Unit 6.3 Backup and Recovery

Need for Backup

- Data is critical to existence of any organization.
- There are some situations in which security and maintenance of data becomes more critical. These situations create the need for Backup of data. e.g.:
    - Accidental erasing data from disk
    - Corruption of data due to power failure or disk crash
    - Virus attack
    - Upgrade of system

Backup Strategies

- Backup strategy defines the kind of backup used or selected
- A full backup means a complete backup of all required files
- The incremental backup means the backup of only those files which have been changed since previous backup.
- All organizations must have a definite backup policy.

Backup and Restore

- Backup and restore essentially consists of the copying of large numbers of files from one place to another.
- UNIX in general provides three tools for the large scale storage of files: tar, cpio and dump/restore. At least two of these systems should be available on any system.

cpio -- copy files to and from archives

- **cpio** stands for "**copy in, copy out**".
- Used for processing the archive files like **.cpio* or **.tar*.
- This command can copy files to and from archives.
- Three modes:
  - **-i** Input
  - **-o** Output
  - **-p** Pass-through

- **-o :Copy-out Mode:**
- Copy files named in name-list to the archive
- **Syntax:**
- cpio -o < name-list > archive

- **Create *.cpio Archive File**
- You can create a *.cpio archive that contains files and directories using cpio -ov

```
$ ls
    file1.o file2.o file3.o
$ ls | cpio -ov > /tmp/object.cpio
```

- The ls command passes the three object filenames to cpio command and cpio generates the object.cpio archive.

- **-i : Copy-in Mode:**

  Extract files from the archive

- **Syntax:**
- cpio -i < archive
- **Extract *.cpio Archive File**

To extract a given *.cpio file, use cpio -iv as follows:

- $ mkdir output
- $ cd output

- $ cpio -idv < /tmp/object.cpio

Here, d option Create directories as necessary.

- **-p :Copy-pass Mode:** Copy files named in name-list to destination-directory
- In copy-pass mode, cpio copies files from one directory tree to another, combining the copy-out and copy-in steps without actually using an archive. It reads the list of files to copy from the standard input; the directory into which it will copy them is given as a non-option argument.
- **Syntax**
- cpio -p destination-directory < name-list

- Unless specifically stated otherwise, options are applicable in all operating modes.
- **-A** (o mode) Append to the specified archive.
- **-a** (o and p modes) Reset access times on files after they are read.

- **-B** (o mode only) Block output to records of 5120 bytes.
- **-C** *size* (o mode only) Block output to records of *size* bytes.

- **-c** (o mode only) Use the old POSIX portable character format.
- **-d** (i and p modes) Create directories as necessary.
- **-E** *file* (i mode only) Read list of file name patterns from *file* to list and extract.
- **-F** *file* Read archive from or write archive to *file*.
- **-f** *pattern* (i mode only) Ignore files that match *pattern*.

tar command

- tar (tape archive) is an archiving utility.
- To create an archive from files or a directory, use: (with option c = create)

  $tar -cvf archive.tar file1 file2 file3
- To extract from an archive use:

  (with options x = extract, v = verbose, f = file):

  $tar - xvf archive.tar

- To create compressed archives, use:

(with option z to compress with gzip)

$tar -cvzf archive.tar file1 file2 file3

- To show all files held in an archive use

  (with option t = list):

```
$tar -tvf archive.tar
```

Monitoring Disk Space

- The one system resource that is most commonly over-committed is disk space. There are many reasons for this, ranging from applications not cleaning up after themselves, to software upgrades becoming larger and larger, to users that refuse to delete old email messages.
- System administrators must monitor disk space usage on an ongoing basis, or face possible system outages and unhappy users.

Using df

- The easiest way to see how much free disk space is available on a system is to use the df command.
- E.g.:

File system	1k-blocks	Used	Available	Use%	Mounted on
/dev/sda3	8428196	4282228	3717836	54%	/
/dev/sda1	124427	18815	99188	16%	/boot
/dev/sda4	8428196	3801644	4198420	48%	/home
none	644600	0	644600	0%	/dev/shm

- As we can see, df lists every mounted system, and provides information such as device size (under the 1k-blocks column), as well as the space used and still available. However, the easiest thing to do is to simply scan the Use% column for any numbers nearing 100%.

dd - block copy and convert

- The ***dd*** command allows you to copy from raw devices, such as disks and tapes, specifying the input and output block sizes. ***dd*** was originally known as the disk-to-disk copy program. With ***dd*** you can also convert between different formats, for example, EBCDIC to ASCII, or swap byte order, etc.
- **Syntax**

  dd[***if***=input_device] [***of***=output_device] [Operand=value]

- **Common Options**

- **if**=input_device the input file or device
- **of**=output_device the output file or device
- If the input or output devices are not specified they default to standard input and standard output, respectively.
- Operands can include:
- **ibs=n** input block size (defaults to 512 byte blocks)
- **obs=n** output block size (defaults to 512 byte blocks)
- **bs=n** sets both input and output block sizes
- **files=n** copy **n** input files
- **skip=n** skip **n** input blocks before starting to copy
- **count=n** only copy **n** input blocks
- **conv=value**[,value] where **value** can include:
- **ascii**convert EBCDIC to ASCII
- **ebcdic**convert from ASCII to EBCDIC etc.

du - report disk space in use

- ***du*** reports the amount of disk space in use for the files or **directories** you specify.
- **Syntax**
- ***du*** [options] [directory or file]
- **Common Options**
- **-a** display disk usage for each file, not just subdirectories
- **-s** display a summary total only
- **-k** report in kilobytes (SVR4)

Examples

- #du /home/sales/

  4244 /home/sales/fans

  6777 /home/sales/tvs

  3229 /home/sales/ovens

It also reports the summary at the end.