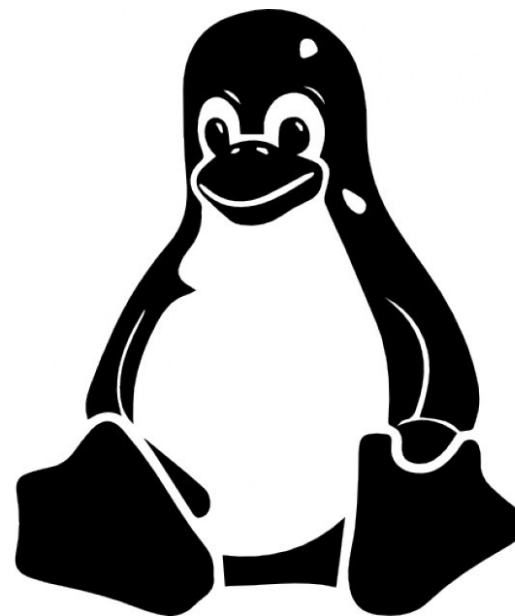


Команды для Linux терминала  
Оболочка Bash



Разраб: Трунин Е.Е.

v 0.1

Краткий конспект книги С.Граннемана Необходимый код и команды. Карманный справочник Linux.

## Содержание

Глава I — Простейшие команды

Глава I I — Объединение команд

Глава III — Потоки ввода-вывода

Глава IV — Отображение содержимого файлов

Глава V — Вывод на печать

Глава VI — Владельцы файлов и права доступа

Глава VII — Создание архивов и сжатие данных

Глава VIII — Поиск данных

Глава IX — Работа с оболочкой

Глава X — Контроль использования системных ресурсов

Глава XI — Инсталяция программного обеспечения

Глава XII — Сетевое взаимодействие

Глава XIII — Работа в сети

///  
//Простейшие//команды//  
//Глава/I//

//LS//  
ls [путь\_к\_каталогу] = выводит список файлов и каталогов  
в данной директории  
Параметры :

- R = Просмотр содержимого подкаталогов
- 1 = Вывод содержимого каталога в один столбец
- m = Вывод содержимого Kanfkjuf с запятыми в качестве разделителей
- a = Отображение скрытых файлов и каталогов
- F = Отображение информации о типах файлов
- l = Информации о правах доступа и владельцах файлов
- r = ВЫВОД информации В обратном порядке
- x = сортировка содержимого каталога по суффиксам имен файлов
- t = сортировка по дате и времени
- s = сортировка содержимого по размеру файлов
- h = предоставление размеров файлов в килобайтах, мегабайтах или гигабайтах

//PWD//  
pwd = определение пути к текущему каталогу

//CD//  
cd [путь\_к\_каталогу] = переход к другому каталогу  
Параметры :  
-(~) = переход в рабочий каталог (cd ~)  
-(-) = переход к предыдущему каталогу (1)

//TOUCH//  
touch [имя\_файла\_для\_изменения\_времени] = изменение сведений о времени файла  
Параметры :  
-a = изменение времени доступа  
-m = изменение времени модификации  
-t [[CC]YY]MMDDhhmm[.ss]= установка произвольного времени для файла (touch -t 197002160701 name.log)

//MKDIR//  
mkdir [имя\_нового\_каталога] = создание нового каталога  
Параметры :  
-p = создание нового каталога и необходимых подкаталогов (mkdir -p pictures/personal/family)  
-v = информация о действиях выполняемых командой mkdir

//CP//  
cp [имя\_файла\_для\_копирования] [имя\_нового\_файла] = копирование файлов  
Параметры :  
-v = вывод подробной информации о копировании файла  
-i = предотвращение копирования файлов поверх существующих  
-R = копирование каталогов  
-a = создание резервных копий

//RM//  
rm [имя\_файла\_для\_удаления] = удаление файлов  
Параметры :  
-v = вывод подробной информации при удалении файлов  
-i = предотвращение удаления важных файлов  
-Rf = удаление файлов и каталогов содержащих данные (2)

/////////////////////////////////MV////////////////////////////////////

mv [путь\_и\_имя\_файла\_копирования]  
[путь\_куда\_копируем\_и\_имя\_файла]= перемещение и  
переименование файлов

/////////////////////////////////RMDIR////////////////////////////////////

rmdir [имя\_пустого\_каталога] = удаление пустого  
каталога

/////////////////////////////////SU////////////////////////////////////

su [имя\_пользователя] = позволяет выполнять действия  
от имени другого пользователя

Параметры :

-l = использование не только имени другого пользователя,  
но и его переменное окружение

/////////////////////////////////WHOAMI////////////////////////////////////

whoami = сообщает имя пользователя с точки зрения  
оболочки

/////////////////////////////////MAN////////////////////////////////////

man [название\_команды] = получение информации о  
команде

Параметры :

- k = поиск команды по выполняемым ею действиям
- f = получение кратких сведений о команде
- u = формирование базы данных команд
- e = вывести справочную информацию на печать

/////////////////////////////////INFO////////////////////////////////////

info [название\_команды] = получение информации о  
команде, организованная в виде разделов

(3)

/////////////////////////////////WHEREIS////////////////////////////////////

whereis [название\_команды\_или\_файла] = определяет путь к исполняемому,  
исходным файлам и страницам справочного руководства

Параметры :

- b = путь только к исполняемым файлам
- m = путь лишь к страницам справочного руководства
- s = путь к исходным файлам

/////////////////////////////////WHATIS////////////////////////////////////

whatis [название\_команды] = выводит сведения о команде (аналог man  
-f)

Параметры :

- w = поиск с применением групповых операций
- r = для работы с регуляторными выражениями

/////////////////////////////////APROPOS////////////////////////////////////

apropos [название\_команды] = поиск информации о команде по  
выполняемым ею действиям (аналог man -k)

Параметры :

- w = поиск с применением групповых операций
- r = для работы с регуляторными выражениями
- e = для точного совпадения слова или фразы при поиске

/////////////////////////////////WHICH////////////////////////////////////

which [название\_файла] = показывает местоположение файла который  
будет запущен командой из строки

(4)

//  
//Объединение/команд//  
//Глава/II//

Чтобы выполнить последовательно несколько команд, необходимо использовать символ (;)

Пр: Команда\_1 ; Команда\_2 ; Команда\_3 ...  
При этом вначале выполняется Команда\_1, затем последующие.

Чтобы выполнить последовательно несколько команд с проверкой на их корректность(смогут ли они выполниться), необходимо разделять команды символом (&&)

Пр: Команда\_1 && Команда\_2 && Команда\_3 ...  
При этом команды также выполняются последовательно, но управление следующей команде переходит только в случае, если предыдущая завершилась без ошибки.

Чтобы выполнить последовательно несколько команд с проверкой на их корректность(смогут ли они выполниться), необходимо разделять команды символом (&&)

Пр: Команда\_1 && Команда\_2 && Команда\_3 ...  
При этом команды также выполняются последовательно, но управление следующей команде переходит только в случае, если предыдущая завершилась без ошибки.

Чтобы выполнить команду при условии, что предыдущая завершилась ишибкой, используется символ (||)

Пр: Команда\_1 || Команда\_2 (5)

При этом Команда\_2 выполнится только при условии, что не выпонилась Команда\_1

Использование выходных данных одной команды при вызове другой команды осуществляется с помощью символов (\$())

Пр: Команда\_1 \$(Команда\_2)  
Пример : mkdir \$(date "+%Y-%m-%d")

//  
//Потоки//ввода-вывода//  
//Глава/III//

Идентификатор	Название	Сокращение	Использование
0	Стандартный входной поток	stdin	клавиатура
1	Стандартный выходной поток	stdout	терминал
2	Стандартный поток ошибок	stderr	терминал

Для передачи выходных данных одной команды на вход другой команды используется символ (|)

Пр: Команда\_1 | Команда\_2  
При этом Команда\_1 будет выступать в качестве источника данных, которые передаются на вход Команда\_2.  
Пример : ls -l | less  
(6)

Для перенаправления выходных данных в файл используется символ (>)

Пр: Команда\_1 > имя\_файла

При этом выходные данные команды Команда\_1 будут записаны в файл имя\_файла

Пример : ls -lF > datafiles.txt

Чтобы предотвратить перезапись файла при перенаправлении вывода, нужно использовать команду noclobber

Она вызывается так : set -o noclobber

После её вызова, перенаправление в существующий файл будет производиться только с нашего разрешения.

Чтобы выключить функцию : set +o noclobber

Чтобы перенаправить выходные данные и записать их в конец файла используются символы (>>)

Команда\_1 >> имя\_файла

При этом выходные данные будут записаны в конец файла, если файла не существует, он будет создан. В отличие от (>) файл не будет каждый раз перезаписываться, а пополняться.

Чтобы использовать содержимое файла в качестве входных данных используется символ (<)

Команда\_1 < имя\_файла

При этом содержимое файла будет выступать в качестве входных данных для команды Команда\_1

```
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////Отображение//содержимого//файлов////////////////////////////////////  
////////////////////////////////////Глава//IV////////////////////////////////////  
(7)
```

```
////////////////////////////////////CAT////////////////////////////////////
```

cat [имя\_файла] = выводит содержимое файла в консоль, а затем возвращает управление оболочке

cat [имя\_файла1] [имя\_файла2] [имя\_файла3 ...] = выводит содержимое сразу нескольких файлов без разделителей.

cat [имя\_файла1] [имя\_файла2] > [имя\_файла3] = объединяет 1 и 2 файл и сохраняет их в файл имя\_файла3

Параметры :

-n = выводит содержимое файлов с нумерацией строк

```
////////////////////////////////////LESS////////////////////////////////////
```

less [имя\_файла] = организует постраничный вывод текста

Имеет множество параметров и комбинаций для внутреннего управления, которые я в данном руководстве рассматривать не буду. Используйте man или info.

```
////////////////////////////////////HEAD////////////////////////////////////
```

head [имя\_файла] = просмотр первых десяти строк файла

head [имя\_файла1] [имя\_файла2] = просмотр первых десяти строк файла1 и файла2

Параметры :

-n [число\_строк] = позволяет отображать указанное количество строк файла

-с [число\_байтов] = позволяет отобразить указанное количество байтов файла

```
////////////////////////////////////TAIL////////////////////////////////////
```

tail [имя\_файла] = просмотр последних десяти строк файла

Поддерживает все функции head.

Параметры : (8)

-f = не завершает функцию и постоянно отображает последние строки файла, позволяя видеть последние изменения.

////////////////////////////////////  
////////////////////////////////////Вывод//на//печать////////////////////////////////////  
////////////////////////////////////Глава//V////////////////////////////////////

////////////////////////////////////LPSTAT////////////////////////////////////  
lpstat -p = получение списка доступных принтеров  
lpstat -d = определение принтера установленного по умолчанию  
lpstat -s = определение расположения принтеров и принтера по умолчанию  
lpstat -t = получение полной информации о принтерах

////////////////////////////////////LPR////////////////////////////////////  
lpr [имя\_файла\_на\_печать] = выводит указанный файл на печать. Для распечатки можно использовать только форматы: txt, PDF, PostScript.

lpr -p [имя\_принтера] [имя\_файла\_на\_печать] = вывод печати на указанный принтер

lpr -# [число\_копий] [имя\_файла\_на\_печать] = распечатка указанного числа копий

lprq = получение списка заданий на печать

////////////////////////////////////LPRM////////////////////////////////////  
lprm = отмена задания, переданного на принтер по умолчанию (9)

lprm [идентификатор\_задания] = отмена задания переданного на произвольный принтер

lprm - = отмена всех заданий на печать

////////////////////////////////////  
////////////////////////////////////Владельцы//файлов//и//права//доступа////////////////////////////////////  
////////////////////////////////////Глава//VI////////////////////////////////////

d|rwx|r-x|r-x| = разделение информации о правах доступа

- 1) обозначение типа файл (- = обычный каталог  
- = исполняемый файл  
d = каталог  
l = символьная ссылка  
s = сокет  
b = блочное устройство  
c = символьное устройство  
p = именованный канал)
- 2) права владельца файла  
3) права группы  
4) права любого другого пользователя

Категории пользователей и их обозначения

Категория пользователей	Сокращение
Владелец	u
Группа	g
Прочие пользователи	o

### Атрибуты, определяющие доступ

Атрибут	Сокращение	Значение для файла	Значение для каталога
Чтение	r	Можно читать	Можно просматривать файл
Запись	w	Можно редактировать	Можно удалять, переименовывать или добавлять новые файлы
Выполнение	x	Можно запускать на выполнение	Можно читать файлы и каталоги и выполнять файлы
suid	s	Любой пользователь может запустить файл на выполнение с правами владельца	Не применяется
sgid	s	Любой пользователь может запустить файл на выполнение с правами группы	Все файлы, вновь созданные в каталоге, принадлежат группе, владеющей каталогом (11)

"sticky bit"	t	Сообщает системе о том, что файл часто используется и должен находиться в области подкачки для быстрого обращения	Удалять или переименовывать файлы, находящиеся в каталоге имеют право только их владельцы или владелец каталога
--------------	---	---	---

cat /etc/passwd = выводит имена существующих пользователей системы, а также группы в которые они входят

////////////////////////////////CHGRP////////////////////////////////  
 chgrp [имя\_каталога\_или\_файла] = изменение групп доступа для файлов и каталогов

Параметры :

- R = рекурсивное изменение принадлежности каталога к группе
- v = получить сведения о действиях команды chgrp
- c = получить сведения только о реальных изменениях командой chgrp

////////////////////////////////CHOWN////////////////////////////////  
 chown [имя\_нового\_владельца] [имя\_файла] = изменение владельцев файлов или каталогов  
 chown [имя\_нового\_владельца:имя\_новой\_группы имя\_файла] = изменение одновременно и владельца и группы для файлов и каталогов



chmod [ugo][+|=][rwx] [имя\_файла] = изменение прав доступа к файлам и каталогам с использованием символьных обозначений

Пример : chmod g+w libby.jpg

chmod [0-7][0-7][0-7] [имя\_файла] = изменение прав доступа к файлам и каталогам с использованием числовых обозначений

Пояснйлочка: здесь 0 = ---  
1 = --x  
2 = -w-  
3 = -wx  
4 = r--  
5 = r-x  
6 = rw-  
7 = rwx

Пример : chmod 664 libby.jpg

Параметры :

-R = рекурсивное изменение прав (к нескольким файлам одновременно)

!Если ввести chmod 000, то производить какие-либо действия с файлом или каталогом можно будет только от имени пользователя root

chmod u[+|=]s [имя\_файла] = установка и сброс suid  
Установленный признак suid означает, что пользователь может запустить файл на выполнение с правами пользователя, владеющего им, т.е. так, как будто этот файл был выполнен самим владельцем

chmod g[+|=]s [имя\_файла] = установка и сброс признака sgid  
(13)

Признак sgid похож на suid, однако он применим не только к файлам, но и к каталогам. Для файлов признак sgid действует так же, как suid, за исключением того, что пользователь запускает файл на выполнение не справями владельца, а справями группы.

chmod [+|=]t [имя\_файла] = установка и сброс признака "sticky bit"

/////////////////////////////////  
////////////////////////////////Создание//архивов//и//сжатие//данных/////////////////////////////////  
////////////////////////////////Глава//VII////////////////////////////////

zip - самый популярный инструмент для архивирования и сжатия на всех платформах.

gzip - имеется практически на каждом UNIX компьютере, но нет на Mac и Windows.

bzip2 - инструмент являющийся лучшим аналогом gzip, сжимает сильнее, но дольше.

tar - инструмент предназначенный только для архивирования. Очень популярен на серверах

////////////////////////////////ZIP/////////////////////////////////  
zip [новое\_имя\_файла].zip [имя\_файла\_который\_будет\_сжат] =  
команда для архивации и сжатия файлов

Параметры :

-[0-9] = указываем коэффициент сжатия от 0 до 9

-r [пароль] = защита архива паролем (не рекомендуем)

-e = шифрует архив и требует ввода пароля (рекомендуем)

(14)

unzip [имя\_архива].zip = разархивирует zip архив

Параметры :

- v = показывает подробную информацию о разархивировании
- l = показывает содержимое архива прямо в командной строке
- t = проверка файлов архива на ошибки разархивации

////////////////////////////////GZIP////////////////////////////////

gzip [имя\_сжимаемого\_файла] = команда которая удаляет файл и оставляет только сжатую копию его.

Параметры :

- с = не удаляет файл, который подлежит сжатию, а выводит в стандартный выходной поток. Для сохранения в файл нужно использовать перенаправление. Пример : gzip -с book.pdf > bookg.gz
- r = рекурсивная обработка файлов посредством программы gzip
- [0-9] = задание коэффициента сжатия.

gunzip [имя\_архива].gz = распаковка файлов, сжатых посредством gzip. Также удаляет архив, оставляя один распакованный файл.

Параметры :

- с = не удаляет файл, который подлежит сжатию, а выводит в стандартный выходной поток. Для сохранения в файл нужно использовать перенаправление. Пример : gzip -с book.pdf > bookg.gz
- t = проверка файлов архива на ошибки разархивации

(15)

////////////////////////////////BZIP2////////////////////////////////

bzip2 [имя\_сжимаемого\_файла] = также как и gzip удаляет файл, оставляя лишь сжатый архив.

Параметры :

- с = не удаляет файл, который подлежит сжатию, а выводит в стандартный выходной поток. Для сохранения в файл нужно использовать перенаправление. Пример : bzip2 -с book.pdf > bookg.bz2
- [0-9] = задание коэффициента сжатия

bunzip2 [имя\_архива].bz2 = команда для разархивации файлов .bz2

Параметры :

- с = не удаляет файл, который подлежит сжатию, а выводит в стандартный выходной поток. Для сохранения в файл нужно использовать перенаправление. Пример : gzip -с book.pdf > bookg.gz
- t = проверка файлов архива на ошибки разархивации

////////////////////////////////TAR////////////////////////////////

tar -cf [имя\_архива].tar [имена\_файлов\_для\_архива или название\_каталога] = tar только объединяет файлы в архив.

////////////////////////////////Поиск//данных////////////////////////////////  
////////////////////////////////Глава/VIII////////////////////////////////

////////////////////////////////LOCATE////////////////////////////////

locate [имя\_файла] = ищет файлы и каталоги по заданному имени на основании базы данных, содержащей имена файлов и каталогов.

(16)

Параметры :

-i = поиск в базе имен файлов без учета регистра.

-n [количество\_выводимых\_результатов] = указываем кол-во результатов

slocate [имя\_файла] = не ищет файлов в тех каталогах, которые не доступны данному пользователю.

updatedb = обновляет базу данных, для программы locate

//////////////////////////////////GREP//////////////////////////////////

grep [ключевое\_слово] [имя\_файла\_для\_поиска] = поиск фрагмента в заданном файле. В качестве файла можно ввести \* и поиск будет произведен во всех файлах. Данная команда имеет достаточно гибкое устройство поиска, т.к имеет возможность использовать регуляторные выражения. В данном пособии регуляторные выражения рассмотрены не будут.

Параметры :

-R = рекурсивный поиск фрагментов текста в файлах

-i = поиск фрагментов текста без учета регистра букв

-w = поиск слов в файлах (производит поиск конкретного слова в файлах, без учета похожих слов)

-n = отображение номеров строк

-A = отображения последующих элементов контекста

-B = отображение предыдущих элементов контекста

-C = отображение всех элементов контекста

-v = отображение строк, не содержащих указанных слов

-l = отображение списка файлов, содержащих указанное слово

(17)

Поиск слов в выходных данных других команд

Пример : ls -l | grep 1960 (из данного списка команды ls будут выбраны только те файлы, которые содержат число 1960 в названии)

Поиск слов в результатах поиска

Пример : ls -l | grep 196[6-7] | grep -v live (указывает список файлов выдаваемых функцией ls содержащей цифры 1966 или 1967 и не содержащих слова live.

//////////////////////////////////FIND//////////////////////////////////

find [путь\_к\_каталогу\_поиска] [команда] [название\_файла]= команда также производящая поиск данных на компьютере, но в отличие от locate она непосредственно ищет файлы по заданным условиям.

Параметры :

-name = поиск файлов по имени или его части

-user = поиск файлов принадлежащих пользователю

-group = поиск файлов принадлежащих группе

-size [размер\_файла]= поиск файлов по размеру. В конце необходимо указать приставку размерности ( b - 512 байтовые блоки

c - байты

k - килобайты

M - мегабайты

G - гигабайты)

-type [знак\_типа]= поиск файлов по типу. В конце необходимо указать приставку типа файла ( f - обычный каталог

d - каталог

l - символьная ссылка

b - специальный файл блочног типа

c - специальный файл символьного типа

p — FIFO (18)

s - сокет)

-a = (and) позволяет объединить несколько условий для поиска

-o = (or) отображение результатов всех в отдельности условий поиска

-n = (not) отображение результатов не соответствующих условиям поиска

-exec = выполнение действий над каждым найденным файлом

-fprint [имя\_файла\_для\_вывода\_данных]= вывод результатов поиска в файл

```
////////////////////////////////////  
////////////////////////////////////Работа//с//оболочкой////////////////////////////////////  
////////////////////////////////////Глава//IX////////////////////////////////////
```

```
////////////////////////////////////HISTORY////////////////////////////////////
```

history = просмотр списка команд введенных ранее.  
Сохраняются последние 500 команд.

!! = выполнение последней команды

![номер\_команды] = выполнение предыдущей команды  
путем указания её номера

![название\_команды] = выполняет команду из истории,  
путем указания её названия

Также стрелочками вверх и вниз на клавиатуре, можно  
выбирать команды из списка history

```
////////////////////////////////////ALIAS////////////////////////////////////  
alias = отображение псевдонимов команд (19)
```

alias [имя\_псевдонима] = просмотр команды определенной данным псевдонимом

alias [псевдоним] = '[команда]' = создание временного псевдонима и присвоение ему определенной команды. Псевдоним, созданный таким способом действует лишь до завершения сеанса работы с оболочкой.

alias [псевдоним] = '[команда]' = для создания постоянного псевдонима, необходимо вписать такую команду в файл .bashrc

unalias [псевдоним] = удаление временных псевдонимов

```
////////////////////////////////////  
////////////////////////////////////Контроль//использования//системных//ресурсов////////////////////////////////////  
////////////////////////////////////Глава//X////////////////////////////////////
```

```
////////////////////////////////////PS////////////////////////////////////
```

ps aux = вывод информации о процессах, выполняемых в системе. В столбце STAT содержится несколько различных букв. Вот основные из них  
( R = выполняется

S = спит

T = остановлен

z = зомби(завис))

ps axjf = просмотр дерева процессов

ps [имя\_пользователя] = отображение процессов, принадлежащих конкретному пользователю

(20)

/////////////////////////////////KILL////////////////////////////////

kill [номер\_процесса] = завершение выполняющегося процесса. Номер процесса можно найти используя команду ps aux.

Параметры :

-1 == -HUP = прекращение выполнения процесса  
-15 == -TERM = "мягкое завершение"  
-9 == -KILL = прекратить все выполняющиеся действия и завершить работу

/////////////////////////////////TOP////////////////////////////////

top = предоставляет динамические данные о происходящих процессах в реальном времени. Чтобы завершить процесс из top, необходимо нажать клавишу k, затем выбрать номер процесса.

/////////////////////////////////LSOF////////////////////////////////

lsf = получение списка открытых файлов. Т.к в UNIX системах все является файлом, данная команда выведет тысячи открытых файлов.

Параметры :

-u [имя\_пользователя] = отображение файлов открытых конкретным пользователем.  
-c [имя\_программы] = отображение сведений о процессах, соответствующих конкретной программе.

lsf [имя\_файла] = отображение имени пользователя, открывшего определенный файл

(21)

/////////////////////////////////FREE////////////////////////////////

free = отображение информации об оперативной памяти системы

/////////////////////////////////DF////////////////////////////////

df = отображение информации об использовании дискового пространства

Параметры :

-h = отображение информации в понятной форме

/////////////////////////////////DU////////////////////////////////

du = отображение разма области дискового пространства, занятого определенным каталогом.

/////////////////////////////////Инсталяция/программного/обеспечения////////////////////////////////  
/////////////////////////////////Глава/XI////////////////////////////////

/////////////////////////////////RPM///пакеты////////////////////////////////  
/////////////////////////////////RPM////////////////////////////////

rpm [u-i] [название\_пакета].rpm = установка rpm пакетов(без установли связей)

Параметры :

-i = инсталяция  
-u = обновление до последней версии и инсталяция  
-h = отображение хода процесса установки  
-v = вывод подробной информации о выполняемых действиях (22)

-e = удаление программных пакетов

/////////////////////////////////YUM/////////////////////////////////

yum [команда] [название\_пакета] = установка программных пакетов с зависимостями.

Параметры :

-install = установка пакета

-remove = удаление пакета

-update = поиск обновлений для управляемых пакетов

-search [ключево\_слово] = поиск доступных к установке программ, содержащих ключевое слово в названии или описании.(не рекомендуем)

-list available = вывод списка всех доступных пакетов

/////////////////////////////////DEB////пакеты/////////////////////////////////

/////////////////////////////////DPKG/////////////////////////////////

dpkg [команда] [название\_пакета].deb = установка Debian пакетов без зависимостей

Параметры :

-i = установка пакета

-r = удаление пакета

/////////////////////////////////APT/////////////////////////////////

apt-get [команда] [название\_пакета] = установка пакетов с зависимостями.

Параметры :

-update = загрузка списка текущих программных пакетов с серверов и поиск новых версий пакетов

-install = устанавливает выбранный пакет

-remove = удаление пакета (23)

-upgrade = обновление установленных пакетов

-clean = удаление ненужных установочных пакетов из системы

apt-cache search [ключевое\_слово\_поиска] = поиск пакетов доступных для копирования в систему

/////////////////////////////////Сетевое//взаимодействие/////////////////////////////////

/////////////////////////////////Глава//XII/////////////////////////////////

/////////////////////////////////IFCONFIG/////////////////////////////////

ifconfig = определение состояния сетевых интерфейсов

ifconfig [имя\_ сетевого\_устройства] [новый\_IP\_адрес] = изменение IP адреса Ethernet карты

ifconfig [имя\_ сетевого\_устройства] -promisc = настройка сетевой карты на прием всех пакетов в поле зрения

/////////////////////////////////PING/////////////////////////////////

ping [адрес\_сети] = проверка способности компьютера принимать запросы

/////////////////////////////////TRACEROUTE/////////////////////////////////

traceroute [адрес\_узла] = контроль прохождения пакета между двумя узлами

/////////////////////////////////HOST/////////////////////////////////

host [доменное\_имя] = нахождение IP адреса, соответствующего данному доменному имени

host [IP\_адрес] = поиск доменного имени по IP адресу (24)

//////////////////////////////////IWCONFIG//////////////////////////////////

iwconfig = получение информации о состоянии сетевого интерфейса беспроводной сети

//////////////////////////////////IFUP//////////////////////////////////

ifup [сетевое\_устройство] = активизация сетевого соединения

//////////////////////////////////IFDOWN//////////////////////////////////

ifdown [сетевое\_устройство] = разъединение сетевого соединения

//////////////////////////////////ROUTE//////////////////////////////////

route = отображение таблицы маршрутизации

В данном руководстве представлены лишь, базовые команды. Не забывайте читать о командах в руководстве man.

//////////////////////////////////Работа/в/сети//////////////////////////////////

//////////////////////////////////Глава/XIII//////////////////////////////////

//////////////////////////////////SSH//////////////////////////////////

ssh [имя\_учетной\_записи]@[IP\_адрес\_машины] = организация защищенного взаимодействия с другим компьютером средствами SSH

Чтобы установить соединение с другой машиной,необходимо выполнить следующие команды : (25)

-ssh-keygen -t dsa

-ssh-copy-id -i ~/.ssh/id\_dsa.pub [имя\_учетной\_записи]@[IP\_адрес\_машины]

//////////////////////////////////SFTP//////////////////////////////////

sftp [имя\_учетной\_записи]@[IP\_адрес\_компьютера] = организация защищенного взаимодействия с другим компьютером средствами SFTP

Основные команды SFTP :

-cd = перейти в другой каталог

-exit = закрыть соединение с удаленным SSH сервером

-get = скопировать указанный файл на локальную машину

-help = отобразить справочную информацию

-lcd = перейти в другой каталог на локальной машине

-lls = отобразить список файлов на локальной машине

-ls = отобразить список файлов в данном каталоге удаленной машины

-put = скопировать указанный файл на удаленный сервер

-rm = удалить указанный файл с удаленного сервера

-scp [имя\_учетной\_записи1]@[IP\_адрес\_машины1]:файл\_1

[имя\_учетной\_записи2]@[IP\_адрес\_машины2]: файл\_2 = защищенное копирование файлов между машинами

-rsync -v = создание защищенной и удаленной резервной копии

//////////////////////////////////WGET//////////////////////////////////

wget [параметры] [ссылка\_на\_источник] = копирование файлов из WEB

Параметры :

-r = задает следование по ссылкам и просмотр каталогов в поисках файлов

-l[уровень\_глубины] = указывает глубину рекурсивного поиска файлов

-nr = запрещает осуществлять рекурсивный поиск в родительском каталоге (26)

-w [значение\_задержки]= задает небольшую задержку перед копированием каждого файла

-A.[формат\_фалов] = указывает, что необходимо скачивание только файлов определенного формата

-R.[формат\_фалов] = указываем тип файлов, которые не нужно копировать

-с = возобновление прерванного копирования

-E = при копировании WEB узла преобразует каждую страницу в html файл

-k = при копировании WEB узла указывает, что надо преобразовывать ссылки в состав WEB-страницы

-р = при копировании WEB узла позволяет корректно отображать скачанный узел, соединяя страницу со стилями и скриптами

culpr = аналог wget, главным отличием которого является возможность указывать последовательность и множества файлов для копирования