Lab 11: Backups - Dump and Restore

*In this lab, you will learn about the **backup** and **restore** commands, which are used to backup and restore filesystems or directory trees. Since we do not have the necessary backup hardware to perform full backups, and because backups can take a fair amount of time, we will simulate the backup process using a small filesystem created on a floppy disk.*

Boot the System

Prepare the system for the current lab.

Step 1. Boot your system into multi-user mode.

Create a Test Filesystem

Create a test filesystem which will be used to produce backups and to be restored.

Step 2. Create a new filesystem to use as our test filesystem. Insert the floppy and create a new ext2 filesystem. Mount the filesystem onto **/mnt/floppy** when it is created

```
# mkfs -t ext2 /dev/fd0
# mount /dev/fd0
```

Step 3. We will use the contents of the **/etc/sysconfig** directory as our test filesystem. To make an exact copy of the filesystem, we'll use the **tar** command to stream an archived directory tree from one filesystem to another. This is a commonly used command pipeline and is worth memorizing. It is better than a recursive **cp** for a variety of reasons, such as proper handling of symbolic links, maintaining dates, file ownership, etc.

```
# tar -cvf - /etc/sysconfig | (cd /mnt/floppy; tar -xpvf -)
```

Step 4. For safety, so that you do not accidentally replace files in **/etc**, rename the **/mnt/floppy/etc** directory that you just created on the **/mnt/floppy** filesystem. Compare the dates and ownership of files in **myetc/sysconfig** with those in the **/etc/sysconfig**

```
# cd /mnt/floppy
# mv etc myetc
# ls -l myetc/sysconfig
```

Backup a Filesystem

Produce a backup archive of the test filesystem.

Step 5. Use the **dump** command to create a full backup of that test filesystem. We will perform the backup into a file, rather than onto a backup device such as a tape drive (pretend the backup archive file is a really a tape cartridge or CD-RW disc, that you would label appropriately). Look up in the **dump** man page each option in the command below, and then run the command. Look at the output and try to understand what it is telling you. If you are not sure, ask!

```
# dump -0u -L "yourinitials" -f /tmp/floppy.full.1 /mnt/floppy
```

Q1. What is the meaning of each of the options and arguments? _____

_____

_____

_____

_____

_____

Step 6. The **dump** command can update the file **/etc/dumpdates** with the dates that backups are performed. Look at the file **/etc/dumpdates**.

```
$ cat /etc/dumpdates
```

Q2. Which command and options causes **/etc/dumpdates** to be updated? _____

Examine the Dump Archive Contents

Run restore in command-line mode to produce a table of contents of a backup archive.

Step 7. The **restore** command can be used to restore individual files, directories, an entire filesystem, or to examine the contents of the archive. Use the **restore** command to examine the contents of the **/tmp/floppy.full.1** dump archive (volume). Pipe the output to **less** if necessary.

```
# restore -t -f  /tmp/floppy.full.1
```

Q3. What are the numbers to the left of each file name? _____

Q4. What command line will verify your theory? _____

Q5. Write down the pairs of file names and numbers: _____

_____

_____

Perform Incremental Backups

Run dump to produce an incremental backup, backing up files changed since the last higher-level backup.

Step 8. Perform a level 5 dump, creating a new dump archive. **Note**: Henceforth, a new dump archive is being created for each of the dumps – this will simulate how the backup media would be changed each evening before a backup is performed.

```
# dump -5u -L "yourinitials" -f /tmp/floppy.inc.2 /mnt/floppy
```

Q6. Which files and directories were archived on this dump? _____

Step 9. In a typical day, new files are created, and in the evening, **dump** is run to backup the filesystem. Add a new file to your **/mnt/floppy** filesystem, and run an incremental dump again, creating a new dump archive:

```
# cp /etc/passwd /mnt/floppy/myetc
# dump -5u -L "yourinitials" -f /tmp/floppy.inc.3 /mnt/floppy
```

Q7. Which files and directories were archived on this dump? _____

Q8. Explain why each was written to the archive: _____

_____

Use restore in Interactive Mode

Run restore in interactive mode to learn about its commands and to restore an individual file.

Step 10. Remove a file to test a restore. Then run **restore** interactively, which will give you a command prompt:

```
# rm /mnt/floppy/myetc/passwd
# restore -i -f  /tmp/floppy.inc.3
```

Step 11. Issue the **help** command in **restore** to see a list of the available commands.

```
restore > help
```

Step 12. Run the **what** command to learn about the dump archive.

```
restore > what
```

Step 13. Now use the **ls** command in **restore** to view the contents of the files in the backup archive. The **ls** and **cd** commands in **restore** are similar to the command line utilities you are already accustomed to. And the Linux version of the **dump** command has built-in command line completion as well.

```
restore > ls
restore > ls myetc
restore > cd myetc
```

Step 14. To restore a file, the file is added to a list of files with the **add** command. This list of files will eventually be extracted with the **extract** command. Add the **passwd** file to this list. The **restore** command will inform you that the **./myetc** directory already exists in the filesystem and that it does not to be restored. More importantly, it is a warning that you may be restoring on top of a directory tree that exists, thus overwriting files. It may be prudent to restore files into a temporary directory and then move the files to their proper location(s). The message can be ignored for this lab.

```
restore > add passwd
restore: ./myetc: File exists
```

Step 15. Run the **ls** command again. Notice the ***** in the output adjacent to the **passwd** file. This is the **restore** command's notation that **passwd** has been added to the extraction list:

```
restore > ls
```

Step 16. Once the desired files have been added to the extraction list, they can be restored. Because a large dump can span across multiple physical media (tapes, CD-R discs, etc.), **restore** needs to know the last volume used in a multi-volume backup. Since your previous dump only used a single volume, you specify 1 as the last one (which is also the first one in our case). Run **restore**'s **extract** command:

```
restore > extract
You have not read any tapes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
```

Step 17. The **restore** command will also ask if you want to set the ownership and permission modes for the current directory (dot). You do not need to change the permissions for dot. Next quit the **restore** command with the **quit** command:

```
set owner/mode for '.'? [yn] n
restore > quit
```

Step 18. Verify that the **passwd** file was restored in **myetc** correctly - use **diff** to compare it to the **/etc/passwd** file:

```
$ diff myetc/passwd /etc/passwd
```

Q9. What is the modification date of the restored file? _____

Q10. Why does the file have this modification date? _____

Restoring a Filesystem

Restore an entire filesystem as would be necessary to restore a filesystem after a disk crash, irreparable filesystem corruption, or to recreate a filesystem with an increased number of inodes.

Step 19. Now we are going to perform a full restore of the **/mnt/floppy** test filesystem. Recursively remove all files under the **/mnt/floppy** filesystem. Be careful!

```
# rm -rf /mnt/floppy/*
```

Step 20. A full restore must be performed only on a new, pristine filesystem. Un-mount the **/mnt/floppy** filesystem, create a new one, and then mount it as before.

```
# umount /mnt/floppy
# mkfs -t ext2 /dev/fd0
# mount /mnt/floppy
```

Step 21. A full restore needs to be initiated with a complete level 0 backup, and then subsequent incremental restores can be applied if necessary. To restore a given filesystem, change directories into the new empty filesystem.

```
# cd /mnt/floppy
```

Step 22. Before you perform the restore command below, be prepared to listen to / watch the floppy drive very carefully. Notice how long it takes until you hear/see activity on this drive! This is an example of caching of filesystem data. The filesystem data that you restore is cached in RAM, and has not yet been written to the disk. If you were to remove the floppy during this time, bad things will happen! Run **restore** with the **-r** option to perform a full restore, observing the caching affects:

```
# restore -r -f  /tmp/floppy.full.1
```

Step 23. Examine the contents of the **/mnt/floppy** filesystem. You should notice that the restore did not produce the desired results.

Q11. Why is the **myetc/passwd** file missing? _____

Step 24. After a full restore, you need to restore the files backed-up in incremental backups. This ensures that everything is up to date with the latest backup. The Linux **dump** command creates a file called **restoresymtable** which is used to assist in this process (it helps remove *ghost* files - those files that were deleted between the time of the full dump and the incremental dumps). Only the most recent backup for any given level needs to be restored. Continue the restore process by restoring the incremental dumps:

```
# restore -r -f  /tmp/floppy.inc.3
```

Q12. Why is it unnecessary to restore files from the **/tmp/floppy.inc.2** archive? (Hint: although this archive contains no files, this is irrelevant. Even if it did contain files, the archive would not be needed). _____

_____

Step 25. Once again, look at the contents of the **/mnt/floppy** filesystem.

Q13. Has the entire filesystem been restored correctly? _____

Q14. Is the **myetc/passwd** file there this time? _____

Step 26. Once the restoration is completed, and there are no more dump archives to process, the **restoresymtable** file is not needed, and can be removed:

```
# rm restoresymtable
```

Step 27. Copy the file **/etc/hosts** to your test filesystem and run a level 9 dump. A higher level should copy only files changed or created since the last dump at this level or less.

```
# cp /etc/hosts /mnt/floppy/myetc/hosts
# dump -9u -L "yourinitials" -f /tmp/floppy.inc.4 /mnt/floppy
```

Q15. The results were probably not what you expected - which files were included in the dump archive? _____

_____

Step 28. This level 9 dump actually contains **all** the files in the filesystem. This occurred because the previous full restore on a new filesystem created the restored files with the inode numbers that were assigned by the operating system. Since the **restore** program has no control over inode number assignment (the kernel does), the inode number for a given file in the filesystem may not match the inode number for that same file in the dump archive. Thus, **dump** must archive each and every file to ensure that inode numbers in the archive match those in the filesystem. Compare the inode numbers of the new archive with the inode numbers you noted in Q5. Also, see the end of the **restore** man page in the **Bugs** section for more information. It is important to start learning these commands in depth, including their **limitations** and **bugs**. Anyone can enter a command line - your challenge is to understand how the commands work.

Restore Individual Files via Command Line

The restore command can be run in non-interactive mode to restore individual files.

Step 29. Files can be extracted non-interactively by supplying the **-x** option to **restore**. Remove the **myetc/hosts** and **myetc/passwd** files, and then extract only the **hosts** file. Confirm that only the **hosts** file was restored.

```
# rm myetc/hosts myetc/passwd
# restore -x -f  /tmp/floppy.inc.4  myetc/hosts
```

Compare Backup Contents against a Filesystem

The restore command can be used to compare the contents of a backup archive with the contents of a filesystem.

Step 30. The **-C** option to **restore** compares the contents of a backup against the filesystem used to create the backup. This is useful to learn what has changed within the filesystem since the last backup. Perform a full backup to a new archive.

```
# dump -0u -L "yourinitials" -f /tmp/floppy.full.5 /mnt/floppy
```

Step 31. Create several types of filesystem modifications by removing one file, changing the modification date of a second, modifying the contents of a third, and finally creating a new file:

```
# rm myetc/hosts
# touch myetc/sysconfig/init
# echo >> myetc/sysconfig/network
# cp /etc/group myetc/group
```

Step 32. Compare the backup against the filesystem:

```
# restore -C -f /tmp/floppy.full.5
```

Q16. What types of modifications does restore **-C** tell you about? _____

Q17. What types does it not? _____

Step 33. Examine the contents of **/etc/dumpdates.**

Q18. How many entries does the file contain? _____

Q19. Why is there only one entry for level 0, even though two level 0 backups were performed?_____