Dump/Resore

Сейчас попробуем вместе рассмотреть варианты создания архивных копий.

Дамп в файл.

```
dump -0aLuf /samba/root.dump /
где 0 – максимальный уровень сохранения информации при дампе, он гарантирует,
что вся информация о системе будет сохранена
a – авторазмер (пропуск всех проверок длины лент, пишет пока хватает места)
L – ключ, используемый при снятии дампа с «живой» файловой системы.
u – обновление файла dumpdates
f – сохраняет дамп в файл
/samba/root.dump – путь к файлу с дампом и, собственно, сам файл с дампом
/ - файловая система для дампа (можно заменить на /dev/ad0s1a)
```

Листинг 01.

Дамп в файл.

```
iv# dump -0aLuf /samba/root.dump /
 DUMP: Date of this level 0 dump: Mon Nov 17 09:53:14 2008
 DUMP: Date of last level 0 dump: the epoch
 DUMP: Dumping snapshot of /dev/ad4s1a (/) to /samba/root.dump
 DUMP: mapping (Pass I) [regular files]
 DUMP: mapping (Pass II) [directories]
 DUMP: estimated 84435 tape blocks.
 DUMP: dumping (Pass III) [directories]
 DUMP: dumping (Pass IV) [regular files]
 DUMP: DUMP: 85364 tape blocks on 1 volume
 DUMP: finished in 9 seconds, throughput 9484 KBytes/sec
 DUMP: level 0 dump on Mon Nov 17 09:53:14 2008
 DUMP: Closing /samba/root.dump
 DUMP: DUMP IS DONE
```

```
iv# ls -l /samba/root.dump
-rw-r--r-- 1 root wheel 87408640 Nov 17 09:53 /samba/root.dump
```

iv#

Дамп в файл с архивацией.

```
dump –0aLu -f - / | gzip -9 > /samba/root.dump.gz
(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)
где 1 – команда дампа
2 – ключи команды, описаны чуть выше
3 – ключ вывода в файл
4 – стандартный вывод
5 – предмет архивации – рутовый раздел
6 – пайп smile
7 – команда архивирования gzip
8 – степень сжатия «-9» (для более подробного в man gzip)
9 – перенаправление вывода в файл
10 – файл в который перенаправлен вывод. Должен оканчиваться на gz.
```

Листинг 02.

Дамп в файл с сжатием файла.

```
iv# dump -0aLu -f - / | gzip -9 > /samba/root.dump.gz
 DUMP: Date of this level 0 dump: Mon Nov 17 09:54:20 2008
 DUMP: Date of last level 0 dump: the epoch
 DUMP: Dumping snapshot of /dev/ad4s1a (/) to standard output
 DUMP: mapping (Pass I) [regular files]
 DUMP: mapping (Pass II) [directories]
 DUMP: estimated 84435 tape blocks.
 DUMP: dumping (Pass III) [directories]
 DUMP: dumping (Pass IV) [regular files]
 DUMP: DUMP: 85364 tape blocks
 DUMP: finished in 40 seconds, throughput 2134 KBytes/sec
 DUMP: level 0 dump on Mon Nov 17 09:54:20 2008
 DUMP: DUMP IS DONE
```

```
iv# ls -l /samba/root.dump.gz
-rw-r--r--  1  root  wheel  33622590  Nov  17  09:55 /samba/root.dump.gz
```

Дамп на фтп сервер

Грамотным решением будет соблюдение безопасности – ограничение

для всех подключений для логина ”epbackup” – никто не обидится.

Листинг 03.

Проверка песочницы для подключений по фтп для логина epbackup.

```
www# cat /etc/ftpchroot | grep epbackup
epbackup /var/backup/today
www#
```

После этого можно дампить:

```
dump -0aLuf - / | ftp -u ftp://epbackup:pamelafktyf@192.168.0.2/root.dump -
(1) (2) (3)(4) (5) (6) (7) (8) (9)
где 1 – команда дампа
2 – ключи к команде (описания смотри выше)
3 – стандартный вывод
4 – предмет дампа – рутовый раздел
5 – пайп
6 – команда ftp
7 – ключ команды. В мане написано что ключ ”-u” нудно вписывать при использовании ссылки : «-u URL file [...]»
8 – собственно ссылка. Состоит из следующих частей: [протокол]://[логин]:[пароль]@[сервер]/[путь/][файл_дампа]
9 – значок тире (стандартный вывод для дампа)
```

Листинг 04.

Дамп с авторизацией и закачкой на фтп.

```
energoproekt# dump -0aLuf - / | ftp -u ftp://epbackup:pamelafktyf@192.168.0.2/root.dump -
Connected to 192.168.0.2.
220 www.enproekt.local FTP server (Version 6.00LS) ready.
331 Password required for epbackup.
230 User epbackup logged in, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
200 Type set to I.
```

```
remote: root.dump
229 Entering Extended Passive Mode (|||61745|)
150 Opening BINARY mode data connection for 'root.dump'.
 DUMP: Date of this level 0 dump: Mon Nov 17 09:58:05 2008
 DUMP: Date of last level 0 dump: the epoch
 DUMP: Dumping snapshot of /dev/mirror/gm0s1a (/) to standard
output
 DUMP: mapping (Pass I) [regular files]
 DUMP: mapping (Pass II) [directories]
 DUMP: estimated 189380 tape blocks.
 DUMP: dumping (Pass III) [directories]
 DUMP: dumping (Pass IV) [regular files]
 DUMP: DUMP: 192006 tape blocks
 DUMP: finished in 21 seconds, throughput 9143 KBytes/sec
 DUMP: level 0 dump on Mon Nov 17 09:58:05 2008
 DUMP: DUMP IS DONE
226 Transfer complete.
196608000 bytes sent in 00:21 (8.63 MB/s)
energoproekt#
```

Листинг 05.

Проверка размера файла на фтп-сервере.

```
www# ls -l root.dump
-rw-r--r-- 1 epbackup wheel 196608000 Nov 17 09:58 root.dump
www#
```

Дамп на фтп сервер со сжатием файла.

```
dump -0aLuf - / | gzip -9 - | ftp -u
ftp://epbackup:pamelafktyf@192.168.0.2/root.dump.gz -
```

Расписывать подробно не буду, в предыдущих пунктах написал вроде бы достаточно подробно, единственное что стоит упомянуть – здесь ставим второй пайп, чтобы вывод команды дамп пошел на вход в качестве аргумента команды ”gunzip -9”, а уже через следующий пайп пошел на вход команде ”ftp -u”. Нагородили в общем :).

В общем, опять же, для экономии места можно сжимать дампы. Жмутся они хорошо, почему бы этим и не воспользоваться. Жать

можно двумя способами: либо на самом сервере с которого делаются дампы, прямо здесь жмём и заливаем на фтп уже зажатый файл, либо жать уже на целевом сервере. Я напишу здесь про первый вариант, считаю его более правильным. Зачем писать несколько скриптов, если всё можно сделать одним. Соответственно будет похоже на строчку в разделе «дамп на фтп», только через пайп пропустим ещё и gunzip, а в названии конечного файла не забудем поставить gz. Итак:

Листинг 06.

Дамп на фтп сервер со сжатием файла.

```
energoproekt# dump -0aLuf - / | gzip -9 - | ftp -u
ftp://epbackup:pamelafktyf@192.168.0.2/root.dump.gz -
Connected to 192.168.0.2.
220 www.enproekt.local FTP server (Version 6.00LS) ready.
331 Password required for epbackup.
230 User epbackup logged in, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
200 Type set to I.
remote: root.dump.gz
229 Entering Extended Passive Mode (|||54755|)
150 Opening BINARY mode data connection for 'root.dump.gz'.
 DUMP: Date of this level 0 dump: Mon Nov 17 10:16:52 2008
 DUMP: Date of last level 0 dump: the epoch
 DUMP: Dumping snapshot of /dev/mirror/gm0s1a (/) to standard
output
 DUMP: mapping (Pass I) [regular files]
 DUMP: mapping (Pass II) [directories]
 DUMP: estimated 189380 tape blocks.
 DUMP: dumping (Pass III) [directories]
 DUMP: dumping (Pass IV) [regular files]
 DUMP: DUMP: 192006 tape blocks
 DUMP: finished in 53 seconds, throughput 3622 KBytes/sec
 DUMP: level 0 dump on Mon Nov 17 10:16:52 2008
 DUMP: DUMP IS DONE
226 Transfer complete.
80281692 bytes sent in 00:54 (1.40 MB/s)
energoproekt#
```

Листинг 07.

Проверка размера закачанного файла на фтп сервере.

```
www# ls -l root.dump.gz
-rw-r--r-- 1 epbackup wheel 80281692 Nov 17 10:17 root.dump.gz
www#
```

Видим, что без архивации дамп занял в два раза больше. Причем это самый маленький раздел у меня. Дампил раздел размером 990 метров, а занято на нем было 185 метров. Вот и делаем вывод. Думаю полезно дампить параллельно с архивацией.

Для восстановления (скорее всего по этому поводу тоже будет статья, но здесь сделаю пометку) пользуемся :

```
gzip -d root.dump.gz | ( cd / ; restore -rf - )
```

Тестирование возможности восстановления из резервных копий.

Во первых, нужно перейти в раздел, дамп которого мы сейчас будем рассматривать. Например, если home.dump, то делаем cd /home. Для восстановления файлов из дампа в диалоговом режиме используем команду:

Листинг 08.

Интерактивное восстановление через restore.

```
www# restore -i -f root.dump
restore > ls
.:
.cshrc cdrom/ home/ rescue/ ttt.sh
.profile compat@ httpd.core root/ usr/
.snap/ dev/ lib/ sbin/ var/
COPYRIGHT dist/ libexec/ squid/
X11R6.tgz dump/ media/ sys@
bin/ entropy mnt/ testparm.core
boot/ etc/ proc/ tmp/
```

```
restore > cd /etc
```

```
restore > ls natd.conf
natd.conf
restore > add natd.conf
restore > ls natd.conf
*natd.conf
restore > extract
You have not read any tapes yet.
If you are extracting just a few files, start with the last volume
and work towards the first; restore can quickly skip tapes that
have no further files to extract. Otherwise, begin with volume 1.
Specify next volume #: 1
set owner/mode for '.'? [yn] y
restore > quit
www#
```

Для ползания по каталогам пользуемся командой cd

Для просмотра файлов командой ls

Для добавления файла в список для извлечения пользуемся командой add с аргументом в виде названия файла. (для удаления файла из списка соответственно команду ”delete”).

Для извлечения пользуемся командой extract (с последующими ответами «1» на вопрос какой том монтировать и «y» на вопрос восстанавливать ли владельцев файла и права доступа.
Примечание: после использования команды add список файлов полезно будет проверить ещё раз с помощью ls – те файлы, которые будут извлекаться, отмечены звездочкой.

Теперь посмотрим как можно использовать команду resore в более менее автоматическом режиме:

restore -vxNf root.dump
где -v – заставляет выводить имя каждого файла (обычно restore работает тихо)
-x – extract – вытащить из архива все файлы
-N – производит извлечение, но не записывает изменения на

диск.
-f root.dump – использует в качестве входного файла root.dump.

Далее смотрим листинг 09 после выполнения команды:

Листинг 09.

Тестовое восстановление через restore с ключем N.

```
iv# restore -vxNf root.dump
Verify tape and initialize maps
Tape block size is 32
Dump date: Mon Nov 17 09:53:14 2008
Dumped from: the epoch
Level 0 dump of / on iv.local:/dev/ad4s1a
Label: none
Extract directories from tape
Initialize symbol table.
Make node ./.snap
Make node ./dev
Make node ./home
Make node ./samba
[i]< … поскипал … >[/i]
Make node ./tmp/ssh-0XhsufqEJC
Make node ./tmp/sarg
Extract requested files
You have not read any tapes yet.
If you are extracting just a few files, start with the last volume
and work towards the first; restore can quickly skip tapes that
have no further files to extract. Otherwise, begin with volume 1.
Specify next volume #: 1
Create symbolic link ./sys->usr/src/sys
extract file ./boot/mbr
extract file ./boot/boot0
extract file ./COPYRIGHT
```

```
extract file ./boot/boot0sio
extract file ./boot/boot
< … поскипал … >
Create hard link ./bin/link->./bin/ln
Create hard link ./bin/unlink->./bin/rm
Create hard link ./bin/[->./bin/test
Create hard link ./bin/tcsh->./bin/csh
Create hard link ./bin/red->./bin/ed
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] y
iv#
```

Перенос системы на другой винчестер.

Тут будет только теория, сам я проделывал это на виртуалках, на серверах с рэйдами пока не представляю как это будет :).

Сначала посмотрим наши разделы в /etc/fstab с помощью cat /etc/fstab. Потом, например, запишем соответствия разделов и точек монтирования:

b – /root
e – /home
g – /tmp
d – /usr
f – /var
(напоминаю что «c» характеризует весь слайс).

Итак, имеем диски ad0 и ad1(нулевой это наш старый, первый будет новым, и уже подключенным к системе), диск LiveCD (это первый диск с фрей). Загружаемся с LiveCD. Размечаем жесткий диск в sysinstall. Сохраняем. Потом выбираем пункт “fixit” – “cd/dvd”.

Для каждой точки монтирования каждый раздел должен быть сначала сконструирован (newfs), затем смонтирован. Затем для восстановления необходимо перейти в этот раздел и только после всего этого пускать dump+restore. Таким образом, после приглашения командной строки для каждого раздела (исходя из буквы девайса в fstab и точки монтирования), делаем следующее:

Листинг 10.

Типовая операция восстановления раздела.

```
# newfs /dev/ad1s1g
# mount /dev/ad1s1e /mnt/home
# cd /mnt/home
# dump 0af - /home | restore xf -
```

Внимание ! Не нужно ставить слэш в конце «/home».
Посмотреть какие разделы смонтированы можно командой «mount -p».
Смонтировать CD-ROM: mount -t cd0660 /dev/acd0 /mnt (ну на всякий случай написал smile)

Небольшой промежуточный вывод:

1.Подключаем второй винчестер
2.Размечаем диск в sysinstall
3.Создаем разделы
4.Конструируем новый раздел (newfs)
5.Монтируем новый раздел к точке монтирования на старом
6.Переходим в смонтированный раздел.
7.Дамп+Рестор
8.Пункты 4-8 повторить для каждого раздела (смотри листинг 10)

После этого можно вроде выключать комп, отключать старый диск, загружать на новом и смотреть, что получилось.

Скрипты для дампов.

Я давно хотел потестить, сколько места будет занимать файл с дампом для каждого раздела. Архиваторы брал gunzip и bunzip2. Получил следующие результаты:

Таблица 01.
Сравнительная таблица размеров архивированных дампов.

Раздел	Gunzup(Mb)	Bunzip2(Mb)
/root	80,2	75,2
/home	713,8	715,6
/usr	6,755,65	6,496,01
/var	2,486,97	2,355,72

Bunzip выигрывает практически во всех случаях, но не очень существенно, так что особенной разницы не увидим. Единственное, от
чего стоит отталкиваться — от вашей общей стратегии копирования, того единообразия, которое должно соблюдаться в любой в
здравом уме настроенной системе.

Сам скрипт с указанием полных путей (по кронтабу FreeBSD 6.2 у меня не понимала что такое dump и поначалу я получил рутом
такое письмо :

Quote

«dump: not found
‘EPSV’: command not understood.
dump: not found
‘EPSV’: command not understood.
dump: not found
‘EPSV’: command not understood.
dump: not found
‘EPSV’: command not understood.»

Листинг 11.

Типовой скрипт для дампа на фтп-сервер по кронтабу.

```
#!/bin/sh
# File : /home/jsand/scripts/backup.sys.dump.sh
# Computer: 192.168.0.1
# Use: make dump of partitition
# Created : 2008.11.11
# Author : Andrey v Strukov
# Changes :
# 2009.01.14 - Added full path to exec files.

# Create, archive and send to ftp dump of / partitition.
/sbin/dump -0aLuf - / | /usr/bin/bzip2 -9 - | /usr/bin/ftp -u
ftp://jsand:pamela@192.168.11.63/today/root.gwsrv.dump.bz2 -
```

```
# var
/sbin/dump -0aLuf - /var | /usr/bin/bzip2 -9 - | /usr/bin/ftp
-u ftp://jsand:pamela@192.168.11.63/today/var.gwsrv.dump.bz2 -

# home
/sbin/dump -0aLuf - /home | /usr/bin/bzip2 -9 - | /usr/bin/ftp
-u  ftp://jsand:pamela@192.168.11.63/today/home.gwsrv.dump.bz2
-

# usr
/sbin/dump -0aLuf - /usr | /usr/bin/bzip2 -9 - | /usr/bin/ftp
-u ftp://jsand:pamela@192.168.11.63/today/usr.gwsrv.dump.bz2 -
```

Соответственно, этот скрипт пихаем в кронтаб через «crontab -e». Строчка «1 0 * * 1-5» означает (если кто забыл или не знал) что скрипт будет запускаться в 00:01 каждый день месяца и каждый месяц в году, а «1-6» означает что дампы будут делаться с понедельника по субботу. НО, учитывайте что 00:01 в понедельник это практически воскресный дамп, а соответственно 00:01 в
субботу это пятничный..

Листинг 12.

Добавление задания скрипта в кронтаб.

```
#
1 0 * * 1-5 /home/jsand/scripts/backup.sys.dump.sh
```

Ещё, кто не знает как работать с vi – вкратце: «a» – переход в режим вставки (типа можно печатать), «esc» – выход из режима вставки и вход в командный режим, тут действуют клавиши «x» – удаление букв по одной и «d» – удаление строчек целиком. Выйти из vi можно через зажатый «shift» и двойное нажатие «Z». За более подробным руководством всем обращаться к википедии (ссылка 4). После всего проверяем листинг заданий в кроне через «crontab -l».

Ссылки на ресурсы, с которых взято большинство информации:

1. http://www.opennet.ru

2. http://www.citforum.ru/operating_systems/manpages/CRONTAB.1.shtml
3. http://rus-linux.net/MyLDP/BOOKS/redhatsec/glava22.htm
4. http://ru.wikipedia.org/wiki/Vi