Experiment 2: Linux OS

1. Aim:

a. Understand the concepts of Linux operating system

2. Requirements: Linux OS

3. Related Theory:

An operating system acts as the interface between human and the machine. It allows us to communicate with our hardware by being an interface.

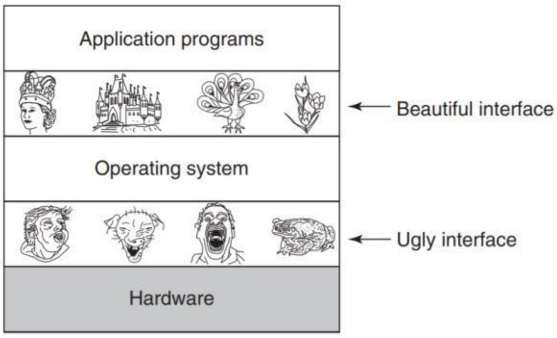

Fig 1: Benefit of OS

Operating System is also responsible for distributing the resources available in the computer (memory, printers, hard disks, etc.) to each program to maintain order in your system. Linux OS has following components:

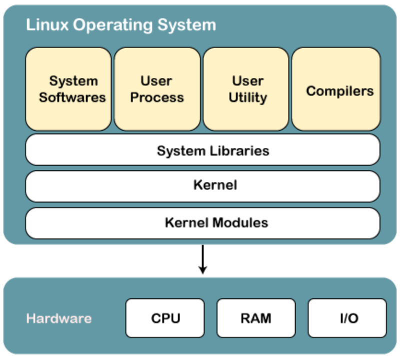

Fig 2: Linux OS

Linux kernel is the core part of the operating system. It establishes communication between devices and software. Moreover, it manages system resources. It has four responsibilities: device management, memory management, process management, and handling system calls.

System libraries are special programs that help in accessing the kernel's features. A kernel has to be triggered to perform a task, and this triggering is done by the applications. But applications must know how to place a system call because each kernel has a different set of system calls.

Programmers have developed a standard library of procedures to communicate with the kernel. Each operating system supports these standards, and then these are transferred to system calls for that operating system.

Linux OS has a set of utility tools, which are usually simple commands. It is a software which GNU project has written and publish under their open source license so that software is freely available to everyone. With the help of commands, we can access your files, edit and manipulate data in your directories or files, change the location of files, or anything.

The process of booting a Linux system typically involves the following steps:

1. The BIOS or UEFI firmware initializes the hardware and starts the boot process.

2. The bootloader, such as GRUB or LILO, is loaded from the boot device (usually a hard drive or USB drive) and begins executing.

3. The bootloader loads the Linux kernel into memory and starts it.

4. The kernel initializes the system and sets up basic services such as memory management and device drivers.

5. The kernel starts the init process, which is responsible for starting other system services.

6. The init process reads the system initialization scripts and starts various services, such as the system logger, the network stack, and the user login service.

7. Once all the necessary services have been started, the init process launches the default user interface, such as the X Window System, or it starts the default shell on text-based systems.

8. The user can then log in and begin interacting with the system.

4. Laboratory Exercise:

Learn the concepts of booting, kernel modules and process management.

5. Post-Experiment Exercise:

A. Conclusion:

#Summarize your experience about the skills acquired from this experiment.

B. Tasks:

1. Discuss the design principles of Linux OS.

2. Explain the following stages in a typical Linux boot process:

    a) BIOS, b) MBR, c) GRUB, d) Kernel, e) Init, f) Runlevel

3. What are Linux Kernel Modules? State the advantages of using Kernel Modules.

4. Perform the following and submit the screenshot of the response

    a) Loading kernel module, b) Unloading kernel module

5. Explain different types of processes in Linux.

6. Demonstrate and infer the basic commands used for process management in Linux.

    a. top

    b. ps (eg ps -f, ps -ef, ps -l etc)

    c. kill (eg kill -9, pkill etc)

    d. nice

    e. renice

    f. jobs

    g. fg

    h. bg

    i. sleep

    j. pstree

7. Demonstrate and infer input redirection and output redirection in Linux. Also explain the concepts of Overwrite redirection, Append redirection and Merge redirection.

Submission Instructions:

For each question, students need to submit the question and handwritten answer.

Wherever required, screenshot of the response should be attached.