

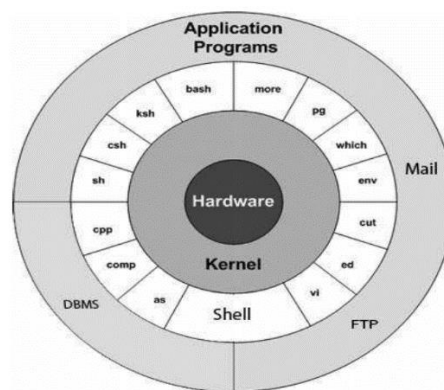
LAB SESSION #1

(Basic of Unix and File and Directory Commands in Unix)

Operating system is a specialized program that controls and monitors the execution of all other programs that reside in the computer, including application programs and other system software. The primary functionalities of any operating system includes:

- ***Memory management***
- ***Processor management***
- ***Device management***
- ***Security***
- ***Coordination between other software and users***

Unix is a C based operating system, originally developed in 1969 at Bell labs. It is a multiuser, multitasking operating system. A multiuser system allows more than one user to use the system simultaneously and in a multitasking system more than one tasks can be performed simultaneously. There are various Unix variants available in the market such as Solaris Unix, AIX, HP Unix and BSD. Linux is also a flavor of Unix which is freely available.



Unix architecture

A unix operating system has the following four layers:

Kernel: The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.

Shell: The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants.

Commands and Utilities: There are various commands and utilities which you can make use of in your day to day activities. cp, mv, cat and grep, etc. are few examples of commands

and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options.

Files and Directories: All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the file system.

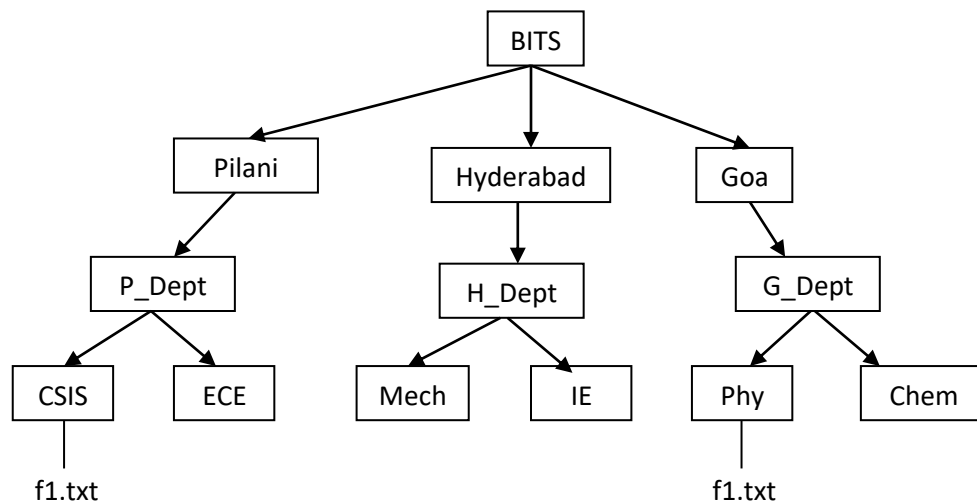
To perform any task in Unix, a user need to execute commands in the shell of the unix. A user typically interacts with a Unix shell using a 'terminal'. To execute commands in the shell, a user first needs to login using his user-id and password. As soon as a user is logged in, Unix places the user in his or her "home" directory. This directory will vary for each user.

Unix uses a hierarchical structure for organizing files and directories. This structure is often referred to as a directory tree.

Root: The node under which all directories and files are stored. A Unix tree has a single root node, the slash character (/).

Home: The directory in which you find yourself when you first login is called your home directory.

Path: To access a resource in Unix you need to specify the path of that resource in the command. There are two different paths in Unix: Relative and Absolute. While relative path refers to the path of a resource from a non-root directory, absolute path refers to the path specified relative to the root directory.



Consider the above directory structure. At some instance, if you are working in the *Goa* directory and you want to display (using *cat* command, which will be introduced a little later) the content of a file *f1.txt* stored in the *Phy* directory. You can display the file content by providing either the relative or the absolute path of *f1* to the *cat* command:

Access using absolute path:

`cat /Goa/G_Dept/Phy/f1.txt` ,here, the first '/' denotes the root directory, which is BITS in this case.

Access using relative path:

`cat G_Dept/Phy/f1.txt` ,here, you have specified the path of f1.txt relative to Goa directory.

Now let us assume that you are in *Goa* directory and you want to display (using *cat* command) the content of a file f1.txt stored in the CSIS directory of Pilani campus.

In this case, you can access the file f1.txt only by using its absolute path. This is because, to reach CSIS, you have to pass through root node. So, the command would be

`cat /Pilani/P_Dept/CSIS/f1.txt`

Given below are some Unix commands that we will be using in this course for file and directory management. Try them out yourself on your terminal!

1. **man**: Displays the user manual of any command that we can run on the terminal.
2. **pwd**: Displays the path of the present working directory
3. **ls**: Lists the content of a directory. Can be executed with various options like `-a`, `-l`, etc.
4. **cd** → changes the current directory of the shell
 - `cd /` // Takes you to the root directory
 - `cd ..` //One step up in the directory tree
 - `cd -` //Takes you to the previous directory you were working in
 - `cd ~` //Takes you to *your home* directory
3. **mkdir** → Making a new directory
 - `mkdir dir` //creates a directory named *dir*
 - `mkdir -p` // create nested directories.
 - e.g. `mkdir -p dir/hello/world` //creates a world directory inside a hello directory which itself is inside *dir* directory
4. **rmdir** → Removing a directory
 - `rmdir dir` //removes directory *dir*, provided it is empty
 - `rm -r dir` //deletes a directory recursively along with its content.
5. **touch** → Creating a file
 - e.g. `touch file1` //creates a file named *file1*
6. **cat** → Displays the content of a file
 - `cat file1` // Displays the content of the file1
 - `cat > file1` // Opens file1 for writing into it, if file1 already exists. Otherwise, creates file1 and opens it for writing into it.
 - `cat file1 file2` //Displays the content of file1 followed by the content of file2
7. **cp** → Copying the content of a file
 - `cp file1 file2` // copies the content of file1 in file2. Overwrites file2's content
 - `cp file1 file2 filen dest` // copies all files into the dest directory
 - `cp file1 ~` // copies file1 into your home directory
 - `cp *.txt dest` // copies all .txt file into dest directory
8. **mv** → Used to move one or more files or directories from one place to another and also for renaming a file or a directory
 - `mv f1.txt f2.txt` // rename f1.txt to f2.txt
 - `mv -b f1.txt f2.txt` // Creates a backup of the file to be overwritten. So, a backup file `~f2.txt` will be created
 - `mv f1.txt dir` //move f1.txt to the directory *dir*
9. **wc** → Counting words or lines
 - `wc -l file1.txt` //counts the number of lines in file1.txt
 - `wc -w file1.txt` //counts the number of word in file1.txt
 - `wc -c file1.txt` //counts the number of bytes in file1.txt

- `wc -m file1.txt` //counts the number of characters in file1.txt
- 10. **head** → print the top N number of data of the given input.
 - `head file1.txt` //Prints the first 10 lines of file1.txt
 - `head -n 5 file1.txt` or `head -5 file1.txt` //Prints the first 5 lines of file1.txt
 - `head -c 2 file1.txt` //Prints the first 2 bytes of file1.txt
- 11. **tail** → print the last N number of data of the given input.
 - `tail file1.txt` //Prints the last 10 lines of file1.txt
 - `tail -n 5 file1.txt` //Prints the last 5 lines of file1.txt
 - `tail -c 2 file1.txt` //Prints the last 2 bytes of file1.txt
- 12. **echo** → used to display line of text/string that are passed as an argument
 - `echo "Hello world"` //will display hello world on the terminal

Few other utilities / commands include:

- **date:** Displays the current date and time, day, month name, day of the month, the time zone name and the year.
- **whoami:** Displays the user-name of the current user
- **who:** Displays information about all users currently logged in.
- **clear:** Clears the screen
- **cal:** Displays the calendar of a specific month or a whole year.
- **hostnamectl:** Displays OS name and version
- **uname -r:** Displays kernel version

Exercise:

Create the directory structure given in Fig. 1. Then first create a new file f2.txt in ECE directory. Store the names of 10 students (any random name you can use) and then copy it's content into f4.txt of Mech directory. To be noted that, you do not have any f4.txt in *Mech* directory and you should not create one using cat command. After copying the content of f2.txt in f4.txt, print the names of the students starting from the 4th line in the file f4.txt and remove the ECE directory from the tree.