

# Table of Contents

<b>6 Руководство пользователя</b> .....	3
<b>Загрузка системы</b> .....	3
<b>Вход в систему</b> .....	3
<b>Документация</b> .....	3
<i>Утилита man</i> .....	4
<i>Утилита info</i> .....	4
<b>Установка дополнительного ПО</b> .....	4
<i>Установка пакета из репозитория</i> .....	5
<i>Установка пакета из файла</i> .....	5
<i>Обновление установленных пакетов</i> .....	5
<i>Установка ПО компании ВАСЭкспертс</i> .....	5
<b>Обзор основных команд системы</b> .....	6
<i>Учетные записи пользователей</i> .....	6
Команда su .....	6
Команда id .....	6
Команда passwd .....	6
<b><i>Основные операции с файлами и каталогами</i></b> .....	7
Команда ls .....	7
Команда cd .....	7
Команда pwd .....	8
Команда rm .....	8
Команда mkdir .....	8
Команда rmdir .....	8
Команда cp .....	9
Команда mv .....	9
Команда cat .....	9
Команда head .....	10
Команда less .....	10
Команда grep .....	10
Команда chmod .....	10
Команда chown .....	11
<b><i>Поиск файлов</i></b> .....	12
Команда find .....	12
Команда whereis .....	13
<b><i>Мониторинг и управление процессами</i></b> .....	13
Команда ps .....	13
Команда kill .....	13
Команда df .....	14
Команда du .....	14
Команда which .....	14
<b><i>Использование многозадачности</i></b> .....	15
Команда bg .....	15
Команда fg .....	15
<b><i>Сжатие и упаковка файлов</i></b> .....	15
Команда tar .....	15



# 6 Руководство пользователя

## Загрузка системы

Запуск VeOS выполняется автоматически после запуска компьютера и инициализации BIOS системы. На экране появляется меню загрузчика GRUB, в котором перечислены возможные варианты загрузки операционной системы.

Стрелками клавиатуры Вверх и Вниз выберите нужный вариант. Дополнительно к основным вариантам запуска ОС из этого меню можно загрузить VeOS в безопасном режиме или запустить проверку памяти. Загрузка операционной системы по умолчанию (первая в списке) начинается автоматически после небольшого времени ожидания (обычно несколько секунд). Нажав клавишу Enter, можно начать загрузку немедленно. Нажатием клавиши E можно вызвать редактор параметров текущего пункта загрузки. Если система настроена правильно, то редактировать их нет необходимости. В процессе загрузки VeOS пользователь может следить за информацией процесса загрузки, которая отображает этапы запуска различных служб и программных серверов в виде отдельных строк, на экране монитора.

Каждая строка начинается словом [Сообщение] (FAILED или OK), являющегося признаком нормального или ненормального завершения этапа загрузки. Слово Сообщение=FAILED (авария) свидетельствует о неуспешном завершении загрузки компонента системы, что требует вмешательства и специальных действий администратора системы. Загрузка операционной системы может занять некоторое время, в зависимости от производительности компьютера. Основные этапы загрузки VeOS — загрузка ядра, подключение (монтирование) файловых систем, запуск системных служб — периодически могут дополняться проверкой файловых систем на наличие ошибок. В этом случае время ожидания может быть занято больше времени, чем обычно. Подробную информацию о шагах загрузки можно получить, нажав клавишу Esc.

## Вход в систему

Стандартная установка VeOS включает базовую систему, работающую в консольном режиме. При загрузке в консольном режиме загрузчик VeOS завершается запросом на ввод логина и пароля учетной записи. В случае необходимости на другую консоль можно перейти, нажав Ctrl+Alt+F2. В случае успешного прохождения процедуры аутентификации и идентификации будет выполнен вход в систему. ОС VeOS перейдет к штатному режиму работы и предоставит доступ к консоли для пользователя.

## Документация

VeOS, как и все системы семейства UNIX, имеет экранную документацию. Её тексты содержат документацию по системным командам, ресурсам, конфигурационным файлам и т. д., а также

могут быть выведены на экран в процессе работы.

## Утилита `man`

Для доступа к экранной документации используется команда `man` (сокращение от `manual`). Каждая страница руководства посвящена одному объекту системы. Для того чтобы прочесть страницу руководства по программе, необходимо набрать `man название_программы`. К примеру, если вы хотите узнать, какие опции есть у команды `ls`, вы можете ввести команду:

```
> man ls
```

Большинство экранной документации написано для пользователей, имеющих некоторое представление о том, что делает данная команда. Поэтому большинство текстов экранной документации содержит исключительно технические детали команды без особых пояснений. Тем не менее, экранная документация оказывается очень ценной в том случае, если вы помните название команды, но её синтаксис просто выпал у вас из памяти.

Поиск по описаниям `man` осуществляется командой `apropos`. Если вы точно не знаете, как называется необходимая вам программа, то поиск осуществляется по ключевому слову, к примеру, `apropos ls` или при помощи ввода слова, обозначающего нужное действие, после команды `man -k` (например, `man -k list`). Слово, характеризующее желаемое для вас действие, можно вводить и на русском языке. При наличии русского перевода страниц руководства `man` результаты поиска будут выведены на запрашиваемом языке. «Страница руководства» занимает, как правило, больше одной страницы экрана. Для того чтобы читать было удобнее, `man` запускает программу постраничного просмотра текстов. Страницы перелистываются пробелом, для выхода из режима чтения описания команд `man` необходимо нажать на клавиатуре `q`. Команда `man man` выдаёт справку по пользованию самой командой `man`.

## Утилита `info`

Другой источник информации о `VeOS` и составляющих его программах — справочная подсистема `info`. Страница руководства, несмотря на обилие ссылок различного типа, остаётся «линейным» текстом, структурированным только логически. Документ `info` — это настоящий гипертекст, в котором множество небольших страниц объединены в дерево. В каждом разделе документа `info` всегда есть оглавление, из которого можно перейти к нужному подразделу, а затем вернуться обратно (ссылки для перемещения по разделам текста помечены \*). Для получения вспомогательной информации о перемещении по тексту используйте клавишу `h`. Полное руководство `info` вызывается командой `info info`. Команда `info`, введённая без параметров, предлагает пользователю список всех документов `info`, установленных в системе.

# Установка дополнительного ПО

Для установки дополнительного программного обеспечения в `VeOS` используются утилиты `dnf` (или `yum`) для загрузки `rpm`-пакетов из репозитория `VeOS` и `rpm` для установки из `rpm`-файлов.



Для установки программного обеспечения у пользователя должны быть права администратора системы (root) или разрешение на запуск утилиты sudo

## Установка пакета из репозитория

Для установки или обновления пакета из репозитория необходимо выполнить следующую команду:

```
> dnf install <имя пакета>
```

Для удаления установленного пакета:

```
> dnf remove <имя пакета>
```

## Установка пакета из файла

Иногда требуется установить локально скачанный rpm-пакет:

```
> rpm -ih <rpm-файл>
```

Для обновления пакета:

```
> rpm -Uh <rpm-файл>
```

## Обновление установленных пакетов

Периодически в репозиториях VeOS появляются обновления установленных пакетов, для установки обновлений пакетов необходимо запустить:

```
> dnf update
```

В процессе обновления утилита обнаружит последние неустановленные версии для всех пакетов системы, вычислит их зависимости, скачает их и установит после подтверждения пользователя.

## Установка ПО компании ВАСЭкспертс

По умолчанию в состав VeOS настроен доступ к репозиториям ВАСЭкспертс и все пакеты компании доступны к установке без дополнительных настроек. Например, установка пакета DPI производится так:

```
> dnf install fastdpi
```

## Обзор основных команд системы

Все команды, приведенные ниже, могут быть запущены в режиме консоли. Для получения более подробной информации используйте команду `man`. Пример:

```
> man ls
```

Примечание Параметры команд обычно начинаются с символа «-», и обычно после одного символа «-» можно указать сразу несколько опций. Например, вместо команды `ls -l -F` можно ввести команду `ls -lF`

## Учетные записи пользователей

### Команда `su`

Команда `su` позволяет изменить «владельца» текущего сеанса (сессии) без необходимости завершать сеанс и открывать новый. Синтаксис:

```
> su [ОПЦИИ...] [ПОЛЬЗОВАТЕЛЬ]
```

Команду можно применять для замены текущего пользователя на любого другого, но чаще всего она используется для получения пользователем прав суперпользователя (`root`). При вводе команды `su -`, будет запрошен пароль суперпользователя (`root`), и, в случае ввода корректного пароля, пользователь получит права администратора. Чтобы вернуться к правам пользователя, необходимо ввести команду:

```
> exit
```

Более подробную информацию о режиме суперпользователя вы можете прочитать в главе [Режим суперпользователя](#)

### Команда `id`

Команда `id` выводит информацию о пользователе и группах, в которых он состоит для заданного пользователя или о текущем пользователе (если ничего не указано). Синтаксис:

```
> id [ОПЦИИ...] [ПОЛЬЗОВАТЕЛЬ]
```

### Команда `passwd`

Команда `passwd` меняет (или устанавливает) пароль, связанный с входным\_именем пользователя. Обычный пользователь может менять только пароль, связанный с его собственным входным\_именем. Команда запрашивает у обычных пользователей старый пароль (если он был), а затем дважды запрашивает новый. Новый пароль должен соответствовать техническим требованиям к паролям, заданным администратором системы. Основные операции с файлами и каталогами

## Основные операции с файлами и каталогами

### Команда `ls`

Команда `ls` (`list`) печатает в стандартный вывод содержимое каталогов. Синтаксис:

```
> ls [ОПЦИИ...] [ФАЙЛ...]
```

Основные опции:

- `-a` — просмотр всех файлов, включая скрытые;
- `-l` — отображение более подробной информации;
- `-R` — выводить рекурсивно информацию о подкаталогах.

### Команда `cd`

Команда `cd` предназначена для смены каталога. Команда работает как с абсолютными, так и с относительными путями. Если каталог не указан, используется значение переменной окружения `$HOME` (домашний каталог пользователя). Если каталог задан полным маршрутным именем, он становится текущим. По отношению к новому каталогу нужно иметь право на выполнение, которое в данном случае трактуется как разрешение на поиск. Синтаксис:

```
> cd [-L|-P] [КАТАЛОГ]
```

Если в качестве аргумента задано «-», то это эквивалентно `$OLDPWD`. Если переход был осуществлен по переменной окружения `$CDPATH` или в качестве аргумента был задан «-» и смена каталога была успешной, то абсолютный путь нового рабочего каталога будет выведен на стандартный вывод. Пример. Находясь в домашнем каталоге перейти в его подкаталог `docs/` (относительный путь):

```
> cd docs/
```

Сделать текущим каталог `/usr/bin` (абсолютный путь):

```
> cd /usr/bin/
```

Сделать текущим родительский каталог:

```
> cd ..
```

Вернуться в предыдущий каталог:

```
> cd -
```

Сделать текущим домашний каталог:

```
> cd
```

## Команда pwd

Команда `pwd` выводит абсолютный путь текущего (рабочего) каталога. Синтаксис:

```
> pwd [-L|-P]
```

Опции:

- `-P` — не выводить символические ссылки;
- `-L` — выводить символические ссылки.

## Команда rm

Команда `rm` служит для удаления записей о файлах. Если заданное имя было последней ссылкой на файл, то файл уничтожается. Предупреждение Удалив файл, вы не сможете его восстановить! Синтаксис:

```
> rm [ОПЦИИ...] <ФАЙЛ>
```

Основные опции:

- `-f` — никогда не запрашивать подтверждения;
- `-i` — всегда запрашивать подтверждение;
- `-r`, `-R` — рекурсивно удалять содержимое указанных каталогов.

Пример. Удалить все файлы `html` в каталоге `~/html`:

```
> rm -i ~/html/*.html
```

## Команда mkdir

`mkdir` — команда для создания новых каталогов. Синтаксис:

```
> mkdir [-p] [-m права] <КАТАЛОГ...>
```

## Команда rmdir

Команда `rmdir` удаляет каталоги из файловой системы. Каталог должен быть пуст перед

удалением. Синтаксис:

```
> rmdir [ОПЦИИ] <КАТАЛОГ...>
```

Основные опции:

- -r — удалить каталог и его потомки.

Команда `rmdir` часто заменяется командой `rm -rf`, которая позволяет удалять каталоги, даже если они не пусты.

## Команда `cp`

Команда `cp` предназначена для копирования файлов из одного в другие каталоги. Синтаксис:

```
> cp [-fip] [ИСХ_ФАЙЛ...] [ЦЕЛ_ФАЙЛ...]  
> cp [-fip] [ИСХ_ФАЙЛ...] [КАТАЛОГ]  
> cp [-R] [[-H] | [-L] | [-P]] [-fip] [ИСХ_ФАЙЛ...] [КАТАЛОГ]
```

Основные опции:

- -r — сохранять по возможности времена изменения и доступа к файлу, владельца и группу, права доступа;
- -i — запрашивать подтверждение перед копированием в существующие файлы;
- -r, -R — рекурсивно копировать содержимое каталогов.

## Команда `mv`

Команда `mv` предназначена для перемещения файлов. Синтаксис:

```
> mv [-fi] [ИСХ_ФАЙЛ...] [ЦЕЛ_ФАЙЛ...]  
> mv [-fi] [ИСХ_ФАЙЛ...] [КАТАЛОГ]
```

В первой синтаксической форме, характеризующейся тем, что последний операнд не является ни каталогом, ни символической ссылкой на каталог, `mv` перемещает `исх_файл` в `цел_файл` (происходит переименование файла). Во второй синтаксической форме `mv` перемещает исходные файлы в указанный каталог под именами, совпадающими с краткими именами исходных файлов. Основные опции:

- -f — не запрашивать подтверждения перезаписи существующих файлов;
- -i — запрашивать подтверждение перезаписи существующих файлов.

## Команда `cat`

Команда `cat` последовательно выводит содержимое файлов. Синтаксис:

```
> cat [ОПЦИИ] [ФАЙЛ...]
```

Основные опции:

- -n, --number — нумеровать все строки при выводе;
- -E, --show-ends — показывать \$ в конце каждой строки.

Если файл не указан, читается стандартный ввод. Если в списке файлов присутствует имя «-», вместо этого файла читается стандартный ввод.

## Команда head

Команда head выводит первые 10 строк каждого файла на стандартный вывод. Синтаксис:

```
> head [ОПЦИИ] [ФАЙЛ...]
```

Основные опции:

- -n, --lines=[-]К — вывести первые К строк каждого файла, а не первые 10;
- -q, --quiet — не печатать заголовки с именами файлов.

## Команда less

Команда less позволяет постранично просматривать текст (для выхода необходимо нажать q). Синтаксис:

```
> less ФАЙЛ
```

## Команда grep

Команда grep имеет много опций и предоставляет возможности поиска символьной строки в файле. Синтаксис:

```
> grep [шаблон_поиска] ФАЙЛ
```

## Команда chmod

Команда chmod предназначена для изменения прав доступа файлов и каталогов. Синтаксис:

```
> chmod [ОПЦИИ] РЕЖИМ[,РЕЖИМ]... <ФАЙЛ>  
> chmod [ОПЦИИ] --reference=ИФАЙЛ <ФАЙЛ>
```

Основные опции:

- -R — рекурсивно изменять режим доступа к файлам, расположенным в указанных каталогах;
- --reference=ИФАЙЛ — использовать режим файла ИФАЙЛ.

chmod изменяет права доступа каждого указанного файла в соответствии с правами доступа, указанными в параметре режим, который может быть представлен как в символьном виде, так и в виде восьмеричного, представляющего битовую маску новых прав доступа. Формат символьного режима следующий:

```
> [ugoа . . . ] [ [+ -= ] [разрешения...]. . .]
```

Здесь разрешения — это ноль или более букв из набора «rwxXst» или одна из букв из набора «ugo». Каждый аргумент — это список символьных команд изменения прав доступа, разделены запятыми. Каждая такая команда начинается с нуля или более букв «ugoа», комбинация которых указывает, чьи права доступа к файлу будут изменены: пользователя, владеющего файлом (u), пользователей, входящих в группу, к которой принадлежит файл (g), остальных пользователей (o) или всех пользователей (a). Если не задана ни одна буква, то автоматически будет использована буква «а», но биты, установленные в umask, не будут затронуты. Оператор «+» добавляет выбранные права доступа к уже имеющимся у каждого файла, «-» удаляет эти права. «=» присваивает только эти права каждому указанному файлу. Буквы «rwxXst» задают биты доступа для пользователей: «r» — чтение, «w» — запись, «x» — выполнение (или поиск для каталогов), «X» — выполнение/поиск только если это каталог или же файл с уже установленным битом выполнения, «s» — задать ID пользователя и группы при выполнении, «t» — запрет удаления. Примеры. Позволить всем выполнять файл f2:

```
> chmod +x f2
```

Запретить удаление файла f3:

```
> chmod +t f3
```

## Команда chown

Команда chown изменяет владельца и/или группу для каждого заданного файла. Синтаксис:

```
> chown [КЛЮЧ]...[ВЛАДЕЛЕЦ][:[ГРУППА]] <ФАЙЛ>
```

Изменить владельца может только владелец файла или суперпользователь. Владелец не изменяется, если он не задан в аргументе. Группа также не изменяется, если не задана, но если после символьного ВЛАДЕЛЬЦА стоит символ «:», подразумевается изменение группы на основную группу текущего пользователя. Поля ВЛАДЕЛЕЦ и ГРУППА могут быть как числовыми, так и символьными. Примеры. Поменять владельца каталога /u на пользователя test:

```
> chown test /u
```

Поменять владельца и группу каталога /u:

```
> chown test:staff /u
```

Поменять владельца каталога /u и вложенных файлов на test:

```
> chown -hR test /u
```

# Поиск файлов

## Команда find

Команда find предназначена для поиска всех файлов, начиная с корневого каталога. Поиск может осуществляться по имени, типу или владельцу файла. Синтаксис:

```
> find [-H] [-L] [-P] [-0уровень] [-D help|tree|search|stat|rates|opt|exec]
[ПУТЬ...] [ВЫРАЖЕНИЕ]
```

Ключи для поиска:

- -name — поиск по имени файла;
- -type — поиск по типу f=файл, d=каталог, l=ссылка(lnk);
- -user — поиск по владельцу (имя или UID).

Когда выполняется команда find, можно выполнять различные действия над найденными файлами. Основные действия:

- -exec команда \; — выполнить команду. Запись команды должна заканчиваться экранированной точкой с запятой. Строка «{ }» заменяется текущим маршрутным именем файла;
- execdir команда \; — то же самое что и -exec, но команда вызывается из подкаталога, содержащего текущий файл;
- -ok команда — эквивалентно -exec за исключением того, что перед выполнением команды запрашивается подтверждение (в виде сгенерированной командной строки со знаком вопроса в конце) и она выполняется только при ответе: y;
- -print — вывод имени файла на экран.

Путем по умолчанию является текущий подкаталог. Выражение по умолчанию -print. Примеры. Найти в текущем каталоге обычные файлы (не каталоги), имя которых начинается с символа «~»:

```
> find . -type f -name "~*" -print
```

Найти в текущем каталоге файлы, измененные позже, чем файл file.bak:

```
> find . -newer file.bak -type f -print
```

Удалить все файлы с именами a.out или \*.o, доступ к которым не производился в течение недели:

```
> find / \( -name a.out -o -name '*.o' \) \ -atime +7 -exec rm {} \;
```

Удалить из текущего каталога и его подкаталогов все файлы нулевого размера, запрашивая подтверждение:

```
> find . -size 0c -ok rm {} \;
```

## Команда whereis

whereis сообщает путь к исполняемому файлу программы, ее исходным файлам (если есть) и соответствующим страницам справочного руководства. Синтаксис:

```
> whereis [ОПЦИИ] <ИМЯ>
```

Опции:

- -b — вывод информации только об исполняемых файлах;
- -m — вывод информации только о страницах справочного руководства;
- -s — вывод информации только об исходных файлах.

## Мониторинг и управление процессами

### Команда ps

Команда ps отображает список текущих процессов. Синтаксис:

```
> ps [ОПЦИИ]
```

По умолчанию выводится информация о процессах с теми же действующим UID и управляющим терминалом, что и у подающего команду пользователя. Основные опции:

- -a — вывести информацию о процессах, ассоциированных с терминалами;
- -f — вывести «полный» список;
- -l — вывести «длинный» список;
- -p список — вывести информацию о процессах с перечисленными в списке PID;
- -u список — вывести информацию о процессах с перечисленными идентификаторами или именами пользователей.

### Команда kill

Команда kill позволяет прекратить исполнение процесса или передать ему сигнал. Синтаксис:

```
> kill [-s] [сигнал] [идентификатор] [...]  
> kill [-l] [статус_завершения]  
> kill [-номер_сигнала] [идентификатор] [...]
```

Идентификатор — PID ведущего процесса задания или номер задания, предваренный знаком «%». Основные опции:

- -l — вывести список поддерживаемых сигналов;
- -s сигнал, -сигнал — послать сигнал с указанным именем.

Если обычная команда kill не дает желательного эффекта, необходимо использовать команду kill с параметром -9 (kill -9 PID\_номер).

## Команда df

Команда df показывает количество доступного дискового пространства в файловой системе, в которой содержится файл, переданный как аргумент. Если ни один файл не указан, показывается доступное место на всех смонтированных файловых системах. Размеры по умолчанию указаны в блоках по 1КБ. Синтаксис:

```
> df [ОПЦИИ] [ФАЙЛ...]
```

Основные опции:

- --total — подсчитать общий объем в конце;
- -h, --human-readable — печатать размеры в удобочитаемом формате (например, 1K, 234M, 2G).

## Команда du

Команда du подсчитывает использование диска каждым файлом, для каталогов подсчет происходит рекурсивно. Синтаксис:

```
> du [ОПЦИИ] [ФАЙЛ...]
```

Основные опции:

- -a, --all — выводить общую сумму для каждого заданного файла, а не только для каталогов;
- -c, --total — подсчитать общий объем в конце. Может быть использовано для выяснения суммарного использования дискового пространства для всего списка заданных файлов;
- -d, --max-depth=N — выводить объем для каталога (или файлов, если указано --all) только если она на N или менее уровней ниже аргументов командной строки;
- -S, --separate-dirs — выдавать отдельно размер каждого каталога, не включая размеры подкаталогов;
- -s, --summarize — отобразить только сумму для каждого аргумента.

## Команда which

Команда which отображает полный путь к указанным командам или сценариям. Синтаксис:

```
> which [ОПЦИИ] <ФАЙЛ...>
```

Основные опции:

- -a, --all — выводит все совпавшие исполняемые файлы по содержимому в переменной окружения \$PATH, а не только первый из них;
- -c, --total — подсчитать общий объем в конце. Может быть использовано для выяснения суммарного использования дискового пространства для всего списка заданных файлов;
- -d, --max-depth=N — выводить объем для каталога (или файлов, если указано --all) только если она на N или менее уровней ниже аргументов командной строки;

- `-S, --separate-dirs` — выдавать отдельно размер каждого каталога, не включая размеры подкаталогов;
- `--skip-dot` — пропускает все каталоги из переменной окружения `$PATH`, которые начинаются с точки.

## Использование многозадачности

VeOS — это многозадачная система. Для того, чтобы запустить программу в фоновом режиме, необходимо набрать «&» после имени программы. После этого оболочка даст возможность запускать другие приложения. Так как некоторые программы интерактивны — их запуск в фоновом режиме бессмысленен. Подобные программы просто остановятся, если их запустить в фоновом режиме. Можно также запускать нескольких независимых сеансов. Для этого в консоли необходимо набрать `Alt` и одну из клавиш, находящихся в интервале от `F1` до `F6`. На экране появится новое приглашение системы, и можно открыть новый сеанс. Этот метод также позволяет вам работать на другой консоли, если консоль, которую вы использовали до этого, не отвечает или вам необходимо остановить зависшую программу.

### Команда `bg`

Команда `bg` позволяет перевести задание на задний план. Синтаксис:

```
> bg [ИДЕНТИФИКАТОР ...]
```

Идентификатор — PID ведущего процесса задания или номер задания, предварённый знаком «%».

### Команда `fg`

Команда `fg` позволяет перевести задание на передний план. Синтаксис:

```
> fg [ИДЕНТИФИКАТОР ...]
```

Идентификатор — PID ведущего процесса задания или номер задания, предварённый знаком «%».

## Сжатие и упаковка файлов

### Команда `tar`

Сжатие и упаковка файлов выполняется с помощью команды `tar`, которая преобразует файл или группу файлов в архив без сжатия (tarfile). Упаковка файлов в архив чаще всего выполняется следующей командой:

```
> tar -cf [имя создаваемого файла архива] [упаковываемые файлы и/или каталоги]
```

Пример использования команды упаковки архива:

```
> tar -cf moi_dokumenti.tar Docs project.tex
```

Распаковка содержимого архива в текущий каталог выполняется командой:

```
> tar -xf [имя файла архива]
```

Для сжатия файлов используются специальные программы сжатия: gzip, bzip2 и 7z.