

Overview of Computer Workshop

Unit-3, Lecture – 2

Some more commands related to Files

- **wc**: It is used to display the total number of lines and characters in one or more files.
- By default it displays four-columnar output. First column shows number of lines present in a file specified, second column shows number of words present in the file, third column shows number of characters present in file and fourth column itself is the file name which are given as argument.

Syntax: `wc filename`

Options:

`wc -w filename`

`wc -l filename`

`wc -c filename`

- **sort**: It sorts its output into alphabetical order line by line.

Syntax: `sort filename`

Options:

`sort -r` Reverse normal order

`sort -n` Sort in numeric order

`sort -nr` Sort in reverse numeric order

- **head** : It prints the first 10 lines of a file.

Syntax: `head filename`

Options:

`head -n filename` (print first n lines of the file)

- **tail**: It prints the last 10 lines of a file.

Syntax: `tail filename`

Options:

`tail -n filename` (print last n lines of the file)

`tail +n filename` (starts printing from the (n+1)th line)

- **cmp**: It prints the first place where two files differ. cmp is used when one wants to be sure that two files really have the same contents. It is fast and it works on any kind of file.

Syntax: `cmp filename1 filename2`

- **diff**: It reports on all lines that are changed, added and deleted. It is used when the files are expected to be somewhat different and one wants to know exactly which lines differ. diff works only on files of text.

Syntax: `diff filename1 filename2`

- **grep**: It prints the line from the file matching with specified pattern.

Syntax: `grep pattern filename`

Option:

`grep -v pattern filename` (prints lines from the file not matching with the pattern)

Pipe

- A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux and other Unix-like operating systems to send the output of one command/program/process to another command/program/process for further processing.
- It is denoted by a vertical bar (|).
- E.g. The next command line displays the number of files in a directory. The `wc` (word count) command with the `-w` (words) option displays the number of words in its standard input or in a file you specify on the command line:

```
$ ls | wc -w
```

Assignment -1 (Additional Questions)

14. Check the output of the following command:

```
cmp reginfo personalinfo
```

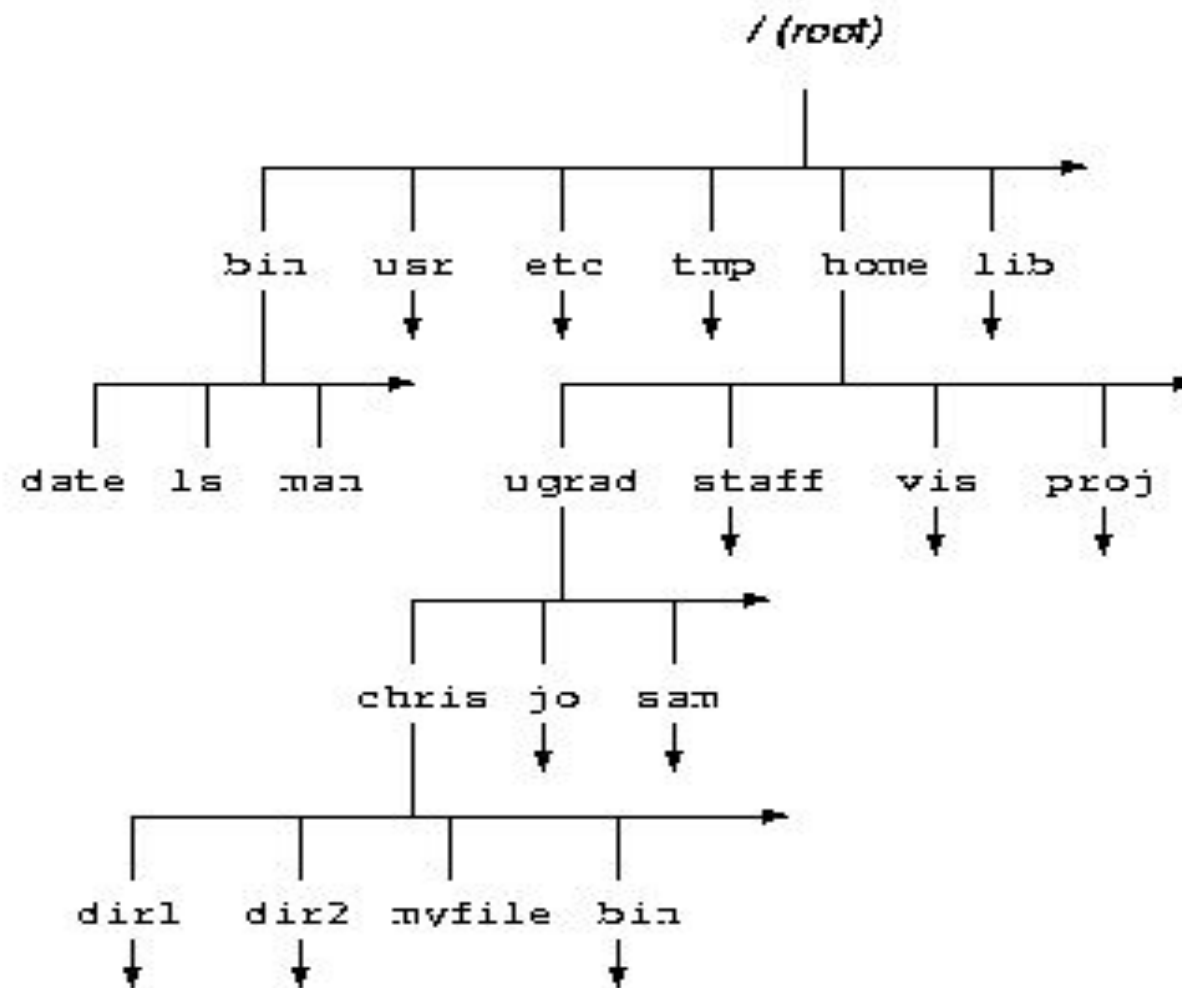
```
diff reginfo personalinfo
```

15. Write a command to count the number of files in the current working directory and display that number.

Linux Filesystem

- **Filesystem:** A filesystem provides the way in which files and directories are placed logically for storage and retrieval on a storage disk.
- The general-purpose computer system needs to store data systematically so that we can easily access the files in less time.
- **Linux Filesystem:** Linux has hierarchical filesystem much like an upside-down tree, with root (/) at the base of the file system and all other directories spreading from there.
- Each directory has a name and can hold other files and directories. Directories, in turn, are arranged under other directories and so forth in a tree like organization.
- All other directories in Linux can be accessed from the root directory.

Linux Filesystem structure



Linux File System Features

- **Specifying paths:** Linux does not use the backslash (\) to separate the components; it uses forward slash (/) as an alternative. For example, as in Windows, the data may be stored in C:\ My Documents\ Work, whereas, in Linux, it would be stored in /home/ My Document/ Work.
- **Partition, Directories, and Drives:** Linux does not use drive letters to organize the drive as Windows does. In Linux, we cannot tell whether we are addressing a partition, a network device, or an "ordinary" directory and a Drive.
- **Case Sensitivity:** Linux file system is case sensitive. It distinguishes between lowercase and uppercase file names.
- **File Extensions:** In Linux, a file may have the extension '.txt,' but it is not necessary that a file should have a file extension.
- **Hidden files:** Linux distinguishes between standard files and hidden files, mostly the configuration files are hidden in Linux OS. Usually, we don't need to access or read the hidden files. The hidden files in Linux are represented by a dot (.) before the file name (e.g., .ignore).

Pathnames

- A path is a unique location to a file or a folder in a file system of an OS. A path to a file is a combination of / and alpha-numeric characters.
- **Absolute Path-name:** *An **absolute path** is defined as specifying the location of a file or directory from the root directory(/). In other words, we can say that an absolute path is a complete path from start of actual file system from / directory.*
- **Relative Path-name:** ***Relative path** is defined as the path related to the present working directory(pwd). It starts at your current directory.*

Links

- In your Linux file system, a **link** is a connection between a file name and the actual data on the disk. More than one filename can **link** to the same data.
- There are two types of links in Linux OS:
- **Hard Link:**
 - They are the low-level links. It links more than one filename with the same Inode and it represents the physical location of a file.
 - When hard link is created for a file, it directly points to the Inode of the original file in the disk space, which means no new Inode is created.
 - Directories are not created using hard links and they can not cross filesystem boundaries.
 - When the source file is removed or moved, then hard links are not affected.
- **Command:**

ln original_filename link_name

■ Soft link or Symbolic Link:

- Soft links are very common. It represents a virtual or abstract location of the file.
- It is just like the shortcuts created in Windows.
- A soft link doesn't contain any information or content of the linked file, instead it has a pointer to the location of the linked file.
- In other words, a new file is created with new Inode, having a pointer to the Inode location of the original file.
- It is used to create link between directories and can cross filesystem boundaries.
- When the source file is removed or moved, then soft links are not updated.
- *Command:*

ln -s original_filename link_name

File Descriptor and Inode

- Linux makes a clear distinction between a file and the information about a file.
- Each file consists of a sequence of bytes. The file does not include any control information, such as its length or an end-of-file (EOF) delimiter.
- All information needed by the file system to handle a file is included in a data structure called an **inode**.
- Each file has its own inode, which the file system uses to identify the file.

Information in Inode

- File type
- Number of hard links associated with the file
- File length in bytes
- Device ID (i.e., an identifier of the device containing the file)
- Inode number that identifies the file within the filesystem
- UID of the file owner
- User group ID of the file
- Several timestamps that specify the inode status change time, the last access time, and the last modify time
- Access rights and file mode