



Визитка

ИГОРЬ ШТОМПЕЛЬ, инженер, системный администратор. Сфера профессиональных интересов — свободное программное обеспечение

Резервное копирование в Linux

Консольные инструменты на страже данных

Как известно, лучший способ предотвратить потерю данных — это провести профилактические мероприятия. Одно из них — резервное копирование (Backup)

Важность осуществления резервного копирования и выделения средств на него, как правило не вызывает недоуменных вопросов и пренебрежительного отношения у руководителей организаций, а если и вызывает, то в основном до первой критической ситуации, повлекшей потерю важных данных.

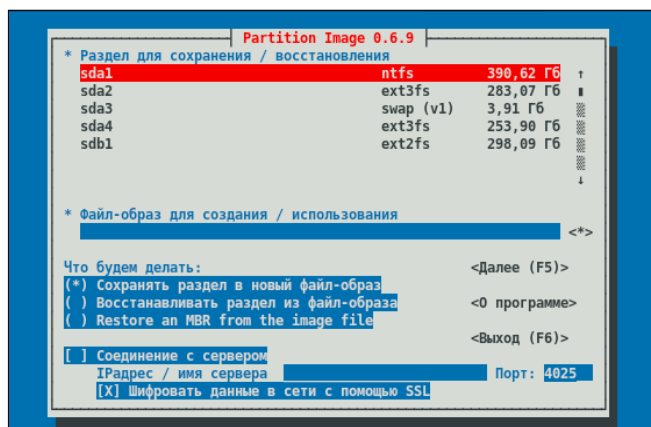
Поэтому нередко регламентация резервного копирования, в том или ином виде, находит отражение в политиках безопасности организаций.

В разных операционных системах доступны, как правило, различные утилиты и программные комплексы для осуществления резервного копирования. Мы же заострим наше внимание на свободных консольных утилитах операционной системы GNU/Linux.

Цель работы

Получение знаний о консольных инструментах операционной системы GNU/Linux, позволяющих осуществлять резервное копирование и его автоматизацию, а также приобретение навыков практического использования этих инструментов с учетом особенностей работы с ними.

Рисунок 1. PartImage запущен



Организация и описание лабораторного стенда

Для успешного выполнения лабораторной работы необходимо организовать рабочий стенд. Нам потребуются один сервер с установленной ОС Linux и внешний жесткий диск. На своей машине я создал следующее разбиение:

- /dev/sda1** – NTFS-раздел;
- /dev/sda2** – раздел с Trisquel GNU/Linux 3.5;
- /dev/sda3** – раздел «подкачки» (вы можете создать его, как и рекомендуется, на логическом, а не первичном разделе);
- /dev/sda4** – раздел с Ubuntu 10.04;
- /dev/sdb1** – внешний жесткий диск (монтируется автоматически).

Также нам понадобится выход в Интернет.

Подготовка стенда

В ходе практической реализации лабораторного стенда нам нужно будет решить следующие задачи:

- > установка необходимых инструментов резервного копирования и автоматизации;
- > подготовка локального хранилища данных.

После решения поставленных задач можно начать непосредственное выполнение лабораторной работы.

Установка необходимых инструментов резервного копирования и автоматизации

Начнем с решения первой задачи. Работа нами будет осуществляться в операционной системе Trisquel 3.5 GNU/Linux, которая совместима с Ubuntu [1].

Итак, для начала проверьте физическое соединение – индикатор сетевой карты, а затем – есть ли выход в Интернет, например, с помощью команды:

```
ping www.ru
```

Поскольку tar, coreutils (нас интересует dd, входящая в его состав) и cron после установки Trisquel уже доступны, нам остается установить только утилиту dump:

```
$ sudo apt-get install dump
```

После выполнения этой команды станут доступны инструменты `dump` (создание архивных копий данных, хранящихся на разделах `ext2`), а также `restore` (работа с архивами, созданными `dump`).

Далее осуществим установку утилиты `PartImage` [2]. Зависимости программы представлены в таблице 1.

Поскольку `PartImage` не доступен из официальных репозиториях Trisquel (впрочем, как и Ubuntu), у нас есть два варианта установки для двух архитектур – мы их рассмотрим ниже. Стоит отметить, что `PartImage` также доступен во многих LiveCD (например, Clonezilla, SystemRescueCD или Recovery Is Possible!), что позволяет избежать его установки и выполнять резервирование/восстановление данных без необходимости запускать ОС, установленную на компьютере (методика известна под термином `bare metal backup`) [3].

Для i386 доступен `deb`-пакет на Launchpad [4]. Загрузим его в домашний каталог, а затем осуществим установку:

```
$ sudo dpkg -i partimage_0.6.7-2ubuntu2_i386.deb
```

Второй вариант – установка `PartImage` на компьютер, имеющий архитектуру `amd64`. В этом случае инсталляция программы будет осуществляться из исходных текстов. Предварительно проверьте, все ли зависимости установлены (см. таблицу 1), а затем произведите установку пакетов для разработчиков этих зависимостей (на этапе сборки `PartImage` понадобятся, например, заголовочные файлы из данных пакетов) – `libbz2-dev`, `libnewt-dev` (в качестве зависимостей автоматически инсталлируются `libslang2-dev`, `zlib1g-dev`) и `libssl-dev`. Только потом переходите непосредственно к установке `PartImage`.

Итак, загрузите исходный код программы, который доступен на официальном сайте [5]. Затем распакуйте архив с `PartImage`:

```
tar -jxvf partimage-0.6.9.tar.bz2
```

Затем перейдите в каталог, в который вы распаковали архив с программой, и дайте следующую команду:

```
./configure
```

После успешного выполнения команды введите и выполните следующее:

```
make
```

А затем:

```
$ sudo make install
```

Проверить, успешно ли была завершена установка, можно, дав команду:

```
$ sudo partimage
```

В моем случае было отображено окно, показанное на рис. 1. Работу с программой мы рассмотрим позже, поэтому пока нажмите `<F6>` – для выхода.

Теперь можно переходить к решению второй задачи.

Подготовка локального хранилища данных

Объем сетевого накопителя, выступающего в качестве физического места хранения резервных данных, зависит от объема этих данных и среднего объема приращения данной информации в сутки. Таким образом, примерный расчет требуемого пространства для хранения резервных копий данных может осуществляться по следующей формуле:

$$V_{st} = V_s + (M_d \cdot T_d)$$

где:

V_{st} – объем хранилища в Гб/Мб;

V_s – объем данных, требующих резервного копирования;

M_d – среднесуточный объем приращения данных Гб/Мб;

T_d – время в днях.

В моей ситуации объем хранилища составляет 320 Гб (327 680 Мб), в то время как $V_s = 7168$ Мб, $M_d = 25$ Мб, $T = 365$. Итого объем хранилища, требуемый для текущего года, – 16 293 Мб. Таким образом, хранилища такого объема (320 Гб) хватит надолго, но в то же время необходимо учитывать срок службы носителей информации.

Кроме того, стоит учитывать, что жесткий диск – не единственный (и, возможно, не самый подходящий) доступный носитель для хранения резервных копий данных. Например, можно использовать оптические диски, ленты,

Таблица 1. Зависимости `PartImage`

Название на официальном сайте	Название пакета в Trisquel 3.5	Описание	Официальный сайт
libz	zlib	Библиотека, позволяющая осуществлять сжатие и распаковку данных в формате zip	http://www.zlib.net
libbz2	libbz2	Библиотека, позволяющая осуществлять сжатие и распаковку данных в формате bzip2	http://www.bzip.org
liblzo	liblzo2	Библиотека, используемая для сжатия данных посредством алгоритма LZO	http://www.oberhumer.com/opensource/lzo/
libnewt	libnewt0.52	Библиотека, позволяющая реализовывать окна в текстовом режиме (цветной текстовый режим, полосы прокрутки, кнопки, перекрывающиеся окна, поля ввода и т. д.) и являющаяся частью набора инструментов Newt	https://fedorahosted.org/newt
libslang	libslang2	Динамическая версия библиотеки S-Lang, позволяющая создавать «дружественные» для пользователя кроссплатформенные приложения	http://www.s-lang.org
libssl и libcrypto	libssl0.9.8	Библиотеки, позволяющие <code>PartImage</code> осуществлять шифрование. Обе являются частью проекта OpenSSL	http://www.openssl.org

zip-дискеты и прочее, но рассмотрение проблемы выбора подходящего носителя находится за рамками данной лабораторной работы.

Кстати, о времени работы с локальным хранилищем. Если мы обратим приведенную выше формулу, то сможем рассчитать количество дней, на которое хватит объема нашего хранилища:

$$Td = (Vst - Vs) / Md$$

Подставив данные из примера выше, мы получим в результате более 35 лет. Но это конкретно для описанной ситуации, естественно, объемы и время могут существенно варьироваться.

Что касается файловых систем, подходящих для использования на нашем локальном хранилище, то мы в определенной степени ограничены в их выборе средствами резервного копирования, которые будем применять, – речь идет о поддержке тех или иных файловых систем в последних.

Забегая вперед скажу, что наиболее подходящим вариантом станет стандартная файловая система ext2. Она поддерживается всеми инструментами резервного копирования, которые мы будем рассматривать подробнее далее.

Кстати, стандартные внешние жесткие диски (USB) поставляются с файловой системой NTFS. Думаю, она не подойдет для решения поставленных в лабораторной задаче – необходима файловая система, которая бы поддерживалась в полной мере всеми рассматриваемыми приложениями.

В нашем случае, например, это условие не выполняется для PartImage (см. таблицу 2) – поддержка NTFS носит экспериментальный характер. Хотя Linux-драйвер NTFS-3g давно и успешно выполняет запись файлов на NTFS, эта файловая система все же не является родной для Linux, и потому от ее использования в нашей лабораторной работе мы воздержимся.

Задействуем внешний жесткий диск в качестве локального хранилища. Создадим на нем раздел с файловой системой ext2. Это можно сделать с помощью консольного инструмента fdisk, работу с которым мы описывали ранее [6].

Итак, в качестве локального хранилища будет использоваться внешний жесткий диск, который подключен через USB к серверу (устройство в системе – /dev/sdb1).

Ознакомление с основными возможностями резервного копирования из консоли и его автоматизации

Для начала рассмотрим возможности утилиты tar как средства резервного копирования [7].

Резервные копии данных будут сохраняться в каталог /dirbrp внешнего жесткого диска (/mnt/sdb1), подключенного к нашему компьютеру.

Например, пусть рабочие данные хранятся в каталоге /home/user/datadir. Тогда создание резервной копии с помощью tar может выглядеть следующим образом:

```
tar -zcf /mnt/sdb1/datadir.tar.gz /home/user/datadir
```

где опции:

z – позволяет произвести дополнительную архивацию с использованием gzip (tar сам не создает сжатых архивов, а использует для этого сторонние утилиты – например, gzip или bzip2. В свою очередь, gzip, как и bzip2 (опция -j), не в состоянии сжимать несколько файлов одновременно, поэтому tar используется для сбора всех архивируемых файлов в один, а затем gzip или bzip2 сжимают его. Кроме того, tar может создавать архивы и в других форматах, например, LZMA (--lzma). Последний отличается высоким коэффициентом сжатия (алгоритм Lempel-Ziv-Markov, алгоритм реализован также и в 7-Zip);

Рисунок 2. Постраничный просмотр содержимого tar-архива

```
user@userdesk:~$ tar -ztvpf etc.tar.gz | more
drwxr-xr-x root/root      0 2010-09-22 10:29 etc/
-rw-r--r-- root/root      12 2009-02-26 18:58 etc/debian_version
drwxr-xr-x root/root      0 2010-03-18 01:44 etc/update-motd.d/
lrwxrwxrwx root/root      0 2010-07-28 11:01 etc/update-motd.d/91-release_upgrade -> /usr/lib/update-manager
/check-new-release
lrwxrwxrwx root/root      0 2010-07-28 11:01 etc/update-motd.d/99-reboot-required -> /usr/lib/update-notifie
r/update-motd-reboot-required
-rwxr-xr-x root/root     149 2009-07-14 05:25 etc/update-motd.d/00-header
lrwxrwxrwx root/root      0 2010-07-28 11:01 etc/update-motd.d/90-updates-available -> /usr/lib/update-notif
ier/update-motd-updates-available
-rw-r--r-- root/root      899 2009-06-08 14:24 etc/gssapi_mech.conf
-rw-r----- root/shadow    939 2010-09-22 10:29 etc/shadow
drwxr-xr-x root/root      0 2010-03-18 01:43 etc/security/
-rwxr-xr-x root/root     1020 2009-09-04 12:25 etc/security/namespace.init
-rw-r----- root/root      0 2010-03-18 01:43 etc/security/opasswd
-rw-r--r-- root/root     2180 2009-09-04 12:25 etc/security/time.conf
-rw-r--r-- root/root      419 2009-09-04 12:25 etc/security/sepermit.conf
-rw-r--r-- root/root     4620 2009-09-04 12:25 etc/security/access.conf
-rw-r--r-- root/root     2980 2009-09-04 12:25 etc/security/pam_env.conf
-rw-r--r-- root/root     2151 2009-09-04 12:25 etc/security/limits.conf
-rw-r--r-- root/root     1442 2009-09-04 12:25 etc/security/namespace.conf
-rw-r--r-- root/root     3456 2009-09-04 12:25 etc/security/group.conf
drwxr-xr-x root/root      0 2010-03-18 01:45 etc/cron.hourly/
-rw-r--r-- root/root      102 2009-09-15 17:12 etc/cron.hourly/.placeholder
-rw-r----- root/root      0 2010-03-18 01:43 etc/.pwd.lock
drwxr-xr-x root/root      0 2010-09-22 10:29 etc/rc5.d/
lrwxrwxrwx root/root      0 2010-07-28 11:01 etc/rc5.d/S99ondemand -> ../init.d/ondemand
lrwxrwxrwx root/root      0 2010-07-28 11:01 etc/rc5.d/S50pulseaudio -> ../init.d/pulseaudio
lrwxrwxrwx root/root      0 2010-07-28 11:01 etc/rc5.d/S20samba -> ../init.d/samba
lrwxrwxrwx root/root      0 2010-07-28 11:01 etc/rc5.d/S99acpi-support -> ../init.d/acpi-support
lrwxrwxrwx root/root      0 2010-07-28 11:01 etc/rc5.d/S99grub-common -> ../init.d/grub-common
lrwxrwxrwx root/root      0 2010-07-28 11:01 etc/rc5.d/S70pppd-dns -> ../init.d/pppd-dns
```

- c** – создание архива;
- f** – имя файла, в котором будет создан архив.

Просмотреть содержимое архива можно с помощью следующей команды:

```
tar -ztvpf datadir.tar.gz
```

Если каталог содержит значительное число файлов, лучше использовать конвейер:

- > для страничного просмотра файлов архива, как показано на рис. 2:

```
tar -ztvpf gdata-1.3.0.tar.gz | more
```

- > для перенаправления вывода в текстовый файл:

```
tar -ztvpf gdata-1.3.0.tar.gz > arch01
```

Стоит иметь в виду, что для сжатого архива большого размера процесс получения листинга будет очень долгим, так как подразумевает распаковку внешнего архива.

Для восстановления данных из резервной копии (в данном случае извлечения из tar-архива) используется опция -x:

```
tar -xzf etc.tar.gz
```

Теперь рассмотрим создание инкрементальных резервных копий средствами tar. Но вначале несколько слов об инкрементальном резервном копировании.

Итак, инкрементальное резервное копирование – это способ резервного копирования, при котором на первом шаге создается полная копия резервируемых данных, а на последующих шагах в нее добавляются только измененные данные.

Вначале создадим полную резервную копию каталога /etc:

```
$ sudo tar -cvfz etc.0.tar.gz -g etc.snar /etc
```

здесь:

- etc.0.tar.gz** – полная копия каталога /etc;
- опция -g** – инкрементальный бэкап;
- etc.snar** – бинарный файл, содержащий информацию о последнем состоянии файлов в каталоге /etc;
- /etc** – каталог для резервирования.

Команда добавления в архив только измененной с последнего резервного копирования информации выглядит следующим образом:

```
$ sudo tar -cvfz etc.1.tar.gz -g etc.snar /etc
```

Восстановить каталог /etc из последней резервной копии можно следующим образом:

```
$ sudo tar zxvf etc.1.tar.gz
```

На этом мы завершим рассмотрение работы с программой tar.

dd как средство резервного копирования

Рассмотрим еще один отличающийся гибкостью инструмент, позволяющий осуществлять резервное копирование, – утилиту dd (dataset definition – определение набора данных) [8]. Среди возможностей программы: создание резервной копии MBR (master boot record, главной загрузочной записи), создание резервной копии жесткого диска сразу на другом диске, создание образа резервируемого диска и другие.

Некоторые опции dd:

bs – задает размер блока;

count – количество блоков.

Приведем примеры работы с утилитой dd. Создадим резервную копию MBR с первого диска нашего сервера:

```
dd if=/dev/sda of=boot.mbr bs=512 count=1
```

где:

if=/dev/sda – входной файл (в данном случае первый жесткий диск сервера);

of=boot.mbr – выходной файл (резервная копия MBR);

bs – задает размер блока в байтах;

count – количество блоков.

Еще один пример – резервирование первого жесткого диска сервера в файл-образ на внешний жесткий диск:

```
sudo dd if=/dev/sda | bzip2 > /media/sdb1/serverf.tar.bz
```

Использование PartImage

PartImage – это свободное программное обеспечение (лицензия GNU GPLv2), которое позволяет осуществлять резервное копирование разделов жесткого диска. Программа поддерживает различные файловые системы, но на разном уровне (см. таблицу 2). В отличие от dd PartImage не может не зависеть от типа файловой системы, потому что копирует только занятое пространство раздела. Как видно из таблицы 2, наибольшую поддержку в PartImage получили файловые системы Linux, хотя поддержка ext4 и Btrfs отсутствует.

Резервное копирование разделов осуществляется посредством посекторного копирования в файл-образ. В процессе информация может подвергаться сжатию в целях экономии времени и дискового пространства (по заявлениям разработчиков на главной странице официального сайта программы, полностью заполненный раздел размером 1 Гб

Таблица 2. Уровень поддержки PartImage различных операционных систем

Уровень поддержки	Название файловой системы	Описание
Стабильный	ext2/ext3	Стандартная файловая система Linux
	ReiserFS	Журналируемая файловая система Linux
	FAT16/32	Файловые системы DOS и Windows
	HPFS	Файловая система IBM OS/2
	JFS	Журналируемая файловая система из AIX
	XFS	Журналируемая файловая система из Irix
Бета	UFS	Файловая система, используемая в *NIX
	HFS	Используется в MacOS
Экспериментальный	NTFS	Файловая система Windows NT

может быть сжат до 400 Мб), а также может быть разделена на части для записи на CD/DVD.

У PartImage есть одно достаточно неудобное ограничение – он не будет осуществлять резервное копирование раздела, который примонтирован. Если вы запускаете PartImage с LiveCD, данная проблема не стоит, но платой за это является необходимость остановить (перезагрузить) сервер для выполнения резервного копирования. Настала пора испробовать PartImage в деле (поскольку, как видно на рис. 1, программа имеет поддержку русского языка, подробное разъяснение всех возможностей, предлагаемых PartImage (путем выбора опций в окнах), не требуется).

Итак, запустим его:

```
$ sudo partimage
```

Далее с помощью клавиш курсора выбираем раздел /dev/sda4 (у вас может быть другой раздел, важно, чтобы это был не тот раздел, на который установлена активная в данный момент система). После наведения полосы-курсора на нужный раздел для его окончательного выбора нажмите клавишу управления курсором «влево». Раздел для «сохранения/восстановления» будет выбран, и курсор автоматически переместится в поле «Файл-образ для создания/использования».

В данном поле мы можем ввести имя файла-образа, в который будет помещена резервная копия раздела, на пример:

```
/media/***/sda4.gz
```

где *** – каталог автоматического монтирования внешнего жесткого диска в /media.

Или же вы можете воспользоваться встроенным файловым менеджером PartImage: нажмем клавишу управления курсором «вниз», курсор из окна присвоения файлу-образу имени переместится на элемент <*>, а последний будет подсвечен красным цветом. Нажатие клавиши <Enter> на нем приведет к запуску файлового менеджера (см. рис. 3).

Перейдем в /media/, а затем в каталог, где примонтирован внешний жесткий диск. Далее перемещаем курсор на «OK» и клавишей <Enter> подтверждаем свой выбор. После этого, мы опять окажемся в главном окне PartImage. В поле «Файл-образ для...» будет автоматически прописан путь к каталогу монтирования внешнего жесткого диска (/media/***/ – см. выше) – остается только дописать имя файла – sda4.gz.

Больше изменять в этом окне нам ничего не потребуется, поэтому нажимаем <F5> для продолжения. Появится окно «Сохранение раздела в файл образа» (см. рис. 4).

Рисунок 3. Файловый менеджер PartImage в работе

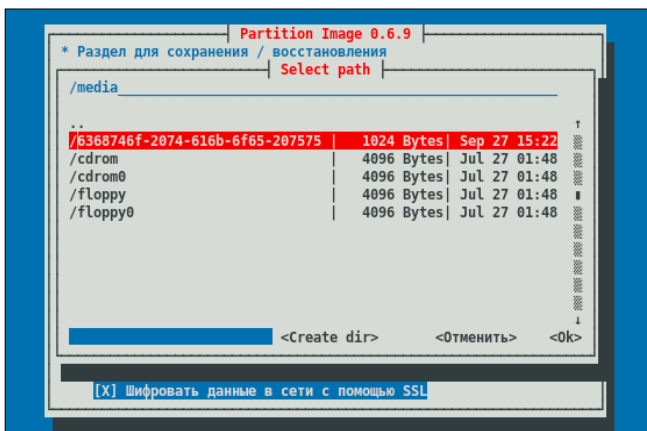


Рисунок 5. Подготовка к созданию файла-образа

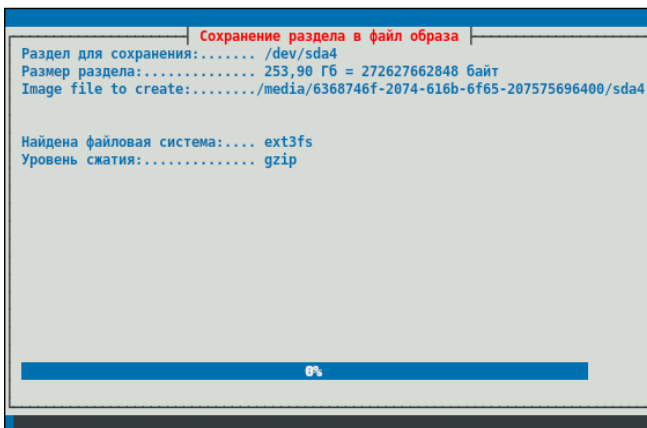


Рисунок 4. Окно «Сохранение раздела в файл образа»

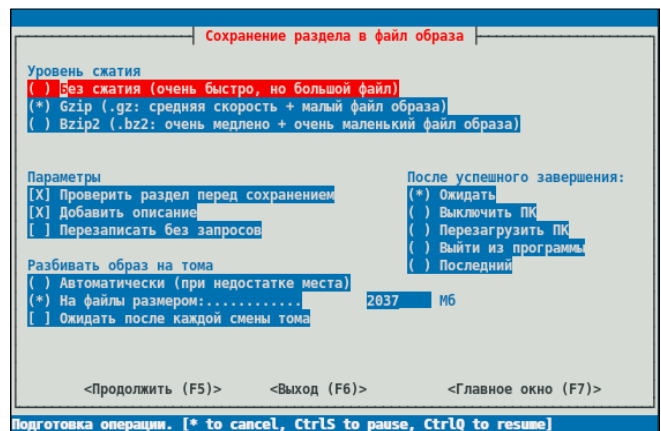
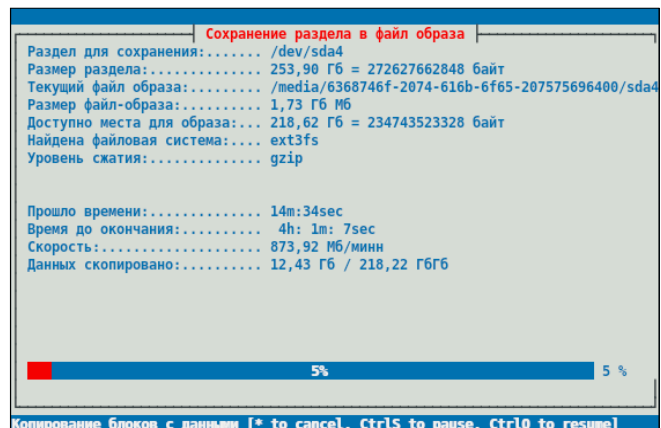


Рисунок 7. Сохранение раздела в файл образа



Как видно из рис. 4, в данном окне мы можем управлять следующими опциями:

- > уровнем сжатия (оставим gzip – средняя скорость + малый размер файла образа);
- > дополнительными параметрами (оставим «Проверять раздел перед сохранением» и «Добавить описание» файла-образа);
- > разбиением файла-образа на тома (оставим по умолчанию);
- > поведение программы после завершения выполнения всех операций (ожидать, выключить ПК, перезагрузить ПК и т.д. – также оставим по умолчанию).

Для продолжения нажимаем <F5>. Появится окно «Описание раздела» (вспомним, при установке дополнительных параметров мы оставили флажок «Добавить описание», установленный по умолчанию). Введем описание и нажмем «ОК». Начнутся подготовительные операции по созданию файла-образа (см. рис. 5).

После чего будет выведена информация о файловой системе раздела, который станет копироваться в файл-образ (см. рис. 6).

Нажимаем «ОК». Начнется процесс «Сохранение раздела в файл образа» (см. рис. 7).

После удачного завершения создания файла-образа PartImage выведет окно «Выполнено», в котором сообщит об успешном окончании, затраченном времени, скорости выполнения в Мб/мин, объем скопированных данных (см. рис. 8). Все, создание резервной копии раздела успешно завершено.

Теперь восстановим раздел из созданной нами резервной копии. Для этого запустим программу:

```
$ sudo partimage
```

Затем в главном окне программы выбираем раздел для восстановления sda4. Далее указываем файл-образ для восстановления (он будет находиться на /dev/sdb1. Кроме того, поскольку мы выбрали при резервном копировании разбиение файла-образа на тома, имена последних получили следующий формат – sda4.gz.***, где *** – число начиная с 000) – sda4.gz.000.

В разделе «Что будем делать:» выбираем «Восстановить раздел из файла-образа». Затем нажимаем клавишу <F5>.

Появится окно с описанием раздела (см. выше – параметр «Добавить описание» – все, что мы ввели на этапе создания резервной копии о данном разделе, сейчас и появится). Нажимаем «ОК» и попадаем в окно «Восстановление раздела из файла-образа».

В разделе «Параметры» (см. рис. 9) мы можем выбрать имитирование процесса восстановления, а также перезапись пустых блоков нулями, а также в разделе «После успешного завершения:», как и на этапе создания резервной копии, – поведение после успешного завершения всех операций.

Нажимаем <F5>. Появится окно с информацией по файловой системе раздела из файла-образа. Нажмем «ОК», и процесс восстановления начнется.

Итак, закончим рассмотрение работы с PartImage.

Автоматизация резервного копирования средствами cron

На этом завершим обзор инструментов резервного копирования и заострим наше внимание на утилите, позволяющей автоматизировать этот процесс, – cron. Cron работает в режиме демона и используется для выполнения различных действий по расписанию [9]. Конфигурационный файл программы – /etc/crontab. Если откроете его, то увидите следующее (возможно, у вас данный файл будет несколько отличаться, для удобства я добавил номера строк):

```
1. SHELL=/bin/sh
2. PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:
   /usr/sbin:/usr/bin
3.
4. # m h dom mon dow user command
5. 17 * * * * root run-parts /etc/cron.hourly
6. 25 6 * * * root run-parts /etc/cron.daily
7. 47 6 * * 7 root run-parts /etc/cron.weekly
8. 52 6 1 * * root run-parts /etc/cron.monthly
```

Давайте рассмотрим конфигурацию cron подробнее.

Итак, первые две строки задают переменные окружения – это указание использовать оболочку /bin/sh для выполнения команд (SHELL), а также указание путей, которые будут использоваться при выполнении команд (PATH).

Четвертая строка является комментарием и содержит описание формата заданий cron:

m – минуты (значение – целое от 0 до 59);

Рисунок 6. Информация о файловой системе

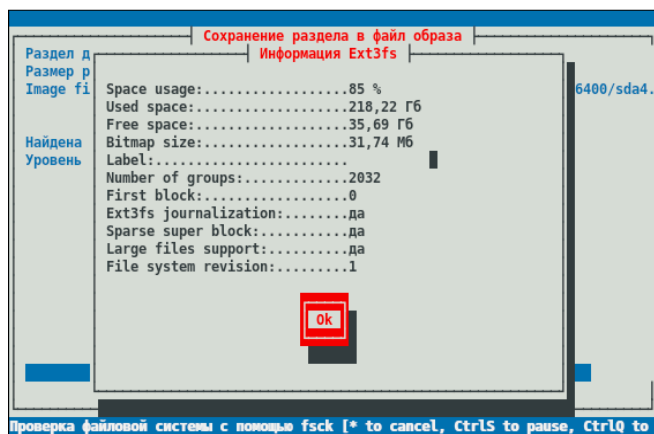
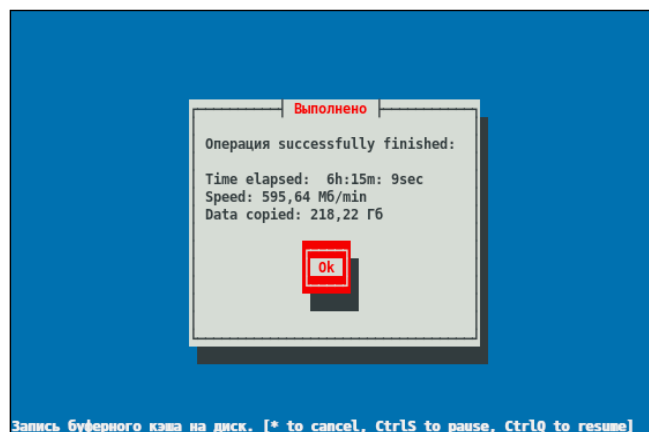


Рисунок 8. Окно PartImage «Выполнено»



h – часы (значение – целое от 0 до 23);

dom – день месяца (day of month, значение – от 1 до 31, но надо иметь в виду, что если вы задали значение месяца, то день не должен выходить за его пределы – например, недопустимо 31 февраля);

mon – месяц (значение – целое от 1 до 12 либо сокращения – jan, feb и т.д.);

dow – день недели (day of week);

user – пользователь (целое от 0 до 7, 0 и 7 – это воскресенье либо сокращения – sun и т.д.);

command – выполняемая команда (например, netstat > 01 либо команда, запускающая скрипт).

Пятая строка – запуск содержимого каталога /etc/cron.hourly на 17-й минуте каждого часа.

Шестая строка – запуск содержимого каталога /etc/cron.daily каждый день в 6:25.

Седьмая строка 7 – запуск содержимого каталога /etc/cron.weekly в 6:47 каждое воскресенье.

Восьмая строка – запуск содержимого каталога /etc/cron.monthly в 6:52 каждый первый день месяца.

Создадим задание cron. Для этого используется утилита crontab, где опции:

-e – редактирование файла crontab текущего пользователя;

-l – просмотреть файл crontab текущего пользователя;

-r – удаление файла crontab текущего пользователя.

Создадим задание cron для текущего пользователя:

```
crontab -e
```

В ходе выполнения данной команды cron создаст временный файл в каталоге /tmp, а последний затем будет открыт на редактирование в текстовом редакторе, указанном в переменной \$EDITOR (на моей системе это Nano). В заголовке программы можно увидеть полный путь к временному файлу, созданному cron.

Пропишем в данном файле следующее:

```
30 17 * * * /mnt/sdb1/datadir.tar.gz /home/user/datadir
```

что означает осуществлять создание резервной копии каталога /home/user/datadir каждый день в 17:30.

После сохранения и выхода из текстового редактора в каталоге /var/spool/cron/crontabs будет создан crontab-файл

текущего пользователя с именем, совпадающим с именем пользователя. Для принятия изменений необходимо перезапустить демон cron:

```
$ sudo restart cron
```

На этом завершим рассмотрение работы с cron.

Задание на лабораторную работу

Определим действия, которые необходимо выполнить, опираясь на созданный нами лабораторный стенд.

Задание 1

Создайте резервную копию данных с использованием утилиты tar, а также с дополнительной архивацией посредством bzip2, lzma, xz (соответствующие опции вы можете найти в страницах руководства tar: команда man tar) и выводом списка обработанных файлов.

Задание 2

Произведите поиск в tar-архиве какого-либо из файлов с использованием утилиты grep.

Задание 3

Восстановите из резервной копии (архив tar) ряд отдельных файлов (имена необходимых для извлечения файлов указываются после имени tar-архива).

Задание 4

Создайте инкрементную резервную копию данных каталогов /etc, /bin, /usr с использованием утилиты tar.

Задание 5

Восстановите данные из инкрементной резервной копии, созданной в результате выполнения задания 5.

Задание 6

Осуществите резервное копирование жесткого диска сервера с использованием утилиты dd сразу на внешний жесткий диск.

Задание 7

Создайте резервную копию MBR локального жесткого диска.

Задание 8

Создайте резервную копию жесткого диска сервера в файле-образе (с использованием gzip) на внешнем жестком диске.

Задание 9

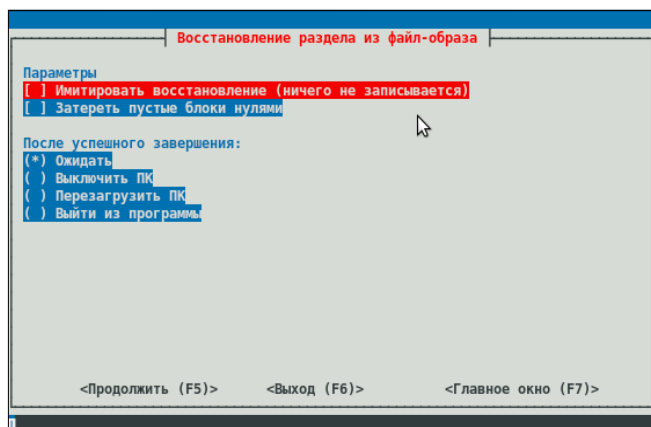
Создайте резервную копию раздела жесткого диска с помощью PartImage, следующим образом:

- > отключите добавление описания файла-образа;
- > установите разбиение образа на тома автоматическое;
- > после успешного завершения перезагрузить ПК.

Задание 10

Осуществите восстановление раздела с помощью PartImage из резервной копии, созданной в результате выполнения задания 7.

Рисунок 9. Окно «Восстановление раздела из файла-образа»



Задание 11

Создайте резервную копию раздела жесткого диска с помощью PartImage, следующим образом:

- > выполните резервирование без сжатия;
- > при автоматическом разбиении образа на тома установить «Ожидание после каждой смены тома»;
- > после успешного завершения выйти из программы.

Задание 12

Осуществите имитацию восстановления раздела с помощью PartImage из резервной копии, созданной в результате выполнения задания 9.

Задание 13

Осуществите восстановление раздела с помощью PartImage из резервной копии, созданной в результате выполнения задания 9.

Задание 14

С помощью cron и tar запланируйте создание резервной копии каталога с данными: в 19:00 в понедельник, среду, пятницу.

При выполнении данной лабораторной работы вы получили знания об основных возможностях осуществления резервного копирования и его автоматизации из консоли в операционной системе GNU/Linux, а также приобрели навыки использования соответствующего программного обеспечения (tar, dd, PartImage и cron).

Использование гибких консольных инструментов позволило нам расширить арсенал доступных средств, а также включить их в процесс планирования и осуществления резервного копирования в будущей профессиональной деятельности. **ЕОБ**

1. Официальный сайт операционной системы Trisquel GNU/Linux – <http://trisquel.info>.
2. Официальный сайт утилиты PartImage – http://www.partimage.org/Main_Page.
3. Официальные сайты операционных систем:
Clonezilla – <http://clonezilla.org>;
SystemRescueCD – http://www.sysresccd.org/Main_Page;
Recovery Is Possible! – <http://www.tux.org/pub/people/kent-robotti/looplinux/rip>;
Статья о Bare-metal restore в английской части Wikipedia – http://en.wikipedia.org/wiki/Bare-metal_restore.
4. PartImage в виде deb-пакета для архитектуры i386 на Launchpad – <https://launchpad.net/ubuntu/+source/partimage>.
5. Официальная страница загрузки PartImage (в том числе, исходного кода) – <http://www.partimage.org/Download>.
6. Штомпель И. Что под «капотом»? Работа с HDD и файловыми системами из консоли. //«Системный администратор», № 1-2, 2010 г. – С. 39-40 – описание работы с fdisk.
7. Официальный сайт утилиты tar – <http://www.gnu.org/software/tar>.
8. Официальная страница, посвященная Coreutils – <http://www.gnu.org/software/coreutils>; dd в русской версии Wikipedia – <http://ru.wikipedia.org/wiki/Dd>.
9. Страница русской Wikipedia, посвященная cron – <http://ru.wikipedia.org/wiki/Cron>.



Информационные Технологии для Страхового Рынка

<SCRIPT 'n' SURE> SUMMIT-2011

Information Technologies for the Insurance Industry

5 апреля 2011

Москва, «Swissotel Красные Холмы»

Информационные партнеры

Системный администратор

Условия участия:

- участие для представителей страховых компаний бесплатное. Организаторы оставляют за собой право ограничить состав участников;
- участие для консалтинговых и ИТ-компаний возможно только в рамках партнерства по данному мероприятию.

Организатор



По вопросам участия, пожалуйста, обращайтесь в редакцию журнала «Русский полис»

тел.: +7(495)612-1093, +7(495)612-8118,
+7 (495) 612-7457

факс: (495) 612-81-28

Контактные лица:

KSmirnova@in-sure.ru – Ксения Смирнова
Orlova@in-sure.ru – Елена Орлова

www.in-sure.ru

Реклама