

# Text-Terminal-HOWTO

David S. Lawyer

Русский перевод Ilgiz Kalmetev [ilgiz@mail.rb.ru](mailto:ilgiz@mail.rb.ru)

v0.05, June 1998

---

*Этот документ объясняет, что такое текстовые терминалы, как они работают, как устанавливать и конфигурировать их и дает некоторую информацию об их ремонте. Он может отчасти использоваться, даже если у вас нет руководства по терминалу. Хотя этот труд написан для реальных терминалов на системе Linux, часть его также применима для эмуляторов терминала и/или других Unix-подобных систем.*

---

## 1. Введение

Если вы хотите попытаться быстро установить терминал, то см. ``Быстрая установка``.

### 1.1 Авторские права, оговорки и об авторе

#### Авторские права

Авторские права принадлежат David S. Lawyer (1998). Пожалуйста, свободно копируйте и распространяйте этот документ. Дайте мне знать (если можете меня найти и если я жив) в том случае, если вы значительно изменяете этот документ. Если некоторые слова - это марки изготовителя, то контекст должен пояснять кому они принадлежат. Например "MS WINDOWS NT" подразумевает, что "Windows NT" принадлежит Microsoft (MS). Mac - компьютер фирмы Apple.

#### Оговорки

Многие положения в этом HOWTO были получены из Интернета и могут быть ненадежными (хотя в некоторых случаях я делал двойную проверку).

В то время как я не собираюсь преднамеренно вводить вас в заблуждение, в этом документе вероятно имеется ряд ошибок. Пожалуйста, сообщите мне о них. Должно быть очевидно, что я не могу по закону отвечать за них.

#### Контакт с автором

Мой email в 1998 - [bf347@lafn.org](mailto:bf347@lafn.org). Пожалуйста, сообщайте мне о любых ошибках в фактах, мнениях, логике, проверке правописания, грамматике, ясности изложения, связях, и т.д. Я понимаю, что этот документ нуждается в некотором усовершенствовании, так что я надеюсь пересматривать его около одного раза в месяц. Пожалуйста, присылайте мне любую информацию, о которой вы думаете, что она относится к этому документу.

## 1.2 Что такое терминал?

Терминал состоит из экрана и клавиатуры, которые совместно используются, чтобы дистанционно связаться с компьютером (хостом). Он используется так, словно он - персональный компьютер, но терминал удален от главного компьютера (стоит у другой стены комнаты или даже в другом полушарии). Программы выполняются на главном компьютере, но результаты отображаются на экране терминала. Вычислительная способность терминала - относительно низкая (иначе это был бы компьютер, а не терминал).

Его вычислительные возможности обычно ограничены возможностью отображать то, что ему послано (возможно, включая полноэкранную графику).

В дни универсальных ЭВМ с середины 1970-х до середины 1980-х большинство людей использовало терминалы, чтобы связываться с компьютерами. Они печатали в программах, запускали программы, писали документы, выдавали команды печати и т.д. Терминал соединялся с компьютером кабелем (часто ненапрямую). Он стал называться терминалом, так как был размещен на терминальном конце этого кабеля (оконечное оборудование).

Если вы использовали компьютер под Linux (исключая использование X-Window) с монитором и клавиатурой, то вы уже знаете что такое терминал, потому что вы его использовали (или точнее "виртуальный терминал"). Монитор (совместно с клавиатурой) называется консолью, но это только эмуляция терминала.

Реальный терминал отличается от монитора, потому что обладает другими электронными параметрами. Текстовый терминал часто соединен с последовательным портом компьютера через длинный кабель. Таким образом, в отличие от монитора, который обычно размещен рядом с компьютером, терминал может быть расположен очень далеко от главного компьютера. Видеоплата внутри компьютера сохраняет видеоизображение, показываемое на экране монитора. Для терминала эквивалент такой видеоплаты встроен прямо в терминал, но так как текстовые терминалы часто одноцветные, без мощной графики, возможности этой "видеоплаты" довольно слабые. Также большинство текстовых терминалов не имеет мышей.

В сетевой терминологии, принято считать, что терминал - это клиент, а главный компьютер - это сервер. Это не совсем так, поскольку процесс "обслуживания" терминала хостом состоит в приеме символов, печатаемых на клавиатуре и реакции на них, также как в случае с компьютером.

Терминал - точно так же как монитор подобен окну в компьютер. Вы, возможно, уже использовали виртуальные терминалы в Linux (нажимая Alt-F2, и т.д.). Реальный терминал работает точно так же, как этот виртуальный терминал, но на собственном экране и клавиатуре. В отличие от использования виртуального терминала в консоли, он позволяет другому человеку находиться у реального терминала и использовать компьютер одновременно с другими пользователями.

## 1.3 Dumb терминалы

Есть множество различных, противоречащих друг другу определений "dumb терминалов", но поскольку время идет, все больше и больше терминалов называются dumb. Этот документ главным образом относится к текстовым терминалам, которые отображают только текст на экране.

Это могло бы быть названо "Dumb-Terminal-HOWTO", но в некоторых компьютерных журналах любые терминалы, независимо от их интеллектуальности, включая предоставляющих полный графический

интерфейс пользователя (GUI), называются dumb. Если все терминалы "dumb", тогда нет смысла приписывать слово "dumb" к терминалу (за исключением коммерческих целей, чтобы продать компьютеры или подобный товар вместо "интеллектуальных" терминалов). Из-за неоднозначного значения "dumb терминалы" не классифицируются здесь как тип терминала.

## 1.4 Типы терминалов

### Текстовые терминалы

Для текстового терминала по кабелю, соединяющему компьютер и терминал, идет двунаправленный поток информации. Этот поток является байтами ASCII, где каждый байт обычно соответствует символу. Байты, печатаемые на клавиатуре идут на компьютер, а большинство байтов от компьютера отображается на экране терминала. Специальные байты (или последовательности байтов) от компьютера сообщают терминалу, куда переместить курсор, что стереть, где начать и закончить подчеркивание и/или мигание и/или утолщение шрифта и т.д.

Зачастую имеется сотня таких специальных команд, и многие терминалов могут даже изменять шрифты.

Соединение использует символы (символы), закодированные с использованием кодовой раскладки для используемого символического набора. Обычно первые 128 байтов из 256 возможных байтов используют коды ASCII. Терминалы для unix-подобных систем обычно соединяются с компьютерами через кабель, установленный между асинхронными последовательными портами (на RS-232-C = EIA-232-D) главного компьютера и терминала. Иногда соединение идет через модем или терминальный сервер и т.д.

Другие названия для текстовых терминалов - "последовательный терминал", "терминал с символическими ячейками", "ASCII терминал", "ANSI терминал", "символьный терминал", "терминал данных", "видео терминал" и "видеодисплейный терминал" (VDT).

В старые дни для терминалов использовался термин "видеодисплейный блок" (VDU), но строго говоря, это определение исключает клавиатуру.

Старые терминалы майнфремов IBM используют "блочный режим". Символы, которые вы напечатаете, временно хранятся в памяти терминала (и их можно отредактировать встроенным редактором в терминале). Затем, когда нажата посылающая клавиша (или что-нибудь в этом роде), блок символов (иногда только строка символов) целиком посылается компьютеру. Такие терминалы не возможно использовать в Linux. Они фактически являются блочными устройствами (а не символическими). См. раздел ``Блочный режим".

### Графические терминалы

В ограниченной степени некоторые ASCII символы могут обеспечивать графику на текстовых терминалах. Можно формировать стрелки: < --- и рисовать рамки с помощью \_ и |.

Со специальными графическими символическими наборами, возможности еще шире. Ни один из них не является действительно графическим терминалом. Однако термин "графический терминал" иногда применяется к всем текстовым терминалам, так как текст - ограниченная форма графики.

Имеются два основных типа графических дисплеев: растровые и векторные (редко используемые). Растровая графика выдает точки на экран горизонтальными строками развертки, рисуемые электронным лучом (или, возбуждением пикселей или точек на плоском экране). Векторные графические дисплеи используют интеллектуальную электронику, чтобы рисовать строки и кривые электронным лучом, который может двигаться в любом направлении. Векторная графика рисует строки высокого качества без зигзагов, но они редки и дороги. Растровая графика сегодня используется почти универсально. Для PC иногда используются изображения, закодированные в векторном графическом формате, но для дисплея они транслируются в формат растровой графики (со снижением качества изображения).

## **Графические терминалы последовательных линий**

Многое в этом документе также применимо к ним. Большинство из них может также функционировать как текстовые терминалы. Protocols для такой графики включают: Tektronix Vector Graphics, ReGIS (DEC), Sixel (DEC) и NAPLPS (North American Presentation Level Protocol Syntax). Все они монохромные??

## **Быстрые графические терминалы (часто известные под другими названиями)**

Ни один из них не рассмотрен в этом документе. Терминал, который заслуживает названия интеллектуального, - это графический терминал, который может отображать полноэкранную графику так же быстро, как монитор PC. Также он будет иметь мышь.

Байты, посланные ему часто являются растрами изображений (и другой графикой). Он часто подключен к компьютеру по быстродействующему соединению, используя витую пару или коаксиальный кабель. Терминалы X-Window являются именно такими устройствами.

Для отображения MS-Windows GUI имеется множество различных типов интерфейсов и терминалов: один из них Winterm, использующий программное обеспечение WinFrame от Citrix. Другой (основанный частично на коде Citrix) - Hydra (кодовое имя) от Microsoft, также известный как "Windows Terminal Server", который работает с версиями MS WINDOWS NT 4 или выше. Citrix использует протокол ICA, и было создано расширение к Hydra, известное как pICasso, чтобы WinFrame (ICA) терминалы могли использовать систему Hydra. Hydra также является многопользовательской. Имеется также "Многоконсольный Персональный Терминал" от Unbounded Technologies и Tektronix имеет многопользовательский интерфейс, но теперь будет поддерживать Hydra. В 1997 году вышла журнальная статья, названная Winterm "dumb терминал", но она по сей день довольно интеллектуальна. Такие терминалы часто называются "тонкими клиентами", но некоторые тонкие клиенты - это больше, чем просто терминалы, поскольку они могут выполнять код Java, посланный им, и т.д.

## **Квази-терминалы (= Квази-компьютеры)**

"Квази" - это моя терминология (не стандартная). Они не являются ни настоящими компьютерами, ни терминалами, а представляют собой нечто среднее. Сетевые Компьютеры (NC'S) - это компьютеры с микропроцессором, но без жесткого диска. Они полностью графические и соединяются с компьютером сервера. Они отличаются от терминалов тем, что выполняют программ на своих собственных микропроцессорах. Они могут выполнять код Java. Они должны работать в IP сетях и могли бы работать под сервером с Linux. Wintel выпустил "NetPC" который, в отличие от NC, является почти компьютером PC.

Однако, на нем не установлены сменные диски, так что пользователи не могут устанавливать собственное программное обеспечение или копировать что-нибудь.

## Эмуляция терминала

Хотя PC имеет экран и клавиатуру (как и терминал), но также он намного мощнее по вычислительной мощности, он может легко использовать часть этой вычислительной мощности, чтобы притвориться текстовым терминалом. Программное обеспечение для этого широко доступно и есть в новых версиях MS WINDOWS. Программа эмуляции часто объединена с программой-звонилкой для модема (типа Minicom для Linux) так, чтобы можно было (например) звонить общественные библиотеки, чтобы использовать их каталоги и индексы (или даже читать журнальные статьи). Чаще всего эмулируют терминал VT100 (текстовый терминал). Иногда эмуляция - не 100%-точная, но обычно это не вызывает много проблем.

Реальный терминал зачастую лучше, чем его эмулятор. Обычно стоит он меньше, часто имеет лучшую разрешающую способность в текстовом режиме, и на нем нет раздражающих своим шумом дисководов. Некоторые реальные терминалы могут эмулировать различные модели терминалов, но все-таки остаются реальными терминалами.

## 1.5 Терминология

Конфигурация означает тоже, что и установку. В то время как команды Linux принимают опции (используя символы), опции в более широком смысле включают различные типы выборов. Установка в широком смысле включает установку (конфигурирование) программного обеспечения и оборудования. Утверждение, в котором я сомневаюсь заканчивается 2 вопросительными знаками:?? Сообщите мне, если вы знаете, как это будет правильно (или об ошибке).

## 2. Быстрая установка

Это быстрая процедура установки терминала без того, чтобы проходить через процедуру ``Установки" и для терминала, и для главного компьютера. Вероятно этот способ не будет работать правильно, если данный терминал вдруг окажется несовместимым с компьютером. Если вы что-то из этого не понимаете, то вам надо будет проконсультироваться в других частях этого документа для принятия большего количества информации.

Чтоб установить терминал, сначала посмотрите в /etc/termcap или terminfo.src, чтобы найти запись о нем (см. ``Terminfo и Termcap (подробно)"). Попробуйте "vt100", если не можете найти что-нибудь подходящее. Выясните, с каким последовательным портом вы соединены и какой tty назначен этому порту (например ttyS1, см. ``Имена устройств"). От root отредактируйте /etc/inittab, и добавьте команду getty к другим командам getty. Формат команды getty зависит от того, какую программу getty вы используете (см. Serial-HOWTO 4.1 и 7.2, и/или напечатайте "man getty").

Для параметров getty используют имя terminfo (или termcap) (типа vt100) для вашего терминала. Напечатайте скорость пересылки данных в бодах, которую поддерживает терминал. Но если вы устанавливаете слишком высокое значение бод, то вам возможно придется использовать (см. ``Управление потоком данных").

Затем физически соедините последовательные порты терминала и компьютера вместе нуль-модемным кабелем, и включите терминал. Не ожидайте, что большинство готовых кабелей подойдет для аппаратного управления потоком данных. Удостоверьтесь, что скорость пересылки данных терминала в бодах установлена такой же, какую вы задали для getty. Затем на консоли компьютера введите "init q", чтобы применить изменения, которые вы сделали в файле inittab. Теперь вы должны видеть подсказку входа в систему в терминале. Если нет, то нажмите клавишу RETURN терминала. Если это не помогло, то читайте этот документ дальше, и/или см. ``Поиск неисправностей".

## 3. Зачем использовать терминал?

### 3.1 Введение

PC настолько мощны сегодня, что один такой компьютер часто может поддерживать работу нескольких людей сразу, особенно, если они выполняют не ресурсоемкие задачи типа редактирования текста, ввода данных, и т.д. Один из способов сделать это состоит в том, чтобы соединить ряд терминалов с хостом по модему или прямым кабельным соединением. Для этого нужна многопользовательская операционная система, такая как Linux. Это называется "режимом разделения времени", но этот термин сегодня уже не так хорош, так как "распределенные" вычисления по сети - это тоже тип режима разделения времени. Вероятно лучше описать его как "централизованное" вычисление. Но центральный компьютер может быть соединен с остальной частью мира через сеть, по которой могут подключаться пользователи терминалов, так что это сейчас это и не "централизованное" тоже.

Терминалы редко использовались с PC, потому что популярные операционные системы, используемые для них (Windows, DOS, и Mac) не были многопользовательскими до 1998 (доступны для MS WINDOWS NT) и не могли поддерживать терминалы. Теперь, когда многопользовательская операционная система Linux является свободно доступной для PC, использование терминалов с PC становится более возможным. Недостаток их в том, что текстовые терминалы, не достаточно интеллектуальны, чтобы поддерживать графический интерфейс пользователя (GUI), на который сегодня рассчитывают многие пользователи компьютеров.

### 3.2 Более низкие аппаратные затраты

Если несколько людей используют один и тот же самый компьютер одновременно, то получается уменьшение количества аппаратных средств, необходимых для того же самого уровня обслуживания. Одна причина сокращения - совместное использование кода. Файлы приложения на жестких дисках разделены также как общедоступные библиотеки в памяти (даже, когда люди выполняют различные программы, если они используют некоторые из тех же самых функций в их коде). Другая причина - уменьшение пиковой загрузки. Аппаратные средства одиночного PC могут быть неактивны большинство времени так как люди медленно печатают информацию, думают, говорят или отлучились со своего рабочего места. Наличие нескольких людей на одном и том же компьютере сразу сокращает время простоя, которое иначе было бы потрачено впустую.

Эта экономия реальна. Можно грубо оценить, что для 9 людей (8 терминалов и 1 консоль) общедоступный PC нуждается приблизительно всего лишь в 3-кратно большей емкости ресурсов (оперативной памяти, дисковой памяти, микропроцессоре(ах), и т.д.), чем отдельный PC, чтобы обеспечить тот же самый уровень обслуживания на человека (9-кратная нагрузка). Таким образом вычислительные аппаратные средства для такой общедоступной системы должны стоить в три раза дешевле в пересчете на одного пользователя.

Однако, стоимость аппаратных средств отображения (электронно-лучевых трубок, клавиатуры, видеоэлектроники, и т.д.) - относительно те же самые для обоих случаев. Терминалы имеют дополнительные затраты за счет требования большего количества последовательных портов на главном компьютере.

Для честного сравнения с PC, терминалы должны иметь те же самые возможности что и мониторы PC. К сожалению, цветные графические терминалы для Linux (X-Window) с высокоскоростной связью - это рыночная ниша с высокими ценами, так что в данном случае особой экономии на аппаратных средствах не будет. Но для текстовых терминалов экономия будет реальной.

### 3.3 Управление программным обеспечением

Для централизованного вычисления, программное обеспечение (и модификации к программному обеспечению) должно быть установлено только на одном главном компьютере вместо нескольких. Человек, отвечающий за этот компьютер, может управлять программным обеспечением, установленным на нем. Это выгодно, если человек, управляющий главным компьютером, - настоящий профессионал и знает о потребностях и предпочтениях других пользователей. Отсутствие на главном компьютере соответствующего программного обеспечения (или применение других мер по ограничению доступа) мешает пользователям играть в игрушки или лазить по Интернет. Нужно или нет централизованное управление - это зависит от ситуации.

### 3.4 Аппаратные обновления

С применением терминалов, аппаратные обновления компьютеров происходят только на одном компьютере вместо нескольких. Это ускоряет и облегчает установку.

В то время как стоимость аппаратных средств для главного компьютерного обновления будет выше, чем одного РС (так как главная ЭВМ нуждается в большей вычислительной мощности, чем РС), общая стоимость обновления будет значительно меньше, чем обновление всех РС, используемых вместо терминалов.

### 3.5 Другие преимущества терминалов

1. В терминалах не шумят вентиляторы и диски, как в компьютере
2. Пользователи терминалов могут совместно использовать данные и файлы и даже посылать электронную почту друг другу. Это является подобием локальной сети.

### 3.6 Главные недостатки терминалов

1. В случае текстового терминала, он не имеет быстродействующего графического дисплея (или графики высокого разрешения), хотя может иногда использовать графические символьные наборы, чтобы рисовать линии и т.д. Этот недостаток ограничивает число используемого на нем программного обеспечения.
2. Если главный компьютер выключается, то терминалы тоже перестают работать (если их нельзя подключить к другому компьютеру).

### 3.7 Являются ли текстовые терминалы устаревшими?

Текстовые терминалы технологически устаревают, потому что за немного большую стоимость можно применить интеллектуальный терминал (с дисплеем того же качества). Не всегда было так, например в 1980 году память стоила тысячи долларов за мегабайт. Сегодня память и процессоры стоят немного, и можно сделать текстовый терминал интеллектуальным за счет увеличения стоимости только на 10% или 20%.

Причины, по которым текстовые терминалы до сих пор не устарели:

- Многие люди не нуждаются в полноэкранной графике.
- Единственный интеллектуальный терминал для Linux - X-Window терминал, который может стоить больше чем PC с запущенным Linux.
- Используемые текстовые терминалы - часто очень дешевы, но могут предоставлять доступ к гораздо более новым (и мощным) компьютерам.

## 4. Кратко о том, как работают терминалы (в Linux)

См. также раздел ``Некоторые подробности о том, как работают терминалы''

### 4.1 Имена устройств

Каждый терминал соединен с последовательным портом главной ЭВМ (часто только PC). Порты имеют названия: ttyS0, ttyS1, ttyS2 и т.д.

Они представляются в виде специальных файлов, находящихся в каталоге устройств /dev. /dev/ttyS0 соответствует COM1 в DOS или Windows. ttyS1 - COM2, и т.д. См. ``Терминальные устройства'' для подробностей.

### 4.2 Вход/выход в/из системы

Когда главный компьютер запускается - он запускает программу `getty` (см. Serial-HOWTO 4.1 и 7.2) на каждом последовательном порту, к которому подключен терминал (как определено в файле `/etc/inittab`). Программа `getty` выполняет программу `"login"`, чтобы регистрировать людей в системе. На экране появляется подсказка `"login: "`. Люди на терминалах входят в систему (при предоставлении ими паролей) и затем получают доступ к компьютеру. Когда приходит время выключения терминала, обычно выходят из системы и отключают питание. См. ``Ограничение входа в систему'' об информации по ограничению входа в систему (включая разрешение корневому пользователю регистрироваться с терминала).

### 4.3 Полу/полный дуплекс

Если кто-то печатает на терминале, то буквы одновременно появляются на экране. Наивный человек мог бы думать, что то, что он печатает, послано непосредственно с клавиатуры на экран, с копированием на компьютер (называется полудуплекс). Обычно то, что печатается на клавиатуре, посылается непосредственно главной ЭВМ (называется дуплекс), которая в свою очередь отправляет обратно на терминал каждый полученный символ. В некоторых случаях (типа паролей или краткого редактора команды) печатаемые символы не отображаются обратно на экран.



## 4.4 Память терминала

Изображение на электронно-лучевой трубке угасает почти немедленно, если его не перерисовывать часто на экране лучом электронов на экране трубки. Так как текст, посланный терминалу должен остаться видимым, изображение на экране должно быть сохранено в микросхемах памяти терминала, и электронный луч должен неоднократно пробегать по экрану (скажем 60 раз в секунду) чтобы поддержать изображение. См. подробности ``Память терминала".

## 4.5 Команды терминала

Терминал управляется компьютером. Компьютер посылает терминалу текст не только для отображения на экране, но также и посылает команды терминалу, которые должны выполняться. Они называются ``управляющими кодами" (байтами) и ``escape-последовательностями". Например, управляющий код CR (возврат каретки) перемещает курсор на левый край экрана. Некоторая escape-последовательность (несколько байтов, где первый байт - управляющий код "Escape") может перемещать курсор в место на экране, определенное параметрами, помещенными внутри escape-последовательности.

``Первые терминалы" имели только несколько таких команд, но современные терминалы, имеют их сотни. Вид дисплея может быть изменен для некоторых параметров: типа яркости, тусклости, подчеркивания, мигания и негативного изображения. Динамик в терминале может "пищать", когда нажата какая-то клавиша или подавать звуковой сигнал, если произошла ошибка. Функциональные клавиши могут программироваться для специальных значений. Можно подключать различные шрифты. Дисплей может быть пролистан вверх или вниз. Определенные части экрана могут быть стерты. Могут использоваться различные типы управления потоком данных, чтобы остановить поток данных, когда байты посланы терминалу быстрее, чем терминал успеет их обрабатывать. Имеется еще много чего, о чем вы можете прочесть в руководстве по терминалу.

## 4.6 Недостаток стандартизации, решенный с помощью terminfo

В то время как терминалы, сделанные для США, используют для алфавита одинаковую для всех кодировку ASCII (кроме IBM-терминалов, которые использовали EBCDIC), они, к сожалению, не используют одинаковые escape-последовательности. Это случается даже после принятия различных стандартов ANSI, так как эти стандарты не продвигались как следует. Кроме того, старые терминалы часто испытывают недостаток возможностей более новых терминалов.

Это может вызвать проблемы. Например, компьютер может послать терминалу escape-последовательность, заставляющую его разбить экран на два окна определенного размера, не понимая, что терминал неспособен это выполнить.

Для преодоления этих проблем была принята база данных, называемая "termcap" (теперь "terminfo"). Эта база данных постоянно находится в некоторых файлах на компьютере и имеет разделы (иногда отдельные файлы) для каждой модели терминала. Для каждой модели (типа VT100) приведен список возможностей, включая доступные для использования escape-последовательности и т.д. См. раздел ``termcap и terminfo (подробно)" для подробностей. Прикладные программы могут использовать эту базу данных, вызывая некоторые программы C-библиотеки. Один большой набор таких программ (более чем 200) называется "ncurses" и перечислен на map-странице для ncurses.

## 4.7 Интерфейс

Переменная среды TERM указывает Linux, какой тип терминала вы используете. Прикладные программы используют ее, чтобы определить возможности терминала в базе данных terminfo, так что переменная TERM должна быть установлена правильно. Но имеется более правильный интерфейс, чем знание компьютера о возможностях терминала.

Для байтов, идущих с компьютера на терминал, терминал должен быть установлен так, чтобы получать байты на той же самой скорости (бит в секунду), с какой они идут с терминала. Если терминал установлен на скорость приема 19,200 бод, а компьютер посылает символы на скорости 9600 бод, то на экране будет виден мусор (или возможно ничего).

Одна из возможностей выбора скорости для терминала (также как и много других возможностей) - из меню "Установка" на терминале. Большинство терминалов имеет большое количество опций в своих меню "Установка".

Компьютерное программное обеспечение последовательного порта также имеет опции, и эти опции должны быть установлены совместимым способом.

## 4.8 Эмуляция

Большинство терминалов сегодня могут работать как несколько типов терминалов (эмуляция).

Номер модели терминала терминалов прежде производимых DEC (Digital Equipment Corporation) начинаются с VT (например, VT100). Многие другие терминалы, не VT100, могут быть настроены на эмуляцию VT100. Wyse - главный изготовитель терминалов, и большинство их терминалов может эмулировать различные терминалы DEC, такие как VT100 и VT220. Таким образом, если вы хотите, скажем, использовать терминал VT320, вы можете или использовать реальный VT320 в "настоящей" индивидуальности, или возможно использовать какой-то другой терминал, способный эмулировать VT320. "Настоящие" обычно имеют большее количество возможностей, так что, при прочих равных условиях, обычно лучше использовать "настоящие".

Наиболее общий тип эмуляции - это использовать PC как будто он - терминал. Это осуществляют программы, загруженные в память PC. В Linux (если вы не находитесь в X-Window) монитор PC (называемый консолью) эмулирует терминал. Даже некоторые окна в X-Window эмулируют терминалам.

## 4.9 Консоль

На PC монитор известен как консоль. Он эмулирует терминал типа "Linux". Регистрируются на нем как на виртуальном терминале. См. ``Консоль: /dev/tty?``. Он получает сообщения от ядра о запуске и выключении системы. Можно сообщения, которые обычно идут на консоль, перенаправить на терминал, исправляя и перекомпилируя Linux. (См., ``Делаем консоль терминалом``).

## 5. Специальные файлы терминалов: /dev/tty ... и т.д.

"tty" - сокращение от "телетайп". Первые терминалы были телетайпами (типа дистанционно управляемых пишущих машинок). См. подраздел ``Телетайпы``.

## 5.1 Терминал на последовательном порту

Компьютер рассматривает терминал как устройство на последовательном порту.

Для каждого такого терминала имеется специальный файл в каталоге (устройств) `/dev`. `/dev/ttyS0` - специальный файл для последовательного порта, известный как COM1 в мире DOS/Windows. Чтобы посылать текст терминалу, вы можете переназначить стандартный вывод некоторой команды в командной строке соответствующему специальному файлу. Например команда `"echo test > /dev/ttyS1"` должна послать слово "test" на терминал, подключенный к COM2 (подразумевается, что вам разрешена запись в `/dev/ttyS1`). Точно так же `"cat my_file > /dev/ttyS0"` пошлет содержание файла `my_file` в COM1 (`tyS0`).

В дополнение к `tyS0` (`/dev/ttyS0`), `tyS1`, `tyS2`, и т.д. ("S" означает последовательный порт) имеется также ряд устройств "cua": `cua0`, `cua1`, `cua2`, и т.д.

`cua0` - тот же самый порт как `tyS0` и т.д. "Cu" `cua` означает CalloUt. Серия устройств `tyS` соответствуют Posix, в то время как при использовании `cua` можно открыть порт, про который линии управления модемом говорят, что он не готов. Это говорит о том, что ряд `cua` включен только для обратной совместимости, так что вы видимо должны использовать ряд `tyS`.

## 5.2 Псевдотерминалы

Псевдо терминалы не имеют уникального физического разъема на компьютере.

Они используются для эмуляции последовательного порта. Например, если кто-то соединяется через telnet с вашим компьютером по сети, они могут подсоединиться к устройству `/dev/ptyp2` (порт псевдотерминала). В XWINDOWS, программа эмуляции терминала `xterm` использует псевдотерминалы.

Также его используют программы любительского радио. При использовании некоторого прикладного программного обеспечения можно иметь 2 или более псевдотерминала, присоединенных на один и тот же физический последовательный порт.

Псевдотерминалы образуют пары типа `ttyp3` и `ptyp3`. `pty...` - это хозяин или управляющий терминал, а `tty...` - подчиненный. `tyq5` - также псевдотерминал, как - `tysc` (с - шестнадцатеричная цифра).

Более точно, главные псевдотерминалы - `/dev/pty[p-s]n`, а соответствующие подчиненные - `/dev/tty[p-s]n`, где `n` - шестнадцатеричная цифра.

## 5.3 Управляющие терминалы /dev/tty

`/dev/tty` означает управляющий терминал (если есть) для текущего процесса (процесс, который использует `"/dev/tty"` в команде). Чтобы выяснять, к которому `ty` присоединен процесс, используют команда `"ps -a"` в командной строке. Посмотрите столбец "ty". Для оболочки, в которой вы находитесь, `/dev/tty` - это терминал, который вы сейчас используете.

Напечатайте "tty" в командной строке, чтобы увидеть название терминала (см. `man tty(1)`). `/dev/tty` - нечто вроде связи с фактически именем устройства терминала с некоторыми дополнительными возможностями для программистов на C : см. `man tty(4)`.

## 5.4 "Терминалы" `/dev/ttyI`?

? означает целое число. Одно из применений их в Linux - пакетный драйвер ISDN: `isdn4linux`. `ttyI?` является неким подобием `ttyS?`. Имеется также `cui?`, который является кое-чем вроде `cua?`. `ttyI` и `cui` эмулируют модемы и могут принимать команды модемов.

## 5.5 Консоль: `/dev/tty?`

В Linux монитор PC называется консолью и имеет несколько специальных файлов устройств, связанных с нею: `tty0`, `tty1`, `tty2` и т.д. Когда вы входите в систему, то попадаете на `tty1`. Чтобы перейти на `tty2` нажмите `Alt-F2`. `tty1`, `tty2`, и т.д. - "виртуальные терминалы" (иногда называются "виртуальными консолями"). вы можете регистрировать на различных виртуальных терминалах и таким образом осуществлять несколько различных сеансов с компьютером в одно и тоже время. вы переключаетесь между ними, используя клавиши `Alt-F?`, где "?" является номером виртуального терминала, на который вы хотите перейти. Консоль также известна как `/dev/tty0`, и сообщения системы могут идти на это устройство и отображаться на вашей консоли. Только система или `root` могут писать на `/dev/tty0`, с которым иногда компонуется `/dev/console`. Сообщения ссемы могут быть также записаны непосредственно по аппаратному адресу платы последовательного порта, таким минуя `/dev/tty0`. Для большего количества информации о консоли см. ``Консоль Linux".

## 5.6 Создание устройства командой `mknod`

В каталоге `/dev` находится множество специальных файлов устройств. Если нужных вам файлов нет там, то вы можете попробовать создать их командой "mknod". См. `man`-страницу `tty(4)` для информации о том, как сделать это для последовательных портов. Чтобы использовать `mknod`, вы должны знать `major` и `minor` числа устройства. вы можете посмотреть нужные вам числа, используя команду "`ls -l`" в каталоге `/dev`. Она отобразит `major` и `minor` числа существующих специальных файлов.

# 6. Некоторые подробности о работе терминалов

Если вы мало знаете о терминалах, предлагаю вам прочесть первый раздел: ``Введение", а также ``Краткий обзор работы терминалов".

## 6.1 Память терминала

Регенерации изображения терминала самостоятельно происходит 60 раз в секунду из изображения, сохраненного в памяти о терминале. Для PC изображение монитора хранится в видеоплате внутри компьютера, а для терминала, эквивалент видеоплаты находится внутри терминала. Для текстового терминала для хранения изображения используется небольшой объем памяти. Вместо того, чтобы хранить в памяти каждый пиксел экрана и тратить память на хранение около четверти миллиона точек, используется намного более эффективный метод хранения.

Экран, заполненный текстом можно представить в памяти терминала ASCII байтами, по одному для каждого символа на экране. Весь экран занимает около 2КБ ASCII байтов. Чтобы отобразить эти символы, терминал должен также знать растр (форма) каждого из почти 100 печатаемых символов ASCII. При использовании на растр символа, скажем, 15 байтов, необходимо приблизительно только 1.5КБ памяти для хранения растров всех символов ASCII (шрифта). Этот текст ASCII и память шрифта просматриваются так, чтобы возникающее в результате изображение выводилось на экран приблизительно 60 раз в секунду. Это форма общедоступной памяти, где одиночный растр символа типа символа "е", разделен между всеми "е" символами, которые появляются на экранном тексте. Низкие требования к памяти привели к низким издержкам производства мониторов в начале 1980-ых, когда стоимость памяти была в несколько тысяч раз выше, чем сегодня (несколько долларов на килобайт).

## 6.2 Ранние терминалы

Первые терминалы были кое-что вроде дистанционно управляемых пишущих машинок, которые могли бы только "отображать" (печатать на бумаге) символьный поток, посланный им из компьютера. Самые ранние модели назывались "Телетайпами". Ранние терминалы могли делать перевод строки и возврат каретки точно так же, как пишущая машинка и подавать гудок, когда приходил символ гудка. Из-за малой функциональности этот первый тип терминала, называется "dumb". Этот тип интерфейса терминала (использование типа терминала, называемого "dumb") иногда используется сегодня, когда компьютер не может выяснить с каким типом терминала он общается.

## 6.3 Управляющие коды и escape-последовательности

Терминалы имеют много возможностей, некоторые из которых доступны всегда, а некоторые из которых требуют активизации (или изменения) подачей команды от компьютера. Чтобы осуществить все эти возможности при управлении компьютером требуется назначить специальные коды так, чтобы компьютер мог сообщать терминалу, что надо сделать. Имеется два главных типа таких кодов: управляющие коды (управляющие символы) и escape-последовательности.

### Управляющие коды

Управляющие коды (или управляющие символы) состоят из первых 32 байтов алфавита ASCII. Они включают такие коды: возврат каретки (переместить курсор к левому краю экрана), перевод строки (переместить курсор вниз на одну строку), возврат на один символ, символ ESC, табуляция и звонок. Они обычно не показываются на экране. Обычно имеется команда, которую вы можете дать вашему терминалу, которая заставит отображать их при получении. Это называется чем-то вроде "Видимость управляющих символов". Если вы сделаете это, то на дисплей могут появиться беспорядочные символы, так как escape-последовательности, которые начинаются с управляющего символа escape, больше не выполняются. Слова, которые должны появиться наверху или внизу экрана, обнаружатся в других местах. Escape-последовательности говорят, что надо позиционировать курсор на экране, но курсор не двигается туда, куда указывает escape-последовательность.

### Escape-последовательности

Так как не имеется достаточного количества управляющих кодов, чтобы делать все, используется множество escape-последовательностей. Они состоят из управляющего символа "Escape" (ESC), сопровождаемого последовательностью обычных символов. После получения символа ESC, терминал исследует символы после него так, чтобы он мог интерпретировать последовательность и выполнить команду компьютера, назначенную этой последовательности. Когда он распознает конец последовательности, дальнейшие полученные символы, отображаются на экране (если они не являются управляющими кодами или большим количеством escape-последовательностей). Некоторые escape-последовательности могут иметь параметры

(или аргументы) типа координат на экране, в которые надо переместить курсор. Параметры являются частью escape-последовательности.

Список escape-последовательностей для вашего терминала должен быть в руководстве для терминала. Если бы терминал не очень старый, то может иметься две или три сотни таких последовательностей. Если у вас нет руководства, то найти его непросто. Для vt100, последовательности доступны в Интернет. При поиске в Интернет одной последовательности (типа ESC [5m) вы можете натолкнуться на длинный список их. Один из способов определить некоторых из них состоит в том, чтобы найти termcap (terminfo) для терминала и декодировать его. См. ``Terminfo и Termcap (подробно)'' в этом документе и/или ``Руководство по termcap'' в Интернет. К сожалению, termcap (terminfo) для терминала часто не перечисляет все escape-последовательности, которые доступны для использования в терминале, но частоупотребляемые, наиболее важные - обычно там есть.

## 7. Специальные возможности некоторых терминалов

### 7.1 Цвет

В то время как обычный одноцветный терминал - не цветной, он может иметь фиксированный "цвет" отображения, отличный от белого типа зеленого или янтарного.

Все терминалы имеют черный (электронный луч выключен = яркость нулевая).

Реальный цветной терминал может изменять цвет текста и фона на много других цветов, в то время как одноцветный терминал может только изменять яркость фиксированного цвета.

Однако, изменение яркости и т.д. дает много возможностей.

Например, черно-белый (одноцветный) терминал может иметь белый, серый и черный цвета, полученные изменением яркости. Некоторые слова могут быть черными на светлосером фоне, в то время как другие высвечены черным по белому.

Кроме того имеется белый на черном, подчеркивание и мигание.

Цвет действует подобно цвету на компьютерном мониторе или экране телевизора.

Электроннолучевая трубка имеет три цвета точек, причем каждый цвет управляется собственным электронным лучом (3 луча). Одноцветный имеет неотъемлемо лучшую разрешающую способность, так как он не зависит от точек, постоянно фиксируемых на экране. Для текстовых терминалов единственное использование цвета должно дифференцировать текст, и это преимущество всегда не стоит стоимости худшей разрешающей способности. Таким образом одноцветный может быть лучше, так как стоит он меньше.

### 7.2 Двойные сеансы

Для этого терминал имеет два последовательных порта. Каждый порт соединен с последовательным портом на различных компьютерах. Таким образом можно регистрироваться на двух различных компьютерах, причем каждый сеанс отображается в окне, занимающем полэкрана. В качестве альтернативы, каждый сеанс может выполняться полноэкранно с переключением между сеансами по "горячей" клавиши (или другим способом). Можно было также соединяться с двумя различными последовательными портами на том же самом компьютере и регистрироваться дважды (типа "виртуального терминала" в консоли).

## 7.3 Порты Printer/Auxiliary

Терминал имеет два последовательных порта, указанные выше, но эти два порта не идентичны в поведении. Если принтер соединен с принтером или дополнительным портом, то нажатие некоторых клавиш приведет к печати экрана. Можно также сделать так, чтобы все что отображается на экране шло также на принтер. Если порт - дополнительный порт, то можно соединять его с другим компьютером и иметь почти двойные сеансы как рассказывалось выше. Однако, видеопамять внутри терминала не может хранить оба сеанса, так что вы должны обновить экран при переключении на другой сеанс.

Это может быть необязательно "горячая клавиша", но возможно запрограммировать функциональную клавишу для этих действий.

Существуют различные комбинации клавиш и escape-последовательности для управления таким портом. Может быть трудно выяснить, как использовать этот порт без руководства терминала.

## 7.4 Символьные наборы

Символьный набор обычно представлен списком (или таблицей, или диаграммой) символов наряду с кодом байта, назначенным к каждому символу.

Код байта может принимать значения от 0 до 255 (от 00 до FF в шестнадцатеричной системе счисления).

В MC-ДОС, таблицы с символьным набором называются "кодowymi страницами". вы должны изучить такую таблицу, если не знакомы с ними. Они иногда включаются в руководства по принтеру и терминалу, но с их поиском могут возникнуть трудности.

ASCII - один из наиболее привычных символьных наборов, используемый в текстовых терминалах.

Это - 7-разрядный код, но может быть сделан 8-разрядным, если первый бит (бит старшего разряда) всегда устанавливать в 0. Другие символьные наборы обычно доступны (за исключением очень старых терминалов, где имеется только единственный набор ASCII). Первая половина большинства символьных наборов - это стандартные символы ASCII, а вторая половина (символы со старшим разрядом равным 1) принадлежат широкому разнообразию символьных наборов.

Символьные наборы - часто являются ISO стандартами. Чтобы получить на терминале не-ASCII символьный набор, вы вероятно должны загрузить программное обеспечение для этого символьного набора в память терминала.

Помимо ASCII, имеется несколько других общих символьных наборов, все - 8-разрядные.

CP означает наборы символов CodePage, изобретенные IBM: CP-437 (DOS ECS), CP-850 (Многоязычный Latin 1 --не то же, что и ISO Latin-1), ISO-8859-1 (Latin-1), ANSI (получен из Latin-1). MS Windows использует ANSI, в то время как Интернет часто использует Latin-1. Имеется несколько символьных наборов ISO-8859 в дополнение к Latin-1. Они включают Греческий (-7), Арабский язык (-6), Восточноевропейские (-2) и Русский (-5). Имеется много других. Например, KOI8-R более обычен для России, чем ISO-8859-5. Unicode - очень большой символьный набор, где каждый символ представляется 2 байтами вместо одного байта.

Вебсайты с символьными наборами:

- <http://www.cc.columbia.edu/kernit/charsets.html> для короткой распечатки различных названий символьных наборов.
- [http://www.pku.edu.cn/on\\_line/w3html/International/Overview.html](http://www.pku.edu.cn/on_line/w3html/International/Overview.html) для информации о символьных наборах и Интернете.
- <http://www.w3.org/International/O-charset-lang.html>
- <http://vancouver-webpages.com/multilingual/>

Если вы нашли заинтересовавшее вас имя символьного набора (или номер), то можете поискать большее количество информации о нем в Интернете.

## 7.5 Шрифты

Большинство терминалов, сделанных после середины 1980-х допускают загрузку программных шрифтов. Это означает, что они могут отображать почти любой символьный набор, если вы найдете программное обеспечение для этого. Если вы не можете найти необходимое программное обеспечение, то всегда можно создать ваше собственное. Редактор шрифтов для этой работы называется BitFontEdit (написан автором этого документа) и может быть найден на [http://www.cs.utk.edu/~shuford/terminal\\_index.html](http://www.cs.utk.edu/~shuford/terminal_index.html)

## 7.6 Клавиатуры и специальные клавиши

Клавиатуры терминала часто имеют ряд клавиш, которых нет на клавиатуре PC. Сейчас немногие терминалы имеют все эти клавиши, и на большинстве их есть дополнительные клавиши, не перечисленные здесь. Некоторые имеют большое количество специальных клавиш вроде терминалов, применяемых с кассовыми аппаратами. Клавиши часто имеют намного больше значений, чем показано здесь, так как клавиши часто изменяют свои значения, когда используются вместе с другими клавишами (типа shift и control).

- BREAK посылает 0 бит (+ 12 V) очень длинной продолжительности от 300 до 700 миллисекунд на главную ЭВМ. Главная ЭВМ может перейти к обработке запроса на прерывание, если stty установил brkint, или проигнорирует это, если ignbrk установлен.
- NO SCROLL останавливает прокрутку экрана, как при нажатиии ^S. Отпускание клавиши продолжит прокрутку. Для этого используются сигналы управления потоком данных.



- REPEAT, если нажата с другой клавишей, заставляет повторять вывод символа этой другой клавиши, даже если выключена опция автоповтора.
- LINE FEED посылает символ перевода строки ^J на главную ЭВМ. Используется редко.
- SET-UP позволяет настраивать терминал вручную через меню. Иногда преднамеренно блокируется специальным язычком под этой клавишей так, чтобы нельзя было нажать ее.
- LOCAL отсоединяет терминал от главной ЭВМ. В локальном режиме то, что печатается идет непосредственно на экран. Полезно для тестирования.
- RETURN - тоже, что клавиша "ввод" на PC. Обычно посылает символ возврата каретки на главную ЭВМ, который обычно транслируется в символ новой строки драйвером устройства главной ЭВМ. На некоторых терминалах он может быть установлен для отправки чего-то еще.
- F1, F2, ... или PF1, PF2, ... - функциональные клавиши, которые обычно могут программироваться, чтобы посылать последовательности байтов (символов).

## 7.7 Консоль Linux

Консоль для системы PC Linux - компьютерный монитор. Он подражает терминалу типа "Linux". Эмуляция гибкая и имеет больше возможностей, чем терминал vt100, который он эмулирует.

Они включают способность использовать заказные шрифты и легко менять раскладку клавиатуры (без того, чтобы исправить исходный текст, и перетранслировать ядро как требуется для случая реального терминала). Эти дополнительные возможности постоянно находятся в программном обеспечении драйвера консоли, а не в программном обеспечении эмулятора, но в результате они как бы часть эмуляции. Есть много команд (см. Keyboard-and-Console-HOWTO, прежде Keyboard-HOWTO), чтобы использовать эти добавленные возможности. Реальные терминалы, которые не используют ни скэн-коды, ни платы VGA, к сожалению, не могут использовать большинство этих возможностей. Можно перекомпилировать Linux, чтобы заставить терминал получать сообщения, которые обычно идут на консоль (см., ``Делаем терминал консолью").

### Делаем терминал консолью

Многие сообщения системы обычно посылаются на консоль.

В то же время, сообщения, появляющиеся при загрузке на консоли, затем можно просмотреть командой `dmesg`. Если начальная загрузка сбойт, то вы можете захотеть отображать на терминале сообщения, появляющиеся в течение загрузки.

Для этого можно отпатчить и перекомпилировать ядро Linux.

Оно пошлет терминалу сообщения, которые обычно посылаются только консоли. Это не превратит полностью терминал в консоль, так как не может ни менять раскладку клавиатуры, ни менять цвета подобно большинству консолей.

Linux Журнал в апреле 1997 выпустил статью по этому вопросу. Чтобы отпатчить ядро Linux добавьте несколько `#defines` в начало `src/linux/drivers/char/console.c`:

---

```
#define CONFIG_SERIAL_ECHO
#define SERIAL_ECHO_PORT 0x2f8 /* Serial port address */
```

---

Если вы в настоящее время используете консоль, чтобы выбрать, какую операционную систему загрузить (используя LILO), но хотели бы делать это с терминала, то вы должны добавить строку в `/etc/lilo.conf` файл. См. map для lilo.conf и поищите "serial=".

## 8. Примечания о фирменных знаках

Этот раздел может в конечном счете стать настолько большим, что станет отдельным документом (или отдельным файлом для каждой марки терминала). Он мог бы использоваться для многих случаев, когда одного руководства недостаточно для управления терминалом. вы могли бы помочь, описав некоторую информацию о терминале, который вы используете и отправив ее по почте ко мне. Иногда, полезная информация для некоторой модели может быть найдена в комментариях к записям terminfo (или termcap) в файле `/etc/termcap` или `terminfo.src`.

### 8.1 Терминалы CIT

CIT терминалы были сделаны в Японии в 1980-ых для CIE Terminals.

Они прекратили импортироваться в недавних 1980-ых. Компания, CIE, все еще делает принтеры CITON (в 1997), но не занимается и не поддерживает терминалами. Ernie в (714) 453-9555 в Irvine CA продает (в 1997) некоторые запчасти для моделей 224, 326, и т.д. но не имеет ничего для 80 и 101. (Документ, который вы теперь читаете, был написан в основном на 101e.)

Чтобы сохранить параметры Setup нажимают ^S в режиме Установки.

cit80: Контраст: кнопка на задней стороне терминала, cit101e: Яркость: используйте клавиши курсора вверх/вниз в режиме Установки.

### 8.2 Терминалы IBM

Не путайте IBM терминалы с IBM PC. Многие терминалы IBM не используют ASCII, а используют 8-разрядный код EBCDIC. В коде EBCDIC порядок передачи разрядов противоположен обычному, старший по порядку бит идет первым. Стандартная связь IBM - тип синхронной связи в блочном режиме (посылает большие пакеты символов). Используются два стандарта - "BISYNC" и "SNA" (который включает сетевые стандарты). Многие из их терминалов соединяются коаксиальным кабелем (RG62A/U), и наивные люди могут подумать, что разъем "BNC" на терминале - для локальной сети ethernet (но это не так).

В то время как эта система IBM фактически более эффективна чем то, что обычно используется в Linux, с этим стандартом IBM в настоящее время не будут работать с Linux. Однако, некоторые IBM терминалы - асинхронные ASCII терминалы и должны работать с Linux на PC. Могут работать модели 31xx за исключением ASCII терминалов 317x и 319x.

Перед приобретением IBM терминала, удостоверьтесь, что имеется termcap (terminfo) для них. Если их нет, вероятно терминалы не будут работать с Linux. Даже если имеется terminfo, они могут не работать. Например, имеется termcap для 327x, но 3270 - синхронный терминал EBCDIC.

Серия 3270 включает 3278 (в последних 1970-ых), 3279 с цветом и графикой, и контроллер терминала 3274 (похож на 3174).

Они могут использоваться, и для BISYNC, и для SNA. 3290 имеет деление экрана (разбиение на четверти).

Синхронные IBM терминалы не соединяются непосредственно с мейнфреймом IBM, а соединяются с "контроллером терминала" (иногда называемым "групповым контроллером" или "контроллером связи"). Некоторые из этих контроллеров могут преобразовывать синхронный сигнал в асинхронный так, чтобы в этом случае синхронный терминал мог бы косвенно соединяться с unix-подобным главным компьютером через последовательный порт. Но все еще имеется главная проблема - блочная передача. См. раздел "Блочный режим".

## 8.3 Телетайпы

Это антиквариат и представляет собой самые старые терминалы. Они подобны дистанционно управляемым пишущим машинкам, но большие и шумные. Сделанные в 1920-ых Teletype Corp. первые модели были сделаны на 30 лет раньше компьютеров. Ранние модели использовали электромеханические реле и вращающиеся дистрибьюторы вместо электроники. Их Baudot код имел только 5 бит на символ по сравнению с 7-разрядным ASCII. См. книгу "Руководство по малым компьютерным системам" Sol Libes, Hayden Books, 1978: pp. 138-141 ("Teletypes").

## 8.4 Терминалы VT (DEC)

Digital Equipment Corporation сделала известную серию терминалов VT, включая эмулируемый обычно VT100. В 1995 они продали свой терминальный бизнес фирме SunRiver, которая теперь называется Boundless Technologies. Подробная информация доступна на Shuford's Website.

Информация о текущих изделиях доступна на website Boundless Technologies. См. "Информация о терминалах в Интернет".

VT220: Некоторые имеют BNC соединитель для видео вывода (не для ввода).

## 9. Последовательная связь на EIA-232 (RS-232)

### 9.1 Введение

Текстовые терминалы на Unix-подобных системах (и на PC) обычно соединяются с асинхронными 232 последовательным портом компьютера. Это - обычно RS-232-C, EIA-232-D, или EIA-232-E. Эти три стандарта почти одно и то же. Первоначальный префикс RS стал EIA (Ассоциация Электронной Промышленности) и позже EIA/TIA после объединения EIA с TIA (Ассоциация Промышленности Передачи данных). EIA-232 спецификации используются также для синхронной (sync) связи, но аппаратные средства для поддержки sync на PC почти всегда отсутствуют. RS обозначение устаревает, но все еще используется. В этой статье будет использоваться обозначение EIA.

Старые терминалы используют 7-разрядный код ASCII для символов, но большинство терминалов, сделанных после 1990 позволяют иметь другие символьные наборы, которые используют 8-разрядные символы (256 возможных символов). Чтобы иметь дело с терминалами, вы должны также прочитать соответствующие части Serial-HOWTO и ознакомиться с таблицей кодов ASCII.

Последовательный порт - это больше, чем просто физический соединитель на конце компьютера или терминала. Он включает электронику, которая должна вырабатывать сигналы, согласовывающиеся со спецификацией EIA-232. Стандартный соединитель имеет 25 штырьков, большинство из которых не используются. Альтернативный соединитель имеет только 9 штырьков. Один штырек используется, чтобы посылать байты данных, а другой, чтобы получать байты данных. Другой штырек - общая земля. Другие "полезные" штырьки используются главным образом для передачи сигналов с устойчивым отрицательным значением напряжения "off" и устойчивым положительным значением напряжения "on".

Микросхема UART (Универсальный Асинхронный Приемопередатчик) проделывает большинство такой работы. Сегодня функциональные возможности этой микросхемы обычно встраиваются в другую микросхему.

## 9.2 Напряжения

### Напряжение для бита

В последовательном порту EIA-232, напряжения биполярны (положительны или отрицательны относительно земли) и по величине должны быть приблизительно 12 вольт (некоторые - 5 или 10 вольт). Для передающих и принимающих выводов +12 вольт - это 0 (иногда называемый "пробелом") и -12 вольт - это 1 (иногда называемая "меткой"). Это известно как инвертированная логика, обычно бит 0 является false и отрицательным, в то время как единица обычно true и положительной. Хотя получающий и передающий вывода - инвертированная логика, другие штырьки (линии управления модемом) - нормальная логика с положительным напряжением, являющимся true (или "on", или "установленными"), и отрицательным напряжением, являющимся false (или "off", или "инвертированными"). Нулевое напряжение не имеет никакого значения (за исключением того, что оно ошибочно).

Допускается отклонение напряжений. Спецификации говорят, что величина переданного сигнала должна быть между 5 и 15 вольтами, но никогда не должна превышать 25 V. Любых напряжения меньше 3 V - это неопределенный логический уровень (но некоторые терминалы принимают более низкое напряжение как допустимое). Иногда ошибочно говорят, что требуется напряжение 5 вольт (или даже 3 вольта), но обычно нужно 10-12 вольт. Если вы используете порт EIA-422 на компьютере Mac как EIA-232 (требует специального кабеля) или EIA-423, то фактически напряжение будет только 5 V. Обсуждаемый здесь интерфейс работает с 12 V. В Интернете много беспорядка в вопросе о напряжениях, и по крайней мере одна книга ошибочно утверждает, что положительное напряжение - это 1, не понимая, что EIA-232 использует инвертированную логику для передачи данных.

Обратите внимание, что нормальные компьютерные логические уровни обычно - 5 вольт (часто даже меньше), так что если вы пробуете использовать испытательное оборудование, разработанное для

тестирования компьютерных логических 5 вольт (ТТЛ-логика) на 12-вольтовых последовательных портах, то вы можете повредить испытательное оборудование.

### Последовательности напряжений для байта

На выводе передачи (© 2) во время паузы находится -12 V (метка). Чтобы начать байт, он переводится в +12 V (пробел) для обозначения стартового бита и остается в +12 V в течении времени передачи бита. Затем приходит бит младшего разряда байта данных. Если он равен 0, ничто не изменяется, и линия остается в +12 V до времени передачи следующего разряда. Затем приходит следующий бит, и т.д. В конце может быть послан бит контроля четности и затем появляется -12 V (метка) - это стоповый бит.

Линия остается в -12 V (пауза) до следующего стартового бита. Обратите внимание, что уровень 0 V отсутствует и таким образом нет простого способа (за исключением сигналов синхронизации) сообщить, где кончается один разряд и начинается следующий, если 2 последовательных бита имеют одинаковую полярность (оба нулевые или оба единички).

2-ой стоповый бит был бы -12 V, также как первый стоповый бит. Так как нет сигнала, который отмечал бы границы между этими битами, то единственный эффект, производимый 2-ым стоповым битом - тот, что линия должна оставаться в состоянии -12 V немного дольше. Приемник не может обнаружить различия между 2-ым стоповым битом и более длинным временем простоя между байтами. Таким образом связь работает хорошо, если один конец использует один стоповый бит, а другой конец использует 2 стоповых бита, но использование, только одного стопового бита очевидно быстрее. В редких случаях используются 1 1/2 стоповые биты. Это означает, что строка сохраняется в -12 V для 1 1/2 периодов времени (подобно стоповому биту, который в полтора раза шире обычного).

## 9.3 Контроль четности, объясненный

Символы обычно кодируются или 7, или 8 битами (данных).

Дополнительный бит контроля четности может (или не может) быть сопровождать байт символа, что дает в результате длину байта 7, 8 или 9 бит. Некоторые эмуляторы терминала и старшие терминалы не работают с 9 битами.

Контроль четности может быть установлен по нечетности, четности или не делаться вообще (контроль четности метки и пробела на некоторых терминалах невозможен). При проверке на нечетность бит контроля по четности выбирается так, чтобы число единичек в байте, включая бит контроля четности, было нечетно. Если такой байт портится, когда один из битов примет другое значение, то в результате проверки на четность будет определено, что байт испорчен. При обнаружении ошибки, если этот символ идет на терминал, на экране появится символ ошибки. Проверка на четность работает подобным способом со всеми допустимыми байтами (включая бит контроля по четности) создавая четное число единичек в байте. При установке, число битов на символ обычно означает только число информационных разрядов на байт (7 для истинного ASCII и 8 для различных наборов символов ISO).

"Метка" - это логическая 1, а "пробел" - это логический 0. При контроле четности метки бит четности всегда одnorазрядный. Для контроля четности пробела он - всегда нулевой. Контроль четности метки или пробела снижает полосу пропускания и нужно избегать его, насколько возможно. "Нет контроля по четности" не означает, что бит контроля четности не добавляется. Для терминалов, которые не работают с 9-битными байтами, должно быть выбрано "нет контроля четности" при использовании 8 разрядных символьных наборов, так как для бита контроля четности не хватает разрядов.

## 9.4 Формирование байта (кадрирование)

При последовательной передаче байтов через EIA-232 порты, сначала всегда посылается бит младшего разряда. Последовательные порты на РС используются для асинхронной связи, когда имеются стартовый бит и стоповый бит, отмечающие начало и конец байта. Это называется кадрированием, кадрируемый байт иногда называется кадром(фреймом). В результате общее количество посланных бит на один байт составляет 9, 10 или 11 битов, причем 10 бит - наиболее часто применяемое количество бит. 8-N-1 означает 8 информационных разрядов, нет контроля по четности, 1 стоповый бит. Это приводит к общему количеству 10 бит, если посчитать стартовый бит. Один стоповый бит используется почти всегда.

На скоростях 110 бит/сек (и иногда 300 бит/сек) использовались 2 стоповых бита, но сегодня 2-ой стоповый бит используется только в очень необычных ситуациях (или по ошибке, так как он по-видимому все еще хорошо работает).

## 9.5 Ограничения EIA-232

### Низкое быстродействие и короткое расстояние

Стандартный последовательный порт EIA-232 имеет низкое быстродействие и строго ограничен по расстоянию. Часто читают "высокую скорость", но он может работать с высокой скоростью только на очень коротких расстояниях типа модема, размещенного прямо рядом с компьютером. Все провода используют общую землю так, чтобы технология витой пары (необходимой для высоких скоростей) не может использоваться без дополнительных аппаратных средств. Однако некоторые компьютеры и терминалы имеют более современные интерфейсы. См. "Преимущества EIA-232".

Трагедия в том, что стандарт RS-232 1969 года не использовал технологию витой пары, которая могла бы функционировать в сотни раз быстрее. Витые пары использовались в телефонных кабелях с 1800-ых. В 1888 (более чем 110 лет назад) "Кабельная конференция" сообщила о поддержке витой пары (для телефонных систем) и подчеркнула ее преимущества. Но спустя более, чем 80 лет, RS-232 не сумел использовать это. Так как RS-232 был первоначально разработан для соединения терминала с низким модемным быстродействием, размещенным поблизости, то потребность в высокоскоростной связи и более длинном расстоянии передачи не была предусмотрена.

### Преимущества EIA-232

Для повышения скорости и увеличения расстояний был принят ряд стандартов EIA, использующих (сбалансированную) технологию витой пары. Сбалансированная передача может иногда быть в сотни раз быстрее, чем несбалансированный EIA-232. Для данного быстродействия расстояние (максимальная кабельная длина) с применением витой пары может быть во много раз больше. Но на РС стоит "устаревший" EIA-232, так как он хорошо работает с модемами, соединенными с медленными телефонными линиями, и он хорошо работает с мышами.

Одно исключение - компьютер Mac от Apple с EIA-232/EIA-422 GeoPort, который использует витую пару (сбалансированную) для передачи и приема. Он использует маленький круглый "мини-DIN" разъем. Он также обеспечивает стандартный EIA-232, но только на 5 вольтах (что допускается по спецификации EIA-232). Однако, вследствие того, что Macs стоят больше чем РС, они редко используются как главный компьютер для терминалов. Некоторые более новые терминалы используют EIA-423, но он точно такой же, как несбалансированный EIA-232, и может быть соединен с EIA-232 портом. Этот EIA-423 использует уровни 5 вольт, но по спецификации обеспечивает скорости выше, чем EIA-232 (который не будет иметь никакую справку на длинном выполнении, где это - неравновесие, которое вызывает вмешательство).

EIA-530-A (сбалансированный, но может также использоваться как несбалансированный) на 2Мбит/сек (сбалансированный) был предназначен для замены EIA-232, но таких интерфейсов было установлено немного. Он использует тот же самый разъем с 25 выводами как и EIA-232. Высокоскоростной Последовательный Интерфейс (HSSI = EIA-612/613) использует разъем с 50 выводами и имеет скорость до 50 Мбит/сек, но расстояние ограничено только несколькими метрами. Универсальная Последовательная Шина (USB) встраивается в PCI чипы. Это - 12 Мбит/сек по витой паре с разъемом с 4 выводами (2 провода - питание), но также ограничен короткими расстояниями.

## **Драйверы линии**

Для текстового терминала, скорости интерфейса EIA-232 вполне достаточно, но длина используемого кабеля зачастую слишком короткая. Сбалансированная технология могла бы это исправить. Общий метод получения сбалансированной связи с текстовым терминалом состоит в том, чтобы установить 2@ драйвера линии на последовательную линию, чтобы преобразовать несбалансированный в сбалансированный (и наоборот).

Они являются специальными продуктами и дороги, если приобретены новыми.

## **9.6 Синхронизация и синхронный**

### **Как синхронизируется "асинхронное" соединение**

В интерфейсе EIA-232 есть только два состояния передающего (или принимающего) провода: "метка" (-12 V) или "пробел" (+12 V). Состояния 0 V нет, и таким образом последовательность логических единиц передается только устойчивым -12 V без каких-либо маркеров между битами. Чтобы приемник обнаружил отдельные биты, он должен иметь тактовый сигнал, который синхронизирован с тактовым генератором передатчика. Такой тактовый генератор генерирует "тик" с передачей (или получением) каждого бита.

Для асинхронной передачи, синхронизация достигается введением для каждого байта стартового бита и стопового бита (выполненных аппаратными средствами). Приемник прослушивает линию, ожидая появления стартового бита, и когда обнаруживает его, запускает тактовый генератор. Он использует эти тактовые импульсы во время чтения следующих 7, 8 или 9 битов. (На самом деле этот процесс несколько более сложен, так как часто принимаются несколько выборок битов, и это требует дополнительных тактовых импульсов синхронизации.) Затем читается стоповый бит, тактовый генератор останавливается, и приемник ждет прибытия следующего стартового бита. Таким образом асинхронный обмен фактически синхронизирован в течение приема одного байта, но синхронизации между двумя соседними байтами нет.

### **Как определить - асинхронный обмен или синхронный**

Асинхронный (asunc) означает "не синхронный". Практически, асинхронный обмен - это когда асинхронный последовательный порт посылает и получает поток байтов, разграниченных стартовым и стоповым битами. Синхронный (Sync) - большинство остальных видов обмена. Но это не объясняет основные понятия.

Теоретически, синхронный обмен означает, что байты посланы с постоянной скоростью один за другим (с промежутками, равными одному такту тактового генератора).

Асинхронные байты могут быть посланы с различными интервалами времени между байтами (подобно тому, как поступают символы с клавиатуры).

Имеются пограничные ситуации, которые должны быть классифицированы или как синхронный, или как асинхронный. Асинхронный последовательный порт часто выпускает байты устойчивым потоком, который мог бы сойти за синхронный, но так как они все еще имеют старт/стоповые биты (из-за которых возможно посылать их с произвольными промежутками), то обмен называется асинхронным. Другой случай - когда байты данных (без старт/стопных битов) помещены в пакеты с возможным ошибочным интервалом между соседними пакетами. Это называется синхронным обменом, так как байты внутри каждого пакета должны быть переданы синхронно.

## **Синхронная связь**

Вы когда-либо задавались вопросом, для чего нужны все неиспользуемые выводы на разъеме последовательного порта с 25 выводами? Большинство их предназначено для использования в синхронной связи, которая редко применяется на PC. Имеются выводы для сигналов синхронизации также, как и для синхронного обратного канала. Спецификации EIA-232 предусматривают и `sync`, и `async`, но используемые в PC чипы UART (Универсальный Асинхронный Приемопередатчик) типа 16450, 16550A или 16650 не умеют работать с `sync`. Для `sync` требуется чип USART или эквивалентный ему, где "S" - означает "синхронный". С тех пор как `sync` занимает рыночную нишу, `sync` последовательный порт, вероятно, будет совершенно дорог.

Помимо `sync` части EIA-232, имеются другие другие синхронные стандарты EIA. Для EIA-232, 3 вывода разъема зарезервированы для тактовых импульсов (или синхронизация). Иногда в задачу модема входит генерация сигналов синхронизации, что делает невозможным использование синхронной связи без синхронного модема (или без устройства, называемого "синхронный модем eliminator", которое вырабатывает сигналы синхронизации).

Хотя немногие последовательные порты - `sync`, синхронная связь часто осуществляется по телефонным линиям с использованием модемов, которые используют коррекцию ошибок V.42. Он удаляет старт/стопные биты и помещает байты данных в пакеты, что приводит к синхронной передаче по телефонной линии.

## **9.7 Блочный режим**

### **Введение**

Блочный режим редко используется в Linux. В блочном режиме, когда кто-то печатает на терминале, результат сохраняется в памяти терминала и не посылается только главному компьютеру. Такие терминалы часто имеют встроенные возможности редактирования. Когда пользователь нажимает некоторые клавиши (типа посылающей клавиши), все, что было сохранено в памяти терминала, посылается главному компьютеру. Сейчас Linux редакторы `vi` и `emacs`, реагируют немедленно на нажатие некоторых клавиш, но в вышеупомянутой ситуации, при нажатии таких клавиш ничего не случится, так как ничто не посылается при нажатии клавиши. Таким образом использование блочного режима терминала не позволит использовать такие интерактивные программ. Старый интерфейс мейнфрейма IBM использует блочный режим, так многие терминалы IBM - работают только в блочном режиме и также синхронные (см. раздел ``Синхронизация и синхронный``).

### **Виды блочных режимов, формы**



Блочный режим может сам иметь различные подрежимы типа "страничного" (страница за раз) и "строчного" (строка за раз). Некоторые терминалы имеют, и блочные режимы передачи, и стандартные символьные режимы и могут переключаться с одного режима на другой. Асинхронные терминалы, которые имеют блочные режимы - это HP2622A, VT130, VT131, VT330, VT340 и Visual500.

Блочные режимы могут включать возможность обработки форм, когда хост посылает форму терминалу. Затем пользователь заполняет ее и нажимает посылающую клавишу, которая посылает только данные из формы обратно на хост. Форма непосредственно (не данные) отображаются на экране в защищенных полях, которые не передаются для главной ЭВМ.

## **Эффективность**

Блочный режим несильно загружает главный компьютер, особенно, если аппаратные средства главного компьютера разработаны для блочных режимов (как в было и есть в IBM). В символьном режиме каждый печатаемый символ посылается в последовательный порт немедленно и обычно вызывает прерывание на хосту. Хост, который получает байт, должна остановить свою работу и принять символ из аппаратного порта. Даже с UART, который имеет мощные аппаратные буферы задержка по времени может быть такой, что прерывание будет выдаваться для каждого печатаемого символа.

В истинном блочном режиме длинный блок символов принимается с использованием только одного прерывания. Если блочный режим используется со стандартными асинхронными последовательными портами, то более новые модели могут теперь получать за одно прерывание по 14 байтов, так как они имеют 16-байтовые аппаратные буфера. Таким образом при использовании блочного режима большинство лишних загрузок и непроизводительных затрат на обработку прерывания удалены, и компьютер может большее количество времени посвятить решению других задач.

Значительная экономия в блочном режиме происходит, если терминал соединен с хостом через сеть. Без блочного режима, каждый печатаемый символ (байт) представлен собственным пакетом, включая все байты заголовков (40 в TCP/IP пакете). При блочном режиме, большое количество символов представляется в виде одиночного пакета.

## **9.8 Книги по EIA-232 (RS-232)**

(Обратите внимание: В первой книге описано гораздо больше, чем только EIA-232.)

- Black, Uyless D.: Протоколы и интерфейсы физического уровня, IEEE Computer Society Press, Los Alamitos, CA, 1996.
- Campbell, Joe: Решение RS-232, 2nd ed., Sybex, 1982.
- Putnam, Байрон В.: RS-232 упрощенно, Prentice Зал, 1987.
- Seyer, Мартин Д.: Легкость использования RS-232, 2-ым редакция, Prentice, 1991

## **10. Управление потоком данных (подтверждение связи)**

Некоторые руководства терминала вызывают подтверждение связи "управления потоком данных". Управление потоком данных должно предотвратить слишком быстрое поступление потока байтов, чтобы не "переполнить" терминал, компьютер, модем или другое устройство. "Переполнение" - это когда устройство не может обрабатывать получаемую информацию достаточно быстро и таким образом теряет байты и/или делает другие серьезные ошибки. Управление потоком данных останавливает поток байтов до тех пор, пока терминал (например) не будет готов к приему следующих байтов. Управление потоком данных посылает сигнал остановки потока в направлении, противоположном тому потоку байтов, который надо остановить. Управление потоком данных должно быть настроено и в терминале, и в компьютере.

Имеются 2 типа управления потоком данных: аппаратное и программное (Xon/Xoff).

Аппаратное управление потоком данных использует специальный сигнальный провод, в то время как программное управление потоком данных сигнализирует, посылкой служебных байтов Xon или Xoff по обычному проводу данных. Для аппаратного управления потоком данных, кабель должен быть правильно распаян.

Поток байтов данных в кабеле между 2 последовательными портами двунаправлен так что имеются 2 различных потока:

1. Поток байтов от компьютера к терминалу
2. Поток байтов с клавиатуры терминала на компьютер.

## 10.1 Почему необходимо управлять потоком данных?

Вы могли бы спросить: "Почему не замедлить скорость передачи так, чтобы устройство успевало принимать информацию и таким образом избавиться от необходимости управлять потоком данных?". Это возможно, но обычно значительно медленнее, чем быстрая отправка и использование управления потоком данных. Одна из причин - это, что нельзя выбрать любую скорость передачи последовательного порта типа 14,500. Имеется только дискретное количество значений скоростей. Скорость должна выбираться немного выше, чем быстродействие устройства, но использование управления потоком данных заставит все работать правильно. Другая причина в том, что максимальная скорость, с которой устройство может работать (без управления потоком данных) часто зависит от того, что именно послано. Посылка escape-последовательностей на терминал, которые выполняют сложные вещи, обычно требует более медленной скорости в бодах. Для модема увеличение эффективности сжатия потока данных, посланных ему, зависит от того, насколько данные могут быть сжаты. Это случайная величина, так что для модемов также необходимо управление потоком данных.

## 10.2 Переполнение последовательного порта

Можно было удивиться, почему возможно переполнение последовательного порта, так как скорости и передачи, и приема байтов данных последовательных портов, установлены равными (в бит/сек) типа 19,200.

Причина в том, что хотя электроника приемника последовательного порта может работать со скоростью входящего потока, аппаратное/программное обеспечение, которые выбирают и обрабатывают байты из последовательного порта, иногда не могут справиться с высокой скоростью потока.

Одна из причин этого - буфер аппаратных средств последовательного порта очень мал. Старые последовательные порты имели размер аппаратного буфера только один байт (внутри микросхемы UART). Если этот один полученный байт данных в буфере не удален (извлечен) командами центрального процессора, то если прибывает следующий байт, этот байт теряется (буфер переполняется). Более новые микросхемы UART, а именно 16550A, имеют 16-байтовые буфера (но могут настроиться на эмуляцию однобайтного буфера) и их переполнение менее вероятно. Микросхемы могут быть настроены на генерацию прерывания, когда число байтов в буфере достигает 1, 4, 8 или 14 байтов. А другая компьютерная микросхема (обычно микросхема центрального процессора компьютера) извлекает входящие байты из этого маленького аппаратного буфера и обрабатывает их (также выполняя и другие задачи).

Когда содержимое этого маленького аппаратного буфера достигает определенного ограничения (один байт для старого UART'S) генерируется прерывание. Затем компьютерные прерывания, которые были вызваны и программное обеспечение выясняют что случилось. В конце концов они определяют, что требуется извлечь байт (или больше) из буфера последовательного порта. Они берут этот байт(ы) и помещают их в больший буфер (также буфер последовательных портов), который ядро поддерживает в оперативной памяти.

Терминалы также имеют последовательные порты и буфера подобно компьютеру.

Так как скорость потока байтов на терминал обычно намного больше, чем поток в обратном направлении с клавиатуры на главный компьютер, то терминал с большей вероятностью подвержен переполнению.

Конечно, если вы используете компьютер как терминал (эмуляцией), тогда он аналогично подвержен переполнению.

Опасные ситуации, когда переполнение наиболее вероятно: 1. Когда другой процесс отключил прерывания (для компьютера). 2. Когда буфер последовательных портов в главной (или терминальной) памяти собирается переполняться.

## 10.3 Посылка сигнала останова

Когда обнаруживается, что приемник почти переполнен входящими байтами, то отправителю посылается сигнал прекращения передачи. Это называется управлением потоком данных, и сигналы управления потоком данных всегда направлены против потока данных, которыми они управляют (хотя не в том же самом канале или проводе). Этот сигнал может быть или управляющим символом ( $^S = DC3 = Xoff$ ), посланный как обычный байт данных по проводу данных ("внутрипотоковая" сигнализация), или переходом напряжения с положительного на отрицательный уровень по rts-cts (или другим) сигнальным проводам (внепотоковая сигнализация).

Использование Xoff называется "программное управление потоком данных", а использование перехода напряжения в специальном сигнальном проводе (внутри кабеля) называется "аппаратное управление потоком данных".

Когда терминал просит остановить посылку данных, терминал "блокирует" клавиатуру. Это редко случается, но когда он это делает, сообщение, или индикатор должны сообщить вам, что клавиатура заблокирована. Все, что вы напечатаете на заблокированной клавиатуре, игнорируется. Термин "блокированный" также используется, когда компьютер просит прекратить передачу данных терминалу. Клавиатура не блокируется так что, все, что вы напечатаете, идет на компьютер. Так как компьютер не может послать что-нибудь обратно вам, символы, которые вы напечатаете, не отображаются на экране словно клавиатура заблокирована, но это не так.

## 10.4 Продолжение передачи

Когда приемник обработал данные и готов получать остальные байты данных, он сообщает об этом отправителю. Для программного управления потоком данных этим сигналом является управляющий символ  $^Q = DC1 = \text{Xon}$ , который пересылается как обычная строка данных. Для аппаратного управления потоком данных напряжение в сигнальном проводе переходит из отрицательного (инвертированного) уровня в положительный (установленному).

Если терминалу говорят, чтобы он продолжил отправку, клавиатура разблокируется, и она снова готова к использованию.

## 10.5 Аппаратное управление потоком данных (RTS/CTS и т.д.)

Некоторые старые терминалы не имеют аппаратного управления потоком данных, в то время как другие для этого используют широкий выбор различных выводов на последовательном порту. Наиболее популярный, кажется, вывод DTR.

### Управление потоком данных RTS/CTS и DTR

Linux PC использует RTS/CTS, но управление потоком данных с помощью DTR (используемое многими терминалами) ведет себя аналогично (за исключением то, что оно однонаправленное). RTS/CTS использует выводы RTS и CTS на последовательном разъеме (EIA-232). RTS означает "Запрос передачи (Request To Send)".

Когда на этих выводах появляется положительное напряжение в приемнике это означает: сохранение посланных ко мне данных. Если RTS инвертирован (напряжение отрицательное), то "Запрос передачи" обратный, что означает: не посылать мне данные" (прекратить посылку). Когда приемник готов опять принимать данные, он устанавливает сигнал RTS для другой стороны, чтобы она продолжила передачу. Для компьютеров и терминалов (оба - оборудование типа DTE) вывод RTS посылает сигнал управления потоком данных, а вывод CTS (Готов к передаче - Clear To Send) получает сигнал. То есть вывод RTS на одном конце кабеля соединен с выводом CTS на другом конце.

Для модема (DCE оборудование) - по другому; модемный вывод RTS получает сигнал, а вывод CTS - посылает. В то время как такая ситуация может казаться запутанной, для нее имеются веские исторические причины, которые также включены в данное обсуждение. Для DTR управления потоком данных в терминале DTR сигнал подобен сигналу, посланному из вывода RTS.

### Связь с помощью интерфейса DTR с RTS/CTS управлением потоком данных

Многие терминалы используют DTR управление потоком данных. Это исключительно одностороннее управление потоком данных для предохранения терминала от переполнения. Это не защищает компьютер от кого-то, печатающего слишком быстро для компьютера. Как можно использовать это с Linux, который использует управление потоком данных RTS/CTS?

Так как вывод DTR ведет себя подобно выводу RTS, то на терминале обрабатывают только вывод DTR, как будто это вывод RTS, присоединенный к выводу CTS на компьютере. Для этого вам вероятно потребуется сделать специальный кабель (или перепаять разъем). Таким образом можно использовать DTR управление

потоком данных на терминальном конце кабеля с RTS/CTS управлением потоком данных на компьютерном конце кабеля. Тогда при использовании этого вы должны "stty local" так как терминальный вывод DTR не может выполнить свою обычную функцию сообщения главному компьютеру, что терминал включен.

## **Отличие от старого подтверждения связи RTS/CTS**

При объяснении значений сигналов возникает путаница, из-за того, что имеется первоначальное значение RTS, которое противоположно вешеприведенному объяснению. Первоначальное его значение: я запрашиваю разрешение на посылку вам данных (Request To Send). Этот запрос был предназначен для посылки с терминала (или компьютера) на модем, который, если решит удовлетворить запрос, pošлет обратно установленный сигнал CTS с вывода CTS на вывод CTS компьютера: для посылки мне все чисто (Cleared to Send). Обратите внимание, что в отличие от современного RTS/CTS двунаправленного управления потоком данных, этот метод защищает поток только в одном направлении: от компьютера (или терминала) к модему.

Для старых терминалов, RTS может иметь это значение и установлен в высокий уровень, когда терминал имеет данные для передачи. Вышеупомянутое использование - форма управления потоком данных с тех пор, если модем хочет остановить передачу с компьютера, он сбрасывает CTS (соединенный с CTS в компьютере), и компьютер останавливает передачу.

## **Обратный канал**

Старые аппаратные терминалы могут иметь вывод обратного канала (типа вывода 19), который ведет себя подобно выводу RTS в RTS/CTS управлении потоком данных. Но этот вывод будет также инвертироваться, если бумага или лента выходит наружу. Часто можно соединить этот вывод с выводом CTS главного компьютера.

Может иметься dip-переключатель для установки полярности сигнала.

## **10.6 Аппаратное управление потоком данных выполнено аппаратными средствами?**

Некоторые думают, что аппаратное управление потоком данных выполнено аппаратными средствами, но (если вы не используете интеллектуальную последовательную плату с несколькими последовательными портами) это фактически выполнено программным обеспечением вашей операционной системы. Чипы UART и связанные аппаратные средства обычно не знают ничего вообще о аппаратном управлении потоком данных. Когда аппаратный сигнал управления потоком данных получен, сигнальный провод меняет полярность, и аппаратные средства дают электрический сигнал прерывания центральному процессору. Однако, аппаратные средства понятия не имеют, что это прерывание означает. Центральный процессор останавливает работу и переходит к таблице в оперативной памяти, которая сообщает центральному процессору, где находится программа, которая выяснит то, что случилось и предпримет соответствующие действия.

Это та программа (часть драйвера последовательного устройства), которая останавливает (или возобновляет) передачу. Эта программа проверяет содержание регистров в чипе UART, чтобы выяснить, какой из проводов изменил полярность.

Затем программное обеспечение понимает, что был получен сигнал управления потоком данных и останавливает (или запускает) поток. Однако, если был получен сигнал останова, поток останавливается

почти немедленно по прибытии сигнала, потому что прерывание останавливает любой процесс, выполняемый центральным процессором (включая программу, которая посылала данные и помещала ее в аппаратные буфера последовательных портов для передачи).

Однако те байты (до 16), которые были уже в аппаратном буфере последовательного порта будут переданы ??

Таким образом аппаратные средства почти немедленно останавливают поток только потому, что это - реакция на аппаратный сигнал, которая прерывает и останавливает все что делал центральный процессор.

## **10.7 Устаревшее управление потоком данных ETX/ACK**

Это тоже программное управление потоком данных и требует драйвера устройства, который знает об этом. Байты отправляются в пакетах (через async последовательный порт), каждый пакет завершается управляющим символом ETX (End of Text - конец текста) .

Когда терминал получает ETX, он ждет, сигнала готовности получить следующий пакет и тогда возвращает ACK (Acknowledge - подтверждение). Когда компьютер получает ACK, он посылает следующий пакет. И так далее. Это не поддерживается в Linux ??

## **11. Физическое соединение**

Терминал может быть соединен с главным компьютером или прямым кабельным соединением, через модем или терминальный сервер.

### **11.1 Прямое кабельное соединение.**

Самый простой и самый дешевый способ соединить терминал с главным компьютером - через прямое соединение с последовательным портом на компьютере. На большинстве PC установлено два последовательных порта, но один из них обычно занят мышью.

Для порта EIA-232 вам нужен нльмодемный кабель, с перекрестными передаваемыми и приемными проводами. Вывод 2 на одном конце соединен с выводом 3 на другом конце. Если вы хотите выключить аппаратное управление потоком данных, то вы должны выяснить, какие выводы терминала для этого используются, поскольку различные модели терминалов используют различные выводы. Диаграммы в Serial-HOWTO 7.1 не будут работать для аппаратного управления потоком данных, если вывод RTS терминала не используется для управления потоком данных. Это редко бывает ??

#### **Многопортовые платы ввода-вывода (адаптеры)**

Можно дополнительно приобрести последовательные платы, которые несут несколько последовательных портов, называемые "многопортовые платы" (см. Serial-HOWTO 2.3 и 2.4).

Одна из компаний, которая имеет цены ниже средних - ByteRunner <http://www.byterunner.com/cgi-bin/goto.cgi?file=iocards.html>. Для некоторых других марок см. Computer ESP [http://www.computer.esp.com/INPUT\\_DEVICES/PARALLEL\\_SERIAL\\_ADAPTERS/](http://www.computer.esp.com/INPUT_DEVICES/PARALLEL_SERIAL_ADAPTERS/)

## Ограничения по длине

Кабель более 50 футов или около того не может работать правильно с высокой скоростью. Намного более длинные длины иногда работают хорошо, особенно, если скорость низкая, и/или кабель - низкеемкостной специального типа и/или электроника приемника более чувствительна. При идеальных условиях на 9600 бод, 1000 футовый кабель работает хорошо. Один из способов покрывать длинные расстояния состоит в том, чтобы установить 2 драйвера линии возле каждого последовательного порта, чтобы преобразовать несбалансированный в сбалансированный (и наоборот) и затем использовать витую пару.

## Тип кабеля

Если вы собираетесь использовать аппаратное управление потоком данных (подтверждение связи), то вероятно вы будете должны соорудить ваш собственный кабель (или переделать готовый). вы должны будете определить, какие штырьки для чего использует терминал и спаять совместимый кабель. См. ``Аппаратное управление потоком данных''. Старые терминалы не могут иметь средств аппаратного управления потоком данных.

Если вы не используете аппаратное управление потоком данных, готовый кабель может работать хорошо (если вы сможете найти достаточно длинный). Нульмодемный кабель часто помечен как кабель принтера с "null modem" in fine print. В этих кабелях некоторые пары проводов "перекрещены" (= переключены, или пересечены, провода. Например, штырек 2 на одном конце идет к штырьку 3 на другом конце, но не все штырьки "перекрещены". Много готовых нульмодемных кабелей имеют перекрещенные RTS и CTS (штырьки 4 и 5), но терминал вероятно использует не эти штырьки для аппаратного управления потоком данных.

Обычный "неперекрестный" кабель не будет работать, если только вы не используете его как удлинитель или вместе с нульмодемным кабелем или нульмодемным адаптером (двусторонним соединителем), чтобы получить эффект пересечения. Удостоверьтесь, что соединители на кабельных концах совместимы с соединителями на аппаратуре. Большинство принтерных кабелей использует соединитель centronics на принтерном конце, который не будет соединяться с терминалами. Можно использовать телефонный витопарный кабель. Экранированный, специальный низкеемкостной кабель лучше всего подходит.

## A Kludge using Twisted-Pair Cable

Хотя ни один из сигналов EIA-232 не рассчитан на витую пару, можно попытаться использовать кабель витой пары. Используйте одну пару для передачи, а другую для приема. Сделать это можно, подсоединив "землю" к одному из проводов каждой из этих 2 пар. Only part of the signal ground current flows in the desired wire but it may help. Due to the lower inductance of the twisted pair circuit, more return (ground) current will confine itself to the desired path than one would expect from only resistance calculations. Это особенно относится к высоким частотам, так как индуктивное сопротивление растет с частотой. Прямоугольная волна на последовательном порту содержит высокочастотные гармоники.

## Заземление кабеля

Штырек 1 должен быть землей (и заземлен), но на дешевых последовательных портах, он может даже не быть соединен с чем-нибудь. Сигнальная земля - штырек 7, и обычно он заземлен на блок. Это означает, что часть сигнала несущей будет течь по проводам "земли" общего монтажа (нежелательно). Кабельные экраны, как предполагается, будут заземлены только на одном конце кабеля, но может быть лучше заземлить оба конца, так как лучше иметь ток в экране, чем на монтаже ??

## 11.2 Модемное соединение

При использовании комбинации модем-терминал (без компьютера) можно соединяться с BBS. Некоторые BBS (свободные сети) разрешают доступ к Интернет через текстовый браузер lynx, который может работать на текстовых терминалах.

Таким образом с помощью старого терминала и внешнего модема можно соединяться с Интернет. Если соединяться с главным компьютером, на котором имеется аккаунт, то можно иногда сохранять работу (или скачанную информацию) на главном компьютере. Конечно проще это делать на компьютере, эмулирующем терминал, используя программу связи типа minicom.

### Звонок с терминала

Вместо того, чтобы подключать терминал (или компьютер, эмулирующий терминал) непосредственно к главному компьютеру, используя кабель, это можно сделать через телефонную линию (или выделенную линию) с модемами на каждом конце провода. Терминал (или компьютер) будет обычно звонить по телефонной линии на главный компьютер.

Этот исходящий звонок обычно делается с помощью PC с модемом, чтобы позвонить на BBS или ISP, когда PC эмулирует терминал при установлении соединения. Если вы для этого используете реальный терминал, это более трудно, так как реальный терминал не очень интеллектуален и не дает такой функциональной обратной связи с пользователем. Для исходящего звонка, многие терминалы могут сохранять один или большее количество телефонных номеров как сообщения, которые могут быть "установлены" в них и посланы на модем нажатием некоторых функциональных клавиш.

Многие модемы также могут сохранять телефонные номера. Номеру телефона должна предшествовать последовательность команд инициализации. Главный компьютер может использовать "mgetty" (getty для использования с модемами) чтобы регистрировать в системе того, кто звонит на главную ЭВМ.

### Прием звонков на терминал

Менее обычная ситуация - главный компьютер звонит на терминал. Одна из возможных причин для этого состоит в том, чтобы сохранить на счетах телефона, где скорости - не симметричные. Чтобы осуществить это, модем на вашем терминале должен быть установлен на автоматический ответ (Если в регистре S0 записано число 2, то модем ответит на 2-й гудок). вы включаете терминал и модем прежде, чем, как вы ожидаете, произойдет обращение и когда обращение приходит, вы получаете подсказку входа в систему и регистрируетесь.

Главный компьютер, который звонит на ваш терминал, может это сделать, выполняя программу Linux "callback" (не очень широко распространенную), иногда называемую "cb". Повторный вызов - это когда на компьютер А звонит компьютер В, затем В бросает трубку и компьютер А звонит обратно. Это - то, что вам нужно, если вы для эмуляции терминала используете компьютер. В случае реального терминала это может быть слишком сложной задачей, так что главная ЭВМ может использовать только "обратную" часть



программы callback. Файл установки для callback должен быть правильно сконфигурирован на главной ЭВМ. callback звонит на терминал и затем mgetty выполняет вход в систему на этом порту. Mgetty сам по себе (с начала 1998 г.) служит только для входных звонков, но ведется работа по включению в него возможности повторного вызова и таким образом он сможет делать исходящие звонки. К тому времени как вы прочитаете это, mgetty может быть имеет все, в чем вы нуждаетесь.

## **Команды модема**

До июня 1997 года в Интернете не было никаких детальных списков команд для большинства модемов. Однако Hayes <http://www.hayes.com/support/techref/> опубликовал список команд и множество предложенных строк инициализации. Эти init строки - возможно не самые лучшие для терминалов, когда посылка информации от терминала идет очень медленно, со скоростью набора текста, но принимать сигналы требуется с максимально возможной скоростью ??

## **11.3 Соединение с терминальным сервером**

Терминальный сервер - нечто вроде интеллектуального переключателя, который может соединять несколько терминалов (или модемов) с одним или больше компьютером. Это не механический переключатель, так что он может изменять скорости и протоколы передачи данных, которые идут через него.

Он используется для того, чтобы подсоединить несколько терминалов (или модемов) к высокоскоростной сети, которая соединяется с главными компьютерами. Конечно сервер терминала должен иметь вычислительную мощность и программное обеспечение, чтобы работать с сетевыми протоколами, так что он немного похож на компьютер. терминальный сервер может взаимодействовать с пользователем и спрашивать, какой компьютер подсоединить, и т.д. Можно иногда посылать задания на принтер через терминальный сервер.

PC сегодня имеет достаточную вычислительную мощность, чтобы работать в качестве терминального сервера для текстовых терминалов за исключением того, что каждый последовательный порт должен иметь собственное аппаратное прерывание. PC для этой цели имеет только несколько запасных прерываний и так как они аппаратные, вы не можете увеличить их число с помощью программного обеспечения. Решение состоит в том, чтобы использовать многопортовую последовательную плату, которая имеет собственную систему прерываний.

Сегодня реальные терминальные серверы обслуживают не только терминалы. Они также обслуживают PC, которые эмулируют терминалы, и иногда соединяются с модемными пуллами, соединенными с телефонными линиями. Если терминал соединен непосредственно с модемом, модем на другом конце линии мог бы быть соединен с терминальным сервером. В некоторых случаях терминальный сервер по умолчанию ожидает, что абоненты используют PPP пакеты, нечто, что текстовые терминалы не посылают.

## **11.4 Создание или изменение кабеля**

### **Купить или сделать?**

Если вам нужны длинные кабели, чтобы соединить терминалы, как вы их получаете? Готовые длинные кабели вероятно трудно найти, особенно, если вы хотите использовать минимум (скажем) 4 проводника. Одна из возможностей получения - заказать их, что вероятно, будет довольно дорого, хотя вы могли бы

найти кого-то, кто делает их по ценам не слишком выше, чем готовый кабель (я их делал). Другой вариант - сделать ваш собственный кабель.

Могут потребоваться специальные инструменты. Если вы можете найти их, использовать кабель может быть выгоднее, но вам вероятно нужно будет перепаять разъемы.

Одно преимущество создания вашего собственного кабеля - то, что знания, которые вы приобретаете, пригодятся если кабель порвется (или испортится) или если вам надо срочно спаять другой кабель.

## **Разъемы и выводы**

К сожалению, большинство кабелей сегодня имеют литые соединители на каждом конце и не могут быть изменены. Другие имеют соединители, которые отвинчиваются и могут быть перепаяны. Если вы делаете свой кабель или изменяете существующий, тогда вы должны знать о выводах. Имеются два типа: паянные и обжатые.

Для обжатия выводов требуется специальный инструмент и кроме этого требуется инструмент для "вставки/вынимания". Но при помощи этих инструментов вы можете делать и изменять кабели гораздо быстрее, чем при помощи пайки. Чтобы вставить вывода, просто втолкните их с помощью инструмента. Удаление вывода с помощью инструмента требует некоторых хитростей. Вы должны поместить щуп инструмента позади вывода. One tool tip requires spreading a little to get it over the wire but for this tip the pin should be easier to remove. Затем мягким толчком сдвиньте провод и инструмент. Если не вышло, значит инструмент был неправильно вставлен или вы должны использовать другой щуп инструмента. Используя этот инструмент можно легко преобразовать обычный кабель в нуль-модемный, итд.

Не пробуйте паять, если не знаете, что вы делаете, или не прочитали, как это делается.

## **12. Установка компьютера и терминала**

### **12.1 Введение**

Часть настроек физического терминала сохраняется в энергонезависимой памяти и устанавливается при его включении. Кроме того, или как вариант, терминал можно конфигурировать посылкой escape-последовательностей на него из компьютера, но эти настройки обычно будут теряться при выключении терминала. Так что, если вы знаете, как устанавливать и сохранять конфигурацию внутри терминала, это вероятно самый лучший способ.

Другая часть установки позволяет компьютеру узнать какой терминал вы подсоединили к нему и как терминал был установлен. Переменная среды TERM содержит имя терминала и по этому имени программы ищут возможности терминала в каталоге terminfo. См. ``Terminfo и Termcap (коротко)".

Программы, называемые "stty" и "setserial", конфигурируют последовательный порт компьютера (включая программное обеспечение драйверов его устройств). Эти программы (если необходимо) должны выполняться при каждом запуске компьютера, так как настройки теряются при выключении компьютера.

Имеется очень много опций, которые вы можете выбрать. Некоторые опции касаются только терминала и не должны устанавливаться в компьютере.

Например: вы хотите видеть черные символы на светлом фоне? (Это легче для глаз, чем черный фон.) Клавиша должна повторять (при удержании ее нажатой) ввод символов? Экран должен переносить слова, когда строка доходит до правого края экрана?

Опции "интерфейса связи" должны быть установлены одинаково и на компьютере, и на терминале: скорость обмена, контроль по четности, биты/символ и управление потоком данных. Если они не совместимы, то удовлетворительная последовательная связь не будет установлена (или даже вообще связи между терминалом и компьютером не будет).

Нужно установить вручную на каждом терминале (или используя некоторый вид специальных кассет на каждом терминале). Главный компьютер конфигурируется добавлением команд в некоторые файлы на жестком диске, которые выполняются при каждом запуске компьютера (или когда люди входят в систему).

Хотя все это может показаться чрезмерно сложным, заставить терминал работать зачастую довольно просто. Раздел ``Быстрая установка" описывает простой путь настройки терминала. Но чтобы дисплей показывал картинку получше и был более производительным, может быть придется выполнить большее количество работы.

## 12.2 Установка терминалов

Как только коммуникационный интерфейс установлен, остальная часть настройки терминалов может иногда производиться отправкой команд на терминал с компьютера. Если у вас много терминалов, то вероятно лучше было бы написать (или поместить) скрипт shell, который будет делать это автоматически. Это может (или не может) быть команда, коорую вы можете посылать терминалу, чтоб он сохранил текущую установку в энергонезависимой памяти так, чтобы она восстановилась при следующем включении терминала.

Если у вас есть руководство по вашему терминалу, то вам нужно также проконсультироваться с ним.

Вы можете посылать команды настройки терминала, используя программы "tput" и "setterm". См. ``Изменение установок терминала". вы могли бы только посылать на терминал строку инициализации из записи terminfo, если хотите таким способом инициализировать терминал. Если вы не планируете посылать такие последовательности с компьютера на терминал каждым раз при включении терминала, то вы должны так или иначе сохранить установки в энергонезависимой памяти терминала.

Чтобы установить терминал, вы обычно должны ввести его в режим установки, нажав клавишу "установка" (или подобную). Появится меню, из которого вы можете выйти в другие меню. По мере перемещения по меню на дисплей могут (или не могут) выводиться другие меню. Посмотрите на клавиатуре метки над верхней строкой цифровых клавиш. Если они существуют, эти метки могут показывать, что эти клавиши делают в режиме установки. Старые терминалы могут иметь только одно меню "установки". Еще более старые имеют физические переключатели. В некоторых случаях не все переключатели хорошо помечены. Они могут даже быть DIP переключателями, скрытыми в разных местах безо всяких пометок.

Если ваш терминал настраивается не переключателями, то ваша установка должна быть сохранена в энергонезависимой памяти терминала так, чтобы она задействовалась при следующем включении терминала. Если вы не сможете сохранить ее, то новая установка будет потеряна при выключении терминала.

Прежде, чем вы перейдете к проблеме установки терминала, удостоверьтесь, что знаете, как сохранять установки. Для некоторых старых терминалах, только в руководстве написано, как их сохранить.

## 12.3 На компьютере

(Конечно вы можете делать это с любого рабочего терминала, соединенного с компьютером.)

### Inittab

Чтобы установить компьютер для обслуживания терминалов надо отредактировать различные файлы.

К счастью, на компьютере надо изменить только один файл: `/etc/inittab`. Добавьте новую команду(ы) `getty` в этот файл рядом с существующими командой(ами) `getty`. `getty` запустит процесс регистрации в системе на последовательном порту(ах) по вашему выбору и сообщит компьютеру какой тип терминала вы имеете через переменную среды `TERM` (см. `Serial-HOWTO 4.1 & 7.2` и/или напечатайте `"man getty"`).

Если вы не используете управляющие линии модема (например, если вы используете только минимальное число 3 проводников: передающий, приемный и землю), то вы должны сообщить об этом `getty`, используя флажок `"local"`.

Формат его зависит от типа `getty`, который вы используете.

### Stty и Setserial

Для установки последовательных портов имеется и команда `"stty"`, и команда `"setserial"`. Некоторые (или все) необходимые `stty` установки могут быть выполнены через `getty` и потребности использовать `setserial` может не возникнуть, так что вам эти команды могут и не понадобиться. Эти две команды (`stty` и `setserial`) устанавливают различные аспекты последовательного порта. `stty` делает больше, в то время как `setserial` работает на более низком уровне типа адресов порта и прерываний. Чтобы "сохранить" установки, эти команды должны написаны в некоторых файлах (скриптах `shell`), которые выполняются при каждом запуске компьютера. вы можете также использовать команды `stty` и `setserial` в командной строке, но такие установки будут потеряны при выключении компьютера.

### Setserial

`Setserial` может изменять и номер прерывания (`IRQ`), и адрес порта. Он может сообщить вам, какие у вас имеются чипы `UART`. Может установить продолжительность работы порта после его закрытия (для того чтобы вывести оставшиеся в его буферах символы в основное ОЗУ). Это необходимо на медленных скоростях 1200 или ниже. Чтобы `setserial` работал, должен быть загружен модуль `serial` (для ядра `Linux`). Если вы используете больше чем 2 последовательных порта, вы должны использовать `setserial`, чтобы назначить уникальный адрес `IRQ`.

### Где запускать setserial?

Чтобы изменить (или добавить) команду `setserial`, редактируют скрипт оболочки, который запускает `setserial`. Где он находится, зависит от вашего дистрибутива. Он может быть в `/etc/rc.d` в файле `"rc.local"` или `"rc.serial"`. В Debian дистрибутиве это был файл `0setserial` в `/etc/rc.boot`, но в конечном счете он должен измениться на файл в `/etc/init.d`.

## Stty

`Stty` производит большинство настроек последовательного порта. Чтобы посмотреть, как сейчас настроен терминала (или консоль), введите в командной строке: `stty -a`. Для других терминалов (типа `ttyS1`) напечатайте: `stty -a </dev/ttyS1`. Вот некоторые пункты настроек `stty`: скорость (в бит в сек), контроль по четности, биты/байт, число стоповых битов, обрезать 8-ой бит?, сигналы управления модемом, управление потоком данных, сигнал прерывания, маркеры конца строки, изменение регистра букв, дополнение, подача звукового сигнала при переполнении буфера, ЕСНО, позволить фоновым задачам писать на терминал?, определите специальные (управляющие) символы (например, по нажатию какой клавиши будет происходить прерывание). См. для подробностей руководство (напечатайте: `man stty`). Для использования некоторых специальных символов см. ``Специальные (управляющие) символы"

В некоторых реализациях `getty`, команды, которые обычно можно было бы давать от `stty`, помещаются в файл конфигурации `getty`: `/etc/gettydefs`. Даже без этого файла конфигурации, строки с командами `getty` может быть достаточно, чтобы настроить все, что нужно и не нуждаться в `stty`.

Можно написать программы C, которые изменяют конфигурацию `stty`, и т.д.

Просмотр соответствующей документации может быть полезен для лучшего понимания использования команды `stty` (и содержит список возможных параметров). Полезно почитать `Serial-Programming-HOWTO`. `man termios` содержит описание структуры языка C (типа `termios`), который сохраняет настройку `stty` в компьютерной памяти. Многие из названий флагов в этой структуре языка C - почти те же самые (и делают те же самые вещи), что и параметры команды `stty`.

## Где поместить команду stty?

Чтобы `stty` устанавливала терминал каждый раз при запуске компьютера, вы должны поместить команду `stty` в файл, который выполняется при каждом запуске компьютера (загрузке Linux). Этот файл должен быть выполнен перед запуском `getty`. Имеются много таких файлов. Если команда помещается в более, чем одно место, и вы знаете (или помните) только одно из тех мест, то вероятен конфликт. Так что обратитесь к документу (возможно руководству вашего терминала), где это написано.

Хорошее место для помещения этой команды - тот же самый файл, который запускает `setserial`, когда система загружается. См. ``Откуда запускать `setserial`?". Казалось бы самое лучшее поместить его после команды `setserial` так, чтобы операции низкого уровня выполнялись первыми. В дистрибутиве Debian имеется скрипт `/etc/init.d/bootmisc.sh`, но в настоящее время он выполняется прежде `0setserial`. Примерная строка: `stty crtscts < /dev/ttyS1`.

## terminfo и termcap (кратко)

См. ```terminfo` и `termcap` (подробно)" для более детального обсуждения `termcap`. Многие прикладные программы, которые вы выполняете используют базу данных `terminfo` (прежде `termcap`). Она имеет записи для каждой модели или типа (вроде `vt100`) терминала и сообщает, что терминал может делать, какие коды

посылать для выполнения различных действий, и какие коды послать терминалу, чтобы проинициализировать его.

Так как многие терминалы (и РС также) могут эмулировать другие терминалы и имеют различные "режимы" операций, то может иметься несколько записей `terminfo`, которые можно выбрать для данного физического терминала. Они обычно будут иметь похожие названия. Последний параметр `getty` должен быть именем терминала `terminfo` (или эмулятора терминала), который вы используете (такой как `vt100`).

`terminfo` - выполняет больше, чем только определяет то, что способен делать терминал и какие коды послать терминалу, чтобы заставить его что-то сделать. Он также определяет, на что будет походить "полужирный" текст, вид курсора, будут ли символы черными, белыми или некоторого другого цвета и т.д. В терминологии РС они называются "параметрами". Они также определяют коды инициализации, которые посылаются терминалу (аналогично `init` строкам модемов). Если вам не нравится как выглядит изображение на экране, вы можете отредактировать (и затем устанавливать) файл `terminfo` (или `termcap`).

## 12.4 Установка интерфейса связи

Эти установки:

- ``Скорость(бит/сек)``
- ``Контроль четности``
- ``Бит на символ``
- ``Управление потоком данных``

Вышеупомянутые установки должны быть установлены одинаково и на терминале, и на компьютере. На компьютере они часто могут устанавливаться, используя параметры для команды `getty` в файле `/etc/inittab`. Все из них могут быть установлены на компьютере командой `"stty"`, которую вы можете поместить в скрипт. Скрипт должен выполняться каждый раз при запуске компьютера. вы могли бы поместить команду `stty` в существующий скрипт, который выполняется во время загрузки. См. ``Где поместить команду stty?``

### Скорость

Она должна быть установлена одинаково и на терминале, и на компьютере. Скорость - биты в секунду (бит/сек или скорость в бодах). Скорость 9,600 - часто достаточно быстра, но если более быстрая скорость более эффективна и не приводит к ошибкам, используйте ее.

На терминале можно установить две скорости: передачи и приема, иногда обозначаемые T и R. Обычно они оба установлены одинаковыми.

Обычные скорости - 300, 600, 1200, 2400, 4800, 9600, 19200, ... Старые терминалы могут даже иметь быстроедействие 2, чтобы иметь совместимость с телетайпами 1920-ых, которые формировали последовательности байтов механическими средствами без электроники.

## Контроль четности

Для определения см. ``Контроль по четности, объясненный''. Контроль по четности часто запрещен по умолчанию. Чтобы разрешить контроль по четности, вы должны и разрешить его, и выбрать проверку на четность или на нечетность. Вероятно, неважно ккую проверку вы выберете. Для терминалов иногда настраивают проверку на четность и для передачи, и для получения. вы должны установить обе эти проверки одинаковыми, так как stty в компьютере не разрешает устанавливать их по-другому.

Должны ли вы использовать контроль по четности? Хотя в действительности контроль по четности не необходим, его неплохо бы включить. Если вы отключили контроль по четности, то вы можете получить неправильные символы в нескольких местах, и начнете исправлять орфографические ошибки, которых на самом деле нет. Однако контроль по четности требует затрат. Во-первых, его несколько сложнее установить, так как значения по умолчанию обычно отключают контроль по четности. Во-вторых, контроль по четности замедлит быстроедействие, с которым байты пересылаются по кабелю, так как на каждый байт добавляется еще один бит. Возможно, что эффективное быстроедействие замедлится.

Например, функции физических терминалов обычно ограничиваются механическим печатанием текста. Увеличение скорости передачи приводит лишь к появлению большего количества сигналов останова, чтобы уравнивать скорость передачи со скоростью механической печати. Из-за этого нет большой разницы в передаче с контролем по четности и без таковой. Такая ситуация типична для большинства терминалов.

Одна возможность состоит в установке терминала без контроля по четности. Далее, если замечаются ошибки контроля по четности, то контроль можно включить. Чтобы определить возможные ошибки при отсутствии контроля по четности, поищите любые орфографические ошибки, которых вы не делали. Если вы найдете такую ошибку, обновите экран (повторно передайте его из компьютера). Если ошибка исчезнет, то это вероятно ошибка контроля по четности. Если появляется слишком много таких ошибок (больше одной на каждую сотню экранов), тогда нужны корректирующие действия типа: разрешить контроль по четности и/или уменьшить быстроедействие и/или использовать кабель покороче/лучше.

Разрешение контроля по четности не уменьшит число ошибок, но сообщит вам, что ошибка произошла.

Противоположная стратегия состоит в первоначальном разрешении контроля по четности. Далее, если ошибок контроля по четности (ошибочные символы на экране) не замечено (за приемлемый период времени, скажем, за месяц или два), то контроль можно безопасно заблокировать.

## Биты/символ

Это - размер символа (число информационных разрядов на символ).

Для ASCII это - 7 бит, но ISO использует 8 битов. Если вы собираетесь использовать только символы ASCII, то выберите 7-бит, так как быстрее передать 7 битов, чем 8. Некоторые старые терминалы отображают только 7-разрядные символы.

## Какое управление потоком данных выбрать?

Выбор осуществляется между "аппаратным" (например rts/cts + dtr/dsr) и "программным" (Xon/Xoff) управлением потоком данных. Хотя аппаратное управление может быть быстрее (если в кабеле есть два

дополнительных провода для этого и если терминал поддерживает этот тип управления), в большинстве случаев Xon/Xoff тоже должен хорошо работать. Некоторые люди сообщают, что они решили какие-то проблемы, используя аппаратное управление потоком данных, но программное управление потоком данных прекрасно работает в других инсталляциях (и у меня лично).

Если вы используете программное (Xon/Xoff) управление потоком данных и имеете пользователей, которые не знают об этом, то они могут случайно послать Xoff главной ЭВМ и заблокировать свой терминал. Пока он заблокирован, они могут пытаться тщетно разблокировать его, нажимая клавиши. Затем, когда будет послан Xon для восстановления связи, все, что печаталось при этих попытках начинает выполняться, возможно с непредвиденными результатами. При аппаратном управлении потоком данных такого не происходит. См. "Управление потоком данных" для объяснения принципов управления потоком данных.

## 12.5 Редко требующийся файл /etc/ttytype

Файл конфигурации /etc/ttytype используется для отображения устройств /dev/ttySn на имена терминалов в terminfo. tset использует его, но если переменная среды TERM уже установлена правильно, то этот файл не нужен. Так как Linux getsy устанавливает TERM для каждого tty, то вы не нуждаетесь в этом файле. В других Unix-подобных системах типа FreeBSD, файл /etc/ttys отображает tty в несколько большее, типа соответствия команды getsy и категории соединения (такие как "dialup"). Пример строки Linux ttytype: vt220 ttyS1

## 12.6 Ограничение входа в систему

По умолчанию, root не может входить в систему с терминала. Чтобы разрешить ему это, вы должны создать (или отредактировать) файл /etc/securetty по man-странице "securetty". Чтобы ограничить вход в систему некоторым пользователям и/или некоторым терминалам и т.п. редактируют /etc/login.access (он заменяет старый /etc/usertty файл??). /etc/login.def определяет, должен ли использоваться /etc/securetty и его можно отредактировать так, что /etc/securetty станет ненужным (или не используемым). /etc/porttime задает временные промежутки, в которые некоторые tty и пользователи могут использовать компьютер. Если фиксируются слишком много неудачных попыток входа в систему каким-либо пользователем, то этому пользователю в следующий раз может быть запрещен вход в систему. На man-странице "faillog" изложена информация по управлению этим процессом.

## 12.7 Выполнять команду только в том случае, если TERM=my\_term\_type

Иногда команды нужно выполнять только для некоторых типов терминалов. Сделать это для команды stty несложно, так как она использует оператор переадресации потока <, чтобы определить, для которого терминала выдана команда. Но как быть с псевдонимами или функциями оболочки? вы можете захотеть сделать функцию для команды ls, так чтобы она выдавала распечатку каталогов в цвете только на цветных терминалах или консолях. Для одноцветных терминалов вы хотите использовать функцию с тем же самым именем (но с другим телом функции), которая использует символы как замену цветовым кодам. Где помещать такие определения функций, которые должны быть различными для различных терминалов?

Вы можете поместить их внутри конструкции "if" в файле /etc/profile, который выполняется каждый раз при входе пользователя в систему. Условное выражение "if" определяет некоторые функции и т.д. только в том случае, если терминал имеет определенный тип. Вот пример работы с dircolors :

```
if [ $TERM = linux ];
then eval `dircolors`;
elif [ $TERM = vt220 ];
then ls () { command ls -F $* } declare -xf ls
else echo "From /etc/profile: Unknown terminal type $TERM"
```



## 13. terminfo и termcap (подробно)

### 13.1 Введение

terminfo (прежде termcap) - база данных возможностей терминала и больше того. Для каждой (ну, почти) модели терминала он сообщает прикладным программам, что может выполнять терминал. Он сообщает то, какие escape-последовательности (или управляющие символы) посылать терминалу, чтобы делать вещи типа перемещения курсор в новое положение, стирания части экрана, пролистывания экрана, смены режима, изменения вида изображения (цветность, яркость, мигание, подчеркивание, негативное изображение и т.д.). После 1980 года многие терминалы стали поддерживать более сотни таких команд.

### 13.2 База данных terminfo

База данных terminfo компилируется и таким образом имеет исходную часть и скомпилированную часть. Старая база данных termcap имеет только исходную часть, но этот источник может быть и преобразован в источник terminfo, и затем скомпилирован одной командой. Чтобы увидеть, есть ли ваш терминал (скажем, vt100) в базы данных terminfo, напечатайте "locate vt100 ". Это команда может выдать /usr/lib/terminfo/v/vt100 или /usr/share/terminfo/v/vt100, которые являются возможными расположениями скомпилированных файлов terminfo. Некоторые старые программы могут ожидать, что они будут находиться в каталоге /etc/terminfo.

Исходный текст, который вы используете, может постоянно находиться в /etc/termcap и/или в terminfo.src. См. man-страницы: terminfo (5) или termcap (5), в которых описан формат, требуемый для создания (или изменения) этих файлов. Файл terminfo.src может быть расположен в других местах, или вовсе может не быть включен в ваш дистрибутив linux. Чтобы попробовать найти его, используйте команду locate. Можно скачать из Интернет (под именем termtypes.ti) из [locke.ccil.org/~esr/terminfo](http://locke.ccil.org/~esr/terminfo)

#### Компилятор terminfo(tic)

Данные в исходных файлах компилируются программой "tic", которая может преобразовывать формат termcap в/из формат terminfo.

Таким образом вы можете создавать скомпилированную базу данных terminfo из исходного текста termcap.

Программа установки, которая использовалась для установки Linux, вероятно установила скомпилированные файлы на ваш жесткий диск, так что вы не должны компилировать что-либо, если не изменяете /etc/termcap (или terminfo.src).

"tic" автоматически установит возникающие в результате компиляции файлы в каталог terminfo, готовые для использования прикладными программами.

#### Рассмотрим ваш terminfo

Это хорошая идея - рассмотреть вашу запись termcap/terminfo (исходный текст конечно) и прочитать комментарии. Быстрый способ посмотреть его без комментариев - только напечатайте "infocmp". Но комментарии могут сообщить вам нечто особое о терминале типа того, как вы должны установить его, чтобы он правильно работал с базой данных terminfo.

### Удаление ненужных данных

Чтобы сохранять дисковое пространство, можно удалить всю базу данных, кроме типов терминалов, которые у вас есть. Не удаляйте ничего из Linux termcap или xterm, если вы используете X-Window. Терминал типа "dumb" используется когда прикладная программа не может выяснять какой терминал вы используете. Если бы программы установки устанавливали terminfo только для терминалов, которые вы имеете, и если бы вы могли бы получать termcap для недавно установленного терминала за доли секунд из Интернет, то это сэкономило бы дисковое пространство вашего жесткого диска.

## 13.3 Инициализация

Часто в terminfo включены пара строк инициализации, которые могут быть посланы терминалу для его инициализации. Он могут изменять вид экрана, изменять режим, в котором работает терминал и/или заставить терминал эмулировать другой терминал. Строки инициализации автоматически не отправляются терминалу, чтобы проинициализировать его.

Чтобы сделать это, вы должны использовать команду, вводимую в командной строке (или в скрипте оболочки) типа: "tset", "tput init" или "setterm -initialize". Иногда ненужно посылать init-строку, так как терминал может устанавливать сам себя правильно, когда включается (используя опции/параметры, которые установлены и сохранены в энергонезависимой памяти терминала).

## 13.4 Переменная TERM

Переменная окружения TERM должна быть установлена по типу терминала, который вы используете. Обычно она устанавливается параметром terminal\_type, переданным программе getty (посмотрите это в файле /etc/inittab

Это имя должно быть в базе данных terminfo. Только напечатайте "set" в командной строке, чтобы видеть, какое значение присвоено TERM (или напечатайте: tset -q). В консоли (мониторе) TERM установлен в "linux", который является монитором PC, подражающим модели несуществующего терминала по имени "linux". Так как "linux" близок к терминалу vt100, а также ко многим другим текстовым терминалам, то обозначение "linux" будет иногда работать как временное с текстовым терминалом.

Если на один и тот же порт могут быть подсоединено более одного типа терминала (/dev/tty ...) (например, если имеется переключатель, разрешающий использовать различные типы терминалов на одном и том же самом последовательном порту, или если порт соединен с модемом, на который звонят люди с различных типов терминалов), тогда TERM нужно устанавливать каждый раз, когда кто-то соединяется с последовательным портом. Часто имеется escape-последовательность запроса такая, чтобы компьютер мог запрашивать у терминала его тип. Другой способ состоит в том, чтобы попросить пользователя напечатать (или выбрать) тип своего терминала.

Возможно для этого действия вам понадобится использовать tset в коротком скрипте shell.

Один способ сделать это состоит в том, чтобы использовать "tset" (см. man). Tset пробует определить имя терминала, который вы используете. Затем он ищет данные в terminfo и посылает вашему терминалу init строку.

Также можно устанавливать значение TERM. Например, пользователь звонит и регистрируется в системе. Выполняется скрипт регистрации в системе .profile, который содержит внутри себя следующее выражение: eval `tset -s? Vt100`. Это приводит к следующему: пользователя спрашивают, использует ли он(а) vt100. Пользователь или отвечает да или печатает имя фактически используемого терминала. Затем tset посылает init строку и устанавливает TERM в это имя (тип) терминала.

## 13.5 Termcap документы

- The Termcap Manual [http://www.delorie.com/gnu/docs/termcap/termcap\\_toc.html](http://www.delorie.com/gnu/docs/termcap/termcap_toc.html) от Richard M. Stallman -- это руководство GNU.
- man terminfo (5) и/или termcap (5).
- /etc/termcap (файл) или terminfo.src содержит информацию о различных версиях termcap файлов, именные соглашения для терминалов и коды специальных возможностей, называемые u6-u9
- "Termcap and Terminfo " -- книга, изданная O'Reilly в 1988.

## 14. Использование терминала

### 14.1 Введение

Это раздел об управлении интерфейсом компьютер-терминал и/или изменении установок терминалов при использовании терминала. Он объясняет (или указывает на объяснения) как пользователь терминала может управлять и просматривать интерфейс и как использовать различные команды, предлагаемые драйвером устройства. Он не объясняет, как использовать многие прикладные программы, оболочки или большинство Linux утилит. Две команды, обычно используемые в терминале:

- clear (чтобы очистить экран)
- reset (чтобы сбросить терминал)

### 14.2 Запуск терминала

Вы включаете питание терминала, нажимаете клавишу "возврат" (или "ввод") и затем получаете подсказку входа в систему. Затем в ответ на запрос печатаете ваше учетное имя и ваш пароль. Если ничего не происходит, удостоверьтесь, что хост работает правильно. Если хост выключен, тогда то, что вы печатаете на клавиатуре терминала может появляться на экране, так как и передающие, и принимающие штырьки на

хосте могут быть соединены вместе, приводя к отображению на экране символов "от" компьютера. Если вы не можете зарегистрироваться, хотя хост работает, см. ``Поиск неисправностей``.

## 14.3 Драйвер устройства терминала

При печати в командной строке, оболочка (типа `bash`) читает то, что вы напечатаете и реагирует на это. То, что вы печатаете сначала проходит через драйвер терминала вашей операционной системы. Этот драйвер может транслировать некоторые символы (типа изменения символа "возврат", сгенерированного клавишей "возврат" в символ "новой строки"). Он также распознает некоторые управляющие коды, которые вы можете напечатать в клавиатуре типа `^C`, чтобы прервать выполнение программы. Он также обычно отображает то, что вы напечатаете обратно на вашем дисплее.

Для настройки драйвера можно использовать ```stty```.

## 14.4 Испорченный интерфейс терминала

### Признаки

Когда изображение на дисплее выглядит неправильным или то, что вы печатаете, не отображается правильно, у вас вероятно испортился интерфейс терминала. Это может также быть ошибкой в программе, которую вы используете, аппаратный отказ или, возможно, неправильная настройка. Если все работало нормально, но внезапно произошел отказ, то может быть интерфейс испортился в результате ваших каких-либо действий. Интерфейс может быть испорчен двумя способами: посылкой двоичного файла на терминал и неправильным выходом из прикладной программы.

### На терминал посланы двоичные символы

Ваш терминал изменит свои характеристики, если на него послать некоторые escape- последовательности или управляющие символы. Если вы по неосторожности пробуете отобразить двоичный файл, то он может содержать такие последовательности, которые могут перевести ваш терминал в некоторый странный режим работы или даже совсем испортить его.

Чтобы этого не произошло, всегда просматривайте или редактируйте двоичный файл программами, разработанными специально для этой цели. Большинство редакторов и просмотрщиков нормально обрабатывает двоичный файл, так что интерфейс не разрушается. Некоторые из них могут отображать сообщение, говорящее вам, что они не могут редактировать двоичный файл. Но использование `"cat ...."` или `"cp .... /dev/tty .."` где .... - двоичный файл, пошлет двоичный файл на терминал.

Порча может также случиться при использовании программы связи, когда удаленный компьютер может послать двоичный файл на ваш экран. Имеется множество других способов, как это может случаться. Даже предполагаемый ASCII файл может содержать нежелательные управляющие коды.

Чтобы поправить эту проблему терминал сбрасывают. Может иметься клавиша, которая делает это непосредственно на терминале (возможно в режиме установки), или вы можете попробовать напечатать или `"reset"`, или `"setterm -reset"` (хотя вы не можете видеть то, что вы печатаете). Это действие пошлет строку `reset` из записи `terminfo` на терминал.

## Неправильный выход из программы

Большие прикладные программы (типа редакторов) часто используют команду `stty` (или подобную) в своем коде, чтобы временно изменить конфигурацию `stty`, когда вы работаете с программой. Они могут переводить драйвер устройства в "пропускающий" режим так, чтобы каждый символ, который вы печатаете шел непосредственно в прикладную программу. Отображение драйвером блокируется, так что все, что вы видите на экране, пришло непосредственно из прикладной программы. Таким образом многие команды управления (типа `^C`) не могут работать внутри таких приложений.

Когда вы приказываете такому приложению завершиться, прикладная программа первым делом восстанавливает установки `stty` на те, что были перед началом работы прикладной программы. Если вы неправильно вышли из программы (может случиться так, что, то что вы печатаете больше не отображается на экране), тогда вы можете оказаться в командной строке будучи в "прозрачном режиме".

Выйти из прозрачного режима и восстановить нормальный тип установок `stty` можно напечатав "`stty sane`". Однако, вы должны напечатать это только после нажатия клавиши "возврат" и закончить это нажатием "возврат". Но нажатие клавиши "возврат" не оказывает действия, так как код "возврат" больше не транслируется в символ новой строки, чего ждет оболочка. Поэтому вместо "возврат" вводите `^J` (символ новой строки). "sane" интерфейс терминала может не быть таким же обычным, но обычно это работает. "stty sane" может быть также полезен для выхода из интерфейса, разрушенного по другим причинам.

## 14.5 Специальные (управляющие) символы

Число управляющих символов, которые вы можете напечатать на клавиатуре, "захватываются" драйвером терминала и выполняют различные задачи. Чтобы увидеть эти команды управления, напечатайте: `stty -a` и посмотрите строки 2-4. Они - разъяснены в `man stty`. Они могут быть изменены на различные управляющие символы или заблокированы, используя `stty`. Таким образом ваши управляющие символы могут быть отличны от тех, что описаны ниже.

Они используются для редактирования командной строки, прерывания, прокрутки, и для прозрачной передачи следующего символа.

### Редактирование командной строки

В то время как драйвер терминала имеет несколько команд для редактирования командной строки, некоторые оболочки имеют встроенный реальный редактор (типа "readline" в оболочке `bash`). Такой редактор - обычно включен по умолчанию, так что вам не надо делать что-либо, чтобы включить его. Если он доступен, то вы не нужно учить множество следующих команд, хотя они часто все еще работают наряду с редактором командной строки. Наиболее важные из них - это `^C` (прерывание), `^D` и прокрутка.

- Delete-клавиша (`stty` показывает ее как `^?`) стирает последний символ
- `^U` удаляет строку
- `^W` удаляет слово назад

- ^R перепечатывает строку. Полезно главным образом на устройствах выдачи твердых копий??

## Прерывание (и выход, приостановка, конец файла, Flush)

- ^C прерывание. Выходит из программы и возвращает Вас к подсказке командной строки.
- ^/ выход. То же самое как прерывание ^C, но более слабое. Также формирует дамп "основного" файла (который вам вероятно не нужен) в вашем рабочем каталоге.
- ^Z приостановка. Останавливает программу и переводит ее в фон. Напечатайте fg, чтобы перезапустить ее.
- ^D конец файла. Если печатается в подсказке командной строки, выходит из оболочки и переходит в состояние, в котором вы находились перед запуском оболочки.
- ^O flush. Не реализовано в Linux. Посылает вывод в /dev/null.

## Прокрутка

Если, что вы хотите остановить прокрутку нижней части экрана, то вы можете послать сигнал "останова" (^S или Xoff) на хост (предполагается, что управление потоком данных" Xon-Xoff включено). Посылка "сигнала продолжения" (^Q или Xon) возобновляет прокрутку. Некоторые терминалы имеют клавишу "No Scroll", которая поочередно посылает Xoff и Xon. Если у вас аппаратное управление потоком данных

- ^S остановить прокрутку (Xoff)
- ^Q продолжить вывод на экран (Xon)

Если вы хотите и остановить прокрутку, и выйти, используйте ^C. Если вы хотите остановить прокрутку, чтобы делать кое-что еще, но хотеть сохранить программу в памяти, так чтобы вы могли вернуться к ней попозже, то используйте ^Z для приостановки.

Альтернативный метод прокрутки - конвейерный вывод через программы pager, такие как more, less или most. Однако, вывод может производиться не только через поток стандартного вывода, но и через поток вывода ошибок, который pager не распознает. Чтобы исправить это, вы можете использовать переадресацию "2>&1", тогда pager заработает правильно. Зачастую проще использовать ^S и ^Q, если вам не требуется листать в обратном направлении.

В консоли PC (эмуляция терминала) вы можете листать в обратном направлении, используя Shift-PageUp. Это часто необходимо, так как прокрутка идет слишком быстро, чтобы использовать ^S. Если вы листали в обратном направлении, то Shift- PageDown будет листать снова вперед.

## Прозрачная передача следующего символа

^V посылает следующий печатаемый символ (обычно управляющий символ) через драйвер устройства непосредственно, т.е. без какого-либо действия или интерпретации.

На экране отображаются два символа ASCII типа ^C.

## 14.6 Проверка интерфейса

Эти утилиты будут выдавать информацию о интерфейсе терминала:

- `gitkeys`: показывает, какой байт(ы) каждая клавиша посылает главной ЭВМ.
- `tty`: показывает, с каким портом `tty` вы соединены.
- `set`: показывает значение переменной `TERM` (имя записи `terminfo`)
- `stty -a`: показывает все установки `stty`.
- `setserial -g /dev/tty??` (подставьте нужное число вместо `??`) показывает тип UART, адрес порта и номер IRQ.
- `infocmp`: показывает текущую запись `terminfo` (меньшее количество комментариев)

## 14.7 Изменение установок терминала

Установки терминала обычно устанавливаются один раз, когда терминал устанавливается, используя процедуры установки в руководстве терминала. Однако, некоторые установки могут быть изменены, когда терминал используется. обычно вы не даёте команд `"stty"` `"setserial"`, когда терминал используется, поскольку они, вероятно, разрушат интерфейс терминала. Однако, есть установки, которые вы можете производить для изменения вида экрана терминала, или изменения его поведения без того, чтобы нарушить целостность интерфейса. Иногда эти изменения делаются автоматически прикладными программами, так что вам возможно не нужно иметь дело с ними.

Один прямой метод создания таких изменений состоит в том, чтобы использовать клавишу установки в терминале и затем использовать меню или что-то подобное, чтобы сделать изменения. Для такой операции вам надо знать терминал.

Другие 3 метода, чтобы делать изменения, посылают escape-последовательность с компьютера на терминал. Эти 3 примера демонстрируют различные методы установки негативного изображения:

```
setterm -reverse
tput -rev
echo ^[[7m
```

**setterm**

Это - самая простая в использовании команда. Она использует длинные опции (но не использует перед ними --) и консультируется с базой данных terminfo, чтобы определить, какой код послать. вы можете изменять цвет, яркость, перенос строк, повторение клавиатуры, вид курсора, и т.д.

## **trut**

Команда "trut" подобна "setterm", но вместо того, чтобы использовать обычные слова как параметры, вы должны использовать сокращения, используемые terminfo. Многие из сокращений очень краткие и трудны для запоминания.

## **echo**

В примере "echo ^[[7m", который устанавливает негативное изображение, ^[ - символ ESC. Чтобы ввести его, напечатайте ^V^ [ (или ^V, сопровождаемый клавишей escape).

Чтобы использовать этот метод "echo", вы должны выяснить какой код использовать из вашего руководства терминала, или из terminfo или termcap. Проще использовать setterm или trut, хотя echo выполнится немного быстрее. Так что вы могли бы использовать echo ... в скриптах шелл, которые имеют дело только с одним типом терминала.

## **Сохранение изменений**

Когда вы выключаете терминал, сделанные вами изменения теряются (если вы не сохранили их в энергонезависимой памяти терминала, нажимая некие клавиши на терминале). Если вы хотите использовать их снова, без того, чтобы заново вводить, поместите команды в скрипт, или сделайте это функцией шелл. Затем выполните его, когда хотите сделать изменения. Один способ сделать изменения полупостоянными состоит в том, чтобы поместить команды в файл, который выполняет каждый раз при входе в систему или запуске компьютера.

## **15. Поиск неисправностей (программное обеспечение)**

Если вы предполагаете, что проблема - аппаратная, см. раздел "Ремонт и диагностика".

Вот список возможных проблем:

- ``Терминал работает хорошо?" Предположим, что терминал нет.
- ``Отсутствует текст" Или пропущен текст, или текст отображается правильно и терминал зависает
- ``Getty Resawning Too Rapidly" (console error message)
- ``Getty Перепорождение Слишком Быстро " (пультовое сообщение об ошибках)



- ``Сбои после входа в систему"
- ``Нельзя войти в систему, но приглашение входа в систему есть.
- ``Искаженное приглашение входа в систему"
- ``Нет приглашения входа в систему"

Имеются два случая, когда терминал ломается. Один - он только что работал хорошо и внезапно перестал работать. Это обсуждено в следующем разделе. Другой случай - неправильная работа терминала, когда вы его устанавливаете. Этот случай описан через раздел.

## 15.1 Терминал работал хорошо

Когда прежде рабочий терминал внезапно ломается зачастую поиск проблемы прост. Проблема может быть очевидной типа сообщения об ошибках, когда терминал включен первый раз. Если он шумит, вероятно требуется ремонт. См. ``Ремонт и Диагностика". Сначала, подумайте от том, что было сделано или изменено недавно, поскольку вероятно это и есть причина проблемы.

Проблема случалась только после того, как было установлено новое программное обеспечение системы или после изменения в конфигурации?

## 15.2 Терминал недавно установлен

Если вы только что соединили терминал с компьютером по инструкции, и он не работает, то этот раздел - для вас. Если терминал, который прежде работал хорошо, не работает теперь, тогда см. ``Терминал работал хорошо".

Если вы предполагаете, что последовательный порт на вашем компьютере может быть испорчен, то вы можете попробовать выполнить диагностическую программу для него, возможно из MS DOS.

Один из подходов, чтобы сначала увидеть, будет ли терминал работать, состоит в попытке скопировать файл на терминал (ср `my_file /dev/ttyS?`) в наиболее простой ситуации. Это значит с заблокированными линиями управления модема и на такой скорости, когда не нужно управление потоком данных (удостоверитесь, что любое аппаратное управление потоком данных заблокировано). Если копирование было успешным, то усложните ситуацию, и см., работает ли это все еще и т.д., и т.д. Когда проблема появляется только после того, как вы сделали изменение, то вероятно, что это изменение и есть источник проблемы. Фактически, более эффективный (но более сложный) метод состоит в переходе от простой ситуации к приблизительно половине конечной конфигурации так, чтобы тест устранил приблизительно половину остающихся возможных причин проблемы. Затем повторите эту методологию для следующего теста. При этом способе достаточно сделать с десятков тестов, чтобы найти причину из тысячи возможных причин. вы должны отклониться немного из этого метода, основанного на догадках и сведениях.

## 15.3 Терминал работает хорошо?

Исправный терминал обычно будет при запуске выдавать слова на экране.

Если эти слова не передают сообщение об ошибках, он видимо исправен. Если не видно признаков подачи питания (не светятся никакие лампочки и т.д.) заново подключите оба конца шнура питания компьютера. Удостоверьтесь, что имеется ток в стенной розетке (или на удлинителе). Попробуйте другой шнур питания, если он у вас есть.

Удостоверьтесь, что терминал включен и что его плавкий предохранитель не перегорел.

Пустой (или тусклый) экран можно иногда исправить использованием кнопок управления яркостью и контрастностью, или нажатием клавиш в режиме установки. Переведите терминал в автономный режим, если это возможно, и попробуйте печатать. Если он все еще не будет работать, см. "Ремонт и диагностика" для советов по починке терминала.

## 15.4 Отсутствует текст

Если некоторый текст отображается на терминале хорошо, а затем он останавливается незаконченным (на середине слова и т.д.), или если куски текста отсутствуют, то вы вероятно имеете проблему с управлением потоком данных. Если вы не можете выяснить сразу же, чем это вызвано, уменьшите скорость. Если это помогло, то вероятно проблема с управлением потоком данных. Может быть, что управление потоком данных не работает полностью из-за ошибок настройки, или из-за неправильной распайки кабеля (для аппаратного потока управления). См. "Управление потоком данных"

Если одиночные символы отсутствуют, возможно последовательный порт - переполнен слишком быстрым потоком данных. Попробуйте более медленную скорость.

Если вы используете скорость 1200 (очень медленно, обычно используемый для старых терминалов и принтеров) и текст остается усеченным, то проблема вероятно в драйвере последовательного устройства. См. Printing HOWTO "Последовательные устройства" для информации о том, как это устанавливается.

## 15.5 getty восстанавливается слишком быстро

### Нет напряжений управления модемом

Если getty не может открыть и/или использовать порт из-за недостатка положительного напряжения управления модемом на одном из выводов, то getty может уничтожиться. Затем командой в inittab, getty повторно порождается и пробует снова, уничтожается снова и т.д., и т.д. Пробуйте использовать опцию "local" с getty и/или проверить установки управления модема и напряжения.

### Клавиша закорочена

Другая возможная причина перепорождения getty - клавиша клавиатуры закорочена, давая тот же самый результат, словно клавиша непрерывно нажата. Если разрешено автоповторение, то в подсказке входа в систему "печатаются" тысячи символов. Экран заполняется одинаковыми символами (в некоторых случаях 2 или более различными символами).

## 15.6 Сбои сразу после входа в систему

Если вы можете войти в систему, но не можете использовать терминал, то возможно что при запуске шелл терминал реконфигурируется (неправильно) командой, которую кто-то поместил в один из файлов, которые выполняются, когда вы входите в систему. Это файлы - /etc/profile и /.bashrc. Ищите команду, начинающуюся с "stty" или "setserial", и удостоверьтесь, что она правильная. Даже если она выполняется хорошо в одном файле инициализации, ее действие может быть неправильно отменено в другом файле инициализации, который вы не знаете. вы можете загрузиться со спасательной дискеты, чтобы войти в систему для фиксации проблемы.

## 15.7 Нельзя войти в систему

Если вы получаете приглашение ко входу в систему, но не получаете ответа (или возможно искаженный ответ) на вашу попытку войти в систему, то возможная причина этого, что связь является плохой односторонней от терминала к компьютер. Если вы еще не используете опцию "local" с getty, сделайте так, чтобы отключить линии управления модемом. вы могли бы также отключить аппаратное управление потоком данных (stty -crtcts), если оно разрешено. Если теперь все работает, то ваши линии управления модемом вероятно или не проводят, или не установлены правильно. Некоторые терминалы позволяют устанавливать различные значения (типа скорости) для передачи и приема, так что прием может осуществляться хорошо, а передача - плохо.

Если вы получили сообщение, говорящее что-то типа "login failed", то, если вы не ошиблись при наборе слов или в пароле, действительно могут иметься некоторые ограничения на вход в систему. См. ``Ограничение входа в систему"

## 15.8 Искаженная подсказка входа в систему

Такое может быть из-за использования неправильного символьного набора, ошибок передачи из-за слишком высокой скорости, несовместимых скоростей или несовместимого контроля по четности. Если это выглядит как набор странных символов, то у вас установлен неправильный символьный набор, или бит старшего разряда устанавливается ошибочно.

Если слова появляются с орфографическими ошибками, попробуйте более низкую скорость. При несовместимости скорости или контроля по четности вы увидите много одинаковых "символов ошибки", который представляет собой реальный символ, который не может отобразиться правильно из-за ошибок в контроле по четности или скорости.

Если вы используетеagetty (часто именуемый getty), то программаagetty обнаружит и установит контроль по четности, если вы что-то печатаете. Пробуйте его с возвратом, чтобы увидеть, исправляет ли она возможные ошибки контроля по четности.

## 15.9 Нет признаков приглашения входа в систему

Если на терминале ничего не происходит, то произошло самое страшное. Одна из первых вещей, которые надо сделать это удостовериться, что все кабельные соединители плотно вставлены и соединены правильно. Другие возможные проблемы: несоответствие скоростей в бодах, неисправные аппаратные средства или не запущен getty.

В этой точке два различных направления подхода (вы можете следовать больше одному):

- Диагностика проблемы с консоли
- Измерение напряжений

## Диагностика проблемы с консоли

В консоли (или другом рабочем терминале) используйте "top" или "ps -al", чтобы увидеть выполняется ли getty на порту. Не путайте ее с программами getty, выполняющимися на других портах или на виртуальных консолях. вы не получите подсказку входа в систему, если getty не запущена. Если она запущена, то вы можете запретить ее, чтобы попробовать скопировать файл на терминал как тест.

Чтобы запретить getty, отредактируйте /etc/inittab, и закомментируйте команду getty значком # в начале строки, чтобы она не порождалась повторно после того, как вы уничтожите ее. Затем уничтожьте старую getty, используя клавишу k в программе "top".

Чтобы скопировать короткий файл на терминал (Хорошо бы попробовать это сделать перед тем, как начать процесс установки перед настройкой getty) используйте команду cp Linux: cp file\_name /dev/ttyS1.

Если это не работает, используйте stty, чтобы сделать упростить интерфейс насколько возможно, отключив все что можно (типа аппаратного управления потоком данных: -crtsets; контроль по четности, и управляющие сигналы модема: clocal). Убедитесь, что скорости и биты/байты - одинаковы. Если ничего не случается, проверьте порт вольтметром (см. следующий раздел).

## Измерение напряжений

Если у вас есть удобный вольтметр, проверьте уровень -12 V (от -5 до -15) на выводе 3 (прием данных) на терминальном конце нуль-модемного кабеля. Положительный вывод измерителя должен быть хорошо заземлен (металлические соединители на концах кабелей часто незаземлены). Если отрицательного напряжения нет, то проверьте его в выводе 2 на компьютере. Если оно есть там, но нет в выводе 3 на терминале, то неисправен кабель (потеря контакта, порван провод, или это не нуль-модемный кабель с выводом 2 соединенным с выводом 3). Если напряжение отсутствует, то не работает последовательный порт на компьютере. Проверьте его диагностическим программным обеспечением или замените это.

Если последовательный порт действует, то вы можете посылать файл в него (с заблокированными средствами управления модема) и посмотрите, принимается ли что-нибудь на него. Чтобы проверить переданный сигнал, посмотрите на индикатор вольтметра в -12 V, когда линия неактивна. Затем начните посылать файл (или запустите getty). вы должны увидеть, что индикатор приближается к 0 и дрожит около 0, поскольку он измеряет короткие импульсы потока битов. вы можете увидеть это также на шкале AC, если ваш измеритель имеет конденсатор, чтобы блокировать внешнее напряжение DC, когда измеряют на AC. Если его нет, то задержка DC -12 V вызовет большую ошибку чтения AC. Без измерителя, вы могли бы подсоединить хорошее устройство (типа другого терминала или внешнего модема) на последовательный порт, и посмотреть работает ли оно хорошо.

## 15.10 Испытательное оборудование для последовательных устройств

Breakout устройства, и т.д.

Хотя возможно, что для нескольких терминалов, вам может понадобиться только мультиметр (используемый как вольтметр), для тестирования линий связи последовательного порта было создано специальное испытательное оборудование. Некоторые называются "breakout ...", где breakout означает - break out проводники кабеля. Эти устройства имеют два разъема и вставляются в последовательный кабель. Некоторые имеют тестовые точки для подключения вольтметра. Другие имеют светодиодные индикаторы, которые светятся, когда на некоторых линиях управления модемом есть сигнал.

Кроме того они могут иметь джамперы для соединения любого провода с любым проводом. Некоторые имеют переключатели. Radio Shack продает "RS-232 troubleshooter" и "RS-232 serial jumper box".

## Измерение напряжений

Любой вольтметр или мультиметр, даже самый дешевый, который продается за \$10, должен прекрасно работать. Попытки использовать другие методы для проверки напряжения сложны. Не используйте светодиоды, если они не имеют шунтирующего резистора. 470 ohm резистор используется для 20 ma светодиодов (но не все светодиоды - 20 ma). Светодиод светит только при соблюдении полярности напряжения на нем, так что вы можете проверить + или - напряжение. Каждый делал такое устройство для тестирования автомобильной схемы?? Логические зонды могут повреждаться, если вы пробуете использовать их для напряжений больше, чем в ТТЛ-схемах, для которых они разработаны - только 5 вольт. Попытка использовать 12-вольтовую лампочку - тоже не слишком хорошая идея. Она не будет показывать полярность и из-за ограниченного выходного уровня в UART, вероятно, даже не будет светиться.

Чтобы измерить напряжение на розеточном разъеме, вы можете воткнуть изогнутый отрезок бумаги. Диаметр должен быть не больше, чем вывода так, чтобы это не повредило контакт.

## Напряжение на вкус

В качестве последнего средства, если у вас нет никакого испытательного оборудования и вы желаете рискнуть получить электрошок (или помереть от электричества), вы всегда можете попробовать напряжение на вкус. Перед касанием одного из проверяемых проводников вашим языком, проверьте их, чтобы удостовериться, что по ним не идет высокое напряжение. Коснитесь обоими проводниками (одновременно) к одной руке, чтобы увидеть есть ли в них ток. Затем, если вы не получили удар, лизните кожу, чтобы она стала влажной и повторите. Если этот тест даст вам удар, вы конечно не захотите использовать ваш язык.

Для проверки 12 V облизайте палец и проведите одним выводом по нему.

Поместите другой проверяемый вывод на ваш язык. Если вывод на вашем языке положительный, то вы ощутите значительный вкус. Вы можете попробовать сначала с батарейками от фонарика, так что вы будете знать какой вкус ожидать.

## 16. Ремонт и диагностика

Восстановление терминала имеет много общего с восстановлением монитора и/или клавиатуры. Иногда встроенная диагностика терминала сообщит вам о неполадке сообщением на экране. Если нет, то по признакам можно часто свести проблему к одной: плохая клавиатура, неработающая электронно-лучевая трубка, отказ цифровой электроники. Самое лучшее иметь инструкцию по эксплуатации, но даже если у вас ее нет, то многие терминалы можно все еще восстановить.

## 16.1 Книги по ремонту и вебсайты

### Книги

- Bigelow, Stephen J.: Troubleshooting & Repairing Computer Monitors, 2nd edition, McGraw-Hill, 1997. Не покрывает поколение символьной электроники и клавиатуру.

### Вебсайты

FAQ <http://www.repairfaq.org> для newsgroup: sci.electronics.repair длинный и всесторонний, хотя и не охватывает терминалы во всем. См. раздел "Компьютер и видеомонитор". Многие из этой информации применимо к терминалам, как например разделы: "Тестирование конденсаторов", "Тестирование трансформаторов" и т.д. Возможно в будущем, "информация" по ремонту в этом HOWTO будет состоять главным образом из ссылок на вышеупомянутый FAQ (или подобный ему документ).

## 16.2 Безопасность

Электронно-лучевая трубка использует высокое напряжение до 30,000 вольт для цветных дисплеев (для одноцветного дисплея эта цифра несколько меньше). Будьте внимательны, чтобы не попасть под это напряжение, если дисплей включен и крышка снята. Вероятно оно не будет смертельным, даже если вы попадете под него. Но вероятен ужасный ожог, вибрация и т.д. Высокое напряжение может пробивать слой воздуха и поврежденную изоляцию, так что держите ваши руки на безопасном расстоянии от него. Вы должны обратить внимание на хорошо изолированный кабель высокого напряжения, соединенный одним концом с ЭЛТ. Даже, когда питание выключено, еще имеется остаточное напряжение на соединении кабеля ЭЛТ, чтобы вы получили удар. Чтобы разрядить это напряжение, когда питание отключено используйте заземленную отвертку (с изолированной ручкой).

Более низкие напряжения (сотни вольт) могут быть даже более опасны, особенно, если у вас влажные руки или если вы носите металлический ремешок для часов, кольцо или подобную вещь. В редких случаях людей убивало этим напряжением, так что будьте внимательны. Самые низкие напряжения в несколько вольт на цифровой схеме довольно безопасны, но не касайтесь ничем (за исключением хорошо изолированного инструмента), если вы не знаете наверняка.

## 16.3 Вид дисплея

Если дисплей слишком тускл, добавьте яркость и/или контрастность, используя кнопки на внешнем модуле (если они существуют). Если ширина, высота или центрирование неправильные, то для них часто имеются ручки управления. Для нескольких старых терминалов нужно нажать клавишу курсора (или подобные клавиши) в режиме установки.

Вам может понадобиться удалить крышку, чтобы внести изменения, особенно на старых моделях. вы могли бы поставить большое зеркало перед терминалом, чтобы смотреть на дисплей в зеркало при внесении изменений. Управляющие элементы могут находиться на печатной плате схемы. Хотя может понадобиться только отвертка (возможно Phillips-head), катушки индуктивности могут потребовать наличия специальных инструментов TV настройки (plastic hex wrenches итд). Сокращенные имена управляющих элементов должны быть напечатаны на плате схемы. Например, есть такие названия:

- V-Size Регулировка размера по вертикали
- H-Size Регулировка размера по вертикали. Это может быть катушка
- V-Pos Регулировка расположения по вертикали
- H-Pos Регулировка расположения по горизонтали
- V-Lin Регулировка вертикальной линейности (Используется, если ширина строк развертки отличается в верхней и нижней части экрана)
- V-Hold Регулировка вертикальной остановки (Используется, если экран не управляет прокруткой)
- Bright Регулировка яркости (также может иметься внешняя кнопка)
- Sub-Bright Регулировка яркости подчиненного режима интенсивности (часто нормальный режим: регулятор освещенности, а не утолщения или яркости).

Изменение линейности может изменять размер так, чтобы приходится его поправлять. Терминал, который хранился в течение некоторого времени, может иметь маленький размер раstra на экране, окруженный большой черной рамкой. Если это трудно подкорректировать, то подождите некоторое время, так как вероятно, что размеры восстановятся

## 16.4 Диагностика

### Терминал издает шум

Если терминал издавал шум перед тем, как испортиться (или когда вы включаете его после поломки), то шум может подсказать, что неправильно. Если вы слышите периодический шум или видите/чуete запах дыма, немедленно выключите терминал, чтобы предотвратить дальнейшее повреждение. Проблема вероятно в высоком питании напряжения в несколько тысяч вольт. Снимите крышку и если неисправный элемент не виден, включите его снова на короткое время в затемненной комнате, чтобы найти появляющийся дуговой разряд. Возможно нарушена изоляция высоковольтного кабеля (между преобразователем обратного хода, и ЭЛТ), что может приводить к появлению дугового разряда. Исправьте пробой изолирующим кембриком или специальной электрической лентой, разработанной для 10,000 вольт.

Преобразователь обратного хода (высоковольтный), если он неисправен, может издавать только слабое щелканье или периодический шум. вы не можете слышать его, пока не выключите терминал на некоторое время, чтобы отдохнуть, и затем включите снова. Чтобы выследить источник шума, вы можете использовать кусок маленькой резиновой трубки (типа используемого в автомобилях) как стетоскоп, чтобы услышать его. Но пока вы слушаете шум, терминал портится все больше, так что попытайтесь найти его быстро (но не так быстро, чтобы попасть под напряжение).

Шум мог быть из-за короткого замыкания в питании. При этом может перегореть плавкий предохранитель. Замена плавкого предохранителя может не решить проблемы, поскольку то же самое КЗ спалит плавкий предохранитель снова. Посмотрите, нет ли каких-нибудь темных нагретых областей и проверьте эти

компоненты. Закороченные мощные транзисторы могут спалить плавкий предохранитель. Они могут быть проверены тестером для транзисторов или даже омметром. Используйте низкоомные значения на омметре так, чтобы напряжение, прикладываемое омметром было небольшим. Это уменьшит вероятность повреждения хороших компонентов.

Если терминал стоял в сыром месте вроде комнаты без вентиляции или рядом с кухней, то ремонт может заключаться в сушке терминала. Таким образом может восстановиться "сломанный" преобразователь обратного хода.

## **Терминал не издает шума**

Пустой экран может быть из-того, что кто-то убавил яркость до самого низкого уровня или ЭЛТ состарилась. Следующая вещь, которую нужно будет сделать, - это проверить целостность кабелей. Если нет признаков питания, замените шнур питания после того, как убедитесь, что в стенной розетке питание есть.

Если подозреваете клавиатуру, пробуйте ее на другом терминале того же самого типа, или замените исправной клавиатурой. Пошевелите разъем кабеля клавиатуры. Провода внутри кабеля могут испортиться, особенно возле его концов. Если определено, что кабель испорчен, то можно или взять новый кабель или надрезать кабель и заново спаять обрывы, и т.д.

Одна из первых вещей, которые делаются, если клавиатура работает, это перевод терминала в автономный режим (в котором он работает как пишущая машинка и разъединен с компьютером). Если он хорошо работает в локальном режиме, то проблема вероятно в соединении с компьютером (или неправильном интерфейсе) или в UART чипах терминала.

Тщательный осмотр схемы часто помогает найти причину проблемы. Ищите сколы, трещины и т.д. Нестабильная работа может быть вызвана нарушением пайки контактов. Обрыв проводника печатной платы может возникнуть из-за изгиба платы. Пропаяйте подозрительные проводники. Пайка может нагреть транзисторы (и другие компоненты) и повредить их, так что используйте теплоотводы, если это возможно.

Если у вас фирменный терминал, то вы можете попробовать найти группу новостей в Интернете, чтобы выяснить наиболее частые типы проблем для вашего терминала и возможно информацию о том, как их устранить.

Чтобы увидеть работает ли цифровая электроника, попробуйте (используя хорошую клавиатуру) напечатать текст на плохом терминале. Попробуйте прочитать этот текст на хорошем терминале (или консоли), используя команды сору или терминальную программу типа Minicom. Вы можете понадобиться нажать клавишу RETURN на терминале, чтобы послать строку. Можно запросить у плохого терминала сведения о нем и т.д. с другого терминала. Это позволит определить работоспособность двухсторонней связи.

## **16.5 Конденсаторы**

Электролитические конденсаторы имеют металлическую оболочку и могут изменить емкость или испортиться, если они пролежали в течение нескольких лет без использования. Иногда только оставление терминала на некоторое время поможет частично восстанавливать их. Если вы можете, включайте терминалы на вашем складе хотя бы раз или два в год.



## 16.6 Клавиатуры

### Взаимозаменяемость

Клавиатура для терминалов не тоже самое, что и клавиатура для PC.

Различие не только в размещении клавиш, но и в генерируемых кодах, когда нажимается клавиша. Также клавиатуры для различных марок и моделей терминалов не всегда взаимозаменяемы друг с другом. Иногда получается "несовместимая" клавиатура, которая частично работает на терминале. Все клавиши ASCII будут работать хорошо, но специальные клавиши типа клавиша установки и прерывания не будут работать правильно.

### Как они работают

Большинство клавиатур только просто замыкает два проводника, когда вы нажимаете клавишу. Электроника в микросхеме в клавиатуре преобразует это замыкание в код, посылаемый по внешнему кабелю клавиатуры. Вместо того, чтобы иметь отдельный провод (или проводник) идущий от каждой клавиши к микросхеме, используется следующий тип схемы. Проводники обозначаются 1-10 и A-J. Например: проводник 3 идет на несколько клавиш, и проводник В идет на несколько клавиш, но только одна клавиша имеет оба проводника 3 и В, идущий на нее. Когда эта клавиша нажата, закоротятся провода 3 и В. Микросхема определяет это замыкание и узнает, какая клавиша была нажата. Такая схема уменьшает число необходимых проводников (и уменьшает число выводов, необходимых на микросхеме). Это напоминает перекрестный переключатель.

### За одно нажатие высказывают 2 различных символа

Если из-за дефекта проводники 3 и 4 закоротятся, то нажатие клавиши 3-В также закоротит 4 и В, и микросхема будет думать, что были нажаты обе клавиши 3-В и 4-В. Это вероятно приведет к печати 2 различных символов, когда вы хотели бы напечатать один символ.

### Современные клавиатуры против старых

В то время как современные и старая клавиатуры выглядят очень похоже, механика обработки в них различна. Старые клавиатуры имеют индивидуальные переключатели клавиши под заглавными буквами клавиши и каждый переключатель вставлен в жесткий пластмассовый корпус. Современные используют большие гибкие пластмассовые листы (мембраны) размером с клавиатуру. Пластмассовый лист с отверстиями в ней зажат между двумя другими пластмассовыми листами, содержащими печатные схемы (включая контактные точки. Когда вы нажимаете клавишу, два "печатных" листа скоприкасаются вместе в некоторой точке, замыкая контакты, напечатанные на листах в этой точке.

### Клавиатура не работает вообще

Если ни одна из клавиш работает, попробуйте подключить другую клавиатуру (если у вас есть), чтобы убедиться, что проблема именно в клавиатуре. Наиболее вероятная причина - испорчен провод внутри кабеля, который соединяет ее с терминалом.

Наиболее вероятные места обрыва - оба конца кабеля.

Пробуйте wiggling концы шнура при нажатой клавише, чтобы увидеть, работает ли она периодически. Вы можете аккуратно надрезать кабель ножом в плохом месте и замкнуть оборванный проводник.

Иногда это не обрыв, а разрушение паяного соединения. Скрепите кабель электрической лентой, клеем или caulk.

## **Печать b отображает bb, и т.д. (удвоение символов)**

Если все символы появляются двойными имеется вероятность, что не в порядке клавиатура. Кроме того, вероятно, что ваш терминал был неправильно установлен для полудуплекса (локальный ECHO ON) и каждый символ, который вы напечатаете, посылается на экран и из электроники внутри вашего терминала, и из вашего главного компьютера. Если два символа - не одинаковые, то может иметься замыкание внутри вашей клавиатуры. См. ``За одно нажатие выскакивают 2 различных символа"

## **Клавиатура печатает самопроизвольно**

Если клавиша замкнута, то она вероятно печатает большое количество одного того же самого символа, если включено автоповторение. Если замкнуть больше, чем одна клавиша, тогда будут печататься несколько повторяющихся последовательностей символов. Это может привести к повторным слишком быстрым восстановлениям getty, если это случилось во время выдачи на экран подсказки входа в систему. См. ``Клавиша замкнута". Это исправляется очисткой контактов, см. главу ``Очистка контактов клавиатуры".

## **Жидкость, пролитая на клавиатуру**

Если на клавиатуре была пролита вода или другая жидкость (или клавиатура попала под дождь, тяжелую росу или лежала в сырости), то некоторые клавиши могут работать неправильно. Сырость может вызывать замыкание клавиши (словно она нажата все время), и вы можете увидеть, что экран заполняется соответствующим этой клавише символом, если допускается автоповторение. Если клавиатура была влажной и затем частично (или полностью) высохла, некоторые клавиши могут не работать из-за налета на поверхностях контакта. Для современного типа клавиатуры, можно легко демонтировать пластмассовые листы внутри и осушить/очистить их. Для старых типов клавиатур можно просушить их на солнце или в духовке (при низкой температуре). Когда клавиатура высохнет, все еще может потребоваться почистить контакты на некоторых клавишах, как это объяснено ниже.

## **Очистка контактов клавиатуры**

### **Клавиатуры с мембранами**

На некоторых новых клавиатурах, пластмассовые листы (мембраны) при необходимости легко убираются для проверки и очистки. вы должны удалить только несколько винтов, чтобы демонтировать клавиатуру и приняться за листы. На некоторых клавиатурах старых IBM листы не могут быть удалены без того, чтобы не сломать несколько пластмассовых заклепок, которые придется при вставке на место смазать клеем. Такую клавиатуру можно иногда починить сгибанием, скручиванием, и/или тряской конструкции, содержащей пластмассовые листы.

## Клавиатуры с индивидуальными переключателями

Что нужно сделать для старых клавиатур, которые имеют отдельные жесткие пластмассовые переключатели для каждой клавиши. Перед началом всей работы по очистке электрических контактов сначала попробуйте перевернуть и порясти клавиатуру и посмотреть не заработают ли неработавшие клавиши. Это может помочь слететь накопившейся грязи, особенно, если вы нажимаете клавишу интенсивно и быстро, чтобы создать вибрацию.

Часто "крышечки" клавиш могут быть удалены, вытащив их вверх, используя маленькую отвертку как рычаг для предотвращения чрезмерного наклона к пальцу. (Предупреждение: На современных клавиатурах это не получится.) "Крышечка" клавиши может наклониться немного и колебаться, так что есть риск ее потерять. Она может даже взлететь вверх и упасть на пол. Далее у вас имеется два варианта как очистить контакты: использовать распылитель очистителя контактов непосредственно на верх переключателя клавиши, или вытащить переключатель клавиши и очистить его. Есть еще другой вариант - заменить переключатель клавиши на новый или использованный.

Непосредственное распыление контактного очистителя или подобного средства на верхнюю часть переключателя клавиши - это самый быстрый метод, но он может не сработать, а также может повредить пластмассу. С включенной клавиатурой (или с контактами клавиши, соединенными с омметром) используют трубку, которая вставляется с распылителем шприца очистителя внутрь переключателя клавиши. Наклоните клавиатуру так, чтобы очиститель стекал на контакты. Для терминала CIT101E с клавиатурой Alps, это значит наклонить цифровым рядом вверх. Понажимайте клавишу вверх-вниз ручкой или маленькой отверткой, чтобы избежать попадания ядовитого очистителя на вашу кожу (или оденьте перчатки). В заключение переверните клавиатуру работая клавишей, чтобы высушить ее от очистителя. Чем больше очистителя вы впрыснете, тем более вероятно, что он очистит контакты, но также более вероятно, что он попортит пластмассу, так что используйте такое количество, какое вы полагаете достаточным. Как только клавиша заработает, поработайте ею вверх и вниз, проверьте спустя полминуты и т.д. чтобы удостовериться, что она будет и дальше нормально работать.

Some switches have the contacts nearly sealed inside so little if any contact cleaner reaches the contacts.

Иногда клавиша работает лучше, когда контакты внутри контактируют с жидкостью очистителя, но когда жидкость через несколько минут высохнет, контакт пропадает, клавиша начинает плохо работать (если вообще будет работать). Работа клавишей, когда жидкость высыхает внутри, может помочь. Некоторые переключатели имеют контакты, почти запечатанные внутри так немного, если любой входит в контакт более чистый, достигает контактов.

Очиститель, попавший на контакты может занести грязь.

Если вам нужно разобрать переключатель клавиши(ключа), сначала рассмотрите его, чтобы увидеть, как он установлен и как вытаскивается. Иногда можно удалить крышку переключателя без того, чтобы вытащить переключатель из клавиатуры. Чтобы сделать это потяните (или перемещайте) верхнюю часть переключателя клавиши после отделения пластмассовых заклепок, которые ее удерживают. Не дергайте слишком сильно, а то вы можете разбить тонкую пластмассу. Если это не может быть выполнено, вам, вероятно, придется отпаять переключатель и удалить их, чтобы брать извлечь переключатель (или заменить его). Как только переключатель был демонтирован, вы можете не увидеть контакты, если поверхности контактов сжаты вместе (почти касаются). вы можете нанести контактный очиститель на контакты, немного разведя проводящие поверхности и впрыскивая очиститель между ними. Могут иметься некоторые типы отсекаелей, удерживающие поверхности контактов вместе, который надо удалить перед разделением этих поверхностей.

С очистителем на контактах, поработайте ими. Наклон или переворачивание клавиатуры может помочь. Соблюдайте осторожность при работе с незакрепленными маленькими детальками, поскольку они могут при демонтаже переключателя клавиши вылететь и потеряться.

## 17. Приложение

### 17.1 Список команд терминала Linux

#### Посылка команды на терминал

- ```setterm```: длинные опции
- ```tput```: краткие опции
- `tset`: только инициализация
- `clear`: очищает экран
- `reset`: посылает строку сброса

#### Конфигурирование драйвера устройства терминала

- ```setserial```
- ```stty```

#### Terminfo

- ```tic``` terminfo компилятор и транслятор
- `toe`: показывает список терминалов, для которых у вас имеется terminfo (termcaps)
- ```infocmp``` сравнивает или отображает записи terminfo

#### Другое

- `gitkeys`: показывает то, какие байты каждая клавиша посылает главной ЭВМ.

- `tty`: показывает с каким `tty` вы соединены
- `set` (или `tset -q`): показывает значение `TERM`, имя записи `terminfo`
- ```tset```: устанавливает `TERM` в интерактивном режиме и инициализирует

## 17.2 Интернет и Книги

### Информация о терминалах в Интернет

- Website Шуфорда [http://www.cs.utk.edu/~shuford/terminal\\_index.html](http://www.cs.utk.edu/~shuford/terminal_index.html) в Университете Штата Теннесси имеет много полезной информации относительно текстовых терминалов.
- Boundless <http://www.boundless.com/textterm/> приобрел VT и терминальный бизнес Dorio у DEC.
- Wyse <http://www.wyse.com/terminal> - главный изготовитель терминалов.
- `comp.terminals` - newsgroup для терминалов

### Книги, относящиеся к терминалам

- последовательный порт EIA-232 см. ``Книги EIA-232 (RS-232)``.
- Ремонт см. ``Книги и вебсайты по ремонту``.
- База данных `terminfo` см. ``Документы по `termcap```

### Все книги по терминалам

Насколько я знаю, по текстовым терминалам удовлетворительной книги нет (если вы не интересуетесь старинными терминалами 1970-ых).

- Руководство по интерактивным компьютерным терминалам Duane E. Sharp; ; Reston Publishing Co. 1977. (Очень устаревшая)
- Communicating with Display Terminals by Roger K. deBry; McGraw-Hill 1985. (в основном о синхронных терминалах IBM )

"РУКОВОДСТВО ..." представляет краткую спецификацию из более чем 100 различных моделей старинных терминалов, сделанных в начале 1970-ых более чем 60 различными компаниями. Оно также объясняет, как они работают физически, но неправильно показывает диаграмму для электронно-лучевой трубки, которая использует электростатическое отклонение электронного луча (р. 36). Терминалы фактически использовали магнитное отклонение (даже в 1970-ых). Эта книга объясняет ряд продвинутых технических понятий (концепций) типа "произвольное сканирование" и "цветной принцип проникновения".

Книга "COMMUNICATING ..." в отличие от "Руководство ..." игнорирует физические и электронные подробности терминалов. Она имеет целую главу, объясняющую двоичные исчисления (что не нужно в книге о терминалах, так как эта информация широко доступна в других местах). Кажется, она сосредоточена на старых терминалах IBM (главным образом 3270) в блочных и синхронных режимах работы. Она мало рассказывает о использовании ANSI терминалов, используемых сегодня на Unix-подобных системах.

Хотя она обсуждает их немного, она не показывает различные схемы соединений, используемые для их соединения с последовательными портами.

### **Книги с главами о терминалах**

Эти главы не рассказывают почти ничего о терминалах непосредственно и их возможностях. Более того, эти главы обычно рассказывают о том, как установить компьютер (и драйвер терминала), чтобы работать с терминалами.

Из-за различий Unix-подобных систем, многое из этой информации не относится к Linux.

- Unix Power Tools. Jerry Peck. Al. O'Reilly 1998. Ch. 5 Установка Вашего Терминала, Ch. 41: Установка терминала и последовательных линий, Ch. 42: Проблемы с терминалами
- Advanced Programming in the Unix Environment. W. Ричард Stevens Addison- Wesley, 1993. Ch. 11: Ввод-вывод терминала, Ch. 19: Псевдотерминалы
- Essential System Administration by Aleen Frisch, 2nd ed. O'Reilly, 1998. Ch. 11: Терминалы и Модемы.

Книга "UNIX POWER TOOLS" книга имеет 3 коротких главы по текстовым терминалам.

Она менее практична, чем этот HOWTO, но дает большее количество примеров, чтобы помочь вам.

"ADVANCED PROGRAMMING ..." Глава 11 покрывает только аспект драйвера устройства (или драйвер терминала) терминала (программное обеспечение, которое включено в операционную систему для работы с терминалами). Она объясняет параметры, которые передаются команде stty, чтобы сконфигурировать терминал.

"ESSENTIAL SYSTEM ..." глава книги относится больше к терминалам, чем к модемам. Кажется она хорошо написана.