Operating Systems

Rakesh Yadav
Associate Professor
Department of Computer Science
Shivaji College, University of Delhi
rakeshyadav@shivaji.du.ac.in

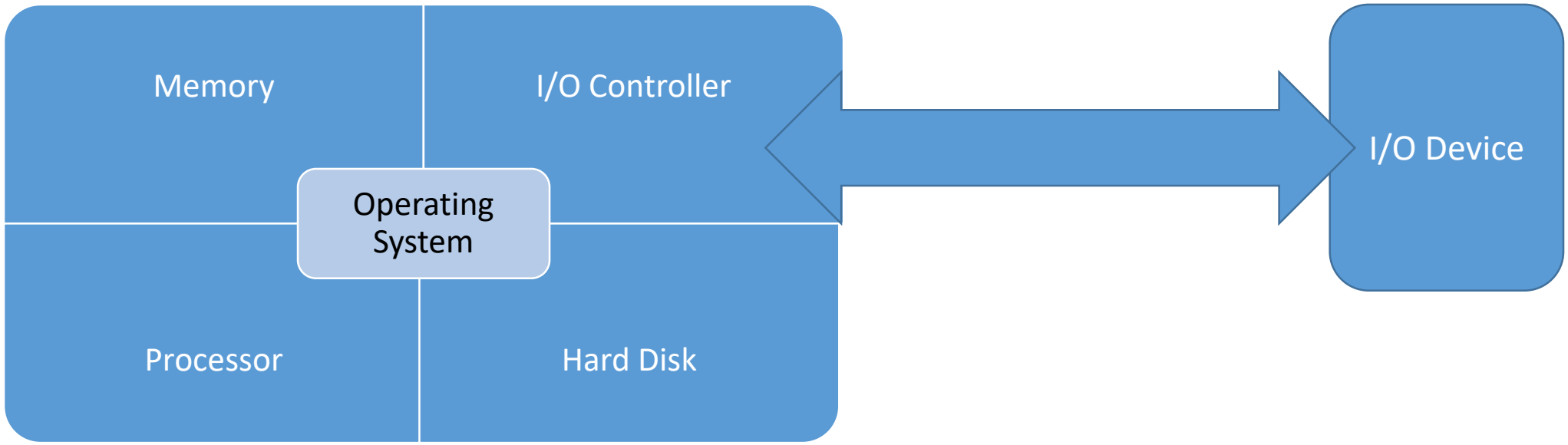
What is operating system

- An operating system is a collection of system programs that together control the operation of a computer system.
- Operating system along with hardware, application and other system software, and user constitute a computer system.
- It is the most important part of any computer system.
- It acts as an intermediate between a user and the computer hardware.

Operating system's objectives

- Firstly, an operating system controls the computer's hardware.
- The second objective is to provide an interactive interface to the user and interpret commands so that he can communicate with the hardware.

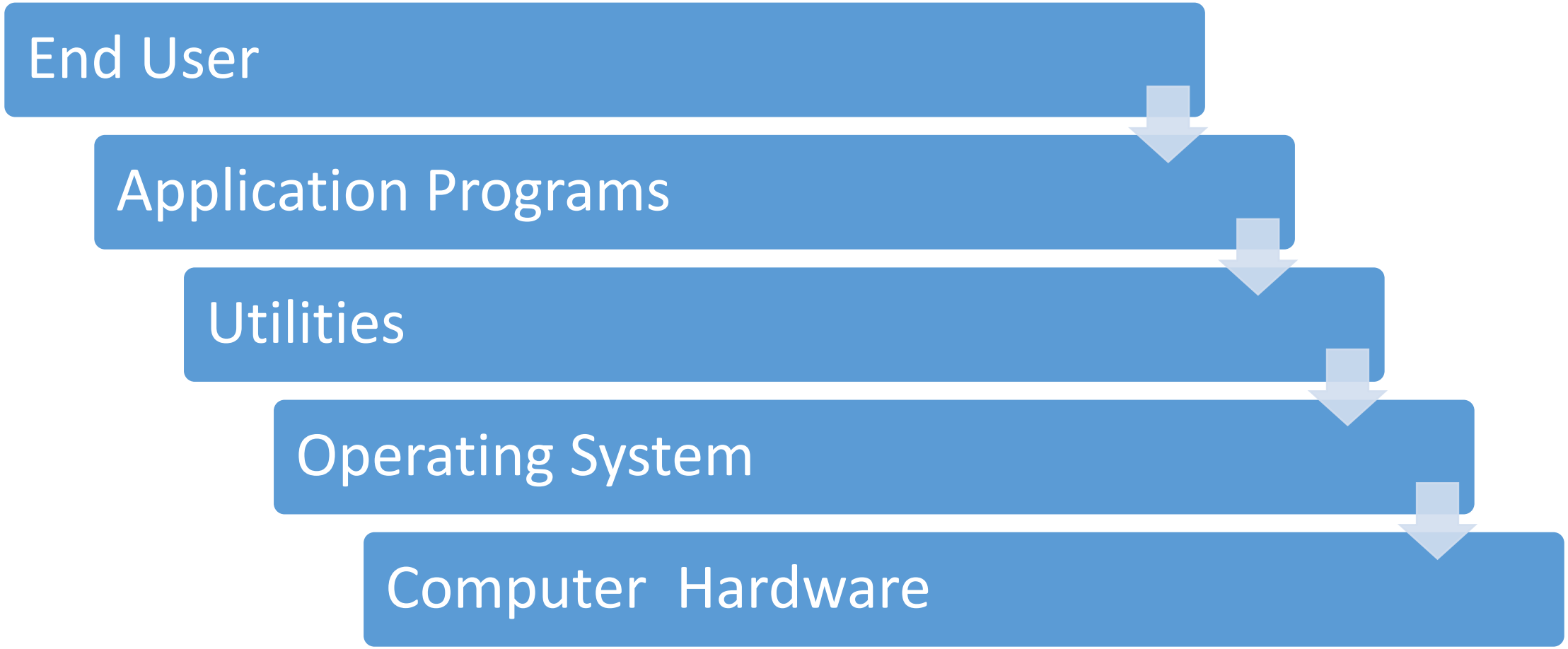
Managing the computer's hardware



Managing the computer's hardware

- The prime objective of the system is to manage and control the various hardware resources of a computer system.
- These hardware resources include processor, memory, disk space etc.
- It receives the user's input from the keyboard, and then outputs the data to the monitor.
- Operating system supervises which input device's data is requesting for being processed and which processed data is ready to be displayed on the output device.
- In addition to communicate with h/w, the OS provides and error handling procedure and displays an error notification.
- If a device is not functioning properly, the OS tries to communicate with the device again.
- If it is still unable to communicate with the device, it provides an error message notifying the user about the problem.

Providing an Interface



Providing an Interface

- The OS organizes applications so that users can easily access, use, and store them.
- When an application is opened, the OS assists the application to provide the major part of the user interface.
- It provides a stable and consistent way for applications to deal with the H/W without the user having to know all the details of the H/W.
- If the program is not functioning properly, the OS again takes control, stops the application, and displays appropriate error message.

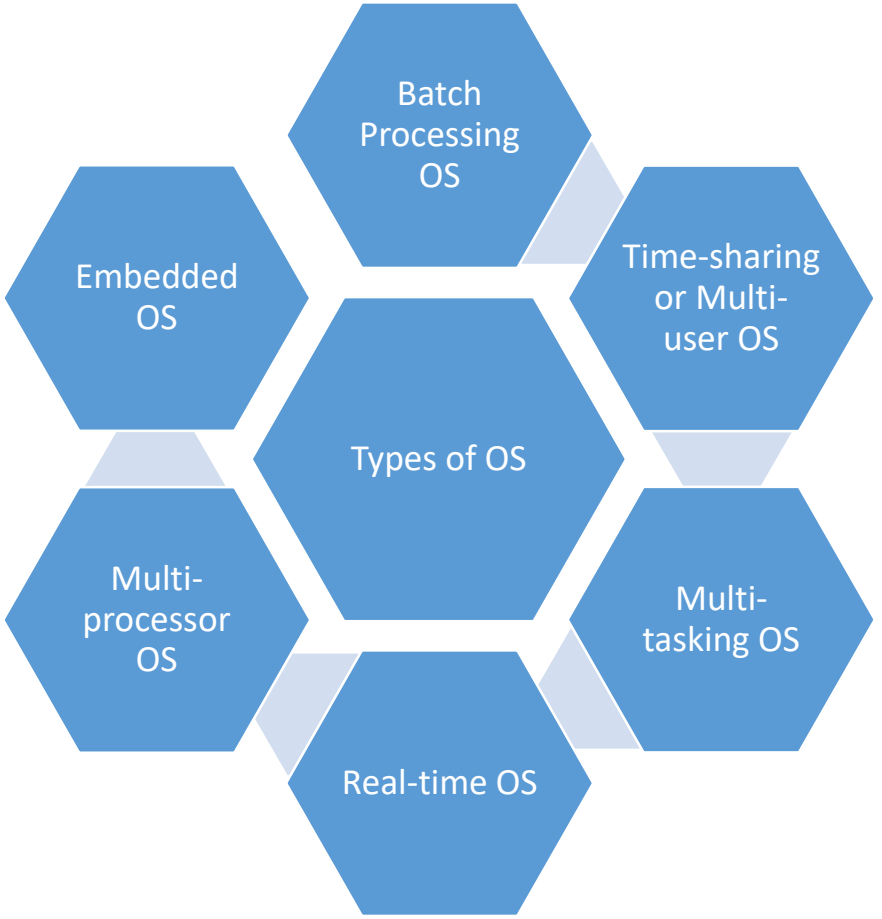
Evolution of Operating Systems

- In the early 1960s when IBM developed the S/360 series of machines.
- Then came the small 4 bit and 8 bit processor known as *microprocessors*.
- The most important of the early OS was CP/M-80 for 8080/8085/Z-80 microprocessors.
- With the development of microprocessor like 386,486, and the PENTIUM series by Intel, the whole computing world got a new dimension.
- AT&T and Microsoft came up with character-based OS, namely UNIX and DOS, which supported the then prevalent hardware architectures.
- After the character-based OS, Microsoft and Apple Macintosh came with their Windows 3.1 and MAC, which were GUI based OS and well suited for the Desktop PC market.
- Today, OS such as Windows 10,MAC,LINUX,Ubuntu and RHL etc. have taken the driver's seat in personal desktops.
- These OS with their remarkable GUI, IoT and network support features can handle diverse hardware devices.

Evolution of Operating Systems.....

- Early Systems (1950)
- Simple Batch Systems (1960)
- Multiprogrammed Batch Systems (1970)
- Time-Sharing and Real-Time Systems (1970)
- Personal/Desktop Computers (1980)
- Multiprocessor Systems (1980)
- Networked/Distributed Systems (1980)
- Web-based Systems (1990)

Types of Operating System



Batch Processing OS

- This type of OS was one of the first to evolve.
- It allowed only one program to run at a time.
- These kind of OS can still be found on some mainframe computers running batches of jobs.
- It works on a series of programs that are held in a queue.
- It is responsible for scheduling the jobs according to priority and the recourses required.
- It is good at churning through large numbers of repetitive jobs on large computers.
- Note:- This OS would be best suited for a company whishing to automate their payrolls. List of employees will be entered, their monthly salary will be calculated and corresponding pay slips would be printed.

Time-sharing or Multi-user OS

- A multi-user OS is used in computer networks which allows different users to access the same data and application programs on the same network.
- It also allows users to communicate with each other.
- The multi-user OS builds a user database account, which defines the right that users have on a particular resource of the system.

Multi-tasking OS

- In multi-tasking OS, more than one process can be executed concurrently.
- The processor is swathed rapidly between the processes.
- A user may run more than one process at a time.
- It is quite common that a user on his computer can have a word processor open and running, an audio DVD/PD playing and he might be browsing the internet, all at same time.

Real-time OS

- RTOS are designed to respond to an event within a predetermined time.
- It is primarily used in process control, telecommunications etc.
- It monitors various inputs which affect the execution of processes, changing the computers model of the environment, thus affecting the outputs, within a guaranteed time period(usually less than 1 second).
- As the real-time OS respond quickly, they are often used in applications such as air flight or railway reservation booking.

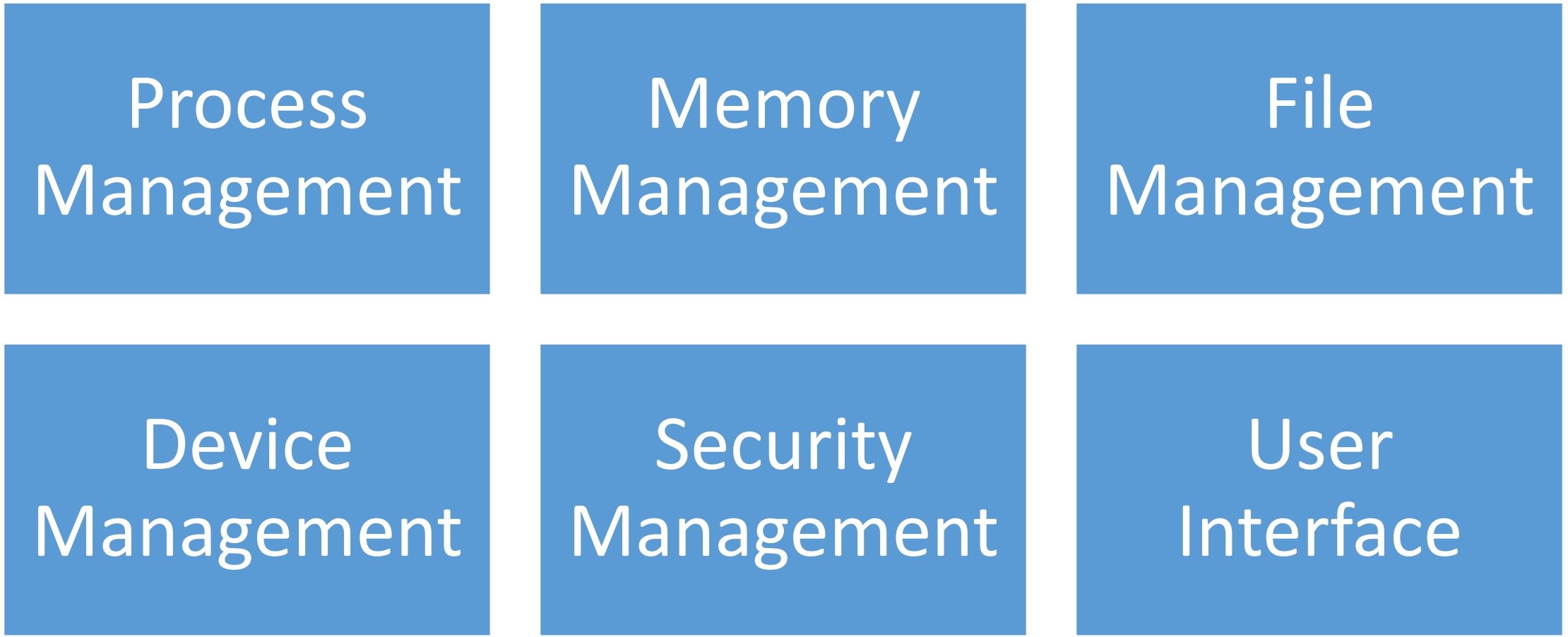
Multi-processor OS

- A multi-processor OS can incorporate more than one processor dedicated to running processes.
- This technique of using more than one processor is often called parallel processing.

Embedded OS

- An embedded OS refers to the OS that is self-contained in the device and resident in ROM.
- Since embedded systems are usually not general-purpose systems, these OS are lighter or less resource intensive as compared to general purpose OS.
- Most of these OS also offer real-time OS qualities.
- Typically systems that use embedded operating systems are household appliances, car management systems, traffic control systems and emergency management systems etc.

Functions of an Operating System



Functions of an Operating System

- Process Management:- As a process manager, the operating system handles the creation and deletion or processes, suspension and resumption of processes and scheduling and synchronization of processes.
- Memory Management:- As a memory manager , the operating system handles allocation and deallocation of memory space as required by various programs.
- File Management:- It is responsible for creation and deletion of files and directories. It also takes care of other file-related activities such as organizing, storing, retrieving, naming and protecting the files.

Functions of an Operating System…

- Device Management:- Operating system provides input/output subsystem between process and device driver.
- It handles the device caches, buffers and interrupts.
- It also detects device failures and notifies the same to the user.
- Security Management:- The OS protects system resources and information against destruction and unauthorized use.
- User Interface:- OS provides the interface between the user and the hardware. The user interface is the layer that actually interacts with the computer operator. The interface consists of a set of commands or menus through which a user communicates with a program

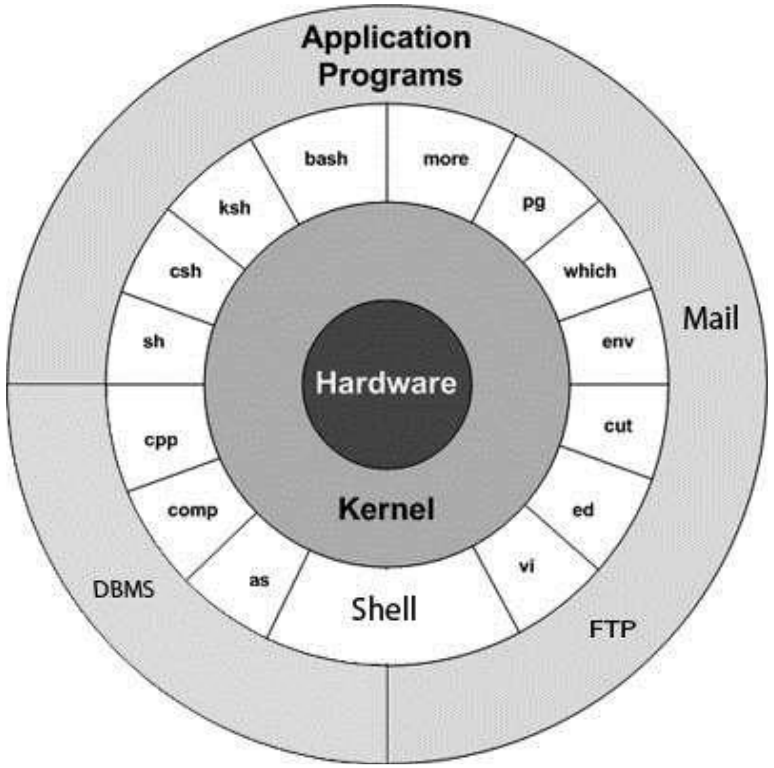
What is Unix ?

- The UNIX operating system is a set of programs that act as a link between the computer and the user.
- The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the operating system or kernel.
- Users communicate with the kernel through a program known as the shell.
- The shell is a command line interpreter.
- It translates commands entered by the user and converts them into a language that is understood by the kernel.

....

- Unix was originally developed in 1969 by a group of AT&T employees at Bell Labs, including Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna.
- There are various Unix variants available in the market. Solaris Unix, AIX, HP Unix and BSD are few examples. Linux is also a flavor of Unix which is freely available.
- Several people can use a UNIX computer at the same time; hence UNIX is called a multiuser system.
- A user can also run multiple programs at the same time; hence UNIX is called multitasking.

Unix Architecture



....

- **Kernel:** The kernel is the heart of the operating system. It interacts with hardware and most of the tasks like memory management, tash 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 most famous shells which are available with most of the Unix variants.
- **Commands and Utilities:** There are various command and utilities which you would use in your day to day activities. **cp, mv, cat** and **grep (**globally search for regular expression and print out) 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 optional options.
- **Files and Directories:** All data in UNIX is organized into files. All files are organized into directories. These directories are organized into a tree-like structure called the file system.

What is Shell?

- The original Unix shell was written in the mid-1970s by Stephen R. Bourne while he was at the AT&T Bell Labs in New Jersey.
- Bourne shell was the first shell to appear on Unix systems, thus it is referred to as "the shell".
- Bourne shell is usually installed as **/bin/sh** on most versions of Unix. For this reason, it is the shell of choice for writing scripts that can be used on different versions of Unix.

What is shell and various type of shell?

- A **Shell** provides you with an interface to the Unix system.
- It gathers input from you and executes programs based on that input.
- When a program finishes executing, it displays that program's output.
- Shell is an environment in which we can run our commands, programs, and shell scripts.
- There are different flavors of a shell, just as there are different flavors of operating systems.
- Each flavor of shell has its own set of recognized commands and functions.

What is shell and various type of shell?

- Shell Prompt
- The prompt, **$**, which is called the **command prompt**, is issued by the shell. While the prompt is displayed, you can type a command.
- Shell reads your input after you press **Enter**.
- It determines the command you want executed by looking at the first word of your input.
- A word is an unbroken set of characters. Spaces and tabs separate words.
- For example of the **date** command, which displays the current date and time −
- $date

Shell Types

- In Unix, there are two major types of shells −
  - **Bourne shell** − If you are using a Bourne-type shell, the **$** character is the default prompt.
  - **C shell** − If you are using a C-type shell, the % character is the default prompt.

The Bourne Shell has the following subcategories −

- Bourne shell (sh)
- Korn shell (ksh)
- Bourne Again shell (bash)
- POSIX shell (sh)

The different C-type shells follow −

- C shell (csh)
- TENEX/TOPS C shell (tcsh)

Various editors present in LINUX

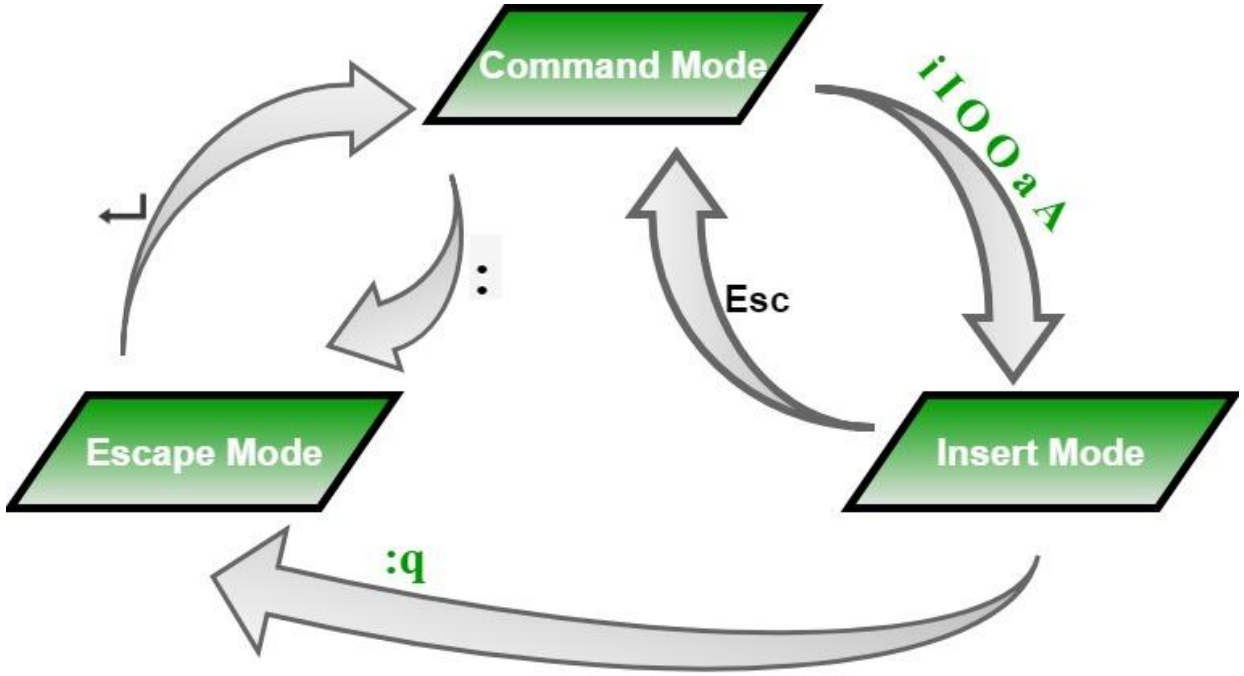
- A text editor is a program used for editing text files. Most configuration of Linux systems is done by editing text files. All Linux distributions ship with multiple text editors included.
- There are two types of text editors in Linux:
- **commandline editors** – vi, nano, pico,emac etc.
- **GUI editors** – gedit (for GNOME), KWrite (for KDE)
- If we want to edit a file called **textfile.txt** using *gedit*, we would type *gedit textfile.txt*. This opens up **textfile.txt** for editing:

Various editors present in LINUX

- The default editor that comes with the UNIX operating system is called vi (visual editor).
- We can edit an existing file or create a new file.
- we can also use this editor to just read a text file.
- An improved version of the vi editor which is called the **VIM (Vi IM**proved) has also been made available now.

Modes of Operation in vi editor

- There are three modes of operation in vi:-



- **Command Mode:** When vi starts up, it is in Command Mode. This mode is where vi interprets any characters we type as commands and thus does not display them in the window. This mode allows us to move through a file, and to delete, copy, or paste a piece of text.
  To enter into Command Mode from any other mode, it requires pressing the **[Esc]** key. If we press [Esc] when we are already in Command Mode, then vi will beep or flash the screen.
- **Insert mode:** This mode enables you to insert text into the file. Everything that's typed in this mode is interpreted as input and finally, it is put in the file. The vi always starts in command mode. To enter text, you must be in insert mode. To come in insert mode you simply type i. To get out of insert mode, press the Esc key, which will put you back into command mode.
- **Escape Mode:** Line Mode is invoked by typing a colon [:], while vi is in Command Mode. The cursor will jump to the last line of the screen and vi will wait for a command. This mode enables you to perform tasks such as saving files, executing commands.

- ***vi filename:*** *Creates a new file if it already not exist, otherwise opens existing file.*
- ***vi -R filename*** *: Opens an existing file in read only mode.*
- ***view filename*** *: Opens an existing file in read only mode.*

Moving within a File(Navigation):

- To move around within a file without affecting text must be in command mode (press Esc twice). Here are some of the commands can be used to move around one character at a time.
- ***k*** *: Moves the cursor up one line.*
- ***j*** *: Moves the cursor down one line.*
- ***h*** *: Moves the cursor to the left one character position.*
- ***l*** *: Moves the cursor to the right one character position.*
- ***0 or |*** *: Positions cursor at beginning of line.*
- ***$*** *: Positions cursor at end of line.*

- ***W** : Positions cursor to the next word.*
- ***B** : Positions cursor to previous word.*
- ***(** : Positions cursor to beginning of current sentence.*
- ***)** : Positions cursor to beginning of next sentence.*
- ***H** : Move to top of screen.*
- ***nH** : Moves to nth line from the top of the screen.*
- ***M** : Move to middle of screen.*
- ***L** : Move to bottom of screen.*
- ***nL** : Moves to nth line from the bottom of the screen.*
- ***colon along with x** : Colon followed by a number would position the cursor on line number represented by x.*

Control Commands(Scrolling): There are following useful commands which can used along with **Control Key**:

- ***CTRL+d*** *: Move forward 1/2 screen.*
- ***CTRL+f*** *: Move forward one full screen.*
- ***CTRL+u*** *: Move backward 1/2 screen.*
- ***CTRL+b*** *: Move backward one full screen.*
- ***CTRL+e*** *: Moves screen up one line.*
- ***CTRL+y*** *: Moves screen down one line.*
- ***CTRL+u*** *: Moves screen up 1/2 page.*
- ***CTRL+d*** *: Moves screen down 1/2 page.*
- ***CTRL+b*** *: Moves screen up one page.*
- ***CTRL+f*** *: Moves screen down one page.*
- ***CTRL+l*** *: Redraws screen.*

Editing and inserting in Files(Entering and Replacing Text):

To edit the file, we need to be in the insert mode. There are many ways to enter insert mode from the command mode.

- ***i*** *: Inserts text before current cursor location.*
- ***I*** *: Inserts text at beginning of current line.*
- ***a*** *: Inserts text after current cursor location.*
- ***A*** *: Inserts text at end of current line.*
- ***o*** *: Creates a new line for text entry below cursor location.*
- ***O*** *: Creates a new line for text entry above cursor location.*
- ***r*** *: Replace single character under the cursor with the next character typed.*
- ***R*** *: Replaces text from the cursor to right.*
- ***s*** *: Replaces single character under the cursor with any number of characters.*
- ***S*** *:Replaces entire line.*

Deleting Characters: list of important commands which can be used to delete characters and lines in an opened file.

- ***X*** *Uppercase: Deletes the character before the cursor location.*
- ***x*** *Lowercase : Deletes the character at the cursor location.*
- ***Dw*** *: Deletes from the current cursor location to the next word.*
- ***d^*** *: Deletes from current cursor position to the beginning of the line.*
- ***d$*** *: Deletes from current cursor position to the end of the line.*
- ***Dd*** *: Deletes the line the cursor is on.*

Copy and Past Commands:

- ***Yy*** *: Copies the current line.*
- ***9yy*** *: Yank current line and 9 lines below.*
- ***p*** *: Puts the copied text after the cursor.*
- ***P*** *: Puts the yanked text before the cursor.*

Save and Exit Commands of the ex Mode : Need to press **[Esc]** key followed by the colon (:) before typing the following commands:

- ***q*** *: Quit*
- ***q!*** *: Quit without saving changes i.e. discard changes.*
- ***r fileName*** *: Read data from file called fileName.*
- ***wq*** *: Write and quit (save and exit).*
- ***w fileName*** *: Write to file called fileName (save as).*
- ***w! fileName*** *: Overwrite to file called fileName (save as forcefully).*
- ***!cmd*** *: Runs shell commands and returns to Command mode.*

Searching and Replacing in (ex Mode):

- **vi** also has powerful search and replace capabilities. The formal syntax for searching is:
- :s/string (For example, suppose we want to search some text for the string).
- :s/shivajicollege.
- The syntax for replacing one string with another string in the current line is:
- :s/pattern/replace/ (Here “pattern” represents the old string and “replace” represents the new string. For example, to replace each occurrence of the word “geeks” in a line with “geeksforgeeks” type).
- :s/geeksforgeeks/gfg/

- The syntax for replacing every occurrence of a string in the entire text is similar. The only difference is the addition of a “%” in front of the “s”:
- :%s/pattern/replace/

Nano Text Editor in Linux

- Nano is a user-friendly, simple and WYSIWYG(What You See Is What You Get) text editor, which improves the features and user-friendliness of UW Pico text editor.
- Unlike vim editor or any other command-line editor, it doesn’t have any mode.
- It has an easy GUI(Graphical User Interface) which allows users to interact directly with the text in spite of switching between the modes as in vim editor.
- Nano is generally by default available in many Linux distributions.

- **1.** To create and open a new file.
- $nano new_filename
- **2.** To save a file
- Press crtl+o
- **3.** To cut paste in a file. *Ctrl+o* is used to cut and *Ctrl+u* is used to paste the text.
- *To cut and paste a whole line*. Move to the line which you want to cut then press *Ctrl+k*. Now the line is moved to clipboard, To paste it, go to the position where you want to paste and then press *Ctrl+u.*
- *To cut and paste the selected text*. Select the text which you want to cut then press *Ctrl+k*. Now the text is moved to clipboard. To paste it, go to the position where you want to paste and then press *Ctrl+u*.

- To search a word in a file. *Ctrl+w* is used.
- *Press Ctrl+w*
  It will ask for a word to search for.
  Enter the word
  It will search for the word and will place the cursor in the first letter of the first occurrence of the word.
- To enable spell check in nano. First, install the spell check package.
- $ sudo apt install spell
- To do spell check first press Ctrl+t
- Now it will ask you to replace the incorrect words
- Enter the word to replace with there
- As soon as you will press the enter key

References

- A Silberschatz, P.B. Galvin, G. Gagne, **Operating Systems Concepts**, 9th Edition,John Wiley Publications.
- Unix: Concepts and Applications, Sumitabha Das, TMH, 4th Edition, 2009.
- Introduction to Computer Science, ITL Education Solutions Limited.
- https://www.geeksforgeeks.org.
- https://www.tutorialspoint.com.