

UNIX operating system

UNIX is a powerful Operating System initially developed by Ken Thompson, Dennis Ritchie at AT&T Bell laboratories in 1970. It is prevalent among scientific, engineering, and academic institutions due to its most appreciative features like multitasking, flexibility, and many more. In UNIX, the file system is a hierarchical structure of files and directories where users can store and retrieve information using the files.

Features of UNIX Operating System:

Let's discuss the features of UNIX OS one by one in detail.

Multitasking: A UNIX operating system is a multitasking operating system that allows you to initiate more than one task from the same terminal so that one task is performed as a foreground and the other task as a background process.

Multi-user: UNIX operating system supports more than one user to access computer resources like main memory, hard disk, tape drives, etc. Multiple users can log on to the system from different terminals and run different jobs that share the resources of a command terminal. It deals with the principle of time-sharing. Time-sharing is done by a scheduler that divides the CPU time into several segments also called a time slice, and each segment is assigned to each user on a scheduled basis. This time slice is tiny. When this time is expired, it passes control to the following user on the system. Each user executes their set of instructions within their time slice.

Portability: This feature makes the UNIX work on different machines and platforms with the easy transfer of code to any computer system. Since a significant portion of UNIX is written in C language, and only a tiny portion is coded in assembly language for specific hardware.

File Security and Protection: Being a multi-user system, UNIX makes special consideration for file and system security. UNIX has different levels of security using assigning username and password to individual users ensuring the authentication, at the level providing file access permission viz. read, write and execute and lastly file encryption to change the file into an unreadable format.

Command Structure: UNIX commands are easy to understand and simple to use. Example: "cp", mv etc. While working in the UNIX environment, the UNIX commands are case-sensitive and are entered in lower case.

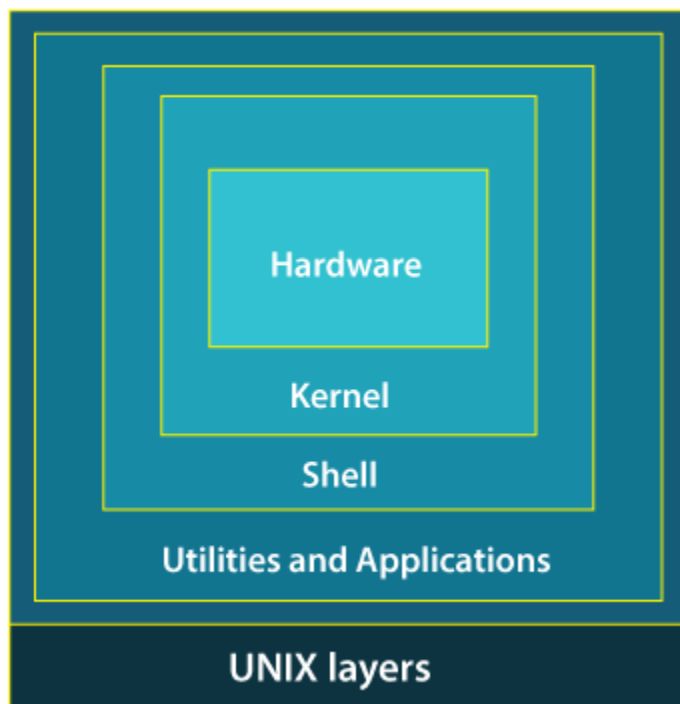
Communication: In UNIX, communication is an excellent feature that enables the user to communicate worldwide. It supports various communication facilities provided using the write command, mail command, talk command, etc.

Open Source: UNIX operating system is open source it means it is freely available to all and is a community-based development project.

Accounting: UNIX keeps an account of jobs created by the user. This feature enhances the system performance in terms of CPU monitoring and disk space checking. It allows you to keep an account of disk space used by each user, and the disk space can be limited by each other. You can assign every user a different disk quota. The root user can perform these accounting tasks using various commands such as quota, df, du, etc.

UNIX Tools and Utilities: UNIX system provides various types of tools and utilities facilities such as UNIX grep, sed and awk, etc. Some of the general-purpose tools are compilers, interpreters, network applications, etc. It also includes various server programs which provide remote and administration services.

The structure of Unix OS Layers are as follows:



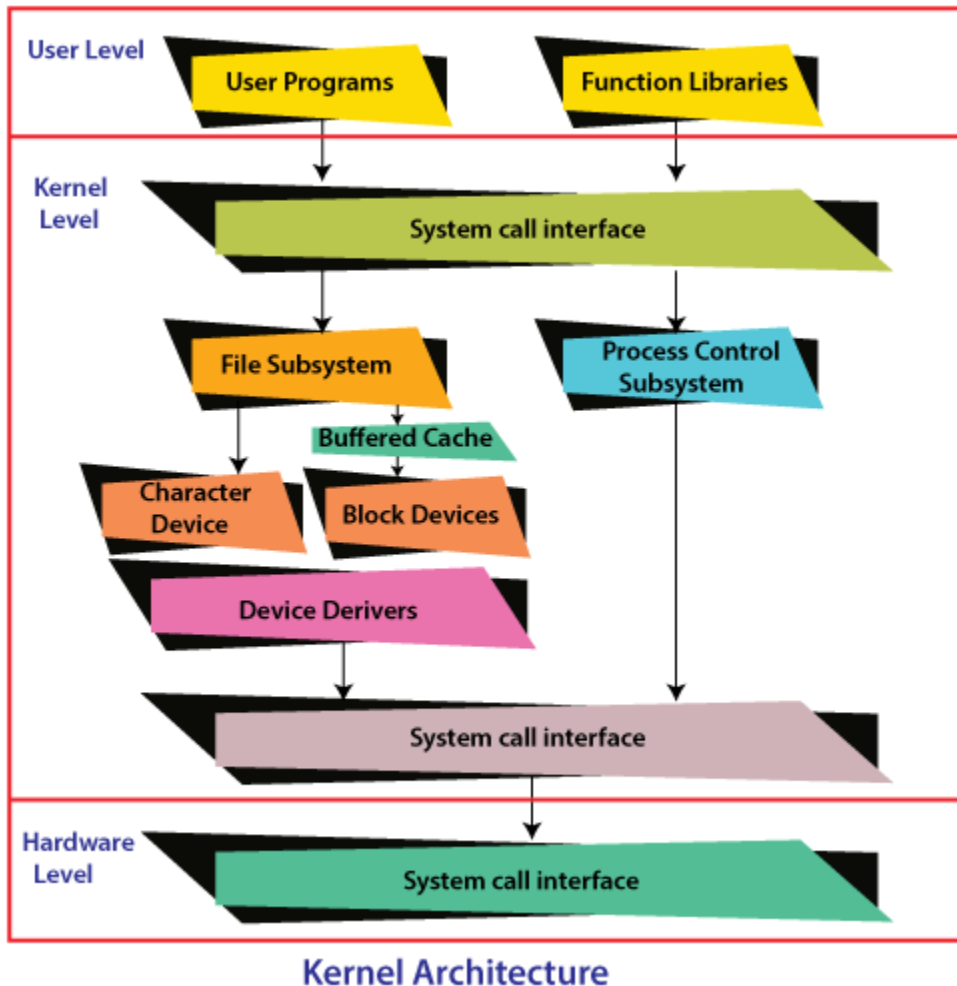
While working with UNIX OS, several layers of this system provide interaction between the pc hardware and the user. Following is the description of each and every layer structure in UNIX system:

Layer-1: Hardware -

This layer of UNIX consists of all hardware-related information in the UNIX environment.

Layer-2: Kernel -

The core of the operating system that's liable for maintaining the full functionality is named the kernel. The kernel of UNIX runs on the particular machine hardware and interacts with the hardware effectively.



It also works as a device manager and performs valuable functions for the processes which require access to the peripheral devices connected to the computer. The kernel controls these devices through device drivers.

The kernel also manages the memory. Processes are executed programs that have owner's humans or systems who initiate their execution.

The system must provide all processes with access to an adequate amount of memory, and a few processes require a lot of it. To make effective use of main memory and to allocate a sufficient amount of memory to every process. It uses essential techniques like paging, swapping, and virtual storage.

Layer-3: The Shell -

The Shell is an interpreter that interprets the command submitted by the user at the terminal, and calls the program you simply want.

It also keeps a history of the list of the commands you have typed in. If you need to repeat a command you typed it, use the cursor keys to scroll up and down the list or type history for a list of previous commands. There are various commands like cat, mv, cat, grep, id, wc, and many more.

Basic unix commands:

1. **who** : The '\$ who' command displays all the users who have logged into the system currently. As shown above on my system I am the only user currently logged in. The thing tty2 is terminal line the user is using and the next line gives the current date and time

```
$ who
Output: harssh tty2 2017-07-18 09:32 (:0)
```

2. **pwd** : The '\$pwd' command stands for 'print working directory' and as the name says, it displays the directory in which we are currently (directory is same as folder for Windows OS users).

In the output we are harssh directory(folder for Windows OS that are moving to Linux), which is present inside the home directory

```
$ pwd
Output: /home/harssh
```

3. **mkdir** : The '\$ mkdir' stands for 'make directory' and it creates a new directory. We have used '\$ cd' (which is discussed below) to get into the newly created directory and again on giving '\$ pwd' command, we are displayed with the new 'newfolder' directory.

```
$ mkdir newfolder
$ cd newfolder
$ pwd
Output: /home/harssh/newfolder
```

4. **rmdir** : The '\$ rmdir' command deletes any directory we want to delete and you can remember it by its names 'rmdir' which stands for 'remove directory'.

```
$ rmdir newfolder
```

5. [cd](#) : The '\$ cd' command stands for 'change directory' and it changes your current directory to the 'newfolder' directory. You can understand this as double-clicking a folder and then you do some stuff in that folder.

```
$ cd newfolder (assuming that there is a directory named 'newfolder' on your system)
```

6. [ls](#) : The 'ls' command simply displays the contents of a directory.

```
$ ls
Output: Desktop Documents Downloads Music Pictures Public Scratch Templates Videos
```

7. [cp](#) : This '\$ cp' command stands for 'copy' and it simply copy/paste the file wherever you want to. In the above example, we are copying a file 'file.txt' from the directory 'harssh' to a new directory 'new'.

```
$ cp /home/harssh/file.txt /home/harssh/new/
```

8. [mv](#) : The '\$ mv' command stands for 'move' and it simply move a file from a directory to another directory. In the above example a file named 'file.txt' is being moved into a new directory 'new'.

```
$ mv /home/harssh/file.txt /home/harssh/new
```

9. [rm](#) : The '\$ rm' command for remove and the '-r' simply recursively deletes file. Try '\$ rm filename.txt' at your terminal

```
$ rm file.txt
```

10. [chmod](#) : The '\$ chmod' command stands for change mode command. As there are many modes in Unix that can be used to manipulate files in the Unix environment. Basically there are 3 modes that we can use with the 'chmod' command

1. +w (stands for write and it changes file permissions to write)
2. +r (stands for read and it changes file permissions to read)
3. +x (generally it is used to make a file executable)

```
$ chmod +w file.txt
$ chmod +r file.txt
$ chmod +x file.txt
```

11. [cal](#) : The '\$ cal' means calendar and it simply display calendar on to your screen.

```
$ cal
Output : July 2017
Su Mo Tu We Th Fr Sa
```

```
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

12. [sort](#) : As the name suggests the '\$ sort' sorts the contents of the file according to the ASCII rules.

```
$ sort file
```

13. [man](#) : The '\$ man' command stands for 'manual' and it can display the in-built manual for most of the commands that we ever need. In the above example, we can read about the '\$ pwd' command.

```
$ man pwd
```

14. [lpr](#) : The '\$ lpr' command send a file to the printer for printing.

```
$ lpr new.txt
```

15. [passwd](#) : The '\$ passwd' command simply changes the password of the user. In above case 'harssh' is the user.

```
$ passwd
Output: Changing password for harssh.
(current) UNIX password:
```

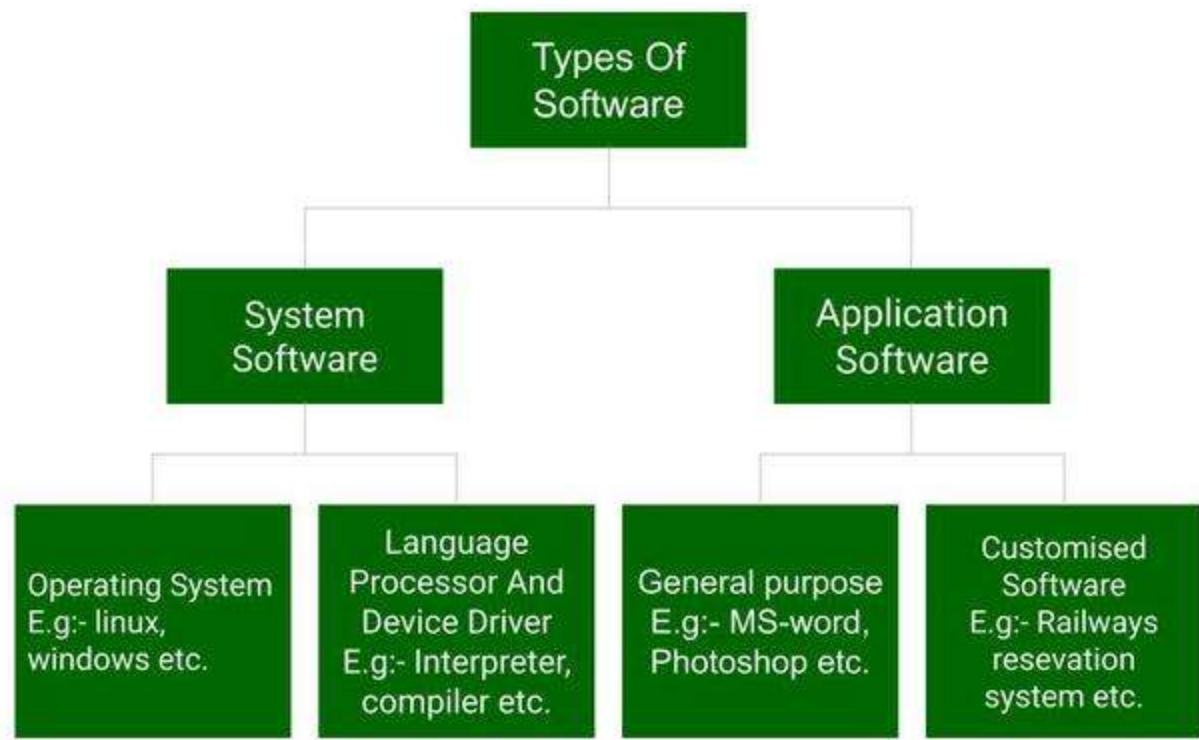
16. [clear](#) : The '\$ clear' command is used to clean up the terminal so that you can type with

more accuracy

```
$ clear
```

Software and its Types

In a computer system, the software is basically a set of instructions or commands that tells a computer what to do. Or in other words, the software is a computer program that provides a set of instructions to execute a user's commands and tell the computer what to do. For example like MS-Word, MS-Excel, PowerPoint, etc. The chart below describes the types of software:



Above is the diagram of types of software. Now we will briefly describe each type and its subtypes :

System Software

System software is software that directly operates the computer hardware and provides the basic functionality to the users as well as to the other software to operate smoothly. Or in other words, system software basically controls a computer's internal functioning and also controls hardware devices such as monitors, printers, and storage devices, etc. It is like an interface between hardware and user applications, it helps them to communicate with each other because hardware understands machine language(i.e. 1 or 0) whereas user applications are work in human-readable languages like English, Hindi, German, etc. so system software converts the human-readable language into machine language and vice versa.

Features of system software:

Let us discuss some of the features of System Software:

1. System Software is closer to the computer system.
2. System Software is written in a low-level language in general.
3. System software is difficult to design and understand.
4. System software is fast in speed(working speed).
5. System software is less interactive for the users in comparison to application software.

Types of system software:

It has two subtypes which are:

1. **Operating System:** It is the main program of a computer system. When the computer system ON it is the first software that loads into the computer's memory. Basically, it manages all the resources such as memory, CPU, printer, hard disk, etc., and provides an interface to the user, which helps the user to interact with the computer system. It also provides various services to other computer software. Examples of operating systems are Linux, Apple macOS, Microsoft Windows, etc.
2. **Language Processor:** As we know that system software converts the human-readable language into a machine language and vice versa. So, the conversion is done by the language processor. It converts programs written in high-level programming languages like Java, C, C++, Python, etc(known as source code), into sets of instructions that are easily readable by machines(known as object code or machine code).
3. **Device Driver:** A device driver is a program or software that controls a device and helps that device to perform its functions. Every device like a printer, mouse, modem, etc. needs a driver to connect with the computer system eternally. So, when you connect a new device with your computer system, first you need to install the driver of that device so that your operating system knows how to control or manage that device.

Application Software

Software that performs special functions or provides functions that are much more than the basic operation of the computer is known as application software. Or in other words, application software is designed to perform a specific task for end-users. It is a product or a program that is designed only to fulfill end-users' requirements. It includes word processors, spreadsheets, database management, inventory, payroll programs, etc.

Features of application software:

Let us discuss some of the features of Application Software:

1. An important feature of application software is it performs more specialized tasks like word processing, spreadsheets, email, etc.
2. Mostly, the size of the software is big, so it requires more storage space.
3. Application software is more interactive for the users, so it is easy to use and design.
4. The application software is easy to design and understand.
5. Application software is written in a high-level language in general.

Types of application software:

There are different types of application software and those are:

1. **General Purpose Software:** This type of application software is used for a variety of tasks and it is not limited to performing a specific task only. For example, MS-Word, MS-Excel, PowerPoint, etc.
2. **Customized Software:** This type of application software is used or designed to perform specific tasks or functions or designed for specific organizations. For example, railway reservation system, airline reservation system, invoice management system, etc.
3. **Utility Software:** This type of application software is used to support the computer infrastructure. It is designed to analyze, configure, optimize and maintains the system, and take care of its requirements as well. For example, antivirus, disk fragmenter, memory tester, disk repair, disk cleaners, registry cleaners, disk space analyzer, etc.