

# Резервное копирование

# 3

## 3.1. Необходимость в резервном копировании

Необходимость в создании резервных копий возникает у каждого администратора, поскольку не хочется:

- объяснять пользователям, куда пропали их файлы, хранящиеся на сервере;
- объяснять начальству, почему сеть целый день не работает (ведь сервер вышел из строя и нужно как минимум переустановить систему и настроить ее);
- тратить время на переустановку и настройку системы с нуля.

Существует множество инструментов для создания резервных копий. Каждый инструмент хорош по-своему. В простейшем случае можно просто использовать команду `cp` для копирования нужных вам данных. А что? Тоже хороший инструмент. Копия создана? Создана. Можно ее восстановить? Да проще простого — скопировать файл на «исходную позицию». К тому же у `cp` есть множество преимуществ перед другими средствами. Во-первых, команда `cp` очень быстра — не нужно тратить время на упаковку и распаковку данных. Во-вторых, для создания и восстановления файлов не нужны дополнительные программы. Инструментом `cp` пользовался каждый администратор; как правило, перед редактированием файла нужно сохранить его копию, чтобы в случае чего можно было легко вернуть все, как было.

Но у команды `cp` есть один недостаток. Она не сжимает данные. Следовательно, для создания копии вам понадобится столько места, сколько занимают исходные файлы, что не всегда удобно с экономической точки зрения. Ведь чтобы сделать копию пользовательских каталогов размером, скажем, в 200 Гбайт, вам придется приобрести еще один жесткий диск. Учитывая, что копию пользовательских данных нужно делать как можно чаще (чтобы удовлетворить примерно следующую просьбу пользователя: «В прошлую пятницу я внес неправильные

изменения в свой `index.php`, пожалуйста, восстановите его копию»), места вам понадобится много.

Поэтому вы будете искать другой инструмент. Можно, например, создать `tar`-архив всех пользовательских каталогов. Очень удобно — все будет в одном файле. К тому же архиватор `tar` поддерживает права доступа к файлам и каталогам, поэтому когда вы распакуете `tar`-архив, все файлы будут помещены на свои места с нужными правами доступа. К тому же `tar` умеет вызывать архиватор и сжимать полученный архив (опция `-z`). Например, для сжатия файлов в текущем каталоге нужно выполнить команду:

```
tar -zcfv archive.tar.gz
```

А для распаковки архива `archive.tar.gz` используется другая команда:

```
tar -zxvf archive.tar.gz
```

Для распаковки из архива только определенного каталога используется примерно следующая команда (в данном случае мы извлекаем каталог `/home/denis`):

```
tar --extract --file=archive.tgz home/denis
```

Казалось бы, у нас есть идеальный инструмент, поддерживающий и структуру каталогов, и права доступа, и сжатие файлов. Но у всего есть недостатки. Архиватор `tar`, да еще и со сжатием, довольно медленный. Попробуйте с его помощью сжать файлы и каталоги общим объемом, скажем, 400 Гбайт — вполне нормальный объем для реальной резервной копии реального сервера.

Продолжим поиски нашего идеального инструмента. Можно использовать программу `dump`, которая работает намного быстрее, к тому же можно добавить сжатие `bzip2`, что позволит сэкономить место на диске. Вот пример создания резервной копии раздела `ad0s1d`:

```
# dump -0ua -L -f- /dev/ad0s1d | bzip2> ad0s1d.img.bz2
```

Распаковать этот архив можно потом так:

```
# bunzip2 --stdout ad0s1d.img.bz2 | restore -rf -
```

Теоретически, вы можете периодически создавать дампы всех BSD-разделов на другой жесткий диск. Затем можно загрузиться с LiveCD (можно использовать команду `Fixit` меню инсталляционного диска) и восстановить по одному все разделы. Конечно, нужно будет по очереди монтировать разделы, для каждого раздела вводить команду восстановления и команду размонтирования. Хотелось бы какой-то автоматизации: загрузился, выбрал команду восстановления, выбрал хранилище резервных копий, выбрал, куда нужно восстановить, подождал немного и после перезагрузки система полностью готова к работе. Можно написать сценарии, автоматизирующие процесс резервного копирования и восстановления данных. Но зачем изобретать колесо, если все сделано до нас?

## 3.2. Почти идеальный инструмент для создания резервной копии и клонирования системы

Для себя я нашел почти идеальный инструмент для создания резервных копий — Clonezilla. Вообще-то Clonezilla — это инструмент не для создания резервных копий, а для клонирования системы. Суть его использования заключается в следующем. Вы устанавливаете систему и настраиваете ее. После запускаете Clonezilla и в результате получаете загрузочный диск с клоном вашей системы. Что делать с этим диском потом, зависит от поставленной задачи. Если вам нужно настроить еще 50 компьютеров аналогичной аппаратной конфигурации, можете сделать еще несколько копий «диска клонирования» и параллельно устанавливать систему. Процесс восстановления системы будет происходить намного быстрее, чем установка. Ведь при установке система спрашивает вас, что вы хотите установить, потом, исходя из вашего ответа, отбирает для установки необходимые файлы (пакеты), переносит их на жесткий диск и распаковывает. Также в процессе установки вам задается ряд вопросов, на которые нужно обдуманно ответить. А вот при восстановлении системы Clonezilla просто переносит ранее созданный образ эталонного жесткого диска на жесткий диск целевой машины. В результате на все про все уходит 10–15 минут. Быстро? Я тоже так думаю.

Даже если вам не нужно клонировать эталонную машину на несколько десятков компьютеров, Clonezilla вам пригодится. Представьте, что жесткий диск вашего сервера вышел из строя. Сколько времени понадобится на восстановление системы? Если не принимать во внимание время, затраченное на поиск нового жесткого диска, то на восстановление системы уйдут те самые 10–15 минут плюс время, необходимое на установку нового жесткого диска (вдруг у вас будет запасной!). Начальство и пользователи будут довольны.

Понятно, что только что установленная и настроенная система занимает совсем немного места, поэтому она поместится на DVD. А что делать, если у вас сервер баз данных и сама база данных разрослась, скажем, до 100 Гбайт? Clonezilla и тут поможет — вы можете хранить образы резервной копии на другом жестком диске (или на этом же, но как тогда вам поможет резервная копия, если выйдет из строя жесткий диск с резервной копией?). Правда, в этом случае использовать Clonezilla не очень удобно, поскольку она инициирует полное восстановление системы. Но зачем нам восстанавливать всю систему, если нужно восстановить всего один файл? Так что Clonezilla поможет вам при полном крахе системы, а вот файлы пользователей и системные файлы конфигурации (каталоги `/etc` и `/usr/local/etc`) нужно будет копировать с помощью `tar` или `dump`.

Программа Clonezilla может использоваться не только для клонирования FreeBSD, но также Linux и Windows. Обычно предприятие закупает компьютеры одинаковой конфигурации, но в процессе постепенного перехода на свободное бесплатное программное обеспечение часть компьютеров будет работать под Windows, часть — под Linux, поэтому можно сказать, что Clonezilla — универсальный инструмент для администратора. К примеру, нужно переустановить Windows со всеми программами. Вы копируете на флешку документы пользователя, затем

загружаетесь с диска восстановления, который вы предварительно создали, и через 15 минут получаете полностью готовый компьютер с установленной Windows и всеми программами.

Давайте разберемся, как использовать Clonezilla. Скачайте с <http://clonezilla.org> ISO-образ Clonezilla и запишите его на болванку. Загрузитесь с него. Сначала нужно будет выбрать язык и клавиатуру, но поскольку русского интерфейса для Clonezilla нет, оставьте все по умолчанию. Затем нужно выбрать команду Start Clonezilla (рис. 3.1).

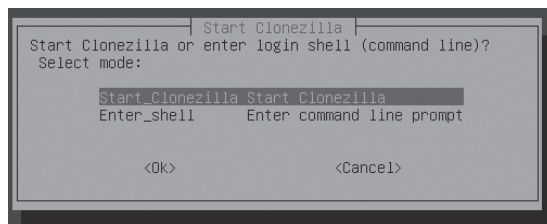


Рис. 3.1. Запустить Clonezilla

Далее нужно выбрать режим работы программы (рис. 3.2):

- device-image — создание образа раздела;
- device-device — резервная копия будет помещена на другой раздел (используется при большом размере раздела).

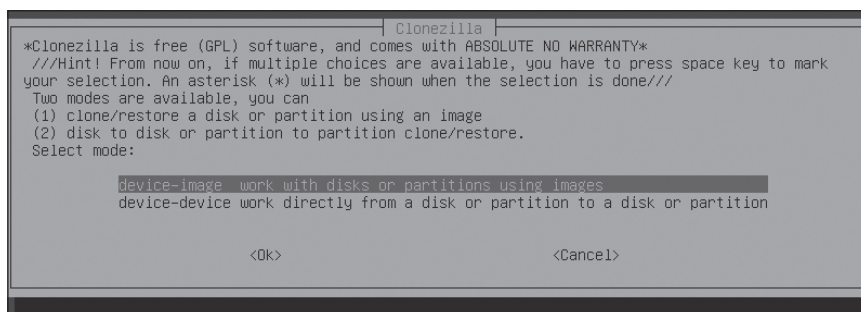


Рис. 3.2. Выбор режима работы

Следующий шаг — выбор типа хранилища резервной копии. Выберите local\_dev (рис. 3.3), что означает запись резервной копии на локальное устройство (в случае создания резервной копии) или чтение резервной копии с локального устройства (в случае восстановления системы). Сразу после этого нужно будет выбрать раздел-хранилище.

После этого нужно выбрать режим работы программы (рис. 3.4):

- savedisk — сохранить весь жесткий диск как образ;
- saveparts — сохранить отдельно выбранный раздел жесткого диска (речь идет о разделах жесткого диска, а не BSD-разделах);

- `restoredisk` — восстановить весь диск из резервной копии;
- `restoreparts` — восстановить раздел из резервной копии;
- `recovery-iso-zip` — создать загрузочный диск восстановления (клонирования) системы;
- `exit` — ВЫХОД.

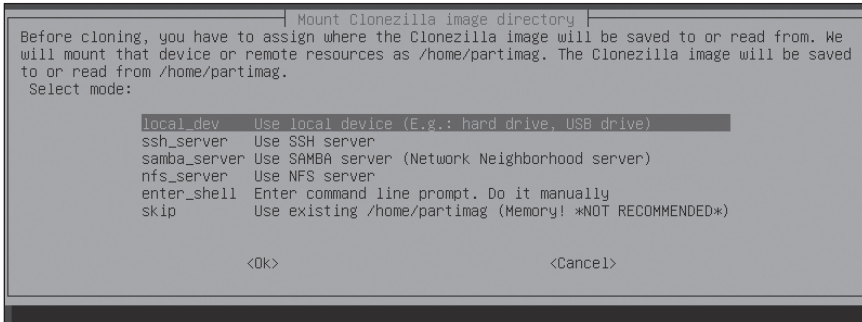


Рис. 3.3. Выбор типа хранилища

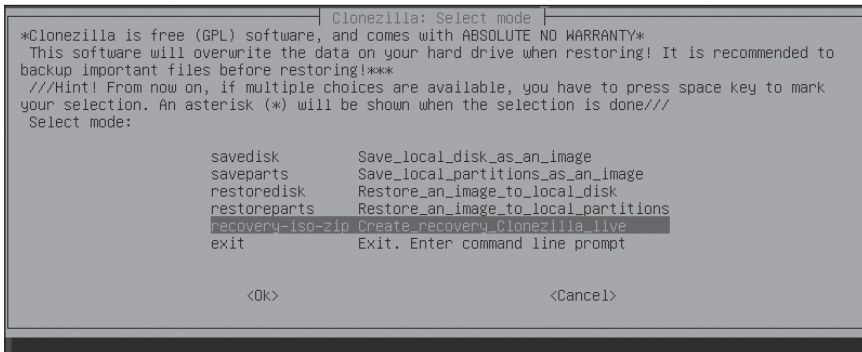


Рис. 3.4. Режим работы программы

Дальше вы справитесь сами. Хочу только отметить следующее: когда программа запросит ввести имя устройства, нужно вводить имя устройства в стиле Linux, то есть `sda1`, а не `ad0s1`.

Принцип работы Clonezilla, наверное, напомнил опытным Windows-администраторам популярный продукт Norton Ghost. Однако, в отличие от «привидения Нортон», Clonezilla — абсолютно бесплатная программа. Так что в этой главе мы познакомились еще с одним бесплатным аналогом популярного коммерческого продукта. Кстати, сейчас бесплатная Clonezilla сэкономила вам 69 долларов (на самом деле чуть больше, учитывая банковские услуги) — именно за столько продается Norton Ghost для Windows 7:

<http://us.norton.com/ghost/>

В заключение этой главы настоятельно рекомендую прочитать страницы справочного руководства по командам `tar`, `dump` и `restore` — эти команды вы также будете использовать при создании резервных копий. Можно также ознакомиться с архиваторами `bzip2` и `bunzip2`, которые мы использовали при создании и восстановлении дампа системы командами `dump` и `restore`.

# Вход в систему и завершение работы

# 4

## 4.1. Вход в систему

В этой небольшой главе мы поговорим о том, как войти в систему после ее установки и как правильно завершить работу. Начнем с FreeBSD. Обычно на FreeBSD не устанавливается графический интерфейс. Такая возможность есть, но большинство администраторов предпочитают работать в консоли (текстовом режиме). И это решение абсолютно правильное. Незачем тратить время на настройку видеокарты (а во FreeBSD время потратить придется — это не Linux, где для корректной работы видеоподсистемы практически ничего не нужно делать), если вам нужно отредактировать несколько конфигурационных файлов, запустить сервисы и забыть вообще о существовании вашего сервера. А все административные действия можно выполнять и в консоли.

После запуска FreeBSD вы увидите приглашение войти в систему:

```
FreeBSD/i386 (den.dkws.org.ua) (ttyv0)
login: root
Password:
```

Введите имя пользователя (root) и его пароль. При вводе символы пароля никак не отображаются — не будет ни звездочек, ни точек, — просто введите пароль и нажмите Enter. Вы вошли в систему. Что делать дальше? А дальше вам нужно вводить различные команды в зависимости от того, что вы хотите сделать. Хотите отредактировать конфигурационный файл? Запустите текстовый редактор и откройте необходимый файл. Командной строке посвящена глава 6. В ней вы узнаете, как выбрать командный интерпретатор, как его использовать и настроить, и, конечно же, познакомитесь с самыми полезными командами UNIX.

А мы продолжаем изучение входа в систему. Настала очередь Linux. Если при установке системы вы выбрали запуск в текстовом режиме (на третьем уровне запуска — см. главу 8), то приглашение будет практически таким же, как и в случае с FreeBSD, только в первой строчке будет отображено название дистрибутива Linux:

```
Fedora release 13 (Goddard)
Kernel 2.6.33.3-85.fc13.i686.PAE on an i686 (tty1)
localhost login:
```

Приглашение Linux более информативно. Давайте сравним приглашение FreeBSD и Linux. В первой строчке FreeBSD выводит: название системы (FreeBSD), архитектуру, под которую оптимизирована система (i386), имя узла (den.dkws.org.ua) и имя виртуальной консоли (ttyv0):

```
FreeBSD/i386 (den.dkws.org.ua) (ttyv0)
```

Переключение между виртуальными консолями осуществляется с помощью нажатия Alt + Fn, n — число. Нумерация консолей в FreeBSD начинается с 0, а в Linux с 1. Поэтому чтобы попасть на консоль ttyv1, в Linux нужно нажать Alt + F1, а в FreeBSD — Alt + F2. Нажатие Alt + F1 в FreeBSD переключит вас на консоль ttyv0. Подробно о назначении консолей, о русификации консоли мы поговорим в главе 6.

Первая строка приглашения в Linux сообщает нам только название (Fedora), версию (13) и кодовое имя (Goddard) дистрибутива:

```
Fedora release 13 (Goddard)
```

Зато вторая строка гораздо более информативна. Она сообщает нам версию ядра (2.6.33.3-85), дистрибутив, для которого было собрано ядро (fc13), архитектуру ядра (i686); режим процессора, под который оптимизировано ядро (PAE). После этого идет указание реальной архитектуры, на которой запущена система (i686), и имя виртуальной консоли (tty1). Как видите, имена консолей в FreeBSD (ttyvN) и Linux (ttyN) немного отличаются. Вот та самая вторая строчка, чтобы вы не листали книгу:

```
Kernel 2.6.33.3-85.fc13.i686.PAE on an i686 (tty1)
```

Нужно сделать несколько замечаний. Во-первых, архитектура, под которую оптимизировано ядро, не всегда может совпадать с архитектурой, на которой запущена система. В нашем случае все совпало. Но бывает, что ядро оптимизировано под i386 (для большей совместимости), а система запущена реально на i686.

PAE (Physical Address Extension) — режим работы блока управления памятью x86-совместимых процессоров. В этом режиме используются 64-битные элементы таблицы страниц (хотя на самом деле используются первые 36 бит), с помощью которых процессор может адресовать 64 Гбайт физической памяти. А если бы использовались 32-разрядные таблицы, то процессор мог бы адресовать всего 4 Гбайт «оперативки». В этом и заключается ответ, почему, если вы установили Linux на ваш новый и ультрасовременный сервер с 16 Гбайт оперативной памяти, Linux «видит» всего 4 Гбайт. В таком не очень приятном случае вам нужно либо перекомпилировать ядро, включив поддержку PAE, либо установить другой дистрибутив, где поддержка PAE в ядре включена по умолчанию. Fedora 13 как раз является таким дистрибутивом.

Далее FreeBSD просит ввести имя пользователя:

```
login:
```

В Linux приглашение ввода имени пользователя выглядит иначе:

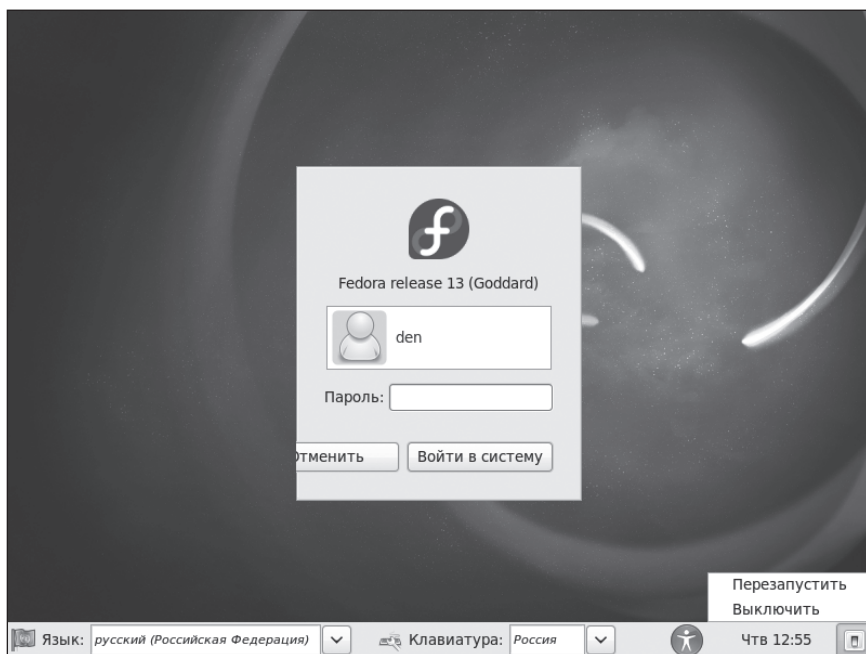
```
localhost login:
```

В этом случае localhost — это имя узла, на котором запущена Linux. FreeBSD выводила имя узла в первой строчке, а Linux выводит в третьей. При локальной



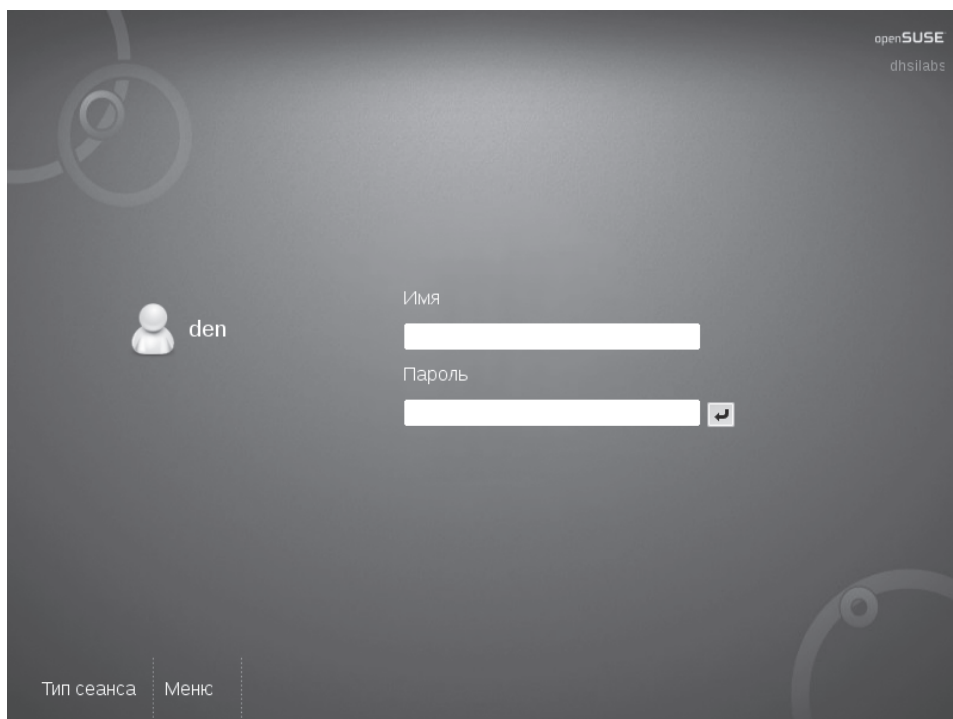
регистрации имя узла не имеет никакого значения, но при удаленной регистрации, особенно когда в вашей сети несколько машин под управлением FreeBSD или Linux, имя узла позволяет узнать машину, к которой вы хотите подключиться.

Большинство современных дистрибутивов Linux по умолчанию настроены на графический вход в систему. На рис. 4.1 изображено окно входа в систему дистрибутива Fedora 13. Сначала вам нужно выбрать пользователя, затем появится поле ввода пароля, как и показано на рис. 4.1. То, что кнопка **Отменить** не поместилась в окне, виноват не я, а разработчики Fedora. Именно поэтому я рекомендую использовать его в качестве серверного дистрибутива. На сервере все равно, как будут выглядеть кнопки, все равно, если некоторые кнопки не переведены на русский язык. А пользователям это не нравится, особенно после «вылизанной» в эстетическом плане Windows. В нижней части окна имеются списки выбора языка и раскладки клавиатуры. Если бы была установлена еще одна графическая среда, например KDE (я обычно устанавливаю одну графическую среду – GNOME), то появилась бы возможность выбрать сеанс (графическую среду). Кнопка в нижнем правом углу позволяет выключить или перезагрузить компьютер без входа в систему.

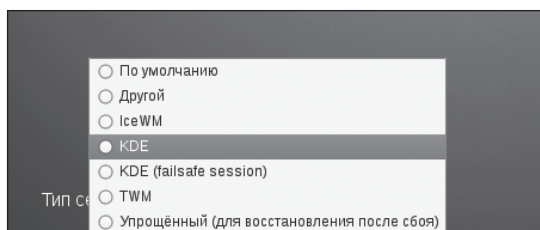


**Рис. 4.1.** Окно регистрации в графическом режиме

Взгляните на окно регистрации дистрибутива openSUSE 11.3 (рис. 4.2). Правда, красивее? Кстати, кнопка **Тип сеанса** позволяет выбрать графическую среду (рис. 4.3)



**Рис. 4.2.** Выход в систему (openSUSE)



**Рис. 4.3.** Выбор графического сеанса (openSUSE)

Разработчики openSUSE уделяют больше внимание мелочам, именно поэтому их дистрибутив подходит как для рабочей станции, так и для сервера.

Даже если Linux загрузилась в графическом режиме, вы можете войти в консоль, нажав **Ctrl + Alt + F1**. Все: теперь вы в текстовом режиме, можете войти в систему и приступить к вводу команд. Хотя если вы новичок, вам будет привычнее оставаться в «графике». Но имейте в виду: чтобы стать настоящим администратором UNIX, переход в консоль неизбежен. Ведь далеко не все административные приложения имеют графическую оболочку, поэтому даже в графическом режиме вам придется вводить команды — в эмуляторе терминала. Но об этом мы поговорим в следующей главе.