

Из DOS/Windows в Linux HOWTO

Guido Gonzato, [ggonza at tin.it](mailto:ggonza@tin.it), перевод Alex Ott, ott@phtd.tpu.edu.ru

v1.3.5, 31 Августа 2000

Этот HOWTO предназначен для всех пользователей (которые скоро станут бывшими?) DOS и Windows, которые решили перейти к использованию Linux---свободной версии UNIX. Целью этого документа является помощь в переносе знаний пользователя о DOS и Windows в среду Linux, а также предоставление советов об обмене файлами и ресурсами между этими двумя операционными системами.

1. Введение

2. Благодарности

Переводчик благодарит Лену Савельеву, вычитавшую этот документ и внесшую исправления, чтобы его могли читать русские люди :))

Особая благодарность автору этого документа, за такое превосходное введение в Линукс.

2.1 Нужен ли Вам Linux?

Давайте будем политически корректными. В этом документе я говорю ``Linux'', но я подразумеваю ``GNU/Linux''. Пожалуйста прочитайте об этом информацию на <http://www.gnu.org/gnu/linux-and-gnu.html>.

Вы хотите перейти из мира DOS/Windows в мир Linux? Это хорошая идея: Linux в техническом отношении превосходит DOS, Windows 9x и даже Windows NT. Но будьте осторожны: это может быть не совсем полезным для вас. Вот основные отличия между DOS/Windows и Linux:

- Под управлением Windows работают Microsoft Office и большое количество игр; Windows воспринимается как легкий в установке и настройке; он, как всем известно, нестабилен; реализован плохо; часто падает (crash).
- Под управлением Linux работает StarOffice, десятки технических программ и некоторое количество игр; он может быть труден в установке и настройке; он устойчив; реализован великолепно; падения очень редки.

Вы должны решить, что вам необходимо. Более того, Linux дает вам мощь, но он требует некоторого времени для того, чтобы научиться, как обуздывать его. Таким образом, если вам в основном нужно коммерческое программное обеспечение, или вы не чувствуете в себе силы для обучения новым командам и концепциям, то вам лучше поискать что-нибудь другое. Сознajte, что много новичков сдались под действием начальных трудностей.

Ведется работа для того, чтобы сделать Linux более простым для использования, но *не ожидайте, что вы станете умелым пользователем до тех пор, пока не прочтаете достаточное количество документации и не будете использовать его по крайней мере несколько месяцев*. Linux не даст вам мгновенных результатов, Я на 100% уверен, что если вы являетесь нормальным пользователем, то Linux станет вашей компьютерной нирваной. Между прочим, Linux + DOS/Win могут счастливо сосуществовать на одной и той же машине.

Начальные требования для данного документа: я предполагаю, что:

- вы знаете основные команды и концепции DOS;
- Linux, возможно с системой X Window System (для краткости X11), правильно установлен на вашем персональном компьютере;
- вашим командным процессором (эквивалент `COMMAND.COM`) является `bash`.

До тех пор, пока явно не будет указано, вся информация в этом документе относится к старому, плохому DOS. И там и здесь существует информация о Windows, но помните, что Windows и Linux полностью отличны друг от друга, в отличие от DOS, который является чем-то вроде бедного родственника UNIX.

Также пожалуйста отметьте, что эта работа не является ни полным учебником, ни руководством по настройке!

Последние версии этого документа доступны в различных форматах по адресу <http://www.linuxdoc.org>.

2.2 Расскажите мне больше

Вы установили Linux и необходимые вам программы на ваш персональный компьютер. Вы создали себе учетную запись (если вы это не сделали, наберите *сейчас* `adduser ваше_имя`) и Linux запущен. Вы просто ввели ваше имя и пароль и теперь вы смотрите в экран и думаете: ``Хорошо, но что теперь?``

Теперь не отчаивайтесь. Вы теперь готовы делать те же самые вещи, что вы делали, используя DOS/Win, а также многое другое. Если бы вы запустили DOS/Win вместо Linux, то вы могли бы выполнять некоторые из следующих задач:

- запуск команд и создание, копирование, просмотр, удаление, печать, переименование файлов;
- Перемещение, создание, удаление и просмотр содержимого ваших каталогов файлов;
- Форматирование гибких дисков и копирование файлов на них или с них;
- Настройка системы;

- Работа с Internet;
- написание командных .BAT файлов и программ на вашем любимом языке программирования;
- Остальной 1% дел.

Вы будете рады узнать, что эти задачи могут быть выполнены в Linux в манере, сходной с DOS. В DOS средний пользователь использует очень небольшое количество из ста с лишним доступных команд: то же самое можно сказать и о Linux.

Предварительные концепции

Лучший способ научиться чему-либо---это окунуться в это. Эксперименты и игры с Linux поощряются: до тех пор, пока вы не войдете в систему как ``root" (администратор), вы не сможете причинить большого вреда системе. Несколько моментов:

- В первую очередь, как безопасно завершать работу с Linux. Если вы видите перед собой экран в текстовом режиме, то нажмите <CTRL-ALT-DEL>, подождите, пока система не перезагрузится, а затем выключите компьютер. Если вы находитесь в системе X Window, то сначала нажмите <CTRL-ALT-BACKSPACE>, а затем <CTRL-ALT-DEL>. *Никогда* не выключайте питание и не нажимайте кнопку сброса сразу: это может разрушить вашу систему;
- В отличие от DOS или Windows, Linux имеет встроенные механизмы безопасности. Файлы и каталоги имеют права доступа, ассоциированные с ними; в результате некоторые из них не могут быть доступны обычному пользователю; (смотрите раздел [Права доступа и владение](#)). DOS и Windows, по контрасту, могут позволить вам вычистить содержимое всего вашего жесткого диска;
- существует специальный пользователь, которого зовут ``root": это системный администратор с правом распоряжаться жизнью и смертью вашей машины. Если вы работаете на своем собственном компьютере, то вы станете таким администратором. Работа под именем администратора *очень опасна*: любая ошибка может серьезно нарушить работу или разрушить систему, почти точно так же как в DOS/Win. Не работайте с правами root, пока это не станет совершенно необходимо;
- большая часть сложности Linux происходит из сильнейшей возможности настройки: почти каждое свойство и каждое приложение могут быть настроено, используя один или несколько файлов настройки. Сложность является ценой, которую надо платить за мощь;
- Перенаправление и конвейеризация являются одним из свойств DOS, но под Linux они являются очень важными и намного более мощными. Простые команды могут быть скомбинированы для выполнения сложных задач. Я предполагаю, что вы научитесь, как использовать эти свойства.

Получение помощи

Существует много способов получить помощь о Linux. Вот наиболее важные:

- *чтение документации*---я это подразумеваю. Хотя документ, который вы читаете, может служить как введение в Linux, также существует несколько книг, которые вы обязательно должны прочитать: Matt Welsh's ``Linux Installation and Getting Started (Установка Linux и начало работы)" (<http://www.linuxdoc.org/LDP/gs/gs.html>), Larry Greenfield ``Linux User Guide (Руководство пользователя Linux)", и Linux FAQ (<http://www.linuxdoc.org/FAQ/Linux-FAQ/index.html>). Чувствуйте себя виноватым до тех пор, пока не прочитаете одну из этих книг;
- документация на установленные на машины пакеты часто может быть найдена в подкаталогах каталога /usr/doc/;
- Для получения некоторой помощи о ``внутренних командах " командного процессора наберите `help` или, еще лучше, `man bash` или `info bash`;
- для получения помощи о команде наберите `man command`, что выдаст справочную страницу(``man") для заданной команды `command`. Также можно набрать `info command`, что выдаст, если доступна, страницу информации (`info`) для данной `command`; `info` является гипертекстовой системой документации, возможно не интуитивно понятной в начале. В конце вы можете попробовать команды `apropos command` или `whatis command`. Во всех этих командах для выхода используется символ `'q'`.
- В Internet: правильным местом для получения помощи является Usenet, подобный [news:comp.os.linux.setup](http://news.comp.os.linux.setup). Пожалуйста, не пишите мне для получения помощи, потому что я слишком загружен работой.

2.3 Соглашения

В течении этого документа примеры часто будут приводится в следующем формате: `<...>` является требуемым аргументом, тогда как `[...]` является необязательным аргументом. Пример:

```
$ tar -tf <file.tar> [> redir_file]
```

`file.tar` должен быть указан , а перенаправление в `redir_file` является необязательным.

``RMP" означает ``пожалуйста Прочитайте Справочные страницы (Man Pages) для более детальной информации". Я не знаю, как подчеркнуть как важно чтение документации. ``DOSWin" означает ``DOS/Windows".

Когда в приглашении командной строки примера имеется знак #, то эта команда может быть выполнена только администратором (root).

3. Для нетерпеливых

Хотите добиться успеха? Посмотрите в эту таблицу:

DOS	Linux	Примечание
ATTRIB (+-)attr file	chmod <mode> file	полностью отличаются то же самое
BACKUP	tar -Mcvf device dir/	

CD dirname\\	cd dirname/	почти тот же самый синтаксис
COPY file1 file2	cp file1 file2	то же самое
DEL file	rm file	будьте осторожны, нет
восстановления файлов		
DELTREE dirname	rm -R dirname/	то же самое
DIR	ls	не полностью похожий синтаксис
DIR file /S	find . -name file	полностью отличаются
EDIT file	vi file	я думаю, вы не полюбите его (а зря!)
	jstar file	выглядит как редактор в dos
EDLIN file	ed file	забудьте его
FORMAT	fdformat,	
	mount, umount	достаточно отличный синтаксис
HELP command	man command,	та же философия
	info command	
MD dirname	mkdir dirname/	почти тот же самый синтаксис
MORE file	less file	намного лучше
MOVE file1 file2	mv file1 file2	то же самое
NUL	/dev/null	то же самое
PRINT file	lpr file	то же самое
PRN	/dev/lp0,	
	/dev/lp1	то же самое
RD dirname	rmdir dirname/	почти тот же самый синтаксис
REN file1 file2	mv file1 file2	не для множества файлов
RESTORE	tar -Mxpvf device	другой синтаксис
TYPE file	less file	намного лучше
WIN	startx	на разных полюсах!

Если вам нужно больше, чем таблица команд, то пожалуйста, посмотрите следующие разделы.

4. Встреча с bash

Хорошая новость: используя Linux, вы набираете немного меньше в командной строке, потому что bash набирает за вас, где это возможно, имеет великолепные средства редактирования командной строки. Для начала, клавиша 'стрелка вверх' вызывает предыдущую командную строку; но это не все. Нажатие на клавишу <TAB> дополняет имена файлов и каталогов, так что набор строки

```
$ ls /uTABloTABbTAB
```

подобен набору

```
$ ls /usr/local/bin
```

Если существует двусмысленность, такая как при наборе

```
$ ls /uTABloTABiTAB
```

bash остановится, потому что он не знает, что вы имеете в виду: /usr/local/info или /usr/local/include. Задайте больше символов, а затем снова нажмите на <TAB>.

Другие полезные комбинации клавиш: <ESC-BACKSPACE> удаляет слово влево, тогда как <ESC-D> удаляет слово вправо; <ESC-F> перемещает курсор на одно слово вправо, а <ESC-B> на слово влево; <CTRL-A> перемещает курсор в начало строки, <CTRL-E> в конец строки. Клавиша <ALT> эквивалентна клавише <ESC>.

Достаточно на первый раз. После того, как вы раз используете эти сочетания клавиш, вы обнаружите, что командная строка DOS очень раздражает...

5. Файлы и программы

5.1 Файлы: Предварительные замечания

Структура каталогов и файлов в Linux очень сходна с той, которая используется в DOSWin. Файлы имеют имена, которые подчиняются специальным правилам и хранятся в каталогах, некоторые из файлов являются исполнимыми и у некоторых из них существуют ключи командной строки. Более того, вы можете использовать символы шаблонов, перенаправление и конвейеризацию. Есть только несколько небольших отличий:

- в DOS, имена файлов хранятся в так называемом формате 8.3; например `NOTENOUGH.TXT`. В Linux мы можем делать лучше. Если вы установили Linux, используя файловую систему, подобную ext2 или umsdos, то вы можете использовать более длинные имена файлов (до 255 символов), и с более чем одной точкой в имени: например, `This_is_a.VERY_long.filename`. Пожалуйста заметьте, что я использовал символы и верхнего и нижнего регистра: в действительности...
- символы верхнего и нижнего регистра в именах файлов отличаются друг от друга. Поэтому `FILENAME.tar.gz` и `filename.tar.gz` являются разными файлами. `ls` является командой, а `LS` является ошибкой;
- пользователи Windows, будьте осторожны при использовании длинных имен в Linux. Если имя файла содержит пробелы (не рекомендуется, но возможно), то вы должны заключить имя файла в двойные кавычки при обращении к нему. Например:

- `$ # следующая команда создает каталог, названный "My old files"`
- `$ mkdir "My old files"`
- `$ ls`
- `My old files bin tmp`

А Также некоторые символы не должны использоваться: это `!*$&#`.

- не существует обязательных расширений типа `.COM` и `.EXE` для программ или `.BAT` для командных файлов. Исполнимые файлы отмечаются символом звездочка `*` в конце их имен, при использовании команды `ls -F`. Например:

- `$ ls -F`
- `I_am_a_dir/ cindy.jpg cjpg* letter_to_Joe my_1st_script* old~`

Файлы `cjpg*` и `my_1st_script*` являются исполнимыми, то есть ``программами". В DOS, файлы резервных копий заканчиваются на `.BAK`, тогда как под Linux они заканчиваются на знак тильды `~`!. Также файл, чье имя начинается со знака точки, считается скрытым. Например: файл `.I.am.a.hidden.file` не будет показан при выполнении команды `ls`;

- Ключи командной строки (switches) программ DOS задаются в виде `/switch`, Linux использует ключи в виде `-switch` или `--switch`. Пример: `dir /s` становится `ls -R`. Заметим, что много программ DOS, такие как PKZIP или ARJ, используют ключи в стиле UNIX.

Вы можете теперь перепрыгнуть к разделу [Перевод команд из DOS в Linux](#), но я надеюсь, что вы будете читать дальше.

5.2 Символьные ссылки

В UNIX имеется тип файлов, которых нет в DOS: символичные ссылки. Он может быть описан как ссылка на файл или каталог и может быть использован вместо файла или каталога, на который она указывает; они сходны с ярлыками в Windows. Примером символической ссылки может быть `/usr/X11`, которая указывает на каталог `/usr/X11R6`, или `/dev/modem`, который указывает либо на `/dev/ttyS0` либо на `/dev/ttyS1`.

Для того, чтобы создать символическую ссылку, используется команда:

```
$ ln -s <file_or_dir> <linkname>
```

Пример:

```
$ ln -s /usr/doc/g77/DOC g77manual.txt
```

Теперь вы можете ссылаться на `g77manual.txt` вместо `/usr/doc/g77/DOC`. Ссылки показываются следующим образом в списке файлов каталога:

```
$ ls -F
g77manual.txt@
$ ls -l
(разные вещи...)          g77manual.txt -> /usr/doc/g77/DOC
```

5.3 Права доступа и владение

Файлы и каталоги DOS имеют следующие атрибуты: А (архивный), Н (скрытый), R (только чтение) и S (системный). В Linux только Н и R имеют значение: скрытые файлы начинаются с точки, а для того чтобы задать атрибут R, читайте дальше.

В UNIX файл имеет "права доступа" и "владельца", который относится к "группе". Посмотрите пример:

```
$ ls -l /bin/ls
-rwxr-xr-x 1 root bin 27281 Aug 15 1995 /bin/ls*
```

Первое поле содержит права доступа к файлу `/bin/ls`, который принадлежит пользователю `root`, группы `bin`. Опуская другую информацию, запомните, что означает `-rwxr-xr-x`, слева направо:

- это тип файла (- = обычный файл, d = каталог, l = ссылка, и т.д.); `rw` являются правами доступа для владельца файла (чтение, запись, выполнение); `r-x` являются правами доступа для группы, к которой относится владелец файла (чтение, выполнение); (Я не хочу делать обзор концепции групп, вы можете выжить без нее, поскольку вы начинающий пользователь ;-)) `r-x` являются правами доступа для всех остальных пользователей (чтение, выполнение).

Каталог `/bin` тоже имеет права доступа: смотрите раздел [Права доступа к каталогам Directories Permissions](#) для дополнительной информации. Вот поэтому вы не можете удалить файл `/bin/ls` до тех пор, пока не станете пользователем `root`: вы просто не имеете прав сделать это. Для изменения прав доступа к файлу используется команда:

```
$ chmod <whoXperm> <file>
```

где `who`---это `u` (пользователь, который является владельцем), `g` (группа), `o` (остальные), `X`---это либо `+`, либо `-`, `perm`---это `r` (чтение), `w` (запись), или `x` (выполнение). Вот общие примеры использования команды `chmod`:

```
$ chmod +x file
```

устанавливает право на выполнения данного файла.

```
$ chmod go-rw file
```

удаляет право на чтение и запись для всех, кроме владельца файла.

```
$ chmod ugo+rwx file
```

дает всем права на чтение, запись и выполнение.

```
# chmod +s file
```

делает так называемый ``setuid" или ``suid" файл---файл, который любой может выполнять с привилегиями владельца. Обычно, это применяется к файлам, владельцем которых является root; часто это важные системные файлы, такие как X-сервер.

Более коротким способом для ссылки на права доступа является использование цифрового обозначения: `rwxxr-xr-x` может быть выражен как `755` (каждый символ соответствует биту: --- равен 0, --x равен 1, -w- равен 2, -wx равен 3...). Это выглядит трудным, но, немного попрактиковавшись, вы поймете концепцию. root, будучи администратором, может изменять права доступа на файлы любого пользователя. RMP.

5.4 Файлы: Перевод команд

На левой стороне находятся команды DOS; справа их аналоги в Linux.

ATTRIB:	chmod
COPY:	cp
DEL:	rm
MOVE:	mv
REN:	mv
TYPE:	more, less, cat

Операторы перенаправления и конвейеризации: < > >> |

Символы шаблонов: * ?

устройство nul: /dev/null

Устройства prn, lpt1: /dev/lp0 or /dev/lp1; lpr

Примеры

DOS	Linux
C:\GUIDO>ATTRIB +R FILE.TXT	\$ chmod 400 file.txt
C:\GUIDO>COPY JOE.TXT JOE.DOC	\$ cp joe.txt joe.doc
C:\GUIDO>COPY *.* TOTAL	\$ cat * > total
C:\GUIDO>COPY FRACTALS.DOC PRN	\$ lpr fractals.doc
C:\GUIDO>DEL TEMP	\$ rm temp
C:\GUIDO>DEL *.BAK	\$ rm *~
C:\GUIDO>MOVE PAPER.TXT TMP\	\$ mv paper.txt tmp/
C:\GUIDO>REN PAPER.TXT PAPER.ASC	\$ mv paper.txt paper.asc
C:\GUIDO>PRINT LETTER.TXT	\$ lpr letter.txt
C:\GUIDO>TYPE LETTER.TXT	\$ more letter.txt
C:\GUIDO>TYPE LETTER.TXT	\$ less letter.txt
C:\GUIDO>TYPE LETTER.TXT > NUL	\$ cat letter.txt > /dev/null
n/a	\$ more *.txt *.asc
n/a	\$ cat section*.txt less

Примечания:

- * намного ``умнее " в Linux: * соответствует всем файлам, за исключением скрытых; .* соответствует всем скрытым файлам (но также текущему каталогу `.` и каталогу уровнем выше `..`: beware!); *.* соответствует только тем, которые имеют `.` в середине имени, или оканчиваются на точку; p*r соответствует и `peter` и `piper`; *c* соответствует и `picked` и `peck`;
- при использовании `more`, необходимо нажимать на клавишу `<SPACE>` для того, чтобы продвигаться по файлу, и ``q'` для выхода. `less` является более интуитивно понятной командой и позволяет вам использовать клавиши управления курсором;
- нет команды `UNDELETE` (восстановления после удаления), так что *дважды подумайте* до удаления чего-либо;
- в добавлении к используемым в DOS' `< > >>`, в Linux имеется `2>` для перенаправления сообщений об ошибках (`stderr`); более того, `2>&1` перенаправляет `stderr` в `stdout`, в то время как `1>&2` перенаправляет `stdout` в `stderr`;
- Linux имеет дополнительный шаблон (wildcard): `[]`. Использование: `[abc]*` соответствует файлам, начинающимся с `a`, `b`, `c`; `*[I-N1-3]` соответствует файлам, оканчивающимся на `I`, `J`, `K`, `L`, `M`, `N`, `1`, `2`, `3`;
- команда `lpr <file>` печатает файл в фоновом режиме. Для проверки состояния очереди печати используйте команду `lpq`; для удаления файла из очереди печати используйте `lprm`;
- нет команды, похожей на команду `RENAME` в DOS; так что команда `mv *.xxx *.yyy` не будет работать. команда, похожая на `REN`, доступна с [ftp://metalab.unc.edu/pub/Linux/utills/file](http://metalab.unc.edu/pub/Linux/utills/file);
- используйте команды `cp -i` и `mv -i` для того, чтобы вас предупредили, когда файл будет перезаписываться.

5.5 Запуск программ: Многозадачность и сеансы работы

Для запуска программы просто наберите ее имя, так как вы делали это в DOS. Если каталог (Раздел [Использование каталогов](#)), где находится программа, находится в переменной среды `PATH` (Раздел [Файлы инициализации системы](#)), то программа будет запущена. Исключение: в отличие от DOS, в Linux программа, расположенная в текущем каталоге, не будет запускаться до тех пор, пока вы не включите этот каталог в `PATH`. Предупреждение: если `prog` это ваша программа, то наберите `./prog` для ее запуска из текущего каталога.

Типичная командная строка выглядит примерно так:

```
$ command [-s1 [-s2] ... [-sn]] [par1 [par2] ... [parn]] [< input] [> output]
```

где `-s1`, ..., `-sn` являются ключами программы, а `par1`, ..., `parn` являются параметрами программы. Вы можете запустить несколько команд из одной командной строки:

```
$ command1 ; command2 ; ... ; commandn
```

Это все о запуске программ, но теперь легко сделать следующий шаг. Одна из основных причин использования Linux---это то, что она является многозадачной операционной системой---она может выполнять разные программы (с этого места мы будем называть их процессами) одновременно. Вы можете запускать процессы в фоновом режиме и продолжать работать. Более того, Linux позволяет вам иметь несколько сеансов работы: Это подобно наличию множества компьютеров на одном!

- Для того, чтобы переключиться в сеанс 1..6 на виртуальных консолях, нажмите <ALT-F1> ... <ALT-F6>
- Для запуска нового сеанса в той же самой виртуальной консоли без выхода из нее, наберите `su - <loginname>`. Пример: `su - root`. Это полезно, например, когда вам надо выполнить задачу, которую может выполнить только root.
- Для окончания сеанса работы наберите `exit`. Если у вас есть приостановленные задания (смотрите дальше), то вас предупредят.
- Для запуска процесса в фоновом режиме, добавьте знак амперсанда `'&'` в конец командной строки:

- `$ progname [-switches] [parameters] [< input] [> output] &`
- `[1] 123`

Командный процессор идентифицирует процессы по номеру задания (например `[1]`; смотрите ниже), и по PID (Идентификационный номер процесса; `123` в нашем примере).

- Для того, чтобы увидеть, сколько процессов запущено, наберите `ps ax`. Эта команда выдаст список запущенных в настоящее время процессов.
- Для того чтобы убить, (завершить) процесс, наберите команду `kill <PID>`. Вам может понадобиться убить процесс в том случае, если вы не знаете, как его правильно завершить... До тех пор, пока вы не станете root, вы не сможете завершать процессы других людей. Иногда процесс может быть завершен только используя команду `kill -SIGKILL <PID>`. В дополнение командный процессор позволяет вам остановить или временно приостановить процесс, перевести процесс в фоновый режим, и перевести процесс из фоновый режим в интерактивный режим. В этом контексте процессы называются ``заданиями``.
- Для того чтобы узнать, сколько заданий существует наберите `jobs`. Здесь задания идентифицируются по номеру задания, а не по их PID.
- Для того, чтобы остановить процесс, запущенный в интерактивном режиме, нажмите <CTRL-C> (это не всегда работает).
- Для того, чтобы приостановить процесс, запущенный в интерактивном режиме, нажмите <CTRL-Z> (то же самое).
- Для перевода приостановленного процесса в фоновый режим, наберите `bg <%job>` (он станет заданием).

- Для перевода задания в интерактивный режим наберите `fg <%job>`. Для перевода в интерактивный режим последнего задания, запущенного в фоновом режиме, просто наберите `fg`.
- Для завершения задания наберите `kill <%job>`, где `<job>` может быть 1, 2, 3,...

Используя эти команды вы можете форматировать диск, сжимать кучу файлов, компилировать программу и разжимать архив в одно и тоже время и все равно иметь командную строку для своих нужд. Попробуйте сделать это в Windows, просто чтобы посмотреть разницу в производительности (конечно, если он не рухнет).

5.6 Запуск программ на удаленных компьютерах

Для запуска программы на удаленной машине, чьим именем является `remote.machine.edu`:

```
$ telnet remote.machine.edu
```

После входа в систему, запустите вашу любимую программу. Излишне говорить, что вам необходимо быть зарегистрированным на удаленной машине и иметь доступ к командному процессору.

Если у вас есть X11, вы даже можете запустить приложение для X на удаленной машине, отображая его на вашем экране X Windows. Допустим, что `remote.machine.edu` -- это удаленный компьютер с X и допустим, что `local.linux.box` -- это ваша машина с Linux. Для запуска с `local.linux.box` программы для X, которая расположена на `remote.machine.edu`, выполните следующее:

- запустите X11, запустите `xterm` или подобный эмулятор терминала, затем наберите:

- `$ xhost +remote.machine.edu`
- `$ telnet remote.machine.edu`

- После входа в систему наберите:

- `remote:$ DISPLAY=local.linux.box:0.0`
- `remote:$ progname &`

(вместо `DISPLAY...`, вам может понадобиться написать: `setenv DISPLAY local.linux.box:0.0`. Это зависит от командного процессора на удаленной машине).

И все! Теперь `progname` будет запущена на `remote.machine.edu` и будет отображаться на вашей машине. Не пытайтесь делать это через модем, это слишком медленно, чтобы можно было пользоваться. Более того, это грубый и небезопасный метод: пожалуйста прочитайте "Remote X Apps mini-HOWTO" по адресу <http://www.linuxdoc.org/HOWTO/mini/Remote-X-Apps.html>.

6. Использование каталогов

6.1 Каталоги: Предварительные замечания

Мы рассмотрели различия между файлами в DOSWin и в Linux. То же самое и для каталогов, в DOSWin корневой каталог обозначается \, в Linux он обозначается как /. Сходным образом, вложенные каталоги разделяются символом \ в DOSWin, и / в Linux. Примеры путей имен файлов:

```
DOS:      C:\\PAPERS\\GEOLOGY\\MID_EOC.TEX
Linux:   /home/guido/papers/geology/middle_eocene.tex
```

Как обычно, . является родительским каталогом, а . текущим. Помните, что система может не позволить выполнять команды работы с каталогами везде, где вы захотите. Каждый пользователь имеет для работы каталог, называемый 'домашним', который дан системным администратором; например, на моем компьютере моим домашним каталогом является /home/guido.

6.2 Права доступа к каталогам

Каталоги также имеют права доступа. Все что мы видели в разделе [Права доступа и владение](#), также применяется и к каталогам (пользователь, группа, и другие). Для каталога *rx* означает, что вы можете перейти в этот каталог, а *w* означает, что вы можете удалить файл в этом каталоге (конечно, соответствуя правам доступа к файлу), или сам каталог.

Например, для того, чтобы запретить другим пользователям рыться в каталоге /home/guido/text, используйте команду:

```
$ chmod o-rwx /home/guido/text
```

6.3 Каталоги: Перевод команд

```
DIR:      ls, find, du
CD:       cd, pwd
MD:       mkdir
RD:       rmdir
DELTREE:  rm -rf
MOVE:     mv
```

Примеры

DOS	Linux
C:\\GUIDO>DIR	\$ ls
C:\\GUIDO>DIR FILE.TXT	\$ ls file.txt
C:\\GUIDO>DIR *.H *.C	\$ ls *.h *.c
C:\\GUIDO>DIR/P	\$ ls more
C:\\GUIDO>DIR/A	\$ ls -l
C:\\GUIDO>DIR *.TMP /S	\$ find / -name "*.tmp"
C:\\GUIDO>CD	\$ pwd
n/a - see note	\$ cd
ditto	\$ cd ~
ditto	\$ cd ~/temp
C:\\GUIDO>CD \\OTHER	\$ cd /other
C:\\GUIDO>CD ../TEMP/TRASH	\$ cd ../temp/trash
C:\\GUIDO>MD NEWPROGS	\$ mkdir newprogs
C:\\GUIDO>MOVE PROG ..	\$ mv prog ..
C:\\GUIDO>MD \\PROGS\\TURBO	\$ mkdir /progs/turbo
C:\\GUIDO>DELTREE TEMP/TRASH	\$ rm -rf temp/trash
C:\\GUIDO>RD NEWPROGS	\$ rmdir newprogs
C:\\GUIDO>RD \\PROGS\\TURBO	\$ rmdir /progs/turbo

Замечания:

- При использовании команды `rmdir` удаляемый каталог должен быть пуст. Для того, чтобы удалить каталог и все его содержимое, используйте команду `rm -rf` (на ваш собственный риск).

- Символ `~` является сокращением для имени вашего домашнего каталога. Команды `cd` или `cd ~` выполняют переход в ваш домашний каталог из любого места; команда `cd ~/tmp` выполнит переход в каталог `/home/your_home/tmp`.
- `cd` - отменяет действие последней команды `cd`.

7. Гибкие и жесткие диски и тому подобное

Существует два способа обслуживания устройств в Linux: DOS-способ и UNIX-способ. Сделайте ваш выбор.

7.1 Обслуживание устройств DOS-способом

Большинство дистрибутивов Linux включают в себя пакет Mtools---набор команд, которые эквивалентны своим копиям в DOS, но начинаются с буквы `m`: т.е., `mformat`, `mmdir`, `mddel`, `mmd`, и так далее. Они могут даже сохранять длинные имена, но не права доступа к файлам. Если вы настроите Mtools, редактируя файл `/etc/mtools.conf` (пример поставляется вместе с дистрибутивом пакета), то вы также сможете иметь доступ к разделам DOS/Win, к устройствам CD-ROM и Zip. Однако для форматирования нового диска команда `mformat` не будет работать. Как root, вам заранее необходимо выполнить команду: `fdformat /dev/fd0H1440`.

Вы не можете иметь доступ в файлам на гибком диске для команд, таких как, например, `less a:file.txt`! Это недостаток DOS-способа для доступа к дискам.

7.2 Обслуживание устройств UNIX-способом

В UNIX имеется другой подход к обслуживанию устройств. Не существует отдельных томов, таких как `A:` или `C:`; диск, будь он гибким или любым другим, становится частью локальной файловой системы через операцию, называемую ``монтирование''. Когда вы заканчиваете работать с диском, до его извлечения вы должны ``отмонтировать'' его.

Физически форматирование диска> -- это одно действие, а создание файловой системы на нем> -- это другое. Команда DOS `FORMAT A:` выполняет оба действия, но в Linux существуют разные команды. Для форматирования гибкого диска> -- смотрите выше; для создания файловой системы используется команда:

```
# mkfs -t ext2 -c /dev/fd0H1440
```

Вы можете использовать `dos`, `vfat` (рекомендуется) или другие форматы вместо `ext2`. Приготовив диск, примонтируйте его с помощью команды

```
# mount -t ext2 /dev/fd0 /mnt
```

указывая правильную файловую систему, если вы не используете `ext2`. Теперь вы можете обращаться к файлам, используя `/mnt` вместо `A:` или `B:`. Примеры:

DOS	Linux
C:\GUIDO>DIR A:	\$ ls /mnt
C:\GUIDO>COPY A:*.*	\$ cp /mnt/* .
C:\GUIDO>COPY *.ZIP A:	\$ cp *.zip /mnt
C:\GUIDO>EDIT A:FILE.TXT	\$ jstar /mnt/file.txt
C:\GUIDO>A:	\$ cd /mnt

```
A:> _ /mnt/$ _
```

После того как вы закончите, до извлечения диска вы *должны* отмонтировать его, используя команду

```
# umount /mnt
```

Очевидно, что вы должны применять `fdformat` и `mkfs` только для неформатированных дисков, никогда до этого не применявшихся. Если вы хотите использовать диск В:, то ссылайтесь на `fd1h1440` и `fd1` вместо `fd0h1440` и `fd0` в вышеприведенном примере.

Излишне говорить, что то, что применяется для гибких дисков, также применяется для других устройств; например вы захотите примонтировать другой жесткий диск или устройство CD-ROM. Вот как примонтировать CD-ROM:

```
# mount -t iso9660 /dev/cdrom /mnt
```

Это был ``официальный'' способ монтирования ваших дисков, но существует один прием. Поскольку немного досадно, что необходимо быть `root` для монтирования гибкого диска или CD-ROM, то каждому пользователю разрешается монтировать их вот таким способом:

- как `root`, сделайте следующее:

- # `mkdir /mnt/floppy ; mkdir /mnt/cdrom`
- # `chmod 777 /mnt/floppy /mnt/cd*`
- # # make sure that the CD-ROM device is right
- # `chmod 666 /dev/hdb ; chmod 666 /dev/fd*`

- добавьте в файл `/etc/fstab` следующий строки:

- `/dev/cdrom /mnt/cdrom iso9660 ro,user,noauto 0 0`
- `/dev/fd0 /mnt/floppy vfat user,noauto 0 0`

Теперь для монтирования гибкого диска в формате DOS и CD--ROM используются команды:

```
$ mount /mnt/floppy
$ mount /mnt/cdrom
```

`/mnt/floppy` и `/mnt/cdrom` могут сейчас быть доступны любому пользователю. Помните, что разрешение всем монтировать диски таким способом -- это зияющая дыра в безопасности, если вас это беспокоит.

Двумя полезными командами являются `df`, которая дает информацию о примонтированных файловых системах, и `du dirname`, которая сообщает размер дискового пространства, занимаемого каталогом.

7.3 Резервирование данных

Существуют разные пакеты, которые помогут вам в этом занятии, но по крайней мере вы должны знать, как делается многотомное резервирование данных (как пользователь `root`):

```
# tar -M -cvf /dev/fd0h1440 dir_to_backup/
```

Убедитесь, что имеется форматированный гибкий диск в устройстве и несколько дополнительных. Для восстановления ваших данных вставьте первый гибкий диск в устройство и выполните:

```
# tar -M -xpvf /dev/fd0h1440
```

8. Что насчет Оконной системы?

"Эквивалентом" Windows является графическая система X Window System. В отличие от Windows или Mac, X11 не был спроектирован для удобства использования, или для хорошего вида, а для обеспечения графических возможностей для рабочих станций с UNIX. Вот главные отличия:

- В то время как Windows выглядит и ведет себя одинаково во всем мире, в X11 это не так: он настраивается более гибко. Весь вид X11 придается ключевым компонентом, который называется "оконный менеджер (w.m.)", и среди которых есть большой выбор: fvwm, простой, но великолепный и эффективный в отношении памяти, fvwm2-95, Afterstep, WindowMaker, Enlightenment, и многие другие. w.m. обычно запускается из .xinitrc;
- ваш w.m. может быть настроен так, что окна работают как в Windows: вы щелкаете по окну и оно появляется на переднем плане. Другая возможность заключается в том, что окно появляется на переднем плане, когда мышь находится поверх его (так называемый "фокус"). Также расположение окон на экране может быть автоматическим или интерактивным: Если вместо программы появляется странный прямоугольник, то щелкните левой кнопкой мыши там, где вы хотите, чтобы появилось окно;
- большинство свойств может быть настроено, редактируя один или несколько файлов настройки. Прочитайте документацию на ваш w.m.: файлами настройки могут быть .fvwmrc, .fvwm2rc95, .steprc, и т.д.. Простые файлы настройки обычно находятся в файлах /etc/X11/window-manager-name/system.window-manager-name;
- приложения X11 написаны, используя специальные библиотеки ("наборы widget"); поскольку доступны разные наборы, то приложения выглядят по-разному. Наиболее просто выглядят те, которые используют Athena widgets (2-D вид; xdvi, xman, xcalc); другие приложения используют Motif (netscape), Tcl/Tk, XForms, Qt, Gtk, и что еще есть. Почти все библиотеки обеспечивают привлекательный вид и feel как и Windows;
- К сожалению, поведение может быть непохожим. Например, если вы выберете строку текста, используя мышь, а затем нажмете на клавишу <BACKSPACE>, вы будете ожидать, что текст будет стерт, правильно? Это не работает в приложениях, основанных на Athena, но будет работать с другими наборами widget;
- как работают полосы прокрутки и изменение размера, зависит от w.m. и набора widget. Совет: Если вы обнаружили, что полоса прокрутки ведет себя не так, как вы ожидали, то попробуйте использовать центральную кнопку мыши или две кнопки вместе для перемещения бегунка на полосе прокрутки;
- приложения по умолчанию не имеют иконок, но они могут иметь много иконок. Большинство w.m. вызывают меню нажатием кнопки мыши на десктопе ("главном окне"); излишне говорить, что меню может быть настроено. Для изменения вида главного окна используйте команды xsetroot или xloadimage;
- Буфер обмена (clipboard) может содержать только текст и ведет себя странно. Как только вы выбрали текст, он уже скопировался в буфер обмена: переместитесь куда надо и нажмите среднюю кнопку мыши для вставки текста. Существует приложение xclipboard, которое обеспечивает несколько буферов обмена;

- drag and drop является необязательным, и он доступен, только если вы используете приложения X11 и/или w.m., которые поддерживают это.

Для сохранения памяти вы должны использовать приложения, которые используют одни и те же библиотеки, но это тяжело осуществить на практике.

Существуют проекты, целью которых является придание X11 вида и поведения, похожего на Windows. Gnome, <http://www.gnome.org> и KDE, <http://www.kde.org>, внушают благоговение. Дайте им шанс: в любом случае вы не будете сожалеть о вашем десктопе под Windows.

9. Настройка системы

9.1 Файлы инициализации системы

В DOS двумя важными файлами являются AUTOEXEC.BAT и CONFIG.SYS, которые используются во время загрузки для инициализации системы, установки некоторых переменных среды, таких как PATH и FILES, и, возможно, запуска программ или пакетных файлов. В Linux существует большее количество инициализационных файлов, некоторые из которых лучше не трогать до тех пор, пока вы не будете знать, что вы делаете. Я скажу, что самыми важными являются:

ФАЙЛЫ	ПРИМЕЧАНИЯ
/etc/inittab	не трогайте его сейчас!
/etc/rc.d/*	так же

Если вы хотите установить переменные среды PATH или другие, или вы хотите изменить сообщение при входе в систему или автоматически запустить программу после входа в систему, посмотрите следующие файлы:

ФАЙЛЫ	ПРИМЕЧАНИЯ
/etc/issue	устанавливает сообщение перед входом
/etc/motd	устанавливает сообщение после входа
/etc/profile	устанавливает переменную \$PATH и другие, и т.п.
/etc/bashrc	устанавливает алиасы и функции, и т.п.
/home/your_home/.bashrc	устанавливает ваши алиасы и функции
/home/your_home/.bash_profile	или
/home/your_home/.profile	устанавливает переменные среды и запускает ваши программы

Если последние файлы существуют (заметим, что это скрытые файлы), они будут считаны после входа в систему и команды, записанные в них, будут выполнены.

Пример---посмотрите на этот .bash_profile:

```
# Это комментарий
echo Environment:
printenv | less # эквивалент команде SET в DOS
alias d='ls -l' # легко понять, что это алиас
alias up='cd ..'
echo "I remind you that the path is "$PATH
echo "Today is `date`" # использовать вывод команды
echo "Have a good day, "$LOGNAME
# следующая часть является функцией командного процессора
ctgz() # вывести содержимое архива .tar.gz
\{
  for file in $*
  do
    gzip -dc ${file\} | tar tf -
  done
```

```
\}  
# конец .profile
```

Вы правильно угадали, что `$PATH` и `$LOGNAME` являются переменными среды,. Существует много других переменных; например, прочитайте справочные страницы для таких приложений, как `less` или `bash`.

Поместив эту строку в ваш файл `/etc/profile` вы получите грубый эквивалент `PROMPT PG`:

```
export PS1="\w\\$ "
```

9.2 Инициализационные файлы программ

В Linux практически все может быть настроено для ваших требований. Большинство программ имеют один или несколько файлов инициализации, которые вы можете править, часто это файлы `.prognamerc` в вашем домашнем каталоге. Первыми файлами, которые вы захотите исправить:

- `.inputrc`: используется `bash` для определения сочетаний клавиш;
- `.xinitrc`: используется `startx` для инициализации X Window System;
- `.fvwmrc`: используется оконным менеджером `fvwm`.
- `.joerc`, `.jstarrc`: используется редактором `joe`;
- `.jedrc`: используется редактором `jed`;
- `.pinerc`: используется программой чтения почты `pine`;
- `.xdefault`: используется многими программами для X.

Со всеми этими и другими файлами вы встретитесь раньше или позже, RMP. Возможно я должен заинтересовать вас документом "Configuration HOWTO", <http://www.linuxdoc.org/HOWTO/Config-HOWTO.html>?

10. Сети: Основы

В Linux доступен не только "Dialup Networking", который более стабилен и быстр. Другая игра называется "PPP" -- это протокол, разработанный для подключения к Internet с помощью модемов. Вам необходимо ядро с поддержкой PPP и утилиты для подключения и создания соединения.

Для получения почты с сервера вашего ISP вам необходима утилита, называемая ``вытягиватель почты (email fetcher)" , которая использует протокол POP; когда почта будет получена, она будет выглядеть, как будто она отправлена прямо на ваше машину с Linux. Затем вы можете использовать MUA (Почтовый агент пользователя), такой как `pine`, `mutt`, `elm` или один из многих других для работы с почтой.

В то время как в Windows программа соединения запускается автоматически, когда вы запускаете приложение для Internet, в Linux это делается наоборот: вы сначала производите подключения, а затем запускаете приложение. Программа, называемая `diald` обеспечивает обычное поведение. Установка и настройка `dialup networking` считается одной из самых трудных вещей в Linux, но не больше: пожалуйста посмотрите в Configuration HOWTO.

В заключение несколько слов о ``Сетевом Окружении": вы можете заставить свою рабочую станцию с Linux появиться как Windows NT/9x в локальной сети машин с Windows! Магическим словом является Samba: это не Бразильский танец, а реализация протокола SMB для Linux. Смотрите на <http://www.samba.org> и в SMB-HOWTO.

11. Немного программирования

11.1 Скрипты командного процессора: Файлы .BAT на стероидах

Если вы использовали .BAT-файлы для создания сокращений для длинных командных строк (я сам так делал), то это может быть достигнуто вставкой соответствующих алиасов (смотрите вышеприведенный пример) в файлы `profile` или `.bash_profile`. Но если ваши .BAT-файлы были более сложными, то вы полюбите язык, предоставляемый командным процессором: он мощен, как старый QBasic, если не больше. Он имеет переменные, управляющие структуры, подобные `while`, `for`, `case`, `if... then... else`, и много других возможностей: он может быть хорошей альтернативой ``настоящим" языкам программирования.

Для написания скрипта---эквивалента .BAT-файла в DOS все, что надо сделать---написать стандартный ASCII-файл, содержащий инструкции, сохранить его и затем сделать его исполнимым с помощью команды `chmod +x <scriptfile>`. Для выполнения скрипта наберите его имя.

Предупреждение. Системный редактор называется `vi`, и по моему мнению большинство новых пользователей найдет его трудным в использовании. Я не буду объяснять, как его использовать; пожалуйста посмотрите эту информацию в книге Matt Welsh или найдите учебное руководство в сети. Здесь достаточно сказать, что:

- для вставки некоторого текста нажмите `i`, и затем вводите текст;
- для удаления символов наберите `<ESC>`, а затем `x`;
- для выхода из `vi` без сохранения изменений наберите `<ESC>`, а затем `:q!`
- для сохранения и выхода наберите `<ESC>`, а затем `:wq`.

Примечание переводчика: На самом деле vi -- мощный редактор. Для получения встроенного учебника наберите <ESC>, а затем :help tutor, для получения справки по какому либо разделу наберите <ESC>, а затем :help имя_раздела или команды. Этот перевод был сделан используя вариант этого редактора под названием Vim, дополнительную информацию смотрите на <http://www.vim.org>.

Хорошим редактором для начинающих является joe: запустив его с помощью команды jstar, вы получите те же самые сочетания клавиш, как и в редакторе, используемом в DOSWin. jed в режиме WordStar или IDE, даже еще лучше. Пожалуйста посмотрите раздел [Где найти приложения](#) для того чтобы узнать, где можно найти данные редактора.

Написание скриптов для bash -- это такой необъятный предмет, что сам по себе заслуживает книги и я не буду углубляться в этот предмет. Я просто дам вам пример скрипта командного процессора, из которого вы можете вычлениить некоторые базовые правила. Прочитайте справочную страницу bash.

```
#!/bin/sh
# sample.sh
# это комментарий
# не изменяйте первую строку, она должна быть такой
echo "Система называется: `uname -a`" # использовать вывод команды
echo "Мое имя $0" # Встроенные переменные
echo "Вы передали мне следующие $# параметров: "$*
echo "Первым параметром является: "$1
echo -n "Введите ваше имя? " ; read your_name
echo заметьте разницу: "hi $your_name" # использование кавычек вида "
echo заметьте разницу: 'hi $your_name' # использование кавычек вида '
DIRS=0 ; FILES=0
for file in `ls .` ; do
  if [ -d ${file} ] ; then # если файл является каталогом
    DIRS=`expr $DIRS + 1` # DIRS = DIRS + 1
  elif [ -f ${file} ] ; then
    FILES=`expr $FILES + 1`
  fi
  case ${file} in
    *.gif|*.jpg) echo "${file}: графический файл" ;;
    *.txt|*.tex) echo "${file}: текстовый файл" ;;
    *.c|*.f|*.for) echo "${file}: файл исходных текстов" ;;
    *) echo "${file}: обычный файл" ;;
  esac
done
echo "обнаружено ${DIRS} каталогов и ${FILES} файлов"
ls | grep "ZxY--!!!WKW"
if [ $? != 0 ] ; then # код завершения последней команды
  echo "ZxY--!!!WKW не найден"
fi
echo "достаточно... наберите 'man bash' если вам нужна дополнительная информация."
```

11.2 C для вас

В UNIX системным языком является C, любите его или ненавидьте. Также доступны другие языки программирования (Java, FORTRAN, Pascal, Lisp, Basic, Perl, awk...).

Считая, что вы знаете C, существует некоторое количество руководств для тех, кто испорчен использованием Turbo C++ или его собратьев по DOS. Компилятор C в Linux называется gcc и не содержит моргалки и дуделки, которые обычно сопутствуют его аналогам в DOS: нет IDE (Интегрированная среда разработки), оперативной подсказки, встроенного отладчика и т.п. Это просто командно-строковый компилятор, очень мощный и эффективный. Для компиляции стандартного hello.c вам необходимо выполнить команду:

```
$ gcc hello.c
```

которая создаст исполнимый файл, названный a.out. Для того, чтобы дать исполнимому файлу другое название, выполните команду:

```
$ gcc -o hola hello.c
```

Для подключения к программе библиотеки функций добавьте к командной строке ключ `-l<libname>`. Например, для подключения библиотеки математических функций:

```
$ gcc -o mathprog mathprog.c -lm
```

(Ключ `-l<libname>` заставляет `gcc` подключить библиотеку `/usr/lib/lib<libname>.so`; так что `-lm` подключает библиотеку `/usr/lib/libm.so`).

Чем дальше, тем лучше, но когда ваша программа состоит из нескольких файлов исходных текстов, то вам понадобится пользоваться утилитой `make`. Предположим, что вы написали разбор выражений: его файл исходных текстов называется: `parser.c` и подключает два заголовочных файла, `parser.h` и `xy.h`. Затем вы хотите использовать подпрограммы из `parser.c` в программе, допустим `calc.c`, которая подключает файл `parser.h`. Что за неразбериха! Что вам надо сделать для компиляции `calc.c`?

Вам необходимо написать так называемый `Makefile`, который сообщает компилятору зависимости между файлами исходных текстов и объектными файлами. В нашем примере:

```
# Это Makefile, используемый для компиляции calc.c
# Нажмите клавишу <TAB> где показано!

calc: calc.o parser.o
<TAB>gcc -o calc calc.o parser.o -lm
# calc зависит от 2-х объектных файлов: calc.o и parser.o

calc.o: calc.c parser.h
<TAB>gcc -c calc.c
# calc.o зависит от двух файлов исходных текстов

parser.o: parser.c parser.h xy.h
<TAB>gcc -c parser.c
# parser.o зависит от трех файлов исходных текстов

# конец Makefile.
```

Сохраните этот файл как `Makefile` и наберите `make` для компиляции вашей программы; вы также можете сохранить его как `calc.mak` и набрать `make -f calc.mak`, и конечно `RMP`. Вы также можете получить некоторую помощь о функциях `C`, которые описаны в справочных страницах, раздел 3; например,

```
$ man 3 printf
```

Для отладки ваших программ используйте программу `gdb`. Наберите `info gdb` для того, чтобы узнать, как использовать отладчик.

Существует большое количество библиотек; среди них первыми вы скорее всего захотите использовать следующие: `ncurses` (для работы в текстовом режиме), и `svgalib` (для графики в консоли). Много редакторов могут работать как IDE; например `emacs` и `jed`, также они производят подсветку синтаксиса, автоматические отступы, и так далее. Также вы можете взять пакет `rhide c` <http://metalab.unc.edu/pub/Linux/devel/debuggers/>. Это клон Borland IDE и есть шанс, что он вам понравится.

11.3 Программирование для X11

Если вы чувствуете себя достаточно образованным для программирования для X11 (это не так трудно), то существует несколько библиотек, которые сделают написание программ для X11 более легким. Основными серверами, которые вам необходимо посетить, являются сервера `GTK+`, <http://www.gtk.org> и `Qt`, <http://www.troll.no>. `Gtk+` является набором элементов, написанных на языке `C`, в начале созданном для графического пакета `The GIMP` (<http://www.gimp.org>), и используется средой `Gnome`. `Kdevelop` основан на `Qt`, написанном на `C++` и используемом средой `KDE`. Вероятнее всего, вы будете использовать один из этих пакетов.

Наилучшими пакетами для визуального программирования являются Kdevelop для Qt, <http://www.kdevelop.org>, и Glade для GTK+, <http://glade.pn.org>. Эта страница имеет большее количество информации: <http://www.free-soft.org/guitool/>.

Многоплатформенной программирование

Было бы хорошо, если бы вы могли писать код, который одинаково компилируется и под *и* под Windows с помощью `gcc`? Во время написания данного документа, существовало несколько наборов элементов, которые позволяли (более или менее) программирование под разные платформы. Из соображений достаточности и стабильности, может использоваться только один: FLTK, Fast Light Tool Kit <http://www.fltk.org>. Он достаточно небольшой, быстрый и стабильный. В его составе также имеется полу-визуальный построитель интерфейса под названием Fluid.

12. Оставшийся 1% дел

В самом деле больше чем 1%...

12.1 Использование tar и gzip

В UNIX существуют некоторые широко используемые приложения для архивации и сжатия файлов. `tar` используется для создания архивов---он похож на PKZIP или Winzip, но не производит сжатие, он только архивирует. Для создания нового архива используется команда:

```
$ tar -cvf <archive_name.tar> <file> [file...]
```

Для извлечения файлов из архива:

```
$ tar -xpvf <archive_name.tar> [file...]
```

Для перечисления содержимого архива:

```
$ tar -tf <archive_name.tar> | less
```

Вы можете сжать данные, используя программу `compress`, которая устарела и не должна больше использоваться, или с помощью `gzip`:

```
$ compress <file>
$ gzip <file>
```

что создает сжатые файлы с расширением `.Z` (`compress`) или `.gz` (`gzip`). Эти программы могут сжимать только один файл за раз. Для расжатия используются команды:

```
$ compress -d <file.Z>
$ gzip -d <file.gz>
```

RMP.

Также существуют утилиты `unrarj`, `zip` и `unzip` (совместимые с PK??ZIP). Файлы с расширениями `.tar.gz` или `.tgz` (архивированные с помощью `tar`, а затем сжатые `gzip`) являются такой же вещью в UNIX как файлы `.ZIP` в DOS. Вот как выдать список файлов в архиве `.tar.gz`:

```
$ tar -ztf <file.tar.gz> | less
```

12.2 Установка приложений

Во первых: установка приложений -- это работа администратора. Большинство приложений для Linux распространяются в виде архивов `.tar.gz`, которые обычно содержат каталог, названный как и архив, который содержит файлы и/или подкаталоги. Хорошим правилом является установка этих программ из каталога `/usr/local` с помощью команды

```
# tar -zxvf <archive.tar.gz>
```

прочитайте файл `README` или `INSTALL`. В большинстве случаев приложение распространяется в виде исходных текстов, которые вам надо скомпилировать; часто достаточно набрать команду `make`, а затем `make install`. Если в архиве есть файл `configure`, то сначала запустите его. Вам также необходимо иметь компилятор `gcc` или `g++`.

Другие архивы должны быть распакованы, находясь в каталоге `/`; это в случае с архивами, используемыми в Slackware. Другие архивы содержат только файлы, без подкаталогов---будьте осторожны, не запутайтесь! Всегда просмотрите список файлов в архиве до его установки.

Debian и Red Hat имеют свой собственный формат архивов -- соответственно `.deb` и `.rpm`. Последний приобретает широкое распространение; для установки пакета `rpm` наберите

```
# rpm -i package.rpm
```

12.3 Советы, без которых вы не сможете жить

Прокрутка назад: нажимая `<SHIFT + PAG UP>` (серая клавиша) вы можете прокрутить назад несколько последних страниц экрана, в зависимости от того, сколько у вас видеопамати.

Сброс экрана: если вы выполнили команду `more` или `cat` для двоичного файла, то ваш экран может быть заполнен мусором. Для того, чтобы исправить это вслепую, наберите `reset` или такую последовательность символов: `echo CTRL-V ESC c RETURN`.

Вставка текста: Как сделать это в консоли, описано ниже; в X нажмите левую кнопку мыши и держа ее выделите необходимый текст, затем щелкните средней кнопкой мыши (или обоими кнопками, если у вас 2-х кнопочная мышь) для вставки текста в позиции курсора. Также есть приложение `xclipboard` (только для текста); не пугайтесь ее медленной работы.

Используя мышшь: если у вас установлена программа `gpm`---драйвер мыши для консоли, то вы можете нажать на левую кнопку мыши и выделить текст, а затем нажать на правую кнопку мыши для вставки выделенного текста. Программа может работать между разными виртуальными консолями.

Сообщения от ядра: войдя под правами администратора посмотрите файл `/var/adm/messages` или `/var/log/messages` для того чтобы увидеть, что ядро сообщает вам, включая сообщения на этапе загрузки. Также удобна команда `dmesg`.

12.4 Где найти приложения

Если вы удивляетесь, должны ли вы заменить ваши старые и привычные приложения под DOS/Win приложениями под Linux, я предполагаю, что вы просмотрели основные архивы программного обеспечения для Linux: <ftp://metalab.unc.edu/pub/Linux>, <ftp://tsx-11.mit.edu/pub/linux>, и <ftp://ftp.funet.fi/pub/Linux>. Другим хорошим местом для поиска является ``Linux Applications and Utilities Page (Страница приложений и утилит для Linux)" <http://www.xnet.com/~blatura/linapps.shtml>, и ``официальную" страницу Linux <http://www.linux.org>.

12.5 Некоторые вещи, которые вы не могли делать

Linux может выполнять ужасно много вещей, которые были громоздкими, трудными или невозможными в DOS/Windows. Вот короткий список вещей, которые могут возбудит ваш аппетит:

- `at` позволяет вам запускать программы в указанное время;
- `awk` -- это простой, но мощный язык для манипулирования файлами данных (и не только). Например, допустим, что `data.dat` -- это ваш файл с множеством полей данных,

- ```
$ awk '$2 ~ "abc" \{print $1, "\\t", $4\}' data.dat
```

напечатает поля 1 и 4 из каждой строки файла `data.dat`, второе поле которой содержит ``abc".

- `cron` полезен для выполнения периодических задач, в указанное время и дату. Наберите `man 5 crontab`.
- `file <filename>` скажет вам, чем является `filename` (ASCII текст, исполнимый файл, архив и т.п.);
- `find` (смотрите также раздел [Каталоги: Перевод команд](#))---это одна из наиболее мощных и полезных команд. Она используется для поиска файлов, которые отвечают нескольким заданным критериям и для выполнения действий над этими файлами. В общем виде командная строка `find` выглядит так:

- ```
$ find <каталог> <выражение>
```

где `<выражение>` включает в себя критерии поиска и действия над найденными файлами. Примеры:

```
$ find . -type l -exec ls -l \{\} \;
```

находит все файлы, которые являются символическими ссылками, и показывает, на какие файлы они указывают.

```
$ find / -name "*.old" -ok rm \{\} \;
```

находит все файлы, соответствующие заданному образцу, и удаляет их, сначала спрашивая у вас разрешение.

```
$ find . -perm +111
```

находит все файлы, чьи права доступа соответствуют 111 (исполнимые).

```
$ find . -user root
```

находит все файлы, принадлежащие пользователю `root`. Здесь есть много возможностей---RMP.

- `grep` находит заданный текст в файлах. Например,

- `$ grep -l "geology" *.tex`

перечислит все файлы `*.tex`, в которых содержится слово `geology`. Вариант этой программы под названием `zgrep` работает со сжатыми файлами. RMP;

- **регулярные выражения**---являются сложным, но мощным способом выполнения операций поиска в тексте. Например, `^a[^a-m]X\{4,\}txt$` соответствует строке, которая начинается на `a`, за которым следует любой символ, за исключением входящих в интервал `a-m`, за которым следует 4 или больше символов `X`, и заканчивается на `txt`. Вы можете использовать регулярные выражения в мощных редакторах, программе `less`, и многих других приложениях. Для введения в регулярные выражения прочитайте справочную страницу для `grep`.
- команда `script <script_file>` выдаст вам на экран содержимое `script_file`, до тех пор пока вы не выполните команду `exit`. Полезно для отладки;
- `sudo` позволяет пользователям выполнять некоторые обязанности администратора (например форматирование и монтирование дисков; RMP);
- `uname -a` выдаст вам информацию о вашей системе;
- `zcat` и `zless` полезны для просмотра и печати сжатых файл без их расжатия. Например:
 - `$ zless textfile.gz`
 - `$ zcat textfile.gz | lpr`
- Следующие команды часто бывают полезны: `bc`, `cal`, `chsh`, `cmp`, `cut`, `fmt`, `head`, `hexdump`, `nl`, `passwd`, `printf`, `sort`, `split`, `strings`, `tac`, `tail`, `tee`, `touch`, `uniq`, `w`, `wall`, `wc`, `whereis`, `write`, `xargs`, `znew`. RMP.

12.6 Использование утилит для UNIX в DOS/Windows

Верьте этому или нет, но существуют утилиты, которые обеспечивают UNIX-подобную среду в DOS/Windows! Одной из них является набор Djgpp (<http://www.delorie.com/djgpp>) для DOS, в то время как Cygnus (<http://www.cygnus.com>) является более сложным портом для Win32. Они оба включают в себя те же средства разработки GNU и утилиты, как и Linux; хотя вы не получите той же стабильности и производительности.

Если вы хотите почувствовать вкус Linux, попробуйте пакет Djgpp. Скачайте и установите следующий файлы (во время написания последней версии была версия 2.02): `djdev202.zip`, `bnu281b.zip`, `bsh1147b.zip`, `fil316b.zip`, `find41b.zip`, `grep22b.zip`, `gwk303b.zip`, `lss332b.zip`, `sh1112b.zip`.. Инструкции по установке включены в дистрибутив и вы можете получить помощь в [news:comp.os.msdos.djgpp](http://news.comp.os.msdos.djgpp).

В частности, использование `bash` в DOSWin является глотком свежего воздуха. Для его правильной настройки отредактируйте поставляемый файл `boot.bat` для того, чтобы он соответствовал вашим настройкам, а затем поместите эти файлы в ваш домашний каталог (на разделе Windows) вместо тех, которые поставляются в дистрибутиве:

```

# это _bashrc

LS_OPTIONS="-F -s --color=yes"
alias cp='cp -i'
alias d='ls -l'
alias l=less
alias ls="ls $LS_OPTIONS"
alias mv='mv -i'
alias rm='rm -i'
alias u='cd ..'
# это bprof
if [ -f ~/.bashrc ]; then
. ~/.bashrc
fi
PS1='\w\$ '
PS2='> '
CDPATH="$CDPATH:~"
# stuff for less(1)
LESS="-M-Q" # long prompt, silent
LESSEEDIT="%E ?!t+%!t. %f" # edit top line
VISUAL="jed" # editor
LESSCHARSET=latin1 # visualise accented letters
export PS1 PS2 CDPATH LS_OPTIONS LESS LESSEEDIT LESSOPEN VISUAL LESSCHARSET

```

12.7 Общие расширения файлов и соответствующие программы

Вы можете обнаружить множество расширений файлов. За исключением экзотических расширений (например, шрифты и т.п.), вот список того, для чего какие расширения файлов используются:

- 1 ... 8: справочные страницы. Читайте их с помощью команды `groff -Tascii -man <file.1>`.
- arj: архивы, созданные программой arj.
- dvi: выходной файл, созданный программой TeX (смотрите ниже). `xdvi` используется для его показа; а `dvips --` для преобразования его в файл PostScript `.ps`.
- gz: архивы, созданные `gzip`.
- info: файл для программы `info file` (альтернатива справочным страницам). Возьмите пакет `texinfo`.
- lsm: Файл Linux Software Map (Карты программного обеспечения Linux). Это обычный ASCII-файл, содержащий описание пакета программ.
- ps: Файл PostScript. Для его просмотра или печати возьмите программу `gs` и, при необходимости, программу `ghostview` или `gv`.
- rpm: Пакет Red Hat. Вы можете установить его на любой системе, используя менеджер пакетов `rpm`.
- taz, tar.Z: архив, созданный `tar` и сжатый программой `compress`.
- tgz, tar.gz: архив, созданный `tar` и сжатый `gzip`.

- `tex`: текстовый файл с информацией для TeX, мощной системы форматирования текста. Возьмите пакет `tex`, доступный в многих дистрибутивах.
- `texi`: файл `texinfo`, вы можете сделать из него и файлы TeX и файлы `info` (`info`). возьмите пакет `texinfo`.
- `xbm`, `xpm`, `xwd`: графические файлы.
- `z`: архив, созданный программой `compress`.

12.8 Преобразование файлов

Если вам необходимо обмениваться текстовыми файлами между DOS/Win и Linux, то будьте осторожны с проблемой ``конца строки". В DOS каждая строка оканчивается символами CR/LF (ASCII 13 + ASCII 10), а в Linux -- символом LF. Если вы редактируете текстовый файл из DOS в Linux, то каждая строка будет оканчиваться странно-выглядящим символом `M'; а текстовый файл из Linux в DOS появится как огромная одиночная строка без разбиения на параграфы. Существуют утилиты `dos2unix` и `unix2dos` для преобразования файлов.

Если ваш текстовый файл содержит символы с диакритическими знаками, то убедитесь, что они набраны в Windows (в Notepad), а не в чистом DOS; в ином случае символы будут потеряны.

Преобразование файлов Word или WordPerfect в текстовые файлы сложнее, но возможно. Вам необходима одна из утилит, которую можно найти на серверах CTAN; один из них <ftp://ftp.tex.ac.uk>. Возьмите программу `word2x` из каталога `/pub/tex/tools/`, или попробуйте программы, находящиеся в каталоге `/pub/tex/support/`. В частности `word2x` преобразует файлы Word 6, тогда как для файлов Word 97 вам необходима программа `mwordview` (<http://www.csn.ul.ie/~caolan/docs/MSWordView.html>), которая преобразует эти файлы в HTML.

12.9 Свободные офисные пакеты

Если преобразования файлов вам недостаточно, то вы можете вцепиться зубами в (свободный!) Microsoft Office--подобный пакет.

В настоящее время StarOffice (<http://www.sun.com/staroffice>) является единственным выбором, хотя Koffice развивается (<http://www.koffice.org>). Он большой, иногда медленен, но в любом случае хорош: Он предоставляет функциональные возможности, которых нет в Microsoft Office. Он также может читать и записывать файлы Word и Excel, хотя преобразования не всегда нормальны.

Другим хорошим пакетом является Corel WordPerfect, свободная версия которой доступна для загрузки. Должен я еще что-нибудь сказать? Скачать его можно с <http://www.corel.com>.

13. Конец, пока что

Поздравляю! Вы теперь охватили небольшой кусочек UNIX и готовы начать работать. Помните, что ваше знание системы все еще ограничено, и что вы будете больше практиковаться в работе с Linux для того, чтобы работа с ним стала комфортабельной. Но если все, что вам нужно было---это получить набор приложений и начать работать с ними, то той информации, которую я включил в этот документ, будет достаточно.

Я уверен, что вы насладитесь работой с Linux и будете продолжать изучать его. Я бьюсь об заклад, что вы никогда не вернетесь к DOSWin! Я надеюсь, что я был понятен и дал хорошую информацию для моих 3-х или 4-х читателей.

13.1 Авторские права

Пока не будет указано что-то другое, документы серии Linux HOWTO принадлежат соответствующим авторам. Документы Linux HOWTO могут воспроизводиться и распространяться все вместе или частями, на любом носителе физическом или электронном, с сохранением этого замечания об авторских правах на всех копиях. Коммерческое распространение разрешено; однако авторы хотят быть предупреждены о любых таких распространениях.

Все переводы, производные работы или работы, включающие любой документ серии Linux HOWTO, должны находиться под действием этого замечания об авторских правах. Таким образом, вы не можете создать производную работу из HOWTO и наложить на нее дополнительные ограничения на ее распространение. Исключения из этих правил могут быть достигнуты при определенных условиях; пожалуйста, свяжитесь с координатором проекта Linux HOWTO по адресу, данному ниже.

Вкратце, мы хотим способствовать распространению этой информации через максимально возможное количество каналов. Однако мы хотим оставить авторские права на документы HOWTO, и хотим быть уведомлены о любых планах распространения документов серии HOWTO.

Если вы имеете вопросы, пожалуйста, свяжитесь с Tim Wynn, координатором Linux HOWTO, используя электронную почту linux-howto@metalab.unc.edu.

13.2 Отказ от ответственности

``From DOS to Linux HOWTO (Из DOS в Linux HOWTO)" был написан Guido Gonzato, REMOVE_MEguido@ibogeo.df.unibo.it (удалите слово ``REMOVE_ME"). Большое спасибо Matt Welsh, автору ``Linux Installation and Getting Started (Установка Linux и начало работы)", Ian Jackson, автору ``Linux frequently asked questions with answers (Часто задаваемые вопросы о Linux с ответами)", Giuseppe Zanetti, автору ``Linux", всем людям, которые прислали мне предложения, и особенно Linus Torvalds и проекту GNU, которые подарили нам Linux.

Этот документ распространяется ``as is (как есть)". Я приложил огромные усилия, чтобы написать его настолько аккуратно, насколько это возможно, но вы должны использовать информацию содержащуюся в нем, на свой собственный риск. Я не несу никакой ответственности за любой урон, нанесенный в результате использования этой работы.

Обратная связь приветствуется. Чувствуйте себя свободно при контакте со мной для любых запросов, предложений, флейма и т.п.

Наслаждайтесь Linux и жизнью,

Guido =8-)