставления обстоятельств и предвидения – примерно как у водителя. Вы сами создаёте инструкцию, или технологию. Многие статьи по разным вопросам освещают лишь самую суть – предполагая, что вы уже знаете все смежные, так сказать, «школьные», вопросы.

Эти статьи, подобно рецептам, могут даже быть устаревшими из-за чрезмерной конкретности. Чтобы сделать их более универсальными и долговечными, можно было бы просто перечислить действия – не уточняя их до конца. Приведу пример. В старых версиях ОС FreeBSD для установки пакета требовалось использовать команду **pkg_add**, а в современных – **pkg**. Команда изменилась, но действие, и соответствующий раздел руководства ОС – остались.

Обычно ничего подобного не пишут в пособиях для начинающих. Часто эти самоучители и подобная литература показывают что автор потрудился лишь описать действия, а объяснение их смысла может вовсе отсутствовать – что мы оставляем на совести таких авторов. Это просто набор действий – мечта многих учеников. Я же тут пытаюсь дать вам нечто большее чем очередное описание действий. Вы же хотите работать так же ловко, и в разных областях, как молодые? Так вот, смысл, как и функция поднимают вашу подготовку на более высокий уровень.

Старшее поколение думает что надо просто нажимать какие-то кнопки. На самом деле всё гораздо сложнее, и, при этом, проще – потому что знание нескольких принципов заменяет знание горы фактов. Старшее поколение стремится запоминать. На самом деле запоминание не требует усилий. Совсем. Это не школа. Тут не надо зубрить. Вы должны не запоминать (и лихорадочно записывать) действия, а стараться уловить их смысл, и используемые приёмы – чтобы применить их в других работах. Если вы хотите равномерно поджарить шницель то вам не нужна инструкция о том когда и как ворочать ножом. Достаточно просто знать о способе равномерной обжарки. Он безразличен к тому что вы готовите. Ученик запомнит то что многократно сделает, и осмыслит.

Что касается ведения тетрадей, то, к моему удивлению, это оказалось не такой простой вещью. Я даже советую вам вести тетрадь. Мы сейчас обсуждаем обычный предмет компьютерных курсов. Посмотрите критически на свои записи. Они должны вам помогать. Не быть просто «кашей», свидетельствующей о вашей спешке. Не быть чем-то загадочным – потом, когда вы забудете урок. Не быть бездумным набором операций – неверных в случае малейших изменений. Например, если вы изучаете поиск в интернет, то на курсах вам могут

сказать набрать поисковый запрос «собака». Из этого не следует что вы при каждом акте поиска должны буквально делать это. Иначе у вас весь интернет станет лишь энциклопедией о собаках. Звучит это смешно, но, увы, недалеко от фактов.

Поскольку это всё так, то не всегда справедливы упреки что «нам не объяснили как делать». Смотря что это значит. Так, я могу ссылаться на пакетный менеджер – не рассматривая его конкретно. Это создаёт проблему? При правильном подходе – нет. Вы сами прочтёте о нём – скорее всего с экрана – именно о том пакетном менеджере, который есть в вашей ОС – если она из семейства UNIX.

Проблема понимания, пришедшего на смену знанию, особенно важна при обучении практике. Поэтому она описана в этой главе.

Проявления программной среды.

Итак, что это за консоли, DE, и вообще взаимодействие с техникой? Консоли, DE, и ОС – это всё проявления программной среды. Она начинает работать после включения техники, и выбора ОС – если её надо выбирать.

- Если вы имеете устройство с Android, то там нет DE, а есть наглядная, упрощённая до крайности рабочая среда, сведённая в общем к нескольким явлениям
 - начальному экрану
 - одному или нескольким «экранам» с панелью внизу – или эта панель на устройстве
 - поставленным вручную программам, и уже поставляемым с устройством. Каждая программа имеет значок и название.
 - «окну», или «окнам» каждой программы. Каждое «окно» занимает весь экран – но не панель
- если вы используете микрокомпьютер или компьютер типа PC, то вы имеете возможность ставить и удалять разные ОС. В общем, эти ОС могут предоставлять вам четыре среды работы
 - консоль – чисто текстовая среда – способная показывать лишь текст
 - консоль с фрэймбуфером – консоль с возможностью показа шрифтов, окон программ которые сами умеют работать через фрэймбуфер, изображений по мере явного запроса на это
 - оконная среда UNIX, работающая поверх консоли – обычно используемая наглядная среда, часто с неким DE. Обязательно есть, или нужна терминальная программа.
 - оконная среда Windows. Есть нечто вроде терминальной программы, но работа в ней не похожа на таковую в UNIX.

DE и оконная среда.

Оконная среда уже есть везде где возможен показ окон. Для этого в среде UNIX используется *оконный сервер Xorg*. Это программа, умеющая создавать окна на экране. Однако этого мало. Чтобы рабочее место было пригодным для использования потребуются как минимум оконный менеджер. Он придаст более понятный и чёткий вид окнам. Обе эти программы ещё не являются DE. Простейшим оконным менеджером является **Openbox**. Эта информация на практике в основном пригодится владельцам слабой техники вроде старых микрокомпьютеров.

Что такое эти DE? Это попытка дать пользователю достаточно наглядную среду, позволяющую уменьшить количество командного управления. Эти среды похожи на оконную среду Windows на первом приближении, далее начинаются отличия.

DE выполняют задачи

- показа окон программ
- управления окнами программ
- выдачи сообщений
- показа индикаторов
- по возможности наглядного управления программной средой, и техникой. В управление программной средой входит, например, регулирование используемых шрифтов.

DE фактически являются набором программ. Ученику желательно иметь не просто DE, а иметь при этом простой способ выхода в интернет с помощью DE. Автор много лет при

наличии проводного подключения к интернет использовал в качестве DE **LXDE**. При таком подключении эта среда не требует ломать голову над выходом в интернет. При этом она «лёгкая» – будет быстро работать даже при малой вычислительной мощности.

DE всегда включают в себя терминальную программу. Зачем одновременно используются оконные среды и консоль? Консоль – действительно более древняя среда. Она всё ещё достаточна для серверов, и иногда становится единственной средой для микрокомпьютера – смотря что там происходит. На сервере оконная среда не столько поможет сколько заберёт оперативной памяти. На некоторых ОС консоль – это всё что вы получаете сразу после установки ОС. В таком случае оконную среду надо ставить, и, возможно, заставить работать после этого.

Что такое эта терминальная программа? Это программа, очень похожая на консоль. Но хорошая такая программа позволяет вам настроить шрифт текста, и раскрашивание текста. Это важно при продолжительной работе. Исходный – обычно белый текст – да ещё мелкий – на чёрном фоне может вас не устроить. В функциональном плане терминал, как и консоль, позволяет выполнять классическое командное управление, заложенное в UNIX изначально.

Окна. Рабочий стол.

Что такое окна? Окно напоминает прямоугольную картонку, и всегда создаётся программой, а сама программа в таком случае называется оконной. Некоторые программы создают или нет окно смотря по тому как их запустили. Окна бывают основными и дочерними. Минимум, создаваемый оконной программой – основное окно. Таким образом, окна есть отображение программ. Окна располагаются на экране так чтобы всегда было можно видеть хотя бы край у каждого окна. Иногда к одному компьютеру подключают два монитора, но оконная среда у них одна; обычно одного монитора достаточно.

Понятие *рабочего стола* обозначает пространство экрана, или экранов, или – в более узком смысле – лишь пространство между панелями, и как бы за окнами. На рабочем столе могут быть значки. Обычно они присваиваются файлам, каталогам, и программам.

Оконная среда DE или Windows состоит из рабочего стола, панелей, окон, и сообщений. В UNIX есть возможность иметь более одного рабочего стола в один момент, но их нельзя отображать одновременно – по крайней мере на одном мониторе. Сообщения существуют как прямоугольные, похожие на окна, и то, что принято называть *ballon*, или всплывающие сообщения. Они часто сами исчезают. С помощью сообщений программы информируют пользователя.

Зачем нужны окна? Окна нужны для показа того, что можно назвать образами. Это не обязательно фото. Во-вторых, на окнах расположены *элементы управления*, с помощью которых вы можете чем-то управлять.

Индикаторы часто расположены на панели. Панель в оконной среде обычно «приклеена» к краю экрана, и может быть настроена. Что показывают индикаторы? Всё, что могут предложить программы, и что позволяет аппаратура. Алфавит, используемый при наборе – то есть раскладку клавиатуры (keyboard layout). Текущее время. Индикатор подключения сетевой платы. Заряд аккумулятора. Значок для регуляции уровня громкости. Уровень нагрузки на микропроцессор. Используемый объём оперативной памяти. Индикатор работы сети Wi-Fi.

Мы рассмотрели оконную среду «классического» типа, а не «окна» в Android.

Теперь рассмотрим неизбежный вопрос – как с этим обращаться. Буду описывать лишь консоль и оконную среду классического типа. Если вы их освоите, то обращение с Android будет более понятно.

Консоль вообще сама не умеет показывать окна. Она всегда занимает весь экран, и отображает лишь текст. Использование фрэймбуфера переводит видеоплату в другой режим, и временно позволяет показывать нечто кроме текста. Обращение с консолью простое – как с текстом. Примерно говоря – как при наборе на печатной машинке. Смотрите главу о командном управлении.

С оконной средой – в DE или без неё – всё намного более сложно. Вы получаете наглядность, но должны понять как с этим работать. Надо научиться обращаться с

- окнами
- панелями
- элементами управления

Пусть рассматривается некая обычная DE. Тогда есть панель, к надписям или значкам

которой привязаны меню. Меню – это набор строк, обозначающие разнотипные объекты, программы, действия, и так далее.

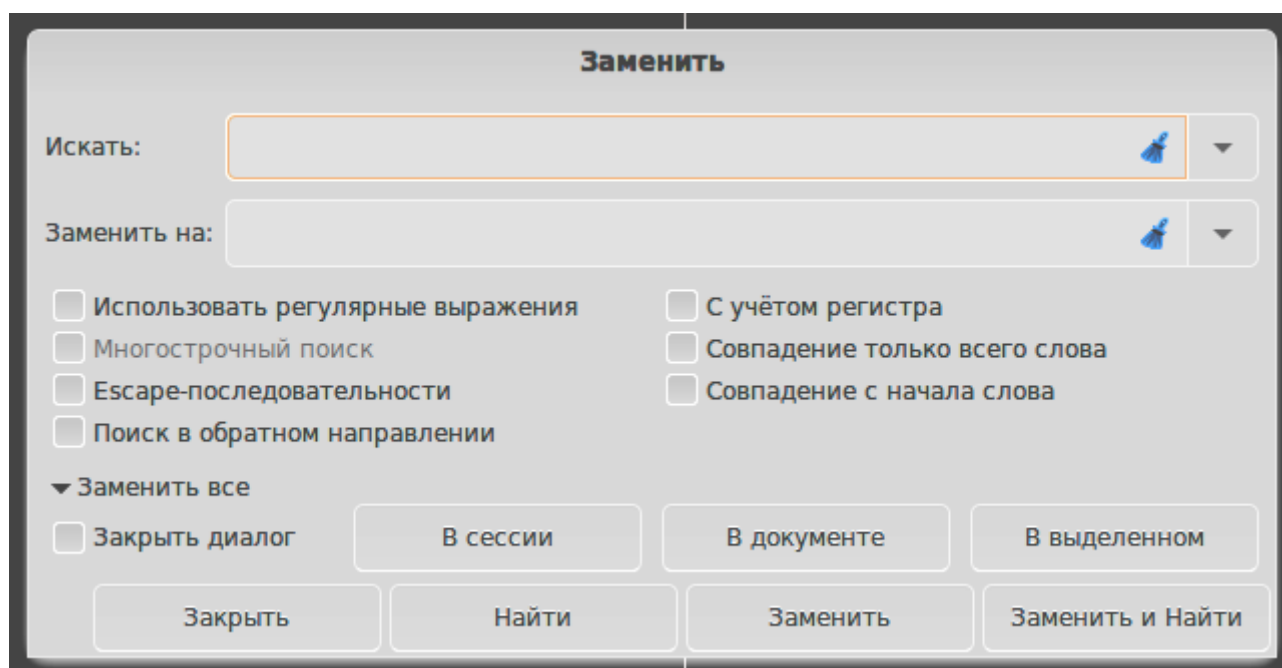
Для управления окнами обычно на их заголовке – вверху окна, есть значки для сворачивания, раскрытия, и выключения окна. ВАЖНО – выключение основного окна обычно выключает создавшую окно программу. Также вы можете

- перетаскивать окно
- менять размеры окна
- сворачивать и разворачивать окно

Кнопки управления окном.



Разворачивание окна – это распаивание и показ окна над рабочим столом. Сворачивание окна это его схлопывание, сворачивание в кнопку на Панели задач. Для каждого окна могут быть установлены ограничения в этих действиях. Так, типично, что дочерние окна нельзя растянуть, и часто свернуть. Чтобы управлять родительским окном часто необходимо закрыть дочернее окно. Типично что дочерние окна нужны для уточнения чего-то или выполнения одного из возможных конкретных действий. Основные окна показаны в главе о браузерах. Вот пример дочернего окна для замены и поиска текста.



Протащить окно (или что-то другое) – это переместить это вместе с курсором мыши. Протаскивание начинается при нажатии левой кнопки мыши, и заканчивается при отпускании. Перетащить окно, развёрнутое на весь экран, нельзя. Этот процесс ещё называется drag-and-drop, DnD – перетащи и брось. Окна протаскивают за заголовок (где написано «Заменить»). Свернуть окно можно в кнопку на панели задач, либо в окно, то есть уменьшить размер окна. В обоих случаях соответствующая программа работает.

Консоль и оконная среда. Простейшие действия с текстом.

Консоль – это строковая среда, где возможны лишь строки. Если быть более точным, есть как минимум два режима работы видеоплаты – консольный и графический. Консоль обычно противопоставляют графическому режиму. Но при работе в консоли можно через фрэймбуфер выдавать на экран нечто кроме строк. Тут я не стану обсуждать глубины вопроса о выдаче чего-либо на экран, и не надо это знать.

Консоль – это то, куда вы попадаете в некоторых случаях и системах. Она не должна вас пугать так как – если вы захотите – вы можете использовать на тех же системах наглядные, неконсольные рабочие среды. Если вы совсем не знаете как работать с компьютером то последние могут также выглядеть загадочно, но порадуют вас обилием цветов и кнопочек. Эти среды – оконные. Существует отдельный раздел компьютерного образования об управлении окнами. Он по существу не связан с *содержимым* окон, как и с работой в окнах.

В консоли вы работаете с помощью команд, в оконных средах вы часто можете этого не делать, но на практике масса литературы и форумов интернета, и статей интернета показывают команды. Их можно выполнять с помощью терминальной программы, или терминала. Если смотреть более внимательно то видно что терминал всё же не равен консоли. Так, например, в терминале **sakura** очень легко изменить шрифт. Для людей с близорукими глазами это важно.

Оконная среда не обязательна в бесплатных ОС, хотя она может оказаться психологически более приемлемой чем консоль. Может быть и наоборот – в зависимости от пользователя. Сочетание оконного менеджера с сервером окон не предъявляет таких больших требований к оборудованию как GNOME, KDE, Cinnamon. На слабом компьютере это позволяет получить более быструю рабочую среду. Если вы ставите FreeBSD 9 то можете в конце установки получить оконный менеджер **twm**. Не следует на него полагаться так как он бестолков. В качестве оконного сервера обычно используется **xorg**. Его в обиходе называют «икс».

Каждый пользователь должен знать как управлять окнами, и часто – как работать в консоли.

Бесплатные ОС даже в спартанском виде обеспечивают вас примерно 8-ю виртуальными консолями. В каждой из них вы работаете последовательно. Нельзя просто подавать команды. Команды можно подавать лишь при наличии приглашения командного интерпретатора. Оно, например, у меня сейчас выглядит как

```
user@mint ~ $
```

Текущая компьютерная работа может надолго отодвинуть во времени момент показа приглашения. В таком случае текущее задание можно отправить в фон. Там оно выполняется. вы же получаете возможность ввода команд.

Для запуска оконного сервера используется команда

```
startx
```

Оконный сервер запустит оконный менеджер или что у вас там. Вы получите оконную среду. В некоторых ОС просто установка KDE или чего-то подобного не позволяет его использовать. Нужна настройка. Она делается в консоли. Во многих ОС вам не потребуется ничего делать для запуска KDE, GNOME или чего-то подобного – они заработают сами.

Все эти вопросы хорошо описаны на сайте freebsd.org.

Многие пользователи будут работать с дружественными ОС, и им гораздо важнее уметь работать с окнами. На домашнем компьютере практически невозможно обойтись без окон.

Окна надо отличать от сообщений. Сообщения так же могут иметь заголовок с кнопками, но это не окна. Пример сообщения



В данном случае заголовок никак не выделен цветом, но он есть вверху сообщения. На нём же кнопки управления окном: свернуть окно, развернуть окно, выключить окно. Окна сворачиваются в кнопки на панели задач, и разворачиваются на весь экран (в данном случае размер окна может быть фиксирован, но кнопка всё равно есть).

Окна бывают основные и дочерние. Выключение дочерних окон не так фатально. У дочернего окна всегда есть родительское основное окно. Вы работаете всегда с активным окном. Его заголовок обычно окрашен не так как у неактивных окон. Чтобы сделать окно активным достаточно щёлкнуть на его заголовке.

Такие программы как GNOME обеспечивают вас панелями по краям экрана. Панель задач обычно внизу экрана. На ней могут быть более одной кнопки окон. Щёлкая на этих кнопках вы разворачиваете и\или выводите на первый план окно.

Щёлкнуть – это нажать быстро и однократно левую кнопку мыши, когда её курсор на чём-то. Курсор обычно имеет вид стрелки, и перемещается при перемещении мыши по столу. При этом кнопки находятся вверху мыши. Следите за пальцами – не должно быть случайных нажатий. Двойной щелчок редко нужен и должен делаться быстро – нажатием левой кнопки мыши. Щелчок средней кнопкой делается однократно и редко нужен.

В показанном окне в правом нижем углу **кнопка** с надписью «Заккрыть» нажимается щелчком мыши. Также кнопки без надписей вверху окна для действий с окном.

Во многих оконных средах есть возможность управлять многими явлениями и программами с помощью нажатий клавиш на клавиатуре. Если вы аккуратно работаете то случайно не нажмёте их. Надо упомянуть курьёзный факт: женщины бюстом могут нажимать кнопки. Никогда не пускайте кошек на стол. Они могут лечь на клавиатуру.

Я сказал о выключении программ. С точки зрения программиста есть некие процессы, часть из которых связана с окнами и нужна вам. Эти процессы в большой степени есть просто работа приложений, и ОС. Даже если вы имеете только консоль то есть 20-40 таких процессов, но вы можете о них не знать. Если вы боитесь обрывать работу программы просто сверните её окно. Программа продолжит работу, но окно уже не будет мешать.

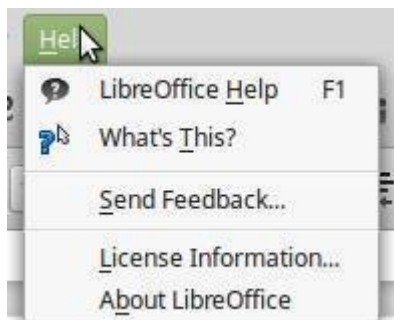
Некоторые программы не выдают на экран ничего, некоторые выдают значок на панель. Однако если нет панели то и значка вы можете не увидеть.

Элементы управления.

Теперь рассмотрим типичные элементы управления – некие видимые части окон с которыми вы работаете внутри окон. Кнопки уже обсуждены.

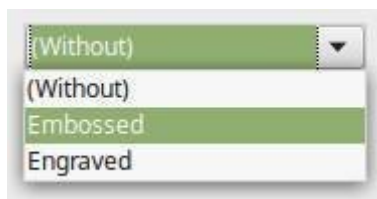
Запуск программ обычно выполняется через основное **меню**. Его вызывает кнопка в левом нижнем углу экрана. Там, кроме меню программ, может быть ещё масса всяких штук. Чтобы

убрать основное меню нажмите на эту кнопку ещё раз. То есть вы щёлкаете на этой кнопке, водите мышью, щёлкаете где надо. Движение мыши по диагонали приводит к ненужным результатам. Также меню бывают в окнах и на рабочем столе. Меню может вызываться однократным коротким нажатием правой кнопки мыши. В таком случае оно может отражать нечто связанное с местом его вызова и называется *контекстным*. Пример меню с англоязычными пунктами меню

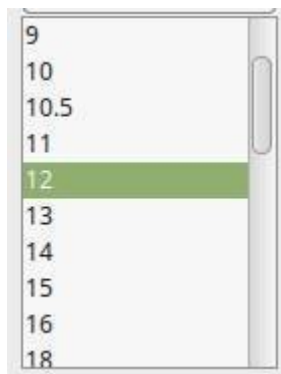


Убрать меню часто можно нажатием **Esc** (escape) на клавиатуре, или щелчком вне меню. Но вы имеете дело с реагирующей средой, и щелчок может вызвать что-то.

Меню надо отличать от списков. **Список** – это набор однотипных по смыслу элементов – обычно строк. Список может позволять редактировать одну из строк, при этом список часто подстраивается под то что вы вводите. Список обычно выглядит как одна строка в рамке, и как правило есть стрелка вниз справа от неё. При щелчке на строке или кнопке список «вываливается» вниз. Не беспокойтесь если он длиннее чем место на экране – это не опасно. Щелчок на строке в списке означает выбор некоего варианта чего-то. Так же **Esc** сворачивает список. Пример списка:



Линейка прокрутки – это элемент управления, необходимый когда не хватает места для показа чего-то. Внешне линейка выглядит как «щель» в которой есть прямоугольный, а иногда овальный ползунок. Минимальный размер ползунка достаточен для попадания в него курсором мыши, вообще длина ползунка показывает какую часть всего того что отчасти скрыто вы видите. Вы можете привести курсор мыши на ползунок и, зажав левую кнопку мыши, протащить ползунок. При этом вы будете перемещаться по тому что видно отчасти. Пример линейки:

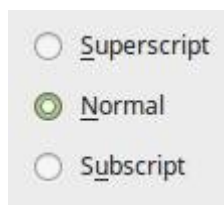


В данном случае неполностью виден список размеров шрифтов. Судя по размеру ползунка

видно не более половины строк с обозначением размеров.

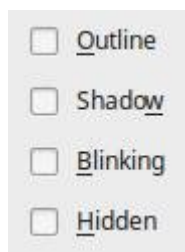
Светофор (радиокнопка) – это элемент управления для выбора одного, и только одного варианта чего-либо. Каждый вариант имеет работающий или неработающий кружок, а кружок с точкой показывает какой вариант выбран. Щелчком мыши на нужном кружке выберите нужный вариант. Если вы не можете так изменить положение точки то светофор отключён.

Пример светофора:



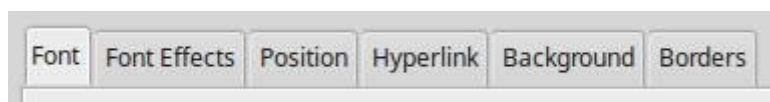
Флажки – это элемент управления для частичного или полного включения какой-то функции или свойства. Для этого надо щёлкать на флажке. Если флажок неуправляем то он отключен. Состояние одних флажков может определять состояние других.

Пример флажков.

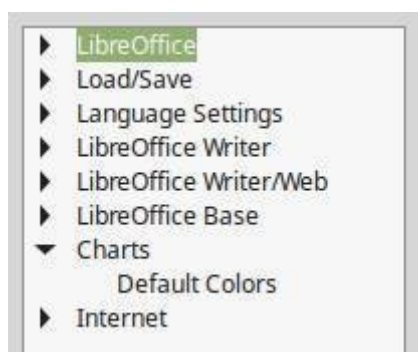


Блокнот – это элемент управления для размещения большого количества чего-то на небольшом пространстве. Блокнот ничего не включает и не выключает, он просто показывает наборы разного инструментария. Блокнот имеет вид телефонной записной книжки и управляется щелчками.

Пример блокнота:



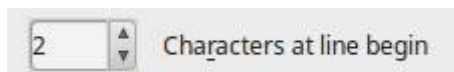
Дерево – это элемент управления для показа отношений или вложенности одной категории в другую. Дерево управляется щелчками. Оно само ничего не меняет но часто используется для создания упорядоченного представления или организации показа параметров. Пример дерева:



Счётчик – это элемент управления для определения числового значения чего-то. Он

показывает текущее значение, и часто это значение редактируемое как текст. Кроме того, типично, что есть кнопки для установки значения щелчком на них. Значение имеет предел, и перейти его нельзя.

Пример счётчика:



Простейшие операции с текстом.

В данном случае текст – это только символы, и тут не рассматривается создание публикаций, но в основе создания публикаций лежат эти простейшие операции.

Типично что для ввода и редакции текста существует прямоугольная область, которая часто зрительно отделена от того что ей не является – рамкой или цветом. Первоначально в ней может не быть текста, и может не быть текстового курсора. Текстовый курсор не должен путаться вами с курсором мыши. Текстовый курсор выглядит как мигающая или немигающая вертикальная черта. Для включения текстового курсора надо щёлкнуть в области для текста (которая может быть узкой и позволять работать только с одной строкой). Если там уже есть текст то так вы ставите курсор на ближайшую позицию в тексте, среди символов и пустых строк.

Забиваемый или вводимый иначе текст вводится на месте текстового курсора. При этом сам курсор перемещается. В простейшем виде набивка текста очень похожа на таковую на машинке, но это так пока вы не делаете опечаток. Для стирания символов есть клавиши **Backspace** и **Del** (delete).

Нажимая клавиши со стрелками, **Page Down**, **Page Up** вы перемещаете текстовый курсор. При этом он не выходит за пределы текста. Текст – это и пустые строки. Если вам надо отодвинуть абзац ниже то можно сделать несколько пустых строк нажимая **Enter**. Так же делается новый абзац.

Для копирования текста его прежде надо выделить. Затем он может быть вставлен в любой редактируемый документ. Для выделения текста надо установить курсор текста в конец или начало текста который вы хотите выделить, и, зажав левую кнопку мыши, протаскать курсор мыши к концу куска текста. Когда вы отпустите кнопку мыши получится заливка текста – обычно чёрная в простых программах. Залитые символы выделены, и стали белыми. Для снятия выделения достаточно двигать текстовый курсор, вернее, нажимать стрелки или щёлкать в тексте. Если есть выделение то клавиши стирания удалят всё что выделено. Также ввод текста (или изображения, и так далее) заменит то что выделено.

Практикуйтесь и научитесь вводить все нужные символы. Язык ввода влияет на ввод.

Глава 18. Как получить помощь.

Специалист, которого можно понять – безграмотен.

Специалист, которого нельзя понять – бесполезен.

(Шутка)

Помощь будем понимать широко. Это особенно оправдано для начинающих, и таких суровых ОС как FreeBSD. Ученик может вообще бояться и не понимать с чего начать, и как подступиться. Не надо форсировать этот этап. Он естествен, и ничего не говорит о вашем уме. Не спешите. Какие есть способы получить помощь?

- Поговорить с более подготовленным человеком. Желательно, чтобы он – или она – были опытными. Тогда количество услышанных вами заблуждений будет не так велико, могут остаться некие личные впечатления. Заметим, что у людей разный опыт, и, имея даже один опыт, можно делать разные выводы. Вам надо иметь в виду чужие мнения. Но и тот факт, что часто даже среди опытных людей мнения очень расходятся. Лучше всего сказать что вы – ученик, «чайник». Тогда можно надеяться на более понятное объяснение. Задавайте вопросы, проясняйте смысл. Разговор может быть долгим, это нормально. Вам потребуется ясная голова и конкретная память. Если вы – за компьютером, то это может быть не лучший вариант. Во многих телефонах есть диктофон. Если ваш собеседник дал вам свой телефон, то важно не злоупотреблять им.

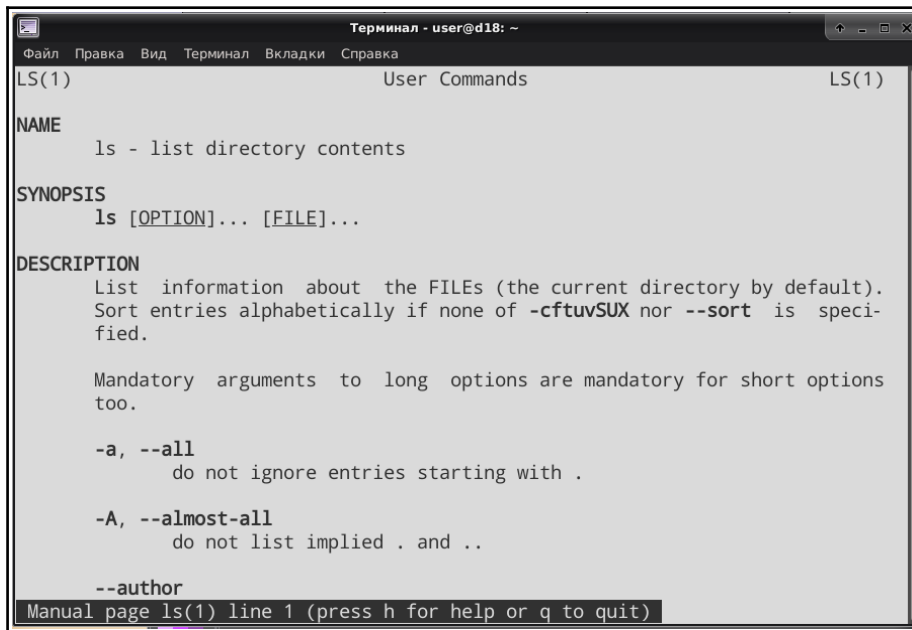
Ни один помощник не будет часами говорить с вами. Не будет по пять раз повторять. Лучше обеспечить удалённый доступ – и как можно скорее. Типичная проблема – «я забыл». Молодые – из-за особенностей мышления – могут прийти в большое раздражение от вашей забывчивости. Это так не потому что у вас плохая память, а потому что им она мало нужна. У них память заменена мышлением, и большой опыт работы.

- Пойти на курсы. Их полезность может быть самой разной смотря по качеству. Ищите курсы с опытными преподавателями. Молодые могут иметь большие трудности в передаче знаний. Так что преподаватель средних лет лучше. Но если он – халтурщик, то ваши деньги пропали даром. Также важна тематика курсов. Не все курсы нужны ученику. Не надо учить всё подряд. Хорошие курсы очень наглядны. Но – внимание! – вы увидите все операции не более двух раз. Хорошая идея – получить видеозапись курсов. Но учитель так рискует остаться без учеников. На курсах надо вести тетрадь. Не стесняйтесь спрашивать. Курсы – не такая глубокая и «неторопливая» вещь как литература. Видимо, лучшие курсы в России – при МГТУ имени Баумана. Курсы имеют мотив грешить делячеством, могут не описывать азов.
- Купить курс на лазерном диске для просмотра в телевизоре. Требуется современный – HD – телевизор. Иначе курс может быть бесполезен. Вы можете смотреть курс любое количество раз. Он – со звуком.
- Смотреть курсы из интернета, или брать оттуда, и записывать на лазерный диск. Очевидно, это трудно для ученика.
- Взять учебник в библиотеке. Основная проблема тут – отсутствие современных книг. Библиотекари в массе своей немолоды, и не понимают что нужна свежая литература. Во-вторых, вам нужна учебная, не специализированная литература – в первую очередь про вашу ОС. Книг по бесплатным ОС мало. Тем не менее стоит пойти в библиотеку. Вложенный в книгу диск может всё ещё быть в библиотеке.
- Купить учебник. Всегда качество текста важнее цены. Вы рискуете купить бесполезную книгу – даже свежую. Не спешите. Убедитесь что книга написана понятно. Я не очень представляю как можно уместить качественный учебник в размер брошюры. Если вы редко можете быть в магазине, то, возможно, стоит купить более одной книги.
- Смотреть журналы. К сожалению, с журналами плохо. Они во многом исчезли с бумаги. Во-вторых, они могут быть малополезны ученику. Так, нет особых надежд на журнал Linux Format. Он – качественный, но для сравнительно опытных людей, и, скорее, для молодых. В России издаётся неплохой журнал «Мой друг – компьютер», но там, похоже, нет ничего про UNIX. В некоторых журналах есть лазерные диски с текстами и прочим полезным. Не воруйте их, но используйте. В библиотечном интернет-классе можно сделать копию диска, и забрать её домой.
- Использовать чужое устройство. Проблема в том что это опасно для данных, и программной среды там. Вам могут отказать. Приходить в библиотеку или интернет-кафе надо уже подготовленным.
- Использовать своё устройство. Часто оно – семейное. Поэтому важно не навредить другим. Стоит попробовать live-диск. Для начала не принципиально какая у вас ОС. Так как вам надо войти в интернет, и читать тамошние статьи. Там же могут быть книги – некоторые из них даже не ворованные, и их можно легально скачать и читать. Упомяну бесплатные курсы института INTUIT. Они, как мне показалось, не глубокие. Зато доступны. Количество английских книг намного больше чем русских. Довольно много русских статей по UNIX. Ниже рассмотрим ваши действия более подробно.

man-страницы.

Если вы пользуетесь UNIX-подобной ОС, то там обычно готовы к просмотру man-страницы. Это не буквально страницы, а статьи без иллюстраций и деления на страницы, которые можно читать даже в консоли. Обычно для программы есть такая «страница», и иногда она ссылается на более подробный документ или книгу. Большинство man-страниц написаны на английском языке. Те же страницы после перевода на русский могут устареть. Обычно man-страницы описывают как минимум использование программы, достаточно часто там есть и другая информация. Страница поставляется обычно вместе с программой – по крайней мере исходная. Её перевод может не устанавливаться при установке программы.

Страницы можно просматривать в окне терминальной программы. Вот как это выглядит.



В данном случае используется терминальная программа. Она позволяет настроить вид страницы.

Для понимания того как читать страницы, или из чего они состоят, подайте в терминальной программе или в консоли команду

`man man`

Практически самое важное – понять как использовать консольные программы. Для этого потребуется понять условные обозначения в странице. В основном они таковы:

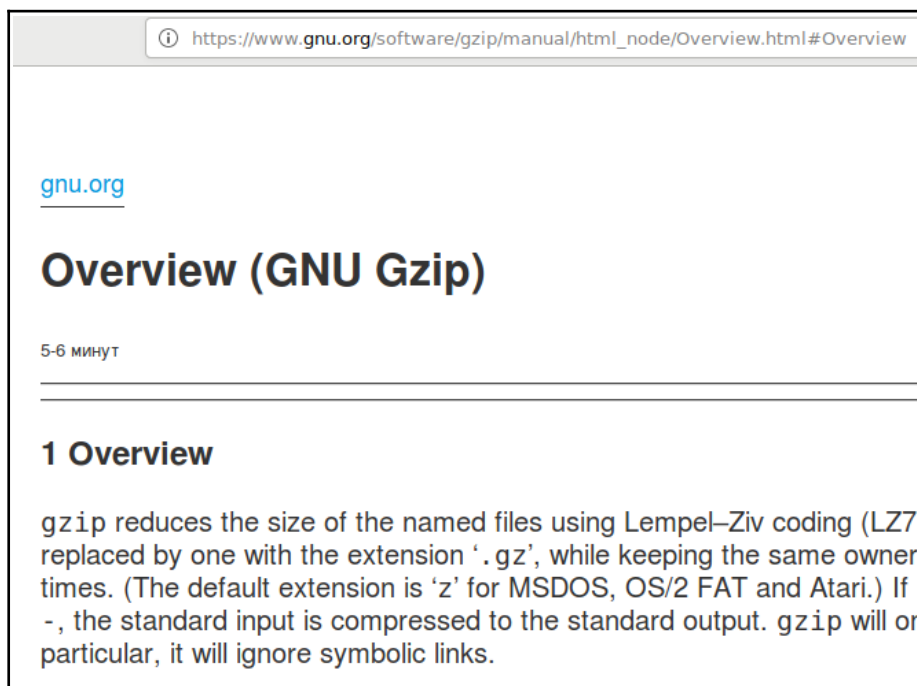
текст жирным шрифтом	набирать точно как показано
текст курсивным шрифтом	заменить соответствующим значением
<code>[-abc]</code>	любой или все параметры внутри <code>[]</code> необязательны
<code>-a -b</code>	параметры, разделённые <code> </code> не могут указываться одновременно
параметр ...	повторяющийся параметр
<code>[выражение] ...</code>	всё выражение внутри <code>[]</code> может вводиться несколько раз
<code><выражение></code>	обозначение выражения, но не его значение

Для выхода из режима просмотра страницы нажмите `Q` при активном состоянии окна терминальной программы. В консоли нет окон, но надо нажимать `Q` в той виртуальной консоли где показана страница. Проще всего листать страницы нажимая кнопки со стрелками на клавиатуре.

Часто вам не обязательно и не нужно читать всю страницу. Особенно интересны примеры и связанные по теме страницы, которые могут быть собраны в конце страницы.

Переводы страниц на русский язык легко найти в интернете. Также небольшое их количество можно найти или поставить из репозитория. Но в общем надо знать английский. Обычно страница написана простым языком. Она не объясняет используемые термины, и ориентирована на практику и наличие словаря или эрудиции читателя.

Вы можете выдать страницу на бумагу. Но это обычно довольно бессмысленно. Люди, собирающиеся работать с компьютером, должны быть способны читать с экрана. По крайней мере надо понимать какие страницы надо читать. Печатать все страницы бессмысленно и дорого. Так, сейчас на моём компьютере около 6000 страниц, и мне надо читать лишь очень малую часть их. Печать страниц имеет смысл для их чтения без компьютера, но лучше печатать руководства по тем же программам. Их прежде всего надо брать с сайта www.gnu.org. Все руководства с этого сайта, которые я видел, набраны довольно мелко, и готовы к печати. Там же есть возможность читать эти тексты, запрашивая их частями, с возможностью регулировать отображение текста (в браузере).



Для поиска слов в страницах проще всего использовать команду **apropos**, или **man** – смотрите соответствующую страницу. То есть подайте команду `apropos <текст для поиска>`

Для получения страниц без использования их ОС стоит посмотреть сайт www.freebsd.org, www.opennet.ru, linux.die.net/man/.

Читаемые с экрана документы.

Это довольно большая тема. Но она по сути не относится к помощи или документации. Она в основном относится к пользованию сайтами и просмотру PDF- и PostScript-документов. В среде Windows также часто вы получаете вместе с программой справку по ней. Программа обычно оконная, и справка показывается в окне, где можно искать слова, и простым образом выдавать на бумагу часть справочного документа или его весь. Справка может быть полноценным пособием. Обычно она не даёт глубокого знания, или вообще сводится к действиям в среде программы. Использование и вызов справок описано на сайте pkprovincia.ucoz.ru, а также rmanual.narod.ru. Руководства по программам для бесплатных ОС надо искать прежде всего на сайте www.gnu.org. Также смотрите сайт вашей ОС и сайт tldp.org. Создано достаточно много готовых к печати руководств, но обычно они на английском, и обычно набраны довольно мелким шрифтом. Стоит поискать их же в другом виде, или исходники для Latex (Latex source) чтобы изменить шрифт. Также для переделывания документа может быть полезна программа **pdftohtml** из набора утилит Попплера (**poppler-utils**). Вот пример готового к печати документа в окне просмотрной программы.

Table of Contents	
Preface	1
Contributors to this Manual	1
The Guile License	1
1 Introduction	3
1.1 Guile and Scheme	3
1.2 Combining with C Code	3
1.3 Guile and the GNU Project	4
1.4 Interactive Programming	4
1.5 Supporting Multiple Languages	5
1.6 Obtaining and Installing Guile	5
1.7 Organisation of this Manual	6
1.8 Typographical Conventions	6
2 Hello Guile!	9
2.1 Running Guile Interactively	9
2.2 Running Guile Scripts	9
2.3 Linking Guile into Programs	9
2.4 Writing Guile Extensions	10
2.5 Using the Guile Module System	11
2.5.1 Using Modules	11
2.5.2 Writing new Modules	12

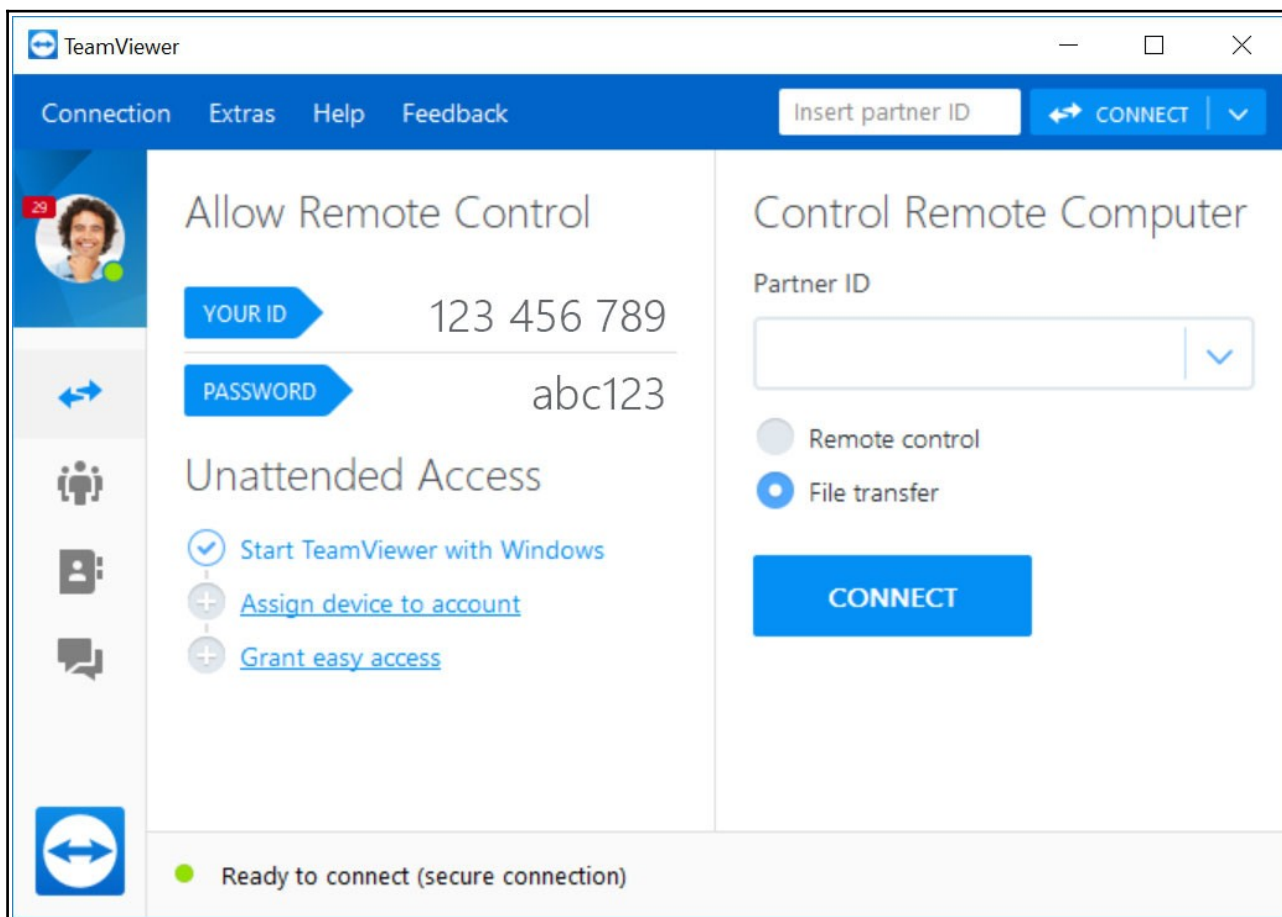
Удалённый доступ.

Это прекрасный способ помощи ученику, но его достаточно трудно объяснить. Под удалённым доступом понимается разное. Мы будем понимать под этим возможность управлять вашим компьютером, данную вашему помощнику – при том что он видит ваш рабочий стол. Такая возможность есть на любой популярной ОС, и есть довольно много программ для этого. Как часть Windows поставляется программа для удалённого доступа, поэтому её не надо даже ставить. В среде UNIX для описанного способа помощи необходима оконная среда. Нас интересует как быстро будет прорисовываться изображение вашего рабочего стола на чужом компьютере. Передача изображения выполняется обычно через интернет, и может сильно зависеть от свойств программы, и её настройки. Кроме того, нам интересно наличие возможности передавать файлы, и шифрование передаваемых данных. Для наибольшей скорости передачи изображения надо минимизировать его сложность. Это, в первую очередь, означает простоту экранной картинки. Кроме того, *разрешение* экрана должно быть не очень велико. Для медленных каналов глубина цвета изображения может быть сильно уменьшена в некоторых программах, а цветопередача – вообще выключена. Передача звука часто не нужна. Некоторые программы умеют сохранять сеанс помощи в видео, и его потом можно посмотреть любое количество раз.

Какие есть программы и способы сделать это? Их много. Самая популярная программа – работающая и на Windows, и на Linux, и на Android – TeamViewer. Она достаточно проста на первый взгляд. Заметим, что все заверения о её высокой скорости работы не означают что вам понравится её работа на очень низких скоростях передачи данных через интернет. Программа великолепна тем что не зависит от фильтрации данных, и адресации. Она заслужила свою популярность. Простейшее использование программы совсем просто, причём обеспечивается шифрование. Оно нужно чтобы посторонние не могли получить экранное изображение, звук и файлы. Первое, что потребуется сделать многим ученикам – поставить программу. Программа распространяется в основном как один файл пакета для ОС Debian, Ubuntu, Red Hat, SUSE, CentOS, Fedora. Это значит, что, скорее всего, потомки Debian, Ubuntu также годятся. Установка программы – обычная, требующая прав администратора. После этого надо запустить программу. Запуск можно выполнить командой

```
teamviewer
```

В результате вы получите окно программы. Наиболее часто обсуждаемый выпуск программы – полный. Окно выглядит так.



Вам надо сообщить помощнику свой идентификатор – в данном случае **123456789**, и пароль – в данном случае **abc123**. Не выключайте программу до конца сеанса помощи. Попутно, как и с другими программами, вы можете говорить по телефону с помощником. Пароль и идентификатор не присвоены лично вам, или навсегда. Они меняются. Показано окно версии 13.

Сеанс помощи сопровождается показом окна с экранной картинкой вашего компьютера на компьютере *помощника*. Он может регулировать параметры изображения и так далее. Картинка обновляется так быстро как позволяет скорость передачи данных. На очень медленных каналах она может замирать. Это не означает конец сеанса.

Трудность – в том что не все знают пароль администратора чтобы ставить программы. Заметим, что не следует самовольно что-то ставить или удалять в библиотеке или интернет-кафе. Есть другой – тоже несложный способ удалённого доступа. Это использование сайта www.screenleap.com. Сайт англоязычный, но простой в использовании. Он обеспечивает передачу экранной картинки через полноценный оконный браузер посредством Java. Если Java есть на компьютере, то вы быстро передадите изображение экрана помощнику. Если Java нет, то придётся её ставить. Это может потребовать прав администратора.

Посмотрите сайт www.webrtc-experiment.com.

Представляют интерес расширения (extension) для браузеров, так как они не требуют прав администратора.

Также есть программы более традиционного вида – в том числе уже поставленные, или поставляемые с ОС. Обращаться с ними несложно (см интернет), но могут быть проблемы из-за отсутствия публичного адреса, и работы фаерволла, или роутера, или закрытого состояния портов через которые работают программы.

Вы также можете получать помощь без оконной среды. Наиболее известное средство для этого – **SSH**. Это тема другой главы.

Глава 19. Трудности, ошибки и освоение программ. Неожиданности.

При всей своей мощности, компьютер требователен и придирчив. Ему нужны верные программы, и то, что мы хотим ему сказать, должно быть выражено точно в каждой мелочи.

Алан Дж. Перлис (из книги «Структура и интерпретация компьютерных программ»)

Компьютер подобен скрипке. Представьте себе новичка, который сначала испытывает проигрыватель, затем скрипку. Скрипка, говорит он, звучит ужасно. Именно этот аргумент мы слышали от наших гуманитариев и специалистов по информатике. Компьютеры, говорят они, хороши для определенных целей, но они недостаточно гибки. Так же и со скрипкой, и с пишущей машинкой, пока Вы не научились их использовать.

Марвин Минский

Трудности возникают почти в любом деле. Если вы хотите чтобы у вас их было побольше, и они нервировали вас сильнее то это очень просто – ничего не читайте, действуйте наугад, а когда что-то не получается – бейте по клавишам кулаком. Так вы узнаете как спокойно и хорошо вы на самом деле жили раньше, и испортите себе самочувствие.

Так вот – трудности естественно возникают в новой области вашей деятельности – особенно пока вы мало знаете. Также особенно если вы неправильно понимаете. Всегда надо убедиться что вы правильно поняли что прочли или услышали. Если вы говорите с кем-то, то всегда старайтесь выяснить точно что имеется в виду. Трудности не обязательно связаны с силой вашего ума, или плохой памятью. Я очень удивлюсь если найдётся хотя бы один человек, который без затруднений освоил компьютер.

Среди некоторых людей распространено мнение что «я такой тупой». Это мнение может иметь исторические причины, не связанные с силой ума. Любой человек может чего-то не понимать. Это не признак тупости. Возможно, ему плохо объяснили. Возможно, он просто не постарался понять. Возможно, он неправильно понял, и попытался использовать знание, и ничего не вышло. Некоторые путают знание с умом, а незнание с тупостью. Называйте вещи своими именами. Вы просто не любите технику? Тогда очень вероятно, что вы не станете с интересом или внимательно *слушать* про неё. Этап номер два: вы сталкиваетесь с техникой, и не можете разобраться – потому что не слушали как надо. Результат – вы ненавидите технику ещё больше, и считаете что это не для вас. Этап номер три: у вас формируется предубеждение против этой техники, и убеждение что «этого мне никогда не понять». Далее, вместо того, чтобы доказывать себе что вы без неё прекрасно обойдётесь, и зато умеете варить картошку, просто дайте себе расслабиться, а потом поищите знающего, терпеливого собеседника, или хорошую литературу.

Так что, я вас хочу затащить в этот отвратительный и неизвестный мир электроники? Нет. Я за компьютером более 20 лет, и не могу сказать что моя жизнь с ним счастливее чем без него. Интереснее – это да. Моя задача в том чтобы все знали куда ведёт тот путь, на который уже многие вступили, и я хочу облегчить продвижение по нему. Если честно, то надо признать обоснованность того *впечатления* что вы много стараетесь, и мало получаете. На самом деле это не так. Это так выглядит пока вас учат основам, и вы не знаете зачем. Если ученик учится сосредоточенно и у него хороший материал то он немного растратит сил, и получит ровно столько сколько возьмёт. Когда вам дают возможности – берите их. Вы же пользуетесь телефоном.

Теперь про ошибки. Это, наверное, для многих – страшная тема. Это от того что вы думаете что ваше устройство сломается. Не сломается. Вы можете беспечно позволить себе ошибки в двух случаях, и относиться к ним без страха:

- когда вы работаете с ОС в эмуляторе, где нет ценных файлов
- когда вы изменяете не ценный файл

Во всех нормально сделанных – во многих – программах заложен механизм предвидения

ошибочных действий человека. Он позволяет избежать ущерба. Кроме того, программы часто позволяют отменить сделанное действие. Заметьте, что источником ошибок может быть ваше невежество.

Теперь про ошибки программ. Программы могут ошибаться. Это так потому что ошибаются программисты. Обычно никаких важных или реальных ошибок программ нет. Тем не менее *сообщения* об ошибках могут быть. Это именно сообщения об ошибках, а не об отсутствии прав и так далее. Понимать это надо так: всякое несоответствие, могущие привести к неправильной работе, будет названо ошибкой (ERROR). Это так потому что программисты понимают последствия, и готовы бить тревогу лишь бы не создать реальный ущерб.

Вы неизбежно столкнётесь с необходимостью освоения программ. Некоторые считают что надо просто «как дать по этому компьютеру», «заставить его работать», и всё. Но ведь это не мясорубка. Не вентилятор, который достаточно включить. И тут не техника – главное. Без техники ничего работать не будет. Но вы работаете с программами, и часто вручную управляете ими. Опять же может возникнуть мысль что надо просто «чтобы они работали». Но даже у самой примитивной программы есть функция, и почти у всех программ есть обрабатываемые по-разному данные, часто разные режимы работы, разные инструменты работы. Разные подходы – иногда к решению одной и той же задачи.

В таких условиях крайне важно понимать что программа – это не просто «механизм», это то, что предполагает ваше знание и понимание того как работать – и с программой, и вообще. Предполагается, например, что вы знаете некую терминологию. На то есть документация и словари.

Но это не всё. Вам потребуется думать. Зачем? Чтобы делать нужную вам работу. Когда вы, например, тянете проводку по дому, то у вас есть инструменты. Сами по себе пассатижи, гвозди, провода, отвёртки недостаточны. Вам нужно *представление* о том что вы хотите и как работать. Вам потребуется знать как обращаться с каждым инструментом. Даже если он прост. Большинство людей, пользовавшихся молотком, попадали себе по пальцу. Хотя это не обязательно. Народный совет: зажмите гвоздь прищепкой. Программы, и их инструменты, могут быть не простыми. Программы развиваются – выходят новые версии. В общем они становятся сложнее. На UNIX всё сказанное усиливается тем что там часто нужно связать несколько программ в технологическую цепочку. Только она вся даст нужный результат. Так что думать приходится.

Что такое это «освоение программ»? Примерно говоря, это либо чтение их руководств, либо разбирательство без чтения. Чем сложнее программа, чем дальше от вашей практики, новее для вас её назначение, тем труднее освоение, и дольше. Гораздо лучше знать английский на уровне школы, и прочесть руководство.

Глава 20. Основы.

Вопросы хранения данных и программ.

Хранение данных и программ внутри компьютерного устройства – имея в виду не окружающие, второстепенные единицы техники – всё ещё обычная практика. Она частично вытеснена в основном благодаря двум явлениям: распространению устойчивых и быстрых каналов связи, и распространению использования тонких клиентов. Тонкий клиент – это устройство, которое способно лишь обеспечивать взаимодействие человека с программной и аппаратной средой без лишних единиц техники или деталей, и единиц программ на этой технике. Это примерно то же что кабина автомобиля без колёс, двигателя и прочего. Тем не менее тонкие клиенты полезны, и востребованы прежде всего в небогатых странах. Набор таких клиентов и необходимого для их полезного использования серверного компьютера позволяет упростить и ускорить администрирование.

Также вы можете использовать так называемые сетевые, или интернет-ориентированные ОС, которые в большой степени бесполезны без выхода в интернет, но могут задействовать доступные по сети мощные компьютеры.

Неким компромиссом между классической схемой с хранением всего на компьютере, и сетевыми ОС, является распространённая практика использования сетевых программ, и сетевых хранилищ. Сетевые программы могут быть проще в плане администрирования, и

могут вообще не храниться на компьютере, но выполняться там. То есть эти программы используют вычислительную мощность компьютера на котором используются. Сетевые хранилища на сегодня обычно это хранилища не программ, а данных. При достаточно быстрой передаче данных, или отсутствии нужды в получении большого их количества такие хранилища полезны, позволяют быстро применить данные, и упростить и ускорить администрирование.

При всём этом надо помнить о том что вы доверяете свои данные и программы людям и программам, управляющим сетевыми программами и хранилищами. Кроме сомнений в их соблюдении ваших интересов есть и другие факторы. Централизованные сетевые системы при сбое или остановке делают бессмысленной или невозможной работу всей сетевой системы.

Практика использования флэш-памяти для хранения программ выглядит сомнительной из-за ограничений на количество актов записи в эту память. Напротив, доступные через интернет дата-центры обладают такими мерами защиты от сбоев которые и не снились автономному пользователю.

С точки зрения пользователя он имеет дело прежде всего с файлами, каталогами, и носителями. Файл представляет собой массив данных или программы, или часть программы. Во всех этих случаях он что-то содержит — либо то что может быть *выполнено* (программа), либо может быть использовано программой, либо то что является объектом работы (данные). В частном случае файл пуст.

Каталог сам по себе ничего не хранит, но может содержать другие каталоги, и содержать файлы. В частном случае каталог пуст.

Носители.

Носитель — это физический объект, на котором записаны файлы и каталоги. Носитель может не иметь чего-то из этого. В частном случае носитель пуст.

В качестве носителей выступают предметы, которые из-за своих свойств могут не позволять изменять записанное.

Носитель	Ёмкость	Комментарии
Жёсткий диск (HDD, SSD)	Определяется размером разделов, суммарно — сейчас сотни гигабайт и более	Обычно встроен в компьютер, не в планшет. Быстро работает. Не терпит ударов если это HDD. Самое дешёвое хранение единицы объёма. При наличии бокса может являться внешним по отношению к компьютеру или планшету, телефону носителем. То же про многие роутеры. В продаже есть совсем небольшие диски размера 1,8 дюйма. Правильная установка диска размером 2,5 дюйма в стационарный компьютер – на салазки. Диск может издавать звуки.
Флэш-брелок	Десятки гигабайт	Удобен для переноса данных и программ. Иногда используется как носитель ОС.
Карта памяти	То же	Иногда используется как носитель ОС. Может иметь аппаратный запрет записи (замок). Часто используется в носимой технике. Для взаимодействия с устройствами используется читатель-писатель к конкретному типу карт - картридер.
Лазерный диск	0,7 гигабайта (CD), обычно 4,7 гигабайта для DVD, обычно 25 гигабайт для BD	Дешёвый носитель. Не терпит солнечных лучей. Должен быть чистым и не подвергаться механическому воздействию. В зависимости от типа может или нет быть очищен от записей. Иногда используется как носитель ОС. Достаточно медленно читается и пишется в соответствующем устройстве.

Стример	Ёмкость кассеты	Дорогое устройство для хранения данных на магнитной ленте. Дешёвое хранение.
Дискета	Практически всегда 1,44 мегабайта	Устаревший носитель. Медленное чтение и запись в своём устройстве. Повреждается магнитными полями. Имеет замок запрета записи.

Эта наглядная и несложная схема описывает известные, наблюдаемые пользователем явления. В действительности всё немного сложнее. Носитель для несения файлов и каталогов должен иметь файловую систему, а она расположена в разделе (то есть партиции, *partition*, в томе). На носителе может быть более одного раздела. Наличные разделы не обязаны иметь одну и ту же файловую систему. В частном случае разделов нет. Но тогда нет и содержимого. На работе такие действия как создание тома или файловой системы на жёстком диске выполняет администратор. Дома это делается частным пользователем. ВАЖНО – эти операции, как и создание таблицы разделов могут привести к потере содержимого раздела или носителя.

Свойства и роль файловой системы.

Файловая система важна потому что её свойства важны, и для использования данных и программ в разделе надо чтобы некая программа могла работать с файловой системой. Такая программа должна показывать другим программам содержимое раздела, и позволять операции с ним. Файловые системы в своих возможностях могут далеко выходить за пределы простой схемы хранения, описанной выше. Для пользователя представляют интерес, например, такие свойства файловой системы:

- файлы какого размера можно хранить. Это особенно важно если вы заняты обработкой видео, или телепередач. Иногда программа для таких работ может работать с несколькими файлами как с одним файлом. Из-за ограничений на размер может не получиться записать, например, файл более 2 гигабайт — при том что может потребоваться записать и файл размером в 4,5, и даже около 12 гигабайт
- какие есть ограничения на длину имени файла и каталога. Это тоже практически важно. Зачастую сохранённые страницы из интернета имеют длинные имена, и в них могут быть необычные символы
- какие есть ограничения на длину пути к файлу. Путь – это обозначение раздела, перечисление каталогов, в которых расположен файл, плюс имя файла; под управлением ОС Windows нужно указывать и имя носителя. Иногда под путём понимается то же, но без имени файла. В документации пишут **file** подразумевая точное описание положения файла — достаточно точное для описываемой программы. Например, файл этой книги сейчас обозначен так:

[/home/user/kniga2.odt](#)

Я использую Linux. В каталоге **home**, в каталоге **user** находится файл **kniga2.odt**. На Windows мне потребуются писать что-то вроде

[C:\kniga2.odt](#)

odt — это расширение. Если бы потребовалось обозначить компьютер, то обозначение могло бы быть, например, таким:

<http://127.0.0.1/home/user/kniga2.odt>

- может ли файловая система содержать кириллические имена файлов и каталогов, какие вообще могут быть символы в обозначении компьютеров, каталогов, и файлов. Простейший способ избежать проблем с именами – делать имена лишь из букв и цифр. Если вы хотите чтобы файл или каталог не вызывал проблем даже на другой ОС то имена должны быть короткими и набраны буквами основной латиницы
- какие есть ограничения на глубину вложенности каталогов
- является ли файловая система журналируемой. Если да, то это означает более высокое качество хранения
- максимальный размер раздела
- возможность сжатия данных и программ. Это позволят уменьшить размер, но не

влияет на содержимое файлов

- можно ли в файловой системе делать ссылки на файлы и каталоги. Ссылки сами не содержат данных или программ того файла на который указывают. Ссылки очень полезны так как иногда позволяют отказаться от копирования файлов и каталогов, то есть уйти от продолжительного процесса, и сэкономить место. Ссылка выглядит как файл или каталог, но не является ими.

Обратите внимание на возможные казусы из-за разных свойств файловых систем. Так, например, пока использовалась ОС DOS она не могла хранить файлы с длинными именами, *копируемые* с носителя с более совершенной файловой системой.

Свойства файлов и каталогов.

Каждый пользователь должен понимать следующие свойства файлов и каталогов:

- расположение. Файл или каталог могут быть не на поверхности носителя, а в каталоге
- имя. Или название. Это основное свойство для различения файлов и каталогов. Имя это строка. Последние символы после последней точки в имени важны как указания на формат, то есть тип файла.
- размер. Под размером каталога иногда понимается размер его содержимого. В частном случае размер равен нулю. Размер есть количество памяти, израсходованной на хранение файла. Размер измеряется в байтах (byte, Б), килобайтах (Kb, Кб), мегабайтах (Mb, Мб), гигабайтах (Gb, Гб), терабайтах (Tb, Тб). Байт часто равен одному символу. Остальные величины получаются из предыдущей умножением на 1024
- видимость и права. В UNIX ОС часто прячет имена, начинающиеся с точки. Права определяют возможные операции для пользователя и администратора. ОС запрещает выполнение не позволенных операций. Это позволяет защитить данные и программы от изменений и других опасных действий⁶
- собственность, или владение. У файлов и каталогов есть владелец. С точки зрения ОС им является пользователь с известным именем. В частном случае это администратор.
- время создания. Полезное на практике свойство
- время последнего изменения. Его часто требуется знать чтобы понять как давно редактировался файл данных. Файлы с одинаковым именем могут иметь разное время редакции. Так вы легко поймете какой из них более свежий. Запомните что это свойство при получении файла из интернета указывает на время получения, а не изменения там откуда скопирован или перенесён файл. При получении из локальной сети может быть иначе

Операции с файлами и каталогами.

Также каждый пользователь должен понимать операции с файлами и каталогами:

- чтение. Если чтение не позволено правами то вы не сможете увидеть содержимое
- запись. Если запись невозможна то нельзя изменить файл или каталог
- выполнение. Если выполнение каталога запрещено правами то нельзя войти в каталог, и видеть что в нём. Если для файла разрешено исполнение то он может быть выполнен. Даже будучи программой, файл не будет выполнен если нет прав на исполнение
- смена владельца
- смена прав (обычно выполняется администратором)
- переименование
- ручное установление времени изменения (timestamp)
- создание и удаление ссылки
- создание
- стирание, или перемещение в Корзину (это не одно и то же!). Корзина является каталогом.
- копирование
- перемещение

Автосохранение редактируемого файла.

В этой главке рассматриваются базовые вопросы хранения данных. Упомяну существенный вопрос об автосохранении файлов. Автосохранение прежде всего выполняется

6 Права подробнее рассматриваются в главе об администрировании

программами для набора текстов, и действий в электронных таблицах. И то и другое очень ценно, и при любых проблемах всегда надо иметь соответствующие файлы в наиболее свежем виде. Типично, например, что текстовый редактор (это программа) сам сохраняет документ каждые несколько минут. Документ технически расположен в оперативной памяти, и оттуда он пишется в файл — возможно, специальный файл. Также всегда желательно иметь автосохранение для всех важных файлов. Это может потребовать настройки программ. Если файл не редактируется то это не нужно.

Автосохранение нужно отличать от резервирования. Последнее занимается созданием копий неотредактируемых, но ценных файлов, и вообще ценных файлов.

Live-диск, live-карта, или live-брелок.

Прежде всего пользователям live-дисков надо понимать что существует три разных состояния хранения данных и программ

- данные находятся в оперативной памяти, в общем случае они изменяются, и должны быть сохранены в ПЗУ⁷ чтобы не потеряться. Это значит, что, как правило, изменённые данные должны быть сохранены в файл в ПЗУ. ОС запускается с live-носителя
- опять же главным местом работы является оперативная память, а постоянная используется лишь для ручного сохранения файлов, но параметры работы наследуются между сеансами. Благодаря этому брелок похож на HDD. ОС запускается с live-носителя
- классический случай когда есть ПЗУ с ОС, и программами для неё. Параметры работы наследуются, и происходит автоматическое или ручное сохранение файлов в ПЗУ. ОС требует однократной установки, и запускается из ПЗУ

Для первых двух случаев характерна большое потребление оперативной памяти, особенно если ОС полностью копируется в неё. Если файловая система на брелке не позволяет запись то брелок похож на live CD. Обычно файлы образов дисков для установки ОС рассчитаны на лазерные диски, а не брелки или карты памяти.

Вопросы работы программ, и программных процессов.

В среде UNIX достаточно часто используются консольные программы. Иногда они работают в фоне, но всё равно есть важное обстоятельство – необычные имена файлов и каталогов могут полностью нарушить работу некоторых программ.

Обычно никаких нарушений нет. Когда вы поставили ОС, то там имена не создают проблем. Что это за опасные имена?

- Записанные экзотическими письменностями. Возможно, я не прав, но лучше знать о такой опасности. Откуда берутся эти файлы? Прежде всего из интернета. Мы живём в мире где каждая письменность должна быть предусмотрена. Но на практике адаптация к письменностям не всегда есть. Так, например, американские ученики русского языка вынуждены вручную настраивать свою ОС для корректной работы с кириллицей. И это ещё не самый трудный случай.
- Слишком длинные. Файловая система может не принять такой файл, или обрезать имя. Такие случаи сейчас маловероятны.
- Содержащие символы, создающие опасность нарушения работы интерпретатора команд, или другой программы.

Конечно, некоторые скажут, что они дают простые буквенные имена, и всё будет хорошо. Но сейчас многие страницы в интернете сохраняются с нарушением этих требований. Суть в том что название страницы становится именем файла, и при этом часто имя содержит опасный символ.

Что это за опасные символы? Проще всего сослаться на руководство по командному интерпретатору, которым обычно является **BASH**. Некоторые символы для программ имеют специальное значение. Эти символы не являются буквами основной латиницы. Также обычно это не цифры. Современные ОС нормально работают с именами файлов и каталогов, набранными кириллицей. Тем не менее вопрос о символах надо иметь в виду. В отдельных случаях он влияет на работу программы.

Основная латиница – это алфавит английского языка. В него не входят некоторые символы других алфавитов.

Кроме того надо понимать роль частей команды. Например, такая программа как **scp**

7 Устройство постоянной памяти

управляется следующей командой

```
scp имя_пользователя@адрес:имя_файла ...
```

Команда может иметь вид

```
scp Aleksey@34.67.38.78:otchet2 .
```

Символы «@» и «:» отделяют части команды. Поэтому они не могут входить в части команды, которые отделяются ими. Также лишний пробел может нарушить работу программы.

Как проверить что нет опасных символов? Для этого не требуется просматривать сотни строк. Достаточно искать опасные символы в именах файлов и каталогов с помощью программ.

Вопросы передачи данных и программ.

Передача данных и программ в обычной практике – это передача файлов прежде всего через интернет и локальную сеть. Требованием к передаче является получение точной копии файла или правильный перенос файла. Первым свойством переданного файла является его размер, который во многих ОС показан примерно. Совпадение размера двух файлов ещё не гарантирует правильности копии. Долгое время для проверки копии использовалась контрольная сумма, но обычно их нельзя узнать так как их никто не считает для оригиналов файлов. Также проверка данных в архиве позволяет проверить копию. Также можно попытаться использовать файл по назначению. С приходом быстрых и стабильных каналов связи ситуация улучшается.

Во-вторых, важна скорость передачи. В локальной сети скорость обычно достаточна. При передаче через интернет она может падать до нуля.

Передача с помощью носителей иногда имеет смысл даже в плане скорости и надёжности так как объём велик, и нет проблем ошибок передачи через каналы связи.

Глава 21. Командное управление и автоматизация.

Командное управление.

Во всех UNIX-подобных ОС командное управление является обычным, часто основным способом управления и работы пользователя. Это по-прежнему так потому что в текстах обычно показаны команды. При этом UNIX давно позволяет работать в оконной среде. В Windows есть возможность использовать как некоторые программы из UNIX, так и встроенную среду командного управления, но она не похожа на таковую в UNIX. В Android стоит поискать программу для командного управления, а почему – будет написано ниже.

Что такое это командное управление? И зачем автоматизация? Командное управление – исторически первый способ работы пользователя. Он реализуется путём подачи команд и работает даже на самых купированных, базовых вариантах установки ОС, не требуя много оперативной памяти, вычислительной мощности или работы оконной среды. Из этого не следует что используемые программы мало требуют, мы говорим о способе управления ими. Команды могут выглядеть непонятно, но сам способ командного управления может быть более понятен, чем выполнение тех же работ в оконной среде. Команды, будучи текстом, можно сохранить, передать и показать в виде текстового файла.

Что такое подача команд? Это их набор или подстановка – с отображением на экране, и нажатие **Enter** для передачи команды командному интерпретатору ОС. Команда – это не слово, а строка. Так, например, есть команда **ls**. Она не является словом, и может применяться с уточнением благодаря *ключам*. Ключи – это части команды, благодаря которым команда становится гибкой в работе, и вообще осмысленной и полезной. Например, я могу просмотреть содержимое каталога и просто подав **ls**, и подав **ls -lh**. Результат будет в виде строк, но он будет выглядеть по-разному.

Не следует думать что достаточно просто подать команду – не понимая относительно чего она будет работать, и какие нужны ключи. ВАЖНО – всегда прежде читайте документацию к команде или программе, а потом пользуйтесь ею.

В командной среде важно какой каталог является *текущим*. Текущий каталог – тот, в котором вы как бы находитесь. При классическом виде приглашения командного интерпретатора (см ниже) имя этого каталога, и всех более общих каталогов показано. В

частности, когда говорят о текущем каталоге, то имеют в виду вообще ваше место в файловой системе. Это может быть поверхность раздела, то есть, так сказать, вершина. Вы не обязаны находиться в каком-либо каталоге. Так, например, если вы – в наиболее общем «каталоге», то команда `ls` будет отображать его содержимое. А именно – у меня, и, наверное, у многих пользователей UNIX оно будет показано как

```
bin  etc  initrd.img.old  lost+found  opt  run  sys  var
boot home  lib  media  proc  sbin  tmp  vmlinuz
dev  initrd.img  lib64  mnt  root  srv  usr  vmlinuz.old
```

Этот же «каталог» при использовании ключа `-lh` будет выглядеть иначе.

```
drwxr-xr-x 15 root root 4,0K июл 24 23:24 lib
drwxr-xr-x 2 root root 4,0K июл 24 22:58 lib64
drwx----- 2 root root 16K июл 24 22:56 lost+found
drwxr-xr-x 4 root root 4,0K июл 25 13:03 media
drwxr-xr-x 2 root root 4,0K апр 27 13:31 mnt
drwxr-xr-x 10 root root 4,0K июл 30 14:13 opt
dr-xr-xr-x 137 root root 0 авг 8 16:19 proc
drwx----- 10 root root 4,0K авг 6 18:12 root
drwxr-xr-x 23 root root 820 авг 8 16:20 run
drwxr-xr-x 2 root root 12K авг 5 17:22 sbin
drwxr-xr-x 2 root root 4,0K апр 27 13:31 srv
dr-xr-xr-x 13 root root 0 авг 8 16:19 sys
drwxrwxrwt 15 root root 4,0K авг 8 16:22 tmp
drwxr-xr-x 10 root root 4,0K июл 24 23:10 usr
drwxr-xr-x 11 root root 4,0K июл 24 23:10 var
lrwxrwxrwx 1 root root 26 июл 24 23:10 vmlinuz -> boot/vmlinuz-4.9.0-9-amd64
lrwxrwxrwx 1 root root 26 июл 24 23:10 vmlinuz.old -> boot/vmlinuz-4.9.0-9-amd64
```

Непонятные слова объяснены ниже.

Многие статьи про UNIX просто указывают нужные команды. Но их надо правильно понять. И важно где заканчивается команда, а где просто переносится строка. Это нужно понять прежде чем что-то делать. На страницах интернета часто в страницу врезана как бы горизонтальная лента большей ширины чем колонка текста – именно чтобы показать набор команд без того чтобы ломать строки где не надо. Если вы неправильно введёте команду, то – ВАЖНО – последствия могут быть разными – в том числе вредными.

Пользование масок файлов и каталогов при операциях с ними.

Маски файлов и каталогов – это небуквальное, обобщённое обозначение имён файлов и каталогов. Программа, в которой используются маски, должна считать некоторые символы особенными; то же верно про ОС. Типичные, классические маски таковы. Тут звёздочка и знак вопроса – особенные символы. Они не должны входить в имена.

Смысл маски	Пример маски
Маска для обозначения имени где каждый вопросительный знак обозначает один символ	filename?
Маска для обозначения имени где звёздочка обозначает несколько или ноль любых символов	filename*

Автоматизация.

Теперь про автоматизацию. Когда строились египетские пирамиды, то тысячи людей с большими усилиями тащили огромные камни. Они построили настолько огромные и устойчивые пирамиды, что они стоят до сих пор, и простоят ещё не меньше. Частный пользователь часто может сам выбирать как ему поступать – уподобляться этим строителям, или найти способ спокойно попить чаю пока компьютер сам сделает всю работу.

Обратите внимание что большая работа не только трудна, но приводит к утомлению, а оно означает падение внимательности и самоконтроля. Из-за этого растёт возможность ошибок.

За примерами далеко не надо ходить: летом 2017 года автор этих строк занимался обработкой больших массивов текста. Каждый из них на бумаге мог занять целый шкаф. Если бы эта работа выполнялась вручную, то всей моей жизни не хватило бы. Работа была сделана

за приемлемое время, применяя несложные команды. Я подавал команду, и, представляя объём работы, уходил гулять. Через минут 40 я возвращался. Работа была сделана – вернее, часть её.

Конечно, автоматизация не всегда выполняется с помощью команд (или программ в консоли). Тем не менее стоит хотя бы иметь представление об автоматизации таким образом. Программы в UNIX в большой степени готовы к автоматизации, и иногда прямо созданы для неё. Эти программы часто управляются командами. Если вам нужна автоматизация, то следует, прежде всего, обратить внимание на непрофессиональные программы для этого, а потом уже – на языки программирования. Этим заявлением я хочу сказать что программирование в чистом виде является профессией, хотя потребность в автоматизации может возникнуть у обычного пользователя-непрограммиста. Ему может быть трудно обеспечить автоматическую работу. Не пытайтесь добиться этого быстро. На работе автоматизацию лучше всего поручить подготовленному человеку. В домашних условиях вы можете поискать готовую программу, но помня, что она может на самом деле вам не подойти, и делать не то что вы хотите.

Зачем простому пользователю автоматизация? И разве в среде Windows люди не обходятся без неё? Примерный ответ таков. С 70-х годов работа не настолько изменилась чтобы совсем не нуждаться в автоматизации. Но на Windows иногда пользователь просто отказывается решать задачу не имея приемлемого способа её решить. Популярная задача – массовое переименование файлов или каталогов. Вы без труда можете переименовать пару файлов. На практике бывает надо переименовать сотни файлов, или тысячи. Делать это руками бессмысленно – утомление сделает вас неработоспособным намного раньше чем вы выполните хотя бы половину работы. Даже не начинайте.

Говоря об автоматизации, мы с вами вступаем в новую область. Это уже не вопросы техники, или хранения данных, или другие базовые вещи. Обозначу важные моменты.

- Команды, как и программы, всегда выполняют точно то что они должны выполнять, но не всегда то что вы хотите. Если программа делает не то что надо, то это всё равно буквально то что вы написали, или взяли где-то.
- Вы создаёте или используете программный механизм, который работает за вас. Результаты его работы могут быть фатальными. Не он отвечает за последствия, а вы – так как вы его применили.
- Так как мы рассматриваем не использование искусственного интеллекта, то программирование и автоматизация всегда делаются после формулировки набора нужных действий, выполняемых при наличии формализуемых, чётко выраженных свойств и обстоятельств. Что это значит? Вы не можете сказать программе «найди мне фото красивой девушки». Ни одна обычная программа не сможет это сделать. Потому что нет применимых, чётко описанных критериев красоты.

Конечно, у вас возникает вопрос – не слишком ли это опасно, как вообще так можно работать? Отвечаю: как программисты. Опасно. Но есть рецепты.

- Всегда избегать ситуаций, грозящих потерей данных. Избегать стирания и переписывания данных.
- Проверять программу. Лучше всего либо понять её, либо проверить с помощью тестовой функции, и\или тестового набора данных.

Вопрос номер два – и как часто удаётся сделать формальное описание? Так как мы не занимаемся поиском красивых девиц с помощью программ, то это очень вероятно, если не сказать сильнее.

Вопрос номер три – как можно убедиться что программа правильно работает? Это может быть трудно. Посмотреть результаты. Проверить их с помощью другой программы. Проверить обрабатываемые данные.

Вопрос номер четыре – разве нет готовых программ? Есть. Но *данные* непредсказуемы. ваша программа часто обрабатывает данные. А данные могут быть самыми разными.

Вопрос номер пять – и кто это делает? Многие – тем более когда у них нет выбора. Программированию учат в школе.

В основе программирования в практике непрограммиста – командное управление. Если понимать под программированием некие приёмы автоматизации, захвата больших массивов данных – в противоположность индивидуальной ручной обработке, то именно в командном управлении программирования почти нет. Вот этим «почти» и займёмся. Что это? В случае

UNIX это

- маски имён файлов и каталогов. То есть обобщённые обозначения имён. Например, «123?» обозначает как «1234», так и «1230».
- регулярные выражения. Это обобщённые описания частей текста.
- Конвейеры. Конвейер – это средство передачи результирующих, обработанных данных одной программы другой программе. Последняя получает эти данные для обработки.

Надеюсь, что это всё. Ко всему этому надо относиться со скепсисом. Так как всё это может подвести.

Базовые команды в Linux и BSD очень похожи, и либо работают сами, либо запускают программы. Перечислю основные команды.

Основные команды.

команда	назначение	Нужно ли быть администратором
su, sudo	Работа от имени администратора	да
adduser	Добавление пользователя	да
userdel	Удаление пользователя, и его данных	да
passwd	Назначение пароля для входа в систему	да
ls	Показ содержимого каталога	нет
chmod	Назначение прав	да
mount	Подключение устройства хранения данных	да
umount	Отключение устройства хранения данных.	да
nano	Ручная редакция текстов	нет
ps	Показ текущих программных процессов	нет
top	Показ программных процессов и их запросов к памяти и микропроцессору	нет
kill	Завершение процесса с вероятностью несохранения обрабатываемых им данных и возникновения нештатных ситуаций	да
man	Показ справочных страниц - в том числе в консоли	нет
apropos	Поиск справочных страниц по слову	нет
mkdir	Создание каталога	нет
rm	Стирание данных.	нет
rename	Переименование файла или каталога	нет
cp	Копирование файла или каталога	нет
mv	Перемещение файла или каталога	нет
7za и пр.	Обработка архивов	нет
more	Вывод строк порциями	нет
poweroff	Программное выключение программ, ОС, и компьютера	да
startx	Старт оконной среды	нет
exit	Выход из терминала или сессии работы в системе	нет
cd	Сменить текущий каталог на другой	нет

pwd	Показать текущий каталог	нет
history	Показать последние введённые команды	нет
lp	Печатать на бумагу	нет
ping	Проверить доступность адреса в сети	нет
ip	Показать сведения о сетевых подключениях, или настроить их	нет
uname	Показать сведения о системе	нет
gparted	Управление разделами (осторожно!)	да
links	Смотреть сайты в консоли - браузер	нет
wget	Скопировать файл или файлы с адреса в сети - скачать	нет
awk, sed	Автоматическая обработка текста	нет
cal	Показать календарь	нет
crontab	Задать задание для автоматического выполнения	нет
du	Показать как занят раздел	нет
grep	Поиск строк	нет
mail, mutt	Работа с электронной почтой	нет
dmesg	Показ сведений об оборудовании и пр.	нет
gufw и пр.	Настройка брандмауэра	да
clamav	Антивирусный тест	нет
mmcli	Настроить соединение с интернетом по модему	нет

Если команда запускает программу то сама программа должна уже быть. Если её нет то программа не работает, и команда не выполняется.

Стандартный ввод, выход и поток ошибок. Передача данных в программу и из неё с помощью пайпа (pipe). Перенаправление в файл и из файла.

Так как UNIX – это семейство операционных систем, то отдельные системы, входящие в него, похожи больше или меньше. Часто в таких системах работает **BASH** или нечто подобное. Такой интерпретатор умеет перенаправлять текстовые данные (на самом деле байты) из одной программы в другую. Что нам от этого? Вагон сахара? Нет. Два вагона? Опять нет. Это чтобы вам было проще. Чтобы была гибкость. Конечно, она нужна не всегда. Гвоздь не может быть гибким. Макароны нельзя забить в доску – они гибкие. Гибкость для построения произвольных, нужных пользователю структур для обработки данных – может потребоваться.

Именно такие темы как перенаправление мотивируют писать что Linux или UNIX надо «очеловечить». Так что – эти системы такие ужасные? Годятся ли они пользователю? Я считаю что правильно ставить вопрос не «годится ли пользователь», а «годится ли система». Один человек написал что это всё для некоторых пользователей – не для всех. Я бы сказал иначе – это для образованных, способных к конструктивному мышлению людей. Конечно, можно сказать «зачем это мне знать да думать». Те, кто уже разобрался с такими вопросами не жалуется. Так же как люди, умеющие писать не спрашивают зачем им это надо.

Перенаправление между программами.

В UNIX есть понятие потоков стандартного ввода, стандартного вывода, и вывода ошибок. Они обозначаются так.

поток	обозначение	типичный источник или получатель
-------	-------------	----------------------------------

стандартный ввод	stdin	0	клавиатура
стандартный вывод	stdout	1	экран
вывод ошибок	stderr	2	экран

Перенаправление между программами имеет общий вид

A|B

где A и B – программы, способные к выдаче данных (A), и получению данных (B). Таких цепочек может быть несколько в одной команде. Данные стандартного вывода подаются на стандартный ввод следующей программы. Символ «|» указывает интерпретатору команд что надо перенаправить данные.

Перенаправление в файл.

Если данные надо прочесть из файла или записать в файл то это делается с помощью символа «>» или «<». А именно, «>>» или «<<» дописывает строки в файл, а «>» или «<» переписывают – ВАЖНО – с потерей прежних данных – файл. Интерпретаторы **BASH** и **Korn Shell** позволяют использовать перенаправление с помощью «>>».

Например, вы можете подать данные программы в файл так.

```
echo -e 'abc\ndef'>testecho
```

Если файл **testecho** не существовал, то он будет создан. Мы подали строку при том что символы «\n» считаются обозначением конца строки – благодаря ключу «-e». Поэтому файл **testecho** будет из двух строк. Проверим это.

```
cat testecho
abc
def
```

Также вы можете увидеть что некая программа имеет вид

AC

Это значит, что файл B подаёт данные в команду A, а результат выводится в файл C.

Если не хватает потоков вывода.

Используйте команду **tee**. Например так.

```
ls -l|tee f1
```

В данном случае вывод команды **ls** подан и на экран, и в файл **f1**.

A теперь всё вместе.

Приведу пример, где используется и перенаправление между программами, и в файл.

```
cat file1|fold -w76>file2
```

Команда **cat** шлёт строки из файла **file1** в программу **fold**, которая разбивает строки по 76 символов, и шлёт в файл **file2**.

Замечание про автоматическую обработку текста.

Мой опыт обработки текста невелик, но важны его выводы. Программа **AWK** часто уже входит в поставку ОС, и часто обсуждается при этой работе. Я тоже её использовал, но считаю своей ошибкой попытку применить её в сложных ситуациях. Она обладает массой возможностей так как в нынешнее время является языком программирования. Руководство по программе – даже если бы оно было на русском языке – само уже является проблемой так как это сотни страниц, и найти нужную часть может быть сложно. В базовом применении она терпима. Видимо по мере роста возможностей потерялась простота. Во многих случаях, видимо, стоит обратить внимание на программу **SED**, которая тоже штатная, но более проста для освоения. Также стоит обратить внимание на программу **m4**, к которой есть документация на русском.

Заметим, что автоматическая обработка текста при сегодняшнем отсутствии искусственного интеллекта никак не связана с достоверностью информации, и может быть не связана с формальной, лингвистической правильностью текста. Так, например, никакая обычная программа не возразит против фразы «Одесса – город на Неве», или «Меня зовут ТУ-154».

Командный интерпретатор и программирование.

Такая программа есть и на Windows, и на UNIX. Но в последнем случае она до недавнего времени была более развита, то есть больше могла сделать. Интерпретатор команд в данном случае есть программа, занятая получением и выполнением команд, или выводом сообщений по поводу поданных команд. Вы можете подать непредусмотренную или недоступную для вас команду, и интерпретатор должен сообщить что такая команда не найдена.

Интерпретатор именно как командная оболочка в ОС принимает команды лишь в своём приглашении. Приглашение – это нестираемая часть экранного текста в консоли или терминальной программе, показывающая готовность интерпретатора принять команду. Типично, что приглашение включает

- имя пользователя
- «собачку» – @
- имя компьютера
- обозначение текущего каталога
- обозначение статуса пользователя – является ли он администратором

Так, например, сейчас у меня приглашение выглядит как

```
[user@localhost ~]$
```

Я работаю в своём каталоге (~), и не как администратор (\$). После набора команды надо нажать на клавиатуре **Enter** для передачи команды интерпретатору.

Некоторые программы имеют свой интерпретатор, и своё приглашение. Например, **Python**.

Что такое язык программирования.

Говоря на эту тему, мы сталкиваемся с массой текстов, которые могут быть не важны для ученика. Также замечено, что программисты являются агитаторами своего языка – не понимая ситуации ученика – особенно если это немолодой ученик. Язык программирования – это средство обработки данных, и, возможно, средство управления передачей и хранением данных. Язык на первом приближении должен удовлетворять как минимум двум требованиям:

- позволять программисту описать нужный процесс, и данные
- осуществить формирование структур данных, и их обработку. Ученику не всегда надо знать как именно.

Кроме того, конечно, язык должен позволять читать из файлов данные, и писать в них.

Пусть нам надо выполнить арифметические действия. Тогда язык должен позволять нам их описывать, и выполнять счёт. Это, разумеется, примитивный и надуманный пример. Тем не менее это так. Например, язык **Python** можно реально использовать как такой калькулятор. «А чего тут сложного, и зачем тут язык?» – спросят некоторые. Отвечаю. В данном случае язык должен различать цифры, знаки действий, и то, что не является ни тем, ни другим. Он должен отказаться вычислять выражение, которое не является арифметическим, например «Автобус отходит в 10:20». Потребуется считать, и примером необходимого, но неожиданного вопроса будет точность и округление чисел. В рассматриваемом случае данными являются числа, а знаки действий – нечто вроде обозначения операций над ними. В реальных языках действия над данными намного более сложны чем операции арифметики.

Язык программирования – по аналогии с естественным языком – есть набор понятий и правил, позволяющий создать текст программы, которая может быть выполнена, то есть реально обрабатывать данные. Зачем нужно программировать? Это реальный способ уменьшения ручной работы – когда она слишком велика. У администраторов могут быть свои причины программировать.

Существуют самые разные языки программирования, и даже набор самих понятий для их описания не совпадает. Тем не менее можно рассмотреть некий типичный язык. Так же как можно рассмотреть учебный язык, или популярный язык. В качестве такового можно взять, например, Pascal. Я не хочу этим сказать что он хорош (или плох) для заработка. Не вдаваясь в полное описание языка, отметим его идеи.

- Как и во многих других языках, постулируются различные *типы* данных, то есть различные виды данных, с которыми работает язык.
- В языке предусмотрена возможность описания условий, которые определяют дальнейший программный процесс, давая оценку выполнению условия в виде «да» или

«нет».

- В языке предусмотрены конструкции для многократного выполнения одной и той же операции. Таким образом, не требуется буквально записывать выражение для выполнения операции столько раз сколько раз надо её выполнить.
- Дана возможность создавать подпрограммы. Это позволяет строить программу из блоков, и простым образом использовать их.
- Язык снабжён набором математических функций, а также функций для преобразования типов данных. Так, например, «123» может пониматься и как строка, и как число. Но строки и числа по-разному *обрабатываются*. И может быть надо преобразовать данные из одного типа в другой.
- Язык позволяет использование модулей, то есть сторонних, но совместимых программ, в создаваемой программе.
- Язык направлен на создание исполнимых файлов, то есть программ, готовых к работе, и представляющих собой файл – выполнимый поверх конкретной ОС. В данном случае исполнимый файл создаётся путём компиляции текста программы.

Заметим, что, например, **Python** – не язык с компиляцией, а язык прямого использования текста программы при его интерпретации, как и **Perl**. Поскольку это так то для работы программы требуется соответствующий интерпретатор. Это программа. Версия языка может быть важна. На сегодня версия **Python 2.7** является встраиваемой во многие ОС семейства Linux – несмотря на существование более поздних версий. Последние могут не подойти для конкретной программы. И наоборот – программы для версий 3-го выпуска могут не подойти к версии 2. **Perl** версии 5 также часто встроен в ОС, а версия 6 совсем другая.

Немного о BASH.

Wikipedia говорит что **BASH** это популярный интерпретатор, способный выполнять программы, записанные в файл. Заметим, что в среде FreeBSD **BASH** не используется, хотя может быть там установлен. В NetBSD также как во FreeBSD используется **ash**, но может быть поставлен **korn shell**.

Изначально установленный интерпретатор не надо ставить. Он будет работать в консоли и терминале. Вы часто сможете видеть что в интернете предлагают программы для **BASH**, однако они могут не подойти, и быть вредны. Заметим, что в курсы по **BASH** часто по непониманию суют тексты про **AWK**, хотя она совершенно не связана с **BASH**. Часто упоминается руководство под авторством Mendel Cooper, но оно написано плохо, хотя и переведено. Его легко найти в интернете. Далее я дам неполное описание **BASH**. В этой книге описано использование **AWK** в качестве примитивной замены BASH.

BASH обеспечивает приглашение ко вводу команд, проверку команд, выполнение команд, выполнение программ-скриптов с расширением SH.

Выполнение скрипта.

Так как **BASH** уже работает, то его не надо запускать. Для выполнения скрипта с именем **test.sh** из текущего каталога подайте команду

```
./test.sh
```

Выдача строк на экран.

Это делается так. Команда

```
echo "123"
```

выдаст на экран строку

```
123
```

Чтение строки в переменную.

Строка в скрипте

```
read VAR
```

записывает строку, введённую человеком, в переменную **var**.

Выдача переменной на экран.

Переменная **var** выводится на экран так.

```
echo $VAR
```

Создание переменных в скрипте.

Переменные это просто название простых данных – строк и чисел. Обычно переменные обозначают простые виды данных. Смысл переменных в том что их данные обрабатываются в скриптах. Чтобы создать переменную ей надо присвоить данные, то есть значение. Сама переменная есть строка, вернее это имя переменной.

```
VAR2=100
```

Тут мы создали переменную с именем **VAR2** и значением **100**.

Операторы в BASH.

Операторы определяют действия, и сравнивают значения. Есть арифметические, строковые, файловые операторы. Многие из них очевидны, например, плюс. Отметим неочевидные операторы.

%	остаток от деления
==	проверка на равенство
!=	проверка на неравенство
-eq	равно
-ne	не равно
-gt	больше
-lt	меньше
-ge	больше или равно
-le	меньше или равно
!	отрицание
-o	«или»
-a	«и»
-z	длина строки равна нулю
-n	... не равна нулю
str	строка не пуста
-b	блочный файл
-c	файл с символами
-d	каталог
-f	обычный файл
-g	файл имеет бит группы
-k	файл имеет бит исполнимого файла
-p	именованный канал
-t	файл связан с терминалом
-u	запуск файла разрешён
-r	файл читаемый
-w	в файл можно писать
-x	файл исполнимый
-s	размер файла больше 0
-e	файл существует

Пока операторы могут быть непонятны. Многие из них просто проверяют условия.

Условие.

Условное выражение позволяет поставить действие в зависимости от состояния, причём состояние должно описываться чётко как существующее или отсутствующее. Общий вид выражения такой.

```
if [ условие ]
then
    действие
fi
```

Например.

```
a=10
b=20
```

```
if [ $a == $b ]
then
    echo "a is equal to b"
fi
```

То есть 10 не равно 20, поэтому на экран не выдаётся что переменные равны. Если бы были равны то на экране было бы

```
a is equal to b
```

Повторение действия.

Повтор действий постоянно используется в программировании. Общий вид выражения в первом случае таков.

```
while условие
do
    действие
done
```

То есть пока условие верно выполняется действие. Например.

```
a=0
while [ $a -lt 10 ]
do
    echo $a
    a=`expr $a + 1`
done
```

То есть пока переменная меньше чем 10 её печатают на экран. Переменная увеличивается на единицу пока она меньше 10.

Другая форма выражения для повтора такова.

```
for var in значения
do
    действие
done
```

Для каждого значения выполняется действие. Например.

```
for var in 0 1 2 3 4 5 6 7 8 9
do
    echo $var
done
```

Если переменная **var** имеет значения 0...9, то она выдаётся на экран.

Стандартный интерпретатор и его конкуренты.

На UNIX наиболее популярным интерпретатором команд ОС является **BASH**, он реализует не очень сложный, но трудный для восприятия язык программирования для автоматизации, выполняемой непрограммистом. Но популярность в данном случае ничего не говорит о пользовательском качестве программы. Лично я считаю что **BASH** устарел, ему есть смена и альтернативы. Контраргументом является масса готовых текстовых программ для него. При этом – подчёркиваю – наличие готовых программ для **BASH** – или любых других – сделанных кустарно, под конкретную, личную задачу – представляет опасность для вашей работы, и ваших данных. Программа, пригодная и безопасная в одних условиях, становится потенциально опасной и бесполезной в других. Это ни что иное как ловушка для ленивых или

малообразованных людей. Кажется, что надо просто взять готовую программу, и не надо учить язык интерпретатора. Да, это просто. Но, если вы хотите избежать опасностей, то вам потребуется понять эту программу. А если вы её понимаете, то – учитывая сказанное – вам надо писать свою программу – которую вы продумали, и понимаете. Но даже ваша программа всё ещё может работать неправильно. Сам факт её правильной работы в одних условиях ничего не говорит о работе в других.

Так как в UNIX многое можно делать командами, и получать текстовый ответ на них, то часто вопросы автоматизации – это вопросы обработки текста.

Какие есть конкуренты **BASH**?

- Несколько командных интерпретаторов, среди которых наиболее интересным, на мой взгляд, является **FISH**, но он не имеет русскоязычной документации. Перевод документации есть к интерпретатору Korn Shell – **ksh**. Он находится по адресу <https://subscribe.ru/catalog/comp.soft.othos.unixman>.
- Наверное, самый подходящий язык программирования для русских учеников – **PHP**. Легко доступная русская документация, простота языка.
- **Python**. Уже присутствующий в поставке ОС язык, очень популярный, и используемый в школьном курсе. Имеется много статей о нём, и, наверное, учебников, но документация к свежим версиям – на английском. Несколько более сложный, чем **PHP**, но всё же простой язык.
- **Perl**. Также часто уже поставленный язык, по которому, наверное, вы найдёте русскоязычный учебник в интернете, но критикуемый за некий анархизм. Обратите внимание, что 6-я версия языка сильно отличается от предыдущих.
- **Lua**. Простой язык, описанный по-русски, но в бесплатном учебнике много заявлений для профессиональных программистов, и это уводит в сторону и затрудняет обучение.
- Такие несложные языки как **Tcl**, **Rexx**, **Scheme** (последний – например в среде **Guile**). Тут я рискую проявить субъективизм, но мне кажется, что это именно так. Русской документации может не быть, или её надо долго искать. Для **Rexx** есть несколько готовых к печати книг в виде файлов в интернете, но они большие, и на английском. Всё же я думаю что этот язык – посильный для учеников, и эти книги может быть надо выдать на бумагу, и, скажем, иметь в библиотеке, как и англо-русский словарь.

Я пока не видел перевод документации по **BASH**. Для понимания используемых терминов потребуется дополнительная литература. Лучше всего записать словарь или специализированным учебником, например, «Языки программирования» (авторы – Голицына, Партыка, Попов). Именно в этой книге много информации, которая может быть не нужна, и не надо учить этот текст.

Особые случаи и предостережения.

Следующая информация имеет истоки в текстах по традиционным интерпретаторам, но может быть полезна и в других случаях. Общее замечание в том, что, как писали авторы книги «UNIX Haters», бесплатные ОС вначале создавались как очередной «хак». Другими словами – паллиатив, костыль. Этот факт не должен вас пугать, и он не отменяет миллионов машин с этими системами, но важны следствия этого. Что именно? Например, следующее.

- слишком много дистрибутивов
- имена команд не очевидны
- документацию (поначалу) трудно понять
- некоторые команды опасны (это так и в других ОС)
- нет команды для восстановления стёртого файла (но при работе таких оболочек как GNOME для них есть мусорная корзина)
- имена файлов не могут быть такими же как части команды
- команды не выдают удобных, ожидаемых сообщений
- ОС не делалась для рядового пользователя (уже не так)
- почтовые программы несовершенны (уже не так)
- администрирование трудно (а где оно легко?)

Что команды опасны это не повод прятаться под стол. Нужно прочесть документацию к ним без спешки.

Команды и другие средства программирования в UNIX.

Рассмотрим некоторые вопросы программирования. В среде UNIX вы обычно имеете несколько программ, которые можно использовать для автоматизации работ. Такая программа сама не обязательно реализует какой-либо язык. Но вместе с другими программами помогает пользователю решать его задачи. Какие это программы?

cat	соединение файлов и выдача их в стандартный вывод
tac	то же но файл перевернут, выдаётся с конца
nl	показ количества строк в файле
pr	выдача на бумагу
fold	формирование файла со строками заданной длины
head	выдача начала файла
tail	выдача конца файла
split	деление файла
csplit	деление файла на части в соответствии с условиями
wc	показ количества строк, слов и байтов в файле
sum	подсчёт контрольной суммы для файла ⁸
md5sum	проверка контрольной суммы или определение её
sort	сортировка строк
uniq	отчёт о повторах строк или игнорирование их
comm	построчное сравнение файлов
tsort	топологическая сортировка строк
cut	отобрать часть файла или вывода строк
paste	соединение строк
join	то же
tr	замена символов
expand	замена табуляций на пробелы
unexpand	наоборот
dir	показ содержимого каталога
ls	то же
vdir	то же
cp	копирование файлов и каталогов
mv	перемещение файлов и каталогов
rm	удаление файлов и каталогов – не в корзину
ln	создание ссылки файловой системы
mkdir	создание каталога
rmdir	удаление каталога – не в корзину
chown	определение владельца каталога или файла
chgrp	смена принадлежности группе

8 Для проверки файла обычно используется программа md5sum

chmod	смена прав
touch	создание пустого файла
du	определение занятого места каталогом
echo	выдача строк
printf	то же
test	проверка типа файла и прочее
expr	вычисления и манипуляции строками
basename	показ лишь имён файлов и каталогов без путей к ним
dirname	показ имени содержащего каталога
pathchk	проверка верности и краткости путей
realpath	показ абсолютного пути
pwd	показ текущего каталога
who	показ вошедших пользователей
date	показ даты и времени
timeout	старт команды с задержкой
kill	оборвать работу программы
sleep	ждать указанное время
less	показать файл порциями, для чтения
sed	отбор и другие операции с текстом
awk	то же

... и другие.

Программа **head** обрывает поток текста, поэтому при её использовании может быть следующее сообщение.

```
IOError: [Errno 32] Broken pipe
```

Это не опасно.

Каким образом происходит это использование программ? Типично, что интерпретатор позволяет

- писать в файл
- читать из файла
- использовать конвейеры и перенаправление
- применять регулярные выражения

Конвейер и перенаправление – это средство передачи данных из одной программы в другую, или из файла, или в файл. Простейший конвейер не передаёт из одной программы в другую так как содержит всего одну программу. Также может быть выполнена передача данных из командной строки, и выдача туда же. Что выбрать? Когда как. Если надо видеть данные то стоит их показать в командной строке. Когда данных много то их пишут в файл. Регулярные выражения – это небуквальное, условное обозначение множества строк. Для того чтобы входить во множество строка должна быть описана в выражении, обладать формальными свойствами вроде длины, входящих символов и так далее.

Проблемы программирования.

Автор считает – а его практика в этом убеждает – что полезно, и даже может быть необходимо уметь автоматизировать свою работу. Тут я опишу своё впечатление от программирования вообще. Эта тема очень велика. Я понимаю что читатели в массе будут уклоняться от всякого программирования. Это не упрёк. Тем не менее может быть полезно знать насколько проблемно выглядит эта работа. Программированию учат в школе. Есть масса

книг. Но и трудностей. Я хотел бы увидеть что программирование стало просто работой. Но пока это не так. То, что молодые люди идут туда – это свидетельство того что они пытаются сделать из своего увлечения работу. Их родители так не пытались жить. Им приходилось трезво смотреть на жизнь, и работать как придётся.

Но это не всё. Литература по этой теме сложна. Я могу понять что в ней используются неизвестные читателю слова. Но если автор перекладывает свой труд на плечи читателя, то это плохая литература. Но она к тому же и дорога. Я ещё могу простить авторов статей интернета. Но авторы книг должны начать уважать свою аудиторию. Не писать для опытных или гениев, или фанатиков. Перестать писать примеры программ, эпиграфом к которым могло бы быть «а теперь догадайся сам».

В работе программиста есть много огорчений и подводных камней. Молодые не могут правильно относиться к неудачам. В одной книге был поставлен вопрос «а что делать подростку в программировании?». Отвечаю – ничего. Иначе он вырастет дистрофиком.

Сами языки программирования иногда выглядят неадекватными. Вам без конца подсовывают программы на C. У меня нет ни малейшего желания его изучать. Пишут, что это хороший язык для отдельных работ. Жизнь показала что нужно людям. Не заумные и сложные вещи. Не «текстовые редакторы» размером в 500 мб, а PHP. Потому что в нём нет почти ничего лишнего. Несмотря на медленную работу процветает Python. Потому что он прост, потому что он описан на многих языках. Некоторые ответят мне «старый дурак». А я скажу «молодые дураки». Вы не понимаете что язык – лишь средство, что компьютеры, ваша работа – лишь средство. Что однажды вы все можете оказаться на совсем другой работе, и сохранить там чистой белую рубашку не получится.

Глава 22. Базовые операции с файлами и каталогами.

Это операции, которые потребуются всем. Файлы и каталоги расположены либо в оперативной памяти – в случае использования live-носителя, либо на носителе – точнее в его разделе, или партиции (partition). В среде UNIX файлы и каталоги представлены в древовидной файловой системе. В данном случае файловая система – это совокупность файлов и каталогов. Файлы и каталоги в большой степени предсказуемы так как операционная система определяет многие из них. Наиболее непредсказуемыми и интересными в среде UNIX являются каталог **home** и каталоги, зависимые от наличных программ. В среде Windows ситуация та же, но расположение и название каталогов другое, и там нет принятого в UNIX способа подключения носителей и доступных по сети файловых систем. Другими словами – пользовательские личные каталоги и личный каталог администратора являются первыми в плане непредсказуемости их содержимого. Личный каталог администратора в UNIX всегда называется **root**, а личные каталоги расположены в каталоге **home**. Оба этих каталога легко найти – они расположены на поверхности того же раздела на котором есть операционная система. Самый верхний уровень составляет так называемый корневой каталог, обозначаемый как «/». Каталоги называются личными так как привязаны к пользователям, и пользователь распоряжается их содержимым. Любые операции – вплоть до чтения – регулируются правами на эти операции. На семейном или общественном компьютере может быть надо запретить просмотр чужих каталогов. Windows уже работает так. UNIX не позволяет изменять их, но этого может быть мало. Операции с посторонними носителями возможны лишь после их подключения. Личный каталог пользователя может называться как угодно, но всегда можно для простоты обозначать его как тильду – «~». Название каталога при этом не ясно, но ясно что это личный каталог пользователя. Тильда не является названием каталога.

Ниже представлены программы или команды UNIX, и их назначение.

просмотр содержимого текущего каталога	ls
переход к другому каталогу	cd
создание каталога	mkdir
создание файла	touch
стирание файла или каталога	rm

перемещение файла или каталога	mv
копирование файла или каталога	cp
переименование файла или каталога	rename
назначение прав для файла или каталога	chmod
просмотр содержимого текстового файла	cat, more
правка содержимого текстового файла	vi, nano
определение суммарного размера файлов в каталоге	du

Запуск программы происходит путём обозначения её головного файла, или он происходит автоматически. Но на этой операции сказывается общее положение о том где должны быть программы. Поэтому, например, две программы с одинаковым названием головного файла имеют разный статус. Операционная система считает что это файл в каталогах, описанных в переменной **PATH**, и именно там ищет. Напротив, файлы данных предполагаются в текущем каталоге. Чтобы указать что файл находится в конкретном месте надо буквально указать путь к нему. Чтобы указать на текущий каталог, достаточно написать перед именем файла «./».

Просмотр и правка нетекстовых файлов возможны, но программы для этого может быть потребуется ставить.

Разумеется, те же операции можно выполнить не этими командами, а с помощью оконных программ – если они есть, и работает оконная среда.

Скорость копирования и перемещения файлов и каталогов

Некоторые ученики могут заметить что в одних случаях копирование и перемещение выполняются почти моментально, а в других, наоборот, происходят очень медленно. Хуже всего что этот факт может вызвать тревогу или страх из-за подозрения о том что техника не в порядке или что-то делается неправильно. Я не могу утверждать что конкретно ваша техника или программы находятся в каком-либо состоянии, но могу объяснить этот факт. Скорость копирования и перемещения зависит от нескольких обстоятельств.

- происходит ли процесс в рамках одного логического диска
- участвуют ли в процессе ссылки файловой системы
- какова скорость передачи данных конкретным способом
- нужно ли выполнять какие-либо операции при идущем процессе
- достаточно ли мощности техники и мешают ли другие процессы

Типично, что логический диск - по крайней мере на жёстком диске - имеет невидимую учеником таблицу разбиения файлов (file allocation table, FAT). На первый взгляд эта таблица не нужна. Но она существует. И приносит пользу. Когда вы копируете или переносите файл или каталог, то фактически просто переписывается эта таблица, и это происходит очень быстро. Если копирование или перенос происходит с переносом данных на другой логический диск, то этот способ не работает.

Ссылки файловой системы легко перепутать с реальными файлами и каталогами. Так как символическая ссылка имеет очень небольшой объём, то её перемещение или копирование происходит очень быстро.

Скорость передачи данных в большой степени зависит от аппаратуры, и линий связи. Также загруженная данными линия связи будет медленно работать. Если нужно быстро перенести данные с одного жёсткого диска на другой то стоит – при выключенном стационарном компьютере – подключить оба диска к шлейфам и проводам питания этого компьютера, и произвести копирование или перенос между ними.

При копировании или переносе могут возникнуть новые технические задачи из-за наличия файлов или каталогов с теми же именами. Они могут уничтожаться, переписываться, дописываться, пропускаться или переименовываться.

Мощность техники не очень важна при копировании и перемещении. Но явно медленный компьютер всё же может заметно отставать. Наличие других программных процессов также может замедлять копирование или перенос данных.

Символические и жёсткие ссылки файловой системы

Ссылки файловой системы – это «отражения» файлов и каталогов, наблюдаемые при просмотре файлов и каталогов программами при условии что эти программы не игнорируют ссылки. Ссылки интересны потому что позволяют почти не расходуя место на носителе размещать один и тот же файл и каталог в нескольких местах, или – используя жёсткие ссылки – моментально «копировать» большие файлы. Пользователи Windows, Android долго обходились без ссылок файловой системы, а в среде Android, насколько мне известно, ссылки не создаются даже сейчас. Тем не менее полезно знать какие есть виды ссылок, и как их создавать.

Для создания ссылки в среде UNIX есть штатная программа `ln`. Она не требует прав администратора. Общий вид команды для создания ссылки таков.

```
ln опции файл-источник файл ссылки
```

Если мы подадим команду

```
ln -s file link
```

где `file` и `link` обозначают файл и ссылку, то получим символическую ссылку к файлу. Штатная команда `ls` покажет её так.

```
lrwxrwxrwx 1 user user 4 апр 9 16:24 link -> file
```

То есть просмотревая команда с ключом `-l` показывает ссылку как часть файловой системы с первым символом строки `l`. Если файл, на который указывает символическая ссылка, будет переименован, или перемещён, или удалён, то ссылка не будет указывать на него. Она станет бесполезной.

Жёсткие файловые ссылки создаются проще, так.

```
ln file link
```

Они имеют следующие ограничения и особенности.

- Работают только в пределах одной файловой системы
- Нельзя сослаться на каталоги
- Имеют ту же информацию `inode` и набор прав что и у исходного файла
- Права ссылки изменяются при изменении прав файла
- Можно перемещать и переименовывать, и даже удалять файл без вреда ссылке

Удаление ссылки выполняется просто, командой `rm`.

```
rm link
```

Рекомендуется попрактиковаться в операциях со ссылками.

Глава 23. Поиск файлов, каталогов и текста.

Каждый пользователь должен уметь искать файлы, и строки в файлах. Зачем? Пока вы имеете один-два файла, вам легко в них ориентироваться. Вы помните что это такое. Но не когда их много. Кроме того, сами файлы могут быть большими. Не идите на поводу у привычки листать тексты. Вы можете пропустить нужный текст.

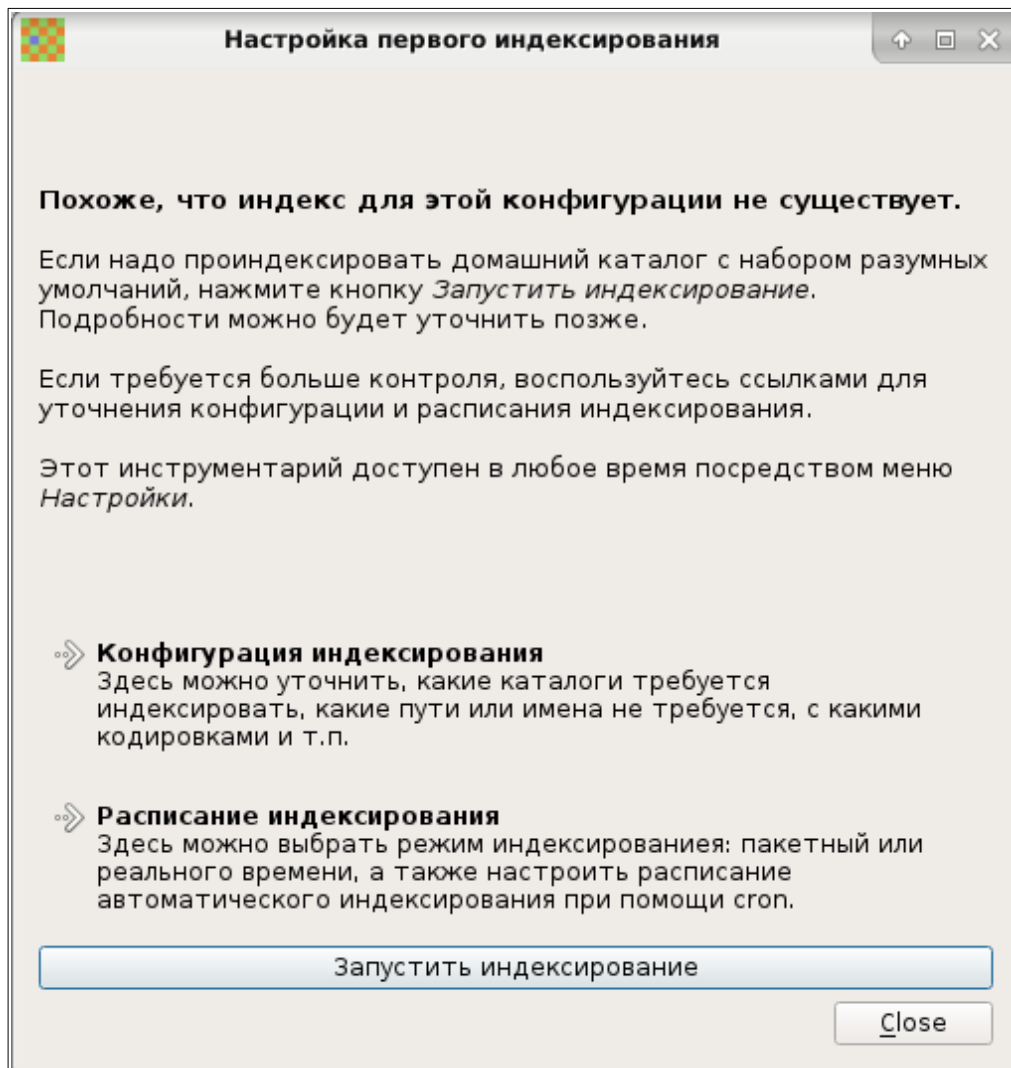
Лучше использовать программы. Первые такие программы – для поиска текста – были созданы очень давно. Что это за программы? Это, например, `recoll`. Эту программу может быть надо ставить так как её нет после установки ОС. Зачем такие сложности? Чтобы искать разные *формы* слова одновременно.

Recoll.

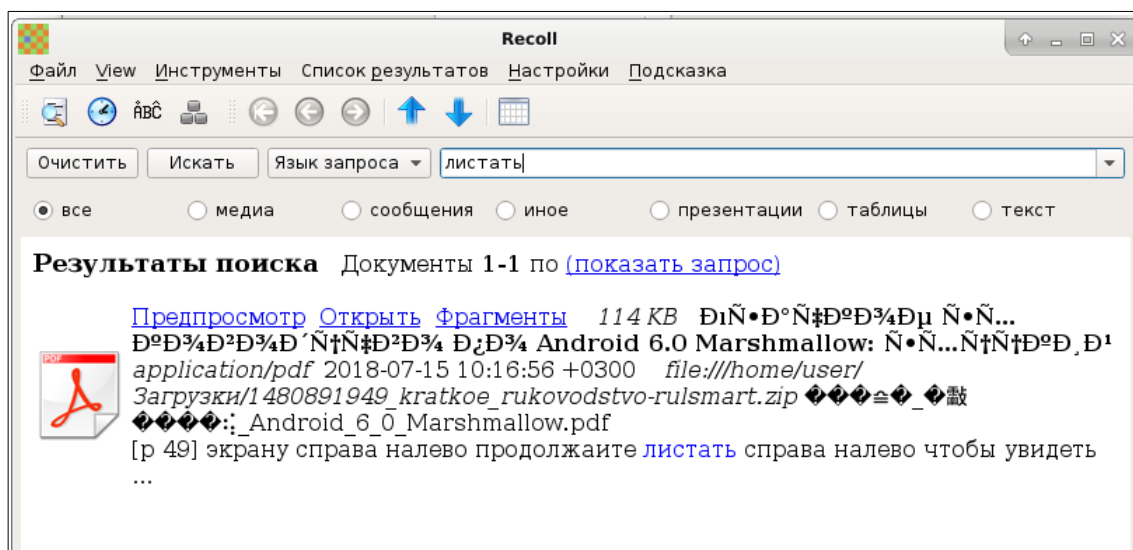
В случае ОС Debian 9, и, скорее всего, у вас, после установки программы следует проиндексировать файлы, то есть создать базу данных для поиска. Для этого подайте команду

```
recoll
```

Будет показано окно для индексирования (тут рассматривается случай с наличием оконной среды).



Нажмите кнопку «Запустить индексирование». На первый план выйдет второе окно этой же программы.



В нём я набрал слово «листать», нажал **Enter**, и получил один документ с этим словом. Программа ищет не по всей файловой системе.

Если надо использовать программу без оконной среды то для индексирования подайте

команду **recollindex**. Для поиска – команду **recoll -t**. Например так.

```
recoll -t UNIX|head
:3:common/rclinit.cpp:308::Configuration directory: /home/user/.recoll
Recoll query: Query((unix OR unix#10))
1513 results
text/plain [file:///home/user/Загрузки/linux_man_pages.zip] [apparmor.d.5.html.txt]
105262 bytes
...
```

То есть программа нашла 1513 вхождений искомого слова «UNIX» и выдала имена файлов с этим словом. Первое имя – **apparmor.d.5.html.txt**.

Может потребоваться собрать именно имена файлов. Для этого можно использовать что-то вроде **AWK**. Имена можно либо использовать сразу, либо записать в файл. Также можно упростить вывод программы так.

```
recoll -t -b UNIX|more
:3:common/rclinit.cpp:308::Configuration directory: /home/user/.recoll
file:///home/user/Загрузки/linux_man_pages.zip
file:///home/user/Загрузки/linux_man_pages.zip
...
```

Так как в данном случае показан лишь файл архива то он указан много раз. В нём находятся файлы где найден текст. Чтобы убрать повторы строк можно фильтровать вывод поиска так.

```
recoll -t -b UNIX|sort|uniq|more
```

Можно сразу затребовать показ содержимого файла с текстом.

```
recoll -t -b -d UNIX|more
:3:common/rclinit.cpp:308::Configuration directory: /home/user/.recoll
file:///home/user/Загрузки/linux_man_pages.zip
[IMG]
  Меню Menu Close
      [сброс] [подчиняются]
Переход к основному содержанию
```

```
Если: ошибка apparmor_2.13.3-5ubuntu1_amd64
```

```
ИМЯ
```

```
apparmor.d - синтаксис профилей безопасности для AppArmor.
```

Но в таком случае вы получите весь файл на экран, и он может быть искажён. Чтобы улучшить результат или сократить работу можно использовать что-то вроде следующего.

```
recoll -t -d UNIX>poiskunix
```

Тут мы создаём файл **poiskunix** с результатами поиска. Но он может не годиться для автоматической работы или быстрого просмотра. Его можно фильтровать так.

```
awk '(/^text/)|(/UNIX/)' poiskunix|sed 's/^text/\n\n/p'|more
```

Имя файла будет показано дважды – фильтр можно сделать лучше. Фильтр показывает только строки с «text» в начале строки или «UNIX». Этого может быть мало для понимания содержимого файла.

Grep.

Другая подобная программа не столь наглядна, но уже есть в поставке вашей ОС. Это **grep**. Так как эта программа не склоняет и не спрягает слова, то нужно подумать что именно искать. Такая программа будет искать буквально то что вы запросите. Например, для поиска слова «xetex» в файлах с расширением «tex» в текущем каталоге надо подать команду

```
grep xetex *.tex
```

Поиск может быть ограничен из-за важного обстоятельства – свойств файлов. Во-первых, файл должен содержать текст. Во-вторых, программа должна понимать кодировку текста. Поиск показанным образом лишь для текущего каталога, но не его подкаталогов.

Какой программой искать имена файлов и каталогов.

Прежде всего на UNIX это программа **find**. Это консольная программа. Всё было бы хорошо если бы не бестолковая документация. Выйти из положения достаточно просто – записать имена в файл, и там искать. Как? Если мы хотим записать все имена в текущем каталоге, то следует подать команду

```
ls -R>names
```

В данном случае **names** – это произвольное имя файла, куда будут записаны имена. Прежде стоит убедиться что вам не нужно содержимое этого файла так как оно будет переписано новым содержимым. Или ещё проще

```
ls -R *.txt
```

– ищем все файлы в текущем каталоге с расширением «txt». Если есть каталог с таким же названием то он тоже найдётся.

Ещё есть много других программ. **grep** ставится при установке системы. Его можно использовать так.

```
cat file|grep «123»
```

В данном случае мы ищем «123» в файле с именем **file** в текущем каталоге.

Если у вас медленный компьютер.

Программа **locate** штатная, она ищет быстро, но по своей базе данных, которая составляется при включении компьютера. Чтобы программа прямо сейчас учла нынешние файлы в базе надо подать команду **updatedb**.

Определение размера содержимого каталогов.

Для текущего каталога будет работать это.

```
du -d 1 -h
```

Пример поиска по свойствам файла.

Можно использовать **find**, но мне всегда казалось что авторы таких программ и их документации надо мной издеваются. Вместо этого можно использовать **AWK**. Команда **ls** даёт много информации о файлах и каталогах. С точки зрения **AWK** она выдаёт строки, разделённые табуляцией. Значения этих строк можно описать. Это либо набор букв, либо цифры. Цифры можно отбирать как большие, равные или меньшие заданной. Символы могут быть равны или не равны известным, возможным. Благодаря этому можно отбирать строки. Например так.

```
ls -l|awk '$5>33333'|more
```

Так мы отобрали файлы, размер (5-я колонка) которых более 33333 байт, то есть (33333/1024=32.5) килобайта. А так мы отберём файлы, размер которых в диапазоне величин.

```
ls -l|awk '($5>33333)&&($5<99999)'|more
```

Теперь отберём лишь файлы (и каталоги), имя которых содержит строку **.odt**.

```
ls -l|awk '/\..odt/'|more
```

Пришлось ставить перед точкой «\» чтобы точка понималась как часть искомого текста. Поиск показанным образом лишь для текущего каталога, но не его подкаталогов.

Поиск по MIME.

Обычно так никто не ищет, но в отдельных случаях это может быть нужно. MIME – это таблица, описывающая соответствие форматов файлов их типу, понятному пользователю. Благодаря этому можно отбирать файлы. Например так.

```
ls -l -Q|awk '/^-*/{print $9}'|xargs file -i|awk '{gsub(":", ""); print $1"\t\t"$2}'|head
```

Тут пришлось сделать больше чем хотелось:

- взять лишь имена файлов
- окружить их кавычками чтобы части имён не казались отдельными файлами
- имена подать программе **file**
- отобрать лишь имена файлов и двоеточия, которые ставит **file**
- выкинуть из этих строк двоеточия

- разделить имена файлов и типы двумя табуляциями

xargs выдаст предупреждение, но ничего лучше пока у меня нет. Чтобы посмотреть типы файлов в подкаталогах, подайте эту команду.

```
ls -l -Q -R|awk '/^-{print $9}'|xargs file -i|awk '{gsub(":", ""); print $1"\t\t"$2}'|head
```

В вашей файловой системе обязательно есть файл или файлы где прописаны типы MIME. В Debian это файл **/etc/mime.types**. Прочтём его.

```
cat /etc/mime.types|more
```

и увидим

```
application/msword doc dot
```

Это значит что файлы с расширением **doc** или **dot** относятся к программе Word, которая не поставляется для UNIX, но можно обрабатывать эти файлы там.

Типы, описанные в MIME могут поначалу ничего не говорить ученику. Стоит посмотреть описание форматов на www.online-convert.com.

Отбор некоторых файлов и каталогов.

Нужные по разным причинам файлы и каталоги могут быть разбросаны куда попало. Такие оконные программы как **Thunar** очень похожи на Проводник Windows и позволяют выполнить нужную работу пока она не слишком велика или сложна. В противном случае можно запутаться и создать мигрень. В худшем случае ученик стирает ценные файлы. Оконная программа поиска файлов может быть слишком примитивной.

Вопрос об отборе сложен. Критериями отбора могут быть, например, права, или глубина вложенности каталога. Критерии могут быть сложными. Программа **find** умеет создавать список файлов с путями к ним, а программа **basename** умеет отделять имя файла в этих строках. Программа **realpath** возвращает приведенное к обычному виду полное имя файла с путём к нему. Программа **tree** показывает иерархию каталогов и файлы, выводя их имена в консоль или терминал. Также она выводит изображение связей в виде «дерева». Поэтому может потребоваться убрать эту графику, которая на самом деле, по-видимому, сложена из символов «|» и «-». Когда вы знаете имя файла или каталога, то показ сведений о нём можно организовать так.

```
ls -lh|grep XXXX
```

где XXXX – имя файла или каталога.

Глубину вложенности можно определить как количество разделителей каталогов минус один так как полный путь начинается с «/». Вам могут помочь регулярные выражения – см соответствующую главу. **AWK** может оказаться необходима для отбора файлов и каталогов. Автор не может описать все возможные рецепты и ситуации так как их очень много. Будем рассуждать логически. Кроме «классических», обычных файлов и каталогов есть ссылки файловой системы. Кроме того есть вероятность появления проблемных символов в именах файлов и каталогов. Прежде всего это пробел. Поэтому придётся окружать имя кавычками. Но они сами могут входить в имя. Имена файлов и каталогов могут повторяться, но не в одном каталоге. Запрет на повтор имён можно использовать. Размер символической ссылки может быть не равен размеру файла или каталога на который указывает ссылка. Также размер каталога не равен его содержимому – такова реальность UNIX. Создаваемые по мере отбора файлы сами входят в какой-то каталог и множество файлов, и меняют суммарный объём. Поиск текста в файлах, отличных от текстовых, выполняется программно, иначе, чем если бы файл был текстовым. Может быть важен номер вхождения текста в файле, или регистр, или строка. Программа, конвертирующая документы, может их исказить. По сути отбор часто есть манипуляция с текстом. Но документы форматов PDF, DOC, RTF, DJVU и многие другие не являются текстом.

А также...

Текстовые редакторы и браузеры умеют искать текст.

Поиск текста в звукозаписях и фильмах пока возможен только при использовании интернета.

При любых проблемах с поиском прежде всего узнайте лучше используемую программу.

Глава 24. Сравнение каталогов и текстов.

Может возникнуть потребность сравнить каталоги или некоторые файлы. Хотя, насколько я могу судить, обычно этого никто не делает. Программы для сравнения файлов созданы давно. Простейшая такая программа называется **diff**. Она не способна сравнивать полезным образом ничего кроме текстовых файлов. То есть она бесполезна для сравнения документов в формате ODT, и подобных. Если вам надо сравнить, например, содержимое электронной таблицы, то можно либо сохранить её как текстовый файл, либо сравнивать другими программами.

Diff работает в консоли или терминале. Она способна находить различия между файлами, и показывать их как строки. С помощью этой программы можно делать патчи, то есть «заплатки» для других файлов – которые надо изменить. Каждая строка рассматривается отдельно, и может

- совпадать со строкой в другом файле
- отсутствовать в одном из файлов
- быть изменённой строкой

Также есть средства для сравнения файлов, требующие оконной среды. Это, в том числе, программа **Kompare**, которая умеет даже сравнивать каталоги.

Глава 25. Перевод англоязычных текстов.

Компьютерные устройства в большой степени связаны с США, и английским языком, который на сегодня является языком международного общения. Обратите внимание – аннотации книг для иностранцев пишут на английском. Интернет даёт вам выход в интернациональную среду, но и там самым общепонятным языком является английский. Это не значит что вы всегда должны писать на нём. UNIX изначально развивался в США. В интернете много русскоязычных текстов, но типично, что уже написанный на английском текст никто не переводит, и достаточно часто он вам нужен. Также надо быть готовым к тому что инструкция, например, к материнской плате будет на английском.

В СССР английский язык поздно начали учить в школе. Насколько трудно выучить английский? Тут нужно уточнить. Вам достаточно понимать в основном конкретные тексты, которые не являются жаргонными или художественными. Часто эти тексты содержат много аббревиатур, но их словарный запас мал. Не считая именно компьютерной терминологии он примерно равен нескольким тысячам слов. Как правило слово в тексте не является результатом словообразования, или одной из форм слова или понятия. Сочетания слов часто обозначают разные понятия, но сами слова в английском не имеют форм. Не нужно зубрить словарь. Английский язык может без усилий войти в вашу жизнь по мере чтения текстов.

Тем не менее англоязычные тексты, и перевод важны, и являются реальным, и большим препятствием на пути ученика. Самым понятным способом перевода является чтение словаря на бумаге. Словарь надо выбирать, а не хватать самый дешёвый или с самым крупным шрифтом. Надо быть готовым к чтению словаря компьютерной терминологии, который будет отдельной книгой. Это может быть, например, словарь авторства Дорот и Новикова, размером в 890 страниц. Каждая библиотека должна иметь и англо-русский, и словарь компьютерных терминов. Библиотекари должны это обеспечить, а не думать что это их не касается.

Также мной в основном из потенциально нужных англоязычных текстов составлена готовая к печати бесплатная публикация «Аннотированная документация UNIX. Дома без компьютера читай и переводи» размером 250 страниц. Её можно скачать с сайта pkprovincia.ucoz.ru. Как и в случае других книг, вам не надо заранее читать все тексты. Тексты читаются по мере надобности.

Там же находится составленный мной неспециализированный словарь именно для перевода компьютерной литературы. Он бесплатен, и имеет размер 613 страниц.

Кроме того, там находятся эти бесплатные книги.

название	Размер, страниц
Словарь компьютерной терминологии	185

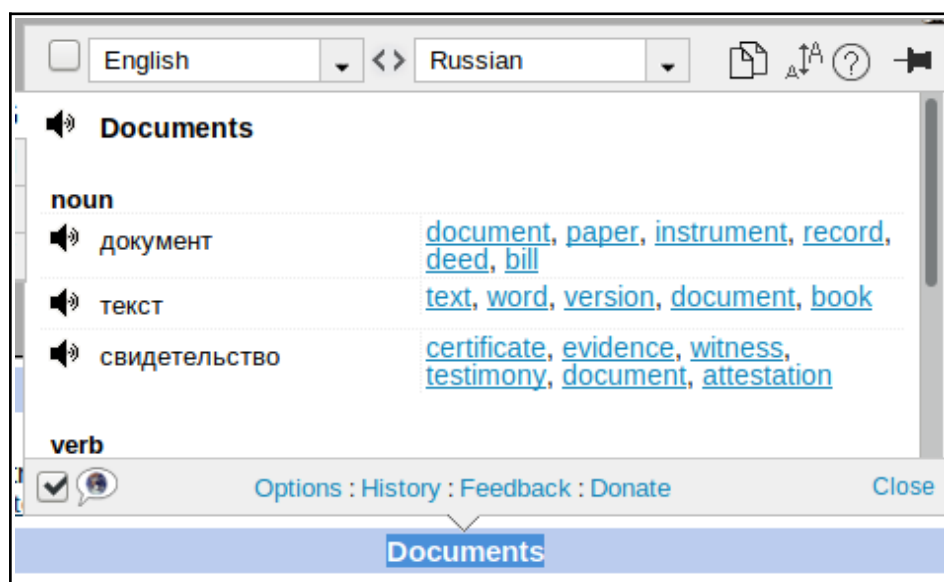
Англо-русский общий словарь (8000 слов)	74
Англо-русский словарь для перевода компьютерной документации (довольно мелкий текст)	94
Linux Dictionary Толковый словарь операционной системы Linux (мелкий текст)	849

Сервис LITRES позволяет покупать в том числе англоязычные книги, и это может быть дешевле чем покупка книг на бумаге, и быстрее чем их доставка в библиотеки обычным образом.

Традиционный перевод с помощью словарей – потенциально самый качественный, но и самый медленный. Как ускорить чтение и перевод? Для этого есть несколько путей.

Перевод в браузере.

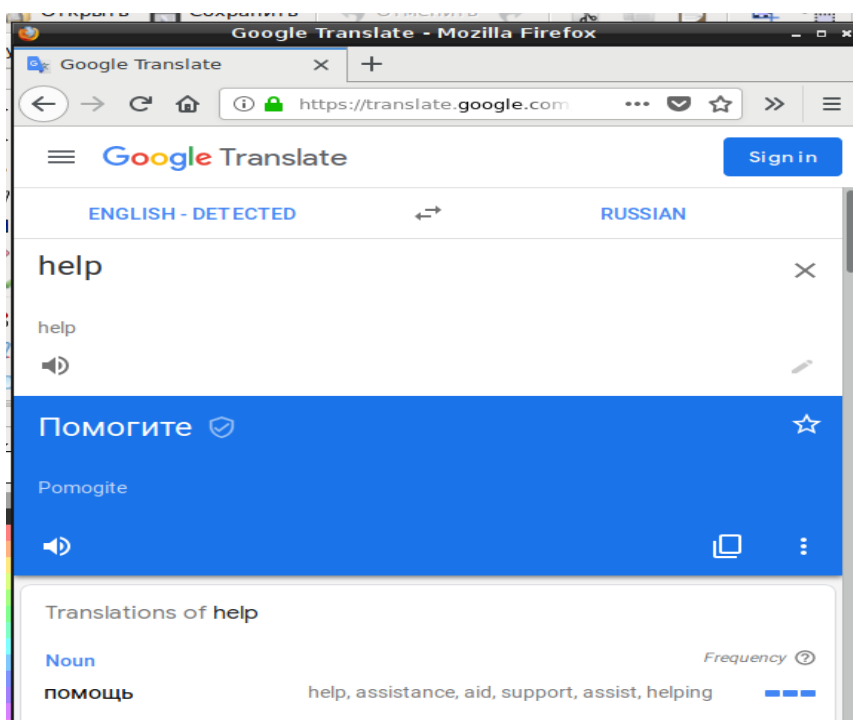
Предполагается, что ученик получил из интернета в полноценный браузер – оконную программу – страницу со словами, которые он хочет перевести. Та же страница, взятая не из интернета, может не обслуживаться. Обычно сам браузер не обеспечивает перевод. Но он позволяет на компьютере поставить *расширение* для перевода. Расширение (extension) может требовать настройки, и выхода в интернет для своей полезной работы. В простейшем случае достаточно выделить текст для перевода, и он будет переведён.



После выделения слова оно становится окрашено, и появляется карточка с переводом. Щёлкните **Close** когда вам надо будет убрать карточку.

В телефонной среде такая возможность может отсутствовать.

Также есть возможность переводить как отдельные слова, так и большие куски текста без установки расширений. Часто это делается с помощью переводчика Google, расположенного по адресу translate.google.com. Первоначально переводчик переводит на английский. Поэтому щёлкните **Russian**. Вы можете либо вставить текст из буфера обмена, либо взять из постоянной памяти вашего устройства, либо набрать. Пример работы переводчика.



Как и многие другие переводчики связного текста, а не отдельных слов, этот переводчик интеллектуален, и поэтому напоминает человека-переводчика. Тем не менее надо помнить что перевод может быть неверным. Показанный переводчик требует выхода в интернет и полноценного браузера.

Заметьте – PDF-документ может быть показан в браузере, но он не является веб-страницей, и любой переводчик может быть не способен его перевести. Это ещё менее вероятно для PS-документов. В браузере Chrome на сегодня можно переводить PDF с помощью расширения «Многоязычный переводчик PDF/HTML/TXT» авторства Bohdan Stasiuk. PS-документ можно конвертировать в PDF.

Также стоит попробовать сайт docplayer.ru. Он показывает PDF-документы, но там работают расширения для перевода.

Как установить расширение для перевода.

Расширение (extension) для перевода существует как модуль к конкретному браузеру, и таких расширений несколько. Установка расширения не требует административных прав, и не делается менеджером пакетов. Рассмотрим установку расширения для браузера Firefox как самого популярного в среде Linux. Перейдите на страницу с адресом https://addons.mozilla.org/ru/firefox/addon/to-google-translate/?utm_source=addons.mozilla.org&utm_medium=referral&utm_content=search , введя этот адрес, или с помощью поиска в любой поисковой машине. Возможно, именно этот адрес будет неверен, но поисковая машина точно найдёт эту страницу. Страница имеет большую кнопку для установки расширения.

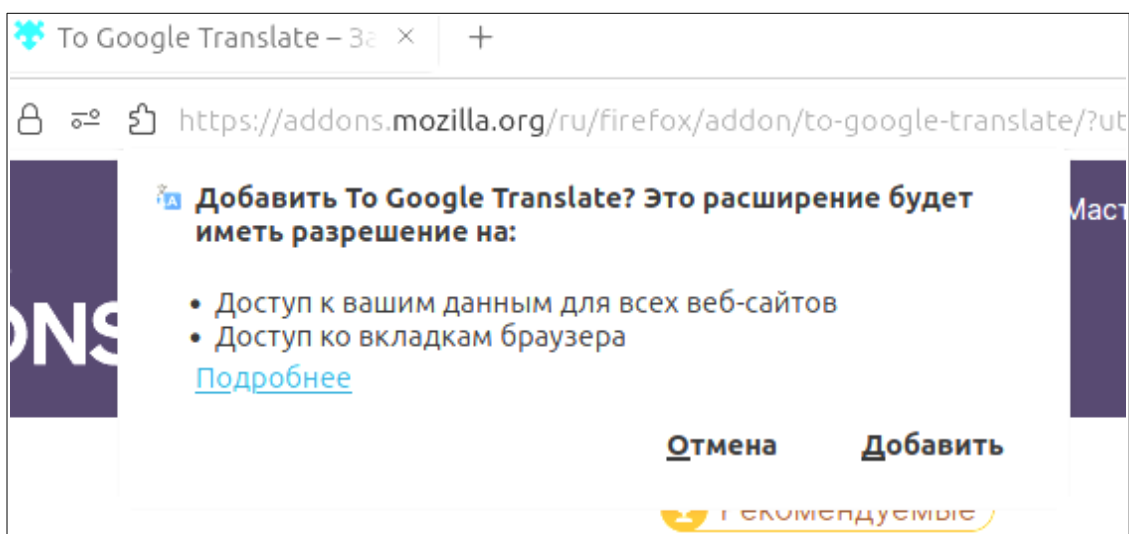
To Google Translate

от Juan Escobar

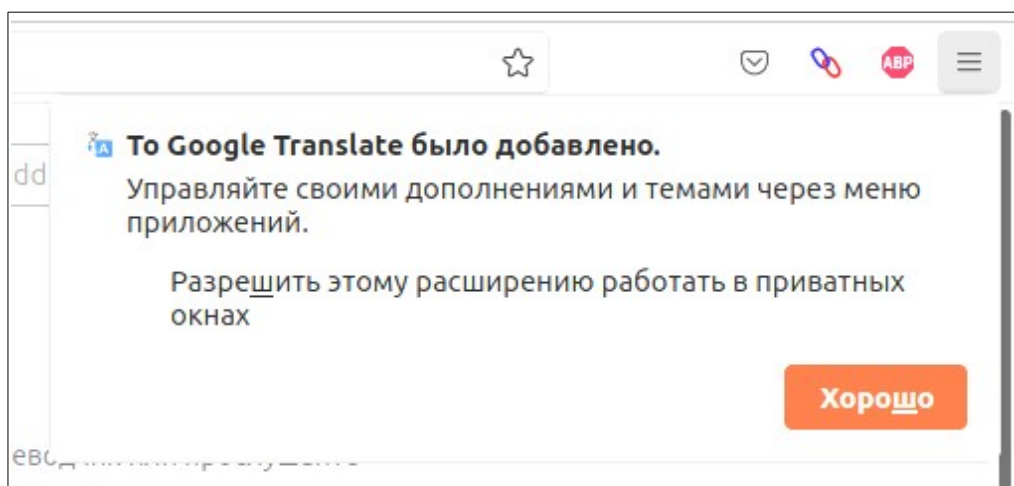
Right-click a section of text and click the Translate icon next to it to text translate or listen to it in your language.

[Добавить в Firefox](#)

Когда вы нажмёте голубую кнопку под адресной строкой возникнет запрос, где надо щёлкнуть кнопку Добавить.



Затем браузер сформирует сообщение об установке расширения.



Тут есть возможность разрешить перевод в частных окнах. Это не очень важно, и можно не ставить галочку для этого.

Теперь зайдём на сайт газеты New York Times и переведём главную страницу нажав **Ctrl+Shift+3** (то есть одновременно). Вот что получилось.

Google Переводчик английский (определен автоматически) → русский


пятница, 14 января 2022 г.
Сегодняшняя газета

Еще New York Times

Мир нас Политика Нью-Йорк Бизнес Мнение Технология Наука Здоровье Виды спорта Искусство Книги Стиль Еда Путешествия

РЕАЛЬНЫЙ
Больницы приближаются к заполнению в 2 дюжинах штатов по мере распространения Omicron

- Чрезвычайно заразный вариант подпитывает волну коронавируса,



На сегодня это, видимо, лучший способ переводить страницы в браузере. Тем не менее нужно учить английский язык.

Перевод в консоли или терминальной программе.

Перевод в текстовой среде плох тем что исходным материалом для перевода является текст, который не равен содержимому страницы публикации. Программы из состава утилит Попплера умеют вытаскивать текст из PDF-документа, но не PS-документа. Полученный текст можно перевести с помощью программы **sdcv**, но для правильного понимания публикации может быть нужно её видеть. Для этого требуется либо иметь оконную среду, либо фрэймбуфер. При фрэймбуфере есть возможность, например, выводить на экран файлы изображений, полученные из страниц PDF или PS документов с помощью программы **imagemagick**.

Программа **sdcv** переводит лишь отдельные слова – даже если вы их вводите вместе для перевода. Интересно переводить весь файл с текстом сразу. Это можно сделать автоматически подавая слова из файла в **sdcv**. Подавать их вручную не надо, используйте что-то вроде **AWK**. Заметим, что переводчик не различает просто слова на иностранном языке и команды, являющиеся этими словами. Поэтому часто перевод статей о бесплатных ОС может сбивать с толку.

Ещё одной из многих программ для перевода является **translate-shell**. Она обращается к интернет-сервисам перевода, поэтому нужен выход в интернет. Готовой версии программы для Debian 9 я не нашёл. Наиболее вероятный работающий способ установки описан на <https://github.com/soimort/translate-shell> и использует программу **git**. После установки программы вы сможете подавать одно слово, либо несколько слов, либо целый файл для перевода. Для получения одним действием перевода целого текстового файла подайте команду

```
trans file://input.txt>perevod.txt
```

В данном случае **input.txt** и **perevod.txt** – это произвольные имена файлов, первый из которых переводит программа, а результат перевода пишется во второй файл. Этот второй файл будет переписан если будет существовать до перевода. В данном случае надо указывать положение исходного файла с указанием протокола, и точно. На большинстве компьютеров с UNIX есть каталог **home** с личными каталогами пользователей. Таким образом, команда будет чем-то вроде

```
trans file://home/andrey/perevody/input1.txt>perevod1.txt
```

Тут предполагается, что файл в каталоге **perevody**, а он в личном каталоге **andrey**. Для перевода man-страницы её можно конвертировать в текстовый файл так.

```
man | col -b > filename.txt
```

Это общее обозначение. Конкретнее, если вам, например, надо создать файл из страницы о программе **elinks**, то надо писать так.

```
man elinks|col -b>elinksman.txt
```

Глава 26. Обработка текста. Вывод публикаций на бумагу.

Возможно, некоторым читателям покажется странным обсуждение, да ещё подробное, этой темы. Есть такое мнение что надо «просто подать команду». Это что-то вроде «как дать по этому компьютеру» чтобы он делал что надо. А что делал? Невежественные люди могут быть поставлены в тупик таким вопросом. «Что надо. (???)». На самом деле компьютеры часто используются для создания публикаций, и для передачи их. Тут надо объяснить зачем это нужно. Сейчас уже ни одна страна не выпускает печатные машинки, или они очень дороги. Даже если у вас их 100 штук, всё равно это гораздо худший, отсталый инструмент. Кстати, если машинка есть, а денег на покупку компьютера нет, то это — гораздо лучше чем ничего.

Под публикацией можно понимать разное. Часто это готовый к печати файл формата PDF или PS (то есть PostScript). Оба они прекрасно подходят для вывода на бумагу. Что это значит? Прежде всего они точно соответствуют печатному результату. Программы для просмотра этих файлов на экране почти всегда уже есть, их не надо ставить. В этих программах вы увидите публикацию в том же виде как на бумаге. Огромное преимущество компьютера и планшета перед телефоном в том что вы можете не напрягая зрение читать документы полиграфического качества. Форматы PDF и PS никак не ограничивают содержимое, отображаемое на бумаге, как и формат DVI. Это не значит что каждый такой файл является готовым к публикации с точки зрения его содержания или специалиста по типографским нормам, но эти файлы позволяют показывать на экране и печатать публикации полиграфического качества. В компьютеризованном мире каждый пользователь может создавать такие издания.

Также под публикацией — только с меньшими основаниями — можно считать, например, страницу сайта. Определённо публикацией можно назвать файл формата EPUB. Также файлы форматов ODT, WPS, DOC, DOCX, RTF можно назвать публикацией, но они к тому же редактируемы — при наличии соответствующих программ.

Просто текстовые файлы — без иллюстраций, состоящие только из текста — я бы не назвал публикацией. Их можно редактировать с помощью примитивного, или более сложного текстового редактора. Это программа. В Windows в качестве таковой выступает штатный **Блокнот**. В UNIX может быть редактор **vi**, но он непригоден для обычного пользователя так как требуется учиться перед его использованием. Вместо него часто есть более понятный и более простой редактор **nano**. Также есть много других редакторов. Часто они требуют оконной среды. **Nano** и **vi** работают в консоли, и в терминале. На Android, скорее всего, уже есть нехитрый текстовый редактор. Также стоит упомянуть консольный редактор **mcedit**.

Какой нам интерес обсуждать публикации, и текстовые файлы? Мы, что, журналисты или учёные? Я вам скажу больше — некоторые из нас и свою работу толком не знают, за всю жизнь после школы написали только пару открыток, или известное заклинание на заборе. Для того чтобы вырезать на берёзе «здесь был Вася» не нужна даже печатная машинка. Тем более это трудно сделать компьютером. Примерный ответ таков: интернет стал площадкой для не подлежащих цензуре публикаций, и количество *авторов* книг на сегодня гораздо больше чем в СССР. Книги делаются на компьютере, причём не требуется мощный компьютер. Кроме того, на работе много безбумажных документов, а печатаемые часто надо править или набирать целиком в компьютере. Текстовые файлы в бесплатных операционных системах часто регулируют работу программ, и тогда они не считаются публикациями, но важнее, что текстовые файлы являются основой публикаций, и, иногда, основой документации на работе. Любая работа может включать хотя бы чтение документов. Работа инженера определённо связана с документированием результатов его работы. То же верно про многие другие профессии.

На сегодня в общем создание публикаций распалось на три технологии:

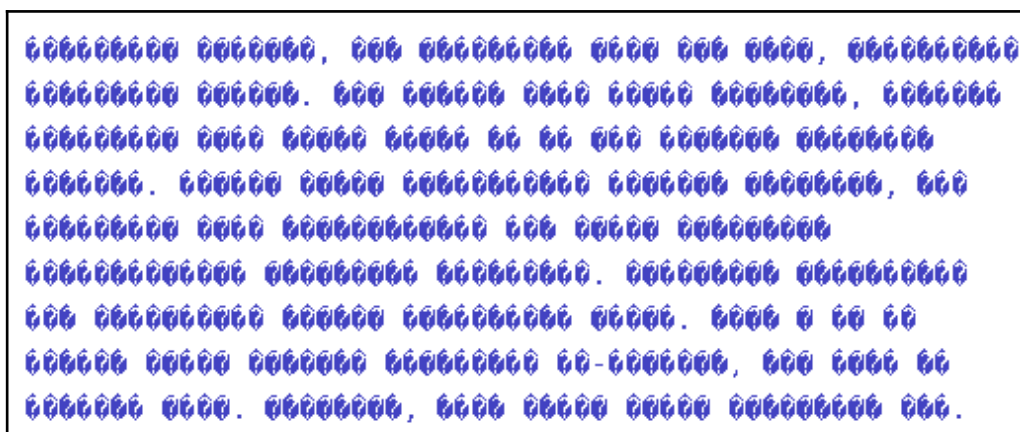
- на основе легковесных языков разметки (LML). Это способ, предполагающий знание

нехитрых обозначений свойств и частей документа, которые прекрасно описаны и легко запоминаются. Затем с помощью некой программы, или нескольких программ, из текстового файла создаётся более похожий на книгу документ (в частном случае страница сайта). Этот способ не подходит для сложно устроенных публикаций, но они часто и не нужны

- на основе таких программ как Word, Writer из Libre Office, Google Docs, AbiWord. На Android есть, например, офисный пакет программ WPS Office. В этом случае вы постоянно видите на экране редактируемый вами документ, и он показан так же как может быть напечатан. Этот способ не годится для очень старых компьютеров, и он может не подойти для научной работы, и вообще особых случаев. Например, если вам надо использовать в документе некие уже существующие статистические данные, и так далее. Заметим, что программа Scribus предназначена для публикаций на бумаге, и она бесплатна
- на основе Latex. На сегодня этот способ несколько старомоден, но позволяет дать в руки учёным и простым пользователям инструментарий для гибкой обработки текста и подготовки самой разной продукции полиграфического качества. Попутно можно заметить что для создания документов таким путём можно использовать легковесную разметку, упомянутую выше. В Latex собственный язык разметки более сложен. Существуют программы для уменьшения потребности запоминать способы разметки. Они сами вставляют «команды» для составления понятного Latex (или xelatex) текстового документа. Latex (или xelatex) обрабатывает текстовый документ. Latex создаёт — в классической технологии — файл формата DVI. В более поздних, уже прижившихся разработках, создаётся файл формата PDF. Всё это можно просмотреть на экране, и проверить результат. Установка и настройка основанной на Latex системы документирования может быть сложнее чем вы захотите. Latex вроде бы не работает на Android, но с планшета и смартфона можно использовать его на сайте overleaf.com. Прежде всего Latex интересен при создании научных публикаций, автоматическом создании документов⁹, при отсутствии оконной среды или экзотических запросах вроде включения в документ шахматных диаграмм. Linux может быть использован на смартфоне или планшете, а в среде Linux Latex работает

Понятие текстового файла.

Википедия говорит, что текстовый файл – это файл, содержащий текстовые данные. Эти данные есть набор символов, поэтому текстовый файл часто похож на то что создаёт печатная машинка. Однако такое определение требует оговорки – что текстовый файл отображается как текст благодаря использованию некоторой кодировки. Кодировка определяет как отображать данные текстового файла. Одни и те же данные могут поэтому выглядеть по-разному, при этом не меняясь сами. Например, этот абзац может выглядеть так.



Пользователю обычно не нужно подбирать кодировку. Текст сразу отображается

⁹ Хотя для этого обычно используется Docbook, или DITA

правильно. Тем не менее он заинтересован мочь подобрать кодировку в случае надобности. Для этого в UNIX есть программа `iconv`. Она позволяет подобрать кодировку, но, насколько знает автор, это требует просмотра результатов её работы. Обычно весь массив текстов, полученный на носителе или из сети, или в одном архиве, имеет одну кодировку.

Заметим также, что некоторые форматы файлов являются ничем иным как текстовым файлом, но использование файла в таком случае не равно его использованию как простого набора символов. Это верно, например, про форматы `Tex`, `SGML`, `RTF`.

В среде Windows текстовые файлы как правило имеют расширение `TXT`. В среде UNIX они могут не иметь расширений. Файл формата `RTF` имеет расширение `RTF` и так далее.

Базовые приёмы работы с текстом.

Вы можете не быть заинтересованы в каких-либо публикациях, но вам потребуется знать базовые приёмы работы с текстом. Они нужны не только для редакции и набора текста в текстовых редакторах, они используются для редакции и ввода текста в оконной среде и консоли вообще. Так, например, если в оконной среде вы пишете электронное письмо, то вам придётся его набирать текст используя эти приёмы.

Надо заметить что приёмы в компьютерной среде сильно отличаются от существующих в средах на устройствах с чувствительным экраном.

Рассмотрим принятые в популярных компьютерных системах приёмы.

- Вы можете изменить лишь текст, который можно редактировать. Иногда файл, содержащий текст, нельзя менять. В таком случае вся редакция текста бесполезна. В лучшем случае вы сможете перенести текст через буфер обмена.
- Иногда для начала редактирования надо щёлкнуть мышью на нужном месте.
- И вообще – для редакции текста вам потребуется текстовое поле. Оно само не считается отдельной программой, но является необходимой зоной, где может быть текст. Текст не может выйти за пределы текстового поля. Обычно оно отделено рамкой от всего остального, или отличается цветом. Вам потребуется понимать где программная среда может ожидать от вас ввода текста. Текстовое поле может быть разного размера и формы.
- Часто для начала редактирования текста надо щёлкнуть мышью в текстом поле, и этим поставить там текстовый курсор. Курсор обычно имеет вид мигающей вертикальной черты.
- Текст вводится на месте курсора. При этом курсор перемещается.
- Для перемещения курсора используются стрелки на клавиатуре. Или щелчок мышью в месте где нужен курсор.
- Обычно текстовое поле позволяет создавать лишь массив текста, а шрифт может быть лишь одного вида.
- В программах, специально созданных для создания публикаций, может быть много приёмов для операций с текстом, но такие приёмы не обязаны повторять то что есть в другой программе.
- Как и во многих других программах, отредактированный материал – в данном случае текст – должен быть явно сохранён. При этом создаётся или изменяется файл.

Также смотрите следующую главу.

Текстовый редактор `nano`.

Текстовый редактор – это программа, которая предназначена в первую очередь для произвольной, ручной, неавтоматической редакции текстов. Она в своей основе – ни что иное как просто подобие листа бумаги, ручки, и ластика. Такой простой текстовый редактор проще всего понять, но некоторые ученики скоро обнаружат что им не хватает его возможностей. Рассмотрим пока простейший редактор – доступный в UNIX. Это `nano`. Как узнать что он есть, установлен? Первый способ – запросить `man`-страницу редактора – подать команду

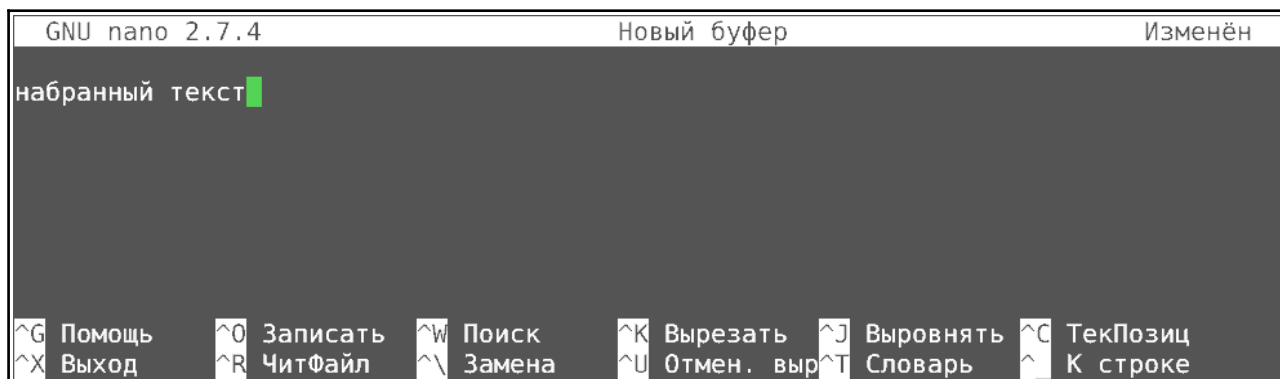
```
man nano
```

(см главу о командном управлении). Если страница показана, то её можно почитать, а проще выключить – нажав `q`. Способ второй – просто запустить редактор – подав команду

```
nano
```

Если редактор установлен – возможно, при установке ОС, то он будет выглядеть примерно

так.



При запуске в консоли конфигурацию «окна» редактора изменить нельзя – она будет повторять соотношение сторон экрана. Редактор должен быть установлен во всех ОС, родственных Debian, и может быть установлен во многих других. Он не установлен в ОС FreeBSD и NetBSD. В таком случае стоит поставить его с помощью пакетного менеджера. Для этого вначале подайте команду

`su`

чтобы ввести пароль администратора. Затем подайте команду для установки (при наличии выхода в интернет).

Операционная система	команда
FreeBSD	<code>pkg install nano</code>
NetBSD	<code>pkg_add -v nano</code>

Пакетный менеджер ответит строками, и может ожидать ответа на вопрос об установке. Следует отвечать утвердительно – **y**, то есть **yes**. Набираем эту одну букву, нажимаем **Enter**.

Затем в общем случае выходим из режима администратора, подав команду

`exit`

Теперь рассмотрим действия с редактором. Если вы стартовали **nano** без указания файла, который вы хотите видеть и\или изменить, то вы увидите пустое окно редактора. Затем вы что-то пишете и редактируете в нём. Потом вы сохраняете текст в файл, и выключаете редактор.

Действие	Клавиши
Сохранение	Ctrl+O
Выключение nano	Ctrl+X

Плюс означает одновременное нажатие. Вначале не спеша нажмите **Ctrl** – она сама ничего не меняет и не вводит. Затем быстро нажмите **O** или **X**. Это буквы латиницы. Их регистр не важен.

Способ переключения с латиницы на кириллицу может быть разным в среде разных ОС, как правило это **Shift+Alt**.

После запроса на сохранение вам потребуется набрать имя файла для сохранения текста. Затем надо нажать **Enter**.

Основы редакции текста в nano.

Если запускать редактор с опцией переноса строк, то они переносятся в пределах его окна или экрана, ломаясь, чтобы быть видимыми целиком. Для этого подайте команду

`nano --softwrap`

Но в таком случае всё же существует лимит на длину одной строки (по крайней мере в моём опыте). И при достижении его ничего нельзя ввести на этой строке. Приходится

нажимать **Enter** чтобы создать новую строку. Но мы не хотим там делать абзац! Мы хотим сами определять где начинается абзац. Что делать? Помощь пришла со стороны. Дело в том что тексты для обработки с помощью **asciidoc**, **Latex** предполагают, что между абзацами есть пустая строка. Если же два абзаца не разделены строкой, то они считаются одним абзацем, и сливаются в один абзац в результирующем файле публикации.

Также проблему лишних абзацев можно было бы решить иначе. Заменой единичной абзацной отбивки на пробел – исходя из того что он разделяет слова, что он нужен. Эта отбивка не видна, и называется концом строки в компьютерной науке. Но, внимание! Просто замена заменит *все* эти концы строк на пробелы. Нам этого не надо. Потому что тогда две подряд идущие отбивки – создающие пустую строку – пропадут. Паллиативный, то есть не очень хороший, но работающий способ правильной замены в том чтобы вначале вместо пустой строки вставить символ, который будет лишь на месте где должен начинаться абзац – и нигде больше, потом заменить концы строк на пробелы, а потом заменить «абзацный» символ на два конца строки.

После таких рассуждений вам может показаться что этот текстовый редактор плох. Скажу немного иначе – он стар. Но вообще надо понимать что работа за компьютером – это не только слушание музыки, но и настоящая работа. Замена слов, например. Заменять вручную ничего не надо. Но думать надо. За вас думать никто не будет.

Замену можно делать в любом текстовом редакторе, или – если это оправдано – с помощью таких программ как **SED**, или **AWK**.

У редактора есть режим жёсткого переноса строк. При этом режиме вводимая строка не слишком длинна, но на её конце редактор сам ставит конец строки. В таком случае для обозначения нужных вам, реальных абзацев придётся делать пустые строки.

Для получения текста помощи надо – уже запустив редактор -- нажать одновременно **Ctrl** и **g**. Расположенные внизу окна или экрана обозначения сочетаний клавиш включают «крышки», то есть «^», а они на большинстве компьютеров обозначают клавишу **Ctrl**.

Для стирания символа перед курсором нажмите **Backspace**¹⁰, на месте курсора – **Delete**. Также для слияния абзацев поставьте курсор после последнего символа в строке, и нажмите **Delete**.

Текстовый редактор Geany.

Если **nano** – это простейший текстовый редактор для консоли, то **Geany** – это популярный текстовый редактор для оконной среды. Выбор текстового редактора это примерно то же что выбор вина. Это дело вкуса, и опыта. Масса пользователей предпочитает что-то другое. Многие предпочитают **Geany**. Можно спорить. А можно просто признать, что часто человек привык к чему-то, или для него его выбор – лучший. Если **nano** часто упоминается просто как некий неказистый, но пригодный для пустяковых работ инструмент, то **Geany** часто называют текстовым редактором, или даже средой для программирования. Что это значит? У программистов свои требования к обработке текста. Если обычный пользователь пишет статьи, то программист пишет программу. Я не претендую на лучший выбор. Почему я пишу про **Geany**? Ответы:

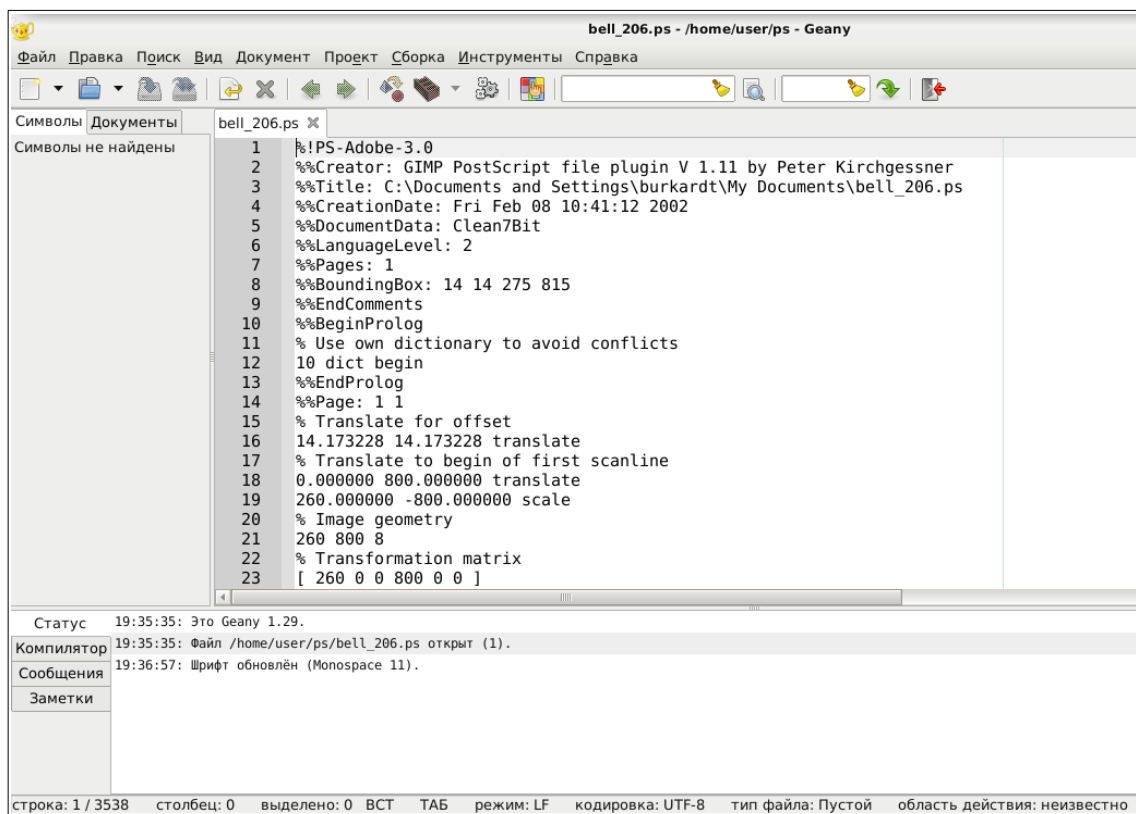
- он очень быстро работает даже с очень большими файлами. На старой технике он будет достаточно скор
- он позволяет выполнять в том числе замену невидимых символов вроде конца строки достаточно наглядным путём
- если вы пишете или правите программу, или просто читаете её, то часто **Geany** сумеет раскрасить её текст. Это может быть полезным. Разные части программы будут окрашены по-разному

Я не думаю что начинающим захочется, или необходимо править программы. Хотя программированию учат в школе.

В основах **Geany** работает так же как **nano**. Вы можете зайти на страницу www.geany.org/manual/0.19_ru/index.html

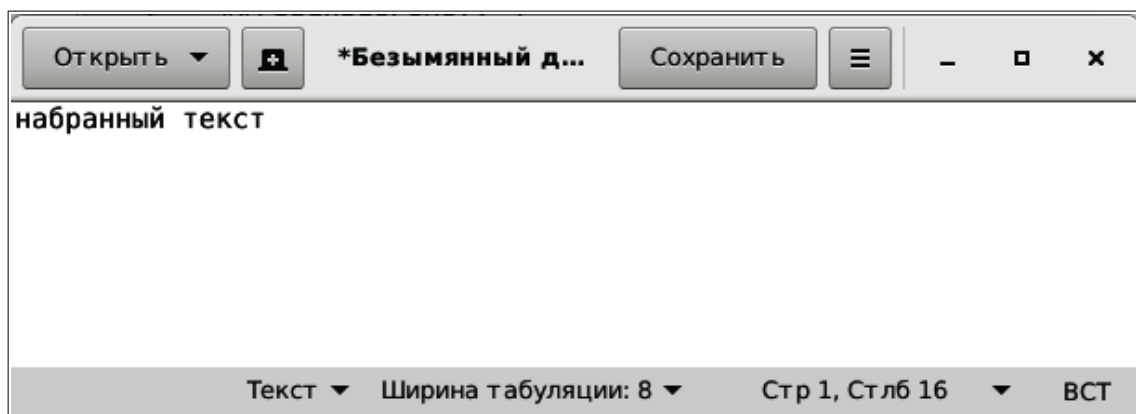
и прочесть русскоязычное руководство – но для старой версии.
Вид окна программы. Уже показан текстовый файл.

10 Обычно кнопка со стрелкой влево над **Enter**.



Gedit.

Gedit – это текстовый редактор, часто уже входящий в программную среду при наличии оконной среды, его поэтому не требуется ставить. На фоне других текстовых редакторов **Gedit** не выглядит достижением, но он является достаточным для простейшего программирования и набора текстов. Настройка шрифта текста возможна, но делается в настройках GNOME. Также стоит отметить, что редактор сам периодически сохраняет открытый файл, уменьшая размер потерь информации при внезапном отключении редактора, сбоя ОС или потере питания. Возможна проверка орфографии текста. Вид окна **Gedit**.



Пара слов о Latex.

На самом деле на эту тему надо было бы написать не пару слов. Почему стоит писать о Latex? Кто-нибудь может посоветовать вам использовать Word. Но, во-первых, его нет для UNIX. Во-вторых, это хороший выбор для тех кто понимает как работать с Word. Так вот, я думаю, что понять как *работать* проще с Latex. Хотя последний и требует более сложной установки и настройки.

На Debian Latex поставляется как часть TexLive. Если у вас есть возможность – просто

скачайте или запросите через почту DVD диск TexLive. Или вам придётся ставить TexLive, или Latex как обычную программу. Современные версии TexLive требуют установки. Сам по себе лазерный диск не позволяет запустить с себя ничего кроме инсталлятора. А он ставит Latex, вернее, множество программ на ваш компьютер. Инсталлятор требует наличия Perl, но он обычно уже есть.

После этого вы получаете возможность создать PDF-файл публикации, но установленный набор программ не содержит ничего для его просмотра. Многие пользователи уже будут иметь нечто вроде **Evince** для этого. Особая ситуация с микрокомпьютерами. Имея 512 Мб оперативной памяти вы не сможете создать публикацию при работе оконной среды. Придётся обойтись без неё. Набор программ может быть большим – занимать несколько гигабайт после установки.

Руководство по TexLive переведено на русский язык, и требует редактировать переменную **PATH** на UNIX для работы программ TexLive. Можно поступить проще – поставить **TeXStudio**, и настроить эту программу для обращения к нужной программе из набора программ TexLive. Так как автор не хочет ограничивать себя в выборе шрифта публикации, он использует в качестве таковой **xelatex**.

Шрифты заслуживают особого внимания. Не следует думать что шрифты не важны. Шрифт должен содержать все нужные символы, и нормально использоваться программами. В 90-е годы было создано много нелегальных, и дефектных шрифтов. Файл или файлы шрифта могут быть неподходящими или не пригодными. Заметьте: шрифт может стоить денег, и поставяться с лицензией. Минимальный набор шрифтов состоит из моноширинного, журнального, и шрифта с засечками. Это могут быть шрифты Free Mono, Free Serif, Free Sans. Шрифты совершенствуются, и имеют нумерованные версии. Для многоязычных, но не иероглифических документов стоит попробовать шрифты Noto. Для многоязычных документов посмотрите шрифт GNU Unifont.

Не выбрасывайте диск с TexLive. Поделитесь им со знакомыми.

Для обучения командам Latex можно читать учебник Львовского – ищите его в интернете. Он бесплатен.

Если такие программы как **Passpartout**, **OpenOffice**, **Lyx**, **Scribus** – тоже создающие публикации – требуют оконной среды, то Latex, как и **xelatex** (то есть **xetex**) сами по себе без неё обходятся. Объясню. Скажем, в OpenOffice вы вынуждены иметь работающую оконную среду не просто потому что иначе программа не запустится, а потому что вам надо будет вручную работать в окне программы. Latex берёт текстовый файл, обрабатывает его, и обычно создаёт файл формата PDF – не показывая его на экране. При этом он может использовать файлы с иллюстрациями, которые – иллюстрации – войдут в результирующий файл. Если что-то не так то будет сообщение об ошибке. Ваши первые опыты с Latex могут быть неудачными не из-за неправильной работы программ, а из-за неправильного текстового файла, набранного вами. Важно знать, что **xetex** требует другого текстового файла так как он не равен Latex. В случае с Latex простой исходный файл выглядит так (по материалам Википедии).

```
\documentclass[12pt]{article}
% Эта строка – комментарий, она не будет показана в выходном файле
\usepackage{ucs}
\usepackage[utf8x]{inputenc} % Включаем поддержку UTF8
\usepackage[russian]{babel} % Включаем пакет для поддержки русского языка
\title{\LaTeX}
\date{}
\author{}
\begin{document}
\maketitle
\LaTeX{} "--- это своего рода препроцессор текста для \TeX{} "---
  программы компьютерной вёрстки. \LaTeX{} является программируемым и расширяемым, что
  позволяет автоматизировать большую часть аспектов набора, включая нумерацию,
  перекрёстные ссылки, таблицы и изображения (их размещение и подписи к ним), общий вид
  страницы, библиографию и многое-многое другое. \LaTeX{} был первоначально написан Лэсли
  Лампортом в 1984-м году и стал наиболее популярным способом использования \TeX{}а; очень
  мало людей сегодня пишут на оригинальном \TeX{}е. Текущей версией является \LaTeXe.
\newline
\begin{eqnarray}
E = & \frac{mc^2}{\sqrt{1 - \frac{v^2}{c^2}}}
\end{eqnarray}
```

`\end{document}`

Вот результат – документ.

L^AT_EX

L^AT_EX — это своего рода препроцессор текста для T_EX — программы компьютерной вёрстки. L^AT_EX является программируемым и расширяемым, что позволяет автоматизировать большую часть аспектов набора, включая нумерацию, перекрёстные ссылки, таблицы и изображения (их размещение и подписи к ним), общий вид страницы, библиографию и многое-многое другое. L^AT_EX был первоначально написан Лэсли Лампортом в 1984-м году и стал наиболее популярным способом использования T_EXа; очень мало людей сегодня пишут на оригинальном T_EXе. Текущей версией является L^AT_EX 2_ε.

$$E = mc^2 \quad (1)$$

$$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (2)$$

Легковесные языки разметки.

Это, наверное, самый лёгкий путь создания более-менее сложных документов, и даже книг. Наиболее известный использованный язык – язык Википедии, но таких языков много. Автор отдаёт предпочтение **AsciiDoc**. Это, как уже сказано, не обычный язык, а способ создания документов. Так как он использует обычные символы и невидимые отметки конца строки, то – ВАЖНО – приходится следовать его правилам, которые сами не позволяют произвольного набора текста. Поэтому нельзя просто взять обычный текстовый файл, и сделать из него публикацию. Создавая ограничения для писателя, язык позволяет ему с помощью простого текстового редактора, и некоторых программ, производить публикацию. Заметим, что обычно такие языки не заняты созданием публикаций с переносом слов, и могут в отдельных случаях расходиться с требованиями русской типографической школы. Но, похоже, сейчас это никого не волнует. Прежде чем писать что-то на этих языках надо понять годятся ли они вообще для того что задумано. В Нидерландах есть фирма, берущая деньги за доводку книги до уровня шедевра. Фирма работает с Latex, не с LML.

Для создания публикации нужен почти любой текстовый редактор, и хотя бы одна программа для редактирования изображений – так как без них обычно нельзя обойтись. Таким образом, нужна оконная среда. Потребуется любое средство проверки ошибок набора – иначе ваш документ будет неполноценным.

Приведу пример. Вам надо вставить картинку. Текстовый редактор этого не умеет. Чтобы она получилась надо обозначить её в тексте. Как? Указать имя её файла, и, возможно, другие свойства. В случае с **AsciiDoc** это делается так.

...

`image:images/logo.png`

...

До и после обозначения картинки должна быть пустая строка. Если вы хотите обозначить

свойства изображения, то строка может быть сложнее, например.

...

```
image:screen-thumbnail.png[height=32,link="screen.png"]
```

...

Распознавание текста.

Распознавание текста – это способ получения текста из графических файлов. Другими словами, из изображений. Такими изображениями обычно являются снимки страниц или листов документа. Это позволяет вам не забивать снова уже существующий текст – при условии что это не рукопись, а именно текст, состоящий из символов шрифта. Это может быть текст, набитый на печатной машинке.

Насколько это нужно? Наверняка есть много людей, которым никогда не потребуется превращать уже напечатанный документ в подлежащий редакции документ. Тем не менее сама эта процедура очень важна так как убирает из работы повторный набор текста. Но оставляет вероятность его правки. Исходное *изображение* страницы превращается в редактируемый *текст*. А в чём разница? Разница очень проста: изображение – это набор пикселей, точек. Текст – это набор символов. И то, и другое можно редактировать. Но редакция текста совершенно не похожа на редакцию изображений. Если вам надо изменить напечатанный документ, или уменьшить объём файла из снимков страниц, то распознавание нужно.

Распознавание выполняется программой, способной найти, выделить символы в изображении. Не требуется понимать как она это делает. Обычно нужно обозначить язык искомого текста, и обозначить части документа: текст, рисунки, таблицы. Это может потребоваться делать вручную. Затем программа начинает готовить результирующий документ.

Результат сильно зависит от исходного файла, или файлов. Обычно требуются изображения с разрешением не менее 300 точек на дюйм. Получить хорошее изображение с помощью камеры телефона может не получиться. Лучшим средством для этого является сканер. Медленно работающие модели сканеров недороги. Но если вам надо сканировать много страниц, то не стоит экономить. Купите более быстрый сканер. Сканер должен быть совместим с вашей ОС. Также есть возможность делать снимки фотоаппаратом, но он должен быть достаточно качественным, и неподвижным при съёмке. Требуется равномерное и сильное освещение оригинала, но без бликов.

Печать.

Печать всё ещё нужна. В городах многие читают с экрана телефона. Но когда человек не умеет этого делать то выбор невелик. Мы живём в прекрасное время когда есть возможность недорого печатать книги. Автор не может описать современные мини-типографии – которые, может быть, станут рентабельными в недалёком будущем. Но уже сейчас есть принтеры и чернила (тонеры), с которыми печать книг не выглядит безумием. В принципе можно даже не слать книги в библиотеки, а печатать на месте. Тем более интересна печать микрофиш. Принтер – это не клавиатура, а устройство, которое само печатает на бумагу.

Стоимость печати складывается из стоимости бумаги, чернил, заменяемых деталей принтера, электричества, обложки, копирки, ленты. Есть дыроколы на десятки страниц. Устройства с Android умеют – может быть, не все – печатать без посредства компьютера. Вы можете использовать альтернативные картриджи – произведённые не той же фирмой что ваш принтер, но убедившись, что они безвредны для техники. Поищите сведения в интернете. Также стоит поискать дешёвые чернила или порошок. Они могут быть альтернативными.

Иногда стоит печатать в специализированной фирме – даже для экономии. Такая фирма может получать документ через интернет, и присылать результат по почте.

Алфавитный указатель.

Важно помнить что на бумаге достаточно трудно искать что-то в тексте. Поэтому всегда желательно иметь алфавитный указатель. Его можно создать не будучи автором книги. Важно лишь не нарушить закон. Исходным материалом может быть даже готовая книга. Она

распознаётся. Полученный файл уже является средством поиска слов в книге. Можно пойти дальше – на его основе создать указатель, и распечатать его. Для этого надо

- сохранить соответствие между частью текста и номером страницы
- отобрать существующие в тексте термины для включения в указатель

Вторая задача пострашнее. Мы говорим не о темах текста. Тут требуется интеллект, или как минимум словарь синонимов. Речь о словах. Мы не можем включить в указатель все слова текста. Можно использовать готовый указатель – например, созданный из словаря. Можно отобрать *отсутствующие* в обычной речи слова. Для этого можно взять, например, словарь Даля. Он есть в интернете как файл, и ничего не стоит. Можно набрать обычных слов из художественной литературы.

Кроме того нам нужно найти все формы слова, а написать в указателе одну форму.

Задача должна быть решена в пределах лингвистики за государственный счёт под эгидой министерства культуры. Тысячи библиотек России нуждаются в этом. Указатель не обязательно печатать. Его можно читать с экрана. Желательно указать номер строки или абзаца, а не только страницы.

Разумеется, делать подобную работу руками не стоит. Нужны программы.

Далее про печать.

Обычно к принтеру прилагается толстая инструкция. Вопрос о работе принтера под управлением бесплатной ОС открыт, и решается индивидуально под каждую модель. Может потребоваться установка драйвера, и CUPS. Всегда надо хорошо знать покупаемый или купленный принтер. Не достаточно уметь просто нажимать кнопку печати.

CUPS – это стандартная система печати UNIX. Она может оказаться более понятной чем её аналоги, и позволяет сделать принтер управляемым через локальную сеть, и интернет.

Заметьте, что принтер может «зажевать» бумагу. Задание печати в таком случае может быть не выполнено.

В простейшем виде печать на компьютере на самом деле не сложна. В программе, где вы видите документ, при работе оконной среды, вы нажимаете кнопку печати, или выбираете строку меню. В последнем случае вам предлагается возможность настроить печать более детально. В частности, отобрать печатаемые страницы. Зачем? Ответ 1: чтобы печатать дешевле. Ответ 2: чтобы печатать лишь чётные или нечётные страницы. Не каждый принтер умеет *сразу* печатать на обеих сторонах листа. Поэтому приходится печатать вначале на одной стороне листов, затем на другой.

Фокус номер 2: печатать в режиме экономии чернил, или осветления. Если его нет, то стоит осветлить сами страницы, превратив их в изображения, или используя возможности программы **Ghostscript** – рецепт описан в интернете.

Фокус номер 3: печать страницы по две на одном листе, на одной стороне.

Фокус номер 4: использовать тонкий шрифт. Но для этого надо редактировать документ.

Также программа, откуда вы печатаете, может влиять на печать.

Всегда надо просмотреть документ перед печатью. Особенно если он отредактирован. В нём могут быть досадные, неожиданные дефекты. Так, например, шрифт Gentium содержит неправильные кавычки. На листе они будут более заметны. Тут может помочь печать в PDF-файл. Так как надо видеть документ, то требуется оконная среда.

Всегда вначале выдайте на печать одну страницу для проверки основного шрифта, и прочего. Сам шрифт не искажается при печати, но может выглядеть не так как на экране. Подняв разрешение печати до 1000 точек и более (на дюйм), вы получаете публикацию, где мелкие части букв видны отчётливо.

По возможности используйте офисную бумагу. Вернее, качественную. Чтобы книга легко читалась, не была слишком тяжёлой.

Конечно, если вы не нуждаетесь в просмотре публикации, то можно давать задания печати с помощью команд, и так же настраивать принтер. В консоли можно настроить программную среду на использование фреймбуфера, и, благодаря ему, видеть страницы публикации – по крайней мере если она в формате PDF. Этот путь позволяет использовать Latex на микрокомпьютере с 512 Мб оперативной памяти.

Стоит подумать о приобретении двух принтеров разных моделей. Один из них печатает быстро, другой – качественно. Публикации может быть надо печатать на обеих, поделив их на страницы. Заметим, что матричные принтеры позволяют дёшево печатать текст, но они сами

дороги.

Подготовка к печати.

Готовить документ может быть не нужно. Это зависит от стоимости печати и других обстоятельств. Не надо особого ума или знания чтобы понять что документ не готов к печати. Лишние тексты, пустые или почти пустые листы, огромный расход чернил, нечитаемость сносок и формул после печати на конкретном принтере – всё это означает что документ не готов.

Подготовка к печати в типографии или печатном сервисе определяется ими. Подготовка к печати лучше всего выполняется когда вы используете программу для вёрстки документа. Документ при этом должен быть редактируемым. Обычно это документы форматов DOC, DOCX, ODT, HTML. Для редакции формата PDF есть как минимум одна бесплатная программа для Linux, но, возможно, лучше конвертировать такой документ в формат, обозначенный выше. Это даст больше возможностей изменения документа. Если документ в формате PS или DJV, то его можно конвертировать в набор изображений, и распознать, а потом сохранить в формате пригодном для вёрстки. Именно формулы лучше всего делать в Latex. Также некоторые программы умеют создавать формулы, которые потом в худшем случае вставляются в документ как изображения.

Подготовка к печати нужна для экономии чернил и бумаги, более лёгкого чтения, исправления ошибок.

Теперь рассмотрим подготовку документов более детально. Я не обещаю что рассмотрю этот вопрос полностью или достаточно подробно.

Если исходный материал существует как набор страниц сайта, то придётся получить их все с помощью **wget**, и превратить в один документ. Это позволит быстрее избавиться от лишних иллюстраций и сверстать материал. Печать каждой страницы отдельно реальна, но утомительна и требует печати рекламы, а также легко приводит к перерасходу чернил и бумаги. Кроме того, это не позволяет иметь оглавление. Режим читателя, или режим чтения в браузере может помочь. После получения страниц можно соединить их путём программирования в одну страницу, которую лучше всего конвертировать в формат ODT или DOC, и редактировать для получения нужного вида документа. Соединение страниц описывается так.

```
cat page1.html>>all.html
cat page2.html>>all.html
...
```

То есть нужно передать содержимое страниц, которое является текстом, в один текстовый файл. То же верно про совокупность страниц на вашем компьютере. Вопросы автоматизации и программирования описаны в этой книге.

Помните, что любое изменение содержимого документа, и даже оформления, или смена листа, как и изменение полей влечёт за собой нарушение нумерации страниц, и оглавление становится лживым. Поэтому даже простая печать двух страниц на одном листе может потребовать менять оглавление. То же верно про алфавитный указатель.

После распознавания или конвертации исходного документа потребуются привести его в пригодный вид. Это может быть довольно долгая и непростая работа. Может быть её будет легче сделать с текстовым файлом, который можно обрабатывать автоматически, и, после того как он удовлетворит требованиям языка **AsciiDoc**, конвертировать в формат ODT или DOC. Обратите внимание на искажения, происходящие при конвертации в текст. Также надо иметь в виду что конструкции **AsciiDoc** могут уже быть в таком файле и исказить результат. **AsciiDoc** является, наверное, самым успешным легковесным языком разметки документов. Он обрабатывается такими программами как **asciidocctor**, **pandoc**, **a2x**. Также можно использовать Latex, хотя это сложнее и не всегда оправдано.

Результат работы любой конвертирующей программы может быть не пригодным для печати, но являться путём к печати. Некоторые такие программы вообще не работают с кириллицей.

Вместо трудной работы по вёрстке, которая может быть слишком долгой, можно печатать с уменьшением насыщенности частей документа, то есть осветлением. Такая возможность может зависеть от драйвера принтера. Всегда желательно испытать подобные трюки с помощью печати в файл. Достаточно простой путь это сделать – использовать программу

imagemagick или **xnview**. Они выручат вас если не получается отрегулировать печать именно средствами настройки программ для печати.

В случае вёрстки документа для уменьшения расхода чернил стоит пройтись по иллюстрациям, и осветлить их. Один из способов это сделать – сохранить документ в формате HTML, и использовать **imagemagick** или **xnview**.

Также такие части документа как окрашенные колонтитулы, фоны, рамки таблиц можно обработать. Часть таких пожирателей чернил можно убрать вообще. Также вы можете уменьшить размер иллюстраций. Такие трудные, по сути монотонные работы можно автоматизировать, например, макросами в LibreOffice. Это не простое дело. Но оправданное. Исходный документ следует сохранить на случай порчи редактируемого.

Читайте книги по OpenOffice, и тексты интернета чтобы прояснить вопросы и облегчить вёрстку. Уже готовый к печати документ можно выложить в интернет чтобы не делать эту работу сотни раз.

Пример устройства с дешёвой печатью – принтер Epson L120.

Я обозначу этот принтер, не стараясь описать всё что может потребоваться знать. Набор моделей меняется со временем. Принтер L120 является очень выгодным для печати большого количества страниц. Для печати фото, а тем более качественной печати фото он может не подойти. Но так как моя задача – помочь в первую очередь библиотекам, то это не так важно. Библиотечная печать скорее ориентирована на текст. Принтер использовался мной в среде Debian Linux 8 и 9. В среде Windows его потребительские качества могут отличаться по причине разных программ, задействованных для печати и обслуживания принтера. Принтер мной не пробован в среде FreeBSD.

Инструкция к принтеру не очень большая. Это недостаток, а не преимущество. Инструкция на русском языке в том числе расположена в интернете, и бесплатна.

Несколько фактов и наблюдений.

- Принтер с чернилами нельзя ставить вертикально. Разумеется, в таком положении его нельзя использовать, но можно по незнанию поставить, желая сэкономить место. Даже если чернила вылиты я считаю опасным ставить «на попа». Можно ли вылить чернила я не знаю. Наклейте на принтер бумагу с предостережением. Не полагайтесь на людей или память.
- Принтер довольно крупный, и длинный. Он не может печатать на листах формата более А4.
- Вам потребуется самостоятельно залить чернила из бутылочек. Ёмкости не используйте повторно.
- Всегда надо правильно проводить печать. В противном случае неправильное завершение работы принтера может привести к плохой печати.
- Без настройки печати принтер печатает грубовато. Это важно, в том числе, при печати чего-то вроде формул.
- Путём настройки и/или замены драйвера можно существенно улучшить печать изображений. Но – ВАЖНО – речь именно о печати прямоугольного изображения, а не страницы, или многих изображений сразу. Как печатать документ? Один из выходов – печатать его как набор именно изображений. Обычно документ представляет собой набор *страниц*, которые могут быть выданы на бумагу, или превращены в изображения – в среде UNIX с помощью программы **imagemagick**, или подобной. Также можно использовать программу **gutenprint**.
- Для улучшения печати при нормальной вычислительной мощности техники и достаточном размере оперативной памяти можно использовать виртуальную машину с Windows. Способ мной не опробован.
- Ничто не мешает вам иметь на одном жёстком диске и Linux, и FreeBSD, получая дешёвую печать, и документированность ОС одновременно. Количество наличных ОС может быть ограничено четырьмя. Их нельзя использовать одновременно. В данном случае я имею в виду не ОС в виртуальной машине.
- Мне удалось получить приемлемые отпечатки фото с помощью программы GIMP.

Печать со смартфона.

Чтобы отправить файл в печать с телефона на принтер, без подключения его к

компьютеру, необходимо купить специальный OTG-кабель (с одной стороны – Micro USB или Type-C, с другой – обычный USB). Одну часть кабеля подключаем к мобильному устройству, вторую к USB-кабелю принтера. Далее следует скачать на смартфон специальное приложение (например, СтарПринт, PrintHand или любой модуль печати от производителя принтера). Чтобы распечатать с телефона нужно сделать следующее:

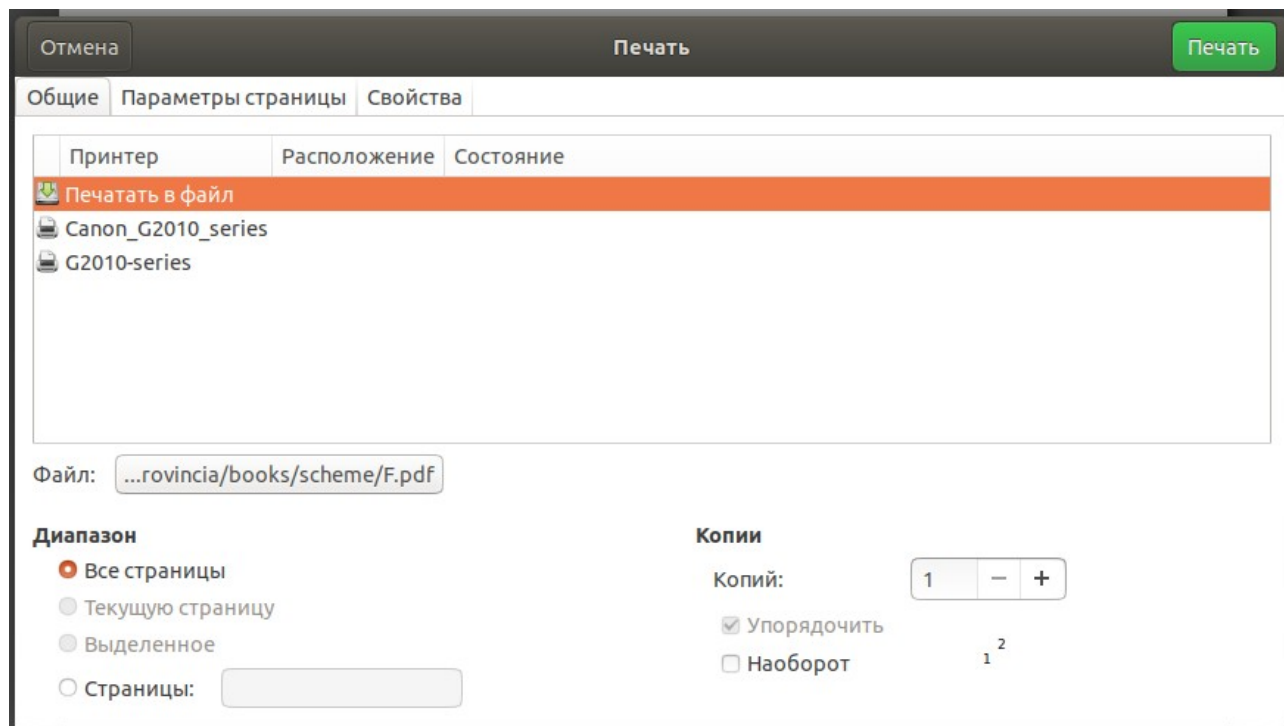
- подключить принтер к телефону через OTG-кабель;
- скачать приложение для печати на телефон;
- включить принтер и открыть ПО на смартфоне;
- выбрать необходимый для печати файл и нажать кнопку «Печать».

Приложение автоматически найдет подключенное печатающее устройство и попросит разрешение на предоставление к нему доступа (в зависимости от программы). Далее нужно будет повторно нажать на кнопку «Печать» или просто подтвердить выбранные действия. Принтеру понадобится какое-то время на подготовку, после чего он сразу же осуществит печать нужного документа, изображения или фотографии.

(по тексту kartridge.org)

Печать в файл.

Это не печать в обычном смысле слова. При печати в файл создается файл, который выглядит как бумажная публикация. Его можно либо передавать именно как файл, либо печатать на бумагу. В среде UNIX при наличии оконной среды и программы, способной печатать – например, Firefox, вы при акте печати выбираете либо физический принтер, либо файл.



Глава 27. Передача данных между программами. Буфер обмена.

Автор, как и читатель, не являются, и не должны быть специалистами, знающими всё про передачу данных между программами. Опишу вопрос по мере своего знания и опыта.

Передача данных между программами важна потому что использование одной программы для решения пользовательской задачи часто невозможно, но обрабатываемые данные при этом всё те же. Наверное, самым понятным ученику способом передачи данных является их сохранение в файл, который затем читается другой программой. Но этот способ имеет как минимум два недостатка: файл может быть слишком велик, и он часто не «понятен»

конкретной программе. Он даже в простейшем случае может иметь кодировку, с которой не умеет работать конкретная программа. Этим способом можно пользоваться.

Другой способ – который должен быть известен на практике каждому ученику – это передача некоего материала – возможно части файла – через буфер обмена, который обеспечивается ОС. Я не уверен что он будет работать, или работать в полную силу в консоли. Там возможна передача текста, но это не похоже на использование буфера обмена в классическом смысле этого слова. Буфер обмена технически представляет собой часть объёма оперативной памяти и/или постоянной памяти, через который данные конкретного вида передаются между программами. Тут важно что именно передаётся, откуда и куда. Получающая и отдающая программа могут влиять на результат. Обычно обе программы – оконные, и наиболее простым и безвредным способом передачи является операция копирования и вставки, выполняемая с помощью мыши. Копируется выделенная часть документа, которая затем вставляется на месте курсора в принимающей программе. Если вы хотите выделить много текста, то лучше использовать для этого не мышшь, а клавиатуру, или меню программы.

Этот способ надо отличать от существующего в среде Windows способа, называемого OLE, который внешне отличается от буфера обмена, и, видимо, совсем не похож на него по сути.

Кроме того, в таких программах как OpenOffice есть возможность явно указать другой способ передачи данных. А именно – указать разные форматы данных, или связь через DDE. DDE – протокол, то есть способ передачи данных, который первоначально был создан производителем Windows, и может оказаться более аккуратным в результате вставки чем простая передача текста.

Так как мы обсуждаем передачу данных, то надо упомянуть связывание одного файла с другим – не ссылкой файловой системы, а так чтобы один файл менялся при изменении другого. Для этого нужна программа которая умеет это делать. Надо понимать что изменяемый файл может не получиться вернуть в прежнее состояние.

Пользователь может экспериментировать с этими способами, и видеть результат.

Кроме того, в среде UNIX традиционно возможна передача байтов данных через трубы (pipe), обеспечиваемые командной оболочкой, или конвейер. Это реальный способ избежать создания больших, и, возможно, ненужных файлов. Также это позволяет уменьшить ручной труд.

Смена кодировки в среде Linux и BSD выполняется с помощью программы **iconv**, а также некоторыми текстовыми редакторами и с помощью браузера.

При копировании и вырезании данных в буфер обмена его прежнее содержимое теряется. Типично что с буфером обмена можно обращаться с помощью клавиш.

Клавиши	Операция
Ctrl+C	Копировать
Ctrl+V	Вставить
Ctrl+X	Вырезать

Глава 28. Электронные таблицы и базы данных.

Это популярные средства и объекты платной, а иногда бесплатной работы.

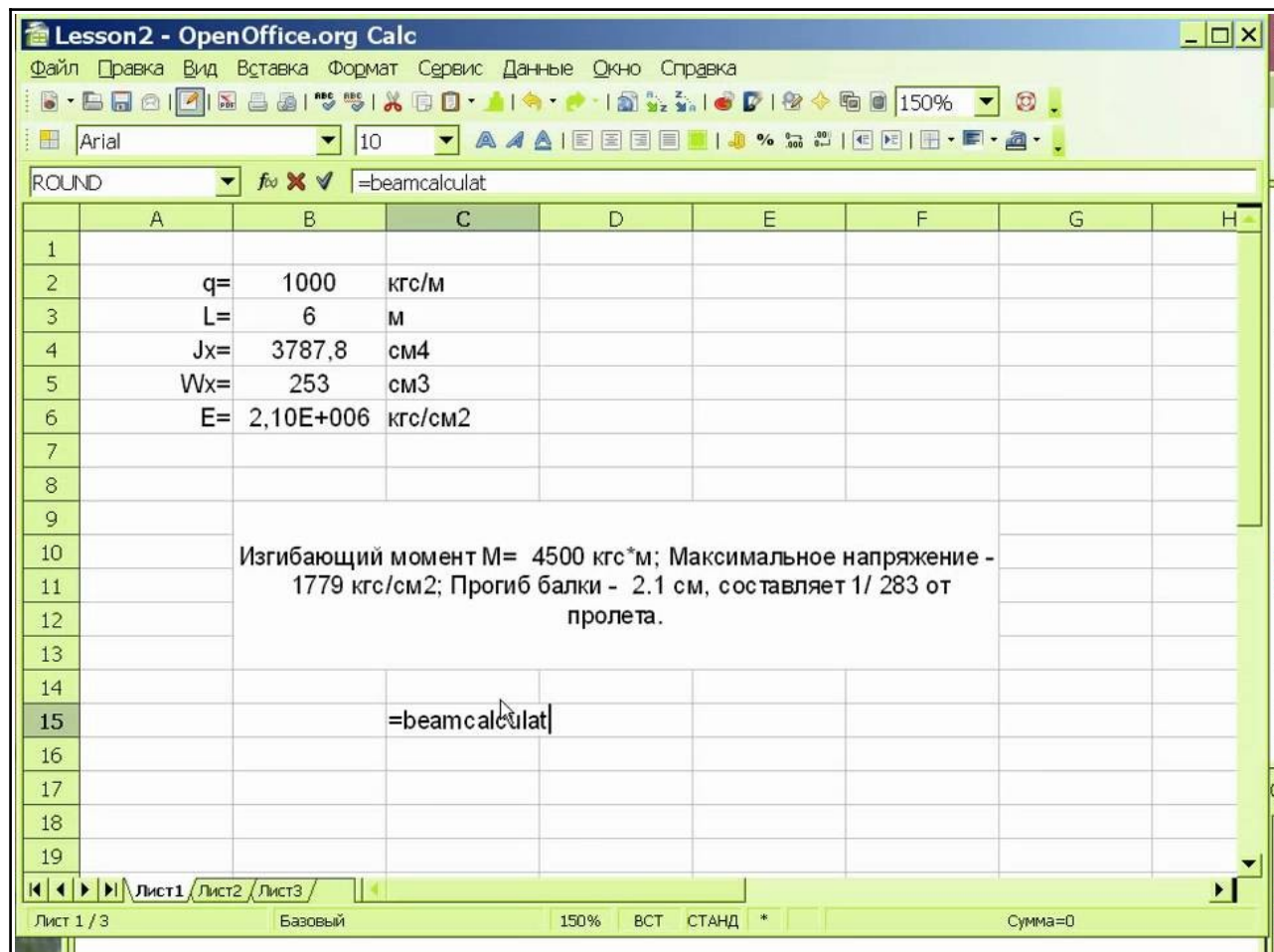
Электронные таблицы – это оконные программы для операций с данными, разложенными по ячейкам таблицы. Это позволяет обрабатывать их как содержимое ячеек. Ячейки имеют понятные адреса. Электронные таблицы часто являются причиной раздражения если пользователь не умеет с ними работать. Результатом работы является таблица, или набор таблиц, отчёт. В частной практике такие программы могут быть достаточно бесполезны. Но используются на работе. В среде UNIX вместо электронных таблиц можно применить некую базу данных – в простейшем случае текстовую – и, например, такое средство обработки текста как AWK

(см руководство по AWK по адресу

https://www.math.spbu.ru/user/rus/cluster/Doc/Library/awk_baluev/awk_bal_oglav.shtml или на

<https://pkp3.home.blog>).

Это старое пособие на русском. Наиболее свежее – на английском – на <https://www.gnu.org/software/gawk/gawk.html>. При этом программа для традиционной обработки электронных таблиц есть. Это Calc из LibreOffice или OpenOffice.



Базы данных – это совокупность организованной информации о чём-то. Само понятие очень общее. Часто база данных – это очень важная собственность коммерческой или бюджетной организации. Они содержат свои данные в базе данных. База данных обрабатывается программами автоматически или вручную. В среде Windows для этого достаточно часто используется программа **Access**. В UNIX для этого есть не только оконная программа LibreOffice Base, но и другие средства – если они вообще подходят к выполняемым работам. Опять же – без образования не начинайте работу.

Базы данных также постоянно используются там где для работы надо через браузер обрабатывать данные. Такая практика имеет свои преимущества.

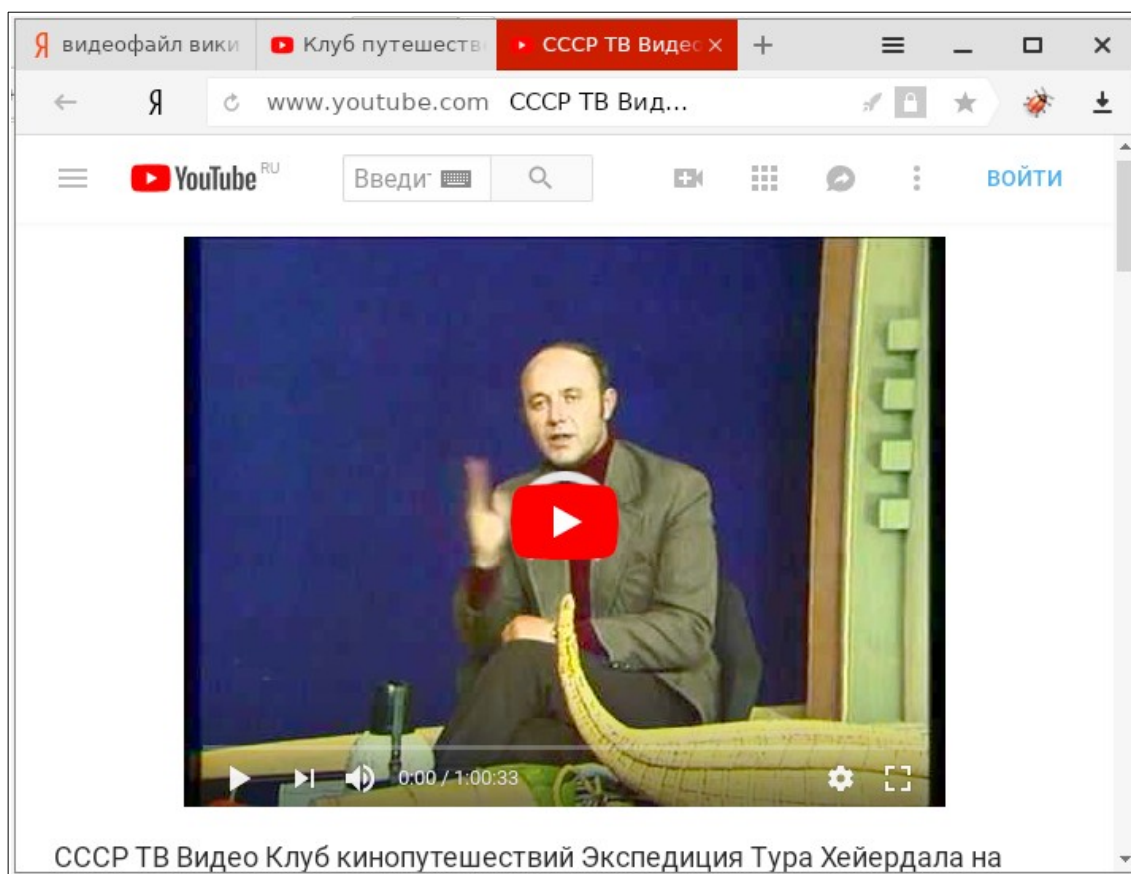
Глава 29. Звук и видео (мультимедиа).

Я не знаю какой нам смысл объединять эти две темы одним словом. Но понятие такое существует. Относительно звука и видео существуют процессы их воспроизведения, записи, редакции, конвертации, создания. Они, очевидно, выполняются с помощью неких программ. Но тут важна и техника. Для профессиональной работы выбирается развитая звуковая-, и видеоплата. Эти же платы, встроенные в материнскую плату – скорее всего, не годятся. Тут речь о компьютере. Планшеты, и телефоны, как и микрокомпьютеры для, хотя бы, редакции видео не годятся. Я не исключаю, что нужная вам программа работает лишь с некоторыми платами. Автор не является специалистом в мультимедиа.

В популярной практике звук и видео чаще всего просто просматриваются или прослушиваются. Видео обычно уже содержит звук. Часто одна и та же программа-плеер умеет

воспроизводить разные форматы звука, и видео. Таких программ достаточно много. В частности, оконный браузер должен быть способен воспроизводить хотя бы те форматы звука и видео, которые популярны, или передаются в интернете – в том числе для немедленного просмотра – трансляцией, без явного или какого-либо сохранения на компьютерном устройстве. Какие это форматы? Это, для звука, конечно, MP3. Это, для видео, конечно, MP4, и, в современных условиях, H.264 – именно так, с точкой. Незрелый или облегченный браузер может не уметь проигрывать желаемый формат. Напротив, такой браузер как Chrome, скорее всего, воспроизведёт любой нужный пользователю видеофайл. Трансляция видео или звука, с точки зрения браузера и плеера, является воспроизведением файла определённого формата.

Некоторые плееры называются универсальными, но не являются ими. Они обычно воспроизводят многие, но не все форматы видео и звука. Все плееры позволяют воспроизводить видео и звук с любого места в видео- или звуковом файле, ставить на паузу проигрывание. Это могут быть поначалу не совсем понятные программы. Ниже показано окно браузера с видео.



Свежий на сегодняшний день формат H.265 может потребовать для своего проигрывания много вычислительной мощности. Думаю, что сам формат – хороший. Но не каждый компьютер сможет проиграть конкретный файл этого формата. Каждый пользователь должен знать, что проигрывание видео может требовать немалой вычислительной мощности. Так же, как редакция, и конвертация видео, и создание.

Для обработки видео и звука могут потребоваться кодеки. Это программы. Но они сами не управляются пользователем, а используются другими программами. Кодеки для бесплатных ОС могут появиться заметно позже чем для Windows.

Один из способов создания видео – захват экранного изображения – часто вместе со звуком с микрофона. Некоторые фирмы так создают презентацию своих товаров. Так создаются уроки по разной деятельности. Их много в интернете. Захват экранного изображения также может требовать не совсем малой вычислительной работы.

Какой нам интерес всё это обсуждать? Во-первых, это интересно. В интернете всё ещё можно найти видео со многими фильмами. Там же лежат архивы радиостанций, и

телепередач. Во-вторых, на сегодня даже телефон является чем-то вроде видеокамеры, а сами видеокамеры достаточно дешёвы. Любой человек может создать свой – обычный, в данном случае – не анимационный фильм.

Конвертация видео в основном нужна для уменьшения его объёма, а звука – для проигрывания на конкретном устройстве. Всё ещё используемые камеры, записывающие видео в формате DV, например, создают очень большие файлы даже при размере кадра, равном телевизионному PAL. Более современные камеры также создают большие файлы. Так как увеличена площадь кадра. Это камеры для кадра размера HD, или для ещё большего кадра.

Автор не имеет опыта в монтаже фильма. Но он может быть не нужен при семейном видео, или любительском кино.

Вообще говоря, лучшим местом хранения готового видео является лазерный диск. Но тут многие совершают ошибку – неправильно создают этот диск. Проигрыватель видео оказывается неспособным проиграть видео. Чтобы не делать таких – подчас непоправимых – ошибок надо понять простой факт: проигрыватель рассчитан лишь на некоторые параметры видео, и на некоторые диски. В этой книге есть глава о таком проигрывателе. Он подключается к телевизору. Всегда убедитесь, что вся техника и программы составляют одну технологическую цепочку, и ваша цель достижима. Возможно, ваш проигрыватель сможет подать видео на телевизор стандарта PAL, но вот *качества* картинки высокого разрешения вы не получите – даже если само видео это позволяет.

Никогда нельзя в спешке, или при невежестве покупать видеокамеру. Качество кадра может вас не устроить. Надо учесть все аспекты вашей будущей деятельности. Некоторые камеры стоят очень недорого, но они не могут дать хорошего изображения, и вы не сможете ничего с этим сделать. Заметим, что многие цифровые фотоаппараты могут снимать видео.

Достаточно часто пользователи сталкиваются с потребностью конвертации видео. Что это такое? Это смена формата, и/или характеристик видео. Что это за характеристики?

- соотношение сторон кадра
- разрешение, то есть количество точек
- количество данных о цвете – глубина цвета
- битрейт – количество данных на одну секунду просмотра
- частота кадров
- субтитры
- и другие...

Конвертация выполняется без просмотра видео, обычно для всего файла без исключений. Популярной программой для этого на UNIX является **ffmpeg**. Как утверждают авторы, она работает очень быстро. Тем не менее конвертация может длиться много часов. С этой программой трудно разобраться. Она не требует оконной среды, и рекомендуется её не использовать при конвертации видео. Другая подобная программа – **mencoder**. Она может оказаться более понятной. В любом случае вначале конвертируйте небольшой файл или часть видео, и проверьте результат. Вам может потребоваться прочесть побольше про свойства видеофайлов. Также вы можете купить аппаратный кодировщик – отдельное устройство для обработки видео. Но прежде убедитесь что он будет работать с вашей техникой и ОС.

На UNIX проще всего создать обычный DVD-диск видео с помощью программы **Devede**. Но прежде посмотрите не слишком ли много вы теряете сохраняя так. Современные видеокамеры рассчитаны на BD-диск.

Оцифровка.

У многих людей в нашей стране всё ещё есть старые, аналоговые, но не всегда плохие носители – виниловые и гибкие пластинки, видеокассеты, аудиокассеты, плёнки любительских камер, и даже бобины с магнитофонной плёнкой, фотографии, фотослайды, фотоплёнки разных типов и размеров. Это аналоговые носители. Они несут тот же материал, что и получаемые из них файлы, но *записан* он иначе. То же верно про книги, газеты, журналы, открытки, альбомы, дневники.

Какой нам интерес это обсуждать? Оцифровка позволяет уместить тот же материал в меньшем объёме, быстрее его копировать, передавать через сеть, и редактировать в компьютере. Кроме того, такие носители как лазерный диск проигрываются иначе. Поэтому качество записи не портится от проигрывания.

Это – большая тема, и тут она будет лишь немного затронута. Как всегда, надо подумать

как и где будет проигрываться видео или звук. В отличие от ситуации с уже записанными в цифровом формате данными видеокамеры, тут потребуется перенести изображения и звук со старых носителей в компьютер, часто – обработать, и потом сохранить на более современные носители.

Относительно подлежащих оцифровке материалов могут быть выполнены процедуры очистки от дефектов, в том числе щелчков и треска, нарезки, и каталогизации.

Нарезка звука с пластинок или аудиокассет – неприятное воспоминание для тысяч людей, получивших мигрень на этой работе. То же верно про видео. Но обе эти задачи в наше время решаются даже без использования искусственного интеллекта. Программы умеют делить видео на сцены, а с пластинками ситуация ещё проще – паузы в звуке «понятны» программе, и они обычно означают перерыв между дорожками пластинки.

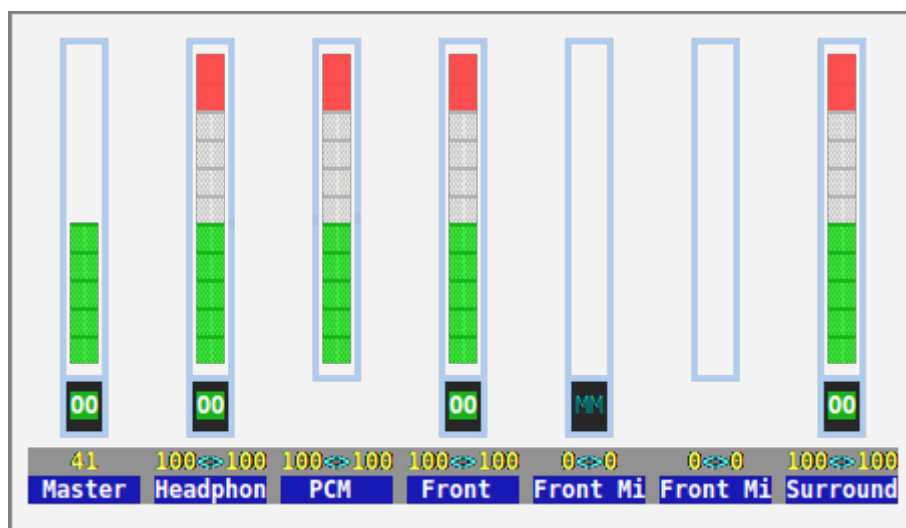
Бумажные материалы сканируются, и, в простейшем случае, просто сохраняются как снимки страниц.

Фото и слайды сканируются, и также могут быть обработаны. Их можно выдать на бумагу – как и кадры видео.

Настройка звука в Linux.

Потребность в настройке звуковой системы обусловлена разными причинами. Прежде всего это прослушивание звукозаписей и просмотр видео. Обычно воспроизведение звука работает если есть оконная среда сразу после установки ОС. Оконная среда не обязательна для работы звуковой системы. Тем не менее настройка звука часто нужна. Её не надо путать буквально с настройкой или программным обеспечением звуковых устройств. Так, например, если новое оборудование требует драйвера, то без него оно не обязано работать, и к состоянию звуковой системы это не имеет отношения. Настройка звука выполняется по-разному в разных ОС, и в зависимости от наличия оконной среды. Простейшая настройка выполняется при наличии достаточно развитых оконных сред, таких как GNOME. Например, в Ubuntu настройка громкости выполняется в том же меню которое вызывается щелчком на значках сети, звука и батареи в правом верхнем углу экрана. Там же есть пункт Настройки. Он выводит окно где можно настроить используемые устройства и громкость звука.

Однако это не всё. В среде Debian может потребоваться использовать программу **alsamixer**. Она работает в консоли, и выглядит так.



Программа позволяет установить уровень громкости на конкретных устройствах.

Если в вашей практике есть потребность работы со звуком то это может потребовать терпения, и чтения форумов интернета. Это особенно верно при обработке звука.

Глава 30. Изображения и анимация.

Тема изображений уже затронута в главе об оцифровке. Изображения окружают нас, и могут быть необходимы для понимания. Они входят в документы. Может быть полезным

различать изображения обычного вида – например, фото, и предназначенные для чего-то вроде научно-технической работы – графики, и подобные изображения для лучшего понимания и более быстрого усвоения сказанного.

С приходом цифровой съёмки фото ситуация с фотоархивами сильно изменилась. Они стали больше, и появилась возможность использовать встроенные данные фотографии для упорядочивания фото, и их поиска. Что это за данные? На сегодня наиболее известная группа этих данных называется EXIF, и содержит такие сведения:

- производитель камеры,
- модель,
- информация о правообладании,
- выдержка,
- диафрагма,
- ISO (светочувствительность),
- использование вспышки,
- разрешение кадра,
- фокусное расстояние,
- размер матрицы,
- эквивалентное фокусное расстояние,
- дата и время съёмки,
- ориентация камеры (вертикально/горизонтально) для камер со встроенным акселерометром,
- тип баланса белого,
- географические координаты и адрес места съёмки

Другой известный набор данных называется XMP. Какой нам интерес это обсуждать? Во-первых, теперь встроенные данные передаются вместе с изображением. Во-вторых, их можно редактировать.

Ещё одна хорошая новость в том что интеллектуальные программы умеют искать похожие изображения, и составлять группы похожих изображений.

Таким образом, поиск фотографий, и их разбор по группам ускорился.

Лучшим средством делать фото всё ещё является фотоаппарат. Продаются в основном цифровые фотоаппараты. Всегда узнайте о вашей покупке подробную информацию, и поищите реальные примеры фото. Дешёвые камеры способны делать неплохие фото лишь в хороших условиях съёмки. Сама по себе электроника или оптика, или реклама ничего точно не говорят о качестве снимков. Также рекомендуется уметь делать фото телефоном.

Создание и правка изображений давно возможны, и осуществляются во множестве программ. Эту работу можно поделить на чисто ручную, полуавтоматическую, и автоматическую. В последнем случае можно использовать, например, программы для составления графиков. Наиболее известная такая программа – **gnuplot**. Она хороша для показа статистики и других подобных применений. Такие программы как **Adobe Photoshop**, или **Corel Draw** дороги, и не нужны среднему пользователю. Он всегда может найти им замену.

Также вы можете конвертировать изображения для изменения их свойств и формата. Часто эта работа делается автоматически, например, с помощью программы **XnView MP**.

В условиях села имеет смысл готовить лазерный диск с изображениями формата JPEG. Такой диск часто можно проиграть на аппаратном плеере. Получится нечто вроде показа слайдов. Но рекомендуется телевизор высокого разрешения.

Достаточно часто книги в интернет можно увидеть в формате **dejavu**. Это не буквально книги – с хранением текста, а снимки страниц. Этот формат перспективен для малообъёмного хранения литературы в библиотеках – по крайней мере в случае чёрно-белого снимка страницы. Есть и другие подобные форматы. Например FLIF, BPG, WebP. Появятся новые, ещё более полезные форматы, использующие ещё более совершенные способы хранения данных.

Анимация на сегодня включает в себя создание фильмов, и поэтому является очень большой темой. Примитивная анимация может быть довольно бесполезна. Именно создание анимационного видео обычно требует много вычислительной мощности при изменениях в сцене с глубиной и объёмом. В научной работе можно попробовать создавать анимации путём программирования. Например, с помощью **VPython**.

Глава 31. Попутные работы.

Под попутными работами тут понимается всё что не относится к главным темам пользовательской деятельности, которые есть: текст, печать, звук, изображения, видео, интернет.

Архивы.

Архивы – это файлы, содержащие в себе как минимум один файл или каталог, а обычно много файлов и каталогов. Это значит, что всё это содержимое архива может быть – при исправности архива и наличии нужных программ – использовано для создания этих файлов и каталогов – используя содержимое архива. Нас, как пользователей, не волнует каким образом устроен архивный файл. Нас интересует польза для нас архивных файлов, их смысл в деятельности пользователя и администратора. Какой нам интерес обсуждать архивы? Архивы обладают полезными качествами, а именно благодаря им мы:

- можем передавать много файлов и каталогов одним файлом
- можем поделить передаваемый материал на несколько носителей – создавая многотомный архив
- часто существенно сокращаем объём передаваемых данных
- получаем способ проверки исправности переданных данных – если тип архива и программа-архиватор это позволяют

Основной интерес, разумеется, это представляет когда надо передать нечто через медленный, или ограниченный по объёму данных канал связи. Попутно заметим, что именно сокращение объёма данных производится не только архивацией, но вообще сжатием данных, сменой формата файла, его свойств.

Поскольку интересно передавать минимум данных через сеть, то часто вы в интернете увидите готовые к получению вами архивы, их содержимое, которое вы хотите получить, само не является архивом, а является, например, текстовым файлом. Так, например, иногда книга, распространяемая через интернет, находится там как *исходники* Latex. В данном случае это как минимум один файл, а скорее набор файлов в архиве. Эти файлы могут быть использованы вами не только для генерации файла книги, но и, при желании, изменения её вида. Этот пример показывает, как передача именно текстового файла вместо файла формата DOC, позволяет уменьшить количество передаваемых данных.

Также программы часто можно получить в виде архивов.

Как понять что вам передают архив? Обычно архивы имеют расширение ZIP, RAR, 7Z, при многотомных архивах за этим расширением может быть номер тома.

Для создания архива в среде UNIX можно использовать популярную программу **7zip**. Или другие, которых много – по крайней мере, для Windows. Создание архива может занять много времени. Если вы прерываете этот процесс то архив будет дефектным. Время создания архива зависит, в том числе, от степени сжатия данных. В результате получается как минимум один файл.

Также обсудим *разархивацию*, то есть извлечение из архива (extract). Тут ВАЖНО иметь в виду что происходит создание новых файлов и\или каталогов, и они могут заменять уже существующие файлы. Это означает угрозу потери информации. Простейшее средство от этой угрозы – разархивация в пустой каталог.

Когда используется архивация? Несмотря на возможность изменять содержимое архива, архивация в основном связана с неизменными файлами – в том числе с резервными копиями. Их создание, то есть резервирование, можно считать работой администратора. Также архивация полезна когда вам надо перенести файл, больший чем пустое место носителя.

Для извлечения из многотомных архивов нужно чтобы все они были в одном каталоге.

То что программа распространяется в виде архива, не всегда означает потребность разархивировать её вручную. На некоторых ОС это важно знать так как, например, файлы формата DEB открываются в архиваторе. В этом нет никакого смысла.

Также каждый архиватор должен уметь показывать содержимое архива. Это значит, что вы узнаете что находится в архиве – без получения его содержимого. Вы узнаете имена файлов и каталогов, и, возможно, их размер, и другую информацию. Это позволит вам решить надо ли, и как обрабатывать архив. Такая возможность особенно полезна когда нет времени разархивировать.

Чтение и запись на носители.

Достаточно часто люди, перешедшие от Windows к UNIX, используют программу **Midnight Commander** – МС. По крайней мере так было. Потому что она позволяла выполнять многие операции без запоминания команд, и вообще напоминала прежде популярную программу **Norton Commander**, и **FAR**. И автор этих строк тоже сразу после установки ОС поставил и запустил МС. Однако она вызвала смущение из-за того что там не было ничего похожего на логические диски (тома). В UNIX – и на Linux, и на BSD – файлы и каталоги подключаемых носителей не рассматриваются как часть отдельной файловой системы, имеющей обозначение в виде буквы, а монтируются в уже существующую файловую систему компьютера и выглядят как её часть. Для этого в файловой системе существует или создаётся – автоматически или вручную – каталог, и он первоначально должен быть пустым. Потом – автоматически или вручную – туда монтируется файловая система носителя. В данном случае мы рассматриваем такие носители как лазерные диски или карты памяти, или флэш-брелки, а жёсткий диск, несущий файловую систему, содержащую ОС, тоже будучи носителем, выступает как устройство с корневой файловой системой, в которую встраиваются файлы и каталоги обозначенных носителей. Так, например, если мы имеем карту памяти с каталогом **PHOTOS**, то после монтирования её файловой системы в каталог **mountdir** содержимое каталога карты памяти будет доступно в **mountdir/PHOTOS**. Не требуется для каждого акта монтирования, или отдельного устройства, создавать новый каталог.

ВАЖНО – всегда следует отмонтировать носитель прежде чем его извлекать из разъёма, или отключать его питание. После отмонтирования, или отмены монтирования, носитель уже не виден как часть файловой системы компьютера. Разумеется, отменять монтирование можно лишь тогда когда файлы и каталоги носителя не используются. Это означает, например, что вы должны прежде дождаться окончания копирования файлов или каталогов, и лишь после этого можно отмонтировать носитель.

Простейший путь работы с носителями реализован в оконных средах вроде XFCE, LXDE, Gnome, KDE. Пользователь, не получая права администратора, может там простым образом отмонтировать носитель, а монтируется он автоматически при подключении – но тогда, когда есть драйвер под его файловую систему. Лазерные диски с файловой системой **iso9660**, так же как носители с файловыми системами, принятыми в среде Linux – например, **ext4** – прочтутся без дополнительных забот. Это значит что практически любой лазерный диск будет прочтён – если он не дефектный, и чистый физически. В файловом менеджере пользователь видит носители, их содержимое, и может их отмонтировать, и монтировать вновь. Файловый менеджер предназначен для выполнения операций с каталогами, файлами, и носителями без автоматизации и использования команд. В среде Windows такая программа называется Проводник.

Если в системе используется окружение рабочего стола, то оно, как правило, берет на себя всю работу чтобы выполнить монтирование флешки или карты памяти в Linux.

Может быть потому, что бесплатные ОС рассматриваются как отдельные от мира Windows, или по другим причинам, например, в среде Debian изначально нет драйвера для полноценной работы с файловой системой NTFS, принятой в Windows. Из этого вырастает факт, что носитель с этой файловой системой после монтирования не может быть полноценно использован. Его содержимое можно читать и копировать, но не перемещать и записывать. Иногда это – не проблема. Можно скопировать файл с носителя, и изменить его когда он записан на жёстком диске. Насколько я знаю, носители с флэш-памятью продаются с файловой системой FAT32. И они полноценно используются на Linux.

Носитель с файловой системой FAT16 имеет ограничение на размер логического диска и файла в 2 или 4 Gb. Этого может быть мало. Такие носители могут вовсе не потребоваться так как вышли из употребления. Администратор может сменить файловую систему носителя если сам носитель это позволяет. При этом – **ВАЖНО** – все данные теряются.

Сразу после установки базовой системы Debian (или чего-то вроде Raspbian lite – на микрокомпьютере) вы *не* получаете оконную среду. В таких условиях тоже можно монтировать носители, пользоваться ими, и отмонтировать. Но операции монтирования и отмонтирования требуют работы от имени администратора. Это означает, например, что коллективный, общественно-доступный компьютер не будет приспособлен к подключению носителей. Так как нельзя всем давать пароль администратора. Даже двум людям давать его

нежелательно так как они могут противоречить друг другу. Или надо настраивать монтирование. Этот вопрос описан в интернете и на man-страницах – в том числе в интернете на русском языке.

Более конкретно это выполняется так.

Если вы работаете как обычный пользователь, надо войти в систему как администратор. Для этого может потребоваться выйти из системы, набрав в консоли команду

```
exit
```

, и нажать **Enter** чтобы команда заработала. После этого вы получаете запрос на логин, и запрос на пароль. В качестве логина для администратора всегда используется **root**. Тут я рассматриваю использование не отдельных команд с правами администратора – при использовании команды **sudo**, а сеанс работы администратора. Команда **sudo** может не работать в Debian.

После входа как администратор вы используете команду для создания каталога – если надо – куда будет монтирован носитель. Например.

```
mkdir mountdir
```

Каталог будет создан в том каталоге, или в корне файловой системы, где вы находитесь. Заметьте, что каталог, созданный администратором, не может быть изменён неадминистратором.

Имеет смысл выйти из созданного каталога на один уровень выше. Потом подайте команду монтирования. Она для лазерных дисков обычно имеет следующий вид.

```
mount -t iso9660 устройство каталог монтирования
```

Чтобы узнать обозначение устройства можно использовать команду **dmesg**, но если у вас всего одно устройство чтения лазерных дисков, то оно называется **sr0**, и как файл расположено в каталоге **dev**. Таким образом, команда принимает вид

```
mount -t iso9660 /dev/sr0 ./mountdir
```

или то же, но с полным указанием пути к каталогу. Так, если каталог создан в личном каталоге пользователя **seloZuzukino**, то команда имеет вид

```
mount -t iso9660 /dev/sr0 /home/seloZuzukino/mountdir
```

После этого ваш лазерный диск станет виден как содержимое этого каталога. Но для записи на лазерный диск с файловой системой **iso9660** не годится простое копирование – как в случае с флэш-памятью.

Затем вы можете читать содержимое лазерного диска – без возможности создавать на нём каталоги и файлы. Часто лазерные диски и не требуется, и не нужно использовать так.

После этого в общем случае вы размонтируете лазерный диск. Для этого, получив права администратора как описано выше, подайте команду **umount**. В нашем случае она имеет вид

```
umount /dev/sr0
```

Для выталкивания лотка с лазерным диском подайте команду

```
eject
```

Использование флэш-брелков и карт памяти не очень отличается от описанного выше.

Подключите флешку и выполните:

```
fdisk -l
```



```

/dev/sdb6 158541824 1953501183 1794959360 855,9G Microsoft basic data
/dev/sdb7 1953501184 1953523711 22528 11M BIOS boot

Disk /dev/sdc: 14,5 GiB, 15514730496 bytes, 30302208 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x000b94f5

Device Boot Start End Sectors Size Id Type
/dev/sdc1 2048 30302207 30300160 14,5G b W95 FAT32

```

Узнайте имя файла, связанного с носителем. В данном случае это /dev/sdc1.

Теперь монтируем флешку с помощью команды mount:

```
sudo mount /dev/sdc1 /mnt/usb
```

Удостовериться что флешка подключена можно посмотрев содержимое каталога, в который мы ее монтировали:

```
ls /mnt/usb
```

После завершения работы с флешкой не забудьте её отмонтировать. Потому что иначе данные могут быть не сохранены или вообще повреждена файловая система флешки:

```
sudo umount /dev/sdc1
```

(по тексту сайта losst.ru)

Взаимодействие с другими устройствами.

Компьютер может взаимодействовать с разными устройствами, но при условии совместимости не только аппаратуры, но и программ. Какие это могут быть устройства?

Принтер. Сканер. Устройства связи – модем, роутер, точка доступа, адаптер для передачи данных через сеть питания. Веб-камера. Устройство бесперебойного питания. Аккумулятор – для ноутбука, смартфона или планшета. Смартфон или планшет. Мобильный телефон. Носители данных, читатели карт памяти.	Диктофоны. Внешние кодировщики видео – в том числе тюнеры. Другой компьютер. Микрофон. Наушники, колонки. Электронная книга. Умные часы. Факс. Фотоаппараты. Видеокамеры.
--	--

К смартфону или планшету интересно прежде всего подключать принтер и носители данных.

Обычно для подключения используется разъём USB – в разных его видах.

Другой компьютер можно подключить через нуль-модемный провод. Так в 90-е годы передавали данные. Звук обычно передаётся через разъём mini-jack. Гнездо Ethernet ещё обозначается как разъём RJ-45.

Мобильный телефон является, наверное, самым проблемным устройством так как он всё ещё используется как модем (не каждый телефон может так работать).

Конкретные указания – в инструкциях.

Глава 32. Интернет и локальная сеть.

Интернет – это всемирная сеть, включающая, помимо обычных, управляемых человеком, другие устройства (интернет вещей). То есть это совокупность соединённых устройств. Локальная сеть – также совокупность соединённых устройств, но гораздо меньшего масштаба.

Можно определённо сказать что доступ в интернет не только является крупным мотивом использования компьютерных устройств, но и существенным достижением. Почему? Интернет является не только доступным источником информации, но и транснациональным местом общения, и условием работы массы прикладных служб, а фактически – производств и организаций – в том числе критически-важных и коммерческих. Исчезновение связи через интернет было бы одной из самых крупных проблем.

Для планирования и обслуживания локальной сети, и получения выхода в интернет можно использовать отдельного специалиста или фирму. Но важно помнить что это лишь средство работы инфосистемы, а она – средство обслуживания людей. Вы можете рассчитать локальную сеть с выходом в интернет или без него с помощью сайта <http://www.netwizard.ru/>.

Зачем нужен выход в интернет.

Тем не менее стоит объяснить какой нам – простым пользователям – интерес иметь выход в интернет. Возможно, кто-то из читателей считает это чем-то вроде детской песочницы, где взрослым людям нечего делать. Это не так. Уже не разговоры, а факты показывают, что интернет стал одним из средств массовой информации. С другой стороны, именно через интернет происходит обновление тех программ и данных, которые устарели, или просто нужны для защиты. Получить их иначе возможно, но часто не так быстро. Интернет позволил объединить компьютеры для выполнения многих, в том числе несомненно гуманитарных задач, примером чего являются проекты в области медицины. Для всех стран возможность работать дома желательна и связана с выходом в интернет. В нашей стране с 2015 года доступны так называемые Госуслуги. Это набор практически полезных услуг. Но это не всё. Выход в интернет особенно интересен пожилым. Он позволяет не делать трудных поездок для получения справок, и так далее. В городах и пригородах, и даже с использованием обычной почты благодаря интернету вы можете делать покупки с помощью интернета даже не имея банковской карты. В частности, это покупки безрецептурных лекарств. Вы можете делать покупки практически любых товаров. Услуг много. Компьютерный ученик может не понимать зачем нужно много услуг, или как пользоваться услугой. Это нормально.

Какой нам нужен выход в интернет.

Теперь рассмотрим следующий вопрос – какой именно выход в интернет нужен.

- Достаточный по скорости для комфортного восприятия и правильного воспроизведения информации. Это свойство пока в России далеко не всегда и не везде обеспечено. Из-за огромной территории, и, в том числе, незнания частью населения о мотивации использования интернета, ситуация именно такова. Наверное, самые большие требования к скорости предъявляет показ видео из интернета. Для этого, ориентировочно, требуется реальная скорость не менее 512 кбит\с. Для комфортного чтения сайтов скорость в 100 кбит\с может быть недостаточной. То же верно про передачу звука, в том числе компьютерную телефонию.
- Непрерывный. Обрывы связи всё ещё возможны. Программы, передающие и получающие что-либо должны уметь минимизировать ваши проблемы из-за обрывов, а обеспечивать работу этих программ приходится в том числе пользователю.
- Без ограничений на объём переданных данных. Пока часто такой выход вне городов довольно дорог или не обеспечен вообще.
- Без тарификации по объёму переданных данных. Наиболее тяжёлая ситуация в этом отношении у пользователей, получающих выход в интернет через GPRS. Обратите на это внимание, владельцы старых телефонов!
- Общедоступный. Это, в том числе, означает наличие общественно-доступных пользовательских устройств. Появление их в заметной степени зависит от ожидаемой популярности, а последняя в условиях безграмотности и недостатка книг под большим сомнением. Заметим, что эти устройства не должны быть общественно-управляемыми так как это означает массу проблем. Для администрирования их требуется

квалифицированный, известный и штатный персонал. Тут – опять проблема для провинциальных районов. Заметим также, что долгая практика сама по себе не означает ни глубокой подготовки, ни правильности мнений. Появление компьютеров и выхода в интернет в библиотеках требует немедленного роста квалификации библиотекарей. Они, как представляется автору, часто не понимают важности этой квалификации, как и её глубины. Для решения этой проблемы нужны в первую очередь понятные учебники и курсы профессионального качества, и они должны быть доступны.

- Не требующий слишком больших трат на оборудование. При цене нового обычного компьютера в районе 20000 рублей это очень трудно. В таких условиях важно уметь оценить подержанное оборудование, и уметь использовать альтернативные устройства. Также это вопрос о работе общественно-доступной техники. Но тут важно понимать годится ли она для конкретного, частного применения. Так, например, одного часа выделенного времени в интернет-кафе может не хватить для продолжительного задания или работы, конфиденциальность под большим вопросом, как и наличие нужных программ. Также нельзя быть уверенным что ваши данные не сотрут – это, несомненно, из-за безответственности персонала, и отсутствия нужного управления со стороны руководства.
- Безопасный. Когда создавались первые сетевые инфосистемы это не выглядело проблемой. Не являясь специалистом в области защиты, могу определённо сказать, что интернет представляет угрозу даже в условиях серьёзных мер по защите. Когда риск слишком велик, или цена вторжения слишком велика, то инфосистемы вообще стараются не подключать к интернету. Каждый руководитель в наше время должен уделять пристальное внимание этому вопросу. То же верно про служащих. Отсутствие регламентации, как и игнорирование правил, невежество, отсутствие контроля и подготовки, беспечность, пьянство могут привести к тяжёлым последствиям. В условиях, когда телевидение и радио вообще не упоминают о важности защиты распространено ошибочное мнение что можно не заниматься защитой, и экономить на администраторах, и не поддерживать порядок в организации. Заметим, что во избежание тяжёлых последствий даже подготовленный рядовой работник не может управлять инфосистемой. Администрирование на работе – это не способ «проедания» бюджета, а очень ответственная, иногда тяжёлая и сложная работа.
- Доступный для понимания неспециалистом. Пока это проблема. В том числе потому что документация, написанная по поводу устройств связи и способа выхода в интернет, неудовлетворительна. Она полна неизвестных потребителю терминов, не всегда на русском языке, или, наоборот, очень поверхностна, или оставляет место для разночтений, то есть проблем. В городах к вам часто могут прийти штатные сотрудники фирм связи, и частные настройщики. Эта возможность не должна вводить в заблуждение что ничего не надо делать и знать. Вообще выход в интернет – часть большого раздела компьютерной науки о сетях. Автор, не являясь специалистом по связи, не может вполне закрыть этой книгой пробелы в российской литературе по сетям. У частного потребителя может не быть времени на освоение нужных вопросов. Решением таких проблем должно заниматься министерство связи. Нужен хотя бы один бесплатный учебник по сетям.
- Свободный от избыточного контроля, с защитой частной жизни и частных данных. На сегодня очевидная потребность в цензуре в широком смысле слова, и уважаемая потребность в приватности и хранении секретов соблюдаются достаточно радикальными мерами. Государства могут узнавать частную информацию, а пользователи и организации имеют возможность прятаться – вплоть до того, что для этого меняется протокол работы сети. Последствия обеих тенденций могут быть тяжёлыми.

Интернет и разные типы сетей.

Технически интернет, как и локальная сеть – это соединённые – не обязательно напрямую – компьютерные устройства. Интернет представляет собой «дерево» линий связи огромного размера. Локальная сеть, в отличие от интернета, часто имеет владельца, который может устанавливать нужные ему порядки в локальной сети. Локальная сеть может не иметь выхода

в интернет. Но даже если он есть, то конкретный компьютер в ней может не иметь выхода в интернет.

В чём особенности и отличия между разными сетями? Различают глобальные (интернет), региональные, городские, локальные, персональные сети. Назовём отличия.

- Административные. Локальные и другие местные сети имеют конкретный состав административного персонала, который может устанавливать конкретные параметры сети.
- Финансовые. Местные сети сами должны заниматься своим обеспечением.
- Функциональные. Локальные сети часто настроены для конкретных функций.
- В доступе. Локальные сети часто не годятся для использования произвольными, посторонними, анонимными пользователями.
- Государственные. Интернет создан и задуман как не принадлежащий ни одному государству. При этом каждое государство может по-своему организовывать свои сети.

Условия выхода в интернет.

Для полезного применения интернета требуется больше чем наличие оборудования, а именно наличие

- поддержки протоколов связи
- акта подключения
- программ, обслуживающих потребности пользователя
- оплаты

Выйти в интернет просто при наличии оборудования, самостоятельно нельзя. Можно выйти в локальную сеть. Выход в интернет обеспечивается фирмами – *провайдерами*, поставщиками услуг связи. Эти фирмы обычно не надо искать. Они известны и доступны простым образом. Часто это те же фирмы которые предоставляют услуги телевидения, телефонии. В России это, в первую очередь, Билайн, МТС, Мегафон, Tele2. Для использования выхода в интернет через спутниковую связь можно купить ресивер фирмы Триколор. Но он – самый дорогой из известных автору.

Несмотря на то что вопросы выхода в интернет обычно описаны провайдером, рассмотрим их.

Протоколы связи.

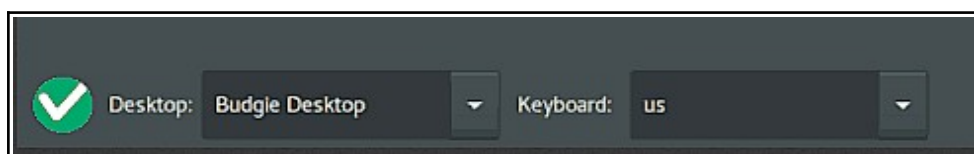
Что такое эти протоколы связи? Протокол связи – это способ организации взаимодействия сетевых, соединённых линиями связи, устройств. Протокол можно сравнить с этикетом. Так, например, при встрече людей принято здороваться. Протокол более сложен, чем простые правила этикета. Сам протокол знать не нужно. Его поддержка обеспечивается программами. Обычно по крайней мере популярные протоколы уже работают в свежееустановленной ОС. Windows давно предоставляет простой и наглядный способ управления протоколами. Вообще управление протоколами так или иначе возможно. Отсутствие их поддержки делает ОС во многом, или целиком, бессмысленной. UNIX изначально разрабатывался как сетевая ОС. Пользователи бесплатных систем могут не сомневаться что популярные протоколы связи поддержаны. В частности, это верно про Android, который на самом деле есть Linux после внесения небольших изменений в основу – ядро, и полной смены более поверхностных, видимых пользователю частей системы. Программы, позволяющие поддержать протокол, бесплатны.

Акт подключения.

Акт подключения существует для начала *сессии* работы, завершающейся актом отключения. В явном виде, без автоматизации, эти акты присутствуют достаточно часто – прежде всего в случае временного подключения к интернету через мобильный телефон. Практически важное преимущество наглядных, дружественных рабочих сред состоит в простом управлении наличием и свойствами *соединений* с интернетом или локальной сетью. Эти соединения не есть сами по себе линии связи, или подключения. Они – лишь наборы параметров, свойственных подключению конкретного типа. В данном случае тип подключения обычно определяется техническими средствами. Например, я пользуюсь – на Linux – рабочей средой Cinnamon. Она предоставляет мне возможность наглядной настройки таких подключений, то есть соединений:

- bluetooth – соединение на очень небольшом расстоянии, с низкой скоростью, обычно не для выхода в интернет
- ethernet – типичное проводное соединение, используемое, в частности, в локальной сети, а иногда для подключения модема, который позволяет выход в интернет не через ethernet, или спутникового оборудования для выхода в интернет
- infiniband – соединение для скоростной сети. Скорее всего, не нужно обычному пользователю, так как сейчас не является способом выхода в интернет, обеспеченным провайдерами
- Wi-Fi – популярное соединение через беспроводную локальную сеть, обычно с выходом в интернет
- мобильные – через мобильный телефон с функцией модема, или многие популярные модемы
- логические – не связанные с техникой для соединения; обычно такие соединения не являются заботой пользователя

Если вам мешает любая оконная среда или нечто вроде XFCE то даже не обязательно удалять такую среду. Достаточно иметь альтернативу, и настроить программную среду. Наглядная настройка делается в некоторых программах, которые заняты входом в систему, например, так.



В данном случае Desktop – это DE, в том числе набор программ. Наиболее спартанский вариант – Openbox. Можно просто временно или навсегда – до отмены – настроить программную среду для того чтобы оконная среда не начинала работу с помощью настройки **runlevel** (см интернет).

Также заметим, что порт USB достаточно часто применяется для выхода в интернет. К нему часто подключается модем, использующий SIM-карту.

Для доступа в интернет через спутниковую связь раньше можно было использовать DVB-плату. Последняя является вставляемой в компьютер платой, альтернативой подключаемых снаружи спутниковых приёмников. Однако она не гарантирует получение полезных данных сама по себе. Обратите внимание что фирмы спутниковой связи могут шифровать данные, и они становятся бесполезными для тех, кто не является их клиентами.

Старым, уже во многом ушедшим способом выхода в интернет является коммутируемый доступ через телефонную станцию (диалап, Dial-Up). Его нельзя путать с соединением через DSL-модем, который тоже использует телефонную станцию. Диалап имеет низкое ограничение скорости связи, поэтому он был проблемным ещё в середине 90-х годов. Дополнительная проблема в том что для этого типа связи типична повременная тарификация. Вы платите за время сессии – даже если вы ничего не получили. Модемы для диалапа могут быть дорогими, и могут поддерживать сжатие данных. Тем не менее избегайте этого способа.

Другой «свиньёй, подложенной под спокойную жизнь пользователя», является модемное или телефонное подключение через GPRS. Оно означает оплату объёма переданных данных – независимо от их полезности для потребителя. Такого соединения также следует избегать.

На сегодня смартфоны стали досягаемы по стоимости, поэтому выход в интернет через них более вероятен. Его нетрудно объяснить. Смартфон получает и передаёт данные либо как мобильные – через технику фирмы, чья SIM-карта в смартфоне, либо через Wi-Fi. С другой стороны, смартфон передаёт данные в компьютер – работая как модем. Так, например, в моём смартфоне есть параметр (то есть «настройка») «USB-модем». Когда эта функция работает то при выходе смартфона в интернет компьютер способен также обмениваться данными с интернетом через провод подключённый к смартфону.

Добиться этого от более примитивных, или старых телефонов может быть сложнее. Простейший путь – купить хотя бы дешёвый смартфон.

Возможно, ваш компьютер имеет встроенный адаптер Wi-Fi, но одно это не означает ни высокой скорости, ни выхода в интернет. В загородных домах может быть оправдано держать

роутер или точку доступа с подключённым к такому устройству USB-модемом. Также вы можете купить адаптер Wi-Fi для порта USB, но не для роутера, а для компьютера или микрокомпьютера. Обратите внимание, что он, как и модем, не обязан работать с редкими ОС – UNIX в частности. Чтобы узнать об этом подробнее смотрите интернет. Модель адаптера UPVEL UA-210WN имеет официальную поддержку Linux, и обеспечивает скорость не более 150 мегабит в секунду, что реально может не выполняться, но в идеальном случае эта скорость для негородских поселений более чем достаточна.

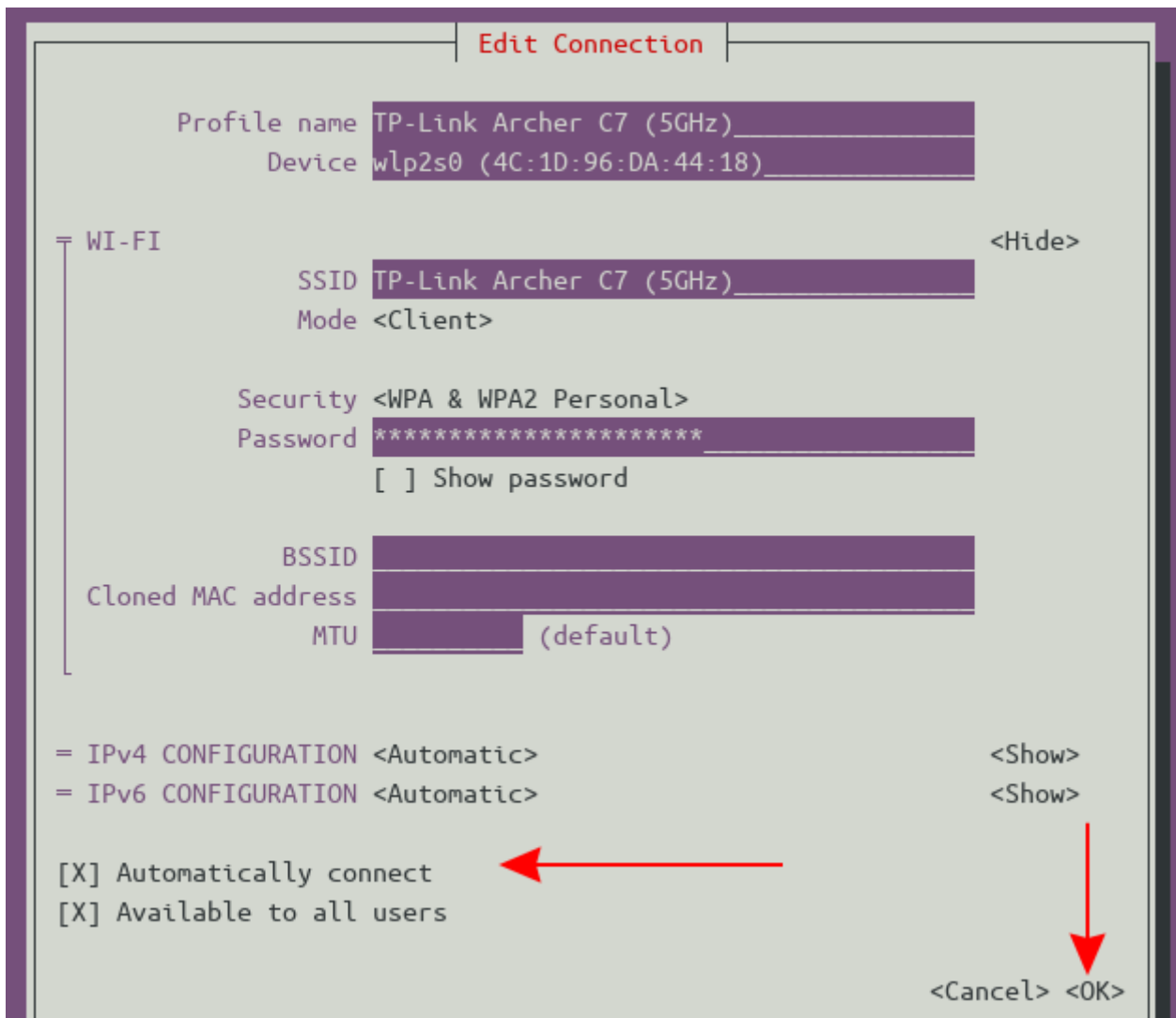
В случаях когда у вас нет наглядной рабочей среды, вы всё же можете настроить соединение, но есть большая вероятность, что провайдер не сможет вам помочь. В таком случае выйдете в интернет с телефона, и поищите там инструкцию.

Сейчас я писал про компьютер. Планшет, и телефон, обычно умеют подключаться к интернету без сложностей. В хорошем случае они умеют подключаться через Wi-Fi. Очень вероятно, что вы увидите значок, обозначающий выход в интернет.

Вы всегда можете позвонить провайдеру по бесплатному для вас номеру телефона. Но требуется ориентироваться в ситуации, как минимум назвать свои действия, описать свою ситуацию. Я пока не видел провайдера, официально поддерживающего какие-либо ОС кроме Windows. Тем не менее другие ОС работают с тем же каналом, но иначе настраиваются.

Проблемой могут стать обрывы связи. В ситуации обрывов связи надо обратить внимание на угрозу получения дефектных файлов. Также обрыв связи останавливает поток видеоданных, общение через интернет-телефонию. Всегда убедитесь что после обрыва связи возможно автоматическое восстановление связи. В моём опыте модем МТС Коннект-4 не обеспечивал эту функцию.

Nmtui является понятной программой для настройки выхода в интернет. Программа понятна при знании вопросов создания соединений, и непонятна без этого знания. Программа работает в консоли и терминале, и позволяет подключаться даже к Wi-Fi-сетям.



Руководство по подключению к интернет с иллюстрациями.

Рассмотрим выход в интернет с помощью средств, предоставляемых наглядными рабочими средами, которые не связаны с основной системы, и используются в разных видах UNIX. Среди таких сред можно выделить следующие.

- Openbox. Рассчитан на подготовленных людей, и даёт высокую скорость работы при низких требованиях к вычислительной мощности и объёму оперативной памяти.
- XFCE. Лёгкая среда, обеспечивающая лишь выход в интернет через Ethernet-соединение. Для управления другими типами соединения требуется дополнительная программа, которую может потребоваться брать в интернете, или настройка.
- LXDE. Лёгкая среда, позволяющая сразу настроить выход в интернет через Ethernet.
- Более требовательные к мощности и объёму оперативной памяти среды – прежде всего GNOME и KDE. Позволяют сразу настроить выход в интернет.

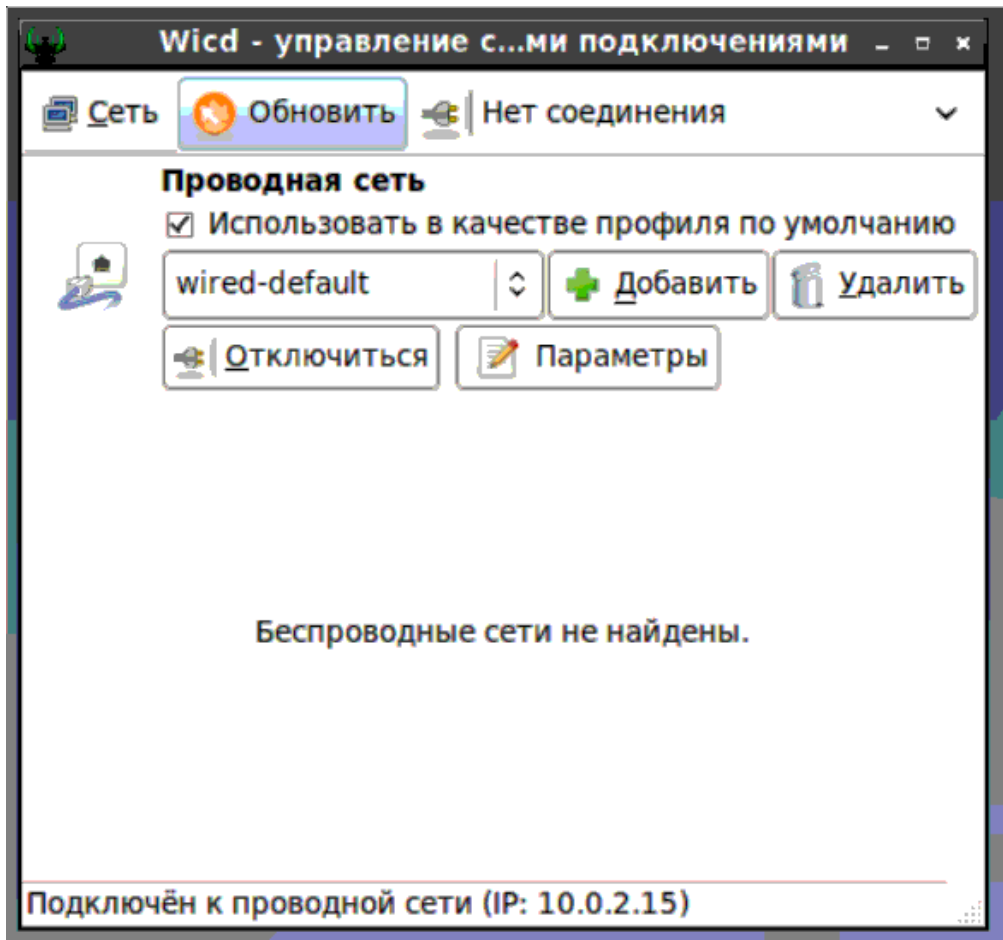
Разумеется, любой выход в интернет настраивается и работает когда его позволяет оборудование, то есть существует линия связи. Обычно под линией связи понимается выделенная линия, тут это обозначает просто канал в интернет.

Если установка ОС проходит при подключённом проводе Ethernet, то после её окончания и перезагрузки компьютера обычно можно видеть значок, указывающий что провод подключён (на примере Debian и LXDE).

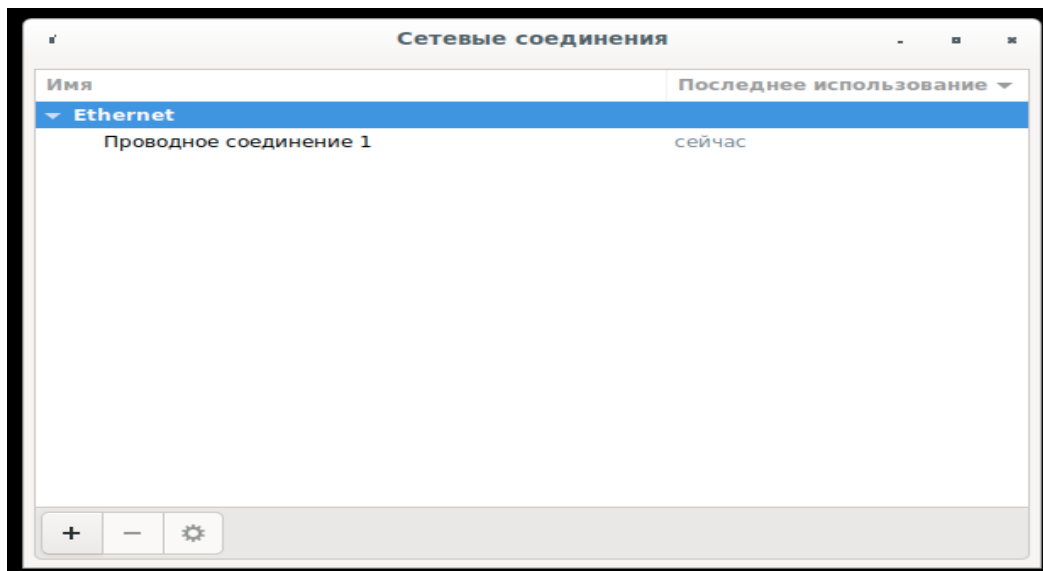


Если при подключённом проводе связи нет, но настройки верны, то стоит проверить провод, и обратиться к провайдеру, или проверить роутер или точку доступа.

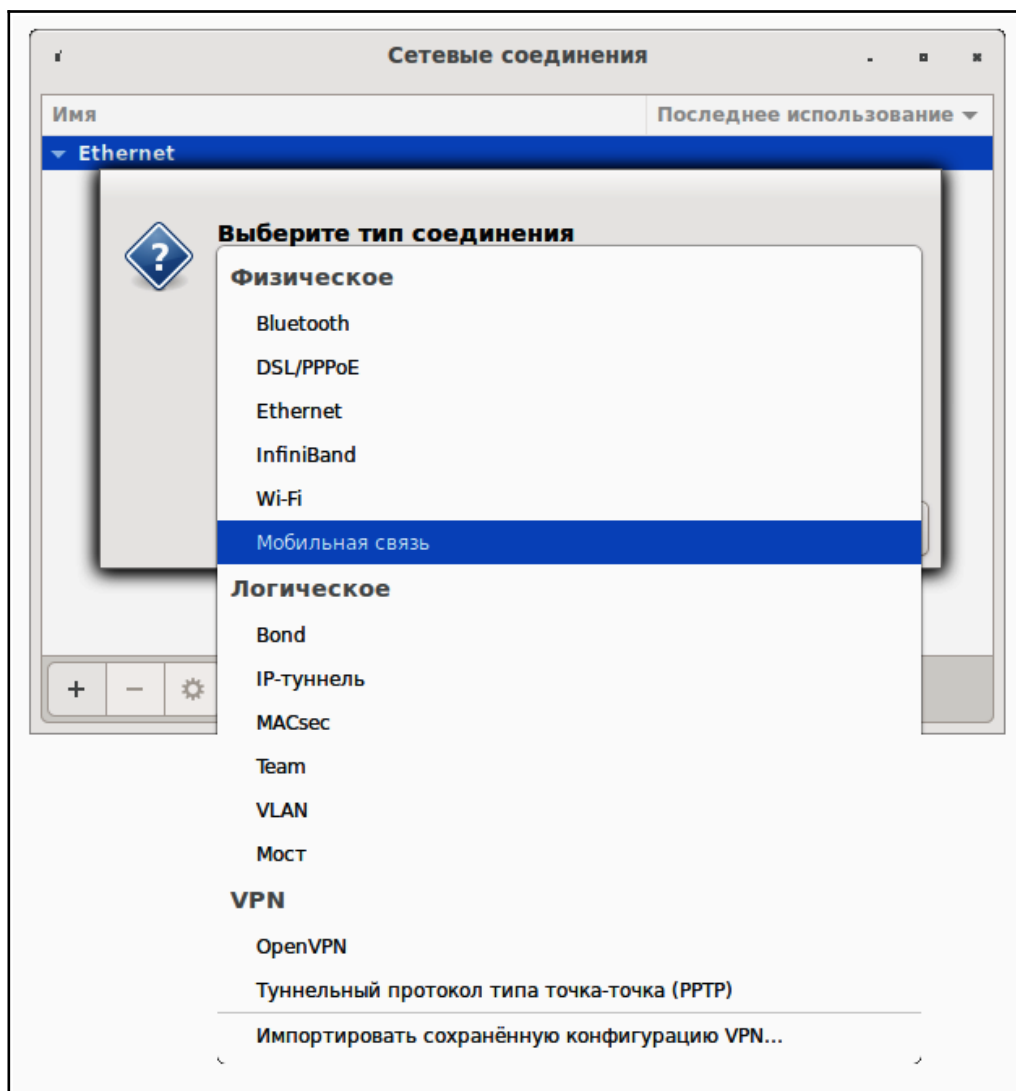
Соединения создаются простым образом. Не обязательно иметь одно соединение. На показанном значке каким-либо нажатием кнопки мыши можно вызвать окно для настройки соединений. В данном случае оно такое.



В случае с Cinnamon ситуация намного лучше для выхода в интернет с телефона или модема. Вам предлагается удобный способ подключения с помощью показанного окна.



В данном случае уже есть соединение через сетевую плату. Оно не обязательно. Нажимая плюс, вы получаете следующее окно с возможностью выбора нужного типа соединения.



Так же создаётся подключение в среде MATE, это DE.

Для подключения в консоли есть программы, не требующие оконной среды. Это, например, программа **ppp** (на Debian). Чтобы узнать о настройке соединений в консоли смотрите страницу на <https://losst.ru/upravlenie-networkmanager-iz-konsoli>, а также на <https://serveradmin.ru/nastroyka-seti-v-centos/>. Это будет нужно при использовании микрокомпьютеров и серверов без оконной среды.

Связанные с интернетом программы.

С интернетом связаны многие программы. Сейчас нас интересуют не все они, а наиболее близко связанные с интернетом. Это, прежде всего, программа, позволяющая работать модему. По поводу её установки и использования лучше всего обращаться к документам провайдера, и его телефонным службам поддержки. В моём опыте установка таких программ на бесплатной ОС не нужна, и даже, возможно, вредна. Вместо этого может потребоваться некая настройка – ищите информацию в интернете. Проще всего купить модем с поддержкой Linux. Сейчас, насколько я понимаю, подключение через модем с SIM-картой может вообще не потребовать настройки. Это, видимо, происходит так из-за особенностей ядра Linux. Я являюсь сторонником и «рекламным агентом» ОС FreeBSD. Там может потребоваться настройка для работы такого модема. Но этот вопрос решается со временем, и по мере изменения обстоятельств. В качестве обстоятельств выступает, например, популярность или новизна техники.

С интернетом определённно связана сетевая подсистема – присутствующая даже в самых базовых экземплярах ОС. Чтобы привести её в состояние, когда возможна передача данных в интернет и из него, может потребоваться перезагрузка этой системы. Другими словами, перезапуск. Проще всего сделать его путём перезагрузки ОС.

С интернетом связана такая известная программа, как **ifconfig**, или **ip**. Она позволяет – в потомках UNIX – посмотреть прежде всего на готовность ОС к работе с интернетом через сетевую плату для Ethernet. Я сейчас именно так выхожу в интернет через городской проводной канал. Команда

```
ip link
```

показывает мне – кроме прочего, следующее:

```
1: lo: <LOOPBACK,UP,LOWER_UP  
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP
```

Это значит, что сейчас сетевая подсистема содержит два сетевых интерфейса: первый – это «петля» – практически бесполезный мне сейчас, и второй – у сетевой платы, работающий (UP).

Показанная команда требует прав администратора.

Также с интернетом связаны многие программы, управляемые в UNIX из консоли. Одна из таких команд – **ping**. Подача её с адресом сайта, *отзывающегося* на эту команду, позволяет проверить наличие выхода в интернет. Выглядит это так:

```
ping -c 1 ya.ru  
PING ya.ru(ya.ru (2a02:6b8::2:242)) 56 data bytes  
64 bytes from ya.ru (2a02:6b8::2:242): icmp_seq=1 ttl=51 time=7.20 ms
```

```
--- ya.ru ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 7.203/7.203/7.203/0.000 ms
```

Здесь **ya.ru** является адресом известного, постоянно (надеемся) работающего сайта, **-c 1** – параметр команды, определяющий количество (1) актов подачи набора данных – пакета, и количество актов ожидания ответа от сайта. Команда не требует административных прав для своего выполнения. Предпоследняя строка показывает что 0% посланных пакетов утеряно (loss), и один получен (received). Мы убедились, что сайт доступен с помощью этой команды. Это значит что выход в интернет есть.

Выход в интернет в большой степени обеспечивается ОС. Тем не менее, существует масса программ, работающих поверх ОС, и связанных с интернетом. Наиболее часто используется браузер, который будет рассмотрен ниже.

На планшетах и телефонах выход в интернет можно проверить выйдя на любой сайт.

Оплата.

Обычно выход в интернет в Москве ничего не стоит, но его использование оплачивается.

Это, в частности, так даже если вы ничего не делали, но потратили время в сессии – при повременной оплате. Способы оплаты определяются провайдером. Он заинтересован дать своим клиентам как можно более простой и достигаемый способ оплаты. Это может быть, например, оплата в банкомате. Вы можете платить вперёд. Это определяется документами, и тарифом. Каждый пользователь должен внимательно читать, и понимать документы провайдера. Оплата часто является ежемесячной. Часто если вам не нужен интернет, то вы можете выполнить блокировку соединения. Это не обычное завершение сессии, а изменение в обслуживании провайдером. Иногда можно найти тариф с подневной оплатой. Это имеет смысл если вы используете интернет не каждый день.

Доступность и защита сети Wi-Fi.

Такая сеть часто используется для комфортного использования прежде всего носимых устройств. Теперь каждый ребёнок может обращаться к интернету из скворечника на вашем садовом участке. Это очень удобно – не тянуть провода линии связи по дому. Скорость связи часто достаточно высока – если вы не ушли слишком далеко.

Беспроводная сеть принесла заметное облегчение в конторы, позволив, например, иногда избавиться от мешающего провода к принтеру.

Каждый пользователь должен знать что реальная скорость связи меньше написанной на коробке с техникой. К тому же бетонные стены, металлические крыши, деревья ослабляют сигнал. Это может привести к падению скорости. Скорость не гарантируется. Для усиления сигнала можно, например, купить повторитель. В качестве повторителя (то есть репитера) иногда можно использовать даже те роутеры, которые для этого не предназначены, и многие смартфоны. При этом скорость будет падать, но охват сигнала вырастет.

Типично, что городская семья, имеющая выход в интернет любого типа, имеет свою локальную сеть Wi-Fi. Это полезно даже просто для переноса данных между семейными устройствами. Для этого используется роутер, или точка доступа. Но в этой ситуации многие пользователи проявляют легкомыслие, и не обеспечивают защиту локальной сети. Такую ситуацию можно понять. Фирмы присылают мастеров, используется планшет, и вообще всё очень просто и здорово. Инструкции к роутеру и точке доступа могут быть непонятны. Однако результат плох: вашим выходом в интернет могут пользоваться соседи. Они не будут за него платить, и могут даже получить доступ к вашим файлам. Защита сети может зависеть от встроенных программ роутера или точки доступа. Вы заинтересованы иметь возможность обновить их. Это простое дело.

В качестве интересного предложения выступает роутер или точка доступа с возможностью подключения к ней модема. Модем является клиентским устройством относительно техники провайдера, и можно не покупать компьютер для соединения его с роутером или точкой доступа. Если вы получили подержанный роутер или точку доступа, то там могут быть нестандартные, посторонние ОС и программы.

Также для создания сети Wi-Fi могут использоваться многие телефоны, планшеты и Wi-Fi-адаптеры. Последние должны уметь работать с вашей ОС. Проверьте это. Однако важно понимать что, создавая сеть, вы, скорее всего, отключаетесь от Wi-Fi сетей. Телефон будет передавать данные как мобильные данные через SIM-карту, или подключённый через провод модем – если удастся заставить его работать.

Теоретически возможно использовать Wi-Fi для поселковой локальной сети. Убедитесь, что договор с провайдером не противоречит этому. Некоторые фирмы выполняют коллективное подключение более чем одного дома.

Проблемы с сигналом мобильной сети.

На огромных просторах нашей страны часто единственным провайдером выхода в интернет является продавец телефонных услуг связи. Провалы скорости – пока ещё обычное явление для такого подключения, но постоянно низкая скорость означает что у вас плохая связь с ближайшим оборудованием провайдера. В такой ситуации можно, например, купить подходящую антенну. Но – ВАЖНО – может потребоваться грозозащита, и многие телефоны не в состоянии работать с антенной – у них нет разъёма для этого. Также вы можете купить усилитель сигнала. В Московской области, например, многие фирмы предлагают быстрый канал в интернет.

Что я получу в интернете?

Вот примерный ответ.

	Что	Где	
	Справочную информацию – часто эта информация быстро и непредсказуемо меняется	Прогноз погоды, курсы валют, телепрограмму, адреса, телефоны, сообщения о транспорте и так далее.	На виджетах в телефонах и компьютерах, и на планшетах, на сайтах. Можно получить с помощью голосового помощника вроде Яндекс Станции (это устройство).
	Новости вроде телевизионных		На сайтах, в лентах новостей, в интернет-видео, в архивах телепередач.
	Статьи.	Самого разного содержания. В том числе старые – информация может быть уже неверной.	На сайтах. В блогах.
	Книги.	В том числе очень старые, часто с нарушением авторских прав.	На сайтах, иногда – в раздаче через торрент.
	Фильмы.	То же.	То же.
	Музыка.	Для не старых звукозаписей – часто с нарушением авторских прав.	На сайтах.
	Поток звука интернет-радио.		Часто на сайтах или получаемый через соответствующие программы – привязанные к радиостанции или нет. Архивы старых передач.
	Телепередачи.		Их легче всего найти на сайтах телеканалов, или в хранилищах видео вроде youtube.
	Поток звука эфирных радиостанций.		На сайтах этих радиостанций, или надо получать с помощью программ для этих радиостанций.
	Общение.	Заметьте, что оно может быть прослушано, и не оскорбляйте собеседника. Не утверждайте того что не знаете точно, и помните, что обвинение может быть поводом для суда.	<ul style="list-style-type: none"> • через электронную почту • через другие программы, пересылающие текст • на сайтах с форумами, на сайтах социальных сетей • через программы, пересылающие звук • через программы, пересылающие звук и видео с камеры
	Интернет-телеканалы.	Требуют высокой скорости передачи данных.	Нечто похожее на интернет-радио, со звуком.
	Программы.	Всегда помните что программа должна быть	Программы обычно поставляются либо как готовые

	совместима с ОС и оборудованием.	к использованию, либо как материал для компиляции. Будьте осторожны, особенно со скриптами.
Документацию и примеры использования программ.	Обычно на английском.	В том числе map-страницы. Достаточно часто – готовые к печати публикации.
Операционные системы.	Не всегда адаптированы для работы с кириллицей, и не всегда пригодные для конкретного человека или вообще.	Обычно поставляются как содержимое виртуальной машины, либо как файл-образ лазерного диска.
Платные сервисы.	Всегда убедитесь что оплата производится через банк или наличными. Другие способы могут быть подозрительными.	Просмотр сайтов и электронная почта почти всегда бесплатны. Сервисы самого разного рода. Интернет-магазины, аптеки и многое другое. Например, аренда ресурсов виртуальных машин.
Доступ к удалённой технике.	Может быть нужно шифрование передаваемых данных.	Не всегда нужно обычному пользователю в домашней практике. Скорее на работе.
Связь с людьми для выполнения платной работы.	То же.	
Электронные платежи, использующие традиционные валюты.	То же.	Скорее всего будет нужно при покупке в интернет-магазинах.
Электронные платежи, использующие криптовалюты.	То же.	Обычно не используются в России.
Бесплатные сервисы.		Особенно полезны владельцам дешёвых и старых телефонов.
Интернет-игры.	Может быть нужен быстрый канал.	Часто – пустая трата времени.
Системы распределённых вычислений.	Полезны в науке.	Обычно не используются, и не обеспечивают оплату.
Передача данных или программ со скрытием адресов или других идентификационных данных. Системы прямой передачи данных.		Часто содержат нелегально распространяемые материалы.
Интернет-хранилища файлов.	Могут быть полезны для доступа к одним и тем же данным с разных устройств.	Иногда платные услуги. Секретность хранения под сомнением.
Интернет-энциклопедии, словари.		Полезны и бесплатны, просматриваются в браузере.
Поисковые машины.	В основном поиск слов для поиска текстов.	Необходимы почти всегда. Бесплатны.

Споры в интернете проходят с тяжёлыми потерями для самолюбия. Вы в России – будьте осторожны.



Браузер.

Слово «браузер» (browser) используется в разных значениях. Нас интересует наиболее часто подразумеваемое значение – программа для просмотра сайтов. Эта глава написана чтобы вы могли с помощью браузера самостоятельно получать информацию, и учиться. Повторяю – мной не ставилась задача написать огромное, полное описание всех вопросов прикладной компьютерной науки. Вы сами должны уметь искать и получать информацию. Первым средством для этого является в наше время браузер.

Прежде чем описывать сами браузеры и другие вопросы, хочу сделать важное замечание, которое особенно важно когда мы обсуждаем браузеры и вообще ситуацию когда вы имеете выход в интернет. Тема браузера и сайтов очень сложна. К счастью, вам не надо знать её как специалисту по оказанию услуг через интернет. Чтобы организовать – в том числе в сознании ученика – информацию по этой теме стоит подумать о том в чём суть описываемых вопросов. Суть проста – вы получаете некие услуги. Поняв это, многие люди делают большую ошибку. Они обращают внимание лишь на то что непосредственно им нужно. Правильный в принципе подход приводит к игнорированию массы вопросов, которые на самом деле важны. Это примерно то же что иметь автомобиль, и думать только о том куда надо ехать – не интересуясь наличием бензина, масла, и прочим. Такой подход мог годиться для примитивной телеги, которая была создана много сотен лет назад. Компьютерное устройство намного сложнее любого автомобиля. Но важнее то что *работа* с ним сложна. Этот факт по-настоящему пока ничем нельзя компенсировать. Заметим, что ленты новостей (RSS) позволяют получать «статьи» так же быстро как страницы сайтов – или даже быстрее – но эта технология не позволяет иметь *взаимодействие* с сервисами сайтов – о чём речь ниже.

Эта тема трудна тем что она – прикладная, и из-за множества смежных вопросов, которые может быть надо затронуть чтобы читатель мог учиться сам, и понимал написанное. Браузеры в том или ином виде есть во многих телефонах, всех планшетах, и обычно есть на компьютерах – по мере надобности. На микрокомпьютере браузер тоже можно иметь, но вычислительная мощность и объём оперативной памяти могут быть недостаточными для конкретного браузера, или его использования. О браузере на роутере или точке доступа можно говорить лишь с оговоркой, что этот браузер должен работать в очень стеснённых аппаратных условиях. Поэтому там полноценный браузер не может работать.

Браузер – это программа, которая требует большого объяснения. Таких программ – браузеров – достаточно много. *Важно понимать что они могут быть неполноценными.* Какое это имеет значение? Использование неполноценных браузеров мне достаточно трудно обсуждать. Я – не специалист в технологиях, на основе которых работают браузеры. Думаю, что задача простого чтения текстов, как и, по-видимому, скачивания файлов, вполне безопасна даже для самых примитивных браузеров. С другой стороны, у пользователя есть гораздо более рискованные задачи – в том числе управление деньгами. Они выполняются лишь с помощью развитых, популярных браузеров. Но даже эти браузеры не гарантируют отсутствие риска. Наименее проблемными, популярными браузерами являются Firefox,

Chrome, Safari, Opera. Я надеюсь, что браузеры для телефонов UC WEB, Opera Mini достаточно развиты и безопасны. На самом деле браузеров много. Думаю, что браузеры Sea Monkey, Vivaldi – сделаны хорошо.

Браузеры – клиентские программы относительно серверных программ, и серверных компьютеров. То есть браузеры являются получателями информационных услуг, предоставляемых серверными программами. В качестве простейшей такой услуги выступает получение информации. Но само это определение грешит неким формализмом. Так как полученная информация может быть, и часто является не пользовательской, а технической. Что это значит? Что часть поступивших данных не предназначена для буквального восприятия пользователем. Такая ситуация типична в работе браузера. Заметим, что эти данные отвечают не только за вид получаемых страниц, но и за *возможности*. Что это за возможности? Это, в первую очередь, возможности встроенных в технические данные программ. Эти программы не нуждаются в установке, и об их работе пользователь может не знать. Это не должно вызывать панику. Это должно подчеркнуть важность грамотной работы, и применение свежих версий полноценных браузеров. В первые дни 2018 года в интернете появилось много сообщений об уязвимости Spectre. Она имеет прямое отношение к браузерам, и вам нужна версия браузера, созданная с защитой от угроз, создаваемых этой «дырой». Вы не обязаны управлять деньгами через браузер, и банки не заинтересованы в потере вами вкладов. При этом есть много других потенциально угрожающих вам ситуаций. По-настоящему их могут описать лишь специалисты – не автор. Ваша текущая задача – получить достаточно знаний для обучения с помощью браузера.

Всё, что было сказано выше, неявно предполагает некий развитый браузер, и это HTML-браузер. Что это значит? Исторически первой функцией браузеров была возможность показа *гипертекста*. Простейшее описание гипертекста – это текст, связанный *ссылками*. То есть у нас есть возможность перемещаться от одного текста к другому, или между частями одного текста с помощью ссылок. Ссылка является текстовой конструкцией – не более сложной чем школьная задача по математике. Но браузер не показывает её буквально в том виде как получает, а использует для перемещения по текстам. Внешне ссылка обычно выглядит как подчеркнутый текст, обычно курсор мыши на ссылке принимает вид руки с указывающим пальцем. Однократный щелчок левой кнопкой мыши на ссылке заставляет браузер показать другую часть текста, или другой текст – возможно, на другой странице. Например, следующий текст «смотри известную статью» включает в себя ссылку во втором слове. Но, разумеется, возможно и просто подчеркивание без ссылки.

Страница – это прямоугольная лента, которая *технически* представляет собой текст, написанный или созданный автоматически в соответствии с правилами HTML. Во внешнем выражении на экране компьютерного устройства страница обычно сильно напоминает газету – с той разницей, что там могут быть видео, анимация и звук, а также некие части, обладающие интерактивностью – например, кнопки.

Так как с помощью ссылки вы обычно запрашиваете другую страницу, то браузер ищет её. Но поиск может окончиться ничем. В таком случае, скорее всего, вы получите сообщение об отсутствии страницы. Возможно, она всё же существует, но находится по другому *адресу*. Или вы получите сообщение об ошибке 404, которое означает то же. Некоторые сайты просто показывают свою главную страницу – которую вы не запрашивали.

Вот мы и пришли к важной теме адресов. Адреса важны потому что они указывают на разные ресурсы, которые могут быть полезны. Обычно не надо запоминать адреса. Они указывают не на название материала, а на его *место* в интернете или локальной сети. Адрес может выглядеть совершенно формально, и вы не сможете его запомнить. Ссылка невидимо содержит адрес. Во-вторых, адрес часто содержит техническое, понятное программам, обозначение того как *создать* нужную страницу. Поэтому часто адреса занимают несколько строк. Неверно, что каждая страница создана вручную. Страницы генерируются с помощью программ, и баз данных. Эта автоматика позволяет получать адекватные страницы в любое время. Они настолько свежи насколько свежи базы данных. Зачем это может быть нужно? Например, есть сервис, то есть, по-русски, служба, обеспечивающая прокладывание маршрута между разными городскими адресами. Так как страницы могут включать в себя интерактивные элементы, и собирать данные, то это возможно. Вы определяете два места, между которыми нужно проложить маршрут. Реально это делается на одной странице. Но мы можем представить нечто другое, сервис, создающий страницу с маршрутом. Тогда некая

программа должна получить с одной страницы оба географических адреса, и создать страницу с маршрутом. Всё это происходит автоматически. Данные об адресах могут быть заложены в адрес создаваемой страницы.

Справится ли с этой задачей примитивный браузер? Возможно, он даст тот же *наблюдаемый* результат. Но это лишь внешняя сторона.

Полное обозначение адреса стандартизировано, и включает в себя обозначение протокола связи. Протокол определяет как будут автоматически взаимодействовать программы на сервере и клиенте. Клиентом в данном случае является техника пользователя. Какой нам интерес обсуждать протокол? Это может быть важно при ограничении на месячный объём передаваемых данных, и при медленном канале. Протокол HTTP, и, возможно, HTTPS является более затратным с точки зрения количества переданных данных, чем FTP, и SFTP. Также я сталкивался с тем что ссылка имела адрес, не обрабатываемый браузером, из-за протокола MAGNET. Возможно, что это вообще не обозначение протокола, но браузер этого не «знает». Протоколы HTTPS и SFTP используют шифрование передаваемых данных. Это, с одной стороны, хорошо – в плане секретности. С другой стороны, есть вероятность получить увеличение вычислительной работы, что для микрокомпьютеров может быть важно. Вам не требуется знать как именно работает протокол.

Также каждый пользователь должен понимать что адрес вида **mailto:XXX@YYY** не является адресом страницы, а представляет адрес электронной почты. Тут буквы XXX и YYY ничего буквально не значат, а просто обозначают некие буквы почтового адреса.

HTML представляет собой, можно сказать, свод правил для оказания сетевых услуг с помощью браузеров. Версия HTML 4 на сегодня, насколько мне известно, уже считается устаревшей. Однако для ученика она – хорошая база для обучения. Пользователю не надо знать HTML. Но я буду опираться на него для описания использования браузера. Этот стандарт позволяет иметь на страницах

- тексты на множестве языков – даже на одной странице. Тексты выглядят примерно так же как в книге. Заголовки, курсивный шрифт, и так далее. Разные шрифты
- статичные иллюстрации, и анимированные иллюстрации
- таблицы разного вида – в том числе без видимого деления на ячейки
- звук
- цветной фон, и цветной текст
- списки, то есть абзацы как пункты, части набора высказываний. Вы читаете список
- видео
- фреймы, то есть нечто вроде страниц на странице
- горизонтальные линии
- программы – скрипты. Это текстовые программы
- элементы управления – в общем аналогичные используемым в окнах программ

Заметим, что обычно при показе страницы полезное содержимое умещается в окне браузера. Если окно меняет ширину, то это содержимое часто перестраивается. И снова умещается в новой ширине. В этом – существенное свойство страниц. Оно позволяет уместить материал на узком экране. Готовая к *печати* публикация – например, в виде файла формата PostScript не может перестраиваться. И не должна. Она рассчитана на листы известной формы – обычно на листы формата А4.

Версия стандарта HTML 5 – большой шаг вперёд. На сегодня ни один браузер до конца не соответствует 5-й версии.

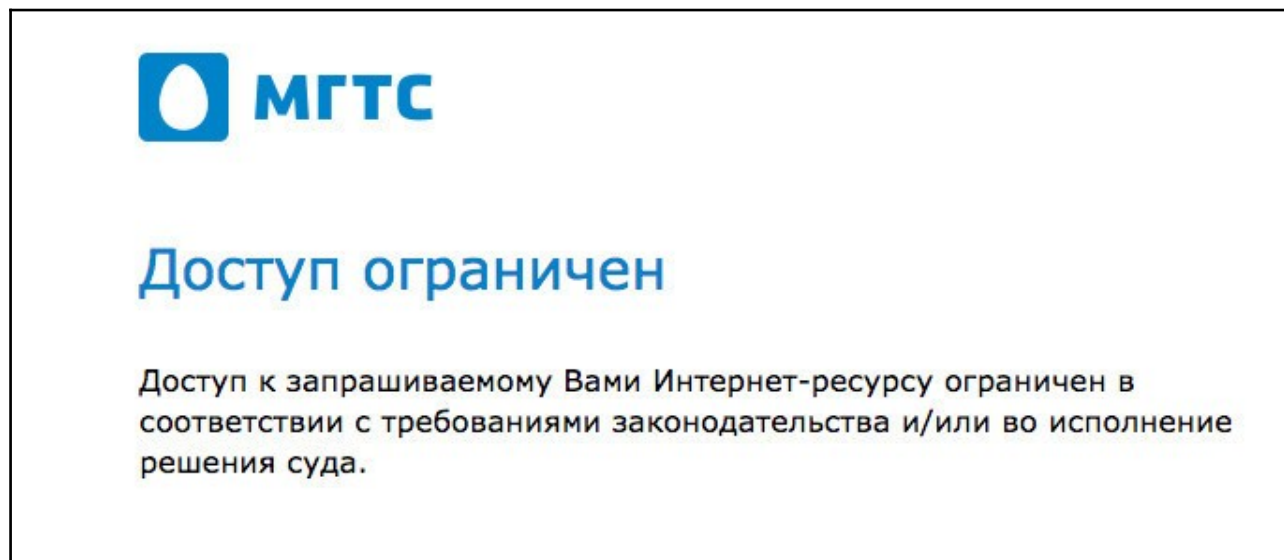
Для написания встроенных в страницы программ широко используется язык программирования JavaScript. Какое это имеет значение? Блокирование этих программ может исказить работу сайта.

Вот мы и дошли до понятия сайта. Сайт достаточно трудно определить так как конкретные сайты могут быть самыми разными. Дать формальное определение достаточно легко. Сайт – это совокупность страниц с одним адресом – точнее, с основной его частью. Работа сайта часто не сводится к простой выдаче страниц, а предоставляет *сервис*. В качестве примера можно назвать карты (географические) Яндекса, сайты государственных услуг, и так далее. Отличие сервиса в более-менее развитой обработке данных, и, в особенности, произвольных *пользовательских* данных. Так, например, сервис Wolfram Alpha обрабатывает произвольные математические выражения. Он готов к неправильным, некорректным выражениям, которые не ломают его. Также сервис SageMath позволяет вводить произвольные выражения, и

обрабатывает их по мере возможности. Подобных примеров можно привести много.

Сайт может хранить на вашем устройстве сравнительно немного данных. Это часто делается с помощью «куков» (cookies). Это текстовые файлы, размер которых ограничен стандартом. HTML 5 вводит в употребление более ёмкое хранилище данных сайта. Блокирование возможности хранения данных может исказить работу сайта.

Подозрительные и опасные сайты могут быть известны поисковой системе. В таком случае она выдаст предупреждение, или даже это сделает браузер (но вряд ли консольный). Такие сайты могут воровать информацию о банковских картах, и так далее. Большинство сайтов не опасны. Кроме того, государство блокирует сайты, занятые раздачей чего-либо с нарушением закона. Так, например, известный сайт, на котором была раздача торрентных файлов с нарушением авторских прав, заблокирован. Вместо этого сайта сделана заглушка. Вернее, сайт существует, но провайдер его не пропускает.



Реклама на сайтах может стать препятствием для использования сайта. Это реальная проблема для ученика, и объяснить рецепт трудно. Важно знать что реклама часто делается как покрывающий слой поверх нужной страницы. Обычно реклама, скрывающая страницу, закрывается сама или её можно закрыть нажатием кнопки на рекламном видео. На смартфонах и планшетах реклама может занять весь экран. Это не сбой, или проявление вируса, а следствие малого экрана, и настройки сайта.

Совокупность возможностей, предоставленных браузером, позволяет создавать сетевые, точнее, браузерные (вэб-) приложения. Их не надо устанавливать, и, скорее всего, они не зависят от ОС. Возможности таких приложений достаточно велики. Заметим, что такие приложения являются потенциально интересными для региональных пользователей так как их не надо устанавливать. Это особенно важно в библиотеках и интернет-кафе, где установка программ разрешена лишь администратору. Так, например, в браузере вы можете выполнять некоторые работы, которые в «классической» программной среде выполняются в отдельной, иногда довольно массивной программе. Браузерное приложение, как и сайт, может очень медленно работать на медленных каналах выхода в интернет. В общем эта возможность – для быстрых каналов – городских, либо библиотечных, или интернет-кафе. Беспроводная сеть может не дать достаточной скорости. Каждый работник интернет-кафе и библиотечного интерне-класса должен понимать что можно использовать удалённые программы сайта rollapp.com или VPS-сервиса через браузер. Вернее, директор библиотеки должен об этом знать, и решить можно ли это.

Также каждый пользователь должен знать о возможности установки расширений и приложений браузера. Они не являются приложениями ОС. По сути это программы. Эти расширения и приложения могут быть очень полезны. Так, они позволяют удобно переводить англоязычный текст. Для установки приложений может потребоваться регистрация в сервисах Google. Она бесплатна и проста.

Так как браузеры работают в связке с серверными программами, то работа серверных

программ отражается на работе браузера. На самом деле браузер может обращаться и к тому же устройству, на котором работает. Но для этого надо чтобы

- на этом устройстве был вэб-сервер, с которым умеет работать браузер
- браузер мог обращаться к своему устройству по адресу

Обычно эта ситуация обсуждается в теме просмотра создаваемого сайта на адресе **localhost** или **http://127.0.0.1**.

Все популярные браузеры умеют работать с вэб-серверами, и FTP-серверами. Вэб-сервер занимается выдачей произвольных файлов и страниц сайтов, а FTP-сервер выдаёт клиентской программе произвольные файлы. В чём разница? Вэб-сервер в основном выдаёт страницы для просмотра на экране. FTP-сервер выдаёт файлы для сохранения их на носителях устройства – включая встроенную память, включая жёсткий диск. Вэб-сервер предназначен, грубо говоря, для сервисов, и чтения страниц. А FTP-сервер передаёт файлы, которые *скачиваются* с него, и сохраняются на устройстве. Пока идёт процесс скачивания не выключайте браузер. Иначе полученный файл может быть дефектным.

Скачивание есть копирование файла из локальной сети или интернета. Его надо отличать от проигрывания файла – обычно это видеофайл. При проигрывании файлов файл может не сохраняться на вашем устройстве. И браузер можно выключить.

Серверные программы обычно достаточно сложны. Они могут выдавать сообщения об ошибках. Но эти ошибки – не повод наказывать пользователя. Вот часто выдаваемые ошибки, их не надо зубрить.

403 Forbidden	запрещено
404 Not Found	не найдено
408 Request Timeout	вышло время ожидания
429 Too Many Requests	слишком много запросов
500 Internal Server Error	внутренняя ошибка сервера
502 Bad Gateway	плохой шлюз
503 Service Unavailable	сервис не работает
511 Network Authentication Required	требуется аутентификация
520 Unknown Error	неизвестная ошибка
524 A Timeout Occurred	вышло время ожидания
526 Invalid SSL Certificate	ненадёжный сертификат

Сообщение **526 Invalid SSL Certificate** описывает ситуацию, когда не удаётся использовать шифрование. Это означает что данные не будут скрыты от посторонних. Вероятно, сам сервис сайта работать не будет.

Сообщение **502 Bad Gateway** обычно означает неправильное состояние сервера, а не вашего устройства или вашего браузера.

Так как FTP-сервер, примерно говоря, занят простой передачей файлов, и отображает файловую систему, то его сообщения об ошибках – это сообщения вроде «файл не найден», или «у вас нет прав просматривать каталог».

Личный кабинет, личный режим и регистрации.

Это важная тема так как личный режим – который не должен быть перепутан с приватным режимом – необходимость для получения многих услуг. Личный режим – это режим работы *сервиса*, при котором он обслуживает вас лично – зная кто вы, и проверив что это именно вы. Браузер и сайты технически могут собрать некие несекретные сведения о пользователе браузера – не зная кто это и даже не зная сколько человек пользовались браузером или сайтом вместо одного. Так, например, некий сервис может проследить на какие страницы вы заходили. Но это отслеживание не вас лично. Оно не является пригодным для тех сервисов где требуется личное обслуживание конкретного лица. Какое это имеет значение? Примерно то же что ваш паспорт. Для личного обслуживания обычно создаётся личный кабинет – то есть

набор персональных услуг и данных. Но ведь вы просто читаете статьи про то как варить малиновое варенье? Тогда вы не получите многих полезных услуг. Почему нужна идентификация вашей личности, проверка того что именно вы пользуетесь сервисом? Я – не разработчик таких сервисов. Но могу предполагать, что это – мера для избегания хаоса и ошибок. Но более важно то что это средство оказания личных услуг. Сервису мало просто знать некие формальные сведения, ему требуется конкретный пользователь. Вы же не хотите чтобы ваше наследство досталось постороннему человеку? Более того – вы не хотите чтобы кто-либо знал конфиденциальные данные о вас. Так, например, есть тайна банковского вклада. Но чтобы такие тайны остались тайнами важно правильно себя вести. А именно:

1. не допускать к компьютерному устройству посторонних, не давать им работать с вами одновременно или попеременно
2. не показывать что вы делаете – тем более при подаче вами идентификационных сведений. Тут важно чтобы любая видео или фото камера не передавала изображение ваших рук или лица
3. не разглашать идентификационные данные, не создавать ситуации когда их можно получить
4. выходить из личных кабинетов немедленно после необходимых действий там. Если произошёл обрыв работы сервиса или программы через которую вы обслуживались лично, то как можно скорее войти в личный кабинет и выйти из него
5. не создавать простые или короткие, или предсказуемые пароли. Пароли подбирают не люди, а специальные программы. Поэтому такой пароль как, например, «Ivanov1955» будет быстро подобран. Пароль «IvanovPetr1955» подобрать чуть дольше. Фамилий, имён, годов рождения реальных людей не так много. Пароль «IvanovPetr1955#» подобрать чуть дольше. Но насколько? Во столько раз сколько символов латиницы на печатной машинке. Это долго? Нет. Это – очень быстро. Также, например, подбирающая пароль программа отлично знает, что «fktrctq» – это Алексей, и так далее (те же кнопки). Она знает все слова русского языка, как и, наверное, самых редких языков.
6. не указывать адреса электронной почты, используемые кем-то ещё кроме вас, или который может быть передан другому лицу. В организациях могут быть не личные, а временно используемые вами адреса электронной почты
7. входить в свой почтовый ящик электронной почты до получения туда письма для смены пароля или подобного, стирать безвозвратно это письмо как можно скорее. Использовать это письмо как можно скорее
8. стирать безвозвратно всю секретную информацию с передаваемых носителей, как и рабочие среды (виртуальные машины и так далее) где может быть такая информация. Конечно, это не значит разрушать работу интернет-кафе – скорее не пользоваться им.
9. штатно, правильно завершать работу в личном кабинете, и с браузером (или другой подобной программой). Это значит явно выходить из личного кабинета или режима персонального обслуживания, и, после этого, дожидаться получения следующей страницы – уже не связанной с вами. Завершить личное обслуживание на всех устройствах, во всех программах
10. не позволять программам хранить даже несекретную, но личную информацию. Программы часто делают в расчёте на личное использование. Это особенно важно в библиотеках, где читатели могут быть совершенно невежественными и беспечными. В частности, нельзя нескольким людям использовать одну почтовую (клиентскую) программу. Личная информация увеличивает риск разглашения секретов

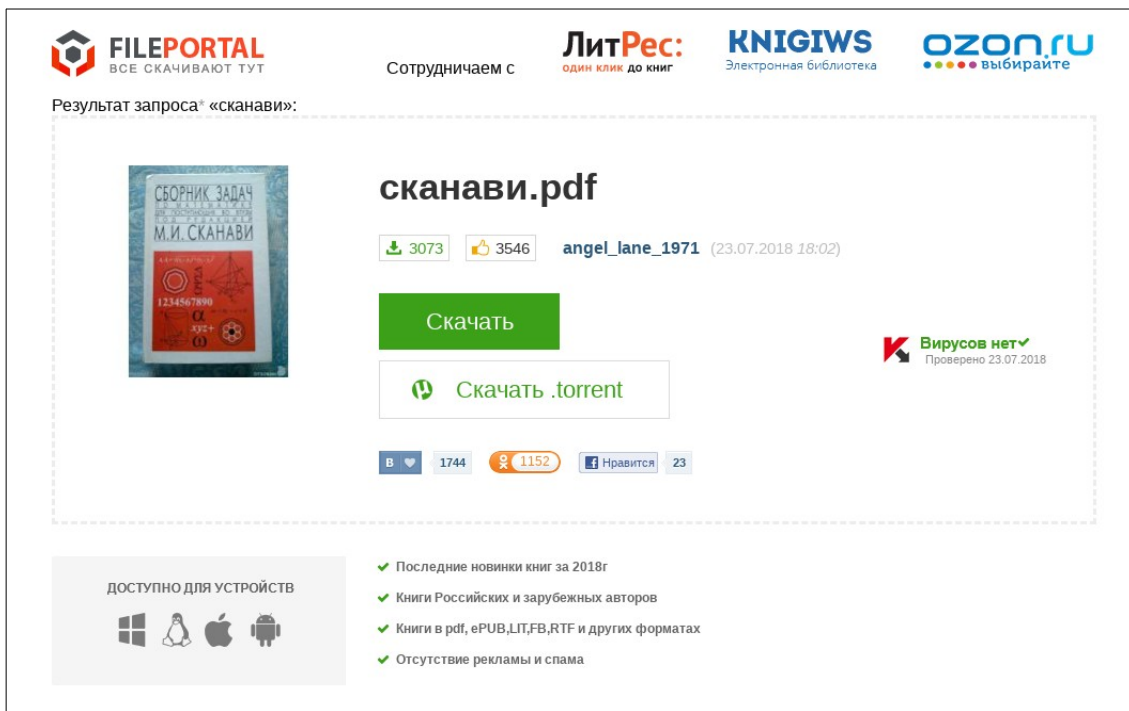
Для создания личного режима нужен вход в *аккаунт* – учётную запись. Для входа в личный кабинет нужно найти соответствующую ссылку или поле ввода на сайте сервиса. Для получения личного обслуживания нужна регистрация, или получение идентификационных данных. Обычно такими данными является логин (то есть имя пользователя, которое не связано с его паспортным именем) и пароль. В качестве логина часто используется адрес электронной почты. Если вы забыли пароль, и даже логин, то его можно пере назначить или получить в письме электронной почты.

Важно использовать свежий выпуск полноценного браузера. Консольные браузеры не годятся.

Автор не является специалистом по защите данных или безопасности.

Платное получение файлов и книг.

Книги часто распространяются как файл. Файлы разного типа и назначения могут распространяться платно. Вы платите не за содержимое файла, а за его получение. В таком случае на той же странице с которой вы получаете файл написано о стоимости услуг передачи вам файлов, и управлении этой услугой. Вот пример страницы. Верх страницы.



FILEPORTAL
ВСЕ СКАЧИВАЮТ ТУТ

Сотрудничаем с

ЛитРес:
ОДИН КЛИК ДО КНИГ

KNIWIWS
Электронная библиотека

ozon.ru
выбирайте

Результат запроса «сканави»:

сканави.pdf

3073 3546 angel_lane_1971 (23.07.2018 18:02)

Скачать

Скачать .torrent

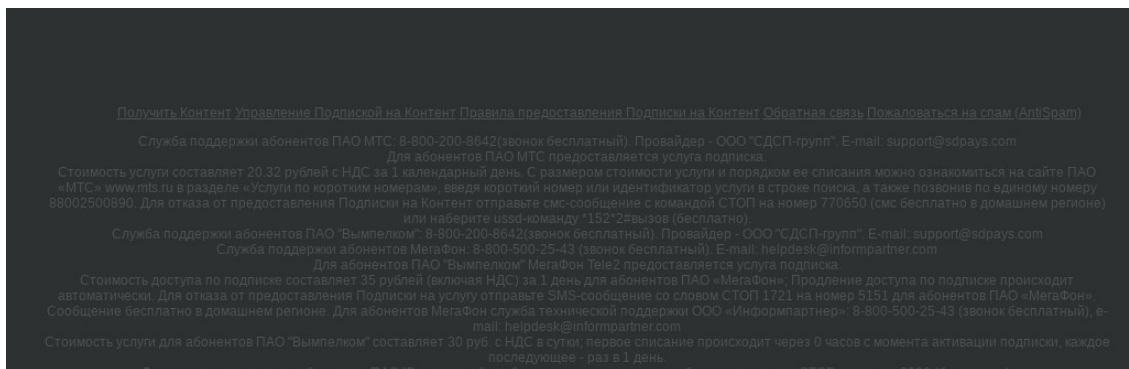
Вирусов нет
Проверено 23.07.2018

1744 1152 23

ДОСТУПНО ДЛЯ УСТРОЙСТВ

- Последние новинки книг за 2018г
- Книги Российских и зарубежных авторов
- Книги в pdf, ePUB,ЛIT,FB,RTF и других форматах
- Отсутствие рекламы и спама

Низ той же страницы.



Получить контент Управление Подпиской на контент Правила предоставления Подписки на контент Обратная связь Пожаловаться на спам (AntiSpam)

Служба поддержки абонентов ПАО МТС: 8-800-200-8642(звонок бесплатный). Провайдер - ООО "СДСП-групп". E-mail: support@sdpsys.com
Для абонентов ПАО МТС предоставляется услуга подписка.
Стоимость услуги составляет 20,32 рублей с НДС за 1 календарный день. С размером стоимости услуги и порядком ее списания можно ознакомиться на сайте ПАО «МТС» www.mts.ru в разделе «Услуги по коротким номерам», введя короткий номер или идентификатор услуги в строке поиска, а также позвонив по единому номеру 88002500890. Для отказа от предоставления Подписки на контент отправьте смс-сообщение с командой СТОП на номер 770650 (смс бесплатно в домашнем регионе) или наберите ussd-команду *152*2#звонков (бесплатно).

Служба поддержки абонентов ПАО "Вымпелком": 8-800-200-8642(звонок бесплатный). Провайдер - ООО "СДСП-групп". E-mail: support@sdpsys.com
Служба поддержки абонентов МегаФон: 8-800-500-25-43 (звонок бесплатный). E-mail: helpdesk@informpartner.com
Для абонентов ПАО "Вымпелком" МегаФон Tele2 предоставляется услуга подписка.
Стоимость доступа по подписке составляет 35 рублей (включая НДС) за 1 день для абонентов ПАО «МегаФон». Продление доступа по подписке происходит автоматически. Для отказа от предоставления Подписки на услугу отправьте SMS-сообщение со словом СТОП 1721 на номер 5151 для абонентов ПАО «МегаФон» Сообщение бесплатно в домашнем регионе. Для абонентов МегаФон служба технической поддержки ООО «Информпартнер»: 8-800-500-25-43 (звонок бесплатный), e-mail: helpdesk@informpartner.com
Стоимость услуги для абонентов ПАО "Вымпелком" составляет 30 руб. с НДС в сутки, первое списание происходит через 0 часов с момента активации подписки, каждое последующее - раз в 1 день.

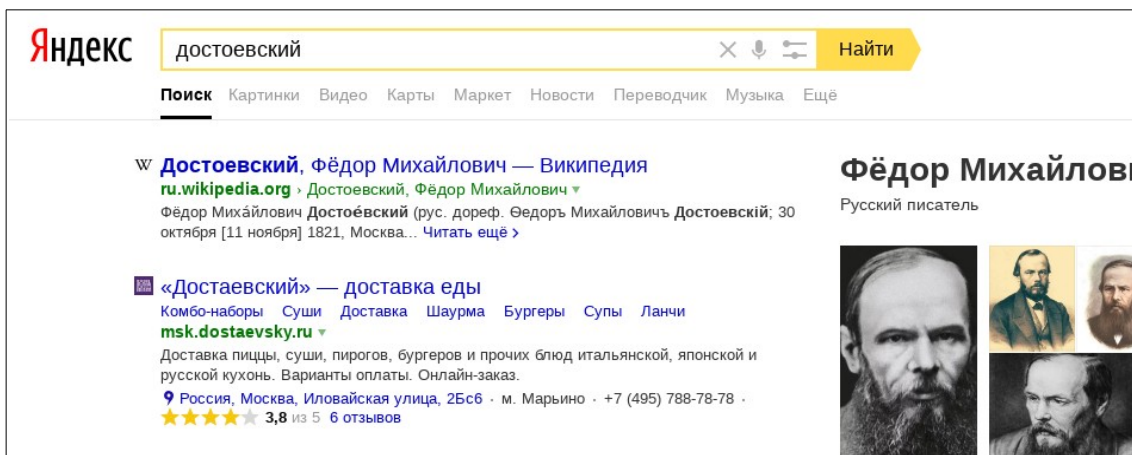
Мелким шрифтом написано о стоимости услуги. Сервис может потребовать ваш номер телефона. Учтите что таким образом вы можете разглашать его, и при незаконном получении файла вас могут наказать.

Поиск в интернете.

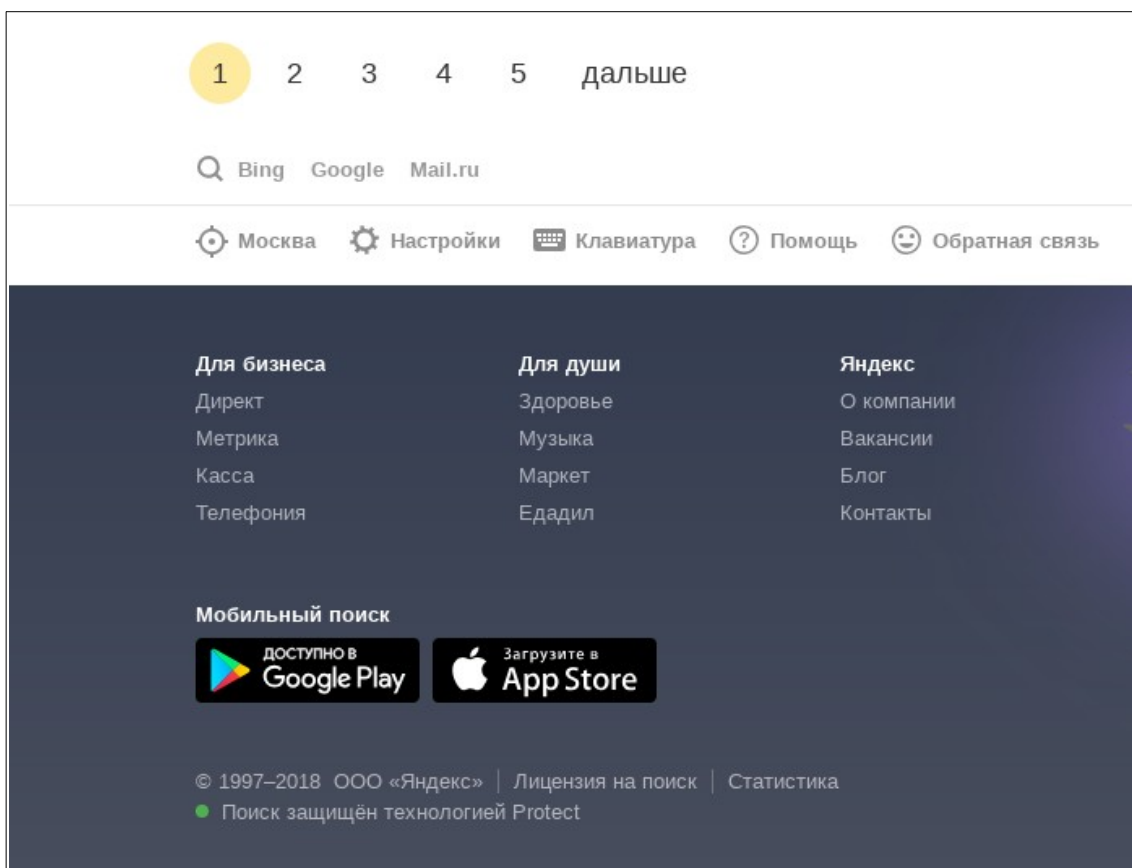
Поиск в интернет практически важен. Каждый пользователь – если он не хочет ударить себе по рукам – должен уметь искать в интернете. Конечно, он должен уметь искать информацию и на носителях. Но интернет гораздо больше чем любой носитель, и там информация обновляется.

Основным средством поиска в интернете являются поисковые машины. Что это? Я – не специалист по поиску. Поисковая машина доступна как сайт и очень похожа на сайт. Но, в отличие от сайта, она соединена с программами для поиска, и базой данных, постоянно обновляемой, и состоящей из данных интернета. Поисковая машина может иметь специализацию, и работать не буквально так как другая поисковая машина.

Поисковая машина позволяет вам ввести текстовый запрос, и выдаёт страницы с результатами поиска. Эти страницы пронумерованы, и внизу каждой страницы есть ссылки на другие страницы в виде номеров. Например.



Низ той же страницы.



Тут цифры – это ссылки на соседние страницы результатов поиска.

На каждой странице крупным шрифтом синего цвета показано название документа, и эта строка является ссылкой на этот документ. Ниже дана дополнительная информация, в частности – куски текста с искомыми словами из этого документа.

Название документа может не отражать его содержания, или быть набором бессмысленных букв и цифр.

После набора текстового запроса надо либо щёлкнуть «Найти», либо нажать **Enter**. Поисковая машина получает запрос, и по своим правилам ищет информацию. Результат поиска сильно зависит от запроса. Каждая поисковая машина может иметь свои приёмы

поиска.

В простейшем виде поиск достаточно полезен и прост. Он один уже достаточная причина для выхода в интернет. Найдя нужный документ, вы можете получить информацию по любой теме. Также вы можете найти товар, человека, журнал, фото картины, звукозапись, фильм, телепередачу, и так далее.

Наиболее популярные, и при этом развитые поисковые машины следующие.

Название	Адрес
Яндекс	ya.ru yandex.ru
Google	google.ru

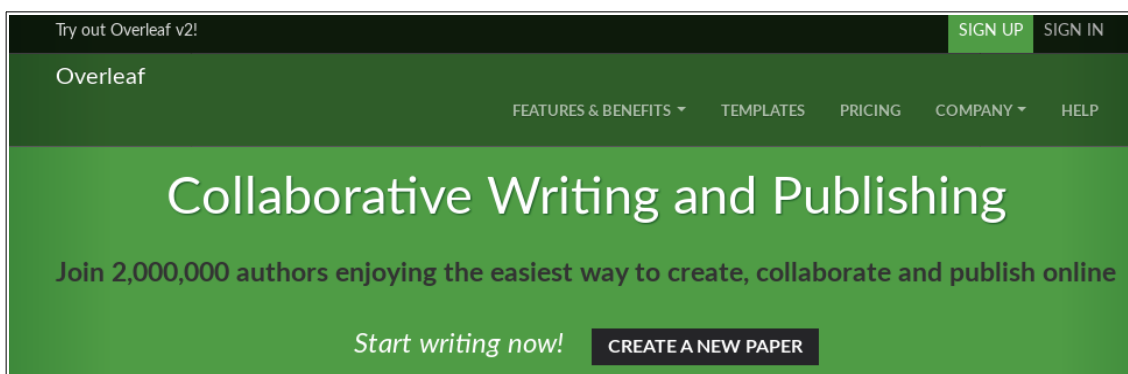
Также могут быть полезными разные рубрикаторы, например, рубрикатор поисковой машины Yahoo. Разница между рубрикатором и поиском слов в том что нужная страница может быть в интересной вам рубрике, но не содержать искомым слов.

Также стоит сказать про поиск сайта. Это поиск лишь по тем страницам, и базе данных, которые выбраны для поиска в пределах сайта. Поиск принципиально такой же как глобальный – в Яндекс и подобных машинах.

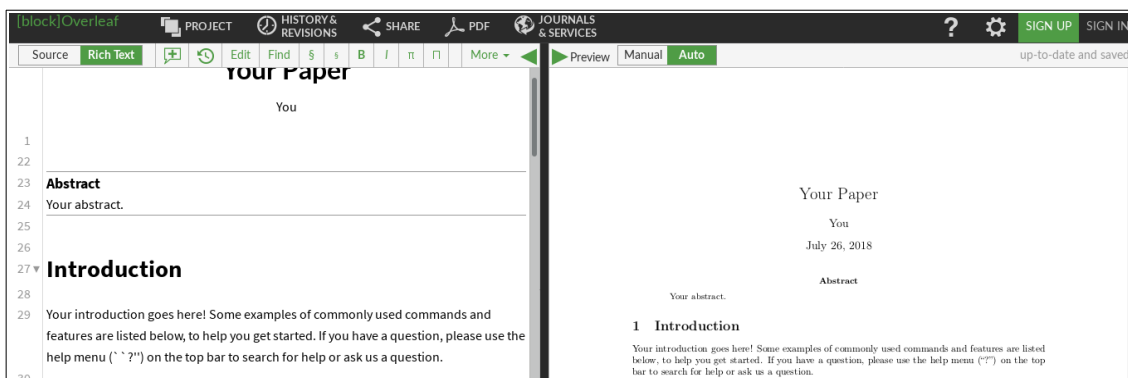
Пример сервиса – Overleaf.

Сервисы в интернет могут быть самыми разными, и предоставлять разные услуги. Для примера покажу сервис Overleaf. Он позволяет вам создавать документы с помощью Latex без лишних трат и сложностей. Но потребует достаточно высокая скорость передачи данных. Если вы хотите использовать Latex обычным образом, то вам потребует множество программ, и вам может быть сложно настроить их. Сервис позволяет достаточно просто решить все эти проблемы.

Вначале – при выходе в интернет – зайдите на сайт сервиса. Вот что вы увидите.



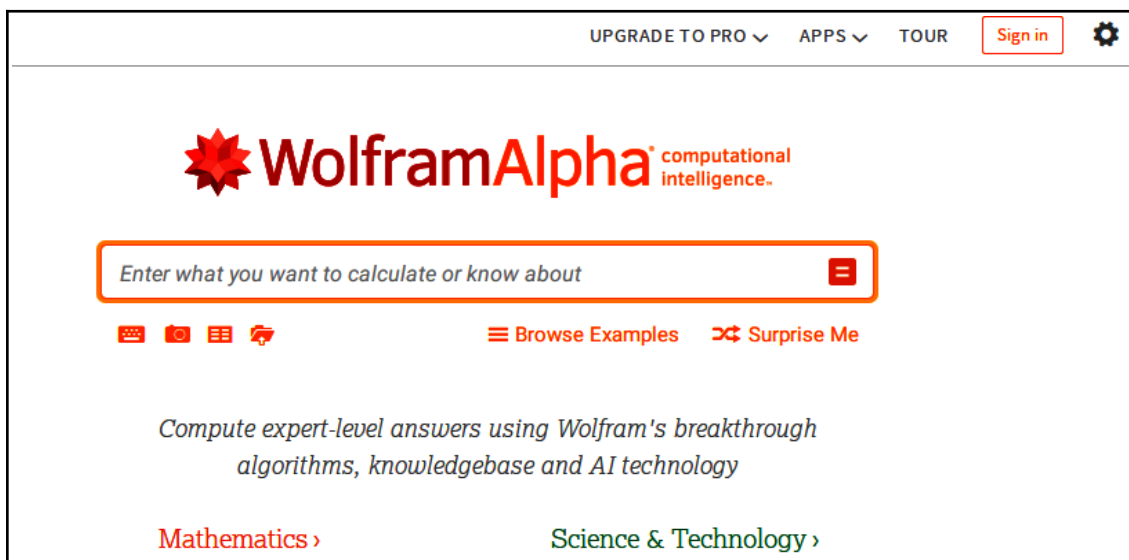
Сайт не имеет русской версии, но позволяет создавать русскоязычные документы. Можно не регистрироваться. Нажмите чёрную кнопку. Вот что вы получите.



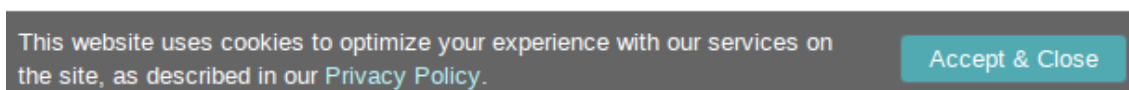
Слева вы редактируете текст, справа он отображается как готовая публикация уже прошедшая механизмы Latex.

Пример сервиса – Wolfram Alpha.

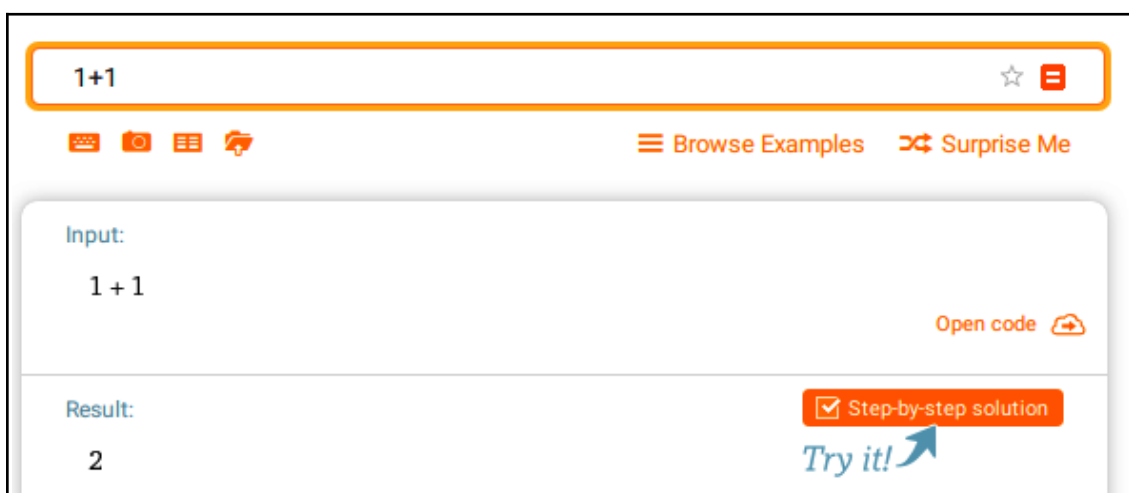
Это сервис, примерно говоря, математической обработки данных. Автор не является математиком, и не имеет существенного опыта использования сервиса. Начальная страница сервиса выглядит так.



Внизу страницы может быть сообщение о том что сайт использует «куки».



Основным инструментом является строка текстового ввода, или поле текстового ввода на первой иллюстрации. Туда можно забить некое – в том числе «классическое» математическое выражение, и получить результат его обработки сервисом. Например.



Сервис можно использовать делая ошибки, или намеренно проверяя его возможности. Он готов к таким ситуациям.

Покупки в интернет. Использование банковских карт.

При коммунизме товары будут заказывать по

телефону, а выдавать по телевизору.

Покупки в интернет – это не перекачивание молока и варенья по проводам, а оформление заказа на товары, с определением способа оплаты и способа получения товара. Какой нам интерес покупать в интернете?

- Это позволяет не ходить в магазин.
- Вы можете прочесть про товар более-менее подробное описание.
- Вы можете быстро найти товар, или узнать что его нет.
- Часто вы можете добиться оповещения о появлении товара в интернет-магазине.
- Иногда это дешевле.
- Вам доставляют товар по официальной цене, и, если надо – с помощью фургона.
- Вы можете покупать сразу много товара, или крупные, тяжёлые вещи.
- Вы можете позвонить в магазин, и вам ответят.

Такие покупки особенно интересны за городом, для больных или пожилых покупателей. Обычно доставка товара платная, зато вы сразу получаете много покупок.

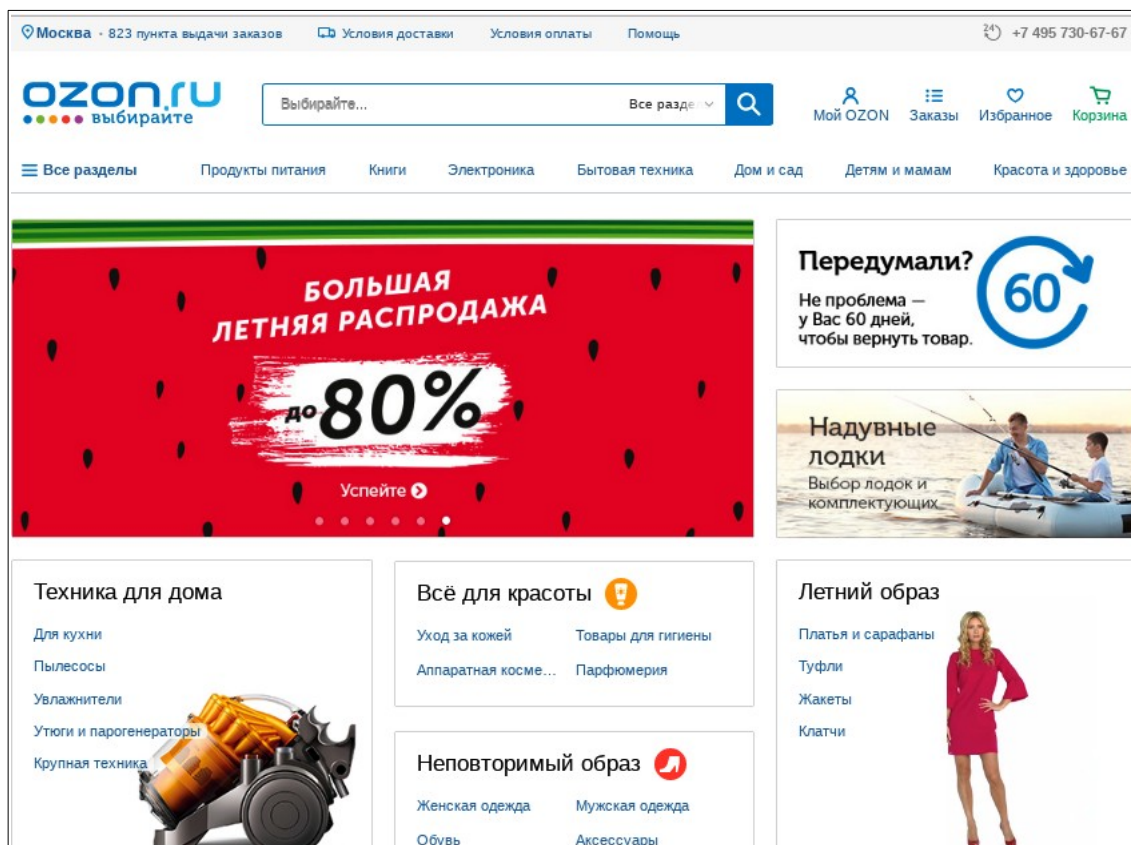
Покупка распадается на несколько этапов.

- Вы ищете в интернет товар или магазин.
- В магазине вы бросаете товар в корзину. Это не реальная корзина, а просто список товаров. Затем находите и бросаете другой товар, и так далее. Вы видите цену и фото товара. Вы определяете количество килограмм или упаковок.
- Вы смотрите содержимое корзины. Если надо, меняете его.
- Вы оформляете заказ. Вам потребуется указать место где вы заберёте заказ, время получения заказа, ваш номер телефона, и способ оплаты. Вашему заказу присваивается номер. Запишите его.
- Вы оплачиваете заказ если так определено вами, и магазин так может работать.
- Вам звонит курьер или водитель магазина – при доставке заказа.
- Вам доставляют заказ, или вы его вывозите сами из магазина или пункта выдачи заказов – ПВЗ.

Самый понятный способ покупок – с доставкой на дом, и оплатой деньгами, наличными. Тем не менее стоит иметь в виду и более дешёвую доставку в пункты выдачи товара, или автоматы. Все вопросы о работе магазина описаны на его сайте, или вы можете задать свой вопрос по телефону. Все крупные магазины имеют бесплатный для вас номер телефона для покупателей.

В Москве сотни магазинов торгуют через интернет, и они популярны. Люди соглашаются платить за доставку вместо того чтобы идти в магазин после работы. Купить можно практически всё что хотите. Иногда можно отложить товар, и приехать за ним в магазин – вернее, на склад. Всегда подробно узнайте как это сделать. Вам надо конкретно знать как проехать и пройти, и что нужно для получения товара. Иногда товар поставляется почтой. В стране работают транспортные компании, и их доставка может быть единственным вариантом в глуши. Оформление заказа при этом не намного сложнее. Вам не важно где территориально находится магазин. Пик популярности магазина китайских товаров Aliexpress, может быть, уже прошёл. Это пример магазина, торгующего лишь при предоплате, и посылающего товар по почте.

Пример сайта интернет-магазина Ozon.



Добро пожаловать в 21-й век!

Магазины могут устраивать акции и распродажи, например, в Чёрную пятницу. Отложите свои дорогие покупки до акции.

ВАЖНО – используйте новые версии программ магазина и банка, свежий браузер, не экономьте на обновлениях ОС.

Банковские карты надо отличать от всего что на них похоже, но не является картой. Оплата фактически происходит со счёта в банке который выпустил карту. При получении карты прочтите текст об использовании её. При утере, краже, застревании карты в банкомате не надо паниковать. При утере и краже карты вы должны немедленно сообщить в банк. Он заблокирует карту. Большинство банков страхуют вклады.

Карты защищены пин-кодом, то есть набором цифр, которые надо помнить. Записывать их нежелательно. Такие программы как Сбербанк-онлайн позволяют сменить пин-код. Для этого не обязательно ходить в отделение банка.

Однако для оплаты при покупках в интернет-магазинах пин-код не нужен. Нужна возможность оплаты, которую надо заранее обсуждать при оформлении карты. Можно оплачивать разными картами в одном магазине. И одной картой во многих магазинах.

Типично что деньгами можно управлять с компьютерного устройства, но автор предпочитает в качестве его использовать компьютер с Linux, а не Android. Походы в банк становятся всё более бессмысленными.

Карта выглядит примерно так.



В данном случае она считается картой платёжной системы VISA. Оплата при использовании браузера при таком виде страницы в оконной среде.

Оплата заказа №11641

С банковской карты

Выберите платёжную систему:

Номер карты:

Месяц и год окончания действия карты:

VALID THRU 01 / 13

Имя и фамилия держателя карты:

Если Вы являетесь владельцем неименной карты, пожалуйста, введите Ваши Фамилию и Имя в произвольной форме латинскими буквами.

CVV2

Данный сайт поддерживает 128-битное шифрование. Информация, введённая на этой странице, защищена специальными средствами.


Описание платежа: Оплата заказа №11641

Валюта: RUB

Сумма: 30781.00

CVV – это номер на обратной стороне карты. После ввода данных браузер автоматически откроет страницу операций банка карты, например так.



Сбербанк

Verified by
VISA

Введите Ваш пароль

Магазин: YM*DROP-DREAM
Описание:
Сумма: **52.80 RUB**
Дата: **02/07/2018**
Номер карты: **** *
Личное приветствие: **None**

Одноразовый пароль был направлен на Ваш номер телефона. Пожалуйста, проверьте реквизиты транзакции и введите пароль из SMS.

[Не получили одноразовый пароль по SMS?](#)

ОТПРАВИТЬ

На ваш телефон послано SMS-сообщение с кодом подтверждения платежа. Введите его, и нажмите кнопку **ОТПРАВИТЬ**. На ваш телефон может быть отправлено сообщение об оплате. Такое обслуживание называется интернет-банкинг. Оно организуется для каждого клиента банка по договоренности с банком, в соответствии с документами.

Всегда записывайте номер заказа. Он может быть нужен при общении с магазином.

Госуслуги.

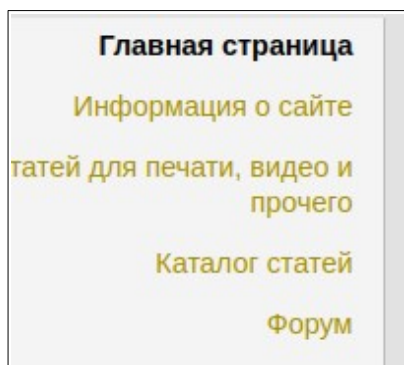
Госуслуги – это попытка дать новые услуги, или старые в новом виде, для нового качества жизни, для большего удобства и получения услуг без очередей, поездок и так далее. Список госуслуг будет расти. На сегодня госуслуги используются не менее чем 64 миллионом человек. Госуслуги на сегодня выполняются в следующих областях.

- Семья и дети
- Паспорта, регистрации, визы
- Транспорт и вождение
- Образование
- Налоги и финансы
- Моё здоровье
- Пенсия, пособия и льготы
- Лицензии, справки, аккредитации
- Квартира, строительство и земля
- Безопасность и правопорядок
- Работа и занятость
- Культура, досуг, спорт
- Бизнес, предпринимательство, НКО
- Производство и торговля
- Информация, связь и реклама
- Природопользование и экология

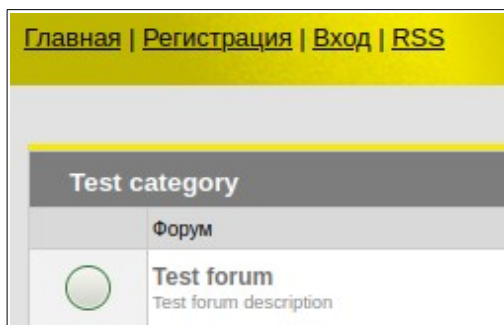
Если вы хотите получить помощь консультанта или автоответчика по вопросам госуслуг, то для вас есть бесплатный телефонный номер 8 800 100-70-10. Но помните, что основная идея госуслуг – в автоматизированном обслуживании, а не с помощью людей. Не злоупотребляйте этим номером.

Как пользоваться форумом (на примере rkrprovincia.ucoz.ru)

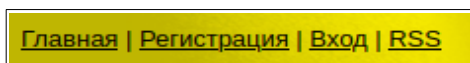
- Зайдите на сайт с форумом
- щёлкните ссылку на форум



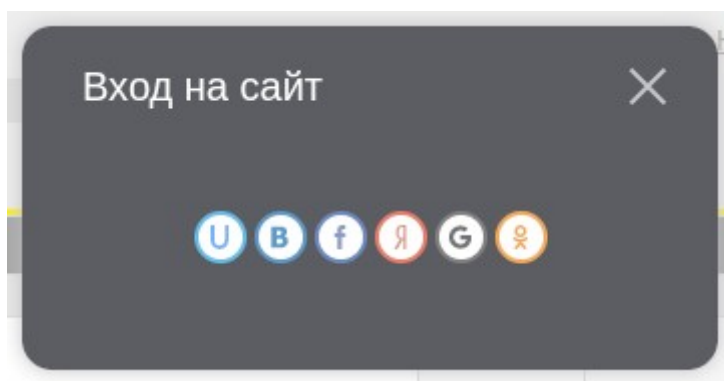
- щёлкните ссылку на тестовый форум (так как другого нет)



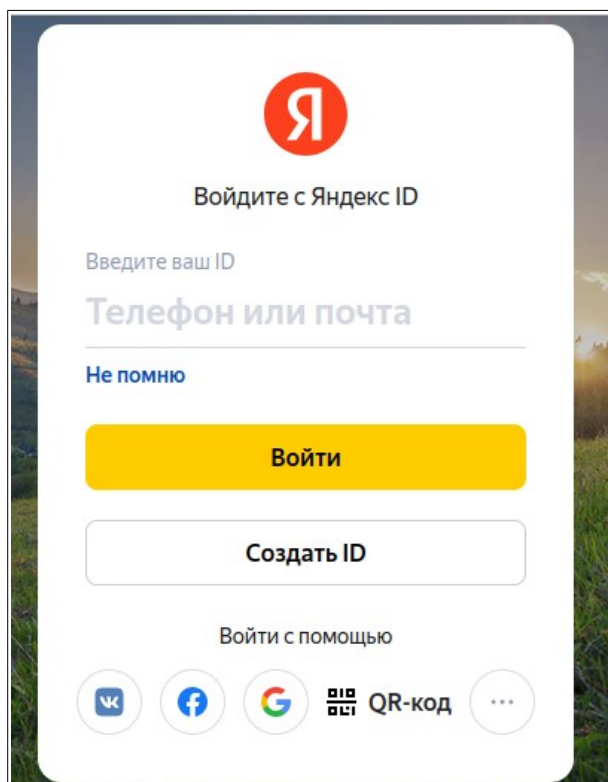
- теперь вы можете читать форум или войти и писать, и читать. Щёлкните Вход.



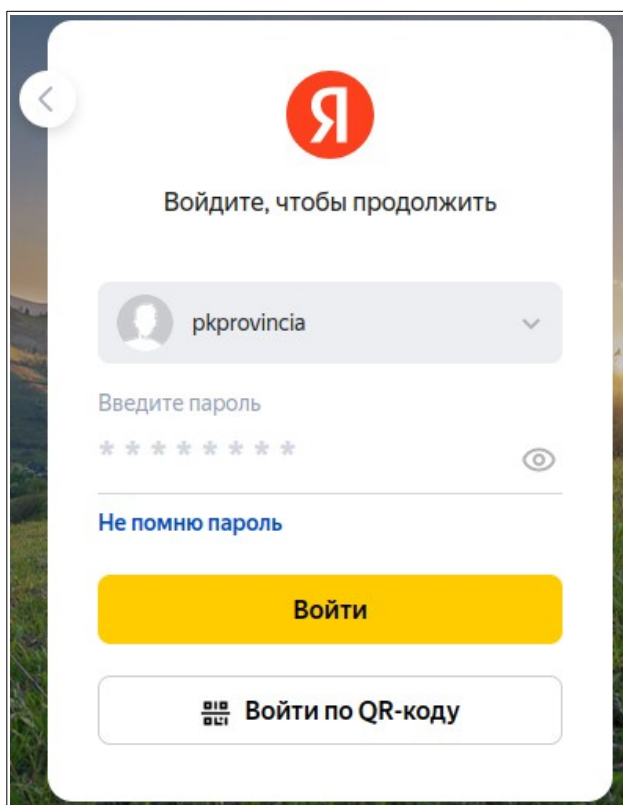
- Щёлкните значок где вы имеете регистрацию.



Пусть это будет регистрация Яндекса. вы получите окно для ввода логина.




Введите логин там где написано «телефон или почта». Нажмите кнопку Войти.



Введите пароль, и нажмите Войти. Поставьте галочку для согласия с политикой сайта, и щёлкните чтобы обозначить что вы не робот.

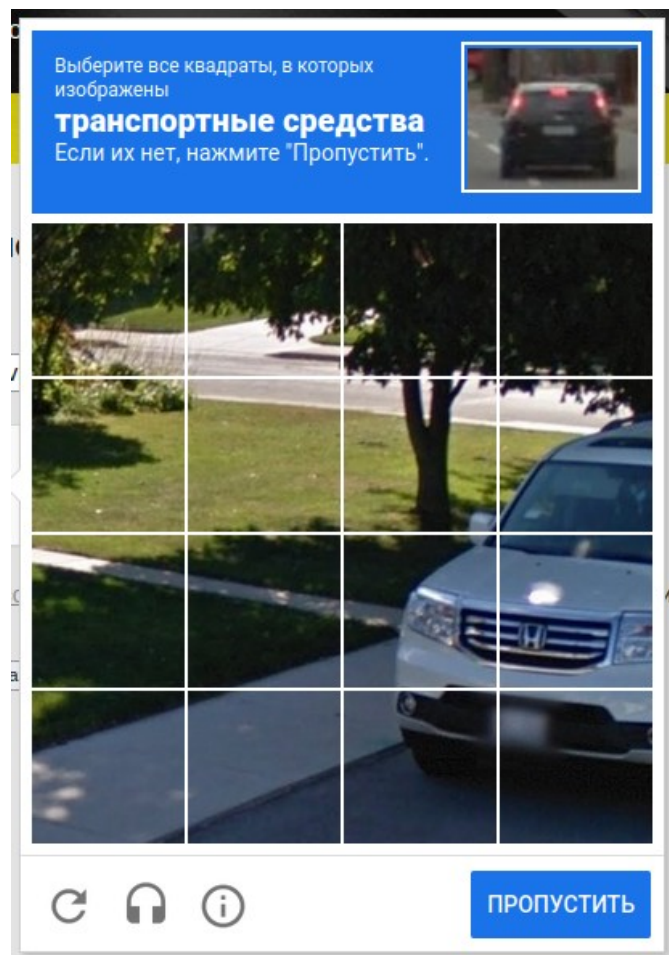
Регистрация

Email

Я не робот  geCAPTCHA
Конфиденциальность - Условия использования

Я согласен на обработку персональных да

Сайт потребует доказать что вы не робот – выбрать щелчками несколько картинок, или одну, или ни одной.



Нажмите Подтвердить. Сайт создаст галочку в строке о роботе. Сейчас меня сайт не пускает. Удалось войти с регистрацией Facebook. Вот что вы увидите.

[Главная](#) | [Мой профиль](#) | [Выход](#) | [RSS](#)

Теперь вы можете читать форум, но и писать в него – в том числе создавая тему обсуждения. Для этого щёлкните **New Topic**.



Форум » Test category » Test forum » Новая тема

Создание новой темы

Название темы:	<input type="text"/>
Описание темы:	<input type="text"/>
Текст сообщения:	<div style="border: 1px solid #ccc; padding: 5px;"><p>b <i>i</i> <u>u</u> SIZE FAMILY COLOR quote code spoiler hide list ...</p><div style="float: left; width: 100px; border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"><p>Все смайлы</p></div> <div style="clear: both;"></div></div>
Опции сообщения:	<input checked="" type="checkbox"/> Включить смайлы <input checked="" type="checkbox"/> Включить подпись
Код безопасности:	<input type="checkbox"/> Я не робот <small>Конфиденциальность - Условия использования</small>

Тут в наиболее крупном окне пишется содержимое вашей темы, ваш вопрос или что-то подобное. Заполните поля названия и описания темы. Нажмите Создать тему.

Примерно так же пишутся ответы и вопросы в теме.

Документация и описание браузеров.

Браузеры, будучи очень часто используемыми программами, как бы стали жертвами своей популярности. Такую сложную программу, да ещё настраиваемую, и связанную с множеством технологий и сайтов трудно описывать. Автор знает лишь два браузера с более-менее хорошим описанием – Firefox и Sea Monkey. Другие – в том числе используемые в телефонах – могут быть вовсе не описаны. Думаю, что их авторы понадеялись на догадливость пользователя, и его опыт. Но проблема в том что этот опыт не у всех есть, и не у всех есть компьютер для использования «классических» браузеров вроде Safari или Internet Explorer. Даже старательно написанное руководство по браузеру будет включать терминологию, которая касается браузера, но по сути уводит в сторону. Это, например, взаимодействие браузера с серверами, или JavaScript, или расширения браузеров. Таким образом, граница желаемого текста уходит всё дальше, и при этом всё ещё остаются вопросы.

Автор сейчас не видит никакой реальной альтернативы браузерам. Конечно, именно для получения неких текстов можно использовать что-то вроде читателей лент новостей. Но самая «соль» и смысл использования браузера не в этом, а в *услугах*, которые в нынешнее время требуют интерактивности. Такая интерактивность есть в программах, но она узко направлена на некоторые функции, и такие программы, примерно говоря, есть лишь для операционной системы Android. Для, например, Linux таких программ нет. И не ожидается так как эта система имеет в качестве аудитории людей уже понимающих браузеры.

Конкретнее об использовании консольного браузера.

Конкретное обсуждение браузера делается описывая либо нечто общее в работе с браузерами, либо конкретный браузер. Второй способ более полезен, и легче для автора и читателя. Каждый пользователь должен понимать что браузеры меняются – и внешне, и в своих функциях, хотя не сильно – по мере выхода их новых версий. Наиболее популярным в среде бесплатных ОС является браузер Firefox, он же есть для ОС Android, но, прежде чем его описывать, надо описать проблемный вопрос о браузере на микрокомпьютере.

Если нужна работа в консоли, то лучшим браузером, наверное, является **elinks**. Если он недосыаем, то используется браузер **Links**. Если у вас на микрокомпьютере установлена урезанная версия ОС, то там может не быть никакого браузера. Его потребуется ставить с помощью пакетного менеджера или просто установкой отдельного файла, который, допустим, вы получили в интернет-кафе. Но надо заметить, что для полноценной установки может потребоваться более одного файла. Интернет и наличные программы – в том числе пакетный менеджер в UNIX вам в помощь. На ОС, родственных Debian, для установки **elinks** надо от имени администратора подать команду

```
apt-get install elinks
```

Вопрос о том откуда файл придёт к вам имеет разные ответы смотря по тому как настроен пакетный менеджер. Наиболее обсуждаемый случай – когда он берёт файлы из интернета.

После установки надо не от имени администратора запустить браузер командой

```
elinks
```

Браузер в телефоне.

Вам прежде всего нужно уметь пользоваться браузером в телефоне так как именно это позволяет понять как работать с планшетом или компьютером. Также это потребуется если планшет или компьютер не способен работать взамен телефона с браузером.

Тут мы приходим к сложной теме так как телефоны разные. Эпоха просто телефонов – не смартфонов – в большой степени закончилась в городах и достаточно богатых пригородах вроде Подмосковья. При этом в более глухих местах всё ещё могут массово использоваться старые или более примитивные модели – в том числе с чувствительным экраном. Чувствительный экран сам по себе ещё не означает смартфон. Поскольку это пока так, то все телефоны можно разделить на три группы.

- Смартфоны – обычно использующие ОС Android.
- Другие телефоны, умеющие выходить в интернет.
- Другие телефоны без возможности выхода в интернет.

ВАЖНО – телефоны без выхода в интернет иногда могут работать в локальной сети Wi-Fi. В таком случае не спешите выбрасывать телефон. Он не слишком примитивен, но требует такой сети, которая сама имеет выход в интернет. Для звонков и получения SMS выход в интернет не нужен.

Есть ещё один важный момент – телефон может иметь выход в интернет, но не обеспечивать достаточно высокую скорость приёма данных оттуда, причём это его техническое ограничение. Если телефон не умеет работать с 3G, то скорость не может быть высокой даже если сама сеть связи её позволяет. Реальная скорость может быть любой, но не выше предела возможностей телефона.

Так как мы говорим о браузерах, то неизбежно деление телефонов на три вида.

- Телефоны, на которых есть браузер, и его можно сменить или добавить другой браузер.
- Телефоны, где теоретически можно поставить браузер, но не обязательно он установлен изначально.
- Телефоны, где браузера нет, и куда его нельзя поставить.

Первая и третья группа малоинтересны. Интересна вторая – рискованная группа. Она по сути связана в большой степени с возможностью установки программ в телефон, и – ВАЖНО – работы этих программ. Почему я так пишу? Например, телефон LG KF300 – прекрасное средство для болтовни и набора цифр руками, но он крайне плохо умеет запускать поставленные на него программы. С другой стороны, телефон без претензий на шик NOKIA 310 не имел таких проблем в моём опыте, и стоил дешевле. И тот, и другой телефон умеет работать с интернет напрямую, и относится ко второй группе. Сейчас явно смартфоны стремятся захватить рынок телефонов, оставив пенсионерам скромную часть рынка кнопочных, простых моделей. 10 лет назад в продаже можно было легко найти телефон с поддержкой JAVA-программ, которых написано много, и, главное – это были недорогие телефоны на которые теоретически можно было поставить браузер. Сейчас это, например, некоторые модели производства Sony Ericsson. Я пишу о JAVA-программах в основном потому что среди них есть несколько развитых браузеров. Но вот относительно их пригодности для финансовых и подобных операций у меня есть сомнения. Если вы хотите иметь нечто дешёвое, но позволяющее эти операции, то, возможно, стоит купить микрокомпьютер.

Тем не менее работающий браузер в телефоне очень важен так как это – реальная помощь

при проблемах и сомнениях.

Браузер в телефоне может не быть JAVA-программой. Он может быть просто программой, поставленной на заводе, и работающей в конкретной модели телефона. Важно, что в последнем случае вы вряд ли сможете обновить браузер. Обновление для браузеров очень важно так как иначе браузер может стать «дырой» в программной среде телефона. Это – ещё один аргумент против простых телефонов. Но – заметьте – Android на сегодня – самая уязвимая ОС.

Реальный недостаток старых и малоразмерных телефонов – малый экран. Он может сильно портить жизнь – тем более когда просматриваемый текст не умещается на экране.

На одном телефоне можно иметь более одного браузера. Пока мы везде обсуждали HTML-браузер, а не WAP-браузер. Последний не очень полезен по сравнению с первым, и интересен лишь для версии WAP2. Такой браузер умеет показывать сайты именно для WAP2, но их количество может быть невелико, и сокращаться со временем.

Оконные браузеры.

Это оконная программа, и она требует оконного сервера, а не фрэймбуфер.

Для читателей этой книги умение пользоваться браузером – ключевое. Браузер на сегодня – замена многим программам в среде Android, и, наверное, самая популярная непрофессиональная программа на всех развитых устройствах. Эта глава есть конкретное описание браузеров. Она направлена на то, чтобы вы, читатель, сами могли узнать с помощью браузера как можно больше, продолжили наращивать своё знание и понимание вопросов информатики.

Браузер с точки зрения ОС есть обычная программа, напрямую связанная с сетями – на практике прежде всего с интернетом. Так как браузер использует оперативную память, и вычислительную мощность, то следует рассматривать его в связи с этими ресурсами. Это – аппаратные, технические ресурсы. Популярные, полноценные браузеры не могут работать на слабых устройствах. Там приходится применять более легковесные, более быстрые браузеры. Автору не очень понятно какой браузер можно использовать на роутере. Но это, можно сказать, нерентабельно – учитывая его конкурентов в плане цены и возможностей. На микрокомпьютере Raspberry Pi B, как пишут, медленно работает браузер Midori. На более мощных микрокомпьютерах можно использовать браузеры Vivaldi и Chromium. Все эти браузеры требуют оконной среды.

На слабых микрокомпьютерах, или в условиях малых ресурсов на сервисе аренды вычислительной мощности, а также при использовании Linux в среде Android пользователь может быть вынужден обходиться без оконной среды. Без оконной среды могут работать браузеры Links, Elinks. Также можно попробовать браузер Arachne, но он вообще-то создан для ОС DOS. Придётся задействовать эмулятор для DOS. Версия браузера для Linux вышла в 2010 году. Она могла устареть. Повторю – все эти браузеры могут быть несовершенными. Последние технологии могут не работать. ВАЖНО – ваша деятельность в любом сервисе может быть нарушена, и представлять угрозу для ценных данных. Я категорически не рекомендую использовать их при операциях с секретными данными или деньгами.

На компьютерах ситуация намного лучше. Даже на технике 10-летней давности производства оконная среда не мешает использовать браузер.

Расход обозначенных ресурсов зависит от использования браузера. Например, проигрывание видео, как и большое количество одновременно открытых страниц, то есть полученных по сети документов, может существенно замедлить работу браузера на старой технике.

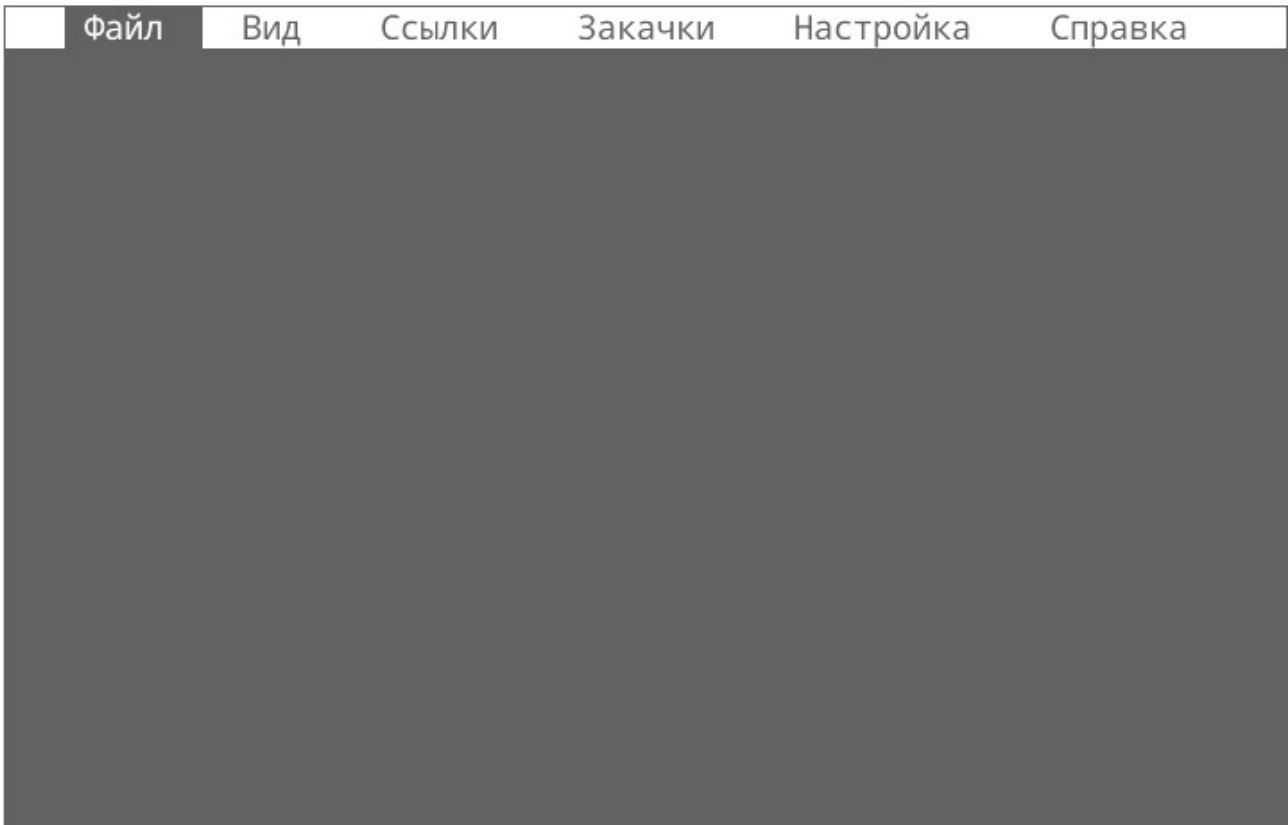
Links.

Браузер работает на многих ОС. В том числе на Linux, и, видимо, на FreeBSD. Очень вероятно, что его можно поставить из официального репозитория. Браузер умеет работать в псевдографическом режиме, но при наличии фрэймбуфера. В таком случае он может показывать иллюстрации. Рассмотрим работу браузера в текстовом режиме, на UNIX. Там он работает в консоли, или терминальной программе. Там для старта браузера подайте команду

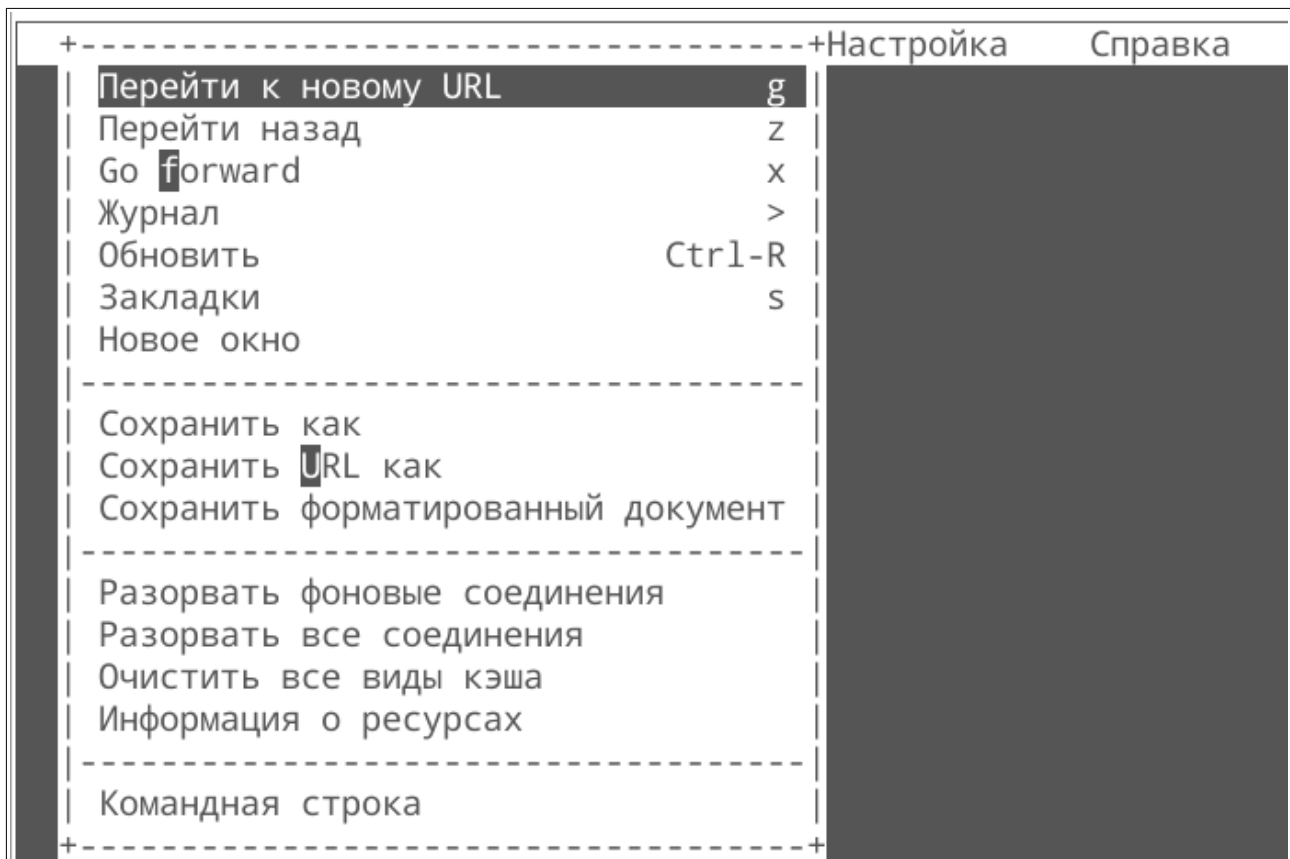
`links`

В результате вы получите либо пустое окно или экран, либо сообщение о работе браузера. На моём компьютере, в терминальной программе, результат – чёрное окно.

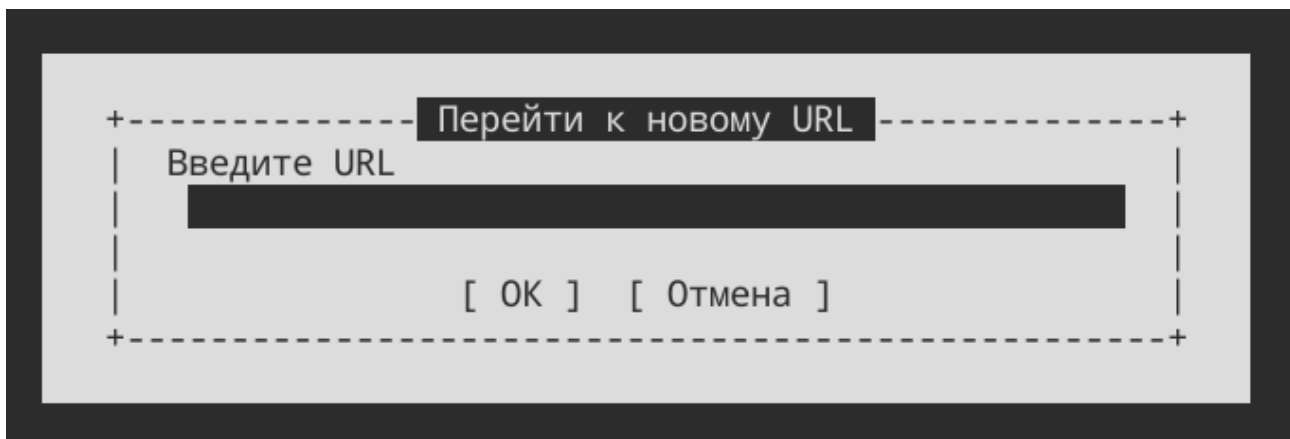
Это – не программная ошибка. Для управления браузером есть меню, но оно не видно. Чтобы его показать нажмите **Escape** (Esc). Результат.



Русскоязычное меню не гарантировано так как может зависеть от настроек ОС. Иначе оно англоязычное, но работает так же. Работа меню обычная, управление меню – обычное. Например, если мы нажмём кнопку **вниз**, то получим.



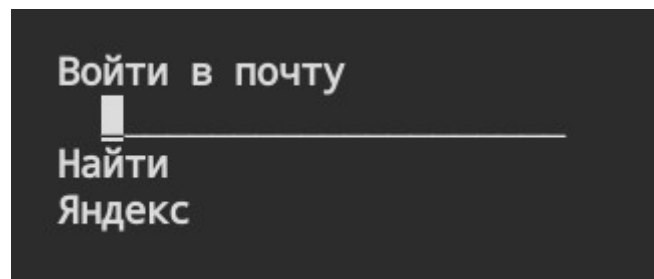
Теперь достаточно нажать **Enter** чтобы заставить работать выбранную строку меню. Перейти к URL – это перейти к новому адресу.



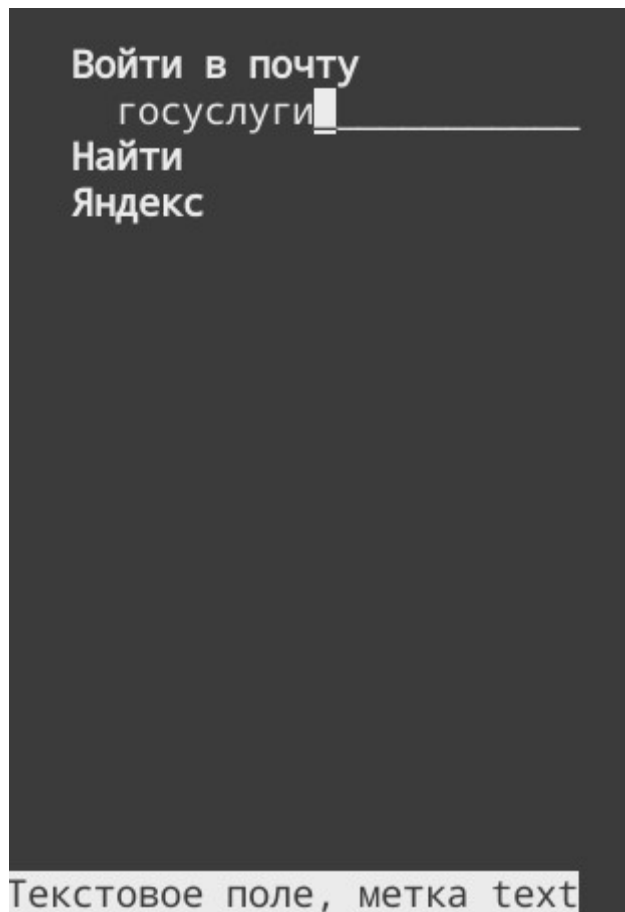
Обычно наиболее практично выйти на быстро получаемую страницу поисковой системы Яндекс ya.ru. Введите этот адрес, и нажмите **Enter**.



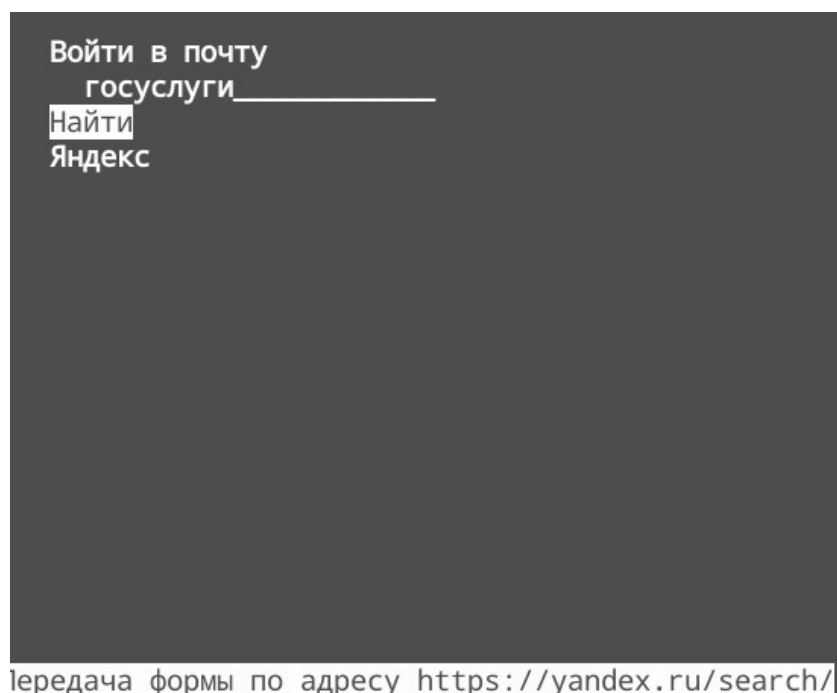
Пока выделена белым цветом ссылка её адрес виден внизу. Чтобы перейти к поисковой строке, обозначенной чертой, нажмите стрелку вниз. Эта клавиша как бы перемещает курсор мыши по ссылкам и по элементам управления на странице.



Вы поставили консольный текстовый курсор в виде «кирпича» в однострочный реактор, который на этой странице ждёт ввода текстового запроса для поиска. Для примера введём текстовый запрос «госуслуги». Прежде чем вводить на русском может быть надо настроить ОС. Но это в консоли может потребовать выключить браузер.



Вы видите что строка внизу может помочь понять где курсор. Теперь нажатие кнопки **Найти** стартует запрос для поиска. Также вы можете редактировать текст запроса простейшими методами работы с текстом. Длина строки запроса может быть больше чем длина однострочного редактора. Нажмите стрелку вниз для перемещения к кнопке **Найти**, она будет окрашена.



Нажатие **Enter** нажмёт кнопку, и вы получите результаты поиска. Они, в принципе, те же что в развитом браузере, но показаны иначе.



Также нажатие **Enter** приводит к переходу по выделенной цветом ссылке. Листание выполняется нажатием **PgUp**, **PgDown**.

Firefox. Установка браузера с запросом установочного файла с помощью Links.

Мне кажется, что этот браузер – самый популярный в бесплатных системах. Вы можете увидеть текст, сообщающий, что Firefox в Debian называется **Iceweasel**. Последним автор мало пользовался. Не уверен что это буквально так. Вы можете поставить и использовать одновременно более одного браузера.

Даже имея оконную среду, пользователь может иметь лишь Links. Если в нём ссылка ведёт к файлу, который должен быть скачан, то вы получите запрос на сохранение. Если просто согласиться на сохранение, то файл будет в вашем личном каталоге.

Прежде чем получать файл, вы, очень вероятно, зайдёте – через поисковую систему – на страницу, где на самом деле есть таблица. Она в Links показана так.

Загрузить Firefox – Бесплатный веб-

```
Link: canonical
Link: alternate x-default
Link: alternate an: aragonés
Link: alternate ar: العربية
Link: alternate az: Azərbaycanca
Link: alternate be: Беларуская
Link: alternate bg: Български
Link: alternate bs: Bosanski
Link: alternate ca: Català
Link: alternate cs: Čeština
Link: alternate cy: Cymraeg
Link: alternate da: Dansk
Link: alternate de: Deutsch
Link: alternate el: Ελληνικά
Link: alternate en-CA: English (Canadian)
Link: alternate en-GB: English (British)
Link: alternate en: English
Link: alternate en-US: English (USA)
https://www.mozilla.org/ru/firefox/new/
```

Таблица содержит ссылки на разные файлы для установки браузера Firefox, собранного для разных языков пользователя. Нажимая стрелку вниз, дойдём до русской сборки.

```
Link: alternate pl: Polski
Link: alternate pt-BR: Português
Link: alternate pt: Português
Link: alternate pt-PT: Português
Link: alternate rm: rumantsch
Link: alternate ro: Română
Link: alternate ru: Русский
Link: alternate sk: slovenčina
Link: alternate sl: Slovenščina
Link: alternate sq: Shqip
Link: alternate sr: Српски
Link: alternate sv: Svenska
Link: alternate sv-SE: Svenska
Link: alternate ta: ௦௧௨௩௪௫௬௭௮௯
AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz
```

Каракули – это не дефект браузера или ОС, а неспособность показать иероглифы – видимо, нет шрифтов. Так как я пользуюсь ОС, настроенной для России. Переходя по ссылке, вы получите то же. Но, двигаясь вниз, вы увидите вот что.

```
Загрузить Firefox – Бес
Быстрый и Ответственный.

Загрузить Firefox – Русский

* Windows 64-bit
* Windows 32-bit
* macOS
* Linux 64-bit
* Linux 32-bit
* Android
* iOS
```

Выберите свою версию в зависимости от того какая у вас ОС – 64-х, или 32-х разрядная.

```
Загрузить Firefox – Русский
*
*
*
*
*
*
*
*
+----- Неизвестный тип -----+
|
|      Тип документа application/x-tar.
|      Желаете файл сохранить или отобразить на
|      экране?
|
| [ Сохранить ] [ Отобразить ] [ Отмена ]
|
+-----+
Ваш
вы можете попробовать одну из этих версий:
```

Тут достаточно нажать **Enter**, и этим нажать Сохранить.

```
+----- Закачка -----+
|
| Сохранить в файл
| firefox-61.0.1.tar.bz2
|
| [ OK ] [ Отмена ]
|
+-----+
```

Тут можно назначить место хранения и имя сохраняемого файла. Можно просто нажать **Enter**.

Сколько времени требуется для получения файла заранее нельзя знать. На медленном канале может потребоваться более часа. В это время не выключайте Links. Чтобы узнать состояние копирования файла из сети используйте пункт меню Links **Закачки**.

Именно в случае Firefox не нужна классическая установка с помощью менеджера пакетов. Показанные действия, разумеется, можно выполнить в браузере, являющимся оконной программой.

Полученный файл подлежит разархивации. В результате вы получите каталог с именем **firefox**. В нём есть файл с именем **firefox**. В отдельных случаях может потребоваться выставить для этого файла права для запуска. После этого стартовать Firefox совсем просто. К этому моменту уже должна работать оконная среда. В терминальной программе подайте следующую команду – исходя из того что каталог браузера находится в известном читателю месте – личном каталоге пользователя seloZuzukino:

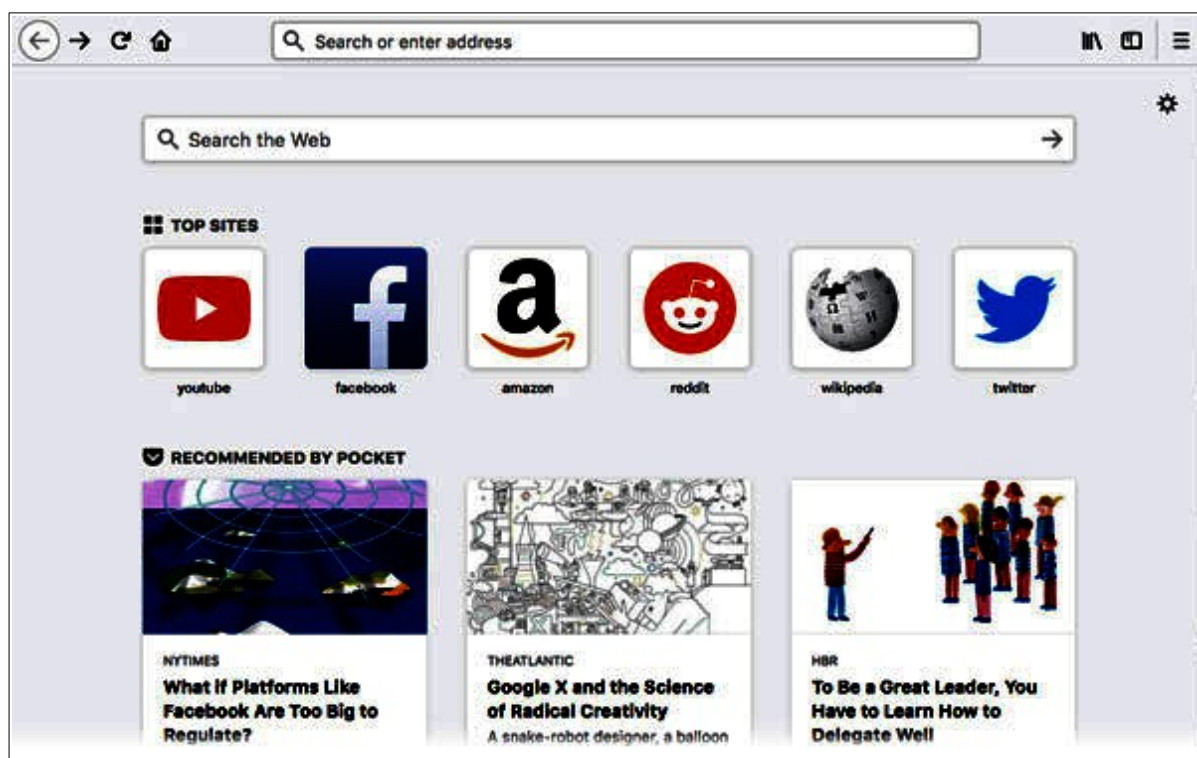
```
/home/seloZuzukino/firefox/firefox
```

ИЛИ

```
./firefox/firefox
```

– если вы уже в личном каталоге, но не в каталоге браузера.

Все популярные браузеры велики, и стартуют медленно. Вот что вы получите (в зависимости от настроек браузера и его версии).



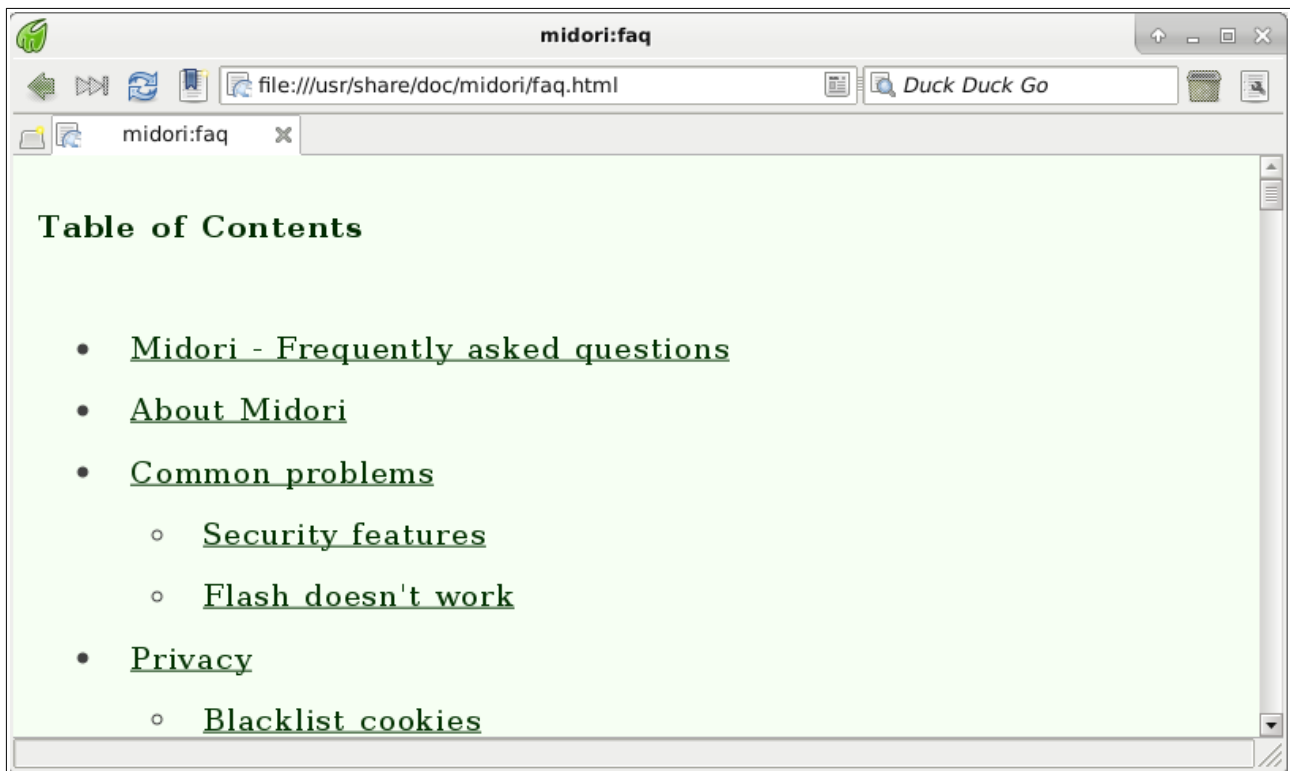
(по опыту установки на Debian 9)

Midori.

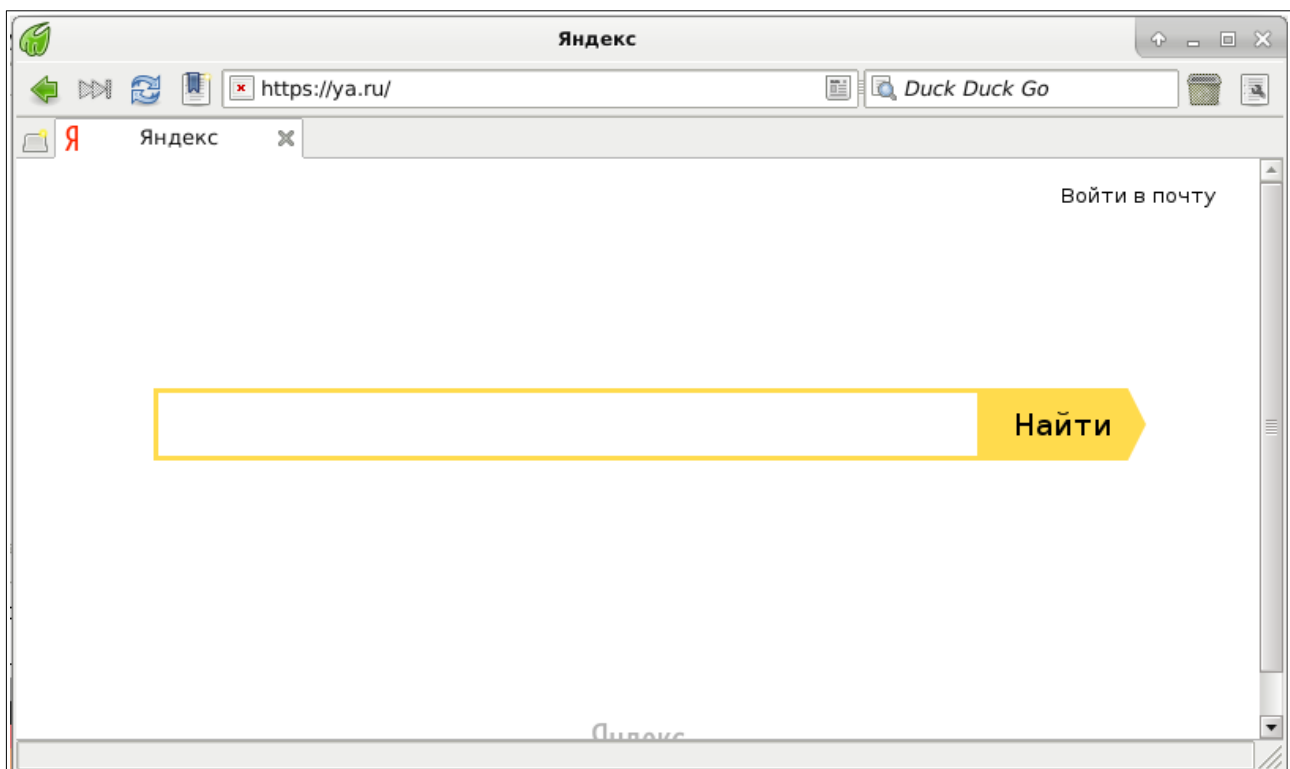
Браузер был штатным для ОС Raspbian, но он отсутствует в выпуске ОС Raspbian Lite. По техническим причинам мне трудно показать установку браузера. Когда он уже установлен, то может быть запущен командой (по аналогии с Firefox)

```
midori
```

Результат.



Вы получаете страницу часто задаваемых вопросов, но она хранится у вас на компьютере. Её показ ничего не говорит о выходе в интернет. Щёлкните в строке адреса сверху, и вместо показанного адреса введите более полезный – например, ya.ru. Нажмите **Enter**. Вот что вы получите (в зависимости от настроек браузера и его версии).



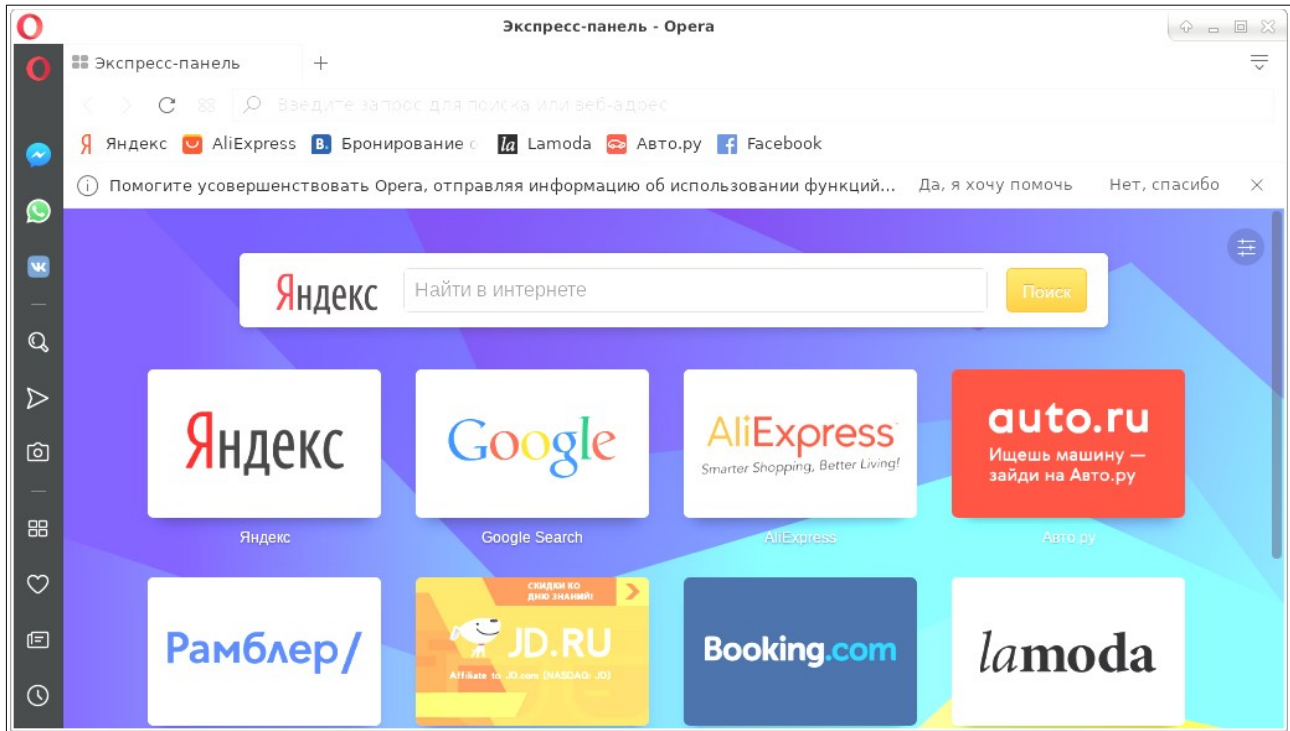
(по опыту установки на Debian 9)

Opera.

Этот браузер интересен встроенной функцией сжатия данных, и относительно небольшими требованиями к мощности и оперативной памяти. Он может быть запущен командой

`opera`

Результат.



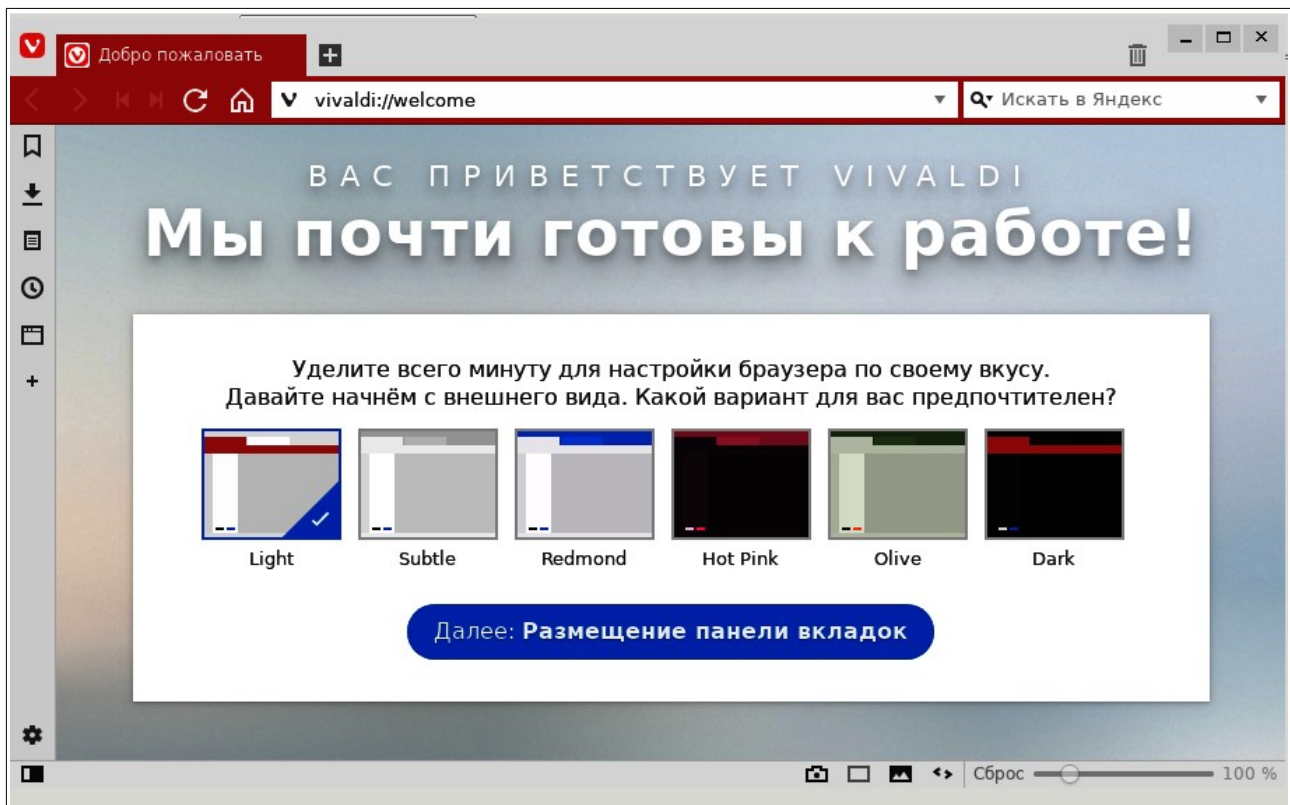
В разных версиях вид браузера может быть разным. Функции почти не меняются.

Vivaldi.

Браузер назван в честь великого композитора. Можно даже понадеяться что программисты его слушали. Браузер можно стартовать командой

`vivaldi`

Результат (в зависимости от настроек браузера и его версии).

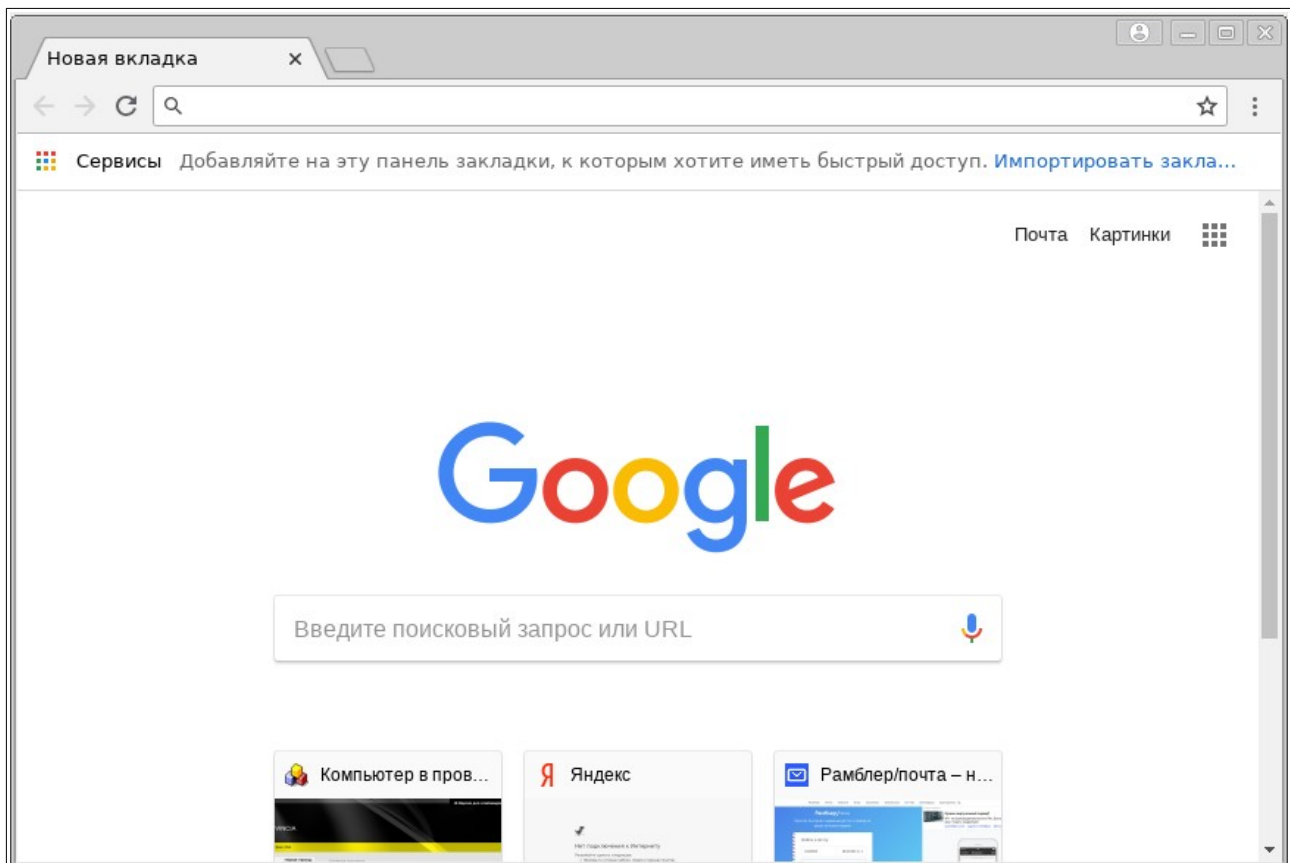


Chrome.

Очень популярный, но довольно требовательный к мощности и оперативной памяти браузер. Современные версии браузера для компьютера работают лишь на 64-битных ОС. Интересен прежде всего массой *своих* приложений, расширений. Магазин приложений и расширений содержит огромное количество этих модулей. Они не работают вне браузера, и не равны в возможностях обычным приложениям. Браузер можно стартовать командой

`google-chrome`

Результат.

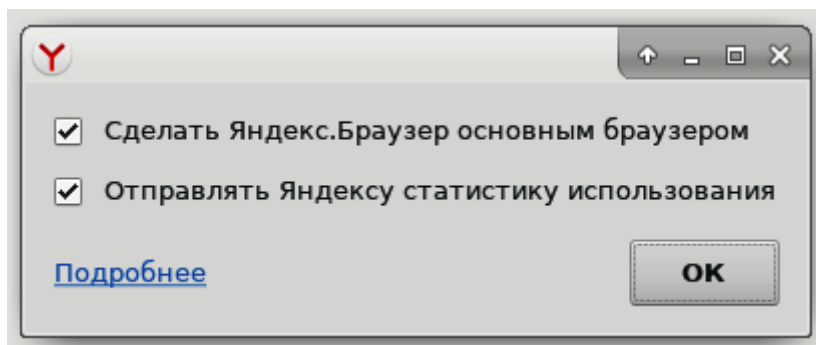


В данном случае показано как браузер выглядит сейчас у меня.

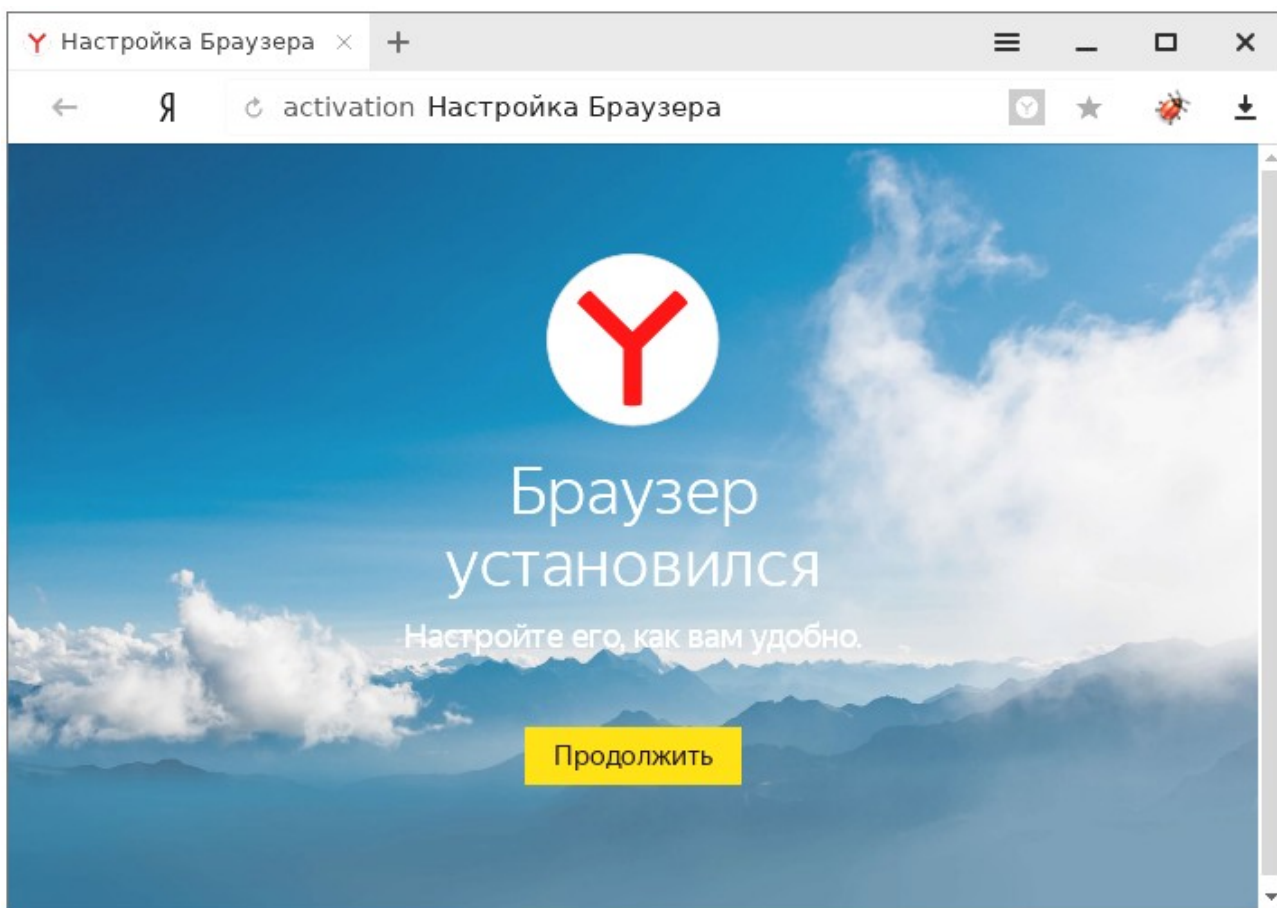
Yandex Browser (Яндекс браузер)

Браузер, всё ещё выпускаемый лишь для 64-битных ОС. Интересен встроенной технологией сжатия данных, и другим. Стартовать браузер можно командой [yandex-browser](#)

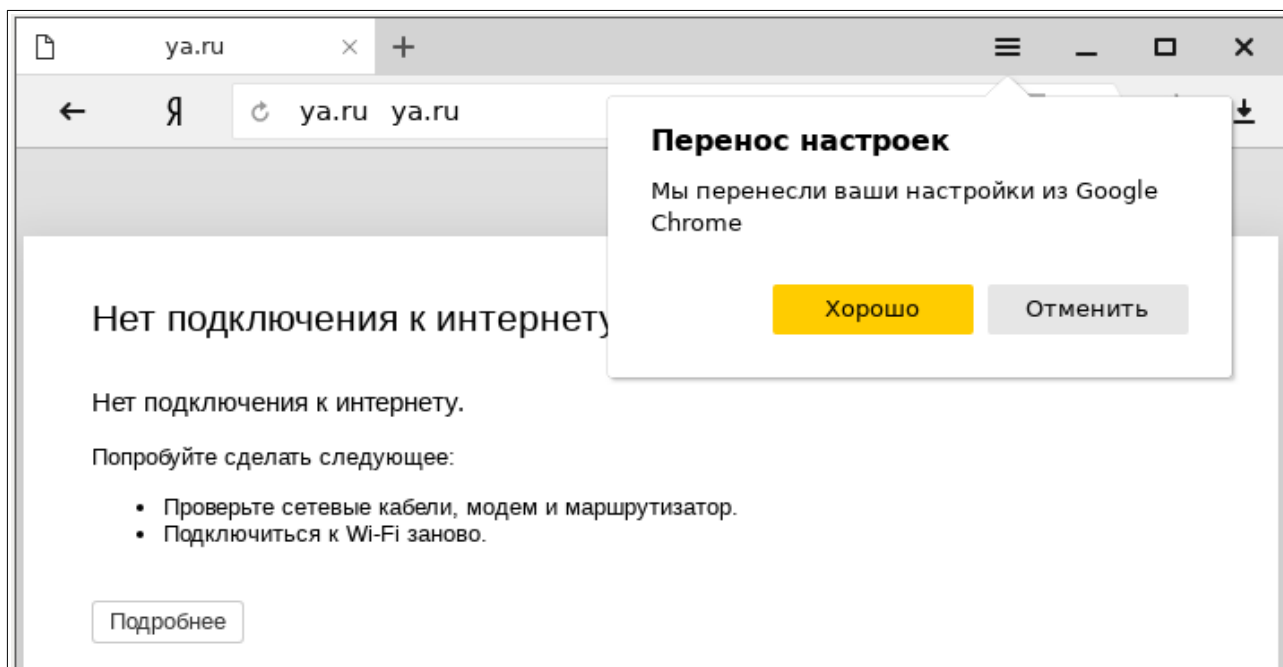
Первое что вы получите после запуска браузера – окно для установки некоторых параметров его работы. Их можно настроить и изменить позже.



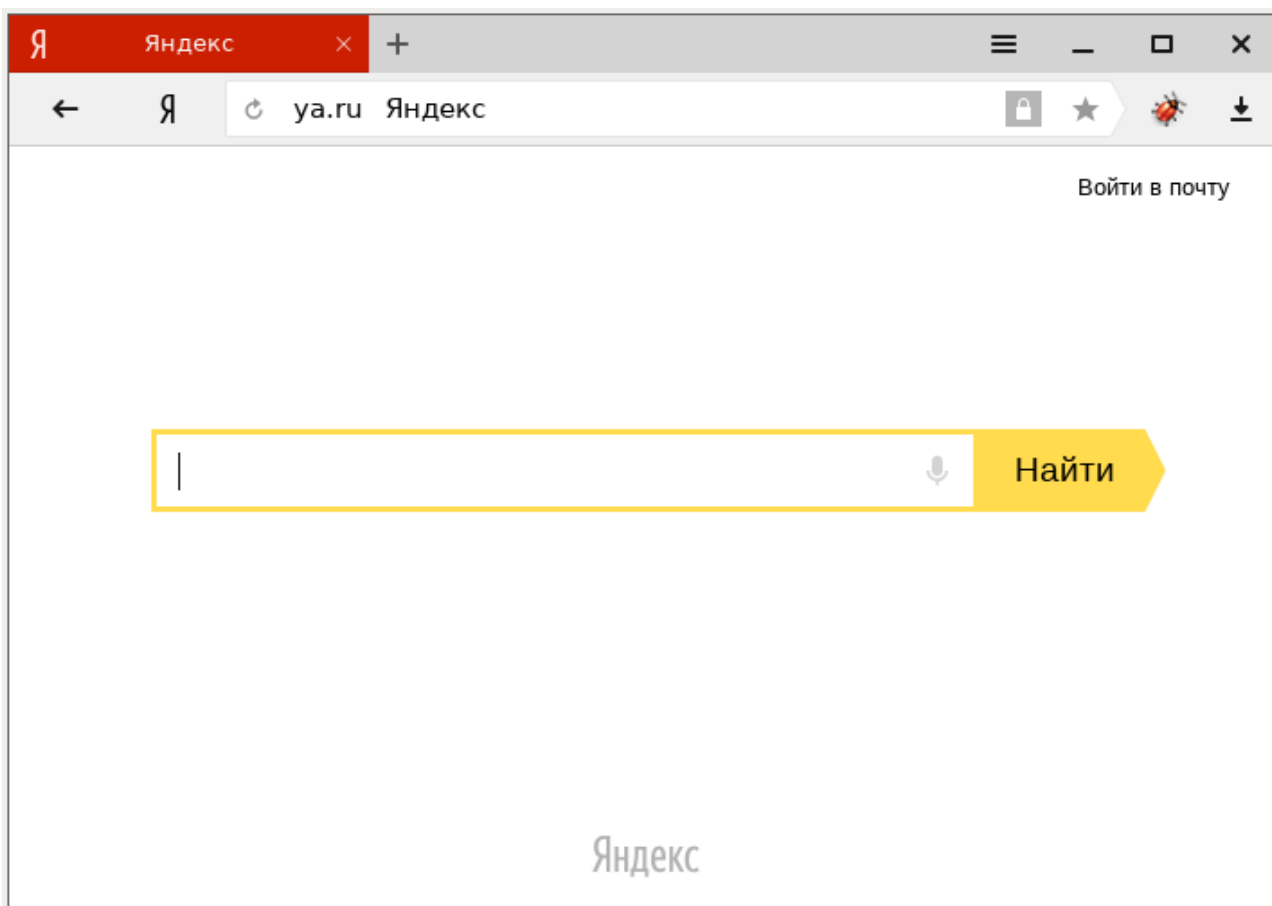
Тут можно ничего не менять, и просто щёлкнуть мышью ОК. Браузер запустится. Настроить его можно позже.



При попытке перейти на адрес ya.ru произошло следующее.



Показана карточка с сообщением о переносе параметров работы из Chrome. На этот момент и правда нет выхода в интернет. Это не поломка. После выхода в интернет, и обновления страницы, или после повторного запроса.



Для скрытия карточки достаточно нажать на ней кнопку **Хорошо**.

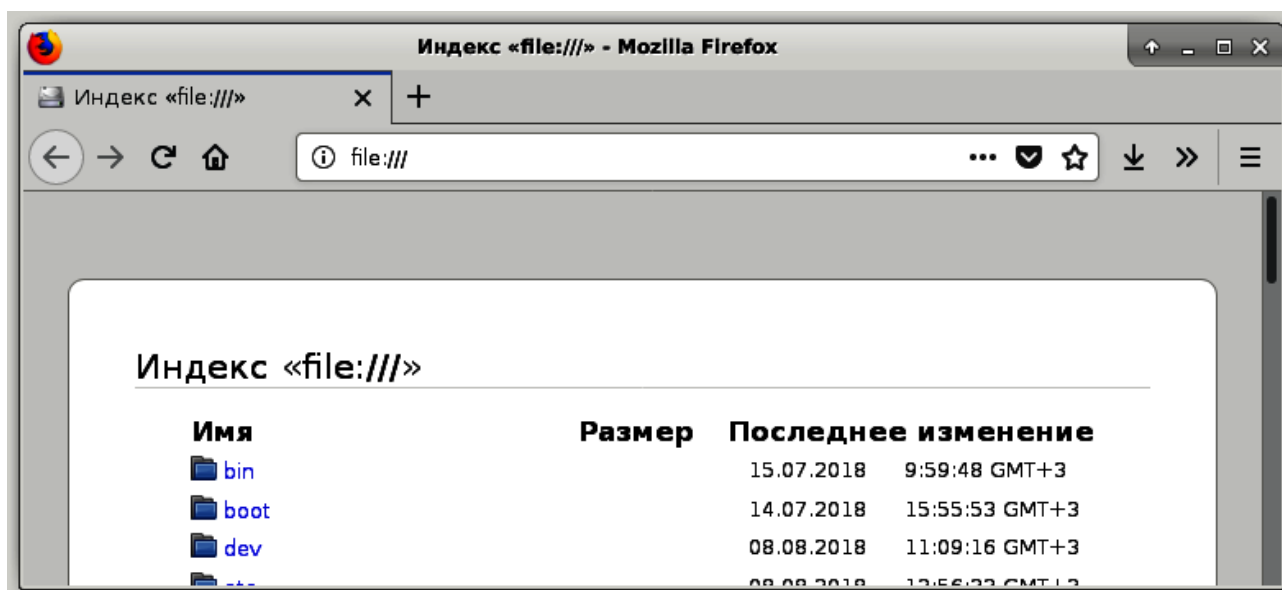
Общее описание окна браузера.

Окно браузера в общем виде выглядит так (опираясь на Firefox).



1. Заголовок окна. Полезен показом названия документа, и кнопками управления окном. Эти кнопки – такие же как в других программах. Их вид определяется настройкой ОС. Также в заголовке окна вы видите символическое изображение, обозначающее браузер. В случае Firefox это картинка с лисой.

2. Не сама вкладка, а её *язычок*. Вкладка по форме похожа на каталожную карточку. Основная её площадь – это площадь места для страницы. На язычке кнопка закрытия вкладки. **ВАЖНО** – это обрывает работу с сервисами, используемыми через эту вкладку. Перед закрытием вкладки выйдите из режима личного обслуживания. Щелчки на язычках сменяют просматриваемые вкладки с документами. Справа от крайнего язычка – кнопка с плюсом для создания пустой вкладки, или вкладки с содержимым, определяемым при настройке браузера.
3. Некоторые кнопки для полезных действий в показанной вкладке. Слева направо: кнопки **Назад**, **Вперед**, **На домашнюю** страницу браузера. Последняя определяется при настройке браузера. Для каждой вкладки есть своя история открытых страниц. Кнопками Вперед и Назад вы перемещаетесь по ней, по страницам.
4. Адресная строка. Она же обычно является строкой текстового запроса если в неё вводить не адрес. Адресная строка позволяет либо перейти на документ с известным адресом, либо обратиться к поисковой машине с текстовым запросом. И для того и для другого после формирования адреса или запроса надо нажать **Enter**. После этого на медленных каналах потребуется подождать. При проблемах с доступом в *интернет* будет сообщение браузера на месте страницы. При отсутствии *страницы* будет сообщение сервера об ошибке 404. Сюда же выдаются сообщения об ошибках сервера, и другие. Эти сообщения означают что вы имеете выход в интернет, но нужного документа по адресу нет, или он есть, но перемещён.
5. Некоторые полезные кнопки, в том числе кнопка меню – последняя на рисунке. После настройки браузера или добавлении расширений там появляются или исчезают кнопки. Это нормально. Набор кнопок определяется браузером.
6. Область где показана страница, или список файлов. В последнем случае окно выглядит примерно так.



Почему это так? Браузер умеет показывать список файлов и каталогов – без возможности изменить их.

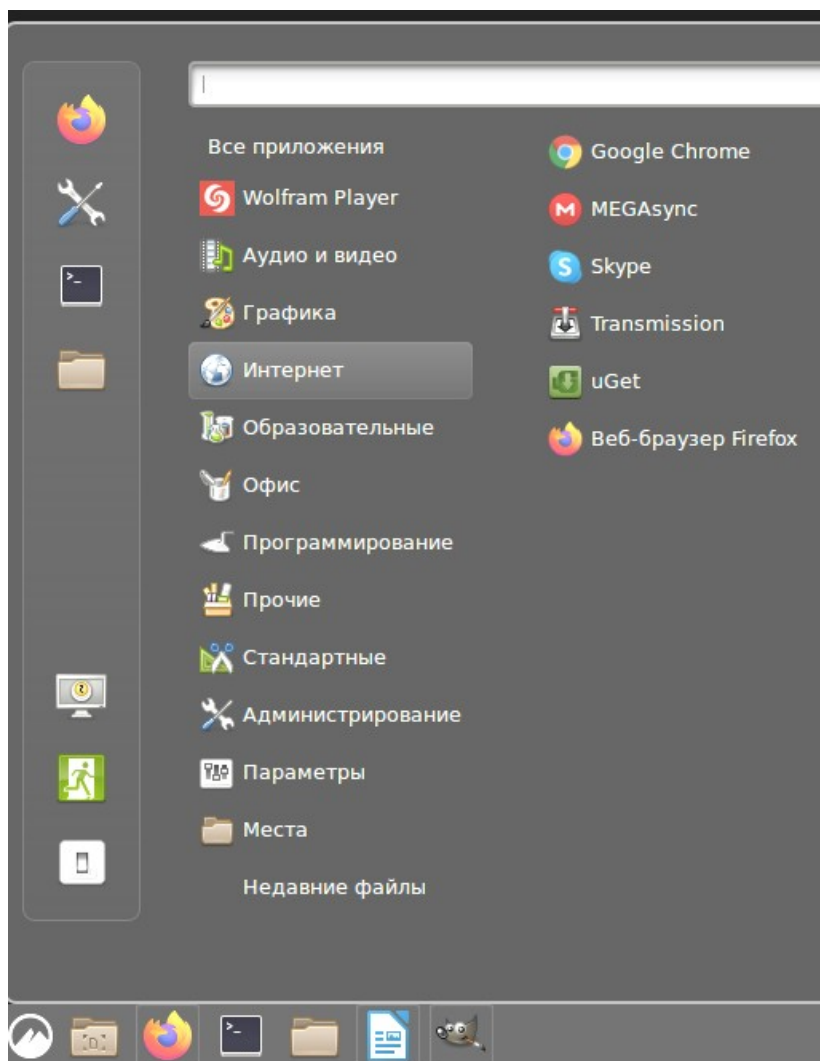
7. Место, где обычно показывается строка статуса. Её невозможно редактировать. Она показывает либо выполняемые браузером операции, либо его состояние, либо адрес того файла или страницы, который привязан к ссылке, на которой курсор мыши.

Тезисно обозначу другие сведения о браузерах.

- Ссылки, перенаправление и переход к странице. Ссылки переносят к другому документу или другой части документа. Перенаправлением пользователь не управляет. Оно происходит автоматически. Например, если на сайте нет нужной страницы то часто сайт перенаправляет браузер на главную страницу сайта.
- История или журнал. Журнал это набор пройденных ссылок, верность которых не

проверяется браузером. Тем не менее это полезная вещь тем более что ученик может искать слова в страницах журнала. Они могут храниться в кэше страниц браузера. Кэш можно очистить. Журнал при этом не стирается.

- Поиск текста. Выполняется поисковыми сервисами. В России это в основном Яндекс.
- Консольные браузеры – скорее всего неполноценные, но пригодны для поиска и чтения текстов, что уже немало.
- Под устройства с архитектурой на основе ARM-процессоров, то есть смартфоны и планшеты, выпускаются многие браузеры – в частности Firefox и легковесные браузеры вроде Yandex Lite.
- Запуск браузера, как и многих других программ при оконной среде часто возможен через меню. Так, в среде Cinppatop в левом нижнем углу экрана есть экранная кнопка для вызова такого меню.



- Домашняя страница это страница которая считается начальной при запуске браузера, или это основная страница сайта.
- На весь экран показывается страница или окно браузера в режиме полноэкранного показа. Нажатие **Esc** на клавиатуре обычно отменяет этот режим.
- Шифрование часто используется для секретности передаваемых данных при работе браузера. Но это не требует действий или знания ученика.
- Сохранение страницы можно сделать вручную для страниц которые наиболее полезны. Сейчас это обычно делается как сохранение в формат PDF в произвольном, выбранном учеником месте в файловой системе, точнее – в личном каталоге.
- Печать страниц давно освоена браузерами, но важно избегать печати лишнего –

например, рекламы. Для этого есть блокировщики рекламы-расширения браузеров, и режим чтения.

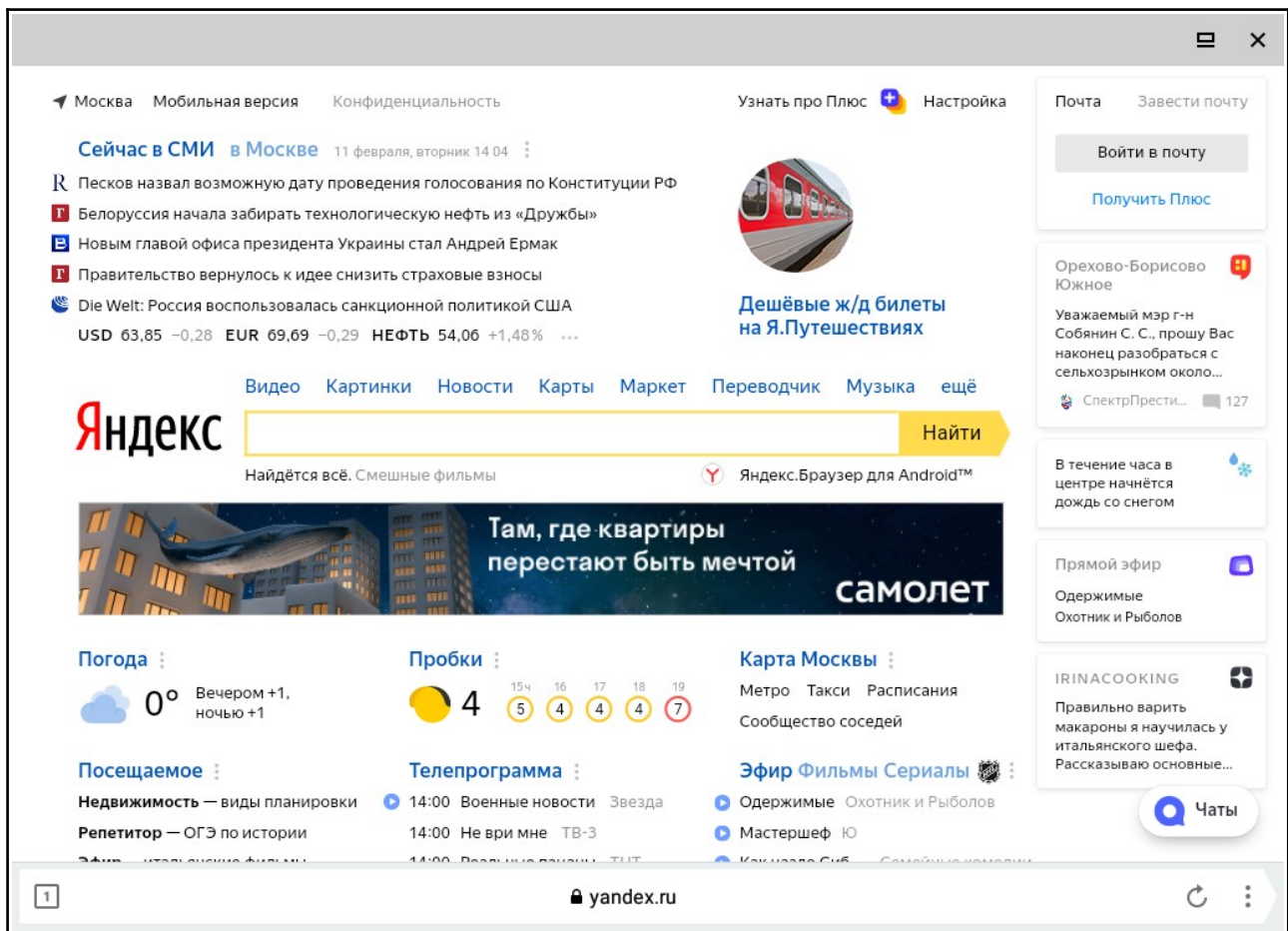
- Кодировка страницы обычно не нуждается в управлении. Тем не менее такая возможность обычно есть в браузере. Кодировка определяет вид страницы. Это особенно важно для старых страниц.
- Копирование ссылок из журнала и браузера – выполняется вызовом контекстного меню на ссылке.
- Режим чтения – если он возможен – регулируется значком в поисковой строке, или строке адреса браузера.
- Листание страницы проще всего выполнить нажатием кнопок «вниз» или «вверх» на клавиатуре при активном состоянии окна браузера. Но при этом важно что некоторые части страницы могут блокировать листание. Так, если текстовый курсор в строке ввода то листание не происходит. Для выхода из таких ситуаций используется щелчок мышью вне этих частей страницы.
- Сведения о странице обычно не нужны, но браузер их предоставляет.
- Пароли – используются для персонального обслуживания, не разглашаются.
- Приватный режим нужен когда требуется скрыть свою деятельность в браузере. Страницы не регистрируются в журнале.
- Разного рода реклама есть пожиратель трафика и времени, и может накладываться поверх страницы.
- Операции с текстом и изображениями могут быть нужны. Это копирование и вырезание текста и копирование изображений. Затем всё это можно вставить в OpenOffice, или другую программу.
- Скачивание файлов через браузер возможно, но оно не является всегда лучшим способом скачивания. Это копирование файла из интернета или локальной сети.
- Группы вкладок могут быть организованы браузером. Обычно это не нужно.
- Автономный режим браузера не означает отключения других программ от интернета, отключается лишь браузер.
- Элементы управления на страницах по сути такие же как на окнах программ, но их состояние и содержимое не сохраняется при сохранении страницы на компьютерном устройстве. Страницы рассчитаны на взаимодействие с серверным компьютером и его программами.

Браузеры в мобильных телефонах.

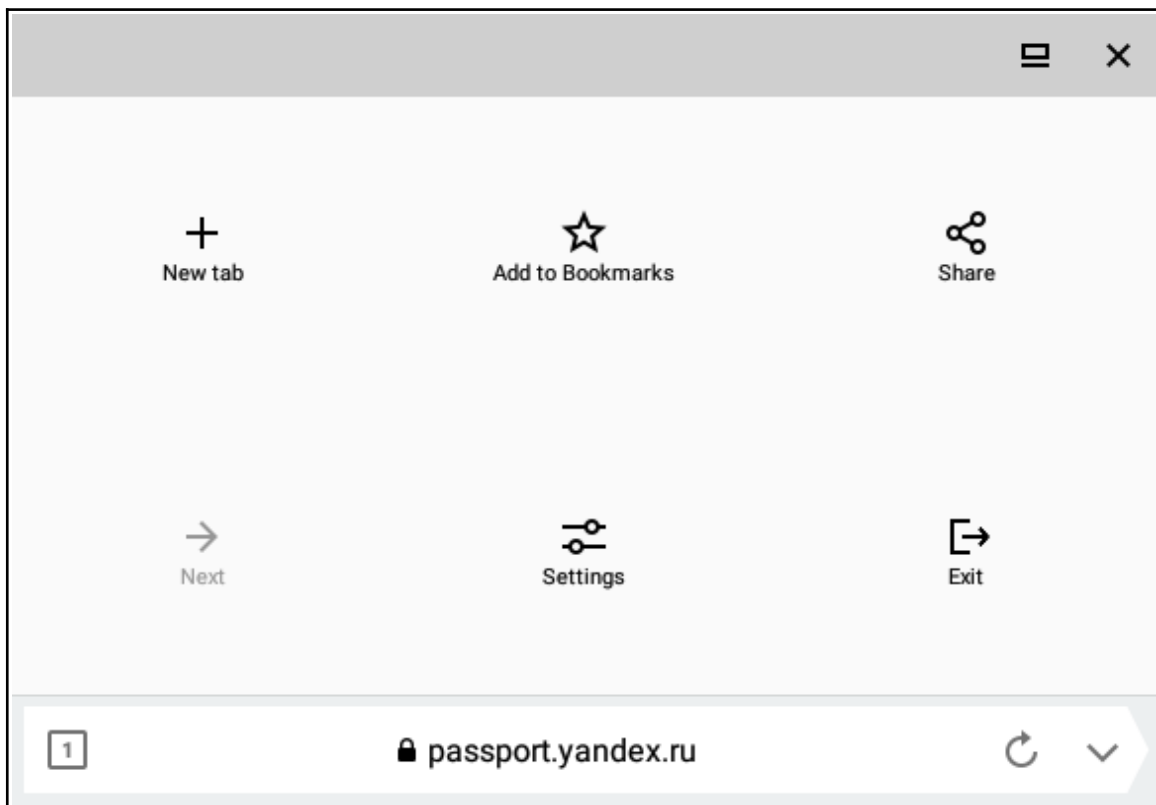
Всё ещё в эксплуатации какое-то количество старых телефонов с возможностью запуска Java-программ. Такими программами могут быть некоторые браузеры, например, UC WEB или Opera Mini. Так как перспектива этих телефонов – обмен на смартфоны, рассмотрим не эти браузеры, а **Yandex Mini** – на сегодня один из небольших браузеров, полноценность которого мне неизвестна, но функциональность достаточна для чтения текстов, поиска текста, просмотра иллюстраций в статьях. Для людей с плохим каналом в интернет это интересная программа.

Yandex Lite.

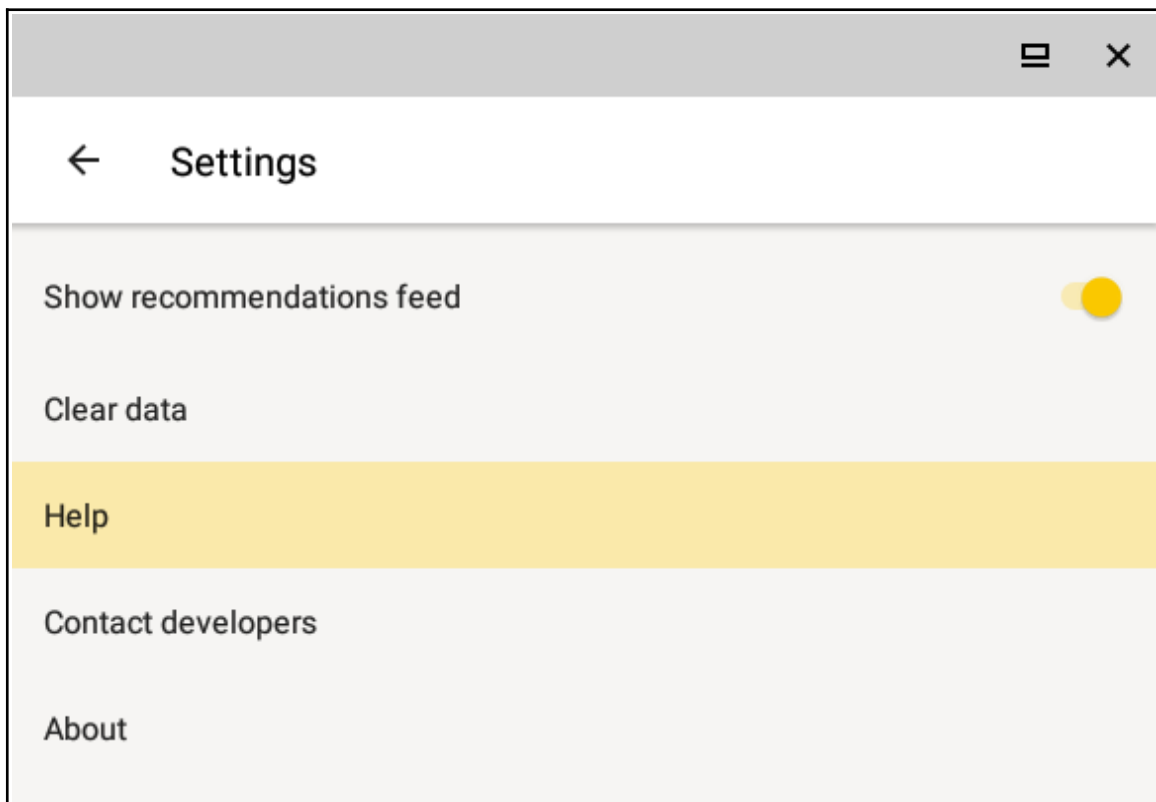
Первоначально её окно выглядит так.



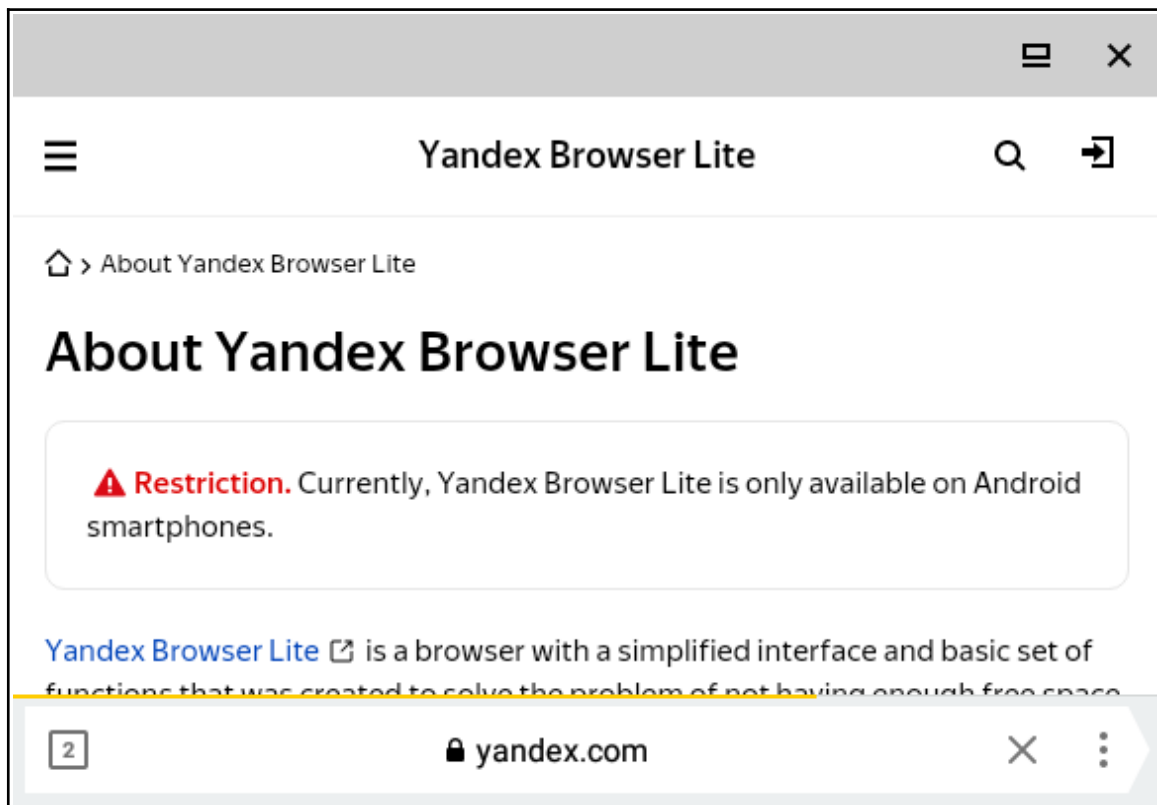
На смартфоне окно было бы уже, но я использую эмулятор **Anbox**, и вид более удобный. Строка адреса и поискового запроса внизу – где замочек и написан адрес страницы. Там же слева количество страниц в Журнале, а справа стрелка для обновления страницы и многоточие для показа меню. Вверху нечто вроде заголовка, но он не создаётся браузером. Нажатие многоточия вызывает не буквально меню, а нечто вроде полноэкранный панели с кнопками-надписями.



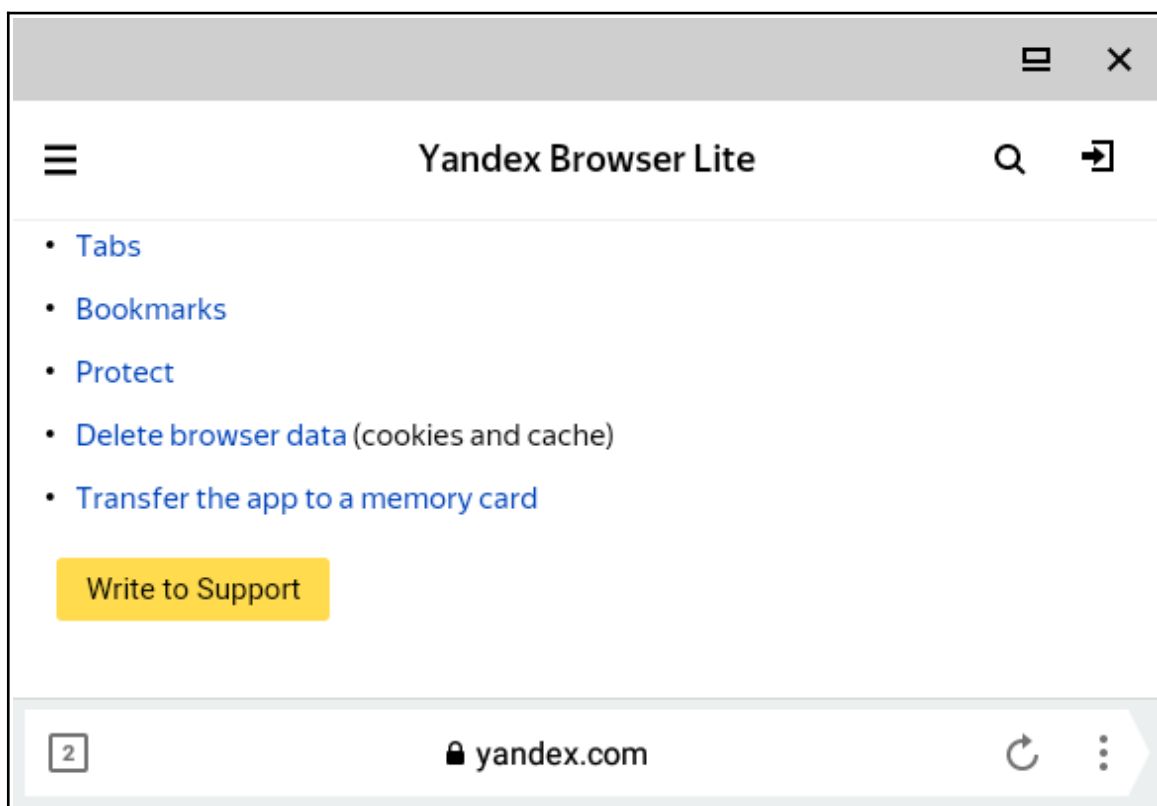
Попробуем настроить браузер нажав **Settings**.



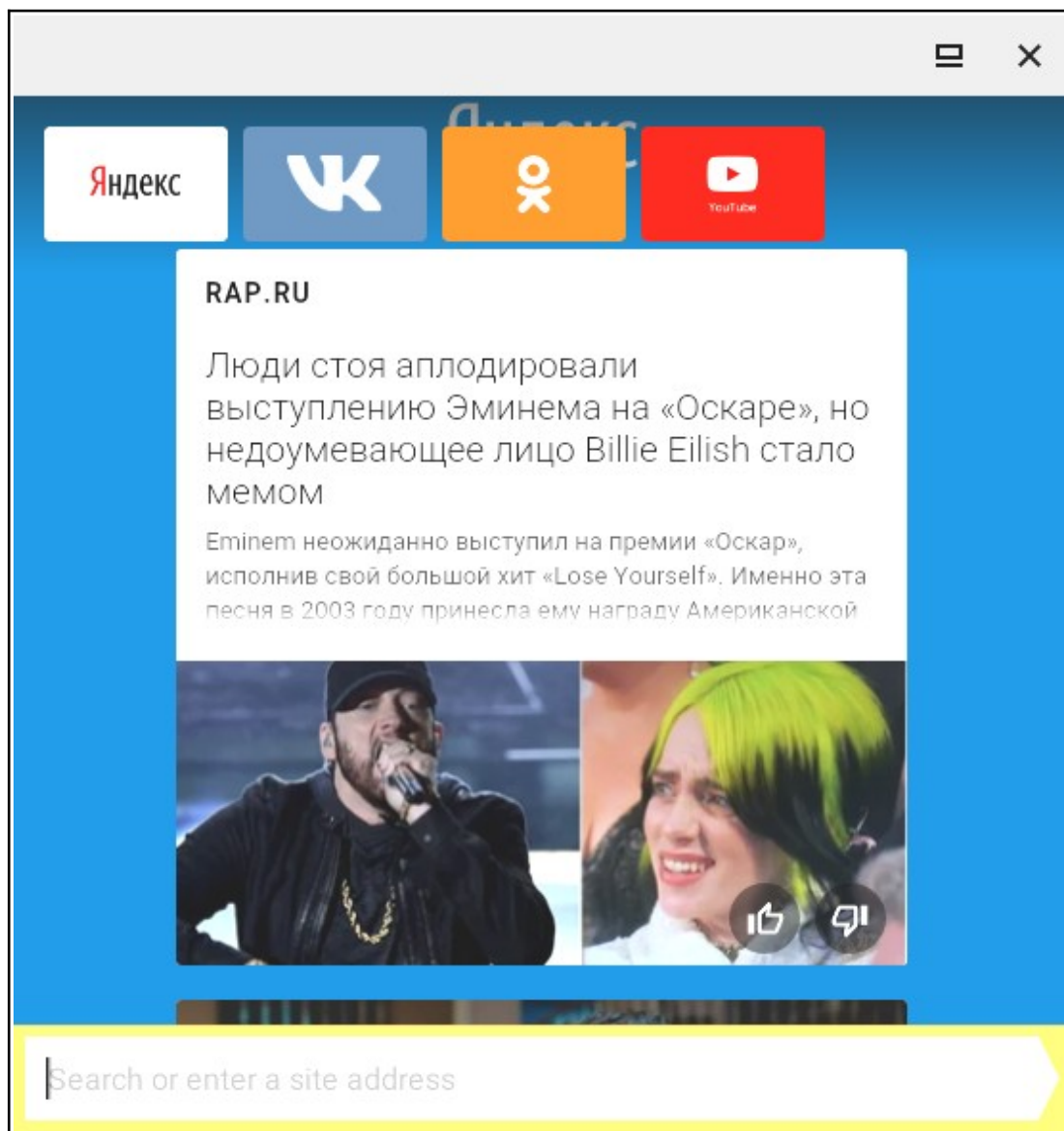
Никакой сложной настройки сделать нельзя. Строка «Show ...» имеет переключатель, который регулирует показ новостной ленты. Она в общем не нужна. Теперь попробуем использовать «Help». Получаем обычную страницу с описанием браузера.



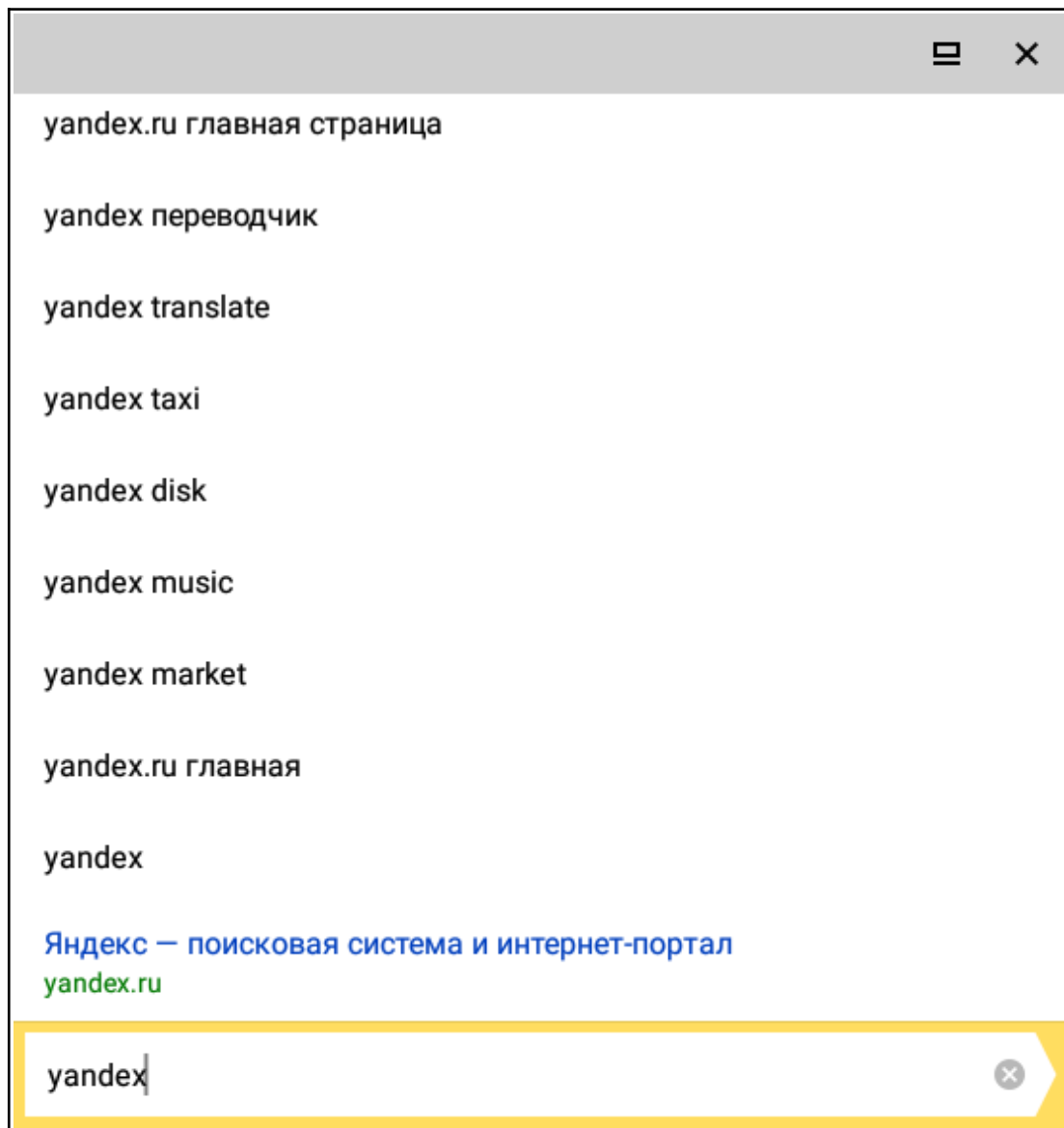
Написано, что программу можно перевести на карту памяти. Ниже кнопка для написания письма разработчикам программы. Но они не обязаны отвечать. Если письма пойдут потоком то они могут их игнорировать.



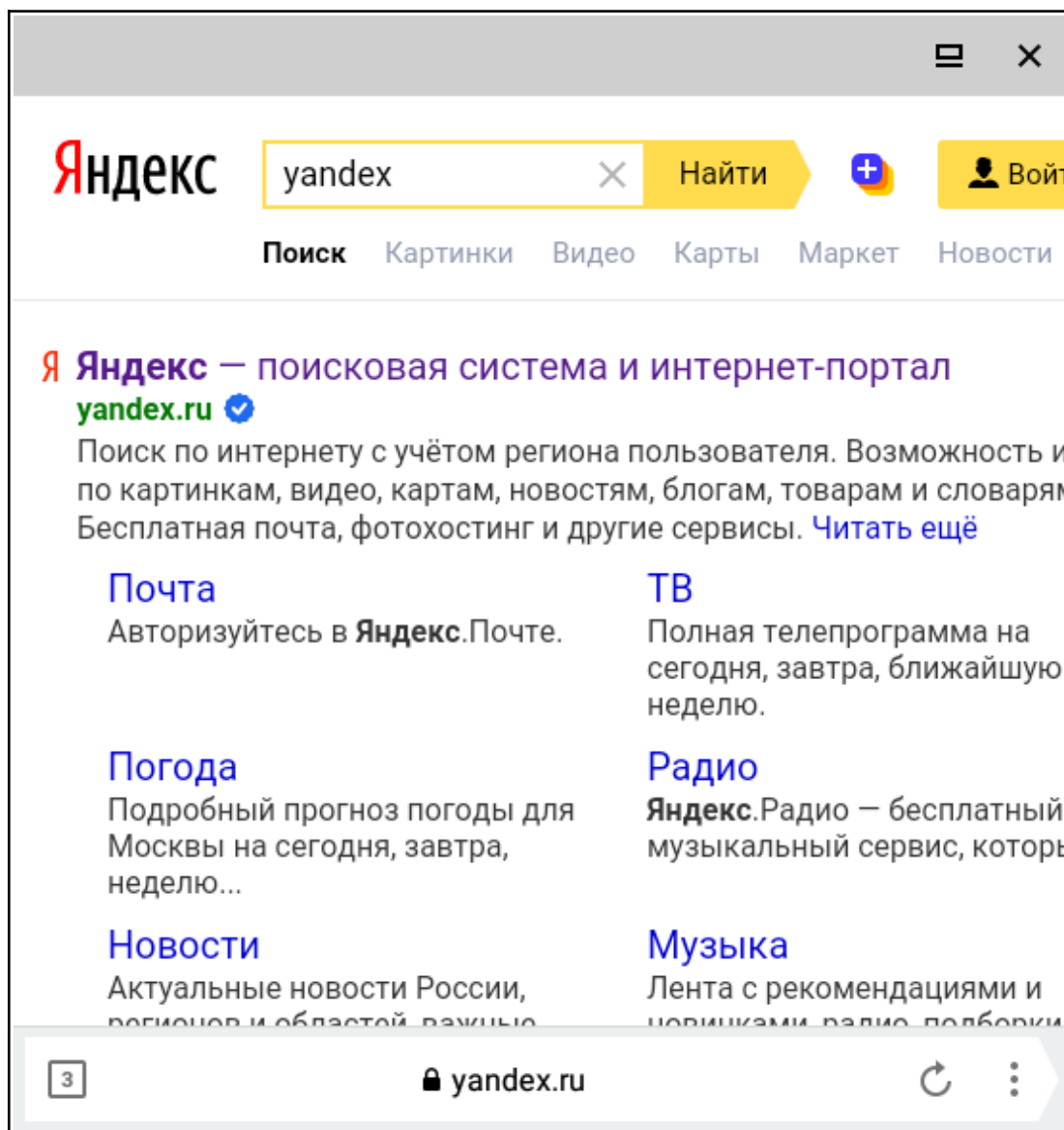
Теперь с помощью «меню» создадим новую вкладку («new tab»). Просто показана страница без язычка с лентой новостей.



Попробуем сделать текстовый запрос. Наберём в строке поиска внизу текст и нажмём кнопку запроса на клавиатуре (на устройстве с Android это экранная кнопка на клавиатуре).



Пока мы набирали браузер подбрасывал подсказки. Вот что получилось.



Искомое слово прежде всего вызвало сведения о фирме и поисковом сервисе Яндекс. В общем никаких особых возможностей. Но и сложностей. Теперь открываем адрес <https://support.mozilla.org/ru/products/firefox> и читаем русскоязычные статьи справки по Firefox.

Электронная почта (e-mail).

Эта книга не задумывалась как всестороннее описание вопросов компьютерной практики или работы различных сервисов. Но тут будет описана работа электронной почты – из-за её особого положения. Электронная почта могла бы вовсе исчезнуть и остаться на свалке истории если бы она не использовалась для регистрации в разных службах. Несмотря на наличие многих других способов общения почта всё ещё может быть необходима. Тут, видимо, сказывается и большое количество её клиентов (то есть людей), и древность самой почтовой службы.

Почтовая служба – это служба, занятая обеспечением личной переписки между людьми с помощью своих протоколов связи POP3 и IMAP. Пользователю не важно как конкретно происходит взаимодействие программ в соответствии с этими протоколами. Но ему важно что при этом происходит. Происходит пересылка писем – которые в простейшем виде есть текст. Почта – являясь личной перепиской, не является сама по себе секретной. Как и в случае бумажного письма сам текст письма в классическом случае не шифрован. Прежде чем

переписываться требуется выход в интернет, и настройка почтовой клиентской программы – если она используется.

Прежде чем настраивать программу, потребуется создать почтовый ящик на одной из конкретных почтовых служб. Эти службы представляют собой беспрерывно работающие инфосистемы, использование которых обычно ничего не стоит клиенту. Почтовая служба организуется и оплачивается некой коммерческой структурой, и эта структура должна обеспечить клиента информацией о своём использовании (а пользователь заинтересован эту информацию прочесть для настройки почтовой программы), а также может получать доходы благодаря рекламе, навязываемой клиентам. Автор пока не сталкивался со случаями прекращения работы таких служб. Но, если это происходит, то такая служба должна обеспечить пересылку писем даже после своего исчезновения так как почтовые адреса уже используются и должны работать дальше. Как это будет сделано – не забота клиента.

Почтовый ящик – это ёмкость для хранения писем, и её объём может быть не бесконечным. При заполнении ящика посланное туда письмо может быть возвращено отправителю. Так работали почтовые сервисы в 90-е годы. Могут работать и сейчас. Почтовая программа может не реагировать на этот факт. Поэтому нужно будет вручную отправить письмо заново. Получатель письма должен стирать или переносить письма из почтового ящика. Для операций с почтой почтовая служба выдаёт ему почтовый адрес и пароль к почтовому ящику. Почтовый адрес можно по мере надобности разглашать – но не буквально, или на бумаге. Иначе ваш почтовый ящик станет наполняться *спамом* – посторонними, ненужными письмами. Пароль вы не разглашаете. В частности, очень глупо и неудобно нескольким людям использовать один почтовый ящик.

Электронная почта в 90-е годы – в условиях очень медленных каналов связи – была основным способом использования интернета, и приносила реальное удобство. Обычно для пересылки малого письма нужно не более часа, некоторые авторы называют намного меньшие сроки. Это верно для всего мира – страна получателя не имеет значения.

Разные протоколы по-разному обрабатывают письма. И при настройке почтовой программы вам потребуется выбрать протокол. Всегда убедитесь, что ваша почтовая программа правильно настроена, и письма доступны лишь вам и получателю.

При официальной переписке неличного характера письма могут быть опубликованы.

При смене или переустановке программ, или ОС, следует *экспортировать* письма чтобы потом *импортировать* их в заново поставленную почтовую программу. Об этом надо думать при смене почтовой программы. Иначе вы потеряете письма.

С помощью почтовых служб работают списки рассылки писем. Это просто набор почтовых адресов. Какой нам интерес использовать эти списки? Ответ очень прост. Они заменяют форумы интернета и в то же время не содержат рекламы. Кроме того, они обеспечивают большее прочтение писем, и большую вероятность помощи благодаря переписке.

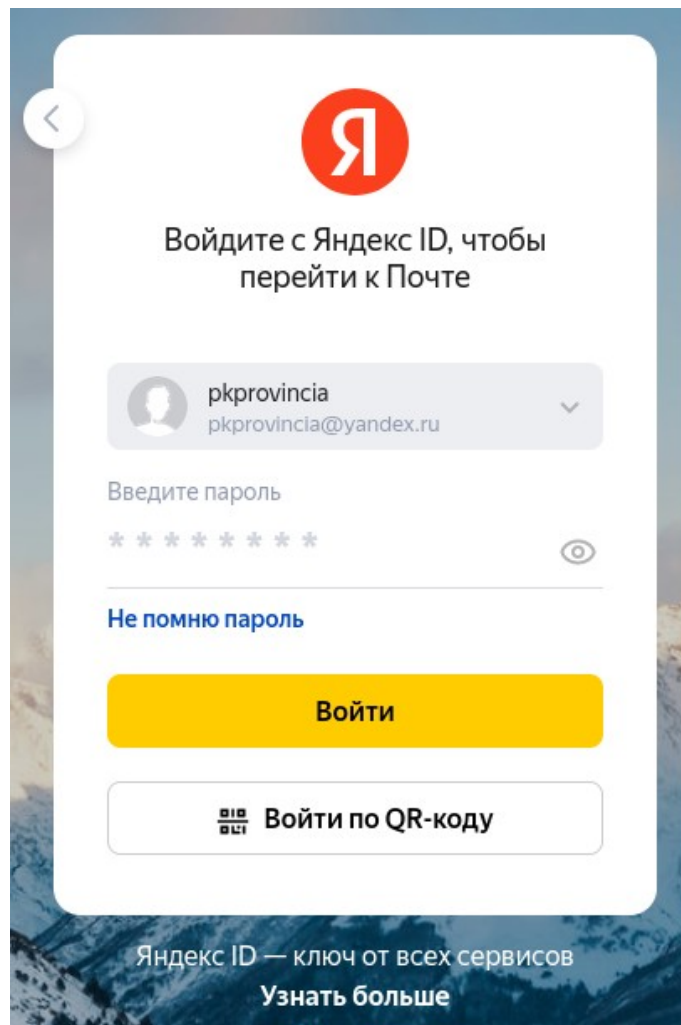
Также многие почтовые службы обеспечили операции с почтой на своём сайте, при помощи полноценного браузера. Операции с почтой выполняются в личном режиме. Поэтому – ВАЖНО – всегда выходите из ящика до использования техники другим человеком.

Перевод документации по почтовой программе **Thunderbird** находится на <https://support.mozilla.org/ru/products/thunderbird>.

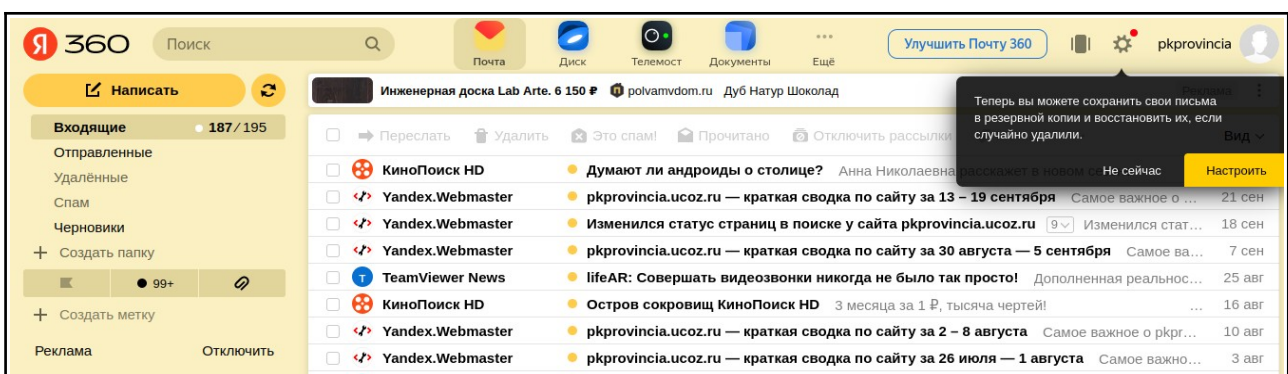
Работа с почтой в браузере.

В интернет-кафе или библиотеке использовать почтовую программу достаточно трудно. К тому же её может не быть. Многие почтовые службы доступны через браузер. В том числе это браузер в смартфоне. Почтовая программа хороша на частной технике. Рассмотрим обращение с почтой Яндекса в браузере компьютера.

1. наберите в строке адреса ya.ru, и нажмите **Enter**. На показанной странице выберите **Войти в почту**.
2. Яндекс выдаст карточку для ввода пароля.



3. Введите свой пароль от почты Яндекса. Он выдаётся при регистрации. Пароль показан точками, а не буквами. Щёлкните кнопку **Войти**.
4. Если всё правильно то Яндекс показывает содержимое вашего почтового ящика. Часть экранного изображения с моим почтовым ящиком.

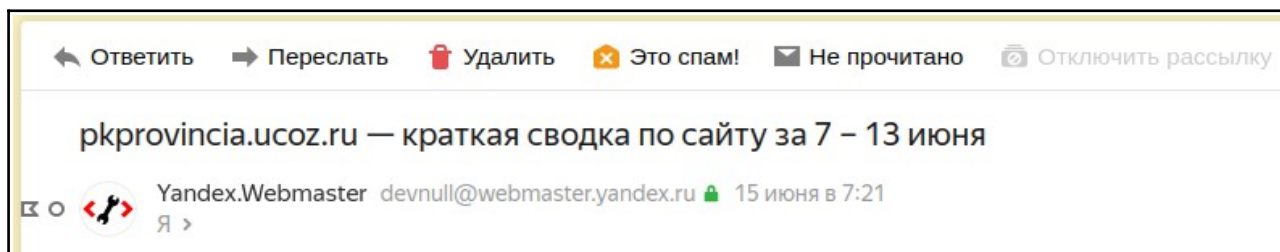


Наиболее свежие письма вверху в списке писем. Такие списки пронумерованы, и доступны по ссылкам внизу страницы. Щелчок по теме письма открывает письмо. Чтобы оттуда вернуться ко входящим письмам щёлкните **Входящие**. Чтобы написать письмо щёлкните **Написать** или **Ответить** внизу письма. В некоторых случаях письма не предполагают ответы, и отвечать бессмысленно. Отправленные письма в папке **Отправленные**. Письма, отправленные вам, могут по ошибке попасть в папку **Спам**, то есть папку почтового мусора. Вы не обязаны читать письма.

5. Чтобы выйти из почтового ящика, и личного обслуживания почтовой службой Яндекса, выберите в правом верхнем углу начало своего почтового адреса, и в появившейся карточке щёлкните **Выйти из сервисов Яндекса**.

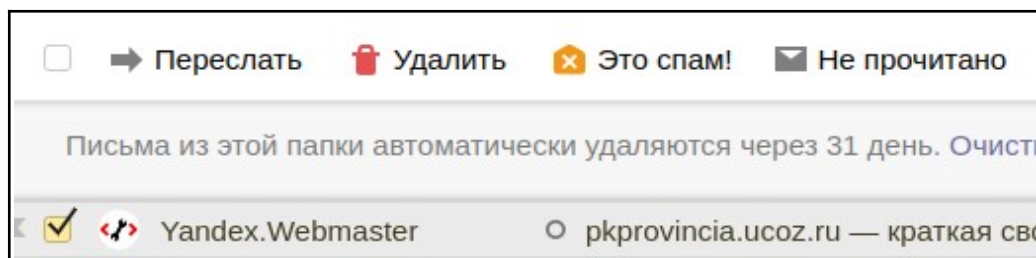
Как удалить письмо.

Напоминаю, что удалять надо не все письма, а те, которые действительно секретны. Обычное удаление есть удаление в **Удалённые**, а не совсем. Вначале зайдите в письмо.



Прочтите его чтобы случайно не удалить ценное. Затем нажмите **Удалить**. Письмо будет перенесено в **Удалённые**.

Щёлкните **Удалённые**. Отметьте галочкой письмо как показано.



Щёлкните **Удалить**.

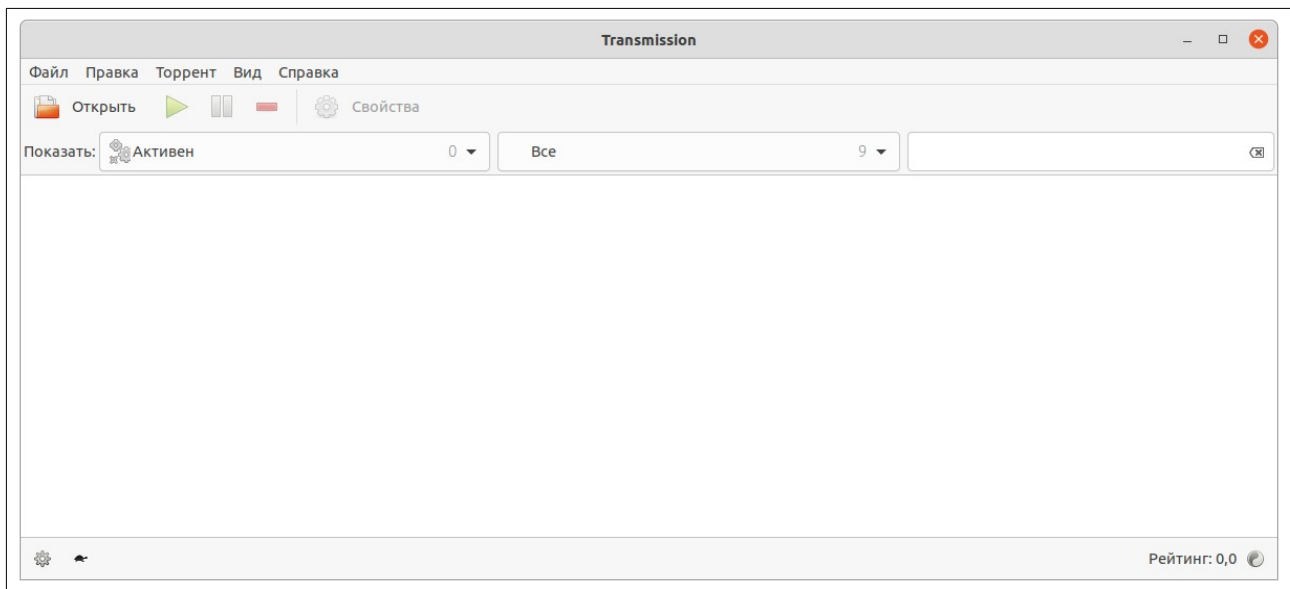
Торренты.

Торрентами называют файлы, которые сами не представляют интереса, но позволяют получать из интернета нужные файлы. Чтобы использовать торрент потребуется программа способная с ними работать. То есть не только получать, но и раздавать файлы. Раздачу можно регулировать. Но если все остановят раздачу то никто не сможет получать. Такая программа позволяет простым образом раздавать сразу много файлов без дополнительных действий. Пользователю обычно не нужно самому создавать торрент. Кроме того можно проверить полученные файлы и убедиться что они правильно передалась. С торрентами можно работать в консоли, но удобнее это делать в оконной среде.

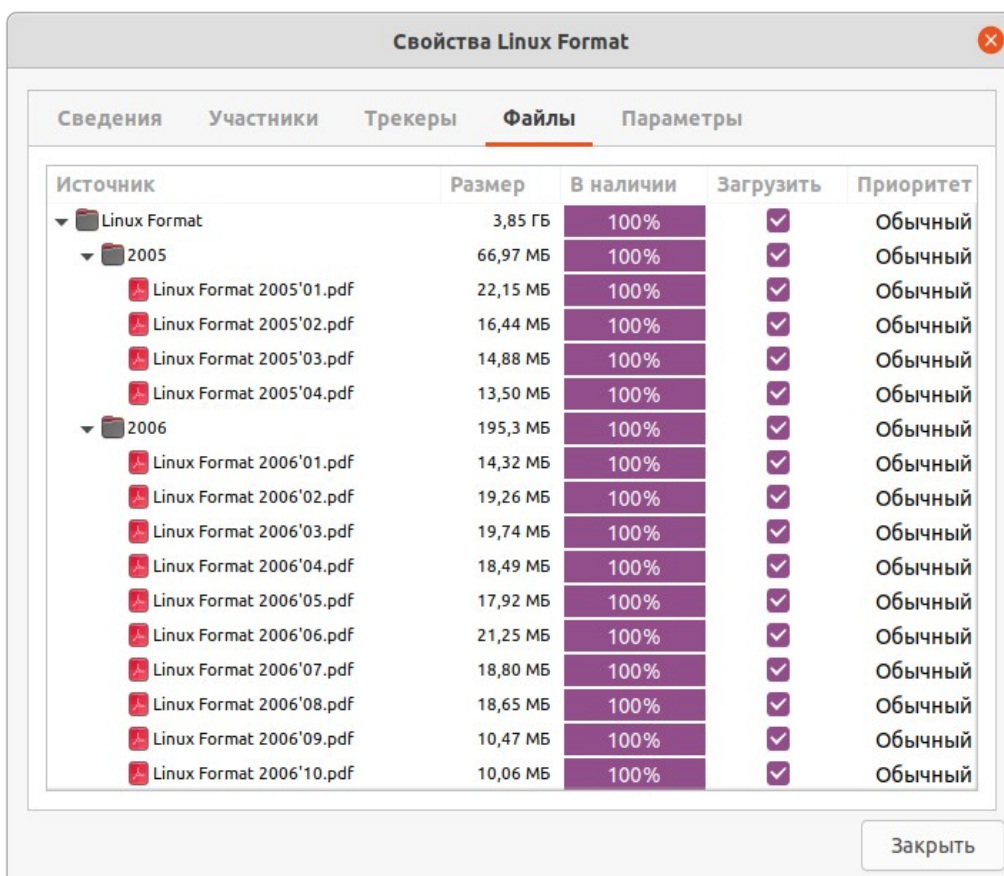
Немного рассмотрим программу **Transmission**, которая часто уже поставляется с дистрибутивами Linux, поэтому её не надо ставить. Торрентный файл берётся в интернете, обычно он имеет размер 20-30 килобайт. По сути это набор ссылок. Понимать и открывать этот файл не надо. Также программа умеет вместо такого файла использовать magnet-ссылку, её копируют со страницы в интернете. Сохраните файл в компьютере. Или браузер сам попытается найти программу для использования файла. Если у вас нет программы в меню программ то её можно запустить в терминале, набрав

`transmission-gtk`

(по опыту работы в Ubuntu). Программа имеет такой вид.



Щёлкните **Открыть**, и выберите файл торрента. Получение любых файлов в конкретный момент не гарантировано, это определяется наличием отдающей стороны и качеством канала в интернет. Программу можно оставить включённой на много часов. Или даже дней. Получение файлов будет отображаться прогресс-индикатором в виде растущей полосы. Иногда не нужно получать все раздаваемые файлы. В таком случае используйте контекстное меню на полосе индикатора; выберите **Свойства**, и получите диалоговое окно где можно отменить получение ненужных файлов.



Yandex messenger.

Яндекс Мессенджер по сути мало отличается от **WhatsApp**. Эта программа позволяет переписку, преобразование речи в текст, звонки – всё это через интернет. Таким образом вам не требуется обычная телефонная связь. Для звонков, как всегда, скорость передачи данных должна быть достаточно большой, и стабильно большой. Но даже на плохих каналах связи можно переписываться. Это – выход для людей, у которых плохо с мобильной связью. Даже под Москвой есть такие места, и в городах. Также с текстом можно отправлять файлы. Если нужно отправить фото то проследите чтобы файл фотографии не был слишком большим. Проще всего этого добиться при съёмке – регулируя размер фото или получаемого файла, то есть количество мегапикселей. Даже большие файлы передаются, но современные смартфоны могут создавать большие файлы низкого качества. Размер прямо не связан с качеством. Для хорошего фото любой смартфон может не годиться, или он будет слишком дорогим. Программа работает на смартфонах, и встроена в распространяемый Яндексом браузер.

Aria2: копируем файлы из интернета, в том числе через торрент.

Торрент – это, по сути, набор адресов в текстовом файле. Благодаря ему можно скачать более одного файла, в том числе с компьютеров, не имеющих белого IP-адреса. В качестве оконной программы для торрентной скачки часто используют **Transmission**.

Программа **aria2** обычно не входит в поставку ОС, её придётся ставить. В Debian это делается так.

```
apt-get install aria2
```

Немного сведений о программе. Программа консольная. Поэтому она не зависит от наличия оконной среды. Для получения файла с известного адреса подайте команду (пример).

```
aria2c http://host/image.iso
```

Для возобновления оборванного скачивания повторите команду в том же каталоге. **aria2** может восстанавливать поврежденные загрузки по информации файлов **.torrent** или **.metalink**.

Это делается так.

```
aria2c -M test.metalink --check-integrity=true
```

Скачка с помощью торрентного файла делается так.

```
aria2c http://www.clearbits.net/get/2068798798.torrent
```

То есть идёт скачка файла с помощью торрентного файла, который берётся в интернете.

Получить несколько файлов или использовать несколько торрентных файлов можно с помощью опции **-i**. Например, так.

```
aria2c -i ariainput
```

Тут **ariainput** – произвольное имя текстового файла с адресами – один адрес на одну строку.

Вывести список файлов из **.torrent**, **.meta4** или **.metalink** можно так.

```
aria2c -S 7687687.torrent
```

Тут **7687687.torrent** – имя торрентного файла для примера, оно могло бы быть другим.

Регистрация Яндекс (Yandex).

Регистрация позволяет вам работать в условиях личного режима не только в браузере, вы идентифицированы, и получаете дополнительные возможности. Например, вы можете не только читать, но и писать в форуме. Так вы сможете задавать вопросы более подготовленным людям. Регистрация нужна для использования электронной почты. Яндекс бесплатно выдаёт почтовый ящик при регистрации. Сама регистрация бесплатна.

Дальше я приведу текст, составленный самим Яндексом.

Имя и фамилия

При регистрации рекомендуется указывать настоящие имя и фамилию. При возникновении проблем с доступом служба поддержки может сверить эти данные с вашим гражданским паспортом, если остальные способы решения проблем не помогли.

Логин

Логин — это уникальное имя вашего аккаунта на Яндексе, которое вам нужно будет вводить при каждом входе в аккаунт. Также логин определяет адрес вашей Яндекс.Почты: если вы выберете логин `ivanov2009`, вашим адресом станет `ivanov2009@yandex.ru`.

Для логинов Яндекс не различает точку, дефис, заглавные и строчные буквы, поэтому логин `ivan.ivanov` считается таким же, как `ivan-ivanov`, `ivan.Ivanov` или `Ivan-Ivanov`. Вы можете указывать имя вашего аккаунта или адрес почты в любом написании.

При этом на сервисах Яндекса логин будет отображаться в том же регистре, в котором вы указали его при регистрации. Изменить это не получится.

Ограничение. Логин не может начинаться с цифры и не должен содержать символ нижнего подчеркивания `_`. После регистрации изменить логин нельзя.

Пароль

Пароль — это главное средство защиты вашего аккаунта. Пароль должен отвечать трем основным требованиям:

- он должен быть известен только вам;
- вам должно быть легко его помнить;
- его должно быть сложно подобрать.

Внимание. Никогда и никому не сообщайте пароль, с помощью которого можно войти в ваш аккаунт. Чем больше людей знают ваш пароль, тем выше вероятность того, что его узнает злоумышленник.

Не используйте простые пароли: `12345`, `qwerty`, `password` и т. п. Также не указывайте в качестве пароля свои имя, фамилию, день рождения или номер паспорта — все эти данные о вас могут быть доступны посторонним.

О том, как лучше защитить пароль и аккаунт, читайте в разделе [Защита вашего аккаунта](#).

Номер телефона

Номер телефона, привязанный к вашему Яндекс ID, поможет вам восстановить пароль с помощью SMS.

Если вы ввели номер телефона при регистрации и успешно зарегистрировались, Яндекс отправит вам код подтверждения. Этот код нужно ввести на странице [Телефонные номера](#).

Подробнее о том, как привязать телефон, см. в разделе [Привязка телефонных номеров](#).

Дополнительный адрес почты

Еще вы можете восстановить доступ к своему аккаунту с помощью другой почты. Если вы забыли пароль, Яндекс может прислать вам письмо с инструкциями на привязанный адрес.

Подробнее о том, как привязать новый адрес, см. в разделе **Дополнительные адреса почты**.

Контрольный вопрос

Если вы не привязали к Яндекс ID номер телефона или адрес почты, сбросить пароль можно, ответив на контрольный вопрос. Вопрос и ответ на него вы задаете сами при регистрации.

Так как контрольный вопрос позволяет сменить пароль, придумывать вопрос и ответ стоит так же внимательно, как и пароль. Не стоит давать ответ, который может знать кто-то, кроме вас. Также не стоит задавать слишком сложный или неверный ответ — его будет легко забыть.

Примечание. Удобнее и надежнее сбрасывать пароль с помощью мобильного телефона или дополнительного адреса почты. Если вы привязали к своему Яндекс ID номер телефона или адрес почты, контрольный вопрос при восстановлении будет недоступен.

Изменить контрольный вопрос или ответ на него можно в настройках аккаунта. Для этого нужно ввести ответ на вопрос, указанный ранее. Убедитесь, что вводите ответ правильно и в нужной раскладке клавиатуры.

Если правильный ответ вспомнить не получается, вам будет предложено обратиться в службу поддержки через специальную форму.

Регистрация Google принципиально ничем не отличается от описанной. Она также бесплатна. Для увеличения своих возможностей рекомендуется иметь обе регистрации. Google так же дарит почтовый ящик.

Парсинг сайтов и автоматизация работы браузера.

Эта тема – сложная, и не входит в опыт автора. Она нужна когда получение любого доступного браузеру материала затруднено, требует, как кажется, большой ручной работы – которой мы избегаем. Парсинг – это автоматизированный процесс сбора данных с сайтов.

Как можно парсить сайты?

- С помощью Selenium. Selenium — это набор инструментов для автоматизации веб-браузеров с открытым исходным кодом. Сценарии Selenium могут быть написаны на различных языках – Python, Java, C#, JavaScript, Ruby. Selenium – настоящий комбайн для парсинга.
- С помощью Web Robots – сервис работающий как расширение для браузера Chrome. Сервис прост в использовании, имеет интерфейс для визуального захвата данных, разобраться с ним не составит особого труда, главное преимущество – сервис автоматизирует действия. Функциональность сервиса так же проста, парсер подойдет для самых простых задач.
- С помощью, например, Python, и программирования произвольным образом.
- С помощью (как я думаю) регулярных выражений.

Последний способ достаточно легко объяснить. Достаточно скачать страницы – возможно, не все подряд, и искать в них ссылки. Это можно сделать, как пишут, с помощью такого регулярного выражения.

```
(https?:\//)?([\da-z\.-]+\.[a-z\.\]{2,6})([\\/\w\.-]*)*\//?
```

Экономия и стоимость.

Выделенные линии.

На сегодня в городах, и, возможно, близких пригородах действует городская техника передачи данных, и городские тарифы. Вот достаточно показательный тариф московского провайдера.

Скорость, до, мегабит	100
Месячная абонентская плата, рублей	450

Лимит на количество переданных данных, в месяц	нет
--	-----

Вот тариф сибирского провайдера.

Скорость, до, мегабит	4
Месячная абонентская плата, рублей	250
Лимит на количество переданных данных, в месяц	нет

Тариф для Омска.

Оборудование, цена, рублей	2900
Подключение и настройка, цена, рублей	300
Скорость, до, мегабит	8
Месячная абонентская плата, рублей	500
Лимит на количество переданных данных, в месяц	нет

Мобильные данные.

Подмосковная расценка.

Скорость, до, мегабит	?
Месячная абонентская плата, рублей	300
Лимит на количество переданных данных, в месяц	7 гб

Специальное предложение для загородного дома.

Монтаж оборудования, и оно само	10000 – 15000
Скорость, до, мегабит	?
Месячная абонентская плата, рублей	500 -1200
Лимит на количество переданных данных, в месяц	нет

Это всего лишь один из многих примеров, стоимость может достаточно сильно различаться, как и скорость.

Экономия при пользовании интернетом. Телефоны для консультаций.

Экономия нужна не только потому что всё ещё часто есть месячный лимит. Экономия позволяет быстрее получать нужное – тем более если учесть что если все будут поступать экономично то скорость будет выше. При выделенной линии это не так уж важно, но при нахождении в деревне, то есть использовании мобильных данных, это определённо важно. С наступлением эры связи пятого поколения, то есть 5G, должна существенно увеличиться скорость, и использование мобильных данных тоже.

Лучший источник информации о ваших услугах и стоимости, а также месячном лимите – ваш провайдер. Он формирует набор тарифов, и может их менять. Но при этом – по-честному – он должен оповещать вас о смене тарифа. Старый тариф может действовать много лет, но уже находиться в архиве, и не использоваться для обслуживания новых клиентов. Обратите внимание на тот факт, что обычно тариф таких фирм как Билайн, МТС, Мегафон, Tele2 включает в себя использование интернет. То есть очень возможно, что клиент уже оплачивает эту услугу, но даже не знает об этом, или не знает что тариф можно сменить, или не знает как этим пользоваться.

Заметим, что при получении телефонной SIM-карты лучше брать конверт с документами.

В некоторых салонах связи торгуют иначе – в том числе из лучших побуждений. Продавец вставит в ваш телефон карту, и выбросит конверт с полезными документами – помогая вам. Но документы в конверте надо не только прочесть, но и хранить. Не у всех твёрдая память. Молодёжи достаточно «голой» карты, но она больше знает и так. В общем о пенсионерах никто по-настоящему не заботится. Почему? Недостаточно ответственных людей. Но есть другой ответ, более «гуманный». Дело в том что фирмы связи и салоны связи имеют молодой персонал. Тем люди даже не очень понимают как можно что-то не понимать или не знать. На сегодня бесплатные телефоны основных провайдеров мобильных данных таковы.

Билайн	0611, +78007008378, +74957972727, 8 800 700 0611 (последний номер – для звонков именно по поводу интернета на телефоне)
МТС	0890, 8 800 250 0890
Мегафон	0500, 8800 550-05-00, +7926 111-05-00 (если вы за рубежом)
Tele2	8 800 555 0611, 611

В Мегафон можно написать SMS на номер 0500.

«+7» может быть надо набирать буквально, но обычно это работает просто как «8».

Какие есть меры экономии данных, переданных от вас и к вам в течение месяца?

- Уже иметь при себе то что вы могли бы взять в интернете. Это особенно касается дачников и командированных.
- Искать где то же имеет меньший объём – как говорят, «вес». Сюда же относятся разные конвертеры, но они не конвертируют программы, а конвертируют документы.
- Использовать сжатие данных. Это может быть платной услугой, и иногда это – возможность браузера. Может быть кто-то посчитает меры по сжатию данных лишними. Но загородному пользователю может быть надо взять из интернета более 200 Мб только обновлений программ и ОС, и это для него при медленном канале – реальная трудность. Жизнь показала что есть много людей, которые готовы что-то делать не ожидая вознаграждения. Молодые люди, в особенности математики могут попытаться хотя бы уменьшить проблемы пользователей. Автор создаёт сжимающую программу. Успех этой работы приведёт к её появлению в интернете. Рабочее название программы – BDK или MULTIC. Программа будет бесплатной, и работать в среде Linux или UNIX. Но автор не будет оказывать помощь пользователям программы.
- Настроить браузер для отказа от запроса изображений, проигрывания звука и анимации, видео. Настроить, и вообще выбрать почтовую программу.
- Настроить наличные программы, и даже всю программную среду для минимизации обновлений программ.
- Выбрать лучший тариф – может быть даже у другой фирмы. Телефонный номер можно не менять.
- Уменьшать передаваемые изображения и видео. Избегать передавать их, и звук. Использовать текстовую переписку – не обязательно через SMS, а, скорее, в программах вроде **WhatsApp**.
- Заказывать нужные файлы по обычной почте. Они приходят на лазерных дисках. Почта может работать медленно. Платная услуга. Пересылать большие объёмы данных на лазерных дисках через обычную почту.
- Кооперироваться с соседями, обмениваться программами и прочим.
- Пользоваться протоколом FTP вместо HTTP где возможно.
- Использовать программу **deltacopy** или подобную. Это позволяет существенно сократить количество данных проходящих через ваш канал в интернет.

Как составить список файлов для запроса с другого компьютера или пересылки почтой.

Всё это важно если у вас нет быстрого, или дешёвого, или надёжного канала в интернет. Вы имеете дело с автоматической системой, а не людьми. Не стоит особенно рассчитывать на людей. Очень вероятно, что от вас потребуется конкретный список файлов, которые будут автоматически скопированы из интернета, и записаны на лазерный диск. Составленный

список надо сразу использовать, пока файлы не исчезли по любым причинам.

Запрос в репозиторий.

Рассмотрим достаточно важный вопрос в его классической форме – вам нужны пакеты из репозитория вашей ОС. Это не обязательно основной репозиторий или его копия. Ваш пакетный менеджер имеет список репозитория, и, возможно, PPA. Администратор может менять этот список. Затем вам надо получить сведения о содержимом репозитория. Эти сведения включают в себя адреса пакетов в интернете. Затем вам надо составить список именно тех пакетов, которые вам нужны или будут нужны в будущем, которое вы можете предвидеть. Полученный список является набором адресов.

В случае использования пакетного менеджера АРТ эти действия осуществляются так.

```
nano /etc/apt/sources.list # если надо настроить список репозитория
apt-get update # обновляем данные о репозиториях
apt-get --print-uris ...>packages # пишем в файл адреса нужных файлов
```

То есть вначале администратор – если надо – редактирует файл репозитория с помощью редактора **nano**, затем обновляет базу данных пакетов (но не сами пакеты), а затем создаётся файл с адресами нужных пакетов. Этот файл не содержит тех пакетов которые уже установлены. Вместо многоточия пишутся названия программ, разделённые пробелом. Результирующий файл содержит адреса пакетов. Название этого файла может быть любым, в данном случае оно **packages**.

Также программа **Synaptic** умеет составлять такие списки.

Рассматриваемый вначале способ даст список с лишней информацией. Чтобы её убрать можно использовать **AWK** так.

```
awk '{print $1}' packages>packages2
```

Файл **packages2** содержит лишь адреса, взятые в одинарные кавычки.

Как узнать размер суммы файлов или одного файла.

Это может быть важно, например, потому, что в интернет-кафе вам не дадут скачивать слишком много. Для определения размера файлов можно использовать пакетный менеджер. Он при реальном запросе пакета из интернета скачивает без разрешения лишь очень небольшие пакеты или файлы. Обычно вы сможете запретить получение, и увидите совокупный размер. Но пакетный менеджер работает лишь с репозиториями и PPA.

Кроме того, вы можете использовать программу **curl**. Для этого подайте такую команду.

```
curl -I <адрес файла>
```

, например

```
curl -I https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-9.4.0-amd64-netinst.iso
```

Вы получите даже больше информации чем нужно. Чтобы отфильтровать лишь нужную информацию можно использовать **AWK**. Размер файла указан в байтах так.

```
Content-Length: 305135616
```

То есть этот файл имеет размер около 300 мб.

Запрос в файловое хранилище.

Есть потребность иметь не обязательно репозиторий, а просто набор файлов, достигаемый вам – например, благодаря раздаче с точки доступа у вас на селе. В таком случае лучше всего чтобы среди этих файлов был их список, который можно скачать, и использовать для составления списка нужных файлов. Опять же – это список адресов. Программа будет запрашивать эти файлы, а вам достаточно ждать пока в ваше устройство будут скопированы файлы. Когда их много то не стоит их запрашивать по одному вручную. Список файлов в хранилище должен указывать размер. Также желательно чтобы был файл с контрольными суммами. Это позволит проверять файлы на повреждения. Если раздача делается через HTTP-сервер, то, скорее всего, он формирует страницы с указанием размера файлов. **AWK** или подобная программа может быть необходима чтобы отобрать нужные файлы. Можно использовать индексный файл описывающий файлы в хранилище, причём индексный файл можно брать из репозитория. Таким образом, важно из какого репозитория создано хранилище. Может быть сам владелец хранилища этого не помнит. В таком случае можно организовать примитивный репозиторий одной командой.

`dpkg-scanpackages . |gzip>Packages.gz`

Команда подаётся администратором хранилища в каталоге хранилища, и создаёт индексный файл **Packages.gz**. Этот файл надо сделать доступным для скачивания потребителями. Вопрос о создании репозитория описан в интернете.

Спутниковый интернет.

На начало 2020 года ситуация со спутниковым оборудованием, как мне кажется, изменилась в лучшую сторону по сравнению с тем что было несколько лет назад. На рынке присутствует, например, фирма МТС, и, кажется, что теперь купить технику не слишком дорого для провинциалов.

Однако именно это в общем-то прогрессивное явление может стать проблемой для людей. Не потому что оборудование стало хуже чем раньше. Оно стало доступнее, но люди к этому не готовы. Они как-то жили вообще без выхода в интернет. Теперь, потратив некую – возможно, достаточно большую сумму – они хотят чтобы всё работало. В таком случае лучше организовать общественно-доступный выход в интернет, и оплачивать коллективно. Но люди, привыкшие к обычной технике, оказываются в ситуации ребёнка, брошенного в воду чтобы он сам научился плавать. Жители городов имеют возможность подключить проводную линию. Это проще. Клиенты спутниковой связи будут звонить в свою фирму. Их, разумеется, выслушают, но потребуются уже быть достаточно грамотными как в области аппаратуры, так и программ. Если вы начнёте говорить не обладая даже понятиями для описания ситуации, или заявляя что-то на основе своего или чужого ошибочного мнения, то помочь будет очень трудно.

Автор на сегодня не имеет опыта использования спутникового интернета. Но именно он часто окажется единственным способом выхода в интернет, обеспечивающим достаточную скорость передачи данных в глуши. Рассмотрим некоторые рыночные предложения.

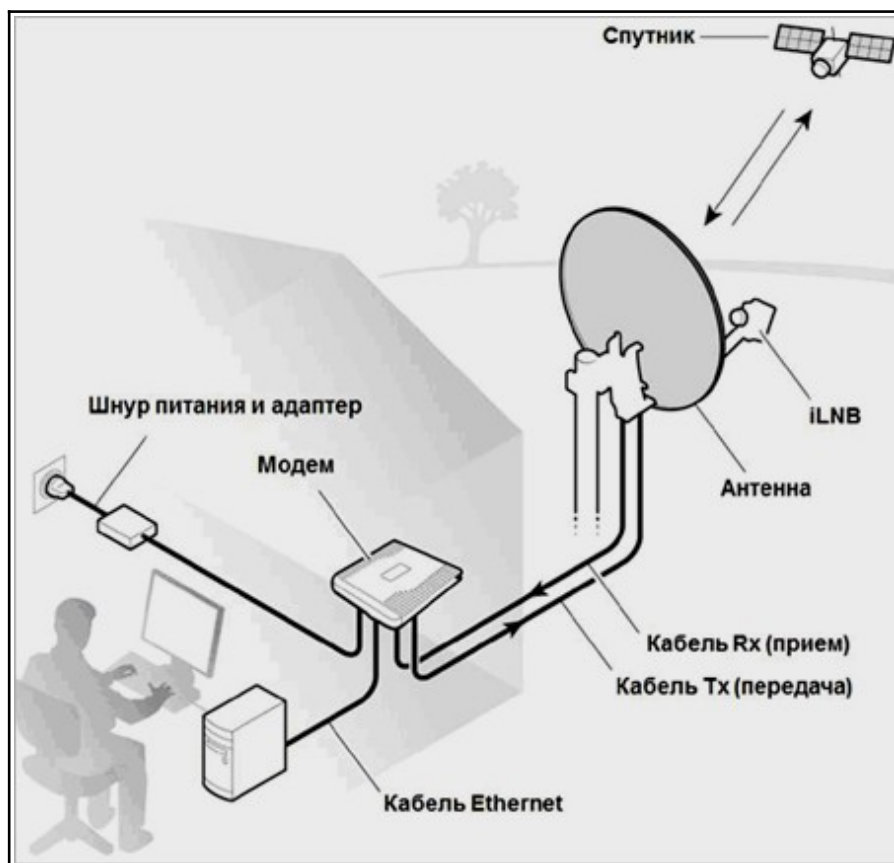
- Спутниковый интернет МТС. Сигнал будет идти почти на всей территории РФ от западных границ до границ Китая. В Карелии сигнал будет не сильный, как и на севере страны. Вы платите от 500 рублей в месяц. Скорость от 200 Мбит. По скорости это городской канал. Инструкцию к оборудованию я не смог найти. Вы можете ставить и настраивать оборудование с помощью мастера. Комплект оборудования от 3000 рублей.
- Спутниковый интернет Триколор. Наверное, самое дорогое предложение. Но оно может быть наиболее качественным и подходящим для коллективного использования. Инструкция на русском размером 96 страниц – в интернете.
- Спутниковый интернет Kitenet. Несколько инструкций на русском. Север страны и Карелия охвачены сигналом. От 1300 рублей в месяц. Комплект оборудования от 8000 рублей.

Рассмотрим оборудование Kitenet. Оно включает в себя следующее.

- Спутниковую «тарелку».
- Конвертер.
- Наушник.
- Устройство Point & play.
- Адаптер питания.
- Модем.
- Кабели, разъёмы и крепёж.

К компьютеру данные идут через провод Ethernet.

Что-то похожее будет у других фирм и в других комплектах. Если вы хотите пользоваться устройством не подключая провод, то нужно организовать беспроводную сеть с помощью роутера или точки доступа.



Глава 33. Телевизионный плеер, интернет-радио, ленты новостей. Чтение публикаций. Мобильный телефон. Телеприставки. Инфосистемы вне городов.

Это – трудная для читателя и автора, но полезная глава. В ней я покажу как можно улучшить информационное обеспечение негородских населённых пунктов. Зачем это надо? В пригородах такой вопрос никто не задаёт. Так, вокруг столицы обширная область обслуживается несколькими фирмами – как поставщиками телефонных услуг, так и другими. В общем смысл выхода в интернет, и информатизации вне городов похож на таковой в городах. Особенно интересна работа на дому. Но есть и особенности, а именно – вне городов мало молодёжи, мало денег, низкие скорости передачи данных, нет нужных магазинов и почтовых отделений, не всегда есть приемлемое электропитание, население не понимает что они могут получить от информатизации. Последний пункт означает что даже при наличии денег люди не заинтересованы что-то менять. Так как средства массовой информации не помогают, то образование является делом самих учеников, или потенциальных пользователей. Такое состояние прямо противоположно нужному для развития страны. Могли бы помочь периодические издания на бумаге, но их мало – если не сказать хуже. Сайт pkprovincia.ucoz.ru в большой степени посвящён инфосистемам вне городов.

Инфосистемы в регионах – в большой степени забота администраторов инфосистем. Иметь такое количество специалистов, тем более в регионах – проблематично. Дилетантизм особенно опасен когда инфосистема связана с

- процессами, которые нельзя останавливать
- процессами, остановка которых особенно дорого обходится
- реагированием на чрезвычайные ситуации
- работой, в которой нельзя допускать ошибки, или их цена очень велика
- передачей секретной информации
- риском от недостатка аппаратных возможностей

- хранением ценной, и фатально-ценной информации

Всё сказанное не должно вызывать ступор, а есть попытка привить ответственное отношение к работе. На практике обычно инфосистемы управляются людьми, чью квалификацию никто не экзаменует. Но это похоже на жонглирование морскими ежами. Такая ситуация связана с низким уровнем понимания, и – в хорошем смысле – тревожности руководства. При этом существует экзамен LPIC, к которому есть тексты на русском в интернете. Также отметим другие материалы интернета, например видеокурсы, которые можно найти и скачать бесплатно.

Библиотеки.

В соответствии с государственной программой все библиотеки обеспечиваются инфосистемами, точнее – чем-то вроде интернет-кафе. Кофе там не дают. В принципе это – большой шаг вперёд. Но он мало касается населения. Даже библиотекари могут быть в растерянности в этой ситуации. Мало иметь технику. Надо иметь литературу.

Кроме того, это добавляет мигрени начальству. Особенно при неорганизованном, безграмотном управлении. Работа библиотек подпадает под регулирование закона – в том числе об авторском праве, и вышестоящих организаций. Об этом приходится писать так как в России люди чуть ли не гордятся игнорированием законов. Но если государство захочет применить закон, то незнание его, как и разговоры о том что «так все делают» ничего не дадут. Так как инфосистемы не являются таким простым явлением как обычная работа библиотеки, то они отнимают много внимания начальства. Вначале надо решить все правовые и организационные вопросы, и лишь потом давать доступ к технике. Можно запросить помощи у министерства культуры, или министерства связи – пусть они отвечают за свои рекомендации и деятельность библиотеки. Они должны обеспечить хотя бы методической брошюрой чтобы библиотекари могли что-то делать. Они же должны были бы обеспечить техобслуживание в виде компьютерного администрирования через интернет.

Автор не работал в сельской библиотеке. Но работал в городской. По-видимому, приход инфосистем не приводит к изменению отношения к работе – оно остаётся таким же как прежде. Компьютер считается чем-то вроде телевизора без «Поля чудес» – если вообще включается, организационные вопросы не решены так как не осознаны, и так далее. Общее наблюдение в том что никто не пытается присмотреться к новым обстоятельствам. А между тем, в отличии от работы с бумажными книгами, в случае инфосистемы воровство книг становится очень возможным. За это отвечает пользователь. Но сами подобные проблемы не осознаны руководством.

Есть и другие важные аспекты. Так, например, может быть низкая скорость интернет-канала. Если есть ограничения на месячный объём переданных данных, то его должно учитывать начальство, и выработать соответствующие меры. Всё ещё в некоторых библиотеках – в том числе из-за проблем с выходом в интернет – есть масса лазерных дисков для читателей. Это, в принципе, хорошо. Но тут есть проблемы: 1) диск легко украсть, 2) постоянная смена дисков быстро изнашивает устройство для их чтения. Вторая проблема без больших трудностей и расходов может быть решена путём создания файлов-образов, но для этого *надо уметь это делать*. Другим, более дорогим решением является устройство для одновременного доступа ко множеству дисков.

Все библиотеки должны иметь компьютерного администратора. В качестве такового не рекомендуется ставить студента – хотя бы потому что ему непонятны занятость и трудность обучения у старших сотрудников. Директор должен быть постоянно и легко достижим для администратора. Общение с администратором может быть очень трудным из-за его неспособности понятно объяснить что он делает, и так далее. Такая ситуация может быть и из-за директора. Это может быть не слишком умный человек, привыкший решать все проблемы исключительно путём выговоров, а тут надо понимать какую-то инфосистему. Администратор может «не видеть за деревьями леса». Директор должен ставить перед ним задачи в русле деятельности библиотеки. Администратор должен именно так понимать свою работу.

В библиотеке может потребоваться принтер. На нём, в частности, можно печатать книги для выдачи читателям. Скорее всего, этот принтер будет устройством для печати на листах типа А4. Единственным типом бумаги, который без проблем используется на всех таких устройствах, является *офисная* бумага. Она сравнительно дорога и толста, и продаётся в пачках по 500 листов. Первые опыты печати лучше делать с такой бумагой. Вопрос о другой бумаге

остаётся открытым. Его лучше всего решать чтением спецификации принтера, или запросом к его производителю или представителю производителя. Каждая библиотека должна знать способы удешевления печати.

Условием работы читателя должно быть выполнение правил, в том числе следующих:

- читатель не может нарушать закон
- читатель не может заниматься настройкой программной среды – в частности, почтовой программы¹¹
- читатель не может без разрешения начинать работу и сам выбирать место
- читатель не может сам посылать задания на принтер
- читатель не может ставить, или удалять программы, как и игры. Тем не менее использование сетевых программ¹², как и арендованных¹³, и арендованной вычислительной мощности – доступной через интернет – не запрещено

Также, во избежание скандалов, библиотека может отказаться от обязательств по хранению или конфиденциальности читательских файлов. Сама возможность их хранения полезна. Заметим, что в среде UNIX (да и других ОС) надо избегать ситуации, когда файлы одного читателя доступны для чтения другим читателем.

Каждая библиотека должна не просто иметь оборудование, а понимать, что ей надо, и что не всегда это оборудование должно быть свежим. При этом надо понимать что носители изнашиваются.

Трудной темой является операционная система, или системы, используемые в библиотеке. Вопрос лицензирования – это ещё не все заботы. Так, например, может потребоваться чтение файлов с компьютера с другой ОС через библиотечную локальную сеть. Стоит обратить внимание на услугу **Яндекс DNS**.

Можно подумать о возможностях библиотек в области информационных услуг в современных условиях. Эти условия состоят в том что читатели не разбираются в вопросах информатики, не хватает денег, и так далее. Такие ограничения сильно влияют на работу библиотеки, делая её не похожей на таковую в городе – тем более крупном. Как можно всё же применить с пользой новые технологии? В качестве наброска можно ответить следующее:

- уйти от каталожных карточек к базе данных книг. Это не отменяет наличие карточек, но позволяет искать книги гораздо быстрее
- печатать книги или микрофильмы – не нарушая авторские права. Для печати надо выбирать принтер в основном смотря на стоимость печати, а не оборудования. Так можно сформировать довольно большой фонд книг
- печатать публикации из интернета. Они могут быть большими, а иногда может быть оправдано создавать сборники статей
- использовать микрофильмы. Оборудование для их чтения должно быть дешевле компьютера
- выдавать читателям аудиокниги, или даже файлы – в том числе с фильмами – не нарушая закон. Библиотека сама может создавать аудиокниги.
- держать в библиотеке энциклопедии и словари, и вообще публикации в виде выводимых на экран, а не бумажных изданий
- использовать компьютерные устройства для обучения информатике, а также небиблотечных функций

У библиотеки, разумеется, может быть банк лазерных дисков, но все подобные ресурсы могут потребовать своей каталогизации.

Микрофильмы и микрофиши хороши тем что с ними просто работать, и оборудование может быть не слишком дорогим. Доставка микрофиш и микрофильмов многократно легче доставки книг. Аппараты для их чтения – по сути проекторы. Вот пример такого аппарата.

11 Тут имеется в виду именно почтовая программа, а не почтовый ящик, доступный через браузер

12 См в главе о программах

13 На сайте rollapp, и так далее



Каждый директор библиотеки должен понимать, что не только РГБ, но и любая библиотека может столкнуться с недостатком места под литературу. Крупнейшая библиотека СССР, «Ленинка» в Москве – даже заслуживая большей заботы чем другие библиотеки, столкнулось с этой проблемой. Количество авторов, и единиц хранения превысило её возможности. Оцифровка фондов стала главной заботой, и новым делом в работе многих людей. Оцифровка совершенно меняет выдачу литературы. Она мотивирует вывод страниц на экраны, и, таким образом, сильно меняет всю работу библиотеки. Компьютерный администратор стал чуть ли не главным работником, оттеснив на второй план профессиональных библиотекарей.

Автор не имеет образования библиотекаря.

Примерно та же ситуация с информатизацией других библиотек.

И в том, и в другом случае создаваемые инфосистемы лишь кажутся простыми на первый взгляд. Более внимательное их рассмотрение покажет новые детали.

Автор, не имея опыта именно библиотечной работы, застал весьма спорное решение вопроса хранения снимков страниц в РГБ. Но, возможно, ещё большей проблемой является то, что директор РГБ, по-видимому, не достаточно уделил внимания этому вопросу. По этому поводу есть хорошая поговорка – «умный человек два раза на грабли не наступает». Все библиотеки – во всех странах – заинтересованы в экспертном решении вопроса хранения своих данных. Надо иметь в виду свежие форматы изображений, например ВРG.

Так как в библиотеках создаются компьютерные классы, то стоит рассмотреть использование тонких клиентов, и **Clonezilla**. Тонкий клиент – это «компьютер», *достаточный* для пользователя компьютерного класса, но не равный полноценному рабочему месту. Он требует наличия центрального, серверного компьютера в той же локальной сети. Сервер в данном случае выполняет вычислительную работу за клиентское устройство. Зачем это может быть надо? Тут есть как минимум два мотива

- уменьшение стоимости класса
- упрощение администрирования

Задача **Clonezilla** несколько другая. Это программа, позволяющая централизованно и быстро распространить на много компьютеров настроенную рабочую среду, и скопировать её же на центральный компьютер на случай её повреждения.

Инфосистемы на селе.

Простейшая инфосистема на селе состоит из развитого мобильного телефона. О примитивных телефонах так сказать можно лишь с натяжкой. И то, и другое не является общественно-доступной техникой. Обычно под инфосистемой будет пониматься компьютерная система. Она может включать и другие устройства.

Разумеется, что в условиях безграмотности не только создание, но и использование инфосистем под вопросом. Трудность для их описания в том что они могут быть самыми разными. В принципе сельская инфосистема должна строиться после понимания того какие её функции будут реально использованы – в конкретных условиях. Само наличие функций ещё не означает их использование. Пример этого – игнорирование функций мобильных телефонов. Люди просто не думают о том что они теряют возможности. Известно, что «там много всякого». А что это и чем полезно – никого из старшего поколения не интересует. В ситуации апофеоза невежества пользователи делают бессмысленные поездки, ломают голову или полагаются на догадки, и так далее. Всего этого могло бы не быть. Когда они начинают изучать свой телефон, то оказывается, что они неспособны понять его работу. Это так из-за отсутствия образования. Причём образование должно быть направлено не столько на знание сколько на понимание. Старшему поколению для обучения нужен стимул. А для знания стимула нужно образование. Поэтому ничего не происходит. Надежда – на молодёжь, или тех, кто был в городах.

Создавать инфосистемы за городом особенно трудно. Хотя бы потому что там часто плохо с каналом интернета. Частное лицо может обеспечить этот канал, но он может потребоваться для других. Абонентская плата и оборудование могут стоить дорого, а коллективная оплата означает коллективное использование. Дополнительная трудность – в непредсказуемости передачи сигнала сети Wi-Fi в условиях помех – деревьев, домов. Это верно и для так называемого моста, то есть приёмопередатчика с узким лучом. Он может работать на расстояние в несколько километров, но без помех.

Одна из интересных идей – в создании локальной сети на селе. Интерес в основном в том чтобы таким образом хотя бы отчасти заменить интернет. Если в простом, «классическом» случае даже соседи общаются через интернет, то так это не нужно. Локальная сеть может оказаться гораздо более быстрой, и скорость, как ожидается, будет меньше колебаться. Так, например, автор, находясь под Москвой, имеет скорость от нуля до 9 мегабит. А скорость в локальной сети, ориентировочно, составит 5-20 мегабит – даже с дешёвым роутером или точкой доступа. В качестве пользовательских устройств подойдут даже мобильные телефоны с поддержкой Wi-Fi, но это – не все требования. Далее мы имеем разнообразные требования, обусловленные спецификой конкретного села. Наверное, самой нужной функцией будет бесплатное речевое общение. Но оно предполагает возможность такого использования *каждого* телефона в сети. То же верно про другие функции. Для реализации этих задач важны реальная работа программ на телефоне. Заявленная поддержка Java¹⁴ на телефоне может быть фиктивной. В таких условиях важно уметь проверить телефон при покупке – насколько это вообще возможно.

Устройства, отличные от телефона, выглядят более предсказуемыми в своих реальных перспективах работы. Нас интересуют устройства с возможностью работы в сети, и выполнения нужных функций – одновременно. Возможно, стоит протянуть проводной канал для улучшения распространения сигнала, а на его концах поставить по роутеру – рассматривая село, или его часть, в виде растянутой области работы сети. Но тут может быть известная проблема настройки сети, и её нельзя избежать.

Следующим по стоимости после телефона устройством является роутер, но он требует прошивки, и труден в использовании. Гораздо более перспективен смартфон или планшет. Также стоит обратить внимание на микрокомпьютеры, и на б\у компьютеры. Даже старый, 10-летней давности компьютер интересен. Он может стоить менее 2000 рублей. Среди микрокомпьютеров особенно интересен Raspberry Pi B. Он подключается к телевизору через разъёмы RCA.

14 Теоретически обеспечивающая возможность работы многих телефонных программ



Никогда не хватайте что попало, а прежде продумайте всю систему.

Проблема в том что обычный пользователь понятия не имеет как строить локальную сеть. Чтобы улучшить ситуацию используйте симулятор локальных сетей, это программа.

Сервис VPS/VDS.

На селе представляет интерес использование аренды вычислительной мощности в городском сервисе VPS\VDS. Зачем? Это позволит использовать компьютерные программы, требующие сравнительно много оперативной памяти. Это позволит использовать быстрый канал в интернет – на этом сервисе, не у вас в деревне. Вычислительная мощность на таком сервисе в общем намного больше чем у старого микрокомпьютера. Характеристики «компьютера» на сервисе можно менять. Иногда можно там использовать выбранную вами произвольную ОС. Этот выбор, конечно, ограничен свойствами сервиса и выданными ресурсами. Мне кажется, что хорошим выбором будет Antix, Slitaz, NetBSD.

Сервис VPS\VDS позволяет вам

- получить условный компьютер без больших потерь денег
- возможность регулировать выделяемые вам ресурсы – в том числе аппаратные. Это виртуальные, не реальные ресурсы.
- в любой момент прекратить использовать сервис
- выключать условный компьютер когда он не нужен, но абонентская плата от этого обычно не зависит
- быть полным хозяином поставленной в условном компьютере ОС и программ
- получить возможность делать и использовать копии ваших данных или копии всей программной среды и данных. Это позволяет вам не всегда беспокоиться о ваших ошибках. Очень вероятно что «откат» к прошлому состоянию условного компьютера делается просто, и даже не требуется куда-то звонить для этого
- хранить личную и секретную информацию – исходя из того что никто не станет пытаться или не сможет её узнать. Степень секретности может быть разной
- использовать не везде возможный или легко достигаемый протокол IPv6
- организовать постоянно работающий сервис – без опасности потери питания или лишнего шума

С другой стороны,

- сервис не позволит вам – обычным образом – подключить к условному компьютеру устройства и носители. Вообще это трудно или невозможно
- сервис не позволит вам иметь произвольное оборудование. Так, вы не сможете использовать произвольную – в частности, иногда необходимую – видеоплату. Скорее всего это будет не нужно

Прежде всего в быту нужно подключать носители. На реальном оборудовании это обычно означает подключение флэш-брелка, или карты памяти, или вставление лазерного диска. С арендованной мощностью всё намного сложнее. Носитель может быть загрузочным¹⁵. В таком случае может помочь его *образ*, то есть файл образа. Если носитель не загрузочный, то его можно рассматривать просто как набор файлов и каталогов на некой файловой системе. Прикладная проблема в том что даже если есть возможность передать некий файл или каталог в условный компьютер, то наличный интернет-канал и тариф делают это дорогим или трудным из-за обрывов связи и низкой скорости передачи данных. Таким образом, задача в том как передать много данных или программ. Достаточно тупой способ в том чтобы послать носитель по обычной почте на сервис виртуализации – убедившись что там его подключат.

15 Не путать с тем что называется autoplay

Более умный способ в том чтобы создать или сделать доступными те же данные или программы для условного компьютера. Очень многое из наличного на носителях уже есть в интернете. Но не всё. Можно использовать интернет-кафе. Не злоупотребляйте выходом в интернет на работе. Это может вызвать гнев начальства. Заметим, что по крайней мере образ в формате ISO можно подключать без устройства для чтения носителя – обычно описывают подключение в среде UNIX. То же может быть верно про формат IMG. Вообще говоря, вы можете конвертировать образы из одного формата в другой.

Программа **NoMachine** позволяет использовать удалённый принтер, но сами VPS\VDS не подключают принтеры – технику, не виртуальный принтер – к условным компьютерам. Эта же программа позволяет задействовать доступные через разъём USB устройства – но все ли? По крайней мере это флэш-брелки.

Вопрос об использовании таких устройств как подключаемые кодировщики или звуковые платы – потенциально сложный, без больших надежд на успех.

Обратите внимание на то что негородские пользователи могут работать с рабочим столом ОС, установленной на сервисе. Это – реальный способ получить быстрый канал в интернет для вашего рабочего места, но при очень низкой скорости передачи данных с сервиса к вашей технике изображение рабочего стола будет застывать. Это не зависание программ. При обрыве связи надо снова войти в персональное обслуживание сервисом, использовать его и завершить работу тамошней системы. Достаточная скорость прорисовки картинку на медленных каналах существует благодаря деградации изображения с помощью упрощения цветопередачи. Поэтому может быть нужно настроить программы – сервер, подающий экранную картинку (например, VNC) или клиент; это означает что работы с качественным цветом сделать будет нельзя. Таким образом даже на слабом компьютере или носимом устройстве можно работать при плохом канале в интернет. Обновления разного рода и большие программы, массивные файлы больше не будут проблемой. Данные, передаваемые для построения экранного изображения, надо шифровать – особенно если вы заняты платной работой.

Ещё о локальной сети.

Сложности настройки локальной сети обусловлены тем что она не обслуживается никаким провайдером. Провайдер отвечает лишь за передачу данных клиенту – если у вас не коллективное подключение. Если вы приглашаете настройщика, то важно всегда любым способом записать все характеристики локальной сети и использованных программ так как смена ОС и другие перемены могут сбить настройку. Сейчас за деньги можно сделать всё. Но какова будет стоимость выезда за 300 км?

Какие могут быть функции, или услуги локальной сети? На первом приближении – те же что у интернета. Но это заявление может вызвать усмешку у специалистов. То что вы получаете благодаря интернету и через него – результат работы многих фирм, с огромным оборотом капитала, парком техники, штатом специалистов, и так далее.

Тем не менее можно назвать в качестве примера поселковый форум. Он, скорее, будет голосовым – из-за практически отсутствия возможности набора текста. Классический форум – текстовый, его содержимое быстрее передаётся, и возможен поиск по тексту. Вставка иллюстраций возможна. В любом случае возникают как минимум два вопроса: какая программа обеспечит работу форума, и как хранить его содержимое. Наличие и удобство использования такой программы на развитом телефоне у меня вызывает сомнение. Использование микро- или обычного компьютера для этой цели многократно описано. Так как работа голосового форума требует передачи звукозаписи на сервер, то должна быть программа для этой передачи. Для обеспечения поиска по, возможно, большому количеству звукозаписей, может потребоваться распознавание речи. Оно может потребовать не микро-, а настоящего компьютера – если вы найдёте нужную для этого программу.

Другая, более далёкая от обычных представлений идея – доступ к поселковому серверу с телефона. Это могло бы быть, например, для чтения почты, или запуска консольных программ.

Также можно организовать хранилище файлов. Это могли бы быть фото, сделанные телефонами. Для обслуживания именно хранилища не требуется мощный компьютер. Один компьютер может выполнять более одной роли в сети одновременно.

Когда говорят что компьютер является сервером, то это значит, что он – благодаря как минимум одной программе – является обслуживающим в области какой-то функции

относительно клиентов – часто благодаря клиентским программам, и клиентским устройствам. Под клиентами понимаются люди, а клиентской может быть программа, которая сама по себе не является таковой, но является клиентской в схеме (или, как говорят, архитектуре) определённого процесса обслуживания клиентов. Сервер предоставляет услуги, а клиентская программа позволяет получать услуги. Такая схема может показаться сложной. Почему она используется? Хотя бы потому что сервером пользуется более одного клиента. С точки зрения администратора иметь один сервер лучше чем его аналоги на каждом компьютере так как настраивать один компьютер проще. Обратной стороной такой ситуации является повышенная вычислительная нагрузка на сервер. Поэтому в целом это компьютер, а не носимое устройство. К тому же сама технология клиент-серверной архитектуры в большой степени пришла из состоятельных стран, где компьютер относительно дешёв. Именно использование компьютера в качестве сервера наиболее часто и полно описано.

В 90-е годы достаточно часто настройка сервера выполнялась вручную. Это не всегда страшное дело. Тем не менее надо знать, что подобные работы сейчас часто заменены на применение готовых рабочих сред – но это не среды пользовательской работы, а среды работы сервиса. Сервис – это то, что предоставляет услугу, или услуги. Обсуждение этой темы заодно затрагивает вопрос об использовании арендованной вычислительной мощности так как в обоих случаях мы сталкиваемся с *виртуализацией*. Сервис может её не использовать.

Почему важна виртуализация? И что это? Примерный ответ таков: это использование отдельных рабочих сред, при том, что происходящий в них процесс или процессы используют виртуальные (а не реальные) аппаратные ресурсы, и не знают что они – эти ресурсы – не реальны. Другими словами, создаётся некая «песочница» для некоего процесса, или процессов, которая отделена от оборудования, а вместо него использует изображаемые ОС устройства. Такой подход позволяет

- игнорировать особенности реального оборудования, и свести его многообразие к одному виртуальному устройству для каждого типа устройств
- благодаря этому простым образом привести в рабочее состояние находящуюся в «песочнице» среду

На сервисе VPS\VDS в качестве такой среды выступает ОС, управляемая клиентом, то есть администратором поселковой инфосистемы. На компьютере этой системы – в том числе там же – в качестве среды сервиса выступает либо ОС, либо контейнер.

Контейнеры.

Рассмотрим для примера популярные контейнеры, управляемые программой **docker**. Что это? Контейнеры – это подготовленные не обязательно пользователем (в качестве которого выступает администратор) экземпляры программ в их конкретном назначении, с параметрами их работы. Параметры работы, как и нужные дополнительные пакеты не стоит лишний раз устанавливать. Поэтому есть интерес использовать именно контейнеры. Контейнеры изначально были задуманы как не обладающие оконной средой. Это часто и не нужно. Но это не принципиальное ограничение, и его можно обойти. Программа **docker** была создана специально для несложного управления «песочницами», и стала очень популярной. Так как контейнер использует не весь набор ресурсов, то процессы в нём в общем идут быстрее, чем в виртуальной машине на VPS\VDS. В последнем случае у пользователя может быть больше возможностей, и используется целая ОС, которая, как правило, хорошо описана.

Всё сказанное не означает что вы всегда сможете просто поставить контейнер, и запустить его. Программисты не очень учитывают невежество и нетерпение потребителя. Надо быть готовым к ручной работе. Инструкция к конкретной работе будет в интернете, и она может исходить из того что вы умеете искать нужные вам объясняющие статьи, и обладаете грамотностью, владеете некой терминологией.

Голосовое общение через Wi-Fi локальную сеть. WhatsApp, Viber, ShareIt на селе.

Голосовое общение – как при разговоре через квартирный телефон или советский уличный телефон-автомат, и все три эти программы работают и при использовании интернета. Но нас интересует именно разговор и передача данных (с помощью ShareIt) в локальной сети. Это могла бы быть Wi-Fi-сеть многоквартирного дома, или Wi-Fi-сеть села. Использование местной сети не затрагивает интернет; возможно, лишь для создания учётной записи или начала связи устройств нужен интернет. Поэтому медленный канал связи с интернетом, и

лимит на него не важны. WhatsApp это крайне популярная программа, используемая в том числе для уменьшения расходов по описанной причине. Основное назначение программы – голосовое общение. Viber также используется для голосового общения. В обеих программах прежде всего заинтересованы пенсионеры, хотя фактически ими пользуются очень многие, в том числе молодые люди.

Википедия говорит что ShareIt это «бесплатное приложение для передачи файлов между устройствами, поддерживающими Wi-Fi протокол. Пользователи могут использовать его для передачи файлов любого типа, включая фотографии, видео, музыку, контакты, приложения и т. д. Программа разработана одноимённой компанией SHAREit Technologies Co. Ltd и доступна на 45 языках, включая русский».

Все три программы бесплатны и рассчитаны на ОС Android. Точнее так.

Программа	Версия программы	Версия Android
WhatsApp	2.21.7.14	4.1 и выше
Viber	Зависит от устройства	Зависит от устройства
ShareIt	5.9.48_ww	4.1 и выше

Рассмотрим программы подробнее.

Программа	Описание
ShareIt	<p>Передача осуществляется при помощи создания точки доступа Wi-Fi на одном из устройств, что и позволяет отправлять файлы на большой скорости. В былое время подобной технологии еще не существовало, поэтому передача происходила при помощи USB-накопителя и Bluetooth. Использование этих методов было неудовлетворительно. Через USB заражались устройства, а через Bluetooth процесс был очень долгим. Компания «Lenovo» этим воспользовалась и сделав SHAREit, вышла лидером на рынке.</p> <p>Возможности</p> <p>Использование SHAREit обеспечивает лёгкую и быструю отправку, но присутствуют другие функции и «фишки»:</p> <ul style="list-style-type: none"> • обмен между ПК и смартфонами; • ненужность в применении интернет-связи; • отправка различного содержимого. От музыки до контактов на вашем смартфоне; • отсутствие ограничения передаваемого размера; • простота в использовании; • возможность переслать установщик клиента через Bluetooth; • просмотр содержимого компьютера со смартфона, используя функцию «Remote View»; • передача уже установленных приложений. (по тексту сайта shareit-rus.ru)
Viber	<p>Приложение-мессенджер, которое позволяет отправлять сообщения, совершать видео- и голосовые VoIP-звонки через интернет. Голосовые вызовы между пользователями с установленным Viber бесплатны (оплачивается только интернет-трафик по тарифу оператора связи). Viber имеет возможность отправлять текстовые, голосовые и видеосообщения, документы, изображения, видеозаписи (по тексту Википедии)</p>
WhatsApp	<p>Популярная бесплатная система мгновенного обмена текстовыми сообщениями для мобильных и иных платформ с поддержкой голосовой связи и видеосвязи. Позволяет пересылать текстовые сообщения, изображения, видео, аудио, электронные документы (по тексту Википедии)</p>

Viber.

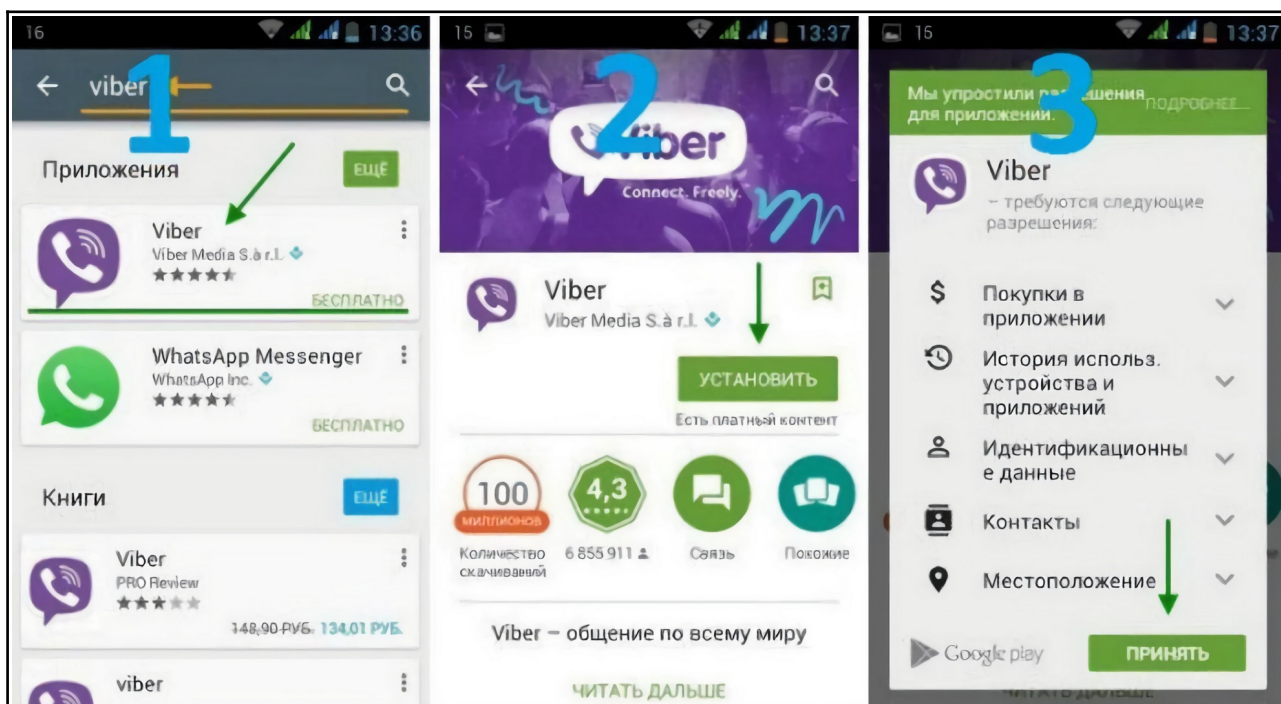
Рассмотрим Viber так как он создаётся в Белоруссии.

Инструкция установки программы «для чайников»:

1. Зайти в магазин приложений (Play Market, Apple Store и др.). Для этого нажмите значок Play Market, он есть на каждом смартфоне.
 2. Найти Viber через строку поиска и щёлкнуть на его значок.
 3. Нажать кнопку «Установить».
 4. Программа выдаст окно со списком разрешений. Выбрать «Принять».
 5. Установка начнется автоматически. После ее завершения нажать кнопку «Открыть».
- Значок Play Market выглядит так.



Теперь наглядно – смотрите картинки.

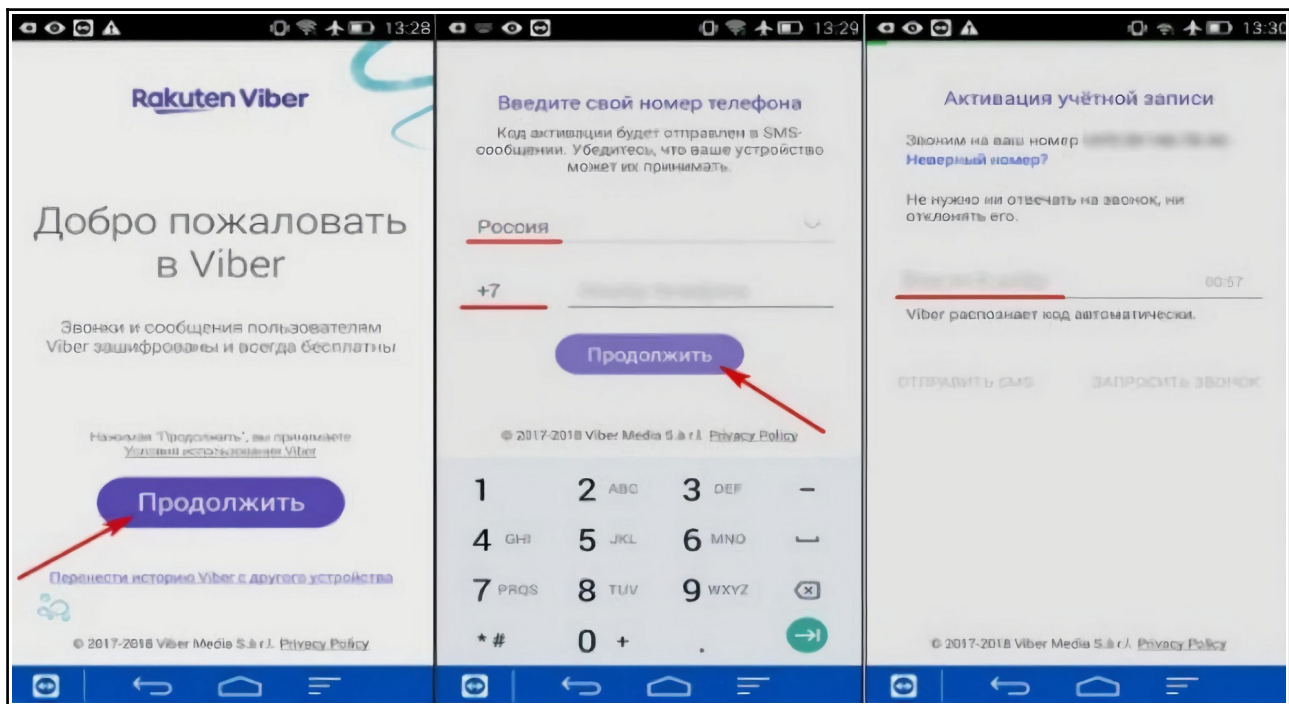


Следующим этапом после установки мессенджера является регистрация. Она предполагает создание нового аккаунта, привязанного к действующему номеру мобильного телефона, и совершается в течение нескольких минут.

Чтобы зарегистрироваться, выполняют такие действия:

1. После первого открытия программы на экране телефона появится приветственное окно. Нажать в нем кнопку «Продолжить».
2. Указать свою геопозицию, выбрав нужную страну из представленного списка.
3. Ввести действующий номер мобильного телефона. Нажать кнопку «Продолжить».
4. Откроется окно для ввода числового кода активации. Бот отправит его на указанный номер в виде SMS.
5. Ввести код, нажать «Готово». Регистрация завершена.

Теперь более наглядно – смотрите картинки.



Отправка сообщений.

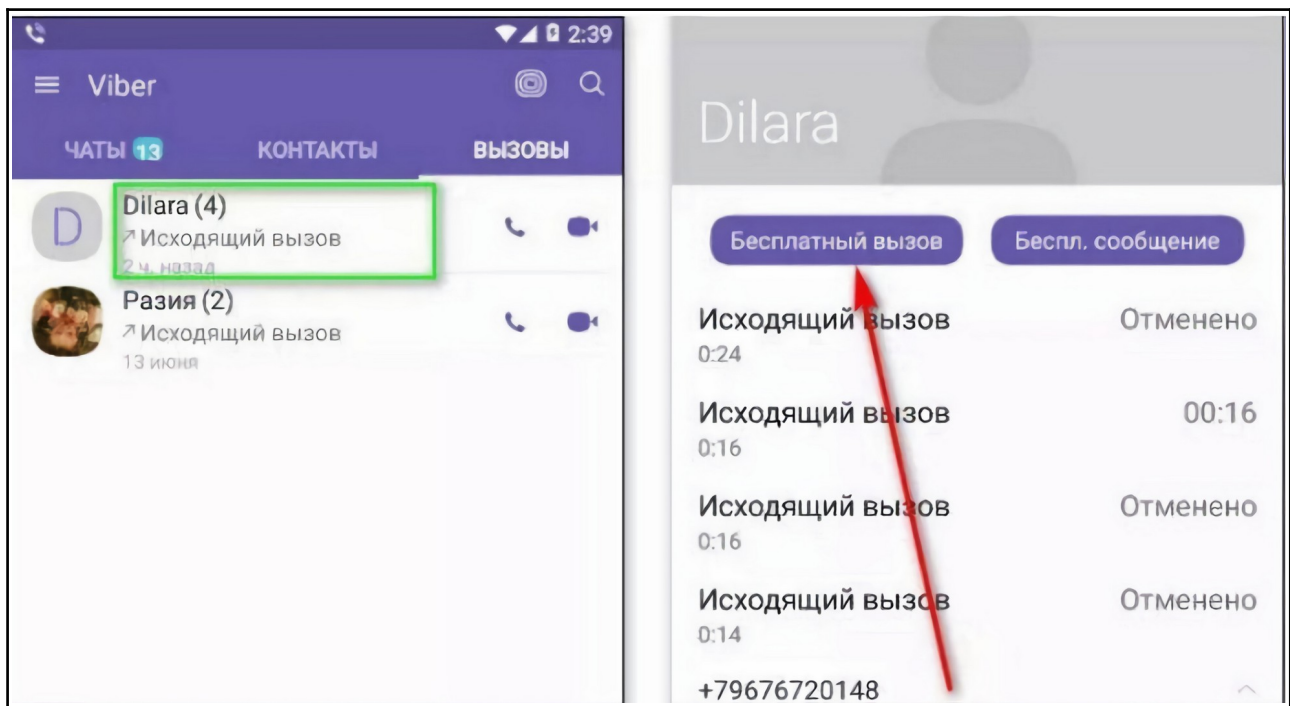
- Открыть закладку «Чаты». Выбрать из списка собеседника или создать новый чат с помощью фиолетовой кнопки с пиктограммой диалога.
- Ввести в расположенной внизу окна строке текст послания. Нажать на фиолетовую кнопку «Отправить».
- Для удаления отправленного сообщения нужно зажать его на пару секунд, а потом выбрать в появившемся меню кнопку «Удалить».

Бесплатные звонки.

Приложение Viber дает своим пользователям возможность совершать бесплатные аудио- и видеозвонки при условии наличия подключения к интернету.

Чтобы совершить аудиозвонок, нужно:

1. Открыть профиль собеседника в списке контактов. Нажать на кнопку «Бесплатный звонок». Можно также зайти в чат с этим человеком и нажать в нем на пиктограмму в виде телефонной трубки.
 2. На экране появится надпись «Вызов». Она означает, что устанавливается соединение с абонентом. Когда он возьмет трубку, можно начинать разговор.
 3. Для завершения беседы нажать на красную кнопку с телефонной трубкой.
- Теперь более наглядно – смотрите картинки.



(по тексту сайта vibir.ru)

Теперь рассмотрим ShareIt.

Программа позволяет автоматический поиск других устройств с SHAREit, находящихся в зоне видимости внутри созданной Wi-Fi-сети.

При передаче файлов Между смартфонами:

- Скачиваем SHAREit (получение и установка – аналогично Viber)
- При включении приложения будет предложено настроить профиль. Указываем псевдоним и свою картинку – аватар, который будет виден при нахождении устройства.
- На смартфоне, который будет получать отправляемые файлы, нажимаем «Получить», а на другом гаджете выбираем файлы для отправки, нажав на «Отправить»
- После выбора передаваемого содержимого и сопряжения устройств, начнётся обмен.

Чтобы выполнить сопряжение необходимо, чтобы наши устройства были подключены к одной точке доступа Wi-Fi.

- Запустив на обоих устройствах SHAREit, нужно выбрать «поиск доступных устройств».
- Программа быстро найдёт другие подключенные к данной точке доступа устройства, а для прямого соединения с необходимым устройством достаточно будет отправить ему запрос для подключения, который необходимо подтвердить.
- Соединение будет установлено, и вы можете отправлять файлы между устройствами с внушительной скоростью.

(по текстам сайтов shareit-rus.ru и droidov.com)

Заметим, что WhatsApp также существует на телефонах производства Nokia с ОС KaiOS. Однако пока эта ОС не может соревноваться с Android количеством программ. Телефоны с KaiOS меньше и легче чем смартфоны, но и экран меньше. Во-вторых, KaiOS пока, насколько я знаю, не снабжена антивирусной программой; также старые телефоны с поддержкой Java не имели такой программы, и выпускались много лет.

Сообщения и звонки без интернета.

Проблема плохих каналов связи, или вовсе отсутствия доступа в интернет с запозданием создала программы, которые могут потребовать на собрании жильцов села обменяться чем-то вроде кодов чтобы общаться в границах населённого пункта; расстояние между смартфонами должно быть не более 100 метров. Используется сеть Wi-Fi. Роутер и администратор не нужны. Приведу тексты интернета.

Миллионы людей используют Signal каждый день, чтобы бесплатно и моментально обмениваться сообщениями в любой точке мира. Отправляйте и получайте сообщения, участвуйте в голосовых и видеозвонках высокого качества и изучайте всё новые и новые функции, которые помогают вам оставаться на связи. Передовая технология обеспечения конфиденциальности Signal всегда включена, поэтому вы можете сосредоточиться на том, чтобы делиться важными моментами с людьми, которые важны для вас.

- Говорите что угодно — новейшая технология сквозного шифрования (основанная на протоколе Signal Protocol™ с открытым исходным кодом) надёжно защищает ваши разговоры. Приватность здесь не дополнительная функция — Signal просто так работает. При каждом сообщении, при каждом звонке, каждый раз.

- Ощутите скорость — сообщения доставляются быстро и бесперебойно даже при медленном соединении. Signal разработан так, чтобы отлично работать даже в самых сложных условиях.

- Почувствуйте свободу — Signal является полностью независимым некоммерческим продуктом. Разработка поддерживается нашими пользователями. Без рекламы. Без отслеживания. Без обмана.

- Будьте собой — вы можете использовать свой действующий номер телефона и адресную книгу для защищённого общения с друзьями.

- Говорите и слушайте — живут ли ваши друзья и родственники за городом или за океаном, благодаря улучшенному качеству звука и видео в Signal вы почувствуете себя рядом.

- Шепчитесь в тени — переключитесь на тёмную тему, если вас не привлекает светлая сторона.

- Найдите свой звук — выберите разные звуки уведомлений для каждого контакта или полностью отключите их. Если предпочитаете звуки тишины, просто выберите "Без звука" в качестве мелодии звонка.

- Творите — используйте встроенные функции редактирования изображений, чтобы рисовать, обрезать и вращать фотографии перед отправкой. Хотя фотография стоит тысячи слов, вы сможете добавить ещё несколько при помощи инструмента "Текст".

Требуемая версия Android – 4.4 и выше.

Briar - это приложение для обмена сообщениями, предназначенное для активистов, журналистов и всех, кому нужен безопасный, простой и надёжный способ общения. В отличие от традиционных приложений для обмена сообщениями, Briar не зависит от центрального сервера - сообщения синхронизируются непосредственно между устройствами пользователей. В случае отсутствия интернета Briar может синхронизироваться через Bluetooth или Wi-Fi, обеспечивая передачу информации в условиях чрезвычайной ситуации. Если интернет работает, Briar может синхронизироваться через сеть Tor, защищая пользователей и их связи от слежки.

В приложении имеется возможность обмена сообщениями, группы и форумы, а также блоги. В приложение встроена поддержка сети Tor. Все, что вы делаете в Briar, хранится только на вашем устройстве, если конечно вы не решите поделиться этим с другими пользователями.

В приложении нет рекламы и отслеживания. Его исходный код полностью открыт для всех и уже прошел профессиональный аудит. Все релизы Briar воспроизводимы, что позволяет убедиться в том, что опубликованный исходный код в точности соответствует опубликованному здесь приложению. Разработкой занимается небольшая некоммерческая команда.

Требуемая версия Android – 4.1 и выше.

Рассмотрим эту программу подробнее.

При первом открытии Briar вам будет предложено создать учетную запись. Вы можете выбрать любой никнейм (имя) и пароль. Пароль должен состоять не менее чем из 8 символов и его должно быть трудно угадать.

Внимание: ваша учетная запись Briar надёжно хранится на вашем устройстве, а не в облаке. Если вы удалите Briar или забудете свой пароль, восстановить учетную запись будет невозможно.

Нажмите «Создать учетную запись». Когда ваша учетная запись будет создана, вы попадете в список контактов.

Briar Setup

Your Briar account is stored encrypted on your device, not in the cloud. If you uninstall Briar or forget your password, there's no way to recover your account and your data.

Choose your nickname

Alice

Choose your password

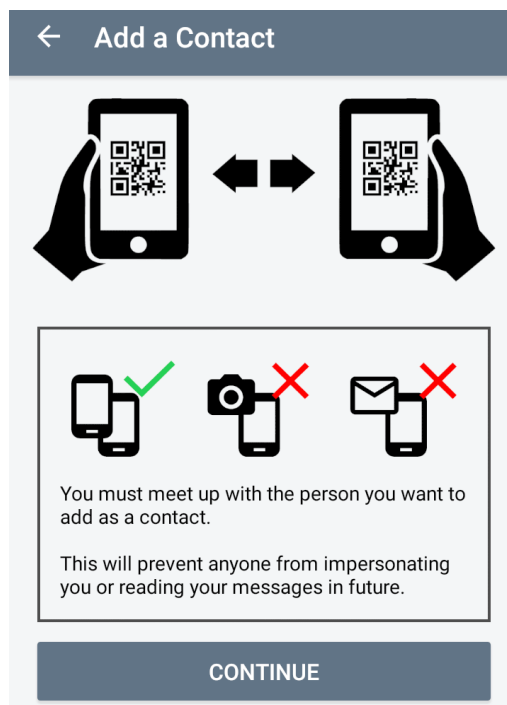
.....

Confirm your password

.....|

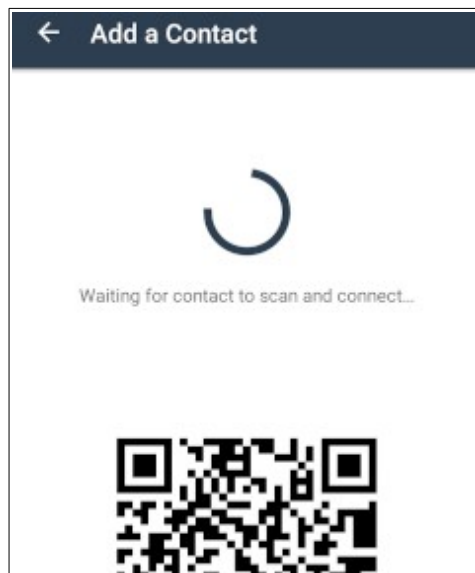
CREATE ACCOUNT

Можно добавить контакт на расстоянии. Другой способ добавить контакт — встретиться с человеком, которого вы хотите добавить. Каждый из вас будет сканировать QR-код с экрана другого человека. Это гарантирует, что вы подключаетесь к нужному человеку, поэтому никто другой не сможет выдать себя за вас или прочитать ваши сообщения.



Выровняйте QR-код вашего контакта в видоискателе. Возможно, вам придется подождать несколько секунд, пока камера сфокусируется.

Когда камера отсканирует QR-код, вы увидите сообщение «Ожидание контакта для сканирования и подключения». Теперь ваш контакт должен отсканировать ваш QR-код.



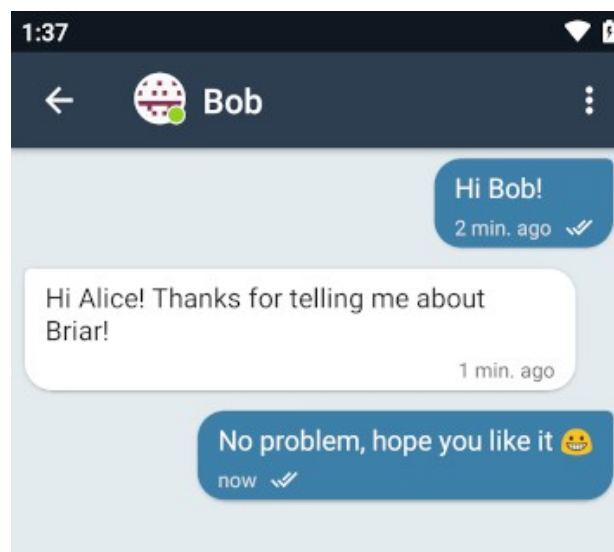
Ваши устройства будут обмениваться информацией, и через несколько секунд вы будете добавлены в списки контактов друг друга. Поздравляем! Вы готовы к безопасному общению.

Обмен сообщениями

Чтобы отправить личное сообщение, коснитесь имени контакта в списке контактов.

Совет: все сообщения в Briar полностью зашифрованы, поэтому никто другой не может их прочитать.

Если ваш контакт не в сети, ваше сообщение будет доставлено в следующий раз, когда вы оба будете в сети.



Представляем контакты

Вы можете представить свои контакты друг другу через Briar. Это позволяет им стать контактами без необходимости встречаться. Чтобы начать знакомство, коснитесь имени контакта в списке контактов и выберите в меню «Создать представление».

Ваши контакты увидят сообщение с вопросом, принимают ли они введение. Если они оба согласятся, они будут добавлены в списки контактов друг друга и смогут безопасно общаться.

(по англоязычному руководству программы)

Другие способы связи.

- Российская система связи на основе трекеров «Гончак». Предназначена для охотников и спасательных служб. Пока не замечена в продаже. Связь на расстоянии до 20 км.
- Связь на основе протокола Lorawan. Не способна передавать большие массивы данных.

Связь на расстоянии до 10 км.

- Связь через тропосферные станции. Планируется как альтернатива спутниковому интернету. Связь на расстоянии примерно 200 км. Видимо станции будут устанавливаться местными властями.

Поселковый форум.

Рассмотрим более конкретно поселковый форум, заодно поучимся. Классический форум состоит в основном из набранных на полноценной клавиатуре текстов-реплик. Он может быть просто установлен с помощью, например, OVA-файла виртуальной машины для форума PunBB. Этот файл не будет работать на 32-битном компьютере, как и на 32-битной ОС. Он имеет размер чуть менее 300 мб. Контейнер для **docker** также предлагается, и может быть не меньше. Более подробная информация на сайте <https://www.turnkeylinux.org>.

Вы сами выбираете какой форум иметь. Я остановил свой выбор на этом форуме так как он быстро работает, создаёт небольшие страницы, и разрабатывается в России.

Для установки PunBB вручную (а не с помощью OVA или подобным образом) надо выполнить три шага:

1. получить из интернета достаточно свежую версию форума. Он поставляется как один файл-архив
2. выложить содержимое архива в один из обслуживаемых веб-сервером каталогов. Это – на этот раз – программа, не устройство
3. открыть в браузере файл **index.php** из этого каталога, и выполнять что там написано. Базовая, необходимая настройка делается за несколько минут

Для установки, работы и настройки форума версии 1.3 есть условия – уже установленные веб-сервер и PHP версии не ниже 4.3.0, база данных – MySQL версии 4.1.2 или новее, PostgreSQL 7.0 или новее, или SQLite 2.

Что такое этот веб-сервер? Это программа, создающая страницы, просматриваемые в браузере. PHP – это язык программирования. Он нужен чтобы обеспечить как функции форума, так и его настраиваемость и способность развиваться. База данных хранит информацию форума, но не его программные механизмы, или параметры работы. Такая конструкция типична для многих программ. Она позволяет гибко менять форумный сервис – без обязательного изменения всех частей из которых он состоит.

Клиентской программой является браузер. Клиентским устройством может быть телефон.

Документация форума призывает: «Работайте с операционными системами типа UNIX! Используйте правильный веб-сервер, например Apache или Lighttpd.».

Больше о форуме написано на сайте <http://punbb.ru>. Сам этот сайт показывает работу форума.

База данных, язык программирования и веб-сервер устанавливаются как обычные программы – в достаточно обеспеченных пакетами ОС. Надо быть готовыми к ручной настройке.

Видео о форуме, думаю, есть в количестве не менее нескольких роликов. Например, на https://youtu.be/c_BNOrj8I5Q

Также об установке форума написано, например, тут – <http://info-comp.ru/internetst/523-how-to-install-punbb-forum.html>.

Настройка форума решает, в том числе, задачу более пригодного для чтения вида страниц.

Звукозаписи реплик можно просто класть в любое доступное хранилище – не расположенное в интернете, а в локальной сети села, и вставлять в форумные реплики ссылки на эти файлы. В таком случае крайне важно чтобы темы обсуждения были текстом, и хорошо описывали тематику. По ним можно будет искать нужную тему. Не обязательно, и не желательно, чтобы звукозаписи скачивались с форума. Достаточно, и гораздо удобнее, если они проигрываются без скачивания. Так как всё это происходит в локальной сети, где минимальная скорость достаточно велика, то звук должен нормально проигрываться.

Доступ к командной строке UNIX.

Теперь рассмотрим более «глубоко расположенную» тему обращения к терминальной строке сервера. Я пишу так потому что, если форум является просто чем-то вроде сельского клуба, то тут нужны объяснения. Доступ к консоли UNIX позволяет делать массу вещей, недоступных на многих телефонах. Зачем? Один из более-менее понятных ответов в том что

так можно конвертировать PDF-файл в набор картинок-страниц, и просмотреть их на дешёвом телефоне. Именно для этой задачи нам потребуется возможность передать эти графические файлы на телефон. Но это уже другой вопрос. На самом деле спектр возможных действий гораздо шире. Вы получаете возможность командного управления процессами на сервере.

Начальные сведения о веб-сервере.

Для простого объяснения часто подходит модель чёрного ящика. Чёрный ящик – это некий механизм, или программа, который не рассматривается ради простоты. Но нам важно что мы получаем с помощью чёрного ящика. Также нам интересно какие условия, и входящие данные нужны для его работы. Веб-сервер можно представлять как чёрный ящик, занятый в основном созданием веб-страниц. Страницы просматриваются в браузере на клиентском устройстве. Также их можно просмотреть на сервере. В качестве исходного материала, входящего в чёрный ящик, могут выступать запросы со стороны клиента, база данных, некие файлы разного типа. На практике часто результирующие страницы формируются под запрос клиента. Для этого часто используются программы на языке PHP. Кроме того, сами страницы могут содержать программы на языке JavaScript. Таким образом можно создавать сложные, полезные, гибкие инфосистемы, подстраивающиеся под клиента, и его устройство. Это – большая и сложная тема. Попутно можно заметить, что не всегда стоит использовать веб-сервер **Apache**. Автор не стремится доказать что он плох. Выбор сервера определяется решаемой задачей. Например, есть сервер **thttpd**, который проще понять и настроить, чем Apache.

Какой нам интерес обсуждать веб-сервер? Нам интересна в основном его возможность создавать веб-страницы, и поставлять их клиенту. Так же сервер может просто выдавать их без создания, но тогда страницы приходится создавать иначе. Страницы гораздо информативнее чем просто текстовые файлы¹⁶. Тем не менее рассмотрим показ текстовых файлов – так как это проще. Сервер может поставлять их как страницу особого типа. Обычные веб-страницы сложнее делать. Приведу пример. Путь есть текстовый файл, содержащий всего одну строку.

Эта строка будет показана в браузере.

Для показа этого файла нам потребуется привести его к следующему виду.

```
<html><pre>Эта строка будет показана в браузере.</pre></html>
```

Можно было бы вывести сотни строк, а не одну. Но тут имеется проблема: так вы *буквально* покажете текстовый файл. Это значит, что его строки могут не уместиться по ширине на малом экране. Кроме того, вы покажете лишь текст – без ссылок, и прочего, что возможно на «классических», обычных страницах. Расширение файла должно быть **html** или **htm**.

Для ручного создания страниц достаточно простейшего текстового редактора, и знания HTML. Также вы можете использовать, например, нечто вроде программы **asciidoc** для простого создания страниц. В последнем случае вам потребуется знать как *обозначать* части и свойства страницы. То есть вам нужно будет знать язык **asciidoc**. Он прост, но ручное создание множества страниц трудно. Для этого потребуется некая автоматизация – выполняемая программами.

Быстрый и простой ввод в эксплуатацию уже настроенных поселковых сервисов.

Настройка и установка программ лишней раз в общем никому не нужна. Но она терпима когда есть квалификация. Ожидая, что большой проблемой будет забывание нужного состояния и действий. Чтобы решить эту проблему можно, например, иметь виртуальную машину с сервисами. Ввести сервисы в рабочее состояние можно за полчаса или быстрее, но слабая техника может не подойти. Виртуальная машина может храниться в формате OVA как один файл. Он будет большим. С контейнерами есть надежда на небольшие файлы.

Автоответчик, голосовое меню и SMS в качестве услуг в локальных сетях.

Голосовое меню существует тогда когда автомат (программа) говорит вам «нажмите ... чтобы узнать ...» или «нажмите ... чтобы попасть в ...». На техническом уровне это способ раскидать услуги по массе пользователей, для пользователей это способ найти нужный

16 Имеются в виду HTML-страницы, и HTML-веб-сервер

аудиоматериал. Это нечто вроде простой поисковой машины или рубрикатора. Поэтому это полезно. SMS сообщения могут принести немалую пользу так как их содержание произвольно. Это могут быть даже ответы на поисковые запросы, выполняемые также в виде SMS. Всё это хотелось бы видеть в негородских условиях в качестве понятных услуг информационного обслуживания. Для этого стоит присмотреться к ОС и программам, предназначенным для создания современной АТС – телефонной станции. Это, насколько мне известно, Asterisk, 3CX Phone System, и, может быть, некоторые другие. Трудность будет в том чтобы это работало в локальной сети.

Мобильный телефон.

Мобильные телефоны имеют прямое отношение к инфосистемам – тем более что они уже куплены. Достаточно развитый телефон уже является частью инфосистемы если он подключён к интернету или локальной сети. Это так потому что тогда он входит в инфосистему, а последняя способна обеспечить владельца телефона инфоуслугами. Сам телефон, конечно, тоже до некоторой степени может обеспечивать этими услугами. Заметьте – это так потому что он подключён к телефонной сети. Телефон может быть подключён к этим трём сетям одновременно. Телефон является доступным и уже распространённым устройством для получения инфоуслуг. Но типично, что старшее поколение плохо знает возможности своего телефона. Ситуация была бы лучше если бы были более качественные инструкции к телефонам. Но для этого надо их читать. Как я понимаю, многие предпочли советы знакомых. Литература на русском также не радуется возможностью понять телефон. Есть проблема человеческого фактора – люди *не хотят* разбираться с телефоном. Это – ошибка. Это примерно такая же глупость как купить плиту, и не уметь пользоваться духовкой. Дополнительные возможности телефона гораздо больше чем возможность печь. Они могут совершенно изменить жизнь владельца телефона.

Чтобы говорить дальше надо разделить телефоны на группы. Примитивный телефон в наше время часто не используется вполне. Типично, что он содержит

- журнал звонков
- календарь
- будильник
- калькулятор
- встроенное радио
- диктофон
- фотокамеру

На рисунке примитивный телефон.



Более развитые телефоны обычно содержат всё это, и ещё многое. Трудность описания телефонов происходит из-за их разнообразия. Знание одной модели может не помочь в освоении другой. На рисунке более развитый телефон.



Надо понимать различие между программной средой телефона и его аппаратными особенностями. Так, например, ОС телефона никак не связана с размером экрана. С другой стороны, она всё же накладывает некие черты на телефон. Так, типично, что у телефонов кнопка включения является в то же время и кнопкой включения экрана, и кнопкой выключения телефона.

Будем рассматривать телефоны с ОС Android, обычно называемые смартфонами. Они очень популярны, и не очень дороги. Есть как минимум одна бесплатная книга по этой системе. Но она – в интернете. Производитель телефона часто вносит небольшие, но сбивающие с толку ученика изменения в систему. Какой нам интерес покупать телефон такого типа? Отвечаю.

- есть шанс найти учебник по ОС
- Вы получаете систему, стремящуюся конкурировать с компьютерной, но не равную ей
- Вы получаете простое управление ОС и программами
- Вы получаете возможность установки массы программ
- обычно вы получаете возможность не только быть клиентом сети Wi-Fi, но и точкой

- доступа этой сети
- Вы можете использовать интернет с более-менее развитым браузером, вы можете выбирать браузер

Что это вам даст?

- Вы получаете устройство, достаточное для многих работ, но ассистирующее, не равное компьютеру с компьютерной ОС. Это уже не мало
- Вы можете носить телефон в кармане
- Вы можете получать услуги через браузер. Вам часто не нужно будет спрашивать или ездить за справками. Интернет-магазины будут работать на вас
- Вы имеете хорошее клиентское устройство для села, и сельской инфосистемы
- часто ваш телефон умеет управлять принтером – но может потребоваться не любой принтер
- иногда ваш телефон можно использовать с компьютерной клавиатурой

На рисунке смартфон.



Для более точного управления сенсорным телефоном вам стоит купить стилус. На картинке стилус в упаковке.



Все смартфоны имеют чувствительный экран. С малым экраном непродуктивно и вредно работать пальцами.

Заметим, что планшет с Android гораздо более пригоден к использованию. Это в основном касается сайтов. Сайты обычно не рассчитаны на малый экран, и часто содержат мелкие детали.

В общем планшет – это реальный помощник в платной работе, особенно интересный в

командировках.

Телевизионный плеер.

Телевизионный плеер на сегодня является, можно сказать, устаревшим классом техники, замещённым более универсальными устройствами. Но эти устройства не превзошли плеер по простоте управления. Такой плеер всегда умеет воспроизводить содержимое лазерных DVD-дисков, и часто имеет и другие функции. Плеер подключается к телевизору – скорее всего через разъёмы «тюльпан» или HDMI.



В данном случае показан плеер BKK DVP030S, стоящий от 1350 рублей. Примерно так же выглядят другие плееры.

Какой нам интерес обсуждать плееры? Плеер позволяет

- проигрывать фильмы
- слушать музыку
- просматривать графические файлы – часто фото, сделанные на фотоаппарат или телефон, или планшет

То есть это в основном «кинотеатр» для села и города. Полезная функция плеера, и часто присутствующая – воспроизведение видео, и изображений, а также звука с флэш-носителей.

Диск или другой носитель можно записать по-разному. Флэш-носитель можно записать с помощью развитого телефона или планшета, или компьютера. Лазерный диск можно записать в основном на компьютере. Проблемой может быть большой объём записываемого видео. В таком случае можно использовать сервисы доставки дисков по обычной почте, и уменьшать размер видео на сайтах где есть конвертер, или на вашем сервисе аренды вычислительной мощности.

Также обратите внимание на возможность переносить публикации на лазерный диск, создавая из страниц изображения, которые просматриваются либо на обычном, либо на современном телевизоре. Использование телевизора высокого разрешения может быть необходимым. Старые телевизоры были предназначены для показа телепередач. Для показа букв они могут не годиться – особенно мелких символов.

Интернет-радио.

Интернет-радио – это поток звука, раздаваемый всем желающим без того чтобы сервис радио знал кто именно слушает его. Это очень простая вещь. Какой нам интерес слушать это радио? Интерес – не меньше, чем слушать обычное радио. Интернет-радио – это часто частное радио, сообщающее частное мнение диктора, или транслирующее звукозапись – возможно, музыку. Всё же я надеюсь, что вероятность попасть на осмысленную речь достаточно высока.

Интернет-радио само не имеет частоты вещания. Оно часто имеет адрес откуда идёт вещание. С этого адреса идёт поток данных, который оплачивается в соответствии с вашим тарифом выхода в интернет.

Реальная проблема в том что качество звука может быть низким. Я слушал радио на городском канале, и оно совершенно нормально звучало. Но на медленных и неустойчивых каналах надо быть готовым к плохому звуку.

Насколько я могу судить, работа интернет-радио обычно рассчитана на полноценный браузер. Там вы набираете адрес, и часто получаете нечто кроме звука. Часто вы можете регулировать битрейт, и поэтому регулировать количество переданных вам данных, и качество звука. Часто интернет-радио хранит старые звукозаписи передач. Их можно послушать.

Организация такого радио не требует разрешений и лицензии, и не ограничена цензурой. Это не очень сложное дело. Поэтому я рекомендую создавать интернет-радио. Особенно когда аудитория не очень хорошо видит, и лучше воспринимает на слух.

В мире создано не менее 28000, в Германии 3340, в России 1000 интернет-радиостанций.

Слушать такое радио можно некоторыми программами-плеерами, специальными программами под одну радиостанцию, и отдельными радиоприёмниками – способными воспринимать хотя бы сеть Wi-Fi. Обычно приёмники дороги, но они более понятны чем компьютерные устройства. Самый дешёвый приёмник стоит менее 1000 рублей, но это – очень редкое явление, и качество этой модели может быть низким. Лучше купить, например, MJBOX101 за 3000 рублей. Можно просто использовать микрокомпьютер за 7 долларов (хотя это лишь основа).

Интересная идея – создать поселковое радио в местной локальной сети. Оно, как ожидается, будет хорошо слышно в пределах сети.

Часто к телефону можно подключить колонки, и звук будет громким.

Не обязательно регулярно вещать, и наговаривать звук. Можно использовать программы-дикторы. Источником текста может быть интернет.

Чтение публикаций.

Чтение публикаций за компьютером в общем никогда не было проблемой. Но мы обсуждаем село, где наиболее популярным устройством является телефон. С телефонами несколько проблем:

- может не быть программ для просмотра публикаций
- программы могут быть плохими или устаревшими, не отображающими правильно файлы
- малоразмерный экран не годится для чтения так как показ страницы целиком по ширине текста не позволяет читать – буквы слишком мелки. Разумеется, можно увеличить масштаб. Но тогда видна лишь часть строки

Проблемы с малым текстом важны при просмотре публикаций где вёрстка фиксирована. Это прежде всего касается формата PDF. Также это верно про часть текста в формате HTML когда строки не переносятся, и поэтому слишком длинны. То же верно про формат PS, но он редко встречается. Документы форматов DOC, DOCX, SXW, ODT, RTF также проблемны.

Для решения этой проблемы есть мобильный вид в некоторых программах. Строки при нём ломаются и размер букв не так мал. Однако ломается и вёрстка.

В общем лучшим носимым устройством для чтения публикаций является планшет. Он может иметь ширину порядка 10 см чтобы его можно было держать одной рукой.

Ленты новостей (RSS).

Термин «лента новостей» может не соответствовать тому что вы получаете или имеете как ленту новостей. Это могут быть не новости. Напротив, я думаю, что достаточно часто это статьи, которые не привязаны к какому-то моменту – хотя на дату их написания – как и в случае со статьёй – надо обращать внимание. Многие тексты устарели.

Какой нам интерес обсуждать или использовать ленты новостей? Можно просто искать статьи в интернете. Но благодаря лентам новостей вы автоматически получаете и можете читать статьи. Вам достаточно выбрать источник или ленту новостей. То есть вам нужен адрес. Это не адрес сайта. С этого адреса в программе, созданной для сбора новостей, будут запрошены статьи. Это можно сделать как на развитом телефоне, так и на планшете, и компьютере. Такая программа может настраиваться. В случае медленных или дорогих каналов выхода в интернет важна возможность не запрашивать иллюстрации. Также можно рассчитывать что в этих статьях будет меньше рекламы, или не будет совсем.

Телеприставки (TV Box).



Основная задача Tv бокса расширить функциональность вашего телевизора. Так же как и на smart-tv вы сможете просматривать видео с популярных ресурсов, а также появится возможность устанавливать дополнительные приложения и игры. Таким образом из обычного телевизора мы сделаем телевизор лучше чем smart-tv. Tv box это своего рода мини компьютером в котором есть своя операционная система, процессор, память, видео чип и различные интерфейсы для подключения периферийного оборудования.

(по тексту zen.yandex.ru/ninja_it)

Инфоматы, голосовые ассистенты.

И то, и другое – попытка адаптировать компьютерную технику к невежественным людям. При этом возможности для этих людей не очень велики – по крайней мере пока. Типичный инфомат выполняет немного функций, и относительно понятен.



Голосовой ассистент должен как минимум уметь отвечать на вопрос. Это происходит благодаря обработке смысла вопроса – скорее всего после передачи звукозаписи через интернет.



Также голосовой ассистент как программа существует для смартфонов.

Микрокомпьютер.

Микрокомпьютер представляет особый интерес для провинции как устройство, похожее на классический – много описанный – компьютер, при том дешёвое. Таких микрокомпьютеров

много. До 1 января 2019 года уже может быть выключен старый телесигнал, и процент телевизоров высокого разрешения имеет причину вырасти. Имея новый телевизор, пользователь может подключать его к рассчитанным на его кабель микрокомпьютерам. Наиболее дешёвый такой компьютер стоит 7\$.

Автор имеет небольшой опыт использования микрокомпьютера модели Raspberry Pi B. Неожиданно возникла проблема недостатка напряжения питания, причём это не сразу стало понятно. Без обеспечения достаточного питания компьютер не может работать. Он продаётся без блока питания. Даже если компьютер нормально работает, но вы используете его провод питания или разъём для подключаемого устройства мощности может не хватить. В такой ситуации нужно иметь блок питания и\или питатель с USB-портами, и повышенным током – более 1 А. Провода также должны быть качественными, иначе они сами могут стать причиной плохого питания.

Скорость работы такого компьютера, как и небольшой (512 мб) объём оперативной памяти мотивируют иметь на компьютере облегчённую ОС, и даже не ставить поверх неё требовательные программы. Заметим, что описываемая модель способна нормально воспроизводить видео формата HD.

Есть сведения что к такому компьютеру можно подключить жёсткий диск типоразмера 3,5 дюйма. Это позволяет иметь достаточно постоянной памяти для более традиционного применения. На таком диске, например, можно хранить фильмы. Также есть сведения что можно подключить портативный читатель лазерных дисков, что для села особенно интересно из-за пересылки дисков по почте.

В любом случае ОС в описанной технологии расположена на карте памяти. В качестве таковой может представлять интерес NetBSD как очень лёгкая система. Но она может быть непонятна пользователю. Может быть хорошим выбором как штатная ОС Raspbian – облегчённый Debian, так и другие, например, DietPi. Последняя имеет выпуск для пробы в эмуляторе на обычном компьютере, но *не* содержит сразу после установки оконную среду. Вы можете в эмуляторе поставить поверх ОС оконную среду, и другие программы. Также представляет интерес ОС OpenWrt. Она заметно отличается от ОС для классических компьютеров, планшетов и телефонов, и может вызывать много затруднений. Поэтому её стоит пробовать лишь при весомых причинах. OpenWrt рассчитана на роутеры и работает быстро. Также стоит иметь в виду ОС Q4OS, которая родственна Debian.

Типичным способом установки ОС на карту памяти является простая, но, возможно, страшная для ученика процедура записи файла-образа с помощью программы **dd** – на Linux. ВАЖНО – запись образа стирает всё что уже есть на карте. На UNIX устройства представлены как файлы. Общий вид команды для записи ОС на карту в простейшем виде таков:

```
dd if=файл-образ of=карта памяти
```

В конкретном случае это может выглядеть так

```
dd if=2018-04-18-raspbian-stretch.img of=/dev/sd1
```

Потребность в компьютере где идёт запись на карту памяти – дополнительная трудность для провинции. Записать ОС на карту памяти можно с помощью телефонной и планшетной программы **DriveDroid**, но она требует Android, и могут быть ограничения; эта возможность не проверена автором.

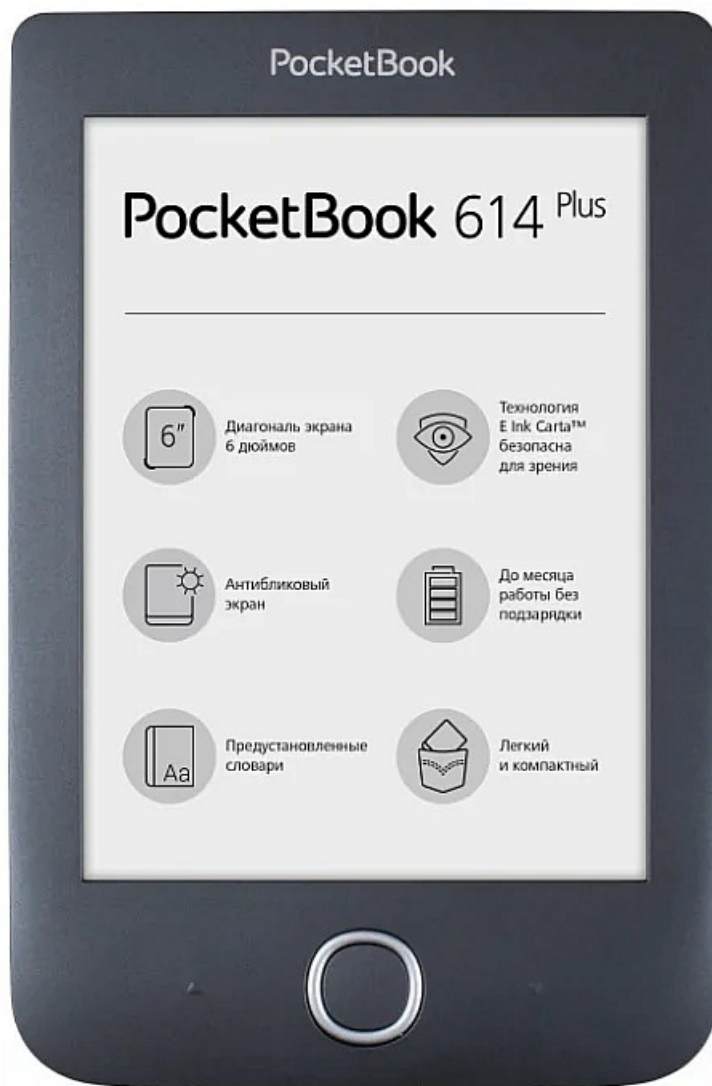
Подключение внешнего читателя лазерных дисков.

Это важно как способ читать файлы, которые могут быть получены из сервиса для отправки файлов на лазерных дисках. Тут могут быть важны 1) модель читателя, 2) качество проводов питания, 3) наличие дополнительного отводка от провода, через который надо запитать читатель, 4) особенности компьютера, и его программ. При удаче вы сможете читать с лазерного диска, но чтобы записывать на него потребуются соответствующая программа, например, **Brasero** (в компьютере).

Глава 34. «Электронная книга».

Электронная книга – это простейший на сегодня способ чтения литературы, и, часто, просмотра изображений (фото), и фильмов, а также прослушивания музыки. Может быть DVD-плеер проще, но его нельзя носить с собой. Прежде чем покупать книгу крайне желательно

узнать как ей управлять – берите инструкции в интернете. Иначе она может стать бессмысленной покупкой. Если ученик не может пользоваться книгой, то, мне кажется, это уже симптом деградации мозга – поймите меня правильно. Самая простая из известных мне книг дорога, но её покупка оправдана так как это то же что купить сотни книг. Книги, и прочее, что можно просматривать часто берутся из интернета и записываются на карту памяти, или в собственную память книги. Наряду с интернет-радио это интересная вещь для невежд.



Глава 35. Некоторые интересные программы для Android.

Я не жду что программы для этой системы будут равны компьютерным – по крайней мере пока размер экрана слишком мал. Тем не менее я признаю полезность устройств, которые можно носить в кармане. Возможно, некоторые программы сократят объём передаваемых данных. Вам могут быть полезны следующие программы.

- Сокращающие лишние поездки.
- Позволяющие заменить газеты и книги.
- Заменяющие радио и телевизор.
- Для заказов товаров.
- Для отслеживания посылок.

- Для банковских операций.
- Для набивки и вёрстки документов.
- Для просмотра видео, и слушания музыки.
- Для доступа к компьютеру – например, в командировке.
- Для замены телефонных звонков.
- Для просмотра электронной почты, ответов на письма.
- Для определения лучшего маршрута пешего или автомобильного движения. Для определения времени ожидания транспорта. Для показа расписаний транспорта.
- Для определения прогноза погоды.
- Для выбора лекарств.
- Записывающие речь.
- Для аренды транспорта.
- Определяющие положение другого телефона.
- Показывающие географические и схематические карты.
- Для просмотра прежде всего PDF-документов.
- Для блокировки камер, то есть пресечения просмотра их изображений вирусными программами и посторонними.
- Для создания программ – если они нужны.
- Для ускорения работы устройства, освобождения его памяти.
- Для помощи именно компьютерным пользователям в разном виде.
- Для поиска общедоступных беспроводных сетей.
- Для перевода текста.
- Для распознавания речи. Если вы находитесь в тихом помещении то не нужно набивать текст. Его достаточно произнести. Такие программы, думаю, пока нуждаются в быстром канале в интернет.
- Дикторы.
- Для показа телепрограммы и телетекста.
- Для печати, взаимодействия с принтером.
- Калькуляторы – в том числе научные и инженерные.

Отметим особенно популярные и важные программы.

- Kaspersky Internet Security – антивирусная программа производства российской фирмы.
- WhatsApp – программа для общения без необходимости спешить к телефону при звонке. Можно пересылать файлы, говорить, писать.
- TP-LINK Tether и подобные – для настройки роутера.
- Play книги – доступ ко множеству книг – в том числе платных.
- Gmail – программа для почтовой переписки с помощью бесплатного ящика Google.
- ReadEra – для чтения книг.
- Яндекс транспорт.
- Яндекс электрички.
- Гисметео – прогноз погоды.
- Яндекс еда.
- РЖД пассажирам.
- РИА новости.
- Доставка Пятёрочка.
- АШАН.
- Все аптеки.
- Яндекс Навигатор.
- Госуслуги.
- Яндекс – с Алисой.
- SHAREit – передача файлов.
- Шахматы уровня 100.
- Родительский контроль Kroha – Детский режим.
- Rar – архивация, извлечение из архива.
- Яндекс.Браузер Лайт.

Алиса – это не сотни телефонных диспетчеров, а интеллектуальный сервис на основе искусственного интеллекта, способный «понимать», и отвечать, а также выполнять действия

на вашем смартфоне.

Глава 36. Элементы автоматизации.

Автоматизация позволяет вам решать ваши задачи путём их описания, практически программирования. Благодаря этому вы сможете быстро и верно – если всё правильно – делать свои дела. Это требует знания и мышления. Но альтернатива ещё хуже. Это ручная работа. В среде Windows возможности автоматизации есть, но UNIX, как более профессиональная среда, даёт пользователю больше средств для неё. Он, и литература, практически подсовывают в руки средства автоматизации.

Мы не станем много говорить об автоматизации. Обычное средство автоматизации – **BASH**.

AWK

Рассмотрим для начала лишь одну программу, которая обычно уже есть даже на урезанных ОС. Это AWK, и её версия **gawk**. По этой программе есть свободнораспространяемая книга и документация в интернете – на английском, а также русский перевод книги конца 90-х годов – например, на

https://www.math.spbu.ru/user/rus/cluster/Doc/Library/awk_baluev/awk_bal_oglav.shtml.

Мой опыт работы с программой не похож на поход в ресторан. Опыт будет определяться вашим знанием документации. У людей с живым умом достаточно быстро будет формироваться представление о том что конкретно делать. Это – общее замечание, верное для всех программ. Если вы знаете документацию, и вообще соображаете, то работа будет не слишком трудна, и достаточно быстра. Но проблема в том что людям трудно осилить том, который надо знать чётко и детально – хотя бы отчасти. Практика поможет запомнить. Нужно терпение.

Программа в своём наиболее популярном применении позволяет отбирать части текста. Благодаря этому вы формируете, например, команды для интерпретатора. Но как неинтеллектуальная вещь, AWK требует формальных признаков, например, разделителя частей текста. Это портит жизнь? Обычно нет.

Рассмотрим простой пример. Программы обычно создают предсказуемые наборы строк. Структура строк часто одинакова, но в этом надо иногда убедиться. Так, программа **ps** создаёт нечто вроде

```
PID TTY      TIME CMD
25824 pts/2    00:00:00 bash
25838 pts/2    00:00:00 ps
```

Лишь первая строка – особенная. Все остальные – не по содержанию, а по структуре – одинаковы. Имеют 4 части. То что выглядит пробелом может быть табуляцией. **AWK** – если вы не настроили её иначе – рассматривает как разделитель строки и пробел, и табуляцию. Вся строка обозначается как \$0, а её части – слева направо – как \$1, \$2, \$3, \$4, и так далее. Благодаря этому если надо отобрать лишь 4 часть строки, то следует подать такую команду с использованием того что называется pipeline, то есть передачей данных от одной программы другой.

```
ps|awk '{print $4}'
```

В результате вы получите.

```
CMD
bash
ps
awk
```

Более сложная конструкция для отбора текста выглядит так.

```
awk '/ps/{print $4}'
```

Что она делает? Отбирает все строки где есть «ps». Это могут быть части слова. Иногда количество строк не позволяет просматривать их глазами, да и не надо.

Важная возможность **AWK** – можно настроить и отбор частей строк, и разделитель для них. Чтобы указать что считать разделителем, нужно явно указать это.

```
ps|awk 'BEGIN { FS = ":" };{print $2}'
```

В результате:

```
00
00
00
```

Если разделителем является двоеточие то второй частью строки являются нули. Если вы подадите такую команду – отбирающую все строки с цифрами

```
ps|awk '/[0-9]/'
```

то получите все строки кроме первой.

AWK умеет обрабатывать более одной строки за раз. Для этого разделитель записей (record separator) должен отличаться от конца строки.

Программа считает строки по мере обработки и соответственно меняет значение счётчика, который можно использовать для отбора данных. Так, если мы имеем файл, содержащий

```
1
3
```

то команда

```
awk '{print NR}' awkz
```

будет печатать номера пройденных строк в файле **awkz**.

```
1
2
```

Конец строки не виден, но записан в файлы. Он также обрабатывается программой. Например так.

```
echo -e "1\n3"|awk '{printf $0}'
```

Мы посылаем **AWK** две строки, разделённые концом строки, то есть «\n». Программа печатает обе строки с помощью функции, которая не делает отдельных строк, она не выводит конец строки. На выходе получается одна строка. Если вы хотите заменить конец строки на что-то, то можно просто добавить это к строке. Если надо потом оборвать строку то используется команда **print** с пустой строкой в качестве печатаемой.

```
echo -e "1\n3"|awk '{printf $0"123";print ""}'
```

Часто надо чтобы некая программа получала определённую строку для своей работы. Допустим, что эти строки уже есть в файле с названием **arguments**. Тогда можно создать новый файл, где этим строкам будет предшествовать имя программы.

```
awk '{print "cat "$0}' arguments>auto
```

Перед каждой строкой будет написано «cat». Результат будет в файле с названием **auto**. Но ОС не даст нам многократно вызвать программу **cat** с помощью этого файла пока сам файл не является исполнимым. Чтобы он стал таким следует подать команду

```
chmod +x auto
```

Именно так автор выполнял автоматизацию – в том числе создавая большие файлы, но простые, и доступные проверке чтением. Это может быть не очень хороший, даже расточительный путь, и уж точно не единственный.

Если вы хотите построить набор строк с изменяемой частью – например, счётчиком, то это можно сделать по-разному, даже с помощью **AWK**. Так, если у нас есть файл **awkcounter** с таким содержимым

```
text
text
```

то команда

```
awk '{print $0NR}' awkcounter
```

выдаст на экран следующие строки

```
text1
text2
```

Зачем это может быть надо? Один из ответов – для создания файлов или каталогов. Их имена не могут совпадать. Счётчик придаст уникальность имени.

Python.

Есть такое мнение что вместо старомодных программ можно использовать

соответствующие возможности **Python**. Спорить с этим я не буду. **Python** позволяет получить значение для счётчика одной строкой. Её вид будет таким

```
python counter.py
```

Тут **counter** – произвольное имя файла, задача которого – создать набор строк со счётчиком. Просто программа для вывода номеров строк может быть в таком виде.

```
i = 0
while i < 10:
    print(i),
    i = i + 1
```

Однако нам надо не буквально это. Надо чтобы именно строки, которые уже есть, содержали счётчик. Пусть у ученика есть файл со строками **strings** с таким содержимым.

```
s1
s2
s3
```

Тогда программа для вывода строк будет такой.

```
import sys
file=open('strings','r')
for lines in file:
    print(lines)
file.close()
```

Программа для вывода тех же строк со счётчиком будет такой.

```
import sys
counter=0
file=open('strings','r')
for lines in file:
    counter=counter+1
    print(lines.replace("\n","")),
    print(str(counter))
```

Она выдаст следующее.

```
s1 1
s2 2
s3 3
```

У нас уже есть строки со счётчиком, но нам может потребоваться чтобы строки включали в себя значения счётчика, а не были отдельно от них. Для этого подадим вывод программы на вход другой программы для замены пробелов так.

```
python pythonstr.py|python pythonstr2.py
```

Программа **pythonstr.py** выводит строки со значениями счётчика, а программа **pythonstr2.py** заменяет пробелы на ничто, то есть просто уничтожает их. Вот она.

```
import sys
for lines in sys.stdin:
    print(lines.replace(" ",""))
```

Результат работы.

```
s11
s22
s33
```

Надеюсь, что это не показалось страшным. Так как **Python** является развитым языком с массой возможностей, то с его помощью можно делать, наверное, всё что потребуется для автоматизации. Советую посмотреть книгу «Автоматизация рутинных задач с помощью Python». Автор – Эл Свейгарт.

Программа paste.

Эта программа, как и **Python**, **AWK**, **xargs**, **tree**, должна быть в вашей программной среде. Ставить её не потребуется. Программа интересна как средство соединения двух файлов для получения третьего файла. Работа с программой очень проста. Пусть у нас есть два файла. Один из них называется **column1** и содержит следующие строки

```
s1
s2
s3
```

Второй называется **column2** и содержит эти строки.

```
s1  
s2  
s3
```

Тогда программа **paste** при команде

```
paste column1 column2
```

выдаст на консоль или в терминальную программу строки

```
s1 s1  
s2 s2  
s3 s3
```

То есть она соединит строки, и вставит между ними табуляцию.

Если вы хотите задать вместо табуляции пробел то команда принимает такой вид.

```
paste -d " " column1 column2
```

Между кавычками – пробел.

Если вы хотите получить результат не на экран а в файл, то используйте перенаправление в файл.

```
paste column1 column2>joinfile
```

В данном случае **joinfile** – произвольное имя файла; если такой файл уже есть, то он будет переписан. Его прежнее содержимое будет потеряно.

Регулярные выражения.

Регулярные выражения (то есть **regex**) интересны возможностью описывать нужные строки в таких программах как **AWK**, но не буквально, а с помощью некоторых правил. Благодаря этому ученик может, например, заменять множество разных строк не описывая их буквально, что было бы в некоторых случаях трудно – если не сказать невыполнимо. UNIX позволяет вам это. Тема для некоторых страшная. На самом деле – не очень. Вы можете почитать, например, руководство по **SED**.

Как это «не буквально», и вообще? Например, чтобы обозначить диапазон цифр, пишут «[0-9]». Это не только 0 и 9, но 7 или 5.

Приведу цитату из аннотации к русскоязычной книге Джеффри Фридла «Регулярные выражения» «...откроет перед вами секрет высокой производительности. Тщательно продуманные регулярные выражения помогут избежать долгих часов утомительной работы и решить проблемы за 15 секунд.».

Для проверки выражений, и лучшего знания смотрите интернет. Например

<https://www.regular-expressions.info/>

<https://rexegg.com/>

<https://regex101.com/>

<http://www.regexplanet.com>

<https://www.regexpal.com>

xargs

Xargs является штатной программой в Linux и FreeBSD для определения параметров работы консольных программ в рамках одного конвейера. Так, например, конвейер

```
echo A B C | xargs -n 1 echo
```

принимает на вход поток вывода от команды **echo** и отправляет его как аргументы в команду **echo**, запуская для каждой группы из одного параметра (т. е. для каждого параметра) отдельный процесс. Это почти то же что подать команды

```
echo A  
echo B  
echo C
```

Рекомендуется экспериментировать с программой чтобы понять как она работает.

Как получить имена всех файлов каталога с путями к ним.

Наверное, самый простой способ – использовать команду

```
tree -i -f
```

для вывода всех файлов и каталогов в текущем каталоге. Текущим каталогом может быть корневой «каталог», то есть самый общий уровень дерева каталогов. Он обозначается как «/». Подав команду

```
cd /
```

Вы попадаете в него.

Как найти 10 самых больших файлов в файловой системе.

Это может быть нужно при очистке постоянной памяти или носителя – например, перед сменой ОС. Чтобы найти 10 наибольших файлов и каталогов выполните команду

```
du -Sh | sort -rh | head -10
```

Команду можно подавать несколько раз чтобы стереть или перенести наиболее крупные файлы.

Языки программирования.

Рассмотрим теперь простейшие способности и свойства, которыми должен обладать не только **BASH**, но и любой, или типичный, язык программирования.

- Разделение инструкций. Указания интерпретатору или компилятору как неинтеллектуальному механизму должны быть чёткими. Простейший вопрос об этом – отделение одной части программы от другой во избежание многозначности.
- Типизация данных. Тип данных важен так как он определяет что можно и нельзя делать с данными. В простейшем виде надо различать числовые и строковые данные. Возможно, конкретный интерпретатор или компилятор сможет догадаться к какому типу отнести данные, но он может и ошибиться.
- Обработка переменных. Переменные, в отличие от данных файлов – это обычно просто строки или числа. Более сложные конструкции не считаются переменными, хотя тоже могут меняться. Также может быть надо передать переменную из одной программы в другую, и это возможно благодаря командному интерпретатору встроенному в ОС.
- Константы. Это неизменные величины. Возможно, потребуется именно константа чтобы избежать случайного изменения значения.
- В языках программирования есть понятие выражений, то есть частей программы, имеющих значение. Так, например, переменная с именем А имеет значение 5, и является выражением.
- Операторы – части программы, служащие для обработки данных, и их оценки. Так, например, оператор «+» в выражении «1 + 1» показывает что нужно сложить числа.
- То, что часто называют управляющими конструкциями. Они по сути являются средством
 - повтора действий
 - перехода к другой части программы
 - обрыва и возобновления действий
 - выполнения действий по условию
- Функции. Это почти самостоятельные программы внутри более крупной программы. Функции позволяют организовать программу из наборов функций. Функция для своей работы должна быть вызвана в более общей программе.
- Обработка ошибок. Это важная часть реальной работы. Ошибки всегда возможны. Во всех развитых языках появившиеся ошибки вызывают сообщения для программиста. Так, например, деление на ноль запрещено, и такая попытка предусмотрена как ошибка программы.
- Ввод и вывод данных относительно программы. Простейшая программа печатает «Hello World!», и этим выводит строку на экран.
- Взаимодействие с операционной системой. Это нужно так как программа находится не в вакууме, и не нужна там.
- Готовность к исключительным ситуациям. Такая ситуация, например, возникает, когда поданы непригодные данные.

Вы можете узнать по документации заранее какие ошибки и ситуации бывают. Это экономит вам массу времени и сил.

Глава 37. Обзор административных работ и вопросов защиты программной среды и данных.

Это трудная тема для автора так как он не является специалистом в ней. Многие вопросы не будут описаны. Администрирование – это по требованиям к исполнителю и возможному эффекту профессиональная занятость. Тем не менее этой работой приходится заниматься частному пользователю – хотя бы потому что больше некому. В эти работы входит прежде всего обеспечение защиты от потери данных, их разглашения и вторжения в компьютер по сети. Признаком административной работы является капитальный, иногда фатальный, влияющий на других пользователей эффект от работы. Административные команды и программы не работают от имени неадминистратора.

Установка и удаление программ – классический путь.

Поставляемая в любом виде ОС уже является набором программ. Но даже в случае её примитивного использования вам потребуется поставить хотя бы некоторые программы.

Тут рассматривается классический случай, когда программы не являются сетевыми. Они могут быть сетевыми по назначению, но не по методу своей работы. Сетевые программы – это программы, которые являются частью сетевого взаимодействия, а не средством для него. Это возможно потому что технически программе не обязательно даже быть установленной или храниться в виде файла или файлов. Программа, тем не менее, работает, просто поступив из сети в оперативную память. Это прежде всего или только в связи с работой развитого браузера.

Установка программы делается однократно. Программу не надо ставить перед каждым запуском. Суть установки в том что программа вписывается в рабочую среду, и становится способной быть запущенной, использованной, выключенной и настроенной. Также её в общем случае можно удалить. Её можно обновить. Чтобы удалить программу надо прежде её выключить, перестать использовать. Всегда желательно проверить на вирусы программу перед её установкой. Иногда программы поставляются как архив, который на самом деле чего-то от вас требует, и может быть опасен.

На Windows установка и удаление программ выполняются без организованного хранения программ, обычно простым образом, а способ установки и удаления программ либо стандартный, либо описан на сайте программы. На Windows реализован неорганизованный, но простой способ выполнения этих работ. Обычно для установки надо либо вставить установочный лазерный диск¹⁷, либо запустить программу, являющуюся файлом с именем INSTALL или SETUP, или запуском файла с расширением MSI. Удаление программ выполняется обычно с помощью **Панели управления**, которая является частью ОС.

На Android установка, обновление и удаление программ также очень просты, и в классическом случае выполняются одинаково и просто с помощью сервиса Google Play, доступного через интернет. Это также неорганизованный способ хранения программ. Удаление программ делается в настройках ОС. Можно использовать F-Droid – каталог свободных приложений с открытым исходным кодом (FOSS) для платформы Android, но этот способ может входить в противоречие с настройкой Android, которая требует установки программ лишь из Google Play.

Источником программ исторически прежде всего были магнитные плёнки, затем дискеты, лазерные диски, интернет, соседние компьютеры локальной сети. Установочный файл можно перенести просто записав его на носитель на одном устройстве, и прочтя на другом.

Однако при этом – ВАЖНО – надо учесть лицензионные ограничения. Программы обычно поставляются под лицензией, текст которой может не поставляться с программой, и не быть переведён на русский язык. То же верно про программы на семейном или общественном компьютере – лицензия может помешать использовать их многим людям или всем кроме одного.

Иногда программу можно не ставить. Она работает без установки. Это обычно либо небольшие скрипты для одной задачи, либо портативные программы. Обновление портативных программ, как мне кажется, пока может быть не приведено в хороший вид.

17 Устанавливаемой программы, не ОС! Аутоплэйный диск. Он может не вызвать запуск установщика – смотря по настройке ОС.

Использование неких наборов команд взятых со страниц интернета или из литературы рискованно. Это так не обязательно потому что такая программа создана как вредная. Предполагаемые условия могут быть непонятны ученику.

Обычно программа поставляется как один файл. Но при этом типично, что после установки она существует как десятки или сотни файлов – прежде всего благодаря архивации, и потому что такой её создали. На Windows обычно программа не требует наличия некой другой программы, а если требует, то эта другая должна быть поставлена заранее – до установки рассматриваемой; вероятно, о нужных программах будет сообщено при установке – которую вы можете без вреда оборвать.

В среде UNIX типично что одна программа требует наличия нескольких программ, вернее один пакет требует наличия других пакетов. Обычно их можно устанавливать одновременно. Имя файла программы часто есть её название, или сокращение от него. Чтобы разобраться с установкой программы лучше всего смотреть её сайт. Часто программа описана на многих сайтах. Также данные пакетного менеджера позволяют узнать о программе. Так, в среде ОС с пакетным менеджером **APT** для поиска подходящей программы прежде всего можно использовать программу **apt-cache**. Подобная программа должна быть и в других случаях – в том числе в среде ОС семейства BSD. Программа умеет показывать названия программ и описания их. Вот пример работы программы.

```
apt-cache search text editor
ed - classic UNIX line editor
emacs - GNU Emacs editor (metapackage)
emacs24 - GNU Emacs editor (with GTK+ GUI support)
emacs24-bin-common - GNU Emacs editor's shared, architecture dependent files
emacs24-common - GNU Emacs editor's shared, architecture independent infrastructure
emacs24-common-non-dfsg - GNU Emacs common non-DFSG items, including the core documentation
emacs24-dbg - Debugging symbols for emacs24
emacs24-el - GNU Emacs LISP (.el) files
emacs24-lucid-dbg - Debugging symbols for emacs24-lucid
emacs24-nox - GNU Emacs editor (without GUI support)
emacs24-nox-dbg - Debugging symbols for emacs24-nox
gedit - official text editor of the GNOME desktop environment
gedit-common - official text editor of the GNOME desktop environment (support files)
gedit-dev - official text editor of the GNOME desktop environment (development files)
gir1.2-gtksource-3.0 - gir files for the GTK+ syntax highlighting widget
gprompter - intelligent predictive GTK+ text editor
gprompter-dbg - intelligent predictive GTK+ text editor (debugging symbols)
inkscape - vector-based drawing program
```

Программа не требует прав администратора. Она зависима от данных репозитория и PPA.

Тема управления наличием программ касается того важного факта что программная среда является системой. Система отличается от хаоса тем что противоречия в ней позволены, но разрешены приемлемым образом. Основным средством поддержания системности является сама операционная система, а в ней эту же функцию выполняет штатный (или другой) *менеджер пакетов* – в UNIX. Менеджер пакетов может быть представлен более чем одной программой, и управляет не параметрами работы установленных, или поставляемых в ОС программ, а в основном установкой *пакетов*¹⁸, в которых находятся программы, и удалением ненужных или мешающих программ среди тех, которые уже есть в программной среде, поставлены поверх ОС. Именно из-за потребности в системности один пакет может быть относительно другого пакета заменяющим или конфликтующим. Также пакет может быть необходимым (depends), рекомендуемым (recommended), предложенным или внушённым (suggests), и расширяющим (enhances)¹⁹. Обычно установка включает необходимые, то есть *зависимости*, и рекомендуемые, и этого достаточно. Потом вы можете добавить остальные. Потребность в пакетах определяется конкретикой использования программы. Также вы можете отказаться от рекомендованных пакетов – например, чтобы уменьшить расход лимитированного месячного объёма данных, но это может не получиться в простой, но

18 Понятие в компьютерной науке используется много раз – в разных смыслах.

19 По опыту использования пакетного менеджера APT.

негибкой программе, обычно называемой **Центр приложений**. Последний сам может нуждаться в установке, и существует лишь в некоторых ОС.

Чтобы лучше узнать о пакетах стоит посмотреть сайт вашей ОС. В случае с Debian у вас есть не очень щедрая на информацию, но понятная аннотация к каждому пакету, и обслуживающий вас набор страниц по адресу www.debian.org/distrib/packages. Также вы можете узнать о пакете с помощью менеджера пакетов – скорее всего, информация будет на английском. Иногда – как в случае с **texlive** – вам по-настоящему разобраться с пакетами таким путём не удастся так как вы не получите достаточно описательной информации по этому огромному набору программ. В таком случае вам помогут многие сайты. Также надо быть готовым к тому что документация по программе поставляется в отдельном пакете – даже в случае когда есть man-страница. Очень вероятно что этот пакет является зависимым или рекомендованным.

Для поиска программ лучше всего использовать поиск в интернет. Также в UNIX вы можете искать программу с помощью менеджера пакетов, но он ищет не в интернете, а по своей базе данных известных ему, подключённых к нему хранилищ.

Для большинства ОС семейств Linux и BSD есть набор пакетов, расположенных в *официальных* хранилищах. Эти пакеты отслеживаются командой, поставляющей и развивающей ОС, и они являются наименее вероятной причиной проблем среди всех источников программ. Также есть неофициальные, независимые от этой команды хранилища. Хранилища пакетов называются *репозиториями*. В частном случае репозиторий расположен на ваших носителях. То же верно для других ОС – вам не обязательно брать программы из интернета, но там они могут быть свежее.

Как понять какие репозитории являются официальными? Посмотреть сайт ОС. Например, Яндекс содержит официальный репозиторий для многих ОС. Хорошее хранилище всегда старается иметь наиболее свежие версии пакетов, но не вообще, а для конкретной версии ОС, и не новее чем в основном хранилище. Брать пакеты из основного хранилища нежелательно так как это означает слишком много запросов на один репозиторий. Вторичные, копирующие оттуда репозитории в случае популярных ОС существуют в количестве 10-20 штук. Выбирайте тот который ближе. В бесплатных системах этот выбор делается просто. Вы не обязаны всё время брать из одного хранилища.

Репозитории в интернете *подписаны*. Они имеют ключ, то есть строку, по которой вы можете проверить хранилище, и убедиться что это оно, а не другое – подозрительное.

То же верно про пакеты. Но с пакетами это важнее так как подмена пакета выглядит более опасной. Обычно никто из пользователей не проверяет пакеты из официальных хранилищ. Это делает пакетный менеджер, но может потребоваться вручную прописать репозиторий как доверенный. При установке пакета может быть вопрос или сообщение по поводу отсутствия ключа репозитория.

Поначалу, пока вы учитесь, вам вряд ли потребуется настройка источников пакетов или добавление репозитория если вы пользуетесь городским каналом выхода в интернет. Напротив, если вы его не имеете, находитесь на краю Земли, то есть вероятность что придётся прописать наличные лазерные диски как источники пакетов. Диски можно купить или платно заказать по почте. Также вероятно что у вас сложится свой набор пакетов, который лучше всего превратить в репозиторий, и поделиться им с другими.

В условиях регионов страны часто нет смысла пытаться получить весь официальный репозиторий. Debian распространяется на дисках, содержимое которых равно официальному репозиторию, но не другим репозиториям. Диски пронумерованы. На сегодня набор программ для этой системы выглядит избыточным для многих ситуаций и пользователей. Думаю, что достаточно первых 10 DVD дисков. В условиях научной или производственной работы надо иметь доступ к наибольшему количеству программ вообще. В учебных заведениях также нужны конкретные программы, которые могут быть не нужны дома. Это в том числе научные программы. На факультетах точных наук, химии, инженерных профессий может потребоваться LaTeX.

Для работ со звуком и видео может потребоваться использовать репозиторий **deb-multimedia**.

Популярные пакеты, управляемые программой **rpm** не являются частью официального репозитория Debian, и пакетами в обычном смысле. То же верно про пакеты программ для **Python**. При этом об этих пакетах надо помнить так как они могут быть нужны.

Некоторые программы – в том числе крупные и нужные – могут никогда не войти в репозиторий. Это, в том числе, программы для ускоренных расчётов, и вообще работ с помощью ускорителей видеокарт (GPU), некоторые программы для профессионального программирования и создания трёхмерных моделей и игр. Программы таких популярных хранилищ как **github**, **gitlab**, любые модули (плагины) для программ, литература не входят в репозитории.

Для ОС, происходящих от Ubuntu Linux, и, возможно, некоторых других, существуют PPA-архивы. Они интересны потому что могут содержать нужные программы, отсутствующие в других хранилищах. Эти архивы также подписаны. Их можно использовать в системах, родственных Debian, но это может быть нежелательным так как я не уверен что их пакеты составят систему с другими. Я пока не читал о частных PPA-архивах, они будут доступны через интернет. Для поиска PPA рекомендуется использовать сайт launchpad.net. Также обычный поиск в интернете может помочь.

Если нужный пакет уже установлен, то его установка не производится, или пакет обновляется – из-за устаревания пакетов относительно репозитория. Так как набор программ и пакетов может быть самым разным, то и запрашиваемый при установке программ набор пакетов тоже может быть разным. Очень вероятно, что уже установленные, и даже удалённые программы хранятся на компьютере, и их можно поставить без запроса их из сети. При типичной установке с помощью команд вы можете узнать размер суммы пакетов, и отказаться от установки. Такой отказ безвреден. В городских условиях – на быстрых, безлимитных каналах, можно использовать «тихий» режим установки без запросов. Это позволяет сделать установку автоматической.

Попутно заметим, что, хотя иногда нужно иметь полную копию репозитория, но не нужно ставить всё подряд что там есть. Обычно поверх ОС поставлено не более нескольких десятков программ – не пакетов – которых может быть больше, а явно используемых, запускаемых – обычно вручную – программ.

Репозиторий не является простым набором пакетов. Это хранилище, построенное в соответствии с конкретным пакетным менеджером. Поэтому они могут взаимодействовать. Это взаимодействие выражено, в том числе, в соответствии индексных файлов и хранилища²⁰. Индексные файлы важны потому, что они позволяют организовать определение списка необходимых пакетов для установки программы без использования хранилища. Это значит что компьютер с репозиторием не перегружен, и может заниматься своим делом – отправлять пакеты. Также он хранит и раздаёт индексные файлы.

Удаление программ означает что они выходят из всех инструментов пользователя, и больше не запускаются. Их файлы стираются, но могут остаться параметры работы – смотря как удалять. Файлы, созданные с помощью программы, не стираются. После удаления программу можно поставить снова.

Программы можно ставить и стирать, обновлять, настраивать любое количество раз. При плохом канале в интернет важно что пакеты могут всё же храниться на компьютере – даже если программа удалена. Поэтому повторная установка проходит без лишних запросов и быстро.

Обновления делятся на необходимые и срочные, и другие. Всегда старайтесь получить срочные обновления как можно раньше. На сегодня сервис спутниковой передачи данных может не удовлетворять этому требованию.

Всё это хорошо – скажете вы – но какие мне нужны программы? Простой ответ: если вы едете в деревню из города, то все. Ответ посложнее: смотрите интернет; обычно нужна оконная среда, браузер, антивирус, архиватор, что-то для редакции текста, просмотра файлов форматов DOC, DOCX, RTF, PDF, PS, FB2, ODT, SXW, XLS. Ответ ещё сложнее: например, на сайте Debian есть список всех официально поставляемых программ к этой системе с короткими описаниями – на английском.

В семействе Linux распространены такие пакетные менеджеры.

пакетный менеджер	расширение имени файла пакета
apt	deb

20 Тут я опираюсь на знание пакетного менеджера АРТ.

rpm	rpm
pacman	pkg ²¹
portage	ebuild
entropy	tbz2

К сожалению, программная среда сложна, и работа пакетного менеджера тоже не совсем проста. Пакетный менеджер приходится знать всем домашним пользователям. На работе это может быть не нужно. Надо быть готовым разобраться с такими вопросами:

- как подключить репозиторий
- как использовать пакеты для прошлого выпуска ОС и бэкпорты. Бэкпорт – это пакет, обновлённый или созданный в старом выпуске ОС из-за его создания или обновления в новом выпуске
- как использовать уже полученные пакеты, а не запрашивать их из интернета
- как обновить программы
- как получить индексный файл (он может уже быть у вас)
- как обновить базу данных пакетного менеджера
- какие есть репозитории
- как уменьшить количество запрашиваемых пакетов
- что делать если пакетный менеджер отказывается ставить программу
- как настроить пакетный менеджер
- как подключить PPA
- как узнать сведения о пакете или программе
- как узнать какому пакету принадлежит файл
- как создать и поддерживать свой репозиторий (для социально-ответственных людей в регионах)
- как поставить или обновить много программ сразу
- как просто удалить ненужные пакеты (особенно нужно с микрокомпьютерами)
- как стереть ненужные пакеты в хранилище ОС (особенно нужно с микрокомпьютерами)
- как запросить пакеты на другом компьютере без установки их там
- как узнать где находится нужный пакет – допустим, на ваших DVD

Особые случаи установки программ.

Это достаточно трудная тема. Так как эти особые случаи могут быть разными. Но это важно. Так как иногда нужна программа, которой нет в репозитории, или PPA. Или у ваших знакомых. Ниже рассматривается ситуация в UNIX.

Выше я сознательно ограничил ситуации использованием пакетов из официальных репозиториях. Теперь исправим небольшую неточность. У читателя могло создаться впечатление что лишь пакеты готовых программ находятся в репозиториях. Это не так. Пакетами выше назывались в основном файлы с компилированными программами, готовые к управлению пакетным менеджером.

Придётся объяснить что такое компилированная программа. Если программа создана на таком языке программирования как C (читается «си»), то она сама по себе не может выполняться. Она является не более чем текстом на этом языке. С точки зрения возможностей полезного управления устройством она так же бестолкова как рассказы Зоценко. Лишь после компиляции этого *исходного текста* программы мы получим полезную, работающую программу. Каждый пользователь должен понимать разницу между текстом программы, который полезен после компиляции, и текстом программы, полезным непосредственно – как в случае с программами на языке **Python**, или любом другом интерпретируемом языке. Что это такое? Некая программа, называемая интерпретатором, получает текст программы, и, читая его, начинает выполнять эту программу. Так происходит, например, с программами на языке **Rexx**, который считается нетрудным для освоения, как и **Python**. Файл с программой имеет расширение, напоминающее название языка программирования на котором написана программа.

21 Эти символы могут быть не последними в имени

Расширение	Язык
py	Python
tcl	TCL
sh	BASH, SH
rb	Ruby
php	PHP
pl	Perl
js	JavaScript
ex	Euphoria
rex	Rexx

Пакеты, или порты именно *программного текста* являются альтернативой прежде рассматриваемым пакетам. И они тоже управляются пакетным менеджером. Эти пакеты контролируются для избежания программных конфликтов, и программу легко удалить.

Либо программные тексты поставляются для ручной установки программ.

Пользователю может быть надо вручную обеспечить свою программную среду компилятором или интерпретатором.

Так какой смысл иметь два способа установки и распространения программ? Обычно обсуждают установку пакетов скомпилированных программ. Это простой путь. Но не всегда лучший. Почему иногда лучше использовать некомпелированную программу? Основная причина в том что иногда так можно уменьшить количество передаваемых данных. Кроме того, некоторые программы просто не поставляются иначе. Почему? Допустим, из-за того что программист не хочет выполнять лишнюю работу. Ведь программа нужна не ему, а вам – вот вы и компилируете. Вы не можете диктовать ему как поставлять программу. Основная работа для человека – не компиляция, а создание исходного текста программы – уже сделана, и, как правило, вы не платите за этот текст. Однажды поставленный компилятор или интерпретатор, или библиотека пригодятся позже. Обычно есть файл, определяющий особенности будущей скомпилированной программы. Его может быть полезно отредактировать в случае старой техники – для более быстрой работы программы. Недостатки тоже есть. Во-первых, поставленную из исходного кода программу может быть трудно удалить. А без этого она может мешать установке других программ. Во-вторых, компиляция может занять часы времени. В-третьих, может быть программный конфликт. В-четвёртых, исходный код не всегда проверяется каким-либо ответственным лицом – майнтейнером, например.

Тем не менее, очень вероятно, что некоторым людям придётся использовать исходный код, то есть, по-английски, *source*. Обычно инструкция по компиляции прилагается к программе. Часто исходный код поставляется в архиве, и вам потребуется его вынимать из архива. То есть нужен архиватор. Это программа. Возможно, она уже есть. Так, например, сайт github.com позволяет получать исходный код в виде архива формата ZIP. Такой архив часто обрабатывается без установки дополнительных программ. Также этот сайт позволяет получать исходный код с помощью программы **git**. Сейчас github не то чтоб в моде – просто это удобно для программистов. Они ведь тоже люди. Их можно понять. Обычно программа на github содержит не только файлы исходного кода, но и все нужные для эксплуатации программы.

При вопросах обращайтесь к текстам интернета.

Портабельные программы.

То есть переносимые. Какой нам интерес рассматривать переносимые программы? Ответ:

- они не меняют – следовательно, не засоряют вашу программную среду
- они существуют как один файл, и его достаточно для установки и работы программы
- они не привязаны к одному пакетному менеджеру или ОС

Провинциалам портабельные программы особенно интересны. Их использование означает что один и тот же файл будет работать на разных ОС, и нет лишних сложностей с дополнительными программами в виде пакетов или портов. То, что это большой файл не важно – ведь он передаётся не через интернет.

Автор, пользуясь городским каналом связи, мало знаком с портативными программами.

Требования к портативной программе отчасти те же что к обычной. Это значит, что вопросы идентификации программы, и её обновления должны быть как-то решены. То есть нужна возможность убедиться что программа не вредна. В официальных репозиториях таких программ нет. Им можно доверять.

Мной создан набор наиболее нужных портативных программ для Linux. Для дальнейшего чтения информации о нём смотрите

http://pkprovincia.ucoz.ru/load/nabor_programm_dostatochnyj_dlja_mnogikh_rabot/1-1-0-60

Это набор программ под архитектуру i386. Я не знаю будут ли они работать на FreeBSD. Об использовании этих программ написано в интернете.

На сегодня, как мне показалось, наиболее полным репозиториумом обладает тип программ Flatpak. Чтобы использовать такие пакеты надо поставить пакетный менеджер для них – скорее всего, без необходимости удалять или настраивать штатный менеджер пакетов. Именно Flatpak позволяет получить в одном пакете всё что нужно для работы программы. Это заметно упрощает управление программной средой, но увеличивает размер пакета. Также вам потребуется некий программный модуль под вашу оконную среду, вернее, DE, который называется **runtime** – необходимые программы (возможно, программные библиотеки). Заметьте, само это слово имеет другой смысл, но в *терминологии Flatpak* имеет этот конкретный смысл. Экосистема Flatpak была раскритикована в том числе из-за проблем с обновлением программ.

По материалам интернета портативные программы требуют меньшего объёма файлов чем контейнеры, но это заявление нужно проверять в конкретной ситуации.

Кросс-системные пакеты.

Конечно, иногда одни и те же пакеты и репозитории пригодны для разных ОС – если они родственны. Тем не менее делались попытки создать универсальные пакеты для разных ОС. Это означает, что пакеты не дублируются, и для них требуется меньше места. Но это – забота разработчиков ОС. Кросс-системные хранилища пакетов представляют особый интерес для провинциальных пользователей. Кроме таких хранилищ, должен быть инструментарий для создания этих пакетов.

Создание портативных программ с помощью CDE.

CDE это программа для создания из обычных программ портативных программ. Автор программы утверждает что такие программы будут работать в течение ближайших лет после создания, и почти на любой ОС на основе Linux. Обращение с программой очень простое.

- Если у вас программы пока нет, то её надо скачать, переименовать в «cde», и сделать выполнимой с помощью **chmod**.

```
chmod u+x ./cde
```

- Находясь в каталоге с программой, запустить программу, которую надо сделать портативной – вот так – для программы **xclock**

```
./cde xclock
```

- Запущенная программа заработает. Её надо выключить не открывая в ней какие-либо файлы.
- При работе программы будет создан каталог **cde-package**. Желательно переименовать его используя название, номер версии, и разрядность создаваемой портативной программы.
- Запакуйте в архив созданный и переименованный каталог. Имя архива желательно сделать таким же как имя каталога, но добавится расширение. Тип архива должен быть популярным. В ОС на основе Linux со спартанской установкой может не быть архиватора **7zip**, но там есть архиватор **gzip**.

Теперь вы можете передать архив в интернет или на другой компьютер. Там достаточно развернуть содержимое архива, и запустить **xclock.cde** (для примера).

Безопасность и защита.

Вопрос защиты должен быть не просто понятен, но и являться постоянным предметом заботы администратора, и пользователя. Компьютерные устройства и сети оказались тем болотом, в котором водятся взломщики инфосистем, разрушители информации, и воры

информации. Не следует думать что ваша техника никому не интересна, или что обязательно кто-то лично пытается взлезть в ваш компьютер. Вторжение, как и защита, выполняется с помощью программ.

Автор не является специалистом по защите или безопасности (security). Обратимся к литературе чтобы указать способы защиты. К сожалению, капитальные труды в русском издании встречаются редко. Вместо них – так сказать, «попса».

Инфосистема, и устройство, а также контейнер или ОС нуждаются в защите. Защита включает в себя такие аспекты

- человеческий – люди не могут нарушать правила работы, люди не могут выдавать секретную информацию посторонним, люди же являются взломщиками защиты; охрана должна быть строгой
- аппаратный и физический – в первую очередь это физические препятствия к использованию и к доступу вроде хорошо запираемых помещений и аппаратных ключей для выполнения действий; недопущение использования посторонних устройств связи и носителей
- программный – сюда можно отнести все способы фильтрации, дублирования и шифрования данных, и обеспечение надёжности программ, а также способы анализа программной и человеческой активности
- сетевой – сети, и, особенно интернет представляют угрозу так как позволяют издали проникать в инфосистему, и скрывать своё местонахождение и другие данные для определения того кто совершил вторжение

Все эти аспекты прежде всего важны для бизнеса и работы с секретной информацией. Но их надо иметь в виду всем. Полезно ставить себя на место вредителей и воров информации, и предвидеть ситуации у рядовых работников. На Западе нанимают взломщиков чтобы видеть недостатки инфосистемы.

Пароли. Двухфакторная аутентификация.

Типичная и классическая ОС обеспечивает парольную защиту, и требует ввода имени пользователя и его пароля для работы. Она же обычно обеспечивает скрытие экранного изображения при отсутствии работы, и требует ввода пароля для получения на экране пригодного для работы изображения. Но из-за интернет-кафе и беспечности людей это так не всегда. Российские библиотеки сейчас тоже могут стать чем-то вроде таких мест, где пользователь как личность не определён. Чтобы улучшить ситуацию нужно регистрировать людей по имени пользователя, которое в базе данных библиотеки связано с паспортными данными. Регистрацию лучше автоматизировать. Начало вредной активности может не совпадать с периодом использования техники, но регистрация всё же улучшает положение.

Степень пригодности, или стойкости к подбору пароля зависит от его длины и набора символов. Чем больше разнообразие символов, и длиннее пароль тем лучше защита.

Далее процитируем Википедию: «Многофакторная аутентификация (МФА, англ. multi-factor authentication, MFA) — расширенная аутентификация, метод контроля доступа к компьютеру, в котором пользователю для получения доступа к информации необходимо предъявить более одного «доказательства механизма аутентификации». К категориям таких доказательств относят:

- Знание — информация, которую знает субъект. Например пароль, ПИН-код.
- Владение — вещь, которой обладает субъект. Например электронная или магнитная карта, токен, флеш-память.
- Свойство, которым обладает субъект. Например биометрия, природные уникальные отличия: лицо, отпечатки пальцев, радужная оболочка глаз, капиллярные узоры, последовательность ДНК.

<...>

Согласно мнению экспертов, многофакторная аутентификация резко снижает возможность кражи личных данных онлайн, так как знание пароля жертвы недостаточно для совершения мошенничества. Тем не менее многие многофакторные подходы аутентификации остаются уязвимыми для «фишинга», «человек-в-браузере», «человек посередине».

<...>

Выбирая для системы тот или иной фактор или способ аутентификации, необходимо, прежде всего, отталкиваться от требуемой степени защищенности, стоимости построения

системы, обеспечения мобильности субъекта.».

Способы контроля пользовательской активности и защиты.

- Просмотр журнала входов в систему. Это позволит как минимум отсеять людей которые, скорее всего, не имеют отношения к рассматриваемому случаю (подозрительному действию). Используйте команду **last**.
- Сбор данных программ, показывающих меняющиеся данные в терминале или консоли. Эти данные – по сути строки. Их можно записывать в файл перенаправлением.
- Просмотр поданных команд. Если все личные каталоги находятся в каталоге **home**, и используется **BASH**, то

```
grep -e "XXX" /home/*/.*bash_history
```

где XXX – интересующая команда.

- Просмотр сетевой активности. Используйте программу **atop**, **Nethogs**, **ntopng**.
- Просмотр процессов. Используйте программу **top** или **ps**.
- Просмотр логов. Используйте программу **swatchdog**.
- Определение открытых файлов. Используйте программу **lsof**.
- Обновление ОС и программ. Обновление (update) – фактор защиты данных, и работоспособности любой инфосистемы, и программной среды вообще.

Обновления.

Обновления программ могут стать проблемой для негородских инфосистем и частных пользователей. Их достаточно бессмысленно делать заказом диска по почте так как почта работает слишком медленно. В городах объём и набор файлов, получаемых для обновлений не важен. Там обновления можно (или даже правильно) делать автоматическими и полными, хотя применение этих файлов даже так требует части вычислительной мощности. Таким образом горожане, и, возможно, жители пригородов могут вообще не думать об обновлениях в рамках их версии операционной системы.

Сейчас я попытаюсь помочь тем кому обновления трудны. Что надо сделать:

- ограничиться обновлениями для безопасности (security update)
- использовать более совершенные методы копирования файлов и обновления программ
- использовать небольшие программы которые требуют небольшого объёма обновлений
- использовать лишь действительно нужные программы чтобы не обновлять остальные
- по возможности использовать пакеты или порты исходного кода
- использовать виртуализацию чтобы со стороны интернета – даже через локальную сеть – была видна лишь программная среда в виртуальной «песочнице», но не основная среда. Это примерно то же что пускать гостей лишь на крыльцо, не впуская в дом. Об этом я нигде не читал, но преимущества велики – не надо много обновлений, и, похоже, даже повреждение виртуальной среды не очень важно.

Получение обновлений может происходить либо с помощью пакетного менеджера, либо иначе – смотря по тому как доступна программа, или файлы для её обновления. Если, скажем, вы компилировали программу, то очень вероятно что придётся компилировать её снова для обновления. Если программа существует как потрабельная или находится в контейнере то обновить её с помощью пакетного менеджера ОС невозможно.

Своевременное обновление программ – наверное самая проблемная тема для пользователей вне городов. Поэтому рассмотрим её более подробно на примере Debian. Наиболее часто обсуждаемый список репозиторийов есть в файле **/etc/apt/sources.list**. Там для репозитория указан не только его адрес, но и обозначение его назначения. Так, например, у меня репозиторий с адресом **http://security.debian.org/debian-security** обозначен как **buster/updates**. При использовании программы **apt-get** можно прописать этот репозиторий с помощью ключа **-t**. В данном случае вы получаете файлы классическим образом, целиком таким образом:

```
apt-get update -t buster/updates
```

Если вы хотите лишь попробовать или проверить этот способ без получения лишних файлов можно использовать симуляцию с помощью ключа **-s**.

Администратор может вручную редактировать файл **/etc/apt/sources.list**.

Обновление расширений браузера.

Выполняется либо благодаря настройке расширения, либо путём установки свежей версии расширения.

Обновление браузера.

Можно осуществить простой установкой его. В принципе возможно организовать сервис для уменьшения количества скачиваемых данных при этом, но это лучше делать как общедоступный сервис. Это были бы хорошие дела большого бизнеса.

Сведения браузера.

Браузер может хранить пароли и другую важную информацию. Чем меньше такой информации он хранит тем лучше.

Беспроводные сети опасны.

Беспроводные сети стали очень распространены, но это может быть от безопасности людей. Перехват их данных – реальная угроза безопасности. Напомню, что такую сеть умеет создавать любой смартфон. Можно купить экранирующий материал для стен, который заблокирует работу смартфонов, и телефонов.

Обновление антивирусных баз.

Обновление или первый запрос антивирусной базы – это то на чём не надо экономить. Передача данных антивирусной базы в условиях плохого канала в интернет – реальная проблема, и она обусловлена плохими каналами связи. Администратор должен добиться чтобы не было нужды делать это для каждого компьютера заново. Для этого можно использовать антивирусную программу на компьютере, с помощью которой локальная сеть получает выход в интернет. Антивирусная программа может быть платной. По крайней мере в организации на этом не надо экономить. В принципе возможно организовать сервис для уменьшения количества скачиваемых данных при этом, но это лучше делать как общедоступный сервис. Это были бы хорошие дела большого бизнеса.

Обновление программ роутера или точки доступа.

Если это не компьютеры, то, скорее всего, обновление достаточно просто в случае пользовательского, не предназначенного для фирм оборудования. Обновление желательно сделать автоматическим. После обновления параметры работы устройства должны сохраниться, или их надо настроить заново автоматически до использования устройства.

Брандмауэр (firewall) и роутер или точка доступа.

Сетевое оборудование может, как и брандмауэр, защищать инфосистему, фильтруя данные, а также собирая статистику сетевой передачи данных.

Временные файлы.

Нужно ограничить возможности и срок существования cookie-файлов браузера. Также любые временные файлы, часто автоматически создаваемые программами при своей работе могут представлять опасность.

Антивирусная проверка.

Такая проверка позволяет найти опасные программы, и является частью работ по восстановлению инфосистемы, как и частью профилактики проблем.

Антивирусная проверка как проверка файлов лучше всего выполняется в начале рабочего дня, или, лучше, заканчивается до его начала. Продолжительность проверки может стать отдельной темой так как большие файлы могут долго проверяться.

Запрет на установку и использование программ, доступ к устройствам и сведениям о местоположении.

Таковыми устройствами являются, в частности, микрофон и видекамера, доступ к которым может дать браузер или программа для видеотелефонии. Также все устройства для чтения посторонних носителей можно отключить на многих компьютерах. Это, в частности, читатели дискет, лазерных дисков и USB-порты. Последние могут читать содержимое телефонов, флэш-брелки и кардридеры для карт памяти. Отключение доступа надо делать так чтобы его не мог

включить рядовой работник. Компьютерный администратор отвечает за защиту инфосистемы.

Браузер или другая программа может собирать данные о местоположении. Такие данные не надо выдавать. Чтобы решить эту проблему можно использовать виртуальную сеть, при которой программы будут дезинформированы.

Доступ плагинов к браузеру, или его расширений, к компьютеру должен быть закрыт. Или, лучше, вообще запрещена их установка.

То же верно про установку программ любым способом – рядовой работник не может этого делать. Запрет исполнения программ Java-Script, Java, любых лишних компиляторов и интерпретаторов может быть нужен. **Perl, Python** на сегодня отключить может быть трудно так как они нужны ОС в рабочем состоянии.

В среде UNIX устройства представлены как файлы, и права для них можно регулировать.

Сайт организации.

Сайт, как и другой сервис, может быть опасно иметь на технике организации. Это лучше делать на арендованной технике или сервисе VPS/VDS. Некоторым организациям вполне можно ограничиться несколькими страницами сайта и доступом к сайту через **SSH**, но их содержимое должно быть верным.

Резервирование данных.

Наилучшей защитой данных обладают датацентры, но, во избежание прочтения ими данных их нужно шифровать. Это не отменяет использование RAID и\или дополнительного носителя в организации. Передача данных через интернет означает что хорошее резервирование пока возможно лишь в городах и некоторых пригородах. Услуги хранения данных – платные. Экономить на них не надо.

Перенос части инфосистемы на специализированный сервис.

В условиях проблем с хранением и передачей данных может быть правильно сделать это. Перенести всю инфосистему нельзя так как рабочие места будут обычно там же где работники. Мотивация, как мне кажется, в том, что передача объёмных и ценных данных достаточно быстро нерентабельна. Рентабельно иметь небольшой поток зашифрованных данных к рабочим местам и от них по мере надобности. Это может касаться того какова база данных организации, и как она меняется. Спутниковый интернет, возможно, обеспечивает высокую скорость подачи данных из интернета, но не на сервис. Под Москвой есть много фирм, которые заявляют, что обеспечивают высокую скорость передачи данных через интернет.

Прочитав Википедию – «Другим ключевым фактором является уровень обслуживания SaaS. SaaS-провайдер способен предложить уровень обслуживания и поддержки ПО в работоспособном состоянии, недоступный для внутренних IT-отделов компаний. ... Многие заказчики опасаются применять SaaS из-за соображений безопасности и возможной утечки информации со стороны поставщика SaaS-услуг. Вопросы, связанные с безопасностью, ограничивают использование SaaS-модели в критически важных системах, в которых обрабатывается конфиденциальная информация. С другой стороны, ответственность за утечку информации со стороны разработчика обычно регламентируется соответствующими договорами, и вероятность такой утечки часто ниже, чем при использовании собственных внутренних систем. Этому, помимо прочего, способствует недоступность программно-аппаратного комплекса, на котором развёрнута система, сотрудникам компании.».

Контроль за работниками.

Работники сами представляют опасность даже если они просто устали. Один из путей увеличения надёжности и уменьшения угроз – давать объектам работы понятные имена, и вовремя их менять. Видеонаблюдение и сопоставление информации помогут предотвратить проблемы.

Напутствие директорам.

В нашей стране директора, как египетские пирамиды, служа не практическим, а ритуальным целям, своей безграмотностью представляют угрозу работе сотен и тысяч людей. Нужно поднимать и поддерживать на мировом уровне свою квалификацию – ничего другого инфосистема не позволяет, и помнить, что любое решение может касаться работоспособности,

рентабельности, и защиты данных организации.

Компьютерный администратор должен быть ответственным человеком. Он не может рассчитывать на снисходительное отношение к своей работе. На курсах, литературе нельзя экономить. Перерыв в работе организации может означать потерю клиентов, или даже судебный иск. По крайней мере надо предупредить клиентов о перерыве.

Сведения этой главы недостаточны для того чтобы директор почивал на лаврах. Фирма Oracle пошла намного дальше автора, и обеспечила свою ОС Oracle Linux не только модифицированным ядром, но и долгим тестированием в условиях эксплуатации. Будущее инфосистем – использование ОС с микроядром, сбой в котором не так разрушителен, но само ядро работает медленнее. Пересборка ядра для ускорения или по другим причинам возможна, и описана в интернете. Также интересна ОС NetBSD, но она может никогда не поддержать технику российского производства.

Права доступа, владение и UMASK.

Права доступа – это права на операции с файлами и каталогами. Именно так выглядит классическая тема прав для UNIX. Права доступа есть и в среде Windows, но там пользователю не надо о них знать.

Права имеют в среде UNIX прямое отношение к работе программ, и доступности файлов. Неправильно выставленные права – пользователем или программой – приводят к неожиданным трудностям, и могут создавать опасные ситуации так как избыток возможностей есть путь к злоупотреблению этими возможностями.

Что это за права? Это права на чтение, запись и выполнение. Выполнение – это работа для программ, и открытие для каталогов.

Как посмотреть какие выставлены права? Простой способ – с помощью команды

```
ls -l
```

В данном случае команда показывает права, и другую информацию о текущем каталоге. Например, так.

```
итого 2156  
-rw-r--r-- 1 user user 2204178 окт 26 07:52 sc34576401.pdf
```

В данном случае для файла **sc34576401.pdf** выставлены права, обозначенные как **-rw-r--r--**, то есть владелец файла (user) может писать и читать файл (read и write), члены его группы могут лишь читать файл, остальные могут лишь читать файл.

Кто выставляет права? Пользователь в своём каталоге. Администратор – где захочет. Программа – если она сама имеет нужные права. Для смены прав используется команда **chmod**.

umask устанавливает права и режим доступа к файлам в файловой системе. Какое это имеет для нас значение? Большое. Так как может быть нужно, например, раздать через Wi-Fi некоторые файлы. Пишу об этом так как в России это является простейшим способом бесплатно получить нужный файл – не запрашивая его в интернет. Пользователи в провинции, делитесь несекретными данными и программами! Не думайте лишь о себе.

Учётные записи и регистрация пользователей.

В Windows, как и в UNIX, что-то делать в системе может лишь вошедший в неё человек. В том числе администратор. Для этого он должен иметь регистрацию в конкретной ОС. Это – не то же, что регистрация на сайте, или в почтовом сервисе. Кроме администратора, в среде всех многопользовательских ОС может быть намного более одного пользователя. Например, при семейном компьютере нет программных ограничений, мешающих создать для каждого члена семьи свою регистрацию. Так и должно быть. Иначе чужие файлы станут доступны, и, что ещё хуже, могут быть отредактированы, украдены, или стёрты. Это особенно опасно в случае когда в семье малолетние дети, и работающие дома на компьютере родители. Windows просто не даёт войти в чужие каталоги, а в UNIX обычно запрещено изменять чужие файлы и каталоги. Но даже их просмотр может быть опасен, как и копирование. Поэтому стоит подумать об этих опасностях, и, при надобности, отрегулировать права.

Регистрация выполняется администратором. Он задаёт не только имя пользователя и его пароль, но и многие другие важные свойства учётной записи. Если на работе надо выполнять регистрацию, то может быть надо не собирать уже созданные пароли и имена, а – во

избежание злоупотреблений – создать новые. Каждый пользователь должен знать свои имя и пароль, и хранить их в тайне.

Один пользователь может иметь более одной учётной записи, но с точки зрения ОС это два разных человека.

Крайне важно иметь непредсказуемый – даже для программ – пароль. Проблема в том что его трудно запомнить. Длинные пароли труднее программно подобрать. Обычно длина пароля должна быть не менее 6 символов. В пароль не только можно, но и нужно вставлять цифры, и другие неожиданные знаки. Пароли всегда задаются латиницей. Как и имена. Имена не могут напоминать паспортные имена.

При обращении к ОС по сети важно, чтобы имя и пароль были зашифрованы, как и все передаваемые данные. Это – лишь часть средств защиты, необходимых при выполнении платной, а тем более секретной работы. Все директора должны постоянно помнить о защите данных, и промышленном шпионаже.

В UNIX пользователь может входить в группу пользователей. Это может быть полезно при большом количестве пользователей. Тогда не надо их перечислять, а указать лишь группу. Кроме того, члены группы могут иметь привилегии по правам. Это важно на работе.

Такие альтернативные средства идентификации как лицо, или отпечаток пальца, мне представляются недостаточно хорошими средствами защиты.

Администратор может узнать кто вошёл в систему, и посмотреть на последствия его действий. Также он может прочесть историю поданных команд.

Выполнение заданий.

В данном случае под заданиями понимаются не задания людям, а программные задания – выполняемые автоматически, и – как принято в литературе по UNIX – не все, а лишь выполняемые по расписанию, или в определённом порядке. Зачем это нужно? Простейший ответ в том что компьютером некому управлять. Это значит, что многие процессы, требующие ручной работы, не произойдут. Тем не менее многое можно заставить работать. Заметим, что есть средства автоматизации управления через оконную среду. Но тут они не рассматриваются.

Так, например, можно представить себе пользователя, желающего получить письма электронной почты ночью, когда обычно скорость существенно выше (но не больше лимита оборудования или тарифа). Пользователь не захочет вставать в 4 часа утра чтобы сделать что-то. Компьютерное устройство должно как минимум быть в спящем режиме до 4 часов, но проснуться в это время. В таком состоянии оно почти бесшумно. Задания широко используются в автоматических инфосистемах, стремящихся быть описанными словами «настроил, и забыл».

На UNIX классическим, наиболее описанным способом выполнения заданий является планировщик **cron**. Это – демон, неспособный вывести компьютер из спящего режима. Но способный либо запустить скомпилированную программу, либо скрипт. Скрипт, то есть текстовая программа, может выполнять разные действия. Всегда надо продумать её работу так как могут измениться обстоятельства. Пользователь может не смотреть на экран, и не знать что происходит. Заметим, что для выдачи звукового сигнала встроенного динамика может потребоваться программа, которую нужно отдельно ставить. Я имею в виду **beep**. Во многих случаях управление такими звуковыми сигналами очень просто, и состоит в запуске этой программы. Можно представить более развитую систему оповещения, требующую настроенной звуковой системы, или посылающую сообщения на другой компьютер или телефон (SMS).

Пробуждение компьютера выполняется настройкой в BIOS, или получением «магического пакета» из сети. Уход в спящий режим регулируется средствами управления питанием ОС. Они – при развитии DE – наглядны.

Задания для **cron** определяются в текстовом файле. Для его редакции не нужна оконная среда. Пример такого файла.

```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task
```

```

#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/

```

Тут ни одна строка не стартует чего-либо – так как все считаются комментариями. Они начинаются с решётки. Если мы сотрём решётку с последней строки, то получим строку задания. Но, чтобы оно работало, файл надо сохранить.

В данном случае программа **tar** запускается ежедневно. Всё, что до «**tar**», является указанием времени и даты задания, а после – параметрами программы **tar**. Программа стартует ровно в 5 часов каждый день. Файл заданий привязан к пользователю, и его правам. Пользователь не может стартовать программы администратора, или он должен использовать команду **sudo**.

Всегда проверьте выполнение задания. Можно задать задание на время когда вы увидите как оно работает. Либо посмотреть лог **cron**.

Настройка сети и роутер или точка доступа.

Эта тема всё равно останется неполно описанной. В основном это так потому что сама тема настройки сети, или компьютерных сетей достаточно плохо освещена в русскоязычной литературе. То, что компьютерные сети «пришли» к нам с Запада – факт, но не большое оправдание. Поэтому дилетантизм остаётся.

Если вы хотите просто «чтобы всё работало через полчаса», то в городских условиях это достижимо достаточно просто. В начале обслуживания провайдером *выделенных линий связи* монтажники могут работать бесплатно – в соответствии с политикой фирмы. Они настроят вам всё что должно работать в пределах квартиры.

В условиях пригорода вероятность получить информацию из интернета с помощью телефона велика. Благодаря этому вы – рано или поздно – настроите сеть. Или можно вызвать настройщика, и оплатить его работу.

В самом плохом положении окажется пользователь в глуши. Тем более что самой простой в понимании линией связи является не работающая там связь через сетевую (Ethernet) плату. Поскольку там нет выделенных линий. Есть подключение спутникового ресивера. Но я не думаю что это просто.

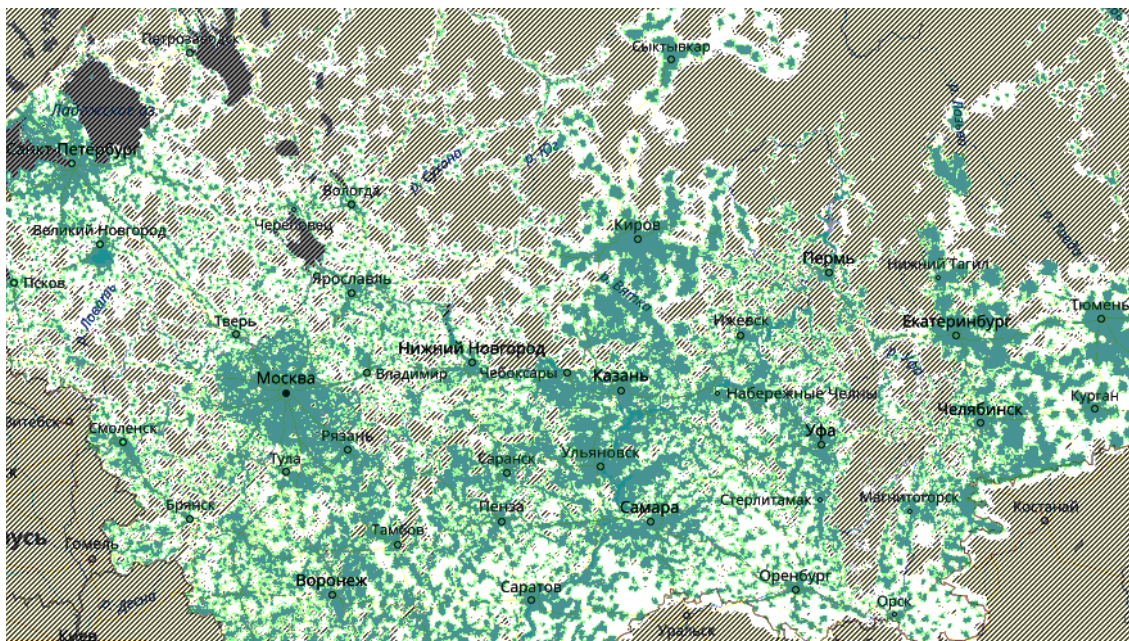
Надо различать выход в интернет конечного пользователя, и вопросы настройки сети – возможно, локальной – у администратора. Конечный пользователь – это обычный пользователь, не занятый вопросами администрирования. Ему достаточно выйти в интернет. После этого при наличии вопросов там он может искать ответы.

Ситуация администратора совсем другая. Ему также нужен выход в интернет – хотя бы для получения помощи. Но ему могут потребоваться знания по теме настройки ОС и программ для работы, например, локальной сети.

В наше время часто роутер обеспечивает локальную сеть в рамках квартиры или дома. Это настоящая локальная сеть. Но пользователю не нужно ей заниматься, или есть инструкция от фирмы. С другой стороны, может быть ситуация, когда локальная сеть требует не только настройки, но и выбора оборудования, когда локальная сеть более сложна, и трудно или невозможно вызвать мастера. Сайт pkprovincia.ucoz.ru в большой степени ориентирован на подобные ситуации. Можно сказать – ситуации вдали от городов. Я сомневаюсь, что за приемлемые деньги кто-то посторонний будет выполнять все эти работы.

Что такое локальная сеть, и зачем она? Это средство обеспечить связь между компьютерными устройствами – в том числе телефонами – в рамках сети. Классическим устройством является настольный или переносной компьютер. Примитивный телефон часто не способен быть частью локальной сети. Передача данных в локальной сети является

альтернативой передачи данных через интернет. Но она не отменяет потребности иметь выход в интернет. Даже развитые телефоны нуждаются в этом – даже без локальной сети – для обновления своей ОС и программ. Если скорость передачи из интернета за городом часто совершенно недостаточна, то в локальной сети ожидаемая скорость гораздо выше. Но она зависит от свойств конкретной сети, и преград – деревьев, домов. Посмотрим на карту сигнала для фирмы Мегафон на сегодня (январь 2019 года).



Тут синим (вокруг Москвы, например) показано наличие сигнала третьего поколения связи (3G) с теоретическим пределом скорости 7 мегабит. Даже если у вас и правда скорость будет равна 3-4 мегабитам, ясно, что пока в России много мест с медленной связью, или вообще без неё. Конечно, есть и другие фирмы. Тем не менее проблема остаётся, и становится всё острее по мере удаления от городов. Типично, что в пригородах в выходные дни скорость сильно падает.

Какие могут быть функции локальной сети на селе?

- Понятные местному населению.
- Посильные по деньгам.
- Нужные.

А именно?

- Конечно, прежде всего это бесплатная голосовая связь внутри села. Она будет особенно полезна зимой.
- Получение лент новостей. Это не всегда буквально новости.
- Поселковое радио в локальной сети.
- Поиск информации – в основном текста – в интернете. Также использование голосовых ассистентов.

Именно для этих функций не требуется высокая скорость. Тут достаточно иметь Wi-Fi сеть стандарта 802.11g, но сейчас не стоит покупать старый роутер или точку доступа. Современные модели способны работать намного быстрее.

Мало купить оборудование. Надо его настроить. Особый интерес представляют модели с возможностью подключения модема для выхода в интернет. Простейшая локальная сеть состоит из одного роутера или точки доступа. Но его радиус действия может быть недостаточен для охвата села. Можно создать несколько сетей. Некоторые точки доступа и роутеры могут работать как повторители сигнала, и увеличивать зону приёма. Также повторители продаются отдельно. Антенна влияет на радиус действия устройств.

Проблема в том что средний человек и средний роутер – две вещи несовместные. Одна из них довольно проста, другая – издевательски сложна. В инструкции может быть написано что-то вроде «нажми розовую кнопку и будь счастлив». Но это не правда. Базовая настройка может быть несложной. Более проблемны случаи когда этого мало. Возня с роутером может занять

много дней.

Радикальным средством решения проблемы низкой скорости входящих (или любых) данных является спутниковая связь. Но оборудование в фирмах дорого. А «народные» средства вроде DVB-платы означают что вы сами должны всё настроить, и то – при понятном сигнале, не зашифрованном фирмой спутниковой связи. Из фирмы приезжает монтажник, он же выполняет настройку и проверку передачи данных. Оплатить такую связь можно коллективно. Это означает коллективное использование выхода в интернет.

Настройка соединения.

Наверное, прежде всего надо позаботиться о выходе в интернет с мобильного телефона. Это не обязательно смартфон. Рассмотрим дешёвый телефон, не являющийся смартфоном. Обычно в инструкции написано что данная модель телефона способна выходить в интернет. Инструкцию можно распечатать в библиотеке или интернет-кафе.

Рассмотрим соединение по модему или через мобильный телефон подключённый к компьютеру. В случае оконной среды этот вопрос совсем прост. В худшем положении окажутся администраторы и пользователи без оконной среды. Фирмы-поставщики услуг связи могут отказаться объяснять настройку соединения в бесплатных ОС. В таком случае после подключения телефона к компьютеру обычно в среде UNIX для подключения используется встроенная в оконную среду программа, где надо определить

- страну
- оператора мобильной сети
- шлюз

Всю эту информацию имеет фирма выпустившая SIM-карту в вашем телефоне.

Если такая возможность отсутствует то используется аналогичная консольная программа.

На более низком, «подвальном» уровне модемы и телефоны обычно управляются AT-командами. Они описаны в литературе, например, в книге «AT-команды. Руководство по использованию AT-команд для GSM/GPRS модемов» размером 432 страницы. Всю эту книгу вряд ли стоит читать и печатать. Городские библиотеки могут выдавать такую книгу в читальный зал, давать её домой нет смысла. В этой же книге есть описание других полезных команд.

Рассмотрим роутер TL-MR3020 – помня что это только пример, и роутеры могут сильно отличаться друг от друга. Роутер способен выходить в интернет через USB-модем. Модем достаточно подключить. В данном случае имеется в виду модем поколения 3G или 4G, это беспроводной модем. Модем питается током через разъём USB. Формально роутер способен передавать данные со скоростью до 150 мегабит, но скорость не выше реальной в конкретных условиях – в зависимости от силы сигнала и загруженности ближайшего оборудования, которое работает примерно так же как радиостанция или Останкинская телебашня. Роутер имеет разъём Ethernet, с его помощью проще всего подключить компьютер. Смартфон или планшет проще всего подключить через создаваемую роутером Wi-Fi сеть. Роутер можно подключить к интернету через кабель. Для настройки роутера используется браузер, операционная система не принципиальна. Следовательно это может быть Linux. Надо понимать что работа интернета нормирована протоколами связи, и эти протоколы поддерживаются почти любой ОС.

Вид роутера.



Роутер не размещают на прямом солнечном свете. Роутер может работать в трёх режимах.

- В режиме маршрутизатора он раздаёт интернет-доступ на устройства в локальной Wi-Fi-сети с модема.
- В режиме беспроводного клиента роутер раздаёт доступ в интернет, при этом сам получая его от беспроводного устройства через Wi-Fi.
- В режиме точки доступа роутер работает либо для расширения проводной локальной сети с помощью своей Wi-Fi-сети, либо расширяет уже существующую Wi-Fi-сеть, либо соединяет две локальные сети, либо позволяет проводное подключение для компьютера.

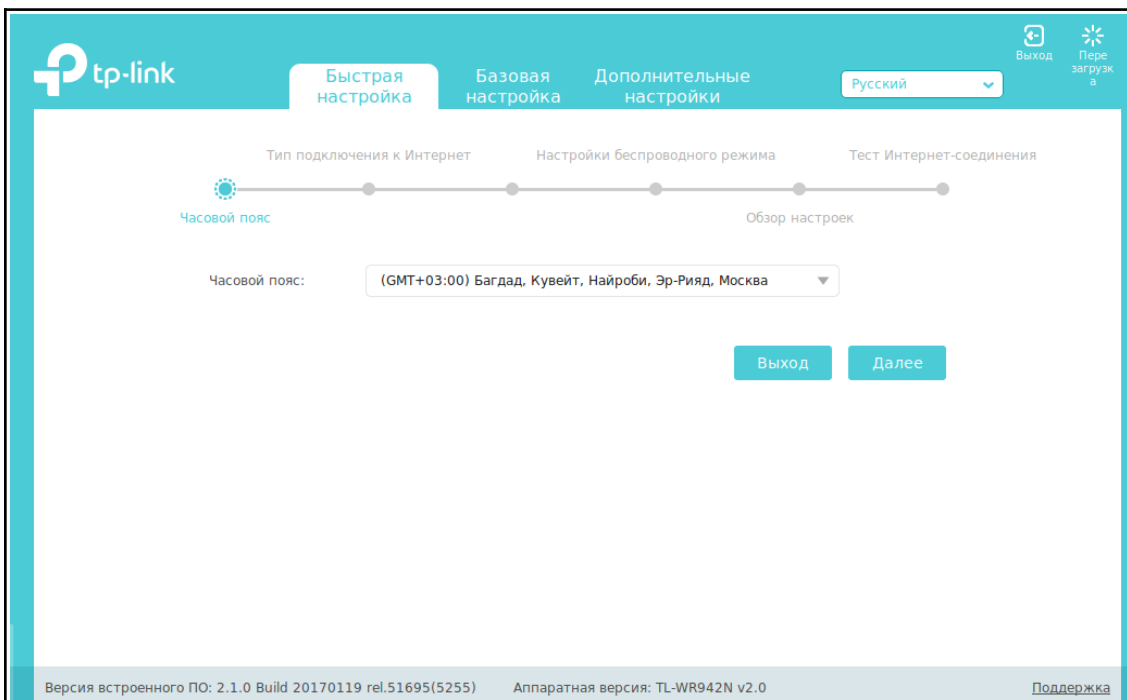
Для подключения в первом случае надо войти в беспарольную Wi-Fi-сеть TP-LINK_POCKET_3020_XXXXXX (где XXXXXX последние шесть цифр MAC-адреса вашего маршрутизатора), перейти по адресу <http://192.168.0.254>, ввести в качестве имени пользователя и пароля «admin», на странице быстрой настройки обозначить

- тип доступа в интернет (3G/4G)
- регион и провайдера
- имя локальной сети и пароль к ней

Нажмите кнопку или ссылку для перезагрузки роутера.

Не все модемы будут работать с роутером, список подходящих модемов – в интернете.

Эмуляторы программ роутера работают в браузере без роутера. Это позволяет ученику попробовать программу в которой обычно происходит настройка роутера. Чтобы это сделать смотрите сайты фирм-производителей роутеров и точек доступа. (см рисунок)



Проблемы соединения.

Следует различать

- отсутствие поступления данных,
- отсутствие нужной программы,
- отсутствие связи,
- разрыв связи,
- ненастроенность роутера или фаервола (или антивирусной программы) и
- ненастроенность ОС.

Ваш модем должен работать с вашей ОС, но это не связано с её официальной поддержкой этим модемом. Провайдер — то есть, например, МГТС в Москве, не обязан обсуждать вопросы связи если данные идут. Тем не менее всегда старайтесь узнать как можно больше. Мой личный опыт говорит о плохой работе 3G модема МТС в Подмосковье.

Всегда читайте целиком все документы об услугах и оплате.

Скорость передачи данных через интернет.

Это важно так как скорость может упасть, и так можно проверить выход в интернет. Если, скажем, в интернет-кафе скорость низкая, то не стоит оплачивать работу там. Скорость можно замерить в консоли или терминальной программе с помощью консольной программы speedtest-cli, распространяемой и описанной на сайте www.speedtest.net. Для этого в среде Debian или Ubuntu подайте следующие команды.

```
sudo apt-get install curl
curl -s https://install.speedtest.net/app/cli/install.deb.sh | sudo bash
sudo apt-get install speedtest
```

Затем вы можете измерить скорость так.

```
speedtest
```

Вот что я получил в результате – кроме заявления о лицензии:

```
Speedtest by Ookla
```

```
Server: MGTS - Moscow (id = 6827)
ISP: Moscow City Telephone Network
Latency: 14.61 ms (8.28 ms jitter)
Download: 0.92 Mbps (data used: 1.1 MB )
Upload: FAILED
[error] Cannot open socket
```


И действительно, интернет работал плохо — написано «Upload: FAILED».

Дисковые квоты.

Дисковая квота по смыслу не связана с диском. Вернее — это ограничения на суммарное количество файлов и каталогов (что обычно не существенно), или их объём для пользователя (что может быть нужно). Зачем нам квоты? Конечно, есть много случаев, когда квоты не нужны, и люди о них даже не знают. Но в случае малообъёмных, и при этом многопользовательских жёстких дисков или флэш-носителей, квоты позволяют избежать оккупации постоянной памяти неким узурпатором. Это особенно важно для микрокомпьютеров. Так как они используют карты памяти. На сегодня карта ёмкостью 64 гб стоит не менее 1000 рублей. Но этот объём — небольшой. Он может быть тесен даже для одного пользователя.

Программа для установки квот называется **quota**.

Полезно знать что есть возможность оповещать пользователей о близости квоты. В-первых, они могут сами запросить сведения об этом. Но человек — самая ненадёжная часть инфосистемы. Самая непредсказуемая. Поэтому лучше прицельно слать информацию — например, с помощью сообщений в оконной среде. При превышении квоты в консоли или терминальной программе уже выдаются сообщения **Disk quota exceeded**.

Подготовка носителей.

Подготовка носителя — это не освобождение места на нём. Просто стирание файлов и каталогов является заурядной операцией, для которой часто не требуется иметь права администратора. Тут подготовка носителей — это менее очевидная и простая работа. Она не часто выполняется, но, когда она нужна, то обойтись без неё не получится.

Думаю, что при наличии старой техники всё ещё используются дискеты ёмкостью 1,44 мб. Это не то чтобы плохо — особенно если ничего другого нет. Относительно дискет может быть выполнена операция форматирования.

Разумеется, в ходу лазерные диски. Большинство их не требуется и невозможно форматировать, но это зависит от их файловой системы. Для перезаписываемого диска возможна его очистка. Но она не является сама форматированием. Форматирование перезаписываемого лазерного диска может быть выполнено для смены его файловой системы.

Очень удобно иметь флэш-носитель — особенно брелок. Его можно форматировать. Вопрос о возможных для брелка файловых системах остаётся открытым.

Также жёсткий диск можно назвать носителем. Тем более если он — автономный, или находится в аппаратном хранилище. И для него определённа возможна операция форматирования, и смена файловой системы.

ВАЖНО — форматирование уничтожает всё что записано на носителе или логическом диске.

ВАЖНО — также смена файловой системы обычно означает уничтожение всего что записано на носителе или логическом диске.

Вопрос о восстановлении данных после этих операций открыт. Беспечным быть нельзя. Нет гарантий восстановления.

Что такое форматирование.

Само форматирование заключается в создании (формировании) структур доступа к данным, например, структур файловой системы.

В процессе форматирования также может проверяться и исправляться целостность носителя.

Лазерные диски пакетной записи.

Возможно, многие не знают, что такие диски бывают, и для такого диска возможно стирание не всех данных, а выбранных пользователем — как на флэш-носителе. Автор пользовался пакетной записью в среде Windows. Действительно, это очень удобно, хотя относительно надёжности хранения есть определённая критика. Очень вероятно, что придётся ставить некую программу для чтения и записи на такой диск.

Сейчас важнее, что без форматирования лазерного диска нельзя сделать его диском пакетной записи. В результате его файловая система сменяется на UDF.

Подготовка жёсткого диска.

Низкоуровневое форматирование обычно либо не нужно, либо не выполняется вполне. Если вы читаете что надо форматировать жёсткий диск, то, скорее всего, имеется в виду не это.

Разбиение на логические диски имеет на практике ту выгоду что вы можете очистить один из логических дисков не затрагивая другой. Также вы можете сменить файловую систему отдельного диска. Кроме того, иногда есть ограничения на объём видимых ОС дисков, поэтому они не могут быть слишком велики. Если вы используете бесплатную ОС, то вам потребуются иметь логический диск для временного хранения данных, и автоматического обращения с ними – своп (swap). В общем случае своп нужен. Если конкретная инструкция требует другого разбиения диска, то она важнее чем общее положение. Лучшим средством для операций с логическими дисками, то есть разделами, является лазерный диск **Gparted Live**, или аналогичный флэш-брелок.

Высокоуровневое форматирование записывает (формирует) логические структуры, ответственные за правильное хранение файлов (файловые таблицы), а также, в некоторых случаях, загрузочные файлы для разделов, имеющих статус активных. Высокоуровневое полное форматирование — процесс, который заключается в создании главной загрузочной записи с таблицей разделов и (или) структур пустой файловой системы, установке загрузочного сектора и тому подобных действий, результатом которых является *возможность использовать носитель в операционной системе* для хранения программ и данных. В процессе форматирования также проверяется целостность поверхности носителя для исправления (блокировки) дефектных секторов. Известен также способ без проверки носителя, который называется «быстрым форматированием». (по тексту Wikipedia).

В связи с загадочностью этих формулировок, и риском потери данных, всегда старайтесь больше узнать о том что вы хотите сделать, и пробуйте операции на пустых, ненужных, или виртуальных (в эмуляторе) дисках.

Резервирование.

Резервирование, или создание резервных копий – это копирование ценных файлов так чтобы минимизировать опасность их потери при разрушении данных. Это может произойти как из-за износа носителя, так и из-за ошибочных действий пользователя или вторжения злоумышленника. Исходный файл может быть файлом любого владельца, но на серверном компьютере наиболее осмысленно резервирование файлов, владельцем которых является администратор. Тем не менее на домашнем компьютере обычно ценными являются прежде всего пользовательские файлы – прежде всего студенческие, или связанные с работой – тем более если вы работаете дома. На компьютере может быть более одной ОС, и более одной системы резервирования, но они *не могут мешать друг другу*.

Что такое эта система резервирования? Это программная система, обеспечивающая прежде всего сохранность ценных файлов благодаря их копированию. ОС не обязана обеспечивать резервирование. Оно достигается благодаря установке программ и настройке. При создании системы резервирования нас интересуют следующие вопросы.

- Когда будет выполнено копирование? В лучшем случае оно происходит сразу после создания или изменения файла.
- Как быстро будет происходить копирование? Нужно чтобы копирование закончилось до следующего акта копирования. Или этот акт должен быть отложен. Для облегчения ситуации можно использовать программу **rsync**.
- Как много места займут копии? Для уменьшения размеров копий часто используется архивация. Также есть техника инкрементального резервирования, когда копируются лишь изменения. Заметим, что большие файлы на домашнем устройстве обычно не надо резервировать. Их обычно не надо менять, и достаточно просто скопировать их на лазерный диск. Напротив, на работе может быть нужно копировать большой файл базы данных. Повреждение или потеря базы данных сломает работу многих организаций. И этот файл постоянно меняется так как переписывается база данных. Возможно, стоит использовать стример – устройство для резервирования на магнитную ленту.
- Что за файлы будут скопированы? Как система резервирования отберёт нужные файлы? Наиболее интеллектуальная система отбора файлов – опирающаяся на нечто вроде

каталожной карточки к файлу. Обычно в домашней практике просто копируют файлы некоторых форматов из нескольких каталогов. На работе ситуация требует больше думать так как там часто используется некая инфосистема, и даже файлы, определяющие параметры её работы могут представлять ценность. Перерыв в работе инфосистемы из-за отсутствия или повреждения этих файлов может не только стоить денег, но и просто быть недопустим из-за особенностей назначения инфосистемы. Наиболее защищёнными от простоя инфосистемами являются системы, где обрыв работы означает ломку технологического процесса (заводы) или угрозу для жизни и здоровья (авиация и прочее подобное). Заметим, что в таких случаях не стоит экономить на RAID-массиве жёстких дисков, и обычная ОС может не годиться. Требуется повышенная надёжность ОС, и может потребоваться ОС реального времени. Например, Debian не является такой ОС. Хотя надёжность её высока, и как ОС, например, для основы сайта Debian вполне подходит. Как мне кажется, директора пока в массовом порядке не знают о таких вопросах. Они должны понимать роль инфосистем, и составить требования к инфосистеме. А компьютерный администратор должен помочь директору.

- Как понять когда были сделаны копии? Простейший способ обозначить дату копирования – вставить её в имя файла. Если не полагаться на дату редакции файла, хранимую файловой системой.
- Как найти нужные файлы копий? Для этого можно создавать список файлов. Но он не всегда достаточно информативен. Так, он не показывает изображений в файлах.
- Где именно будут храниться копии? Хранение копий в интернете может быть неприемлемым из-за опасности просмотра данных посторонними, или плохого канала в интернет.
- Потребуется ли какой-то особый способ восстановления копий, или поиска в них? Будет ли он выполнен в условиях сбоя? Особые способы могут потребовать специальных программ.
- Как будет происходить обновление данных для поиска?
- Как быстро можно восстановить работу сервисов – это для администраторов общественно-полезной техники? Напомню, что лазерные диски читаются медленнее чем флэш-память.
- Как избежать повторного хранения файлов в копиях.

Есть много программ для резервирования. Одна из них входит в Windows. Заметим, что большинство программ работают по расписанию, или после ручного запуска, не реагируя на изменение в файлах. Для автоматического резервирования сразу после изменения файла на Linux есть программа **inotify-tools**. На FreeBSD есть подобная программа.

После получения всей этой информации некоторые решат, что «как-нибудь пронесёт», или что «это не для средних умов». И не станут вообще заниматься резервированием. ВАЖНО – именно тут кроется опасность и ошибка. Гораздо лучше иметь примитивную, или любую систему резервирования, чем не иметь её вообще. Обычно в домашней практике есть некая резервирующая программа, она работает по расписанию, и назначено куда, что, и откуда копировать. Для организации такого примитива надо мало времени. Для бизнеса резервирование обязательно.

Интересные программы.

Назову несколько претендентов, имея в виду и домашнего пользователя, и применение на работе; но на работе нужно *выбирать* программу. Это делает администратор.

- **Time Vault**. Как и многие другие утилиты резервного копирования, этот пакет создаёт инкрементные резервные копии файлов (сохраняет только изменения относительного некоторого первоначального состояния — прим.пер.), которые позже могут быть использованы для восстановления. Его мгновенные снимки являются копиями директория в определённый момент времени. Снимки, сделанные для файлов, которые не изменились с момента предыдущего снимка, занимают очень мало места. Это связано с тем, что вместо создания резервной копии всего файла без его изменения, в снимках используется жёсткая связь с существующей резервной копией файла в её первоначальном состоянии.
- **Tardis** - это система резервного копирования в стиле TimeMachine, которая позволяет

производить инкрементное резервирование и зашифровывать backup-копии. Для домашних или начинающих пользователей сложновата. Скорее для работы. Хорошо описана на английском.

- **Bareos** (Backup Archiving REcovery Open Sourced) - это форк известного проекта bacula.org. Развивается с 2010 года и имеет много улучшений по сравнению с исходным проектом. Скорее для работы.
- **Deja Dup**. Любопытна она своим минималистичным функционалом и быстрой работой. На начальном экране Deja Dup всего две кнопки – Восстановление и Резервное копирование. И в дальнейшем интерфейс программы не отличается сложностью. Хранение резервных копий возможно на локальном жёстком диске в том числе и USB, удалённом ресурсе – FTP, WebDAV или на узле с доступом по SSH, также поддерживается облачное хранилище Amazon S3. Программа умеет шифровать и сжимать данные, имеет планировщик резервного копирования и функцию поэтапного резервного копирования, что позволяет восстановиться из любой конкретной копии.
- **dar** is a shell command that backs up from a single file to a whole filesystems. Хорошо описана на английском.
- **Back In Time** — простое и удобное приложение для локального резервного копирования (backup) и восстановления пользовательских файлов и каталогов.
- **Arkeia Smart Backup**. Цена программы пропорциональна объёму информации, который необходимо защитить, и если он меньше 50 ГБ, то программу можно использовать бесплатно.

(по текстам интернета, в основном по <http://linsoft.info>). Большинство указанных программ бесплатно.

Отслеживание процессов и управление ими.

Простоты ради можно считать что процесс – это экземпляр работающей программы. Само понятие процесса используется по-разному. Какой нам интерес заниматься процессами? Они ведь, наверное, сами как-нибудь там без нас происходят? Можно сказать – да. Тем не менее тема процессов практически важна. Почему это так? Причина первая – программа может «зависнуть», то есть перестать управляться вами (или вообще чем угодно). Причина вторая – программа может забрать слишком много ресурсов. Причина третья – вам может быть надо немедленно выключить устройство или освободить захваченный программой носитель. Не пугайтесь – «зависание» программы случается редко. Исключить в принципе такую ситуацию нельзя.

Приведу пример. Я создавал файл с помощью программы **AWK**. Я ожидал что она выполнит свою работу за пару секунд. Мне пришлось оборвать её работу когда я увидел что создан файл размером в сотни мегабайт, и он мне не нужен. Если не остановить такой процесс, то вся ёмкость логического диска может быть забита файлом. Работать с ним может не получиться. Мне пришлось оборвать процесс. Такая ситуация может быть из-за неправильно написанной или неправильно используемой программы.

Рассмотрим программы в UNIX. Классической программой для просмотра сведений о процессах является программа **top**. Она работает даже в консоли, и показывает сведения о текущих процессах. Эти сведения обновляются примерно раз в секунду. Вот пример работы программы.

```
top - 14:27:05 up 3:35, 2 users, load average: 0.33, 0.30, 0.28
Tasks: 125 total, 2 running, 123 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.2 us, 1.7 sy, 0.0 ni, 93.3 id, 0.0 wa, 0.0 hi, 0.8 si, 0.0 st
KiB Mem: 882772 total, 172976 used, 709796 free, 56084 buffers
KiB Swap: 102396 total, 0 used, 102396 free. 65320 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
626	root	20	0	104380	18416	11076	S	19.2	2.1	38:47.43	picam
949	root	20	0	0	0	0	S	1.7	0.0	0:04.03	kworker/u8+
71	root	1	-19	0	0	0	S	0.7	0.0	1:49.22	VCHIQ-0
78	root	-51	0	0	0	0	S	0.7	0.0	1:33.04	irq/92-mmc1
992	pi	20	0	5112	2528	2164	R	0.7	0.3	0:00.29	top
3	root	20	0	0	0	0	S	0.3	0.0	0:06.70	ksoftirqd/0
15	root	20	0	0	0	0	S	0.3	0.0	0:03.50	ksoftirqd/2
578	ntp	20	0	5688	3684	3248	S	0.3	0.4	0:01.44	ntpd
632	www-data	20	0	15868	3128	2164	S	0.3	0.4	0:08.27	nginx
1	root	20	0	23892	4016	2744	S	0.0	0.5	0:04.54	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:+
7	root	20	0	0	0	0	R	0.0	0.0	0:04.40	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0.0	0.0	0:00.01	migration/0
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/1

Наверное, самое интересное тут – столбцы CPU и MEM показывающие использование вычислительной мощности устройства и его оперативной памяти. К тому же видно какие процессы есть – например, **systemd**. Он запущен от имени администратора (**root**).

Другая подобная программа – **ps**. Она выдаёт список процессов, и этот список можно фильтровать для быстрого получения ответа на вопрос есть ли интересующий процесс. Вот пример работы программы, запущенной с ключом **-e**.

```
PID TTY TIME CMD
1 ? 00:00:02 systemd
2 ? 00:00:00 kthreadd
3 ? 00:00:00 ksoftirqd/0
5 ? 00:00:00 kworker/0:0H
7 ? 00:00:02 rcu_sched
```

Для фильтрации списка процессов можно применить следующую команду.

```
ps -e | grep nano
```

В данном случае мы ищем процесс **nano**, ожидая, что название процесса совпадает с названием программы.

Прежде чем обрывать процесс надо понимать последствия этого. Все данные – и файлы вашего устройства, и других устройств, и вовсе не записанные в файлы – перестанут обрабатываться оборванным процессом. Это, можно сказать, нештатная ситуация, которую вы сами создаёте. Обрыв обработки данных означает что они не будут вообще никак или правильно сохранены в файле или файлах. Обрыв процесса выполняется администратором с помощью команды **kill**. Все описанные программы работают в консоли и терминальной программе, и работают даже при установке только базовой части ОС.

Cpulimit

Эта программа позволяет ограничить использование микропроцессора конкретной программой. Это может быть нужно чтобы освободить вычислительную мощность для других программ. Программа достаточно просто управляется. Чтобы управлять процессом указав его PID-идентификатор дайте команду

```
cpulimit -p 1234 -l 50
```

Тут мы указали PID 1234 и уровень занятости 50%.

Чтобы ограничить нагрузку указав имя файла программы подайте команду

```
cpulimit -e foo -l 50
```

Тут мы указали программный файл **foo** и уровень занятости 50%.

Переключение между программами.

Пользователь может иметь в работающем состоянии всего одну требовательную к производительности техники программу, и переключение между программами станет медленным, и может раздражать. В UNIX есть средство сделать переключение более быстрым. Программа **nice** определяет степень способности программы уступить место другой программе, то есть сменить активное окно. Если программа запущена без явного указания для быстрого переключения – например, через меню, то уровень «уступчивости» такой программы равен 10 в диапазоне от -20 до 19. Обычно этого достаточно при условии что ваша ОС не слишком «тяжела» для техники. Пользователь без прав администратора может регулировать значение уступчивости в интервале от 0 до 19, иначе будет сообщение о недостатке прав.

Может потребоваться увеличить уступчивость для уже работающей программы. Для этого используется программа **renice**. Пользователь без прав администратора может увеличивать значение уступчивости в интервале от 10 до 19, иначе будет сообщение о недостатке прав.

Установка уступчивости делается с помощью ключей **n** и **u**:

```
nice -n 15 nano  
renice -u -19 nano
```

Демоны.

Понятие демона (daemon) отсутствует в Windows, но оно часто используется в UNIX. Что это такое? Демон – это автоматически запускаемый процесс, который, как можно ожидать, полезен, и часто из-за своих функций должен постоянно работать в интересах пользователя – в том числе администратора. Демон обычно не выдаёт на экран окон или индикаторов. Тем не менее он работает. Что это может быть за программа? Если рассматривать спартанские системы, то в первую очередь приходит на ум **cron** – планировщик заданий. На компьютерном сервере демонов может быть много – для разных целей. Так, например, веб-сервер может быть демоном. Вы можете не видеть окна сервера, но он работает. Заметим, что сервер как компьютер обычно вообще не имеет оконной среды. Простейший, и несовершенный веб-сервер уже встроен в **python**, и в его работе – после запуска – можно убедиться, запросив в браузере адрес **http://localhost:8000**.

Журналы (логи).

Логи, или логфайлы – это файлы, хранящие сведения о работе и данных какой-либо программы. Так, например, веб-сервер **lighttpd** может сохранять эти сведения в лог. Зачем? Чтобы вы в удобное вам время, или в любое время могли узнать что происходило. Заметим, что некоторые программы очищают свои логи, или придерживаются сложной политики их создания и редакции. Поэтому может быть нужно делать резервирование логов – смотрите соответствующую главу. Содержание логов часто можно настроить. Журнал вашего браузера не считается логом.

Нечто вроде лога можно организовать для полезного применения самостоятельно. Скрипт может, например, регулярно проверять некое значение или состояние, и записывать в файл. Это можно достаточно просто сделать. Благодаря этому нет надобности постоянно кому-либо следить за состоянием инфосистемы. Но тут важно понимать, что совсем без внимания персонала ни одна инфосистема не обходится.

Фаервол (брандмауэр).

Брандмауэр является программой, предназначенной для управления потоками данных в сети. Такая программа предназначена не только для серверов, но и обычных устройств. Если говорить о среде UNIX, то там обычно обсуждаются не очень простые и наглядные программы этого типа. Зачем это нужно?

Межсетевой экран, сетевой экран – программный или программно-аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него сетевого трафика в соответствии с заданными правилами (текст Википедии).

В среде UNIX часто обсуждаются сложно управляемые сетевые экраны. В случае ученика стоит поступить так: 1) поставить программу **gufw**, 2) настроить её для режима «дом». Для

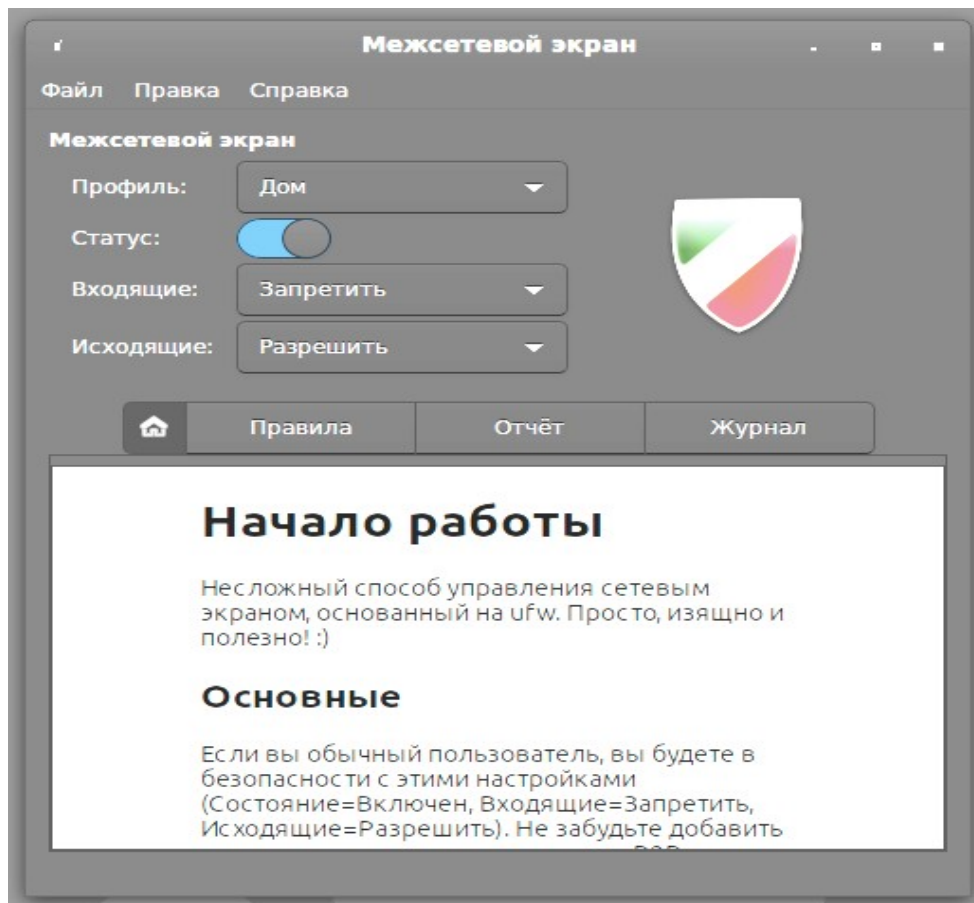
сервера этого в лучшем случае мало. Зато просто. Более конкретно:

- скомандовать

```
apt install gufw
```

(в случае ОС, родственной Debian)

- настроить экран как показано после запуска программы – требуются права администратора



Об SSH и доступе к удалённому компьютеру.

Какой нам интерес иметь доступ к удалённому компьютеру? Ответов может быть несколько. Один мотив – в получении помощи, об этом уже написано. Другие мотивы: управление другим компьютером (а может быть планшетом), в частности управление арендованной мощностью через интернет, или доступным через интернет хранилищем файлов. Также это нужно администратору если он должен быстро чинить инфосистему – находясь далеко от неё. Выше мы рассматривали подключение к рабочему столу. Это – работающий, наглядный способ. Но он становится проблемным при медленном канале. Его можно назвать дилетантским. Знающие администраторы используют подключение к командной строке UNIX. Но чтобы это было полезным, надо уметь делать там нужные операции. Классическим способом выполнить это подключение является **SSH**, то есть защищённая оболочка. Она реализована с помощью программы-сервера **SSH**, и обеспечивает шифрование передаваемых данных. Сервер позволяет работать с командным интерпретатором ОС обычным образом, но после обеспечения шифрования, и подключения к компьютеру, на котором вы хотите что-то сделать. Это может быть виртуальная машина, что не вносит изменений в вашу работу. Для копирования файлов или каталогов между вашим и удалённым компьютером используется программа **scp**.

Также я хочу замолвить слово за потенциально дешёвые хранилища в интернете, доступные через программы, работающие с протоколом SFTP. Это – то же что FTP, но с

шифрованием передаваемых данных. На UNIX есть возможность использовать программу **FAR** как клиентскую для сервера SFTP. Она позволяет достаточно просто манипулировать файлами и каталогами.

Производительность.

Производительность, или вычислительная мощность – свойство микропроцессора, и некоторых видеоплат. Оно *само по себе* не зависит ни от чего кроме разгона – который мы тут не обсуждаем. Но производительность единицы оборудования, или инфосистемы, напротив, зависит от многих факторов. Что это за факторы? Скорость работы ОС, скорость используемой для рассматриваемого процесса программы, скорость работы носителей и сетевых соединений, объём оперативной памяти. Даже файловая система, скорость поиска в базе данных, а тем более применение RAID²² влияют на скорость. Каждый пользователь должен знать, что старый компьютер может ещё послужить достаточно долго – если применить эти факторы. Обычно компьютер, выпущенный 5-10 лет назад всё ещё можно использовать для более-менее традиционных функций. Более старым достаются функции брандмауэра, и так далее – совсем малотребовательные. Обычно основным способом ускорения является переход к быстрой ОС – например, на Slitaz. Такие ОС – интересная вещь для провинциалов так как у них часто нет денег на более дорогую технику. Такой покупатель может купить подержанный компьютер с большой уценкой.

Профилактика.

Это можно уподобить заботе об автомобиле. В отдельных случаях отсутствие этих мер ведёт к существенным потерям – в разном смысле слова. Поскольку профилактика не является захватывающим занятием её лучше делать регулярно, а в программном отношении – автоматически. Задания планировщика в общем – хороший способ автоматизации, но задания пропадут при смене ОС. Если на их спасение рассчитывать нельзя то просто приклейте на дисплей липкую бумажку с датой последней профилактики.

Температура.

Переносной и стационарный компьютер может быть использован только в условиях когда испарина от разницы температур не покрыла его детали, кроме того перепады температур и перегрев могут – ВАЖНО – привести к остановке процессов (теоретически вплоть до разрушения микросхем).

Перегрев как угроза давно осознан техниками и инженерами, но действия пользователей пока не свидетельствуют о достаточном понимании проблемы. Вы можете установить программу отслеживающую температуру микропроцессора, правда вас интересует не интерполированная, а прямая оценка. Эта микросхема нагревается сильнее других. Если степень нагрева рассчитывается от количества работы то это вызывает сомнения.

А произойти это может ещё из-за повышения температуры воздуха вообще, нагрева корпуса и плохого теплоотвода и остановки вентиляторов. Поэтому тем более нельзя оставлять переносной компьютер на солнечном свете – даже весной, а уж про лето и говорить нечего. То же верно про салон машины – там может быть жарко.

Говоря о теплоотводе я имею в виду слой пыли на микросхемах. Он мешает им охлаждаться. Поскольку они являются источниками тепла они должны отдавать его.

Для улучшения вентиляции можно сменить шлейфы перейдя к современным узким моделям. Вы можете свернуть шлейф и закрепить липкой лентой или резинкой.

Вода и моющие средства.

Несмотря на то что напольное положение компьютера является самым устойчивым оно же самое опасное в плане температуры, пыли и случайных ударов ногами. Мокрая обувь или моющий пылесос уборщицы может закортить контакты которыми иногда снабжают лицевую панель. Например, там может быть разъём USB.

Пыль и грязь.

Не используйте пылесос для очистки деталей поскольку он несёт заряд статического электричества. Возьмите кисточку или резиновую грушу. В блоке питания даже в

22 Аппаратная конструкция из жёстких дисков, позволяющая увеличить скорость чтения с них.

отключённом состоянии имеется конденсатор поэтому возможен удар током. Поэтому на блоке питания наклеена яркая предостерегающая бумажка. Это не абракадабра, это предостережение об опасном напряжении на английском языке.

Запомните что нельзя лезть в блок питания где расположен вентилятор металлическими предметами, особенно пока машина работает. Второй вентилятор находится на микропроцессоре. Возможно есть ещё один-два. С ростом мощности растёт температура. Они издают шум независимо от того в каком состоянии но тембр может меняться. Продаются бесшумные модели компьютеров.

Продувать машину лучше всего на балконе в ветреный день поскольку пыль очень мелкая, летучая и часто её много. Она легко попадёт в лёгкие если её занесёт воздух.

CD\DVD дисковод можно очистить только в ремонтной конторе.

Стёкла некоторых дисплеев можно протирать только специальной жидкостью иначе вы смоете антибликовое покрытие. Не используйте растворители.

С другой стороны, вода совершенно бессильна перед старыми грязевыми полосами. Не играйте в русскую рулетку часами протирая мокрой тряпкой корпус. Используйте моющие средства, в компьютерных магазинах есть специальные, и салфетки для очистки. Салфетки хороши тем что из-за малой толщины просто не могут удержать много жидкости поэтому они только влажные но не мокрые и не дают капель. Когда ваш «железный друг» будет помыт он будет вызывать совсем другие эмоции. Делать это надо, разумеется, при отключенном питании. Если в компьютер попала вода обращайтесь к специалистам.

Протрите принтер так как многие модели сделаны там что вместе с листом в них входит и пыль с корпуса. Убедитесь что стекло сканера чисто. Сальные пятна могут отпечататься на оригинале.

Вентилятор – одна из немногих движущихся частей поэтому его может остановить пыль и волоски которые всегда присутствуют в воздухе; сигаретный дым играет ту же роль. Они могут прилипнуть к масляной поверхности вала.

Программные процессы.

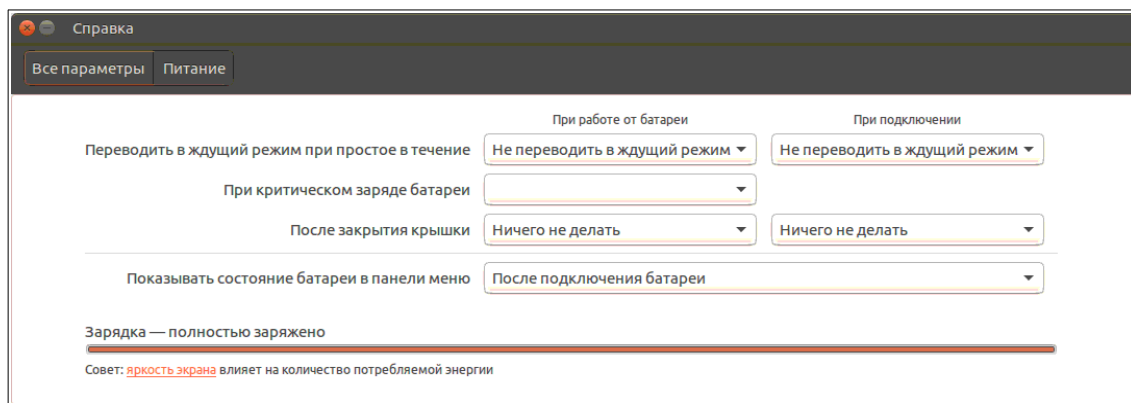
Теперь несколько слов о программной стороне. В качестве дежурных мер которые можно автоматизировать с помощью заданий планировщика назову дефрагментацию диска и его очистку (см справку ОС). Некоторые приложения созданы специально для этого. Сюда же можно отнести антивирусную проверку и конечно резервирование. Кроме того при старом винчестере нужна проверка его состояния. Возможно он поддерживает Self-Monitoring, Analysis and Reporting Technology (**SMART**). В таком случае специальная программа или BIOS может показывать результаты этой самопроверки. Также в интернете вы найдёте описание параметров которые определяются программами диагностики.

Электропитание и ACPI.

Система ACPI нам интересна в основном как средство управления питанием – не обязательно всего устройства, а, возможно, его частей. Это компьютерная система, хотя планшеты и смартфоны также часто можно настраивать в плане питания и включения или выключения. ACPI нам интересна по следующим причинам.

- Она позволяет пробудить компьютер, вывести его из состояния низкого энергопотребления, которое может быть полезно пока компьютер не используется.
- Она может управлять частями компьютера, вернее, их питанием, а именно питанием
 - монитора
 - сетевой карты
 - других устройств
- она может управлять производительностью компьютерной работы. В данном случае это не производительность в понятиях пользователя – измеряемая, скажем, количеством распознанных страниц – она зависит от программ – а техническая производительность микропроцессора. Последняя, тем не менее, влияет на то что называется быстрой работой техники.

В ОС с оконной средой, обеспеченной программами для наглядного управления аппаратурой есть возможность без лишнего образования настроить нужные свойства. Прежде всего это отключение экрана, которое нельзя путать с его блокировкой. Так, например, в Ubuntu Linux есть возможность настроить это свойство в окне.



Также может быть очень важно отрегулировать питание сетевой платы так как часто именно через неё подаётся «пробуждающий» набор данных. Очень вероятно, что это придётся делать в BIOS, где часто или всегда есть пункт **Wake on LAN** или что-то подобное.

Управление контейнерами docker.

Эта глава будет более понятна при дальнейшем образовании. Контейнер – это исполняемый экземпляр, который инкапсулирует требуемое программное обеспечение. Он состоит из образов. Его можно легко удалить и снова создать за короткий промежуток времени. По сведениям Википедии сейчас программа работает лишь на аппаратуре с архитектурой 86_64 и ARM. Каждый контейнер имеет свой программный, не требующий отдельного или дополнительного оборудования выход в интернет. Возможно записать все изменения, сделанные в контейнере, в новый образ. Образ – базовый элемент каждого контейнера. В зависимости от образа, может потребоваться некоторое время для его создания. Docker имеет пакетный менеджер Docker Compose, позволяющий описывать и запускать многоконтейнерные приложения. Docker полезен для маленьких и средних приложений, когда вам хочется получать больше чем с виртуальными машинами при имеющихся аппаратных ресурсах. Созданные контейнеры можно запускать, останавливать, проверять их статус и т.д. Для того чтобы контейнеры могли общаться между собой и с внешним миром, Docker автоматически поднимает виртуальный сетевой мост и настраивает правила маскардинга (NAT) для внешнего сетевого интерфейса. Это значит, что извне достучаться до контейнеров не получится. Однако мы можем настроить проброс портов, чтобы запрос к определенным портам внешнего сетевого интерфейса машины автоматически перенаправлялся на указанные порты контейнера. Русская документация, например, тут – <https://docker.crank.ru/docs/>.

Rclone и экономия при получении файлов из интернета.

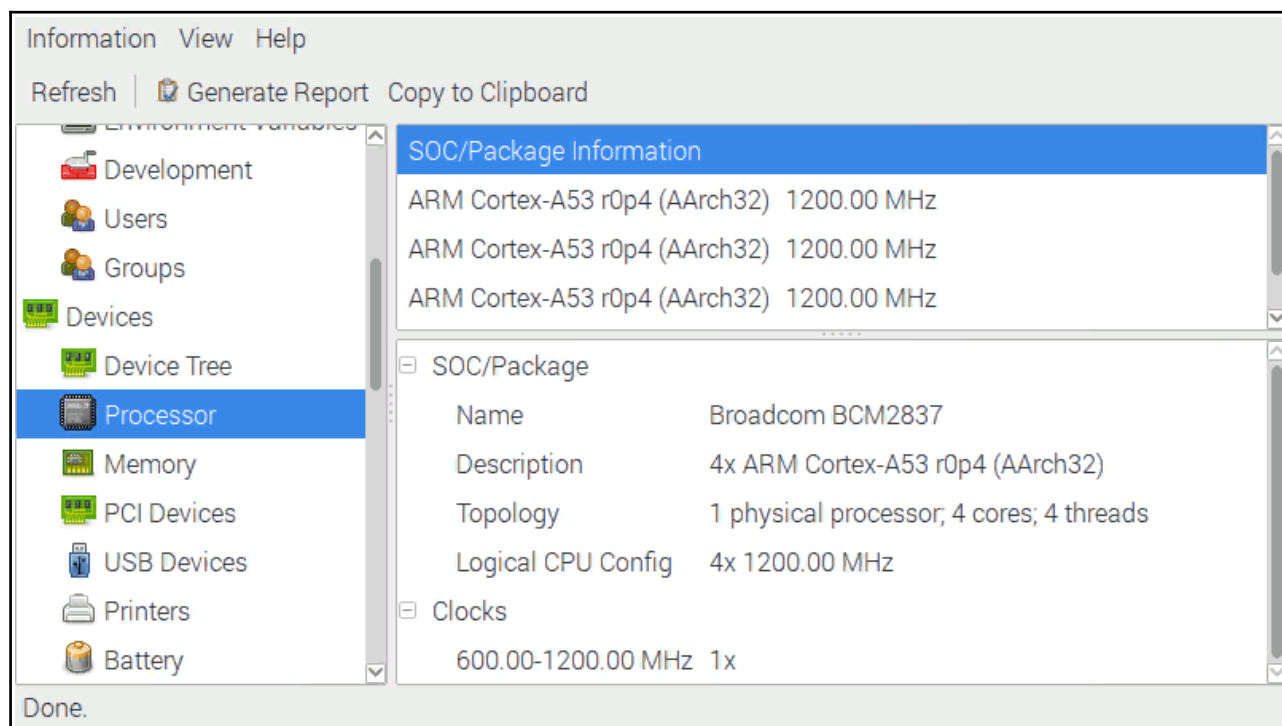
Основной интерес состоит именно в экономии при получении файлов из интернета, хотя их можно получать и из локальной сети или подключённого устройства или носителя. Вы заинтересованы минимизировать количество переданных данных так как это уменьшает стоимость, время передачи и опасность повреждения данных при обрыве связи. Конечно, это не значит что вы не можете в среде UNIX просто использовать программу **wget**. Она как раз умеет докачивать файл после обрыва связи. Также нетрудно выбрать по возможности FTP-сервер вместо HTTP при запросе файла. Не всегда у вас будет этот выбор, но он очень вероятен при запросе пакетов программ. Начинаям достаточно пользоваться **wget** или браузером. Но останавливаться на этом в провинциальных условиях не надо. Следующим шагом является освоение программы **rsync**. Её нельзя использовать на каждом шагу так как она работает с собственным программным сервером, а он есть далеко не везде. Можно было бы самостоятельно создать хранилище с этим сервером на VPS, но есть способ дешевле и проще. Все владельцы почтового ящика, предоставленного Яндексом, получают 10 Гб первоначально пустого места для произвольных файлов – важно лишь не нарушать соглашение с Яндексом. Это не очень много. Другие хранилища могут быть больше, и также ничего не стоят. Но тут вы получаете (?) русскую техподдержку, и возможность использовать программу **rclone**. Зачем нам эта программа? Это несложный способ минимизировать объём данных, получаемых при

скачивании из хранилища. При простом скачивании без помех при передаче данных экономии нет. Однако если полученный файл повреждён при передаче, то эта программа позволяет исправить его передавая лишь повреждённую часть файла, а не его весь.

Кроме того, надо всегда помнить, что распространяемые вами файлы должны быть как можно – в разумных пределах – меньше. Для этого часто используют архивацию. Также для уменьшения размера файла можно применить смену формата, и/или свойств файла.

Сбор сведений об оборудовании.

Одна из задач администратора – иметь под рукой описание техники даже – и тем более – если она не работает. Для этого есть средства в том числе в среде Linux. Это в первую очередь программа **hardinfo**. Она требует оконной для показа окна либо использоваться в консоли с перенаправлением описания техники в текстовый файл.



Кроме того, можно использовать консольную программу **dmesg**. Она выдаёт на экран менее понятные строки. Их можно записать в файл простым перенаправлением в файл.

В любом случае на предприятии, в организации надо не жалеть бумаги, а распечатать сведения о технике. При поиске драйверов или подобных вопросах это будет полезно.

Использование образов дисков.

Образ диска – это файл. Смысл использования образа в том что не нужно писать и иметь реальный диск. В наше время образы легко умещаются на наличных носителях. Образы каких дисков бывают? Прежде всего это образы лазерных дисков, их размер равен размеру данных на лазерном диске. Это обычно файлы формата ISO. Во-вторых, это файлы формата IMG, являющиеся копиями флэш-носителей – обычно брелков. В-третьих, это разные файлы, используемые для виртуализации – копии реальных файлов-дисков для таких программ как Virtualbox. Смысл таких образов в том что не требуется самому ставить и настраивать ОС. Она уже поставлена и настроена, но надо знать пароль и логин к ОС. Образы могут поставляться в архивах.

Образы лазерных дисков.

Использование образов лазерных дисков позволяет не изнашивать аппаратный читатель этих дисков. Для этого потребуется сделать образ диска. Это проще всего сделать в таких программах как **Brasero** – занятых в основном записью данных на лазерные диски. Та же операция возможна в консоли UNIX. Нельзя создать ISO-файл из диска с защитой от

копирования.

Для этого подайте команду вида

```
dd if=устройство читателя of=файл образа
```

Например, так

```
dd if=/dev/cdrom of=/home/username/iso-name.iso
```

Тут проблемой может быть обозначение читателя. Обычно это устройство по адресу `/dev/cd0`. В UNIX каждому устройству сопоставлен файл. Если вы введёте не то обозначение то, в худшем случае, скопируете не тот диск. Если непонятно что писать то используйте программу **hardinfo** или **dmesg**.

Подключение образа для чтения его содержимого выполняется так – с правами администратора.

```
mount file.iso /media/iso/ -t iso9660 -o loop
```

Так мы примонтировали образ. Теперь его содержимое видно в каталоге, обозначенном в команде. Для отмонтирования образа подайте команду

```
umount /media/iso/
```

Или используйте такие программы как **ISOmorphin** или **Gmount-iso**.

Образы флэш-памяти.

Образ обычно интересен потому что так создаётся брелок для использования его как загрузочного носителя некой ОС. Заметим, что использование такого брелка в библиотеке или интернет-кафе может привести к скандалу. Первоначально есть образ как файл, и брелок или карта памяти. Чтобы узнать каким устройством считается брелок подайте команду

```
lsblk
```

Текстовый вывод команды покажет устройства, и разделы на них. Тут надо быть осторожным так как нельзя ошибиться. Жёсткий диск, скорее всего, обозначен как **sda**, а брелок как **sdb**. Разделы на носителях обозначены цифрами, например, как **sdb1**. Запись образа стирает всё на брелке, и выполняется так.

```
dd if=/home/vlad/ubuntu.img of=/dev/sdb
```

Для подачи команды нужны права администратора. В данном случае брелок имеет обозначение **sdb**, а образ называется **ubuntu.img**.

Образы жёстких дисков для виртуализации.

Так как виртуализация выполняется конкретной программой, то она может требовать какого-то своего типа файлов. Так, Virtualbox умеет читать файлы с расширением VMDK, и умеет писать в них. Эта программа поставляется так что можно в командной строке UNIX даже конвертировать один тип образа в другой. Обычно это не нужно, и образ несёт установленную ОС, которая может использоваться в эмуляторе, в виртуальной машине.

Организация фрэймбуфера.

Эту не совсем простую тему я затрагиваю именно в интересах провинциалов, которые заинтересованы не использовать оконную среду, и этим экономить ресурсы техники, но при этом хотят получить изображения на экране.

Я не жду что можно будет выдавать любые изображения, или буквально повторить оконную среду. Но многие изображения как файлы можно конвертировать в файлы, которые можно показать.

Обратимся к статьям интернета:

В Ubuntu включить фрэймбуфер можно следующим образом:

1. Открываем файл **/etc/modprobe.d/blacklist-framebuffer**:

```
nano /etc/modprobe.d/blacklist-framebuffer.conf
```

теперь прокомментируем строку:

```
#blacklist vesafb
```

и сохраняем файл.

2. Открываем файл **/etc/initramfs-tools/modules**:

```
nano /etc/initramfs-tools/modules
```

добавляем строчки:

```
fbcon  
vesafb
```

и сохраняем файл.

3. Пересобираем initramfs image

```
update-initramfs -u
```

4. Для Grub, первой версии, открываем файл `/boot/grub/menu.lst`

```
nano /boot/grub/menu.lst
```

для Grub2 открываем файл `/boot/grub/grub.cfg`

```
nano /boot/grub/grub.cfg
```

В опции ядра добавляем "vga=789" (для разрешения 800x600) в след. строку:

```
linux /boot/vmlinuz-2.6.32-19-generic root=UUID=fcf2d8bc-8ef4-40e1-adb9-c6a0304cb2a4  
ro quiet vga=789 splash
```

Если у вас другое разрешение - смотрим тут:

http://en.wikipedia.org/wiki/VESA_BIOS_Extensions#Linux_video_mode_numbers

Перезагружаемся и наслаждаемся

(по <http://linux-bash.ru/menusistem/13-framebuffer.html>)

Глава 38. Идентификация, индексация и каталогизация.

Википедия говорит что идентификация — установление тождественности неизвестного объекта известному на основании совпадения признаков. Наверное, чаще всего имеется в виду идентификация людей. Но она может применяться и к другим объектам.

Индексация – это – в узком смысле, связанном с поиском информации – создание массива данных для поиска – прежде всего – текста. Смысл этого в том чтобы связать слова с конкретным объектом; обычно с документом. Для чего это может быть нужно? Чтобы ускорить поиск. Чтобы искать текст без обращения к набору документов. Индексация обычно нужна перед тем как искать текст в документах. Также можно индексировать, например, фотографии или видео.

Каталогизация – это процесс, который является частью работы по соотносению свойств документа и конкретного документа. Отличие от индексации в том что каталогизация имеет дело с абстрактными категориями, которые могут не упоминаться в тексте, но быть средством рубрикации текстов. То же верно про другие виды информации.

Все рассмотренные вопросы практически важны. Их решение отчасти связано с наличием баз данных, отчасти – с индивидуальными свойствами документов, в качестве какихых сегодня выступают идентификаторы в виде текста – например, номера, штрих-коды, QR-коды, и RFID-метки. Созданы системы считывания этих идентификаторов, которые иногда способны и записывать в идентификатор новые значения. Пользу от таких систем поясню на примере. Некоторые библиотеки вынуждены расставлять книги собирая их по размеру. При этом нарушается любое соответствие положения книги и её тематики. Или можно рассмотреть случай, когда поддержание порядка в документах невыгодно. В таком случае идентификатор позволяет быстро найти документ.

В практике пользователя каталогизация часто выполняется относительно фотографий. В результате может создаваться база данных. Программа **AWK** способна работать с небольшими текстовыми базами данных. В качестве идентификатора или его части можно использовать контрольную сумму файла. Программа LibreOffice Base создана для управления базами данных.

Искусственный интеллект позволяет решать задачи классификации без ручной работы.

Глава 39. Искусственный интеллект.

Искусственный интеллект уже не тема фантастических романов, но пока он лишь начал участвовать в производственной сфере, а также оказании услуг. Через 30 лет он может принять совсем другой вид. Будущее за развитым искусственным интеллектом, который – как мне кажется – не только обеспечит людей информационными услугами, но и совершенно

изменит производственные процессы. В результате в странах где он будет работать может возникнуть растущий класс людей, работа которых не нужна, которые живут на государственном пособии – достаточном для их быта. Это будет огромный шаг вперёд для развитых стран. Свобода от работы будет расти, а финансовые возможности граждан не уменьшатся, а будут возрастать благодаря новым производствам. Пока искусственный интеллект обеспечивает автоматизацию в следующих областях.

- Финансы
- Военное дело
- Тяжёлая промышленность
- Медицина
- Управление человеческими ресурсами и рекрутинг
- Музыка
- Новости, издательство и писательство
- Онлайн и телефонные службы поддержки клиентов
- Техническое обслуживание телекоммуникаций
- Развлечение и игры
- Транспорт

(по тексту Википедии)

Глава 40. Как настроить шрифты в UNIX.

Шрифты важны не только когда вы создаёте документы, но и при показе строк в терминальной программе, и в консоли, и в оконной среде. Эта глава мотивирована прежде всего тем что на телевизоре при использовании старых моделей вроде Sony Trinitron текст может отображаться плохо, и его понимание становится во многом результатом догадки. Это важно прежде всего при использовании микрокомпьютеров.

Сразу надо оговориться что автор не будет вдаваться в глубины этого вопроса, и в разных ОС настройка шрифтов выполняется по-разному.

Даже базовая часть ОС умеет выдавать строки в консоль. Если есть оконная среда, то уже есть хотя бы 10-20 шрифтов. Но неизвестно устроят ли они вас.

К шрифту предъявляются многие требования. В частности, важно чтобы в нём присутствовали нужные символы. Процент шрифтов без присутствия кириллицы очень высок. Latex предъявляет некоторые требования к шрифтам, и это ограничивает набор используемых там шрифтов.

Напомню, что использование шрифтов оговаривается лицензией, и шрифт может стоить денег.

Настройка шрифтов в консоли Debian.

Это особенно важно при работе сервера, или на VPS/VDS. Первоначально используемый шрифт может быть малопригодным. В таком случае подайте команду от имени администратора

```
dpkg-reconfigure console-setup
```

В появившемся окне псевдографического режима выберите кодировку, с помощью которой интерпретируется набор символов консоли. Скорее всего достаточно выбрать кнопку **Ок**. Для этого может потребоваться использовать табуляцию или стрелки. Когда нужная кнопка выделена цветом, нажмите её с помощью **Enter**. То же верно про строки, перечисляющие возможные кодировки.

В следующем окне идёт выбор набора символов. Опять же – скорее всего достаточно согласится на уже выбранный, выделенный вариант.

Затем выберите сам шрифт. И размер шрифта. Часть размеров не будет использована без фрэймбуфера.

Ваша консоль перестроится под новый шрифт.

Не исключено что можно настроить и цвет символов, но это не так важно.

Всё написанное должно работать и на Raspbian.

Настройка шрифтов в терминальной программе Mlterm.

Эта программа может устроить ученика без настройки. Если надо отрегулировать размер символов то подаётся команда

```
mlterm -w 20
```

То есть размер шрифта равен 20-ти кеглям. Шрифт имеет лишь некоторые возможные значения размера. Если вы захотите невозможный размер то будет использован предыдущий.

Шрифты в оконной среде.

С этой темой многим придётся столкнуться. Оконная среда или DE требует шрифтов, и поставляется с ними; она же обычно позволяет их выбирать. Такие программы как LibreOffice Writer добавляют много шрифтов. Но фактически выбор невелик. Многие шрифты либо очень похожи, либо не устроят при наборе русских текстов. Поэтому стоит поискать шрифты в интернете. В результате вы получаете либо один файл шрифта, либо архив с несколькими такими файлами. В среде Debian простейший способ сделать шрифты доступными одному пользователю – поместить их в каталог `.fonts` в личном каталоге. Такой каталог можно создать вручную. LibreOffice Writer придётся перезапустить чтобы шрифты стали ему доступны.

Некоторые рекомендации.

Многие шрифты созданы лишь для особых случаев. Для больших текстов они непригодны. Мозг привыкает читать определённые тексты. В России это всё ещё тексты, набранные шрифтом Times. Именно Times может не годится из-за юридических аспектов или оплаты. Тогда попробуйте шрифт Nimbus Roman или PT Serif. Самый легко читаемый шрифт – Verdana. Но он журнальный и довольно широкий. Иногда имеет смысл купить диск со шрифтами. Когда шрифтов много используйте программы для их просмотра и других дел. Назову несколько – на мой взгляд – хороших шрифтов.

DejaVu Serif	Nakula
Noto Serif	Vollkorn
Times New Roman	IBM Plex Serif
Trebushet MS	PT Sans Narrow
Ubuntu	Alegreya
URW Gothic	Prata
Bitter	DejaVu Sans Mono
Spectral	Oswald
Roboto	Marmelad
Lora	Georgia
Cormorat Garamond	IBM Plex Serif
Merriweather	Literata
Playfair Display	Oranienbaum

Глава 41. Программирование в интересах регионов.

Программисты продемонстрировали весьма прохладное отношение к тому факту что миллионы людей не имеют хорошего канала в интернет. Созданы монстроподобные, и при этом не всегда быстрые программы размером в десятки мегабайт, а иногда намного больше. Никто не будет заботиться о регионах лучше чем они сами. Программирование как создание программ – сложная работа, требующая сосредоточенности, организованности и ответственности. Это самая сложная работа среди всех профессий. Однако пугаться её не стоит – пугаться надо бессонных ночей. Например, простейшая программа для создания пустого окна с помощью Python версии 3 занимает всего несколько строк.

```
from tkinter import *  
window = Tk()
```

```
window.title("Python GUI App")
window.configure(width=500, height=300)
window.configure(bg='lightgray')
window.mainloop()
```

Программирование на сегодня касается массы вопросов. Я – не специалист в этой области. Но мнение у меня есть. Правильный выбор не только языка, но и соответствующих задаче инструментов обозначает всего один ответ на вопрос какие это инструменты и языки – подходящие. Такая постановка вопроса, конечно, может оттолкнуть так как это означает знать более одного языка. Если кого-то это не устраивает то можно поискать достаточно хороший язык с достаточно большими возможностями. Может быть это Julia, Java, Python, C, Kotlin. Может быть кто-то выберет Lua.

Для полноценного программирования важно избегать ошибок, вызванных неподходящими данными. Поэтому важна типизация данных. Не все языки учитывают это обстоятельство – и тем не менее существуют.

На сегодня известны «лекарства» от известных недостатков разных языков. От медленной работы Python – многозадачность, **PyPy**. От старомодности C – компиляция в него простым путём. От медленной работы Java – использование **GraalVM**. От слабой типизации – проверка данных.

Не имея существенного опыта или знания по теме программирования, обозначу некоторые перспективные для негородских поселений технологии.

- Все способы улучшить жизнь с помощью программ для Android.
- Java, вернее, JVM, предоставляющий набор ресурсов для работы потенциально небольших программ.
- Все проекты, предполагающие использование браузера как среды для программ произвольного назначения. На сегодня наиболее известный такой проект – **Electron**. Но он требует получать из интернета много данных.
- Все способы создания небольших программ. Это могут быть программы на языках Haskell, WebAssembly, и так далее.
- Совершенствование способов сжатия данных – в том числе «на лету», при работе браузера. На сегодня архиваторы PAQ, в особенности такой архиватор, встроенный в **PeaZip** уже позволяет сжать существующие текстовые файлы – например, исходники программ – на, примерно, 30% больше чем обычным путём – с использованием архиватора **xz**.

Глава 42. Как подключить читатель лазерных дисков, мышь и клавиатуру к смартфону.

Лазерные диски – прежде всего DVD – всё ещё нужны если пользователь имеет на них ценные файлы. Это могут быть диски, полученные по почте – по заказу из города.

Мышь и клавиатура нужны как замена касаниям экрана, для более точной работы – например, набивки и правки текста.

Чтение лазерного диска важно как способ читать файлы, которые могут быть получены из сервиса для отправки файлов на лазерных дисках. Тут могут быть важны 1) модель читателя, 2) качество проводов питания, 3) наличие дополнительного отводка от провода, через который надо запитать читатель, 4) особенности смартфона, и его программ. По сведениям интернета на некоторых читателях работает режим для таких целей, он включается после подключения к смартфону, но при выдвинутом лотке долгим нажатием кнопки лотка на читателе. Успех дела не гарантирован. Смотрите описание читателя в интернете. При удаче вы сможете читать с лазерного диска, но не записывать на него.

Проводной способ в том чтобы подключить через OTG-переходник разветвитель (то есть хаб), а к нему – мышь и/или клавиатуру (оба устройства должны иметь разъём USB); есть сведения что важна версия разъёма USB. Беспроводной способ состоит в использовании Bluetooth-соединения. Через Bluetooth не получится подключить читатель лазерных дисков. Смотрите сведения о Bluetooth в интернете.

Глава 43. Лазерные диски, дискеты. Каталогизация данных носителей.

Лазерные диски.

Лазерные диски во многом потеряли свою ценность в условиях городов так как там используются безлимитные каналы в интернет. Напротив, в других местах лазерный диск всё ещё представляет интерес как дешёвый носитель, который легко передать – в том числе послать почтой. Проигрыватели лазерных дисков с музыкой могут не работать с дисками DVD, но в общем такой диск является предпочтительным так как вмещает больше данных чем CD. Обычно рассматриваются наиболее популярные диски DVD ёмкостью 4,7 гигабайта. Диски типа BD (bluray) пока менее распространены. В случае хранения больших объёмов данных они интересны, но не у всех есть читатель таких дисков. Не пытайтесь записывать что-то на штампованный лазерный диск. Всегда следите за чистотой диска, храните его так чтобы избежать царапин и солнечного света. Уже записанный диск надписывайте – иначе вы очень скоро забудете где что записано. Для этого есть специальные, или перманентные фломастеры; писать ими надо на той стороне где написан тип диска. Также можно создавать надписи на другой стороне, но это зависит как от диска, так и писателя. Такая возможность интересна для организаций, когда надо обеспечить автоматическую маркировку дисков. На неперезаписываемые диски пишут данные которые не требуется изменять. В противном случае сейчас может быть лучше использовать флеш-носитель. Учитывая сложное строение флеш-брелка всё ещё лазерные диски – особенно BD – могут быть лучшим способом архивного хранения информации. Всегда желательно, а иногда необходимо проверять записанный диск. Проверка делается после записи с помощью программ, пишущих на диск. Для оконной среды это, например, **Brasero**, а для консоли - **xorriso**.

Дискеты.

Дискеты во многом ушли из употребления так как их заменили флеш-брелки и карты памяти. Всё ещё можно купить читатель-писатель дискет как отдельный узел, но автору неизвестно будет ли он работать при использовании UNIX. В старых версиях ОС эта возможность была, но тогда читатель подключался к материнской плате. Большинство дискет имеют ёмкость 1,44 мегабайта. Недостаток дискет в том что они медленно заполняются данными и читаются. Если на вашей технике есть читатель дискет то его не следует снимать и выбрасывать. Дискеты может быть трудно купить. Старые дискеты могут нести ценную информацию. Лучше всего скопировать её на более современный носитель.

Каталогизация.

Частный пользователь или организация могут иметь множество дисков любого типа. Скорее всего это лазерные диски. Чтобы не перебирать их в поисках нужной информации требуется каталогизировать диск. То есть сохранить отдельно некую информацию о данных на диске. Наиболее очевидным способом каталогизации является сохранение списка файлов и каталогов на диске вместе с указанием их размера и даты редакции. Однако на самом деле это очень скудная информация. Имя файла или каталога может быть информативным если оно что-то значит. Получаемые из интернета файлы могут иметь совершенно бессмысленные имена. Интерес представляют программы для индексации текстов, и другие подобные. Простейшая каталогизация выполняется с помощью команды

```
ls -lhR каталог монтирования диска>diskname
```

где **diskname** – информативное имя файла, который сам подлежит хранению как ценная информация. Размер файла будет небольшим. Его удобно хранить в интернете.

Глава 44. Как попробовать другую операционную систему. Литература.

Проба другой ОС.

Проба другой ОС интересна, прежде всего, для будущего использования при нелегально полученной и применяемой сейчас ОС. На некоторых компьютерах всё ещё нелегальная ОС, и

это Windows. Её можно заменить, и наиболее перспективной альтернативой является популярная система семейства Linux, либо FreeBSD. Заметим, что ОС PC-BSD по опыту автора не обладает достаточным качеством чтобы ей пользоваться. Её преимущество в виде готовой оконной среды несущественно так как примерно то же можно получить на FreeBSD путём добавления и настройки программ за один день – если не за два часа.

Вопрос о выборе ОС семейства Linux можно рассматривать исходя из свойств разных ОС. Однако при этом ясно, что ученику наиболее интересны немногие системы. А именно:

- Mint, Ubuntu, Runtu, Fedora, OpenSUSE – наиболее простые в эксплуатации
- Debian – наиболее простое получение массы программ для вывоза их в регионы, часто обсуждаемая, надёжная система
- Lubuntu, Antix, Q4OS, Slitaz – для старых компьютеров, или при большой нагрузке на современные

Эта книга пытается помочь администраторам будущих инфосистем в регионах. Именно для задач такого рода, как кажется автору, стоит попробовать прежде всего Oracle Linux, Calculate Linux, FreePBX (последняя – для телефонных услуг). Для случаев повышенной нагрузки на аппаратуру может быть оправдано использование DragonflyBSD.

NetBSD, несмотря на её нетребовательность к аппаратным характеристикам, и, как пишут, устойчивость в работе, а также достаточно хорошее руководство, всё же будет трудна для начинающих. Ещё более трудным представляется использование OpenBSD, для которой её разработчики за многие годы не потрудились написать пособие. ОС Q4OS и Slax основаны на Debian, который хорошо описан и обсуждён, но эти системы не имеют своей документации, или она очень мала. Владельцам дешёвых микрокомпьютеров всё же стоит попробовать NetBSD, которая своей быстрой работой во многом компенсирует медленное оборудование. Однако тут потребуются терпение, и лучше всего это удастся людям с опытом конструктивной работы.

Имеет смысл попробовать Android для компьютера из-за простоты работы с этой системой. Но она почти не документирована, и в плане гибкости и набора возможностей уступает традиционным системам. Android – хороший выбор для неспособных к обучению людей.

Все обозначенные системы способны работать с кириллическими текстами – возможно, после некой настройки.

Легче всего найти информацию про Debian, Ubuntu, FreeBSD. Oracle Linux снабжён большим англоязычным руководством.

Ubuntu.

Ubuntu всегда упоминается как самый распространённый, адаптированный к простому пользователю дистрибутив Linux. С документацией ситуация достаточно хороша. Без лишних трудностей работают PPA, и это дистрибутив, официально поддерживающий CUDA, то есть ускоренную обработку данных на видеокартах (не на всех).

FreeBSD

Автор считает эту систему перспективной для России. Потому что она уже снабжена русскоязычным руководством почти по всем вопросам. В случае других систем это либо сделано хуже, либо не сделано никак (пока). Общий объём руководства, готового к печати, составляет на сегодня сотни страниц – в моей вёрстке – более крупной, и с сокращениями ненужных частей это примерно 860 страниц (см pkprovincia.ucoz.ru). Чтение руководства предполагает уже существующее знание как минимум на уровне школьной подготовки. Там могут быть неизвестные термины, важно их не путать с названием программ. Привлекательность проекта в том что он постоянно содержит и переписывает готовое к печати руководство. Конечно, пользователь любой ОС должен уметь читать с экрана, но в данном случае можно читать пособие до покупки компьютера, дома. Нынешнее руководство набрано довольно мелким шрифтом, ВАЖНО – там есть лишний, бесполезный для пользователя раздел о ключах; его не надо печатать.

1.3.4. Текущая версия FreeBSD

FreeBSD - это свободно доступная, с полными исходными текстами, основанная на 4.4BSD-Lite версия для компьютерных систем, основанных на Intel i386™, i486™, Pentium®, Pentium® Pro, Celeron®, Pentium® II, Pentium® III, Pentium® 4 (или совместимыми), Xeon™, DEC Alpha™ и Sun UltraSPARC®. В основном она базируется на программном обеспечении от группы CSRG, U.C. Berkley, с некоторым дополнениями из NetBSD, OpenBSD, 386BSD и Free Software Foundation.

С момента выпуска FreeBSD версии 2.0 в конце 1994 года, производительность, возможности и стабильность FreeBSD существенно возросли. Самое большое изменение - это полное обновление системы виртуальной памяти с объединением виртуальной памяти и буферного кэша файловой системы, что не только увеличивает производительность, но и уменьшает количество используемой FreeBSD памяти, делая 5 Мбайтовую конфигурацию более приемлемым минимумом. Другие улучшения включают полную поддержку клиента и сервера NIS, поддержку транзакций TCP, поддержку «дозвола по запросу» в PPP, встроенную поддержку DHCP, улучшенную подсистему SCSI, поддержку адаптеров ISDN, ATM, FDDI, Fast и Gigabit Ethernet (1000 Мбит), улучшенную поддержку новейших контроллеров Adaptec и многие тысячи исправленных ошибок.

В дополнение к базовой системе, FreeBSD предоставляет коллекцию портированного ПО, включающую тысячи популярных программ. На момент подготовки этого документа в ней было более 24,000 портов! В кол-

Также в интернете часто обсуждается эта ОС.

Разработчики утверждают, что заняты безопасностью системы не менее, чем её надёжностью, а надёжность такова что многие сайты и другие важные инфосистемы многие месяцы работают без вмешательства людей и остановки. Заметьте, что это инфосистемы не требующие ручного управления.

Насколько мне известно, FreeBSD сильно нагружает жёсткий диск, что может означать её меньшую пригодность в случае использования флэш-памяти. С другой стороны, FreeBSD малотребовательна в микрокомпьютерах в других отношениях.

9-я версия этой ОС содержала удобный настройщик, затем его перестали поставлять, и, думаю, что теперь вы даже его не сможете поставить как отдельную программу. Но его функции не исчезли, а просто выполняются другими программами.

Авторы некоторых публикаций честно предупреждают о том что в первый раз может не получиться поставить систему. Также при домашнем, несерверном применении её придётся дополнять программами и настраивать. Но эти трудности происходят лишь при невежестве. Это вам не Fedora, тут мало уметь гонять мышь по столу.

Есть поддержка старой техники (с архитектурой i386). Много программ, так что об этом не стоит волноваться.

В руководстве написано что FreeBSD может быть использована как основа

- сервисов, доступных через интернет
- образования
- исследований
- оконной среды, рабочего места неспециалиста
- программирования

Debian.

По сравнению с FreeBSD тут поддержано больше оборудования, и больше пользователей. Если программа сделана не вообще, в виде исходного кода, и в её документации упоминаются какие-то бесплатные системы, то почти наверняка обсуждается работа на Debian. Очень много готовых к использованию пакетов скомпилированных программ, также есть их исходный код. Есть поддержка старой техники (с архитектурой i386), хотя, конечно, Debian не так скор, как, например, Slitaz. Несмотря на свою, можно сказать, суровость, или отсутствие оформления для детей, это очень популярная система. Часто обсуждаемая. Debian знаменит надёжностью своей основы, и несвежими выпусками программ. Но они не подводят – проверены. Свежесть программ не так важна как надёжность. Тем не менее при любой ОС лучше пользоваться свежими программами для этой системы. Много программ для любых целей.

Руководство по Debian развивается как отдельный проект, и на сегодня в большой степени переведено на русский. В моей вёрстке оно содержит 417 страниц.

Предисловие

Debian — очень успешная операционная система, участвующая в нашей цифровой жизни в гораздо большей степени, чем многие часто предполагают или знают. На момент написания Debian — самый популярный вариант GNU/Linux на серверах: согласно [W3Techs](#) более 10% всемирной паутины работает на Debian. Задумайтесь, сколько веб-сайтов Вы бы недосчитались сегодня без Debian? Среди наиболее впечатляющих применений — использование Debian на Международной космической станции. Следили ли Вы за работой астронавтов на МКС, скажем через присутствие в социальной сети NASA или других международных организаций? И сама работа, и сообщения о ней стали возможны благодаря Debian. Бесчисленные компании, университеты и государственные учреждения полагаются на Debian в своей повседневной работе, доставляя услуги миллионам пользователей по всему миру... и даже на орбиту!

OpenSUSE.

Эта ОС привлекательна в нескольких отношениях:

- поддерживается коммерческой фирмой, хотя техподдержки вы, скорее всего, не получите
- снабжена большим руководством на английском, готовым к печати
- не требует тратить большого объёма данных при обновлении программ
- администрирование сделано более наглядным при наличии оконной среды
- есть поддержка старого оборудования, и экзотических архитектур

Вид руководства. Текст набран слишком просторно, это – одно из руководств, OpenSUSE Reference, размером в 538 страниц.

13 Basic Networking

Linux offers the necessary networking tools and features for integration into all types of network structures. Network access using a network card can be configured with YaST. Manual configuration is also possible. In this chapter only the fundamental mechanisms and the relevant network configuration files are covered.

Linux and other Unix operating systems use the TCP/IP protocol. It is not a single network protocol, but a family of network protocols that offer various services. The protocols listed in *Several Protocols in the TCP/IP Protocol Family*, are provided for exchanging data between two machines via TCP/IP. Networks combined by TCP/IP, comprising a worldwide network, are also called “the Internet.”

Руководство по безопасности имеет 458 страниц.

Fedora.

Fedora, может быть, не очень популярна, но наглядна. Она обладает простым и удобным установщиком. Но вопросы установки не так важны как вопросы эксплуатации ОС. Fedora спонсируется коммерческой фирмой. Поэтому она будет поставляться ещё неопределённо долго. Обновления программ требуют меньше данных. Flatpak поддерживается изначально.

Oracle Linux.

Эта система есть модифицированная ОС Red Hat Linux. В основном представляет интерес из-за документации. Вид руководства для версии 7.

Preface

Oracle® Linux 7: Administrator's Guide provides introductory information about administering various features of Oracle Linux 7 systems, including system configuration, networking, network services, storage devices, file systems, authentication, and security.



Note

The information in this book has been migrated to separate and more updated documentation. For the most current information, refer the individual, topic-based guides in the [Oracle Linux 7 Documentation](#).

Этот документ имеет 84 страницы, но он может быть неполным. Пример документации Red Hat Linux.

- Remotely with the server's IP address: **<https://192.0.2.2:9090>**
If you use a self-signed certificate, the browser issues a warning. Check the certificate and accept the security exception to proceed with the login.

The console loads a certificate from the `/etc/cockpit/ws-certs.d` directory and uses the last file with a `.cert` extension in alphabetical order. To avoid having to grant security exceptions, install a certificate signed by a certificate authority (CA).

NixOS.

Эта система интересна не большой популярностью, а, в основном, уменьшением страха поломки системы, который может испортить жизнь ученику. Вот автоперевод рекламы с сайта NixOS.

«Воспроизводимый»

Nix строит пакеты изолированно друг от друга. Это гарантирует, что они воспроизводимы и не имеют необъявленных зависимостей, поэтому, если пакет работает на одной машине, он также будет работать на другой.

Декларативно

Nix делает тривиальным совместное использование сред разработки и сборки для ваших проектов, независимо от того, какие языки программирования и инструменты вы используете. (это, похоже, для программистов)

Надежный

Nix гарантирует, что установка или обновление одного пакета не приведет к нарушению других пакетов. Он позволяет откатиться к предыдущим версиям и гарантирует, что ни один пакет не находится в несогласованном состоянии во время обновления.»

Насколько я помню, также старые параметры не пропадают, и их как минимум можно прочесть. Для этой системы есть 60000 пакетов. Это много. Непосредственно на сайте описано использование контейнеров. Есть руководство по системе на английском. Насколько мне известно, система не родственна таким монстрам как Debian. Это недостаток. Так как нельзя заглянуть в чужую документацию. Но нервы тоже дороги. К тому же можно спросить других пользователей, но общение может быть на английском – как и в случае того же Debian.

Установка на сегодня не наглядна, но есть готовая виртуальная машина с оконной средой для Virtualbox размером 1,5 гб.

NixOS считается «лёгкой» системой – пригодной для слабой техники.

Android.

Android интересен и как ОС для компьютеров типа PC, и как ОС для планшетов и телефонов. Эту ОС не надо долго изучать – по степени упрощения управления она далеко обгоняет «традиционные» ОС – в том числе Windows. Тем не менее естественно, что многие нуждаются в чтении руководства по Android. Оно не поставляется вместе с устройствами, но доступно в интернете. Для версии 5 оно называется Android Lollipop Quick Start Guide, и содержит 70 страниц.

В этом руководстве содержатся сведения о системе Android 5.0 для устройств семейства Nexus и серии Google Play Edition. Инструкции по установке SIM-карты, технические характеристики и другую информацию об этих устройствах можно найти на странице support.google.com/android. Кроме того, на ней содержатся подробные сведения обо всех телефонах и планшетах Android, а также о других версиях этой операционной системы.

Android на PC-компьютере может быть не совсем адекватен, то есть иметь некие ограничения. Но сама идея практична – дать пользователю простую среду деятельности.

ULBSD

Эта система есть ни что иное как FreeBSD с оконной средой, что позволяет ученику не делать дополнительных действий для создания наглядного рабочего места. Последний выпуск платный (сейчас 800 рублей за сбережённые нервы), прежние – бесплатны. Однако ВАЖНО помнить что установочная программа уничтожает всё на целевом носителе, которым обычно является жёсткий диск. Создатели системы утверждают что почти не вносят изменений в базовую ОС. Система представляет интерес прежде всего для домашних пользователей так как они вынуждены самостоятельно добавлять оконную среду поверх FreeBSD.

Haiku.

Эта ОС не имеет отношения ни к Linux, ни к BSD. Она интересна прежде всего потому что быстро работает на старых компьютерах, обеспечивая при этом оконную среду, и примерно ту же простоту какая есть в Windows. Скорее всего Haiku удастся попробовать с помощью Virtualbox. Готового диска для этого может не быть в интернете.

Hyperbola.

Дистрибутив Linux, который, как мне кажется, хорош для старой техники, делается добросовестно.

Литература и сайты.

Кроме уже указанных пособий, Вам особенно интересно прочесть или просмотреть следующее.

Источник	Язык	Объём
Linux Foundations	русский	Более 173 стр
Introduction to Linux – A Hands On Guide	русский	Более 216 стр
Операционная система Linux. Курс лекций. Учебное пособие.	русский	348 стр
Кто был ничем тот станет всем Кузнецов И.А.	русский	Более 852 стр (в версии для малых экранов)
UNIX для студента	русский	?
Введение в бесплатные программные системы	русский	Около 40
Компьютер и информатизация на селе	русский	Около 40
Королёва, В. А. Инновационные технологии современного офиса (Облачные вычисления)	русский	88
Linux Network Administrators Guide Olaf Kirch, Terry Dawson	английский	505
Fundamentals of the UNIX System Instructor	английский	1147

Guide		
BSD UNIX TOOLBOX	английский	339
Linux Networking Cookbook Carla Schroder	английский	640
Network Administration with FreeBSD 7 Babak Farrokhi	английский	278
Effective awk Programming, 4th Edition: Universal Text Processing and Pattern Matching Arnold Robbins (бесплатный учебник)	английский	602
Таненбаум Э.С. Компьютерные сети	русский	960
Д. Куроуз, Т. Росс Компьютерные сети. Настольная книга системного администратора	русский	912
Collins. Большой толковый словарь компьютерных терминов Синклер Айен	русский	?
Ваулина Е. Мой компьютер. Толковый словарь	русский	
Информатика. Новый систематизированный толковый словарь-справочник Воройский Ф.С.	русский	760
Большой англо-русский толковый словарь по вычислительной технике и информационным технологиям Э.Пройдаков, Л. Теплицкий	русский	1600
Современный англо-русский словарь компьютерных и телекоммуникационных терминов Ников Виктор Алексеевич	русский	256
Основы ОС Linux/UNIX для пользователей Садов О.Л.	русский	106
Лукас Майкл. FreeBSD. Подробное руководство. (2009)	русский	864
Rexx Programmers Reference – Howard Fosdick (бесплатный учебник)	английский	723

Некоторые книги устарели, но полезны. В разных изданиях может быть разное количество страниц.

Также стоит посмотреть следующие сайты.

- losst.ru
- pkprovincia.ucoz.ru
- rus-linux.net
- askubuntu.org
- tldp.org
- distrowatch.com
- linux.softpedia.com
- www.ibm.com
- www.linuxquestions.org
- www.gnu.org
- freebsd.org
- oracle.com
- lifelacker.ru/topics/technology/android/
- www.android-x86.org
- www.opennet.ru
- runetbsd.ru
- <https://www.intuit.ru/studies/courses/673/529/info>

Также вам может быть полезна книга про программу **git**. Знать всё об этой программе не

надо. Но стоит иметь в виду что она позволяет получать многие программы. Эти же программы часто можно получить и иначе. Книга называется «Pro Git», переведена на русский, готова к печати, и имеет 282 страницы.

Как попробовать другую систему.

Нужно понимать что проба может выполняться с ограничением возможностей. Рассмотрим разные способы сделать это.

- Попробовать систему, доступную через интернет, но без её установки вами, не на VPS\VDS. Для многих проблемой будет низкая скорость интернет-канала. Сама система может не иметь выхода в интернет.
- Попробовать вашу систему на VPS\VDS. Платная услуга, и опять могут быть проблемы из-за низкой скорости. Но не требуется получать через интернет много программ, и саму систему.
- Установить Linux на смартфон с помощью **Drive Droid**, **Linux Deploy**, **Debian no root** или других подобных программ. Это программы для Android. Требуется достаточно места либо в памяти смартфона, либо на его сменной карте памяти.
- Использовать live-диск. Файлы компьютера будут доступны только для чтения, и требуется много оперативной памяти, система будет работать медленно, установка программ будет в лучшем случае работать в течение одного сеанса работы диска. Системы не могут работать одновременно.
- Купить или создать карту памяти или флэш-брелок с системой. Часто небольшой объём постоянной памяти, не всегда достаточное качество брелка или карты (что грозит потерей данных). Системы не могут работать одновременно.
- Пробовать систему в виртуальной машине. Повышенные требования к объёму оперативной памяти, часто необходимость поддержки виртуализации аппаратурой.
- Использовать контейнер с системой. Надо уметь управлять контейнерами. Потребуется заставить работать оконную среду в них.
- Поставить систему на жёсткий диск как вторую. Требуется разобраться со стартом систем. Нужен достаточно большой диск. Надо освободить место на нём. Системы не могут работать одновременно.
- Поставить систему на отдельный жёсткий диск. Требуется настройка BIOS. Системы не могут работать одновременно.
- С помощью Microsoft Windows Services for UNIX. Прочтите статью на https://ru.bmstu.wiki/Microsoft_Windows_Services_for_UNIX.
- С помощью **andLinux**. Так можно использовать почти все программы для Linux. Но прежде прочтите, например, статью на <https://ru.bmstu.wiki/AndLinux>.

На старой технике могут быть препятствия для использования виртуальной машины. На очень старых компьютерах могут быть ограничения на объём жёсткого диска. Там неплохо выглядит полная замена системы, которая подробно описана. Либо использование диска **Super Grub2 Disk**, который позволяет не заниматься вопросами старта систем.

Всегда прежде убедитесь, что операционная система, или подготовленный лазерный диск (или что там) для её установки подходят вам, и вашей аппаратуре. Почитайте про ОС.

Достаточно часто для пробы системы пользователи скачивают из интернета образ установочного диска этой системы. Это файл. Так как он может быть искажён при его передаче требуется его проверка. Такая проверка выполняется путём вычисления контрольной суммы для этого файла, и сравнением её с правильной контрольной суммой. Но это тогда когда правильная сумма вам известна. Сейчас в отдельных случаях правильная контрольная сумма вам не известна. Тогда можно скачать файл через качественный канал – где повреждение файла маловероятно. Причём можно качать не к вам на компьютер, а в любое хранилище в интернете – лишь бы там можно было вычислить контрольную сумму. Ещё проще написать письмо создателям ОС. Иногда при установке ОС можно проверить правильность установочного диска – это функция программ на этом диске.

Образ установочного диска не является контейнером.

Для пробы системы есть немного более быстрый и более простой путь – использовать уже созданный виртуальный диск или виртуальную машину. Иногда эмулятор, в котором работает виртуальная машина, сам называют виртуальной машиной. По сути это верно, но в документации виртуальной машиной называют не его, а именно конкретную среду работы

системы, и таких сред – управляемых эмулятором – может быть много. Виртуальный диск, как и виртуальная машина может быть представлен как один файл. В случае виртуального диска вы получаете файл, который в созданной кем-то виртуальной машине изображал жёсткий диск. Вопрос о необходимом объёме оперативной памяти для системы на этом диске открыт. В случае виртуальной машины вы получаете все файлы для использования системы, но эта виртуальная машина может не работать вообще никак или быстро если ей не хватит оперативной памяти. То же с виртуальными дисками. Можно ли использовать виртуальный диск или виртуальную машину для установки их содержимого, то есть системы, на ваш жёсткий диск? Я не могу это обещать. Поэтому есть вероятность что эти полезные и большие файлы будут использованы лишь в эмуляторе.

Эмулятор есть программа, создающая некую среду, «песочницу» для другой системы, но эта другая система не может работать без работы основной системы. В качестве эмулятора хорош Virtualbox, он описан по-русски в интернете. Также иногда стоит попробовать QEMU, но с этой программой будет больше вопросов. QEMU работает медленнее, но требует меньше скачивать для своей установки. Эмулятор хорош тем что вы не рискуете что-либо испортить в основной системе. Заметьте, что эмулятор позволяет поставить в качестве пробуемой ОС Windows, но она в общем от этого не становится бесплатной.

Вы можете обратиться к своим знакомым, или связаться с кем-то через интернет. Вам даже обычно не обязательно чтобы этот человек тратил время и использовал лазерные диски. Любой носитель годится если он достаточно ёмок и его можно прочесть. Как минимум вы получите набор пакетов. Вам потребуется образ первого, установочного диска. Это может быть образ небольшого размера. Важно лишь что он позволит поставить ОС. Всегда будьте внимательны к тому что конкретно получаете. Если рассматривать Debian, то его просто передавать как набор файлов-образов дисков. Диски нумерованы. Чем меньше номер тем нужнее диск. Вы можете взять не все вообще доступные диски. У некоторых людей может сформироваться набор пакетов или образов полученных от разных людей или из интернет. Это нормально, и все их можно использовать при совместимости. Источник сам по себе не важен. Так, например, ваш сосед по району может без трудностей «залить» на вашу карту памяти довольно большое количество программ. Если для этого используется носитель, на который может быть поставлена – или уже стоит – некая ОС, то копировать пакеты или образы может быть лучше в отдельный раздел носителя. Это позволит не возиться с лазерными дисками, и сменять ОС не стирая полученные пакеты и образы. Если вы доверяете вашему помощнику можно просто переслать носитель по обычной почте. Ищите дешёвые и ёмкие карты памяти.

Как и в случае любых больших файлов представляет интерес рассылка лазерных дисков с ними. Наверное, самый известный сервис для этого – <http://www.filespost.ru/>. Эта услуга – платная, но она обходится дешевле чем получение этих же файлов из интернет. Не пытайтесь скачивать большие файлы в библиотеке – вас могут наказать, или оборвать передачу файла. То же верно про интернет-кафе. Лучше всего спросить разрешения на такой запрос. Так как получение больших файлов может быть долгим, но требует лишь несколько процентов вычислительной мощности, вы можете уступить место за компьютером другим пользователям. Они смогут использовать почти всю мощность компьютера. Ещё более продуктивный путь – найти человека, у которого есть нужный вам файл. Это может быть компьютерный администратор той же библиотеки. Также не скачивайте большие файлы на работе.

Что бы вы не скачивали – лучше это делать не браузером, а такой программой как **wget** или **rsync**, но последняя интересна тем, что умеет работать с rsync-сервером. Такой сервер есть программа, обеспечивающая минимизацию потока данных при исправлении неверных файлов которые уже скачаны. Заметим, что хранилище Яндекса (это фирма) также позволяет (?) вам исправить неверный файл, но это несколько сложнее чем просто скачать его. Это хранилище хорошо подходит в качестве зеркала репозитория, например, для Debian. В данном случае зеркало – это просто дополнительный репозиторий. Если вы в Москве, то оно может обеспечить очень высокую скорость подачи данных – возможно, большую, чем ваш канал в интернет. Видимо, для него, также как для торрентной скачки, есть льготы по скорости.

Полученный образ установочного лазерного диска хорош тем что с его помощью можно поставить ОС как в эмулятор, так и на аппаратуру. Классический путь установки ОС – подробно и много раз описанный – по крайней мере для популярных ОС – это установка с

лазерного диска. В наше время это скорее DVD диск чем CD. Вопрос о том что лучше вам – дискуссионный. На практике для сервера скорее нужен CD. Но так как полученная с его данными и программами рабочая среда для многих будет менее понятна, то может быть всё же надо брать образ DVD. Для домашнего применения лучше DVD, но при условии что у вас есть устройство под него. Обычно есть. Также можно получить образ BD-диска, но устройств под него часто нет. Образ для создания ОС на карте памяти или брелке предлагается редко, и интересен когда вас устроит именно использование этих носителей вместо жёсткого диска.

Создание диска из образа не обязательно делать на том компьютере где будет опробована или поставлена ОС. Тут важно выполнить именно *запись образа*, а не его файла обычным путём. Сам процесс записи прост, и занимает десятки минут. Для минимизации вероятности ошибок при записи надо установить минимальную скорость записи.

После этого вам не следует бездумно ставить ОС с лазерного диска. ВАЖНО – прежде надо скопировать всё ценные данные и программы с того раздела или устройства, куда будет поставлена новая ОС. Это достаточно простой, но продолжительный процесс. Не спешите, и внимательно посмотрите что есть ценного у вас на компьютере. Помните что нужные файлы могут быть не только у вас, но и у других пользователей компьютера. Не настраивайте и не ставьте ОС на оборудование на работе или в интернет-кафе, или в библиотеке. Для этого есть администратор.

Установка ОС хорошо описана в интернете. Она обычно включает следующие этапы

- Старт установщика новой ОС.
- Создание пустого раздела для новой ОС.
- Определение имени и пароля пользователя, или нескольких пользователей. Определение административного пароля.
- Определение языка рабочей среды, и ОС. В России это русский язык. Тем не менее достаточно много программ написаны для англоговорящего пользователя. Эти программы будут работать и на ОС с настройкой на русское оформление рабочей среды.
- Определение часового пояса для ОС.
- Определение набора программ, устанавливаемых кроме базовой системы.
- Установка программ с лазерного диска на раздел на жёстком диске.
- Определение других ОС на том же жёстком диске, и установка селектора ОС – обычно это GRUB.
- Изъятие лазерного диска.
- Перезагрузка компьютера, и старт новой, или другой выбранной в селекторе ОС.

Но пользователю важно не просто поставить ОС, а создать полезную и понятную среду. Это обстоятельство объясняет то, что ему потребуется ставить некие программы, и настраивать ОС.

Установка с CD-диска, как и ограниченная установка вообще, может быть далека от выполнения этого требования. Я считаю интересной для мало подготовленных людей ОС FreeBSD. Но она не создаёт желаемой рабочей среды после установки. Так же как ограниченная установка Debian, и многих других систем. В тяжёлом случае после установки новая система начинает работу, но вы видите лишь запрос логина, и несколько достаточно бесполезных строк. Из такой ситуации можно при быстром канале интернета выйти за недолгое время, но при достаточном знании. То, что вы видите – это не повод выбрасывать компьютер в окно. На микрокомпьютере или старой обычной технике вы получаете наиболее быструю среду с возможностью заставить работать оконную среду, и дополнительные программы.

Вам потребуется знание того как ставить программы, и настраивать ОС. Вы даже можете использовать любую литературу по UNIX, но понимая что вы можете расстроить систему. Это не фатально пока у вас есть возможность поставить её заново, и вы не теряете при этом ценные данные. Также очень важно иметь выход в интернет. Это более вероятный путь к нужным статьям и книгам чем местная библиотека.

Возможно, вы решите использовать готовую виртуальную машину чтобы избежать трудностей установки системы и программ. В целом это довольно обманчивый путь, хотя совершенно реальный. Так как вы всё равно будете иметь трудности, и вопросы.

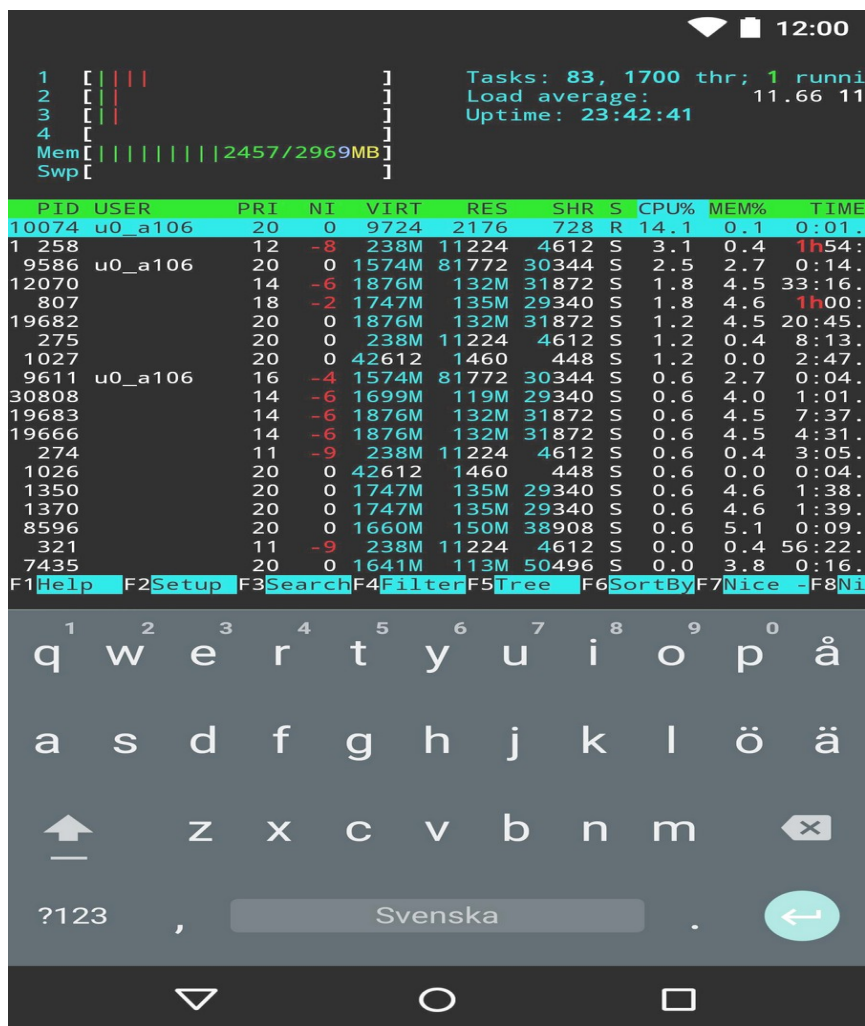
Чтение файлов другой операционной системы.

Другая ОС обычно на разделе с другим типом файловой системы. С этим могут быть трудности. Для чтения файловых систем **ext2/3/4** в среде Windows есть драйвер **Ext2Fsd**. Мне пока не удалось найти способ чтения файловой системы WinFS в среде UNIX. Поэтому может быть нужно создать раздел для передачи файлов между разными ОС. Он должен быть доступен для записи и чтения. Для Linux лучше использовать наиболее свежую файловую систему из вообще подходящих.

Установка Linux на смартфон или планшет.

Такая установка может быть простой, и не сносит уже работающую ОС. Она не вынуждает вас тратить деньги на дополнительную технику и не мотивирует заниматься выходом в интернет в новой ОС. Автор использует программы **Termux** и **AndroNix**. Они должны быть вместе на устройстве. Аппаратные и другие требования мне не известны и могут зависеть от версии этих программ. Это, наверное, самый простой способ получить Linux на носимом устройстве. Во второй программе вы выбираете конкретную ОС, копируете в буфер обмена Android соответствующую строку команды, и вставляете её в терминал первой программы. После запуска команды – при наличии выхода в интернет – начинается получение оттуда файлов, и установка ОС. Так как это довольно большой объём, то нужно хотя бы на час иметь быстрый канал (зайдите в интернет-кафе с Wi-Fi в ближайшем городе – всё равно ездите туда за шашлыками).

Результат в случае ОС без оконной среды может быть таков.



```
12:00
Tasks: 83, 1700 thr; 1 runni
Load average: 11.66 11
Uptime: 23:42:41
Mem [|||||]2457/2969MB
Swp [ ]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME
10074 u0_a106 20 0 9724 2176 728 R 14.1 0.1 0:01.
1 258 12 -8 238M 11224 4612 S 3.1 0.4 1h54:
9586 u0_a106 20 0 1574M 81772 30344 S 2.5 2.7 0:14.
12070 14 -6 1876M 132M 31872 S 1.8 4.5 33:16.
807 18 -2 1747M 135M 29340 S 1.8 4.6 1h00:
19682 20 0 1876M 132M 31872 S 1.2 4.5 20:45.
275 20 0 238M 11224 4612 S 1.2 0.4 8:13.
1027 20 0 42612 1460 448 S 1.2 0.0 2:47.
9611 u0_a106 16 -4 1574M 81772 30344 S 0.6 2.7 0:04.
30808 14 -6 1699M 119M 29340 S 0.6 4.0 1:01.
19683 14 -6 1876M 132M 31872 S 0.6 4.5 7:37.
19666 14 -6 1876M 132M 31872 S 0.6 4.5 4:31.
274 11 -9 238M 11224 4612 S 0.6 0.4 3:05.
1026 20 0 42612 1460 448 S 0.6 0.0 0:04.
1350 20 0 1747M 135M 29340 S 0.6 4.6 1:38.
1370 20 0 1747M 135M 29340 S 0.6 4.6 1:39.
8596 20 0 1660M 150M 38908 S 0.6 5.1 0:09.
321 11 -9 238M 11224 4612 S 0.0 0.4 56:22.
7435 20 0 1641M 113M 50496 S 0.0 3.8 0:16.
F1 Help F2 Setup F3 Search F4 Filter F5 Tree F6 SortBy F7 Nice F8 Ni

1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p å
a s d f g h j k l ö ä
↑ z x c v b n m ↵
?123 , Svenska . ↵
```

В данном случае Android настроен на язык, отличный от русского. Программу для набора символов можно менять. Автор использует для этого 1С Клавиатура. При наличии оконной

среды крайне желательно чтобы работала мышь. Она создаст курсор. В противном случае управление пальцами иначе как издевательством назвать нельзя. Также хорошо заставить работать аппаратную клавиатуру. Смотрите интернет.

Проба ОС на VPS\VDS.

Этот способ хорош тем что вам не нужно ничего скачивать и ставить у себя на технике. Но услуги аренды вычислительной мощности в целом платны, и вопрос в том как вы будете платить. Вам могут дать пробный бесплатный период. С помощью браузера вы обращаетесь на сервис, там уже должна быть ваша регистрация. Обычно там не надо ставить ОС, но можно ей пользоваться. Скорее всего сервис сам обеспечит установку базовой, консольной части свободнораспространяемой ОС, Windows обычно там не ставят. Вероятно, вы сможете через браузер работать в консоли, и, если есть достаточно аппаратных возможностей, поставить программы для оконной среды, и использовать её. Но оконная среда требует быстрой передачи данных чтобы отображаться на вашем экране.

Live-диск или live-флеш брелок.

Такие диски стоит поискать как приложения к книгам и журналам, но тамошние ОС должны быть не слишком стары. Также брелок иногда можно заказать и купить через интернет. Оба способа позволяют поработать с ОС, но не равны её работе после классической установки. Основное отличие в том что если явно не сохранять что-то на носитель, то эти данные потеряются при окончании сеанса работы ОС. Также не сохраняются параметры работы программ, и их данные, например, закладки браузера. Содержимое жёсткого диска не меняется, ваша ОС там не меняется и не заменяется.

Работа этих носителей происходит тогда, когда они находят BIOS раньше чем жёсткий диск. Многие компьютеры умеют использовать флэш-брелок как загрузочное устройство, а загрузка с лазерного диска – вообще классика. Live-диск часто можно использовать для полноценной установки ОС.

Подробнее о Virtualbox.

Эта книга старается облегчить работу ученика. Virtualbox может не работать на слабой технике, вернее, там не будет достаточно оперативной памяти для второй ОС, и он сам требует не менее 512 мб оперативной памяти; реально для каждой ОС обычно нужно не менее 1-2 гб оперативной памяти. Сам Virtualbox работает в среде Windows и Linux, а также FreeBSD и Solaris на аппаратуре с архитектурой x86 или AMD64, которую иногда обозначают как 86_64.

Установка Virtualbox может оказаться более сложной чем хотелось бы. После моих экспериментов я предпочитаю ставить версию для RHEL, которая годится для всех, или многих систем семейства Linux. Также Virtualbox можно поставить с помощью пакетного менеджера если он есть как программа для него.

Таким образом, если у ученика есть достаточное терпение он получает несложный способ попробовать другую ОС.

Запуск Virtualbox не требует прав администратора. После этого ученик получает простое окно, где одна из кнопок стартует одну – выбранную – ОС. Та работает в своём окне. Само это окно управляется так же как другие окна используемой ОС. Либо можно работать в полноэкранном режиме, но потребуются знать как выйти из него. У ученика есть несколько способов выключить это окно вместе с его ОС – в том числе сохраняя состояние ОС.

Практически важно правильно настроить работу мыши. Это важно потому что на некоторых клавиатурах нет правой кнопки **Control** (Ctrl), которая переключает мышь в среду основной ОС.

Для увеличения возможностей можно подключить к Virtualbox пакет расширений, а к работающей в эмуляторе ОС – дополнения к гостевой системе. Это можно сделать даже когда некая ОС уже поставлена в эмулятор.

Источником программ для установки ОС в эмулятор может быть в первую очередь образ установочного диска для этой ОС. Если это так, то потребуется поставить ОС на виртуальное оборудование, обеспеченное эмулятором. Установка ничем существенным не отличается от установки на физический компьютер. Установленную ОС можно обновлять штатным для неё образом, ставить на неё программы и так далее.

Virtualbox обеспечивает выход в интернет для гостевой – только что обсуждаемой – ОС.

Обычно это происходит с помощью NAT. Это пригодный способ для большинства случаев, но он не позволяет обращаться из интернета к гостевой ОС.

Virtualbox умеет работать с файлами OVA, которые содержат готовые виртуальные машины – с возможностью их настройки.

Другие эмуляторы.

- Gnome Boxes

Это эмулятор, который, насколько мне известно, может не работать на Windows, но хорош тем что прост, и этим же плох. Это оконная программа, которая, скорее всего, может быть поставлена с помощью пакетного менеджера. Для начинающих это почти идеальный вариант. Эмулятор даёт выход в интернет для гостевой ОС. Информации об эмуляторе мало. Возможно, он умеет заставить работать ОС для экзотических архитектур аппаратуры.

- QEMU

Я плохо знаю QEMU, и имею неудачный опыт в её использовании. QEMU формально не считается эмулятором, но позволяет заставить работать разные ОС – в том числе для редких архитектур аппаратуры. Это в общем программа для редких случаев, и большого терпения. С её помощью можно, например, попробовать NetBSD или Raspbian. Поставленная ОС может работать в своём окне. QEMU работает на Windows, Linux, FreeBSD.

- WSL

Это вообще не эмулятор, а специальный слой программ для запуска программ для Linux в среде Windows. Проект пока молодой, и может вызвать неудачи и трудности. Требуется свежий выпуск Windows.

- VMware Workstation Player

Если вы не смогли поставить Virtualbox, то стоит попробовать этот эмулятор. Иногда его установка более сложна чем хотелось бы, но это может зависеть от наличия программ на компьютере. Когда плеер уже установлен, то он очень прост в использовании. Программа работает как на Windows, так и на Linux.

Глава 45. Вопросы качества и производительности работы.

Я надеюсь, что эта книга будет прочтена в том числе руководителями организаций, которые могут оказаться самыми невежественными людьми на своей работе, поэтому работа – это не лишняя тема. На работу надо приходиться работать. У автора нет опыта создания инфосистем для платной работы людей. Работа за компьютерным устройством, или работа этого устройства оценивается в общем так же как любая другая – в первую очередь по результату. Но и процесс работы надо знать, и продумать.

Наибольшей ценностью является работоспособность и рентабельность, но они находятся в зависимости от хранения данных и состояния программной среды. Данные должны дублироваться, храниться на пригодных носителях, и, возможно, шифроваться. Администратор как компьютерный специалист должен иметь максимум информации по всем вопросам хранения, передачи, и обработки данных в своей инфосистеме. Обычный администратор должен понимать роль инфосистемы в организации. Оба должны быть готовы к сбоям инфосистемы.

Человеческий фактор.

В России это не одна проблема, а комплекс препятствий на пути нормальной работы. Эти препятствия ставят под угрозу любые планы и бюджеты. Пора заняться психологией работника, но это не моё дело. Тем не менее можно заметить, что любая смена технологии трудна – особенно при непонимании выполняемых действий. Работнику надо, как минимум, создать условия для работы, и это значит в том числе не перегружать его. Возможность бесконтрольной редакции баз данных или подобных основ работы угрожает организованным искажением работы – вплоть до её блокирования. Причины этого могут оказаться не очень велики. В частности, вроде бы полезные, но на практике трудные в исполнении, или даже разрушительные указания и правила могут быть причиной. Смена персонала в условиях весьма разной квалификации и организованности может оказаться проблемой. Для России

типично непонимание роли правил, и руководители могут не поддерживать и создавать порядок, а разрушать его противоречивыми указаниями.

Люди являются самой ненадёжной и медленной частью инфосистемы. Они нуждаются в документации, или курсах. Курсы можно записать на видео, и сделать доступными на рабочем месте. Также всегда полезна возможность помощи при показе экрана, но без разглашения секретов и служебной информации.

Всегда учитывайте юридический аспект – не только закон, но и лицензии программ, и соглашения, в которых вы участвуете.

Технический аспект.

Это – большая тема, и директор должен в ней разбираться. Всякая смена оборудования должна рассматриваться как альтернатива смены технологии, программ, каналов связи – а не сама по себе. Возможно, имеет смысл создать дублирующие рабочие места или устройства на случай сбоя – если надо добиться непрерывной работы. Их можно арендовать.

Для увеличения скорости обработки данных можно использовать ускорители (GPU) видеоплат, и RAID-зеркалирование. На сегодня самая быстрая передача данных обеспечена, по-видимому, оптическими каналами связи, и интерфейсом SATA Express, работающим внутри компьютеров. Вне компьютеров будут самыми быстрыми интерфейсы Thunderbolt3 и USB4. Скоростные беспроводные сети сами по себе недороги, но важно чтобы ничто не снижало скорость. Есть потребность в дальнедействующих передатчиках для беспроводной сети, но сведения об этом некоторыми рассматриваются как анекдот. На самом деле направленная передача по Wi-Fi может быть осуществлена более чем на 20 км, и техника для этого легко доступна. Провода могут мешать, и их перенос труден, но они обеспечивают бо'льшую секретность. Оценка износа носителей должна быть автоматической и регулярной. Шумная техника вроде матричного принтера может мешать, отдел для телефонной связи должен быть отдельным помещением. Проблемы с электричеством, перепадами температуры и протечками воды должны быть предусмотрены.

Программный аспект.

Программный аспект очень важен. Программы, используемые 10 или 20 лет назад часто делали примерно то же что сейчас, но не требовали нынешней аппаратуры. Существенное ускорение может быть получено применением, например, более быстрой ОС, или изменением ядра Linux, а также – если надо – сменой типа виртуализации. Одна программа может делать работу в десятки раз быстрее другой. Большие или сложные файлы могут только открываться так долго, что за это время можно выпить чаю – это не следствие поломки или вируса²³. Также все должны знать о возможности простейшего программирования – сводимого часто к написанию программы длиной в одну или несколько строк. Всегда надо пользоваться свежими программами для шифрования данных. Правильная работа программ никем не гарантирована.

Важна не только скорость технических процессов, но и скорость работы людей. Для этого их рабочая среда должна быть понятной. Можно переслать документ на принтер в другом конце страны, и он там будет напечатан. Не нужны курьеры. Сетевые средства доступа вроде Samba могут заметно ускорить работу.

Использование консольных программ может в наше время ускорить работу аппаратуры во много раз. Создать такую среду можно с помощью Rexx или любого подходящего языка.

Программа Recovery Star, и подобные, позволяют многократно поднять вероятность восстановления повреждённых данных. Однако эта программа не работает в среде UNIX, и она «шифрует» файлы. Поэтому обычная каталогизация сталкивается с трудностями.

Вернуть состояние обрабатываемых данных, и программ, а также всей рабочей среды реально, но может потребоваться вернуть состояние программ не меняя данные. Сам этот вопрос должен рассматриваться заранее, при планировании инфосистемы.

Контроль рабочего процесса и журналирование.

Возможность узнать как именно шла работа, и кем выполнена операция – очень важны – в том числе при сбоях и неправильной работе. Но чтобы полностью использовать возможность

23 Тут может помочь создание составных документов

контроля надо обеспечить знание того кто работал. Это можно сделать путём видеонаблюдения, а также программной записи событий. Всегда надо обеспечить правильную запись времени, и ход аппаратных часов. Все совещания должны храниться как видеозаписи.

Глава 46. Что-то не работает или работает не так?

*Журавли летят
И солнце скрылось в тростнике
(Из японской поэзии)*

К сожалению, это возможно. Например, автор застал такую ситуацию в банке. А мог бы – в пожарной части. Требуется не просто иметь под рукой администратора, но и мудрого человека в его лице. Это – не преувеличение. Инфосистемы не просты. И прежде чем действовать надо понимать и думать.

То же верно про частную технику – с той оговоркой что она обычно не имеет дело с обязательствами, деньгами, и рисками коммерческих и других организаций. В наиболее опасном положении – работающие дома пользователи, владельцы изношенных носителей, и слабо защищённых ОС. Защищённость зависит от использования разных сетей или способов передачи данных, а также работы брандмауэра и сетевого оборудования. В случае когда надо минимизировать риск лучше пользоваться проводной сетью, и не использовать Bluetooth так как последний очень слабо защищён от взлома. Защита напрямую связана с обновлением программ – в том числе в роутерах, антивирусных баз, работой антивирусной программы, и проверкой жёсткого диска.

Полезно понимать что по сути проблемы обычно либо аппаратные, либо программные. Одна из этих причин может сказываться на другой.

Надо всегда помнить что пользователь может неправильно понять ситуацию. Это реальный источник проблем. Так, например, неожиданный результат работы программы может быть следствием её дефекта, но, скорее всего, условия работы программы, либо представления неверны, а результат верен.

В среде UNIX обычно не используются программы и драйверы Windows, и часто аппаратура обеспечивается совершенно другими драйверами – не похожими на таковые для Windows. Из-за этого не только работа программ, но и оборудования может быть отличной. Это не результат поломки или вирусной активности.

В отдельных случаях аппаратура вообще не может работать в среде UNIX, или конкретной ОС. Это наиболее вероятно для экзотической техники, или дорогой, хотя, например, микропроцессоры видеокарт, позволяющие получить кратное ускорение работ, поддержаны достаточно хорошо.

В UNIX работа Wi-Fi долгие годы была достаточно плохо обеспечена. Ситуация в общем будет улучшаться, хотя по мере выпуска новой техники не гарантировано своевременное создание нужных программ. Лучше использовать не встроенный в технику модуль беспроводной сети, а роутер или точку доступа; есть достаточно дешёвые модели, они являются средствами защиты вашей техники. В среде других ОС ситуация может быть лучше или хуже, но надо понимать, что для Windows есть больше вирусных программ, и часто нужно обращаться в интернет за драйверами. Для Linux это не характерно. UNIX прекрасно умеет работать с обычным Ethernet-проводом, через который подключается роутер и точка доступа. Также он может использовать смартфон – подключённый через USB – для связи через Wi-Fi сеть.

Наверное, наиболее популярной мерой восстановления работы программ является перезагрузка компьютера, но надо помнить что это может не только повредить данные, но и оборвать соединения, и нарушить параметры работы программ.

Также может помочь переустановка программ.

В среде Windows очень легко «откатить» состояние программной среды назад, но для этого нужно *прежде* создать точку восстановления – раньше чем возникла проблема.

Но все эти меры выглядят очень далёкими от сущностного разбирательства в ситуации. Такое разбирательство может включать в себя просмотр логов, проверку работы сети, просмотр файлов настроек и другие меры. Старайтесь обеспечить себя выходом в интернет для поиска решений проблем, и собирайте как можно больше сведений об условиях в которых

возникла проблема.

Точно знать работу программы может лишь её создатель, программист. Тем не менее логика и опыт позволяют другим людям достаточно оптимистично смотреть в будущее. Не спешите вызывать «мастера» – тем более что мастером он является не всегда.

Переустановка ОС является сравнительно несложной, но достаточно долгой процедурой для решения проблем с работой любых программ – если сами программы в порядке. ВАЖНО – прежде чем это делать надо скопировать всё ценное на посторонний носитель. Лучше всего если таким носителем будет карта памяти. Во многих случаях переустановка избавит вас от проблемы. Но она потребует заново настраивать программы, и ставить их, и обновления ОС. При наличии городского канала в интернет эта работа займёт, скорее всего, не более одного дня. В условиях регионального доступа надо перенести хотя бы часть файлов пакетов из прежней установки. Быстрее и проще поставить новую ОС в виртуальную машину. Но это не всегда правильно и возможно.

Не спешите, и позаботьтесь о том чтобы «новая» ОС могла загружаться. Если по любой причине этого не происходит, то используйте диск Super GRUB 2. Благодаря этому вы сможете сразу работать с новой ОС не тратя время на её переустановку.

Аппаратура обычно работает нормально, поэтому о ней можно много не писать. Для неё есть ремонтные конторы, или просто магазины. Так, например, сетевая плата стоит очень мало. Её не нужно чинить. Вам может потребоваться программа для диагностики аппаратуры. Её можно поставить и запустить используя live-диск.

Одна из наиболее трудных и огорчительных ситуаций – потеря данных. При частном использовании техники можно попытаться восстановить данные с помощью созданных для этого программ. В остальных случаях важно понимать что вы отвечаете за последствия своих действий. Поэтому – если вы администратор – может быть правильно сразу нести жёсткий диск, или другой носитель, или всё устройство в сервис для восстановления данных. Жёсткий диск может быть повреждён после ударов и падений. Чтобы не попадать в такую ситуацию каждая организация должна иметь систему резервирования, или RAID-массив носителей.

В организациях важно провести анализ причин чтобы устранить опасность взлома или расстройства инфосистемы в будущем. Это оправдано так как тут риски гораздо больше чем у частного пользователя, а в коммерции потеря данных или правильной работы – реальная угроза репутации и финансам организации.

Что может выглядеть проблемным?

- Любая система или сервис не пускает вас, пишет что данные идентификации неправильны. Скорее всего вы допустили опечатку, или ввели строки не на том языке или не в том регистре.
- Случайные нажатия, или слишком быстрые или недостаточные нажатия кнопок или клавиш.
- Зависания программ из-за большого объёма работы или слабой техники – в данном случае не патологичные.
- Агрессивная реклама или неожиданные интерактивные реакции сайта.
- Трудности передачи данных. Скорость может резко и значительно меняться в условиях загородных способов выхода в интернет. Такие препятствия как мокрая листва могут повлиять.
- Читатель лазерных дисков или принтер долго готовятся к работе. Скорее всего это нормально.
- Компьютер начал шуметь при отсутствии управления пользователем. Может работать антивирус. Также есть вероятность что с помощью просматриваемого сайта на вашей технике производится некая вычислительная работа не в ваших интересах.
- Сообщения о том что страница сайта не найдена. Возможно она перемещена.
- Неожиданное отключение техники. Возможно, кончился заряд аккумулятора.
- Медленный запуск программ. Это во многом определяется особенностями программы.
- Сообщение о том что у вас недостаточно прав или подобное. Это может быть от того что уже идёт некий процесс, запущенный от имени администратора. Настройкой прав занимается прежде всего администратор.
- Проблема нарушения программной среды, управляемой пакетным менеджером. Проблема решается.

- Трудности при установке пакетов. Проблема решаема.
- Отказ пакетного менеджера использовать сомнительный репозиторий. Это форма защиты вашей программной среды. Проблема решаема.
- Отсутствие man-страницы. Может быть нормальным явлением. Иногда требует установки дополнительного пакета.
- Отсутствие нужной программы в виде готового пакета. Если такой пакет есть в бэкпортах следующего выпуска ОС, то его можно оттуда ставить. В среде Linux каждая, или почти каждая программа имеется в виде исходного кода – не обязательно из репозитория. Вы можете компилировать из него программу сами, но это может потребовать установки компилятора и других программ, и занять много времени²⁴. Неудача компиляции может означать неполноценность программы, и вы можете даже обратиться к её автору или мантейнеру.

Все подобные ситуации я даже не пытаюсь перечислить. С ростом опыта ученик меньше пугается. Полезно понимать что если проблема лишь у вас, то, вероятно, она крайне редка, или так или иначе иллюзорна. И наоборот, если проблема была у кого-то и раньше, то, скорее всего, она уже описана, и часто решена. Где есть потребность в программе, там, скорее всего, есть и программное удовлетворение потребности.

Важно отличать поломки аппаратуры от отсутствия настройки программ. Это верно, в том числе, про роутеры и точки доступа, которые могут работать неприемлемо без настройки. Их встроенные программы нужно обновлять так быстро как они появляются в обновлённом виде. Эти программы не входят в репозитории.

Глава 47. Будущее инфосистем.

Вычислительная техника и программы так быстро меняются что это – вопрос практики, а не фантастики. Напоминаю, что автор не специалист в области аппаратуры или администрирования. Попробую обозначить некоторые тенденции.

- Несомненно, что в России будет использоваться собственная техника. На ней на сегодня работает модифицированный Debian, и Alt Linux.
- Интеллектуальные программы будут применяться всё больше. Это может быть не всегда и не для всех возможностью меньше знать. Компьютеры будут всё более походить на людей.
- Будут появляться специализированные микросхемы для интеллектуальных задач.
- Вопросы обращения с криптовалютами станут важны для всех.
- Недорогой телефон будет заменять сканер.
- В быт войдут мощные и дешёвые микрокомпьютеры.
- Скорость и надёжность каналов интернета будет возрастать, и быстрые каналы придут в провинциальные районы.
- ОС семейства Linux будут иметь общий набор пакетов.
- BD-диски станут обычным явлением.
- Заключение контрактов и сделок будет выполняться в смартфоне.
- Биржевые роботы станут обычным явлением.
- Создание собственного фильма перестанет быть редким явлением. Анимационные фильмы неотличимые от обычных станут делаться без больших трат.

Alt Linux.

Это российская ОС, снабжённая своим репозиторием. Вот её некоторые особенности.

- Возможность автоматизированной установки ОС.
- Выпуски как для серверного компьютера, так и для пользовательского.
- На сегодня есть выпуск с лёгкой средой XFCE. Есть и LXDE, Cinnamon.
- Самостоятельно сформированный набор пакетов и репозиторий.
- Адаптация к российским микропроцессорам.
- Подправленный менеджер пакетов APT.
- Уже готовы несколько вариантов ядер ОС.

²⁴ По поводу долгой компиляции, особенно характерной для некоторых ОС, есть известная шутка: «Бог тоже мир семь дней компилировал».

- Возможность сборки своих дистрибутивов.
- Поддержка системы виртуализации OpenVZ без дополнительных действий.
- Alterator – штатное средство настройки системы. На основе альтератора, прежде всего, построен инсталлятор системы. Может показаться, что в уже установленной системе альтератор уже не нужен, однако это не так. Альтератор предоставляет возможность выполнить заново те шаги, которые были пропущены либо сделаны неправильно при установке. Многим системным администраторам интересна удаленная настройка системы по сети. Именно для этого и предназначен интерфейс, называемый **alterator-fbi**.

Глава 48. Блокировка рекламы.

Сайт pkprovincia.ucoz.ru был создан несколько лет назад для информирования по вопросам использования сетей, компьютерных устройств, и так далее. Этот сайт на данный момент многократно упомянут, и переместить его будет трудно так как сменится адрес. На нынешнем адресе он бесплатен для меня как администратора сайта, и расположен на домене сервиса ucoz. Сервис менялся со временем, становясь всё более проблемным для провинциальных пользователей. Руководство сервиса вовсе не думает о них. Масса рекламы в виде изображений, а также реклама в виде видео, выдаваемого поверх полезной информации примерно 2 раза за одно посещение сайта – это неприемлемо не только потому что раздражает, но и потому что видео щедро расходует ваш часто лимитированный месячный объём данных. Что делать?

- использовать браузер с фильтрацией рекламы
- использовать расширение браузера для фильтрации рекламы
- отключить в браузере запрос на изображения, воспроизведение анимации, и звука
- использовать расширение браузера для отключения запроса на то же, и видео

Рассмотрим простой способ убрать рекламу на множестве сайтов. Толку от рекламы обычно нет, а поток данных может быть ограничен тарифом. К тому же вы делите канал с другими при использовании выхода в интернет через мобильный телефон или смартфон. Так я, находясь под Москвой, делю общий поток данных со множеством людей, а обслуживает нас одна вышка или антенна. Нет смысла забивать канал рекламой. Мы поставим известный фильтр для рекламы Adblock Plus. Но – ВАЖНО – он может исказить работу сайтов. Для конкретного сайта его можно отключить.

Перейдите на сайт adblockplus.org.

Покажем кусок главной страницы.

Пользуйтесь интернетом без раздражающей рекламы


- ✓ Получите более чистый и быстрый доступ в Интернет и блокируйте раздражающую рекламу
- ✓ Допустимая реклама по умолчанию разрешена для поддержки сайтов ([Подробнее](#)) ^[1]
- ✓ Adblock Plus является бесплатным ПО с открытым кодом (GPLv3+)

Нажав на кнопку ниже, вы соглашаетесь с нашими [Условиями использования](#).



[Загрузить Adblock Plus для другого браузера](#)

Нажмите на кнопку загрузки. Текст кнопки может быть другим, я ставлю в браузер Firefox. Вот что вы получите.



Adblock Plus

от [Adblock Plus](#)

⚠ Mozilla не осуществляет активного контроля за безопасностью этого дополнения. Убедитесь, что вы доверяете ему, перед установкой.


[Подробнее](#)

Этот бесплатный блокировщик рекламы для Firefox блокирует раздражающую рекламу на любых сайтах,

[Добавить в Firefox](#)

Нажмите на кнопку **Добавить в Firefox**. Далее идёт установка расширения как уже описано

выше. В результате:



Честная система

Команда Adblock Plus приветствует вас. Спасибо за установку Adblock Plus!

Adblock Plus – программа, распространяемая по честной цене. **Это означает, что мы верим во взаимную добросовестность:**

- Вы заплатите за **Adblock Plus ровно столько**, сколько можете себе позволить. Мы позволяем **вам самостоятельно** выбрать цену.

Вы не обязаны платить. Рядом со строкой адреса появится значок фильтра. При щелчке на нём можно отключить фильтрацию для текущего сайта.

(сделать картинку для этого не получилось)

Глава 49. Общие соображения.

Эта книга далеко не достаточна для получения нормальной квалификации по вопросам информатики, или компьютерной практики, и, тем более, теории. Учитывая ожидаемую аудиторию, я считаю важным подчеркнуть, что очень большая ошибка – думать, что то, чего я не описал, или не описал другой автор – не существует. Мир быстро меняется, появляются новые факты, средства, подходы, идеи. Разговоры вроде «А я и не знал что такое есть», или «А я не знал что так можно» – в данном случае скорее показывают неверный подход, чем верный. Вернее другое мнение – стремящееся к тому что обозначил Сократ, говоря «Я знаю лишь то, что ничего не знаю». Мой читатель пока многого не знает. Конечно, это уводит ученика с твёрдой почвы кажущихся или реальных фактов в болото сомнений. Что делать? «Выбросить компьютер в окно» – скажут одни. Но там может быть дворник, или собака. «Застрелиться» – решат другие. «Рвануть в Израиль» – подумают некоторые. В самом деле – что делать? Больше читать, и думать.

Глава 50. Словарь основных компьютерных и смежных терминов.

это не специализированный словарь, в него не входят многие технические и профессиональные термины, а также названия программ, обозначения протоколов связи, типов файлов (расширений имён файлов) и прочего, имена людей. смотрите словари терминов на Яндекс Маркет. составитель -- не профессионал в компьютерной сфере.

3D -- трёхмерный, с глубиной

3D-принтер -- принтер для создания объёмных фигур

3G -- третье поколение устройств связи, с достаточно высокой скоростью если не пытаться смотреть видео в формате HD. Реальная скорость может быть низкой.

4G -- четвёртое поколение устройств связи, со скоростями, достигающими скорости проводных сетей. реальная скорость может быть низкой.

аватар -- обозначение человека картинкой

автоподатчик -- устройство для подачи бумаги автоматически

авторские права -- Авторское право — в объективном смысле — институт гражданского права, регулирующий правоотношения, связанные с созданием и использованием (изданием, исполнением, показом и т. д.) произведений науки, литературы или искусства, то есть объективных результатов творческой деятельности людей в этих областях. Программы для

ЭВМ и базы данных также охраняются авторским правом. Они приравнены к литературным произведениям и сборникам, соответственно. Английский термин копирайт (англ. copyright, от «копировать» и «право») не идентичен по смыслу выражению «авторское право», которое часто[уточнить] означает только права авторов, но не смежные права. (текст Википедии)

- агрегатор новостей -- программа для получения лент новостей (RSS) – наборов статей
- адаптер -- контроллер, дополнительное устройство для соединения устройств
- администратор -- (компьютерный) -- человек, отвечающий за корректную работу программно-аппаратной системы
- адрес -- текстовая строка из цифр, точек, букв, позволяющая обращаться к сервису или устройству
- аккаунт -- учётная запись, совокупность личных данных о человеке (возможно, вымышленном) и некоторых технических данных. типичная учётная запись содержит реальное имя человека и его пароль
- аналоговый -- аппарат -- не цифровой, запись -- с плавным, непрерывным изменением уровня силы сигнала
- анонимайзер -- то что прячет или заменяет личность
- апгрейд -- повышение характеристик и\или увеличение набора деталей техники
- аппаратные часы -- встроенные часы аппаратуры. они хранят не только время, но и дату, и сменяют их даже при отсутствии питания от сети питания
- аппаратный -- технический, не программный
- апплет -- текстовая программа на языке программирования Java, работающая без её компиляции
- архив -- файл, содержащий в себе файлы и\или каталоги. обычно сумма их объёмов больше чем объём архива. в таком случае архив сделан со сжатием данных.
- архитектура -- тип аппаратуры
- аутентификация -- процедура проверки подлинности
- аутоплей -- автоматический запуск программы или открытие файла при вставлении лазерного диска
- баг -- ошибка программы, дисфункция программы
- база данных -- организованная совокупность данных
- базовая система -- программная основа операционной системы
- баннер -- анимированная или неподвижная картинка, обычно небольшая, на сайте
- белый айпи -- публичный IP-адрес
- бенчмарк -- замер скорости
- беспроводной -- передаваемый через радиосвязь, основанный на радиосвязи (не путать с обычным радио)
- блог -- дневник, набор статей
- бот -- робот
- браузер -- программа в основном для показа сайтов, печати их страниц, поиска текста на них
- буфер обмена -- ёмкость, через которую передаются данные путём их копирования или вырезания, и последующей вставки
- бэкап -- резервная копия или процесс её создания
- бэкэнд -- невидимая, но тесно связанная с обрабатываемыми данными часть программы или программа, отделяемая от основной программы условно или реально
- веб камера -- камера с невысокими характеристиками, для показа лица собеседника
- Веб-интерфейс -- наглядный способ регулировки чего-либо через браузер
- веб-мастер -- специалист, занятый созданием сайтов
- версия -- выпуск программы
- взлом -- нелегальное проникновение в информационную систему или расшифровка пароля к ней
- виброзвонок -- вибрирование без звука звонка
- видеоплата -- адаптер дисплея, монитора
- винт -- винчестер, HDD
- винчестер -- HDD
- виртуализация -- часто процесс перехода от использования реального оборудования к виртуальному

виртуальная консоль -- в UNIX и его потомках одна из нескольких текстовых рабочих сред
виртуальная сеть -- сеть с соединениями, не равная физической сети
виртуальный -- мнимый
вирус -- вредная программа
витая пара -- тип кабеля компьютерных систем, типичный кабель выделенной линии связи и локальной сети, на концах имеет обычно прозрачные разъёмы, кабель толщиной около 7 мм
ВОЛС -- волоконно-оптическая линия связи
временная зона -- параметр операционной системы, определяющий время через положение компьютера
вставить -- записать 1 файл или 2 данные в документ
вход в систему -- процедура, необходимая для начала работы в программной среде, часто операционной системе
вход звука -- гнездо, принимающее электросигнал звука
выделение -- 1 процедура обозначения части документа или 2 видимое отличие, показывающее что инструмент воспринимает ваши сигналы
выделенная линия -- личная скоростная линия связи
выйти -- 1 выключить программу или 2 выйти из программной системы или операционной системы, потерять связь с ней и возможность работать с ней
выключение -- 1 программы или 2 аппаратуры. завершение работы программы, не обрыв её работы (обрыв может разрушить данные, а выключение не может)
вырезать -- изъять файл или часть документа
выход звука -- разъём для выдачи электросигнала звука
гаджет -- небольшое устройство
гарнитура -- приспособление для освобождения рук при разговоре
гашение экрана -- выключение изображения, не дисплея
гигабайт -- единица ёмкости памяти и размера данных и программ, 1024 Мб
главная страница -- страница сайта, с которой рекомендуется начинать чтение
ГЛОНАСС -- система определения положения (российская)
глюк -- дисфункция программы
госуслуги -- услуги государственных организаций, оказываемые через интернет
графический интерфейс -- обычный сейчас способ работы, основанный на окнах программ
графический редактор -- программа для изменения изображений
группа -- группа пользователей
гуглить -- искать с помощью поисковой машины Google
данные -- не программа, пассивные записи
деинсталляция -- грамотное удаление программы
дельта -- изменённая часть данных
дерево каталогов -- набор каталогов и подкаталогов, преим. в UNIX
дефрагментация -- процедура соединения данных в кластерах носителя
диалог -- дочернее окно
динамический айпи -- потенциально меняющийся IP-адрес
дискета -- носитель ёмкостью 1,44 Мб
дисковод -- преим. устройство чтения и записи на дискеты
дисплей -- монитор, устройство зрительного вывода информации
дистрибутив -- поставка операционной системы, набор дисков или один диск для этого
документ -- компьютерная "статья", аналог бумажного документа
домен -- часть адреса сайта. не адрес файла, часть адреса компьютера.
драйвер -- программа для работы устройства
ЕМНИП -- "если память не изменяет мне"
ждущий режим -- режим работы компьютера с пониженным энергопотреблением, прекращаемый при вашей работе с компьютером.
жёсткий диск -- HDD
ЖК -- жидкокристаллический
зависание -- патология работы программы при которой она не реагирует

загрузка -- 1 вход в рабочее, полезное состояние операционной системы или 2 копирование или перенос файла на одно устройство с другого

загрузочный носитель -- носитель, с которого берутся программы при загрузке операционной системы

загрузчик -- программа для загрузки операционной системы

закладка -- пометка сайта, хранимая в базе данных браузера

закрывать -- перестать использовать файл, разорвать его связь с редактирующей его программой

запуск -- старт работы программы

захват видео -- копирование видеоматериала в компьютер

звуковая плата -- плата для работы со звуком

зеркало -- обычно копия сайта или данных, доступных по FTP

ИБП -- источник бесперебойного питания. устройство, необходимое при обрывах и потере питания.

ИМХО -- "по моему скромному мнению" (в России -- "имею мнение, хрен оспоришь")

имя -- совокупность символов для обозначения файла, каталога, и пр.

инициализация -- присвоение начальных значений параметров

интернет -- всемирная сеть компьютеров и других устройств

интернет канал телевидения -- то же что интернет-радио, но с передачей видео

интернет-банк -- сайт для выполнения банковских услуг

интернет-радио -- сервис передачи звука через интернет. не требует регистрации и разрешения, требует адрес и постоянно работающий компьютер

интерфейс -- способ и средства взаимодействия программ, деталей, компьютеров, людей

инфомат -- автомат для выполнения информационных услуг

информатизация -- процесс внедрения современных информационных технологий.

информационные технологии -- технологии обработки информации с помощью компьютера и компьютерных сетей в первую очередь.

инфракрасный порт -- приёмник или передатчик сигнала с помощью инфракрасного излучения

исходный код -- текст программы

картридер -- читатель и писатель с\на карты памяти

картридж -- обычно ёмкость с чернилами или порошком для печати

каталог -- ёмкость для файлов и подкаталогов. каталог сам не содержит полезной информации

килобайт -- единица размера файлов, 1024 байта

киоск -- примитивный программный комплекс для невежественных или нетребовательных пользователей

клавиатура -- пульт компьютера или другого устройства, в основном для набора текста или команд, а также для управления программами

клиент -- то, что получает, принимает обслуживание

ключ -- часто ключ шифрования -- строка, позволяющая расшифровку

ключевые слова -- слова для поиска или отражающие суть текста

кнопка -- часто экранный элемент изображения, работающий при использовании курсора мыши на нём путём нажатия левой кнопки мыши

кодек -- программное средство работы с видео или звуком

кодировка -- способ хранения текста

команда -- строка, регулирующая работу программы, запускающая программу

командный интерпретатор -- программа, стоящая между операционной системой и пользователем, принимающая команды от пользователя

комбинация клавиш -- набор одновременно нажимаемых кнопок клавиатуры

комментарий -- замечание по поводу уже прочтённого текста

компиляция -- перевод программ в вид, пригодный к использованию. Для работы программы компиляция может быть не нужна, и иногда даже невозможна.

комплектующие -- детали и устройства

компьютер -- ЭВМ, вычислитель, электронный аппарат для обработки данных или информации с помощью программ, хранения данных, передачи данных.

конвейер -- набор команд в UNIX и его потомках, передающих данные по цепочке, средство уменьшения ручной работы

консоль -- текстовая среда в UNIX и его потомках, недоступная при работе графического интерфейса

контент -- содержимое

контрольная сумма -- строка, рассчитываемая по данным

конференция -- общедоступная переписка

конфигурация -- набор параметров работы или деталей

копипастить -- копировать без добавления своего текста уже написанные тексты

копировать -- создавать копию -- обычно файла

корзина -- ёмкость для ненужных, мусорных файлов и каталогов

криптовалюта -- вид электронных денег, обслуживаемый путём шифрования

куки -- небольшие порции служебных данных сайтов, помогающие при работе браузера

кулер -- охладитель

курить мануалы -- читать руководства

курсор -- 1 мыши -- обычно стрелка или нечто вроде буквы I 2 текстовый -- обычно в виде вертикальной черты

кэш -- хранилище передаваемых по сети данных

лазерный диск -- пластиковый диск обычно диаметром 12 см, для передачи данных или программ, операционных систем

ламер -- слабоподготовленный пользователь

лиса -- браузер Firefox

лицензия -- юридический документ -- часто об использовании программы или операционной системы

личный кабинет -- воображаемая личная ячейка с информацией и средствами выполнения операций

логин -- имя пользователя

логический диск -- часть энергонезависимой памяти, могущая нести независимый набор каталогов и файлов

локализация -- адаптация к особенностям страны

локаль -- набор параметров программ или операционной системы, связанных с положением компьютера -- язык и пр.

локальная сеть -- местная сеть, управляется конкретным человеком.

макрос -- текстовая программа, работающая лишь в определённой программе, преим. для выполнения отдельного набора повторяемых операций

ман -- текстовая информация о программе в UNIX и его потомках

маршрутизатор -- роутер

маска -- набор символов для обозначения имён

маска сети -- строка, характеризующая адрес на программном уровне

мгновенные сообщения -- текстовые сообщения, передаваемые очень быстро (не электронная почта)

мгц -- мегагерц, единица измерения частоты и скорости работы вычислительной техники и её частей. 1000 мгц -- гигагерц

мегабайт -- единица размера файла или ёмкости памяти, 1024 Мб = 1 гигабайт

меню -- набор строк для управления программой в графическом интерфейсе

мессенджер -- программа для обмена сообщениями

метка -- строка, обозначающая носитель или логический диск

микропроцессор -- основная микросхема вычислительных устройств, "мозг" их

многотомные -- архивы на нескольких носителях или в нескольких файлах

модем -- одно из устройств связи в сети

модератор -- человек, редактирующий послания в форуме

монитор -- дисплей

моноблок -- компьютер без разделения системного блока и дисплея

монтирование носителей -- подключение их к дереву каталогов в UNIX и его потомках

мост -- средство соединения сетей
 мощность -- 1 характеристика техники, показывающая какое количество вычислительной работы выполняется за секунду 2 количество функций программы
 мультимедиа -- видео, звук, средства работы с ними
 мультитач -- готовый к множественному одновременному прикосновению
 МФУ -- многофункциональное устройство
 мышь -- устройство для перемещения курсоров и прочих действий при перемещении его по столу
 настройка -- определение параметров работы программы или аппаратуры
 нетбук -- ноутбук для работ с интернетом, без большой мощности
 неттоп -- небольшой и не мощный настольный компьютер
 НЖМД -- HDD
 ник -- имя в форуме. часто не совпадает с реальным именем человека
 носитель -- физический объект, способный нести информацию
 ноутбук -- переносной компьютер
 облако -- хранилище в интернете
 обновление -- 1 процесс обновления операционной системы или программы 2 данные для этого (update)
 образ -- часто файл для создания загрузочного лазерного диска операционной системы
 обрыв связи -- потеря понятных программе данных, не обязательно обрыв провода
 объём -- количество технической, не пользовательской информации
 огнелис -- браузер Firefox
 ОЗУ -- оперативное запоминающее устройство, устройство оперативной памяти
 окно -- прямоугольный "лист" с внешним выражением программы
 онлайн -- при выходе в интернет
 операционная система -- набор программ для базовых операций
 оптика -- оптическая линия связи
 оптический привод -- читатель\писатель лазерных дисков
 ОС -- операционная система
 открыть -- часто показать и подготовить к редакции файл с помощью программы
 офисный пакет -- набор типичных программ для офисных работ, а именно редакции текста, обработки бухгалтерии, ведения базы данных, создания презентаций, планирования работ и пр.
 оффлайн -- без подключения к интернету
 оффтоп -- посторонняя информация
 оцифровка -- перевод в цифровую запись, потенциально понятную компьютеру и смежным устройствам
 ошибка -- обычно ошибка программы
 пакет -- 1 единица поставки программ в UNIX и его потомках 2 единица передачи информации 3 набор программ
 пакетная запись -- способ упрощённой записи на лазерные диски
 память -- ёмкость для данных и программ
 панель -- обычно полоса на краю экрана со значками и кнопками
 панель задач -- панель с кнопками окон
 папка -- почти то же что каталог
 параметр -- часто параметр программы -- свойство программы
 пароль -- строка для проверки личности пользователя
 патч -- корректирующий файл
 паутина -- совокупность сайтов или ресурсов, доступных в браузере
 перезагрузка -- выключение и включение операционной системы
 переменные -- именованные части данных программы
 перетаскивание -- таскание чего-либо по экрану с помощью мыши
 переустановка -- обычно операционной системы или программы. удаление и установка заново
 периферия -- подключаемые, не всем нужные устройства
 ПЗУ -- постоянное запоминающее устройство, устройство

энергонезависимой памяти
 пиксель -- точка на дисплее
 письмо электронное -- обычно текст с адресом получателя и отправителя
 ПК -- персональный компьютер
 плагин -- программный подключаемый модуль
 планшет -- "компьютер" в виде книги, со встроенным экраном, экранной клавиатурой
 по умолчанию -- первоначально, до указания
 повторитель -- устройство, повторяющее сигнал беспроводной сети
 пользователь -- человек, работающий за компьютером
 порт -- 1 разъём 2 единица поставки программ в UNIX и его потомках 3 выпуск операционной системы NetBSD 4 выпуск программы
 постоянная память -- энергонезависимая память, не стираемая при отключении питания
 потеря данных -- потеря файлов, не потеря их нужного состояния, или потеря информации
 почтовая рассылка -- рассылка электронной почты
 почтовый ящик -- ёмкость для электронной почты
 права -- права на выполнение операций, преим. с файлами и каталогами
 приватность -- от слова "частный" (частная переписка и пр.)
 приложение -- прикладная, полезная программа
 принтер -- устройство для печати на бумагу
 провайдер -- организация, обеспечивающая выход в интернет
 программа -- "данные", способные что-то делать, выполнять работу в программируемом устройстве
 программирование -- создание программ
 программист -- создатель, редактор программ
 программный -- не технический
 прокси-сервер -- сервер, представляющий кого-то или что-то
 проникновение -- часто синоним взлома
 протокол -- способ взаимодействия программ
 прошивка -- 1 процесс смены программной среды 2 файл для этого
 прямая печать -- печать без компьютера
 публикация -- готовая "книга" или статья, преим. читаемая с экрана
 публичный айпи -- IP-адрес, достигаемый через интернет
 путь к файлу -- строка, показывающее положение файла
 рабочий стол -- основное пространство на экране при работе графического интерфейса
 пока нет окон программ
 раздел -- часть ёмкости HDD
 разметка -- процесс разделения HDD
 разработчик -- программист
 разрешение -- количество точек
 разъём -- гнездо
 раскладка клавиатуры -- набор вводимых символов
 расширение -- 1 последние символы имени файла 2 дополнение к браузеру
 расширения браузера -- дополнительные программные модули к браузеру
 регистрация -- создание учётной записи
 регулярные выражения -- строки для выполнения операций над множеством строк или всем текстом
 редактор -- программа для изменения
 резервирование -- резервное копирование. копирование для спасения ценных данных
 рип -- копирование видео или звука
 рунет -- российская часть интернета
 русификация -- адаптация к русскому языку
 сабж -- тема
 сайт -- совокупность страниц, обычно с текстом
 сборка -- 1 сборка техники 2 операционной системы 3 программы
 свисток -- портативный модем

своп -- раздел для сброса данных из оперативной памяти
сервер -- обслуживающий компьютер
сериальный номер -- уникальный номер, нужный для работы или установки программы
серийник - сериальный номер
серый айпи -- IP-адрес, недоступный через интернет
сетевая карта -- типичное оборудование, плата для выхода в интернет или подключения к локальной сети через провод витую пару
сетевая установка -- установка операционной системы или программы при которой идёт сбор программ из интернета
сетевой принтер -- принтер, доступный по сети
сетевой сканер -- сканер, доступный по сети
сетевой фильтр -- обычно фильтр напряжения питания
сеть -- вычислительные устройства и линии связи
сжатие данных -- процесс уменьшения количества данных
синхронизация -- копирование данных, повторение их в другом месте
система контроля версий -- система для хранения данных, позволяющая легко находить даже данные в старом виде
системные требования -- требования к аппаратуре и программам, операционной системе
системный блок -- основной блок компьютера
сканер -- устройство для "фотографирования" плоских объектов, преим. непрозрачных
скачивание -- копирование данных, доступных по сети
скорость -- количество данных, передаваемых за единицу времени
скринсейвер -- программа для уменьшения выгорания экрана, и блокировки работы с ним
скриншот -- снимок экрана
скрипт -- программа как текст. компиляция не нужна.
слэш -- косая черта, "\" или "/"
смайлик -- набор символов для передачи настроения, например скобка – улыбка
снимок -- часто совокупность данных операционной системы в какой-то момент, сохранённая любым способом с возможностью вернуться в это состояние
совместимость -- возможность совместной работы техники или программ, техники и программ
совместный доступ -- одновременный или последовательный доступ нескольких пользователей к ресурсу
сообщение -- обычно нечто вроде окна с текстом, который надо прочесть. способ общения программ с людьми.
софт -- программы, операционные системы. не данные.
сохранение -- запись в энергонезависимую память.
социальная сеть -- сеть для общения, обмена чем-то. не физическая и не виртуальная сеть.
спам -- рекламные, ненужные письма
справка -- вид документации о работе программ, операционной системы
спящий режим -- почти то же что ждущий режим
ссылка -- средство переноса к другому документу, или части того же документа, реже -- средство запуска процесса. Часто – подчёркнутый текст.
стандартный вход -- stdin в UNIX и его потомках. обычно ввод с клавиатуры
стандартный выход -- stdout в UNIX и его потомках. обычно вывод текста на экран
стандартный лист -- скорее всего лист формата А4
стартап -- новый проект
статистика -- скорее всего статистика событий на сервере
статический айпи -- неизменный IP-адрес
стилус -- "перо" для планшета или телефона, для управления ими и рисования, печати текста
стример -- одно из средств резервирования, аппарат
струйник -- струйный принтер
суперкомпьютер -- компьютер, занимающий целый зал.
тактовая частота -- количество актов работы в секунду

текстовый файл -- не двоичный, файл без изображений
 телетекст -- текст, передаваемый по телеканалу
 терминал -- обычно программа для работы с командами (в UNIX и его потомках)
 тильда -- "~"
 том -- обычно логический диск или том LVM
 тонер -- порошок для печати
 тонкий клиент -- "компьютер" только для основных функций, бесполезный без сервера и локальной сети, или интернета
 торрент -- файл для торрентной раздачи и скачивания, набор ссылок
 точка восстановления -- момент создания снимка операционной системы
 точка доступа -- оборудование для раздачи интернет-доступа по wi-fi
 трансляция -- передача видео или звука по сети
 трафик -- поток данных, количество данных
 троян -- "троянский конь" -- разновидность вредной программы
 тэг -- 1 ключевое слово 2 "команда" языка разметки HTML и прочих подобных
 тюнер -- устройство для захвата телевизионного сигнала и пр.
 убить -- выключит программу несмотря на то что она к этому не готова. это может привести к порче обрабатываемых этой программой данных
 удалённый доступ -- доступ из интернета с возможностью такой же работы какая делается рядом с компьютером
 усилитель сигнала -- устройство для увеличения радиуса приёма сигнала
 утилиты -- служебные программы, бесполезные сами по себе
 файл -- именованный массив данных или программа
 файл-проектор -- проектор изображений, хранимых в файлах
 файловая система -- способ хранения файлов
 факс -- факсимильный аппарат -- "принтер + телефон"
 фидбэк -- отзыв, сообщения разработчикам
 флавор -- разновидность выпуска операционной системы
 флуд -- болтовня
 флэшка -- брелок флэш-памяти
 форк -- программа, основанная на исходном коде другой программы
 формат -- способ хранения данных
 форматирование -- подготовка к работе носителя, разметка и смена файловой системы, стирание
 форум -- общедоступная переписка
 фриланс -- домашняя официальная работа
 фронтэнд -- "вывеска", внешняя программа, обслуживаемая более глубоко заложённой, невидимой программой
 хаб -- разветвитель
 хак -- 1 взлом 2 приём хакера
 хомячок -- малоподготовленный, мало умеющий, нелюбопытный, робкий пользователь
 хостинг -- 1 хранение данных сайтов, обеспечение их работы 2 VPS, VDS
 хранитель экрана -- скринсейвер
 цена печати -- стоимость бумаги + порошка + барабана
 ЦОД -- центр обработки данных
 чайник -- начинающий пользователь
 чат -- средство текстового общения, или общения через веб-камеру
 шрифты -- в компьютере файлы, содержащие очертания букв. Благодаря им документы похожи на книги
 электронная книга -- 1 устройство для чтения электронных публикаций 2 эти публикации
 электронная почта -- пересылка электронных писем
 электронное правительство -- средства предоставления услуг и данных с помощью информационных технологий
 элементная база -- простейшие детали и микросхемы
 элементы графического интерфейса -- простейшие, элементарные видимые средства

работы при работе графического интерфейса

юзабилити -- от "использовать", удобство, логичность

юзер -- пользователь

ЯД -- Яндекс Деньги

ядро -- основная часть операционной системы

язык программирования -- язык для написания программ

Яндекс -- российская фирма и поисковый сервис

A4 -- наиболее популярный в России лист бумаги

abort -- обрыв

access -- доступ

ADSL -- технология передачи данных со скоростью, достаточной для большинства работ

Android -- операционная система, в основном используемая на планшетах и многих телефонах

API -- средства управления программой для специалистов

ATA -- старое обозначение интерфейса PATA

bad gateway -- "плохой шлюз" -- одна из причин непоказа нужного сайта

Bada -- операционная система для некоторых телефонов Samsung

bash -- командный интерпретатор, популярный в потомках UNIX

BD -- тип лазерного диска, bluray disk

binary -- двоичный, программа

BIOS -- нестираемая программа материнской платы для определения важных свойств материнской платы

bitcoin -- наиболее известная криптовалюта

Bluetooth -- короткодействующая система передачи данных с небольшими скоростями, популярная в основном на телефонах

bluray -- тип лазерного диска

boot -- загрузка операционной системы или начало работы компьютера

bridge -- мост, соединение двух сетей

broadcast -- тип адреса в сети

BSD -- семейство операционных систем, первоначально разработанных в Беркли (США)

bug fix -- устранение ошибок

CD -- один из типов лазерного диска, обычно ёмкостью 700 Мб

COM -- тип порта с низкой скоростью, на новых компьютерах его нет

data -- данные

default -- по умолчанию, срыв значений к начальным

deny -- от "запрет"

desktop -- настольный компьютер обычного применения, не сервер

DHCP -- протокол для раздачи адресов

dial-up -- способ выхода в интернет по телефонной линии, устаревший

direct connect -- система передачи данных, используемая в некоторых виртуальных сетях

DNS -- сервис имён доменов

DOS -- уже почти нигде не используемая операционная система, популярная примерно до 1995 года

dpi -- "точек на дюйм"

DSL -- технология передачи данных

DVD -- один из типов лазерного диска, обычно ёмкостью 4,7 Гб

DVI -- тип файла, аппаратно-независимый файл

e-mail -- электронная почта

EDGE -- технология передачи данных в мобильной сети

encoding -- кодировка

encription -- шифрование

Enter -- клавиша, часто нажимаемая

environment -- окружение

error -- программная ошибка

error 404 -- отсутствие страницы по набранному адресу

ethernet -- технология передачи данных в сети, используемая сетевыми платами
fail -- крах, неудача
FAQ -- часто задаваемые вопросы
features -- возможности
firewall -- брандмауэр, программа для фильтрации сетевых данных
firewire -- протокол, часто использовавшийся для работы с цифровыми видеокамерами
flash -- технология, включающая в себя обеспечение многих сервисов в браузере, и видеоформат
forbidden -- запрещено
free software 1)бесплатные программы 2)программы, распространяемые при том, что вы можете устанавливать программное обеспечение на столько машин, сколько хотите. В один и тот же момент времени это программное обеспечение может использовать любое количество людей. Вы можете делать столько копий данного программного обеспечения, сколько хотите и раздавать их кому хотите (свободное или открытое распространение). Нет никаких ограничений на изменение данного программного обеспечения (если вы сохраните в неизменном виде некоторые замечания). Нет никаких ограничений на распространение или даже продажу этого программного обеспечения.
freeware -- бесплатные программы
FTP -- протокол передачи файлов, сервер для этого
getting started -- простое начало работы с чем-то
git -- программа для контроля версий
Google -- фирма и поисковый сервис
GPL -- одна из лицензий
GPRS -- медленный способ передачи данных из интернета на телефон
GPS -- система определения местоположения
GUI -- графический интерфейс
HD -- высокое разрешение
HDD -- типичный носитель энергонезависимой памяти, встроенный жёсткий диск компьютера
HDMI -- тип разъёма дисплея, телевизора
hostname -- имя компьютера
HTML -- язык для создания сайтов
HTTP -- протокол по которому передаются данные сайтов
HTTPS -- зашифрованный HTTP
IBM-PC -- персональный компьютер
ICQ -- программа для текстового общения, одна из них
ID -- идентификатор
IEEE1394 -- firewire
internal server error -- внутренняя ошибка сервера
iOS -- операционная система телефона iphone, планшета ipad
IP -- протокол интернета, адрес, формируемый для устройства
iptables -- программа для управления брандмауэром
IPv4 -- протокол интернета версии 4
IPv6 -- протокол интернета версии 6
IRDA -- инфракрасный порт
ISO -- расширение файла-образа
Jabber -- протокол текстового общения, один из них
jack -- звуковой разъём
Java -- технология и язык программирования
javascript -- язык программирования, работающий в основном на сайтах, в браузере
JPEG -- тип файлов с изображениями, способ сжатия графики
JRE -- набор программ для работы java-программ
JVM -- виртуальная машина java
Linux -- семейство операционных систем, точнее, их ядро
live -- лазерный диск, при помощи которого можно поработать с операционной системой без её установки
loop -- петля, интерфейс взаимодействия компьютера с самим собой

LPT -- принтерный, сканерный параллельный порт
LTE -- высокоскоростной стандарт для мобильных устройств
mac-адрес -- неизменный адрес сетевой карты
magnet-ссылка -- нечто вроде указателя на файл или адреса
MBR -- главный "загрузчик" операционной системы
md5 -- контрольная сумма, рассчитанная по методу md5
memstick -- "флэшка", или образ диска для записи на неё
MICRO USB -- тип разъёма
MINI USB -- тип разъёма
MIT -- одна из лицензий
MP3 -- способ хранения звука
MPEG4 -- способ хранения видео
NAS -- доступное по сети хранилище данных
networking -- сетевые работы
no carrier -- нет несущей, отсутствие сигнала
OCR -- распознавание текста, вынимание текста из изображений
partition -- раздел HDD
PATA -- параллельный интерфейс HDD
PATH -- переменная для поиска программ
PC -- персональный компьютер
perl -- язык программирования, один из них
permission -- права
php -- язык программирования, один из них
pocket writing -- пакетная запись
print screen -- печать экрана, сброс экранной картинки в файл или на бумагу
PS/2 -- разъём для клавиатуры или мыши
QWERTY-клавиатура -- полноценная клавиатура
RAID -- массив HDD для увеличения скорости и\или надёжности хранения данных и программ
RCA -- разъём "тюльпан"
reference manual -- уточняющее руководство
remote -- удалённый, находящийся далеко
requirements -- требования
root -- 1 корневой каталог 2 администратор
RSS -- ленты новостей
SATA -- тип интерфейса HDD
security -- безопасность
service -- службы
sha256 -- способ расчёта контрольной суммы
shared -- совместный, отдельный
shareware -- способ распространения программ, при котором можно пользоваться программой до её оплаты
SIM-карта -- необходимая карта телефона, она не привязана жёстко к номеру
Skype -- программа для видеотелефонии
slice -- раздел в терминологии BSD
SMART -- система самодиагностики HDD
smart TV -- телевидение с выходом в интернет
snapshot -- часто снимок состояния виртуальной машины
soft -- программы
SOHO -- малый офис, домашний офис
Solaris -- семейство операционных систем
source -- исходный код программы, текст её
SSD -- жёсткий диск с флэш-памятью
SSH -- программа для шифрованного управления компьютером
stdin -- стандартный вход (см тут же)

stdout -- стандартный выход (см тут же)
storage -- хранилище
superuser -- администратор
Symbian -- операционная система для некоторых телефонов
TCP\IP -- семейство протоколов интернета
text wrap -- перенос строк
Thunderbolt -- порт, тип разъёма
timeout -- ограничение по времени
tuning -- тонкая настройка
tv box -- телеприставка, превращающая телевизор в компьютер с Android
TV OUT -- телевизионный выход, выход телесигнала
Unices -- всё семейство UNIX
unicode -- способ хранения текста
UNIX -- семейство операционных систем с более ранними корнями чем у Windows
update -- обновление
URL -- указатель, адрес
USB -- тип разъёма
USB 2 -- тип разъёма
USB 3 -- тип разъёма
user guide -- руководство пользователя
VDS -- служба, обеспечивающая автономную операционную систему и приложения, доступные через интернет
VGA -- разъём дисплея
VoIP -- голос через протокол интернета
VPN -- виртуальные частные сети
VPS -- служба, обеспечивающая автономную операционную систему и приложения, доступные через интернет
wake ower lan -- "включение" компьютера через сеть
WAP -- протокол, позволяющий простые службы, доступные на мобильном телефоне через интернет
warning -- внимание
webmoney -- популярный тип электронной валюты
Wi-Fi -- один из беспроводных способов связи
wiki -- набор статей
wikipedia -- доступная в браузере энциклопедия
WIMAX -- скоростной тип связи для мобильных устройств
Windows -- семейство операционных систем (все платные)
WOL -- wake on lan
WYSIWYG -- "что видите то и получаете"
X -- обиходное название оконного сервера xorg, программы для организации графического интерфейса в UNIX и его потомках