3. Edit the New_Policies.txt file for casing errors and generate the word count.	a. To view the contents of the file, enter **cat New_Policies.txt** b. Observe that casing of the letter "Y" in the word "EmploYees" is incorrect. c. To change the casing of the letter "Y" in the word "EmploYees" in the output, enter **tr 'Y' 'y' < New_Policies.txt** d. Observe that the casing of the letter "Y" has changed in the output. e. To count the words in the file, enter **wc -w New_Policies.txt** f. Observe that the word count of the file is displayed.
4. Create a hard link for the file.	a. To create a hard link for the file, enter **ln New_Policies.txt New_Policies_09.txt** b. To view whether the hard link has been created, enter **ls New*** c. Observe that the hard link is created. d. Clear the terminal screen.

ACTIVITY 4-8

Creating Links to Frequently Used Files

Data Files:

- Attendancelogfile.txt
- Currentdatetime

Before You Begin:

1. You have logged in as root in the CLI and the first terminal is displayed.
2. To copy the data file, enter `cp /085099Data/Managing_Files/Attendancelogfile.txt /opt`.
3. To copy the data file, enter `cp /085099Data/Managing_Files/Currentdatetime /opt`.
4. To modify the file to make it an executable file, enter `chmod +x /opt/Currentdatetime`.
5. To clear the terminal screen, enter `clear`.

Scenario:

You would like to add hard links to files that you use frequently. Both files are located in the /opt directory. One of the files is a text file and the other is an executable program. You have to make these files easily accessible from the root directory without disturbing their original location.

What You Do	How You Do It
1. **True or False? When you delete a hard link, the file to which the hard link is set will also get deleted.** __ True __ False	
2. Create a hard link named "attlog" in the root directory for the Attendancelogfile.txt file located in the /opt directory.	a. To create a hard link, enter **`ln /opt/Attendancelogfile.txt⇒ attlog`** b. To view the hard link, enter **`ls -l attlog`**
3. **True or False? If an executable file is located in the search path, then the user can run the file from any location.** __ True __ False	

4. Create a symbolic link named "Time" for the executable program, Currentdatetime, located in the /opt directory.

   a. To create a soft link, enter

   ```
   ln -s /opt/Currentdatetime⇒
   /bin/Time
   ```

   b. To view the soft link, enter **`ls -l /bin/Time`**

   c. To execute the file, enter **`Time`**

   d. To clear the terminal screen, enter **`clear`**

Topic F

Back Up and Restore Files

In the previous topics, you created, edited, located, and linked files. It is essential that you also know how to back up and restore these files when the need arises. In this topic, you will back up and restore files.

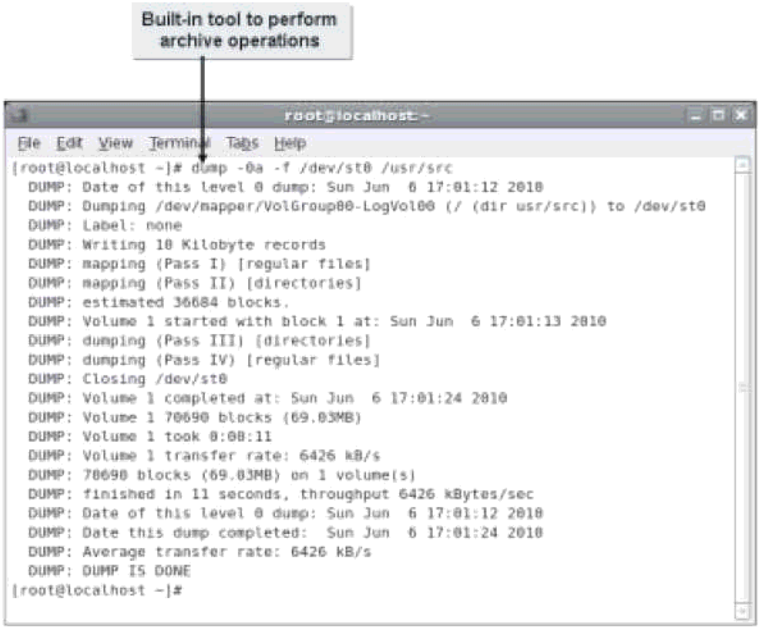
Learning how to back up and restore files will save you countless hours of repairing your system after a system failure. Backing up and restoring files allow you to keep an additional copy of files on your system because they existed at a specific point in time. If you ever have a system failure, these files can be used to restore your system.

Archiving

Definition:

Archiving is a method of storing data by copying data from a system disk drive into a backup device. This is done to preserve a record of the data for future reference or to create data dumps. In the event of a network disruption resulting in data loss, the data can be retrieved from archives.

Example:



Built-in tool to perform archive operations

```
[root@localhost ~]# dump -0a -f /dev/st0 /usr/src
  DUMP: Date of this level 0 dump: Sun Jun  6 17:01:12 2010
  DUMP: Dumping /dev/mapper/VolGroup00-LogVol00 (/ (dir usr/src)) to /dev/st0
  DUMP: Label: none
  DUMP: Writing 10 Kilobyte records
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 36684 blocks.
  DUMP: Volume 1 started with block 1 at: Sun Jun  6 17:01:13 2010
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /dev/st0
  DUMP: Volume 1 completed at: Sun Jun  6 17:01:24 2010
  DUMP: Volume 1 70690 blocks (69.03MB)
  DUMP: Volume 1 took 0:00:11
  DUMP: Volume 1 transfer rate: 6426 kB/s
  DUMP: 70690 blocks (69.03MB) on 1 volume(s)
  DUMP: finished in 11 seconds, throughput 6426 kBytes/sec
  DUMP: Date of this level 0 dump: Sun Jun  6 17:01:12 2010
  DUMP: Date this dump completed:  Sun Jun  6 17:01:24 2010
  DUMP: Average transfer rate: 6426 kB/s
  DUMP: DUMP IS DONE
[root@localhost ~]#
```

Figure 4-20: *Archiving files using the dump command.*

The cpio Command

The *cpio* command copies files to and from archives. It is included in standard Linux distributions. The `cpio` command has three operating modes.

Operating Mode	Description
Copy-out `cpio -o`	In this mode, the command copies files into an archive. It reads the standard input to obtain a list of file names and then copies those files to the standard output.
Copy-in `cpio -i`	In this mode, the command copies files from an archive. It extracts files from the standard input.
Copy-pass `cpio -p`	In this mode, the command copies files from one directory tree to another. It reads the standard input to obtain the list of file names that are created and copied into the destination directory.

The dd Command

The *dd* command copies and converts files to enable them to be transferred from one type of media to another. The `dd` command has various options.

A selected input file is copied to a selected output file. If no files are selected, the standard input and the standard output are used.