

UNIX/Linux системы в инфокоммуникациях

Продвинутый пользовательский курс в UNIX/Linux системах

В. Титов, О. Садов
tit@linux-ink.ru

Университет Информационных технологий, Механики и Оптики
кафедра СиОТ

23.11.2015

- ❶ `uname [-amnrsv]` — вывод информации о данной операционной системе;
- ❷ `date [ММДДччмм[ГГ]]` — вывод/установка текущей даты и времени.
- ❸ `cal [-y3] [месяц] год` — просмотр календаря;
- ❹ `id` — получить идентификатор пользователя;
- ❺ `who` — вывести список пользователей, работающих в системе;
- ❻ `write [пользователь]` — посылка сообщения другому пользователю;

Работа с файлами/каталогами

```
uname -a
```

```
Linux felix 2.6.32-220.2.1.el6.i686 #1 SMP Thu Dec 22 11:16  
CST 2011 i686 i686 i386 GNU/Linux
```

```
cal
```

```
          Декабрь 2012  
Пн Вт Ср Чт Пт Сб Вс  
                1  2  
 3  4  5  6  7  8  9  
10 11 12 13 14 15 16  
17 18 19 20 21 22 23  
24 25 26 27 28 29 30  
31
```

```
id user
```

```
uid=1000(user) gid=1000(user) группы=1000(user)
```

Работа с файлами/каталогами

- ❶ `ls [-aciltuFR]` — вывести имя текущего каталога;
- ❷ `pwd` — вывести имя текущего каталога;
- ❸ `cd [каталог]` — сменить текущий каталог;
- ❹ `cp [-i] файл ... файл или каталог`
или
`cp -r [-i] каталог ... каталог`
копировать файлы/каталоги в файл или каталог,
указанный последним параметром;
- ❺ `ln [-fs] файл или каталог ... [ссылка или каталог]` —
установить ссылку в файловой системе;
- ❻ `mv [-i] файл_или_каталог файл_или_каталог` —
переименовать/переместить файл или каталог;
- ❼ `rm [-r] [-f] [-i] файл/каталог ...` — удалить файл/каталог;
- ❽ `rmdir` каталог — удалить каталог;
- ❾ `mkdir` — создать каталог.

Работа с файлами/каталогами

```
$ ls -la
итого 224
drwx----- 16 user  user  4096 Map 21 18:16 .
drwxr-xr-x   3 root  root  4096 Map 21 17:53 ..
-rw-----   1 user  user    21 Map 21 17:54 .bash_history
-rw-r--r--   1 user  user    33 Дек 16 22:42 .bash_logout
-rw-r--r--   1 user  user   176 Дек 16 22:42 .bash_profile
-rw-r--r--   1 user  user   124 Map 16 22:42 .bashrc
drwxr-xr-x   2 user  user  4096 Map 21 17:53 Desktop

$ ln -s a b
$ ls -l
итого 8
-rw-rw-r--   1 guest guest  0 Map 21 18:57 a
lrwxrwxrwx   1 guest guest  1 Map 21 18:57 b -> a
```

Просмотр содержимого файлов

- ❶ `cat [-n] [файл]` — вывести на стандартный вывод содержимое файлов или, если файлы не указаны, содержимое стандартного ввода;
- ❷ `more [+номер_строки][+/образец_поиска] [файл]` — постраничный просмотр;
- ❸ `less [+номер_строки][+/образец_поиска] [файл]` — постраничный просмотр;
- ❹ `head [-n строк] [файл. ...]` — вывести первые строки файла;
- ❺ `tail [-n строк] [файл. ...]` — вывести последние строки файла;
- ❻ `tee [-a] [файл. ...]` — разделить входной поток;
- ❼ `wc [-clw] [файл. ...]` — подсчитать байты, строки, слова;
- ❽ `[e]grep [-cinvg] образец [файл. ...]` — найти строки по заданным образцам.

Просмотр содержимого файлов

```
$ cat > test
```

```
one
```

```
two
```

```
three
```

```
four
```

```
five
```

```
six
```

```
seven
```

Вывести содержимое файла test:

```
$ cat test
```

```
one
```

```
two
```

```
three
```

```
four
```

```
five
```

```
six
```

```
seven
```

```
$ head -n 5 test
```

```
one
```

```
two
```

```
three
```

```
four
```

```
five
```

```
$ tail -n 5 test
```

```
three
```

```
four
```

```
five
```

```
six
```

```
seven
```



```
$ grep -n ve test  
5:five  
7:seven
```

```
$ grep -v ee test  
one  
two  
four  
five  
six  
seven
```

- ❶ `tar [-]c|r|t|u|wv [f файл] [режим сжатия] файл или каталог...` — архивировать;
- ❷ `cpio -i[cdkmrtuv] [-H формат]`
или
`cpio -o[acvAL] [-O файл] [-H формат]`
или
`cpio -p[lmuvL] каталог`
архивировать;
- ❸ `gzip [-cdv] [файл...]`
`gunzip [-cv] [файл...]`
сжатие/разжатие файлов
- ❹ `bzip2 [-cdv] [файл...]`
`bunzip2 [-cv] [файл...]`
сжатие/разжатие файлов (более эффективное, по алгоритму Burrows–Wheeler);
- ❺ `md5sum [-c] файл`
`sha256sum [-c] файл`

❶ `find` каталог ... выражение

Выражение:

- `-name` файл;
- `-perm` права_доступа;
- `-type` [b|c|d|p|f];
- `-links` число_связей;
- `-user` пользователь;
- `-size` размер[c];
- `-[a|m|c]time` дней
- `-exec` команда [{ }];

Поиск и проверка состояния файлов

- ❶ `which` команда;
- ❷ `type` команда;
- ❸ `file` файл ...
- ❹ `test` выражение
или
`[выражение]`
вычисление условного выражения.

Поиск и проверка состояния файлов

```
$ find /usr/bin -name 'gcc*'
```

```
/usr/bin/gcc34
```

```
/usr/bin/gcc
```

```
/usr/bin/gccmakedep
```

```
$ file *
```

```
5: ASCII text
```

```
a: directory
```

```
birds.avi: RIFF (little-endian) data, AVI, 576 x 320
```

```
video: XviD, audio: MPEG-1 Layer 3 (stereo, 48000 Hz)
```

```
Booklet.doc: Microsoft Office Document
```

```
Booklet.pdf: PDF document, version 1.4
```

```
Desktop: directory
```

```
index.html: HTML document text
```

```
linux-ink_logo.gif: GIF image data, version 89a, 200 x 47
```

```
Nau.png: PNG image data, 270 x 55, 8-bit/color RGB
```

```
rn.sh: Bourne shell script text executable
```

```
test.tar: POSIX tar archive
```

Что нужно сделать, чтобы установить новое программное обеспечение?

- Настроить (возможно, применив патчи)
- Скомпилировать
- Установить (разместить файлы в нужных местах)
- `configure`
- `make`
- `make install`

- Легкость использования (rpm вместо configure/make/make install)
- Ориентирование на понятие «пакет»
- Возможность обновления пакетов (см. след. пункт)
- Отслеживание межпакетных зависимостей (в отличие от MSW, где ОС управляет приложениями с точки зрения всей системы)
- Возможность различных запросов
- Возможность верификации
- Поддержка различных процессорных архитектур
- Использование «чистых» исходников

RPM-пакет

RPM пакеты поставляются в виде сжатых архивов, которые содержат не менее 1 файла, а также инструкции по установке этих файлов, включая права доступа, которые должны быть применены к каждому файлу в процессе установки. Эти инструкции также могут содержать скрипты, которые запускаются перед или после установки (удаления) пакета. Имя пакета делится на **четыре** части, отделяемые друг от друга дефисами или (последние две части) точками:

имя-версия-релиз.архитектура.rpm. например:

`kernel-2.6.32-279.5.1.el6.i686`

`kernel-headers-2.6.32-279.14.1.el6.i686`

Имеется два типа пакетов: бинарные пакеты и пакеты с исходным кодом.

`kernel-2.6.32-279.5.1.el6.src`

Программа/система управления пакетами (rpm) ведет базу данных всех установленных в системе пакетов. Она находится в директории `/var/lib/rpm`.

Обновление/установка	-U	--upgrade
Установка	-i	--install
Удаление	-e	--erase
Режим запросов	-q	--query
Верификация	-V	--verify
Проверка подписи	-K	--checking
Обновление в режиме freshen	-F	--freshen
Инициализация БД	Нет	--initdb
Перестройка БД	Нет	--rebuilddb
Установите/удалите пакет (/var/lib/tftp...)		

```
# rpm -U foo-0.1-1.i386.rpm
```

```
# rpm -e foo
```

Установлен ли пакет, номер версии:

```
rpm -q foo
```

Вместе с опцией -q можно также использовать другие опции запроса.

```
rpm -qa foo
```

```
rpm -qa | grep foo
```

Информация о пакете:

```
rpm -qi foo
```

Список файлов пакета

```
rpm -ql foo
```

Список пакетов, от которых зависит этот пакет:

```
rpm -qR foo
```

Можно управлять форматом вывода информации:

```
rpm -q kernel \  
    --queryformat="%{NAME} %{RELEASE} %{SOURCERPM}\n"
```

```
rpm -qa --last | head
```

```
rpm -V initscripts
```

Запросы: --requires, --provides, --whatrequires,
--whatprovides, , --scripts, --triggers.

```
yum [options] [command] [package ...]
```

Команды yum:

- `install` — используется для установки последней версии пакета или группы пакетов, обеспечивая удовлетворение всех взаимных зависимостей пакетов;
- `update` — при запуске без указания имени пакета обновляет все установленные в системе пакеты, если указано имя пакета, обновляется только указанный пакет;
- `remove` или `erase` — используется для удаления из системы указанных пакетов;
- `list` — используется для просмотра информации об имеющихся пакетах;
- `info` — получение информации о пакете.
- `provides|whatprovides` — зависимости.
- `groupinstall|groupinstall|groupremove|groupinfo`.

- `localinstall` — используется для установки пакета с локального компьютера. При этом будут автоматически найдены и установлены пакеты, необходимые для разрешения зависимостей;
- `localupdate` — используется для обновления пакета с локального компьютера;
- `deplist` — показывает список имеющихся репозиториев. По умолчанию показываются только разрешенные репозитории.
- `replist` — показывает список имеющихся репозиториев. По умолчанию показываются только разрешенные репозитории.
- `search` — поиск по всем доступным репозиториям.

Yum, опции

-h, --help	Справка
-t, --tolerant	Режим игнорирования некритичных ошибок
-C	Работа из кеша (не обновлять его)
-c config_file	Альтернативный конфиг-файл
-R minutes	Максимальное время ожидания в минутах
-d debug_level	Уровень отладочных сообщений
-e error_level	Уровень многословности сообщений
-y	Автоматически отвечать "yes"
--version	Вывести версию yum
--installroot=path	Указать корневой каталог установки, отличный от каталога по умолчанию (корневой)
--enablerepo=repo	Сделать доступным доп. репозитории
--disablerepo=repo	Сделать недоступными репозитории
-x, exclude=package	Исключить пакет по имени или шаблону
--nogpgcheck	Отключить плагины yum
--nogpgcheck	Отключить проверку подписи gpg
--disableplugin=plugin	Отключить конкретный плагин по его имени

- yumex,
- PackageKit,
- Nautilus.

- Планирование того, что желательно собрать (в качестве образца можно использовать уже имеющийся спес-файл из `src.rpm`)
- Комплектование ПО для пакета (архивы с исходным кодом, патчи)
- Накладывание при необходимости патчей на исходный код
- Осуществление воспроизводимой сборки ПО (`configure/make/make install`)
- Планирование обновлений (как обновлять/удалять версии пакета)
- Отслеживание зависимостей (определить зависимости)
- Сборка пакета
- Тестирование пакета

Дерево директорий:
<rpmbuild>/

{SOURCES,SPECS,BUILD,RPMS/{noarch,i686},SRPMS,BUILDROOT}
Поместить исходный код (.tgz) в SOURCES, спец-файл —
в SPECS.

Секция общей информации:

Summary:

Summary (ru):

%define version 1.1

Licence:

Group: Development/Languages

Name: xxx

Prefix: /usr

Provides: xxx

Release: 1

Source: xxx-%version.tar.gz

URL: <http://ibm.com/developerworks/opensource/jikes>

Version: %version

Buildroot: /tmp/xxx

%description

%description -l ru

Содержание спес-файла

Секция prep:

```
%prep  
%setup -q  
%patch
```

Секция build:

```
%build  
./configure --prefix=$RPM_BUILD_ROOT/usr ...  
make
```

Секция install:

```
%install  
make install
```

Секция clean:

```
%clean  
rm -rf $RPM_BUILD_ROOT
```

Секция files:

```
%files
%defattr(-,root,root)
/usr/bin/xxx
%doc /usr/doc/xxx-%{version}/license.htm
%doc /usr/man/man1/xxx.1*
```

Секции post, preun, postun:

Команда rpmbuild

`rpmbuild -bСтадия_ сборки имя_ пакета.spec`

- ba Собрать бинарный пакет и пакет с исходным кодом
- bb Собрать бинарный пакет
- bc Скомпилировать программу, но не собирать rpm-пакет, то есть выполнить до секции %build включительно
- br Выполнить подготовку и остановиться сразу после завершения стадии %prep
- bi Выполнить сборку бинарного пакета и остановиться сразу после завершения стадии %install
- bl Выполнить проверку списка файлов для пакета и вывести резюме ошибок, если корневой каталог сборки не содержит каких-то файлов из списка
- bs Собрать только пакет с исходным кодом

```
rpmlintl <файл пакета.rpm>
```

Скрипты

```
rpm -qa | grep rpm
```

```
#!/bin/sh
```

```
tid='rpm -q --qf "%{INSTALLTID}\n" $*'
```

```
rpm -q --tid $tidrpm -qa | grep rpm
```