OPERATING SYSTEM

OS – 3 Lectures:

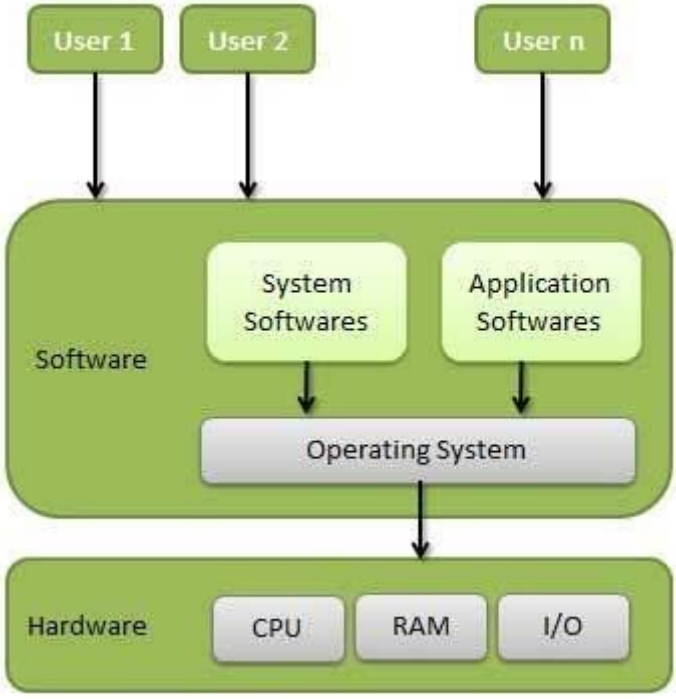
- Introduction
- Memory Management
- Scheduling/Dead Lock

Networks – 2 Lectures

System Design – 3 Lectures

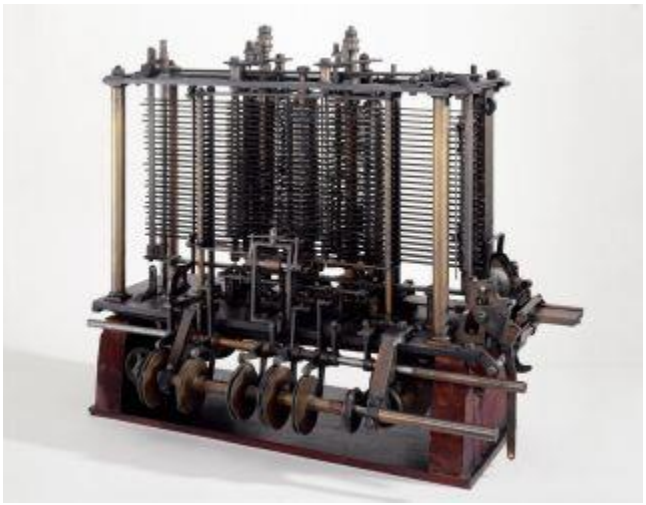
Operating System:

An Operating System (OS) ***is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.***



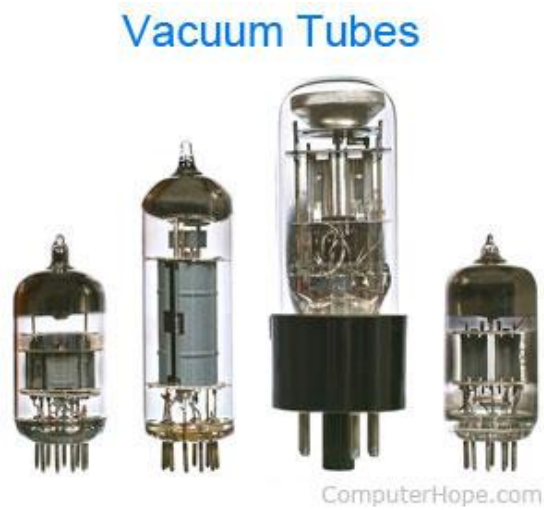
Five of the most common operating systems are Microsoft Windows, Apple macOS, Linux, Android and Apple's iOS.

History:



The first computer invented by man was **ABACUS**. In Abacus, we were able to do calculations only till 10 digits. To do calculations for more than 10 digits, Charles Babbage, an English Mathematician, conceives of a steam-driven calculating machine that would be able to compute tables of numbers. He invented a mechanical machine which could do calculations. He is called the Father of Computer.

During WW1, countries sent their spies into the enemy territory, so that they would give them information in such a way that other people would not understand. To get this secret message or encrypted message, it was put into such machines that would give decrypted messages. At this point, the importance of computer grew.



After WW1, **vacuum tubes** referred to as an **electron tube** first developed by **John Ambrose Fleming** in **1904** is a glass tube that has its gas removed, creating a vacuum. They contain electrodes for controlling electron flow and were used in early computers as a switch. In the **1950**s, the ***transistor*** started to **replace** the **vacuum tube**. As computing devices started to become smaller in size, transistors were more ideal to use due to their smaller size as well. **Vacuum tubes** used for the **initial computers** were of **huge size** (two floor-building).

Second Generation of Computers

Transistor

But the **invention of transistors** changed the whole idea of computer from being so huge and not affordable by many to pocket size and in-hand for everyone. People never thought the a computer could be carried in a pocket.

People would have a very big computer of about two floor height. As a college student, if you need to get some output, you supply some input and the computer will give you output. So, if you talk to the operator, he would give you ‘punch card’. There was no storage concept. You write your input on the punch card and give it to the computer and it would give you output. It was a time taking process. But now every one has his/her own PC. All this development has occurred in span of 50 years.

As technology grew, people use keyboard, mouse, RAM, monitor and all of this is present in your laptop. All of these things are managed by **OPERATING SYSTEM**. Operating System is a interface between user and hardware that helps us to manage all the applications and hardware in a computer.

In a computer all the instructions are in binary (0’s and 1’s). To write a program in binary code is a tedious process. This is where OS comes into play. Whenever we write a python code and run it, it is converted into machine code or assembly code by operating system.

What happens when you switch on your computer?

Every computer has a motherboard. It has all the important things that a computer needs. It has space for CPU, RAM, ROM, Hard-disk, BIOS Chips etc. When the power switch is pushed, the operating system code which resides in your hard disk; is read by ROM – read only memory (it can never be deleted) in this there is a boot loader. It knows where in HDD the operating system code is present. It fetches the code and puts in the RAM; this switches our computer ON.

Memory:

There are two types of memory;

1. Primary Memory – Volatile (RAM)
2. Secondary Memory – Non-Volatile

Primary Memory – Volatile (RAM)

The data in this memory will persist as long as the electricity is available. Once you turn off the computer the Primary Memory is lost. Primary memory is computer memory that is accessed directly by the CPU. This includes several types of memory, such as the processor cache and system ROM. However, in most cases, primary memory refers to system RAM.

Secondary Memory – Non-Volatile (HDD, SDD, ROM)

Secondary memory refers to storage devices, such as hard drives and solid-state drives. It may also refer to removable storage media, such as USB flash drives, CDs, and DVDs. The data in this memory is stored. Unlike primary memory, secondary memory is not accessed directly by the CPU. Instead, data accessed from secondary memory is first loaded into RAM and is then sent to the processor. The RAM plays an important intermediate role, since it provides much faster data access speeds than secondary memory. By loading software programs and files into primary memory, computers can process data much more quickly.

All the programs which are running in our computer is stored in RAM. And once the computer if turned off, the RAM is made empty. Simple example, if we have more RAM, then we can run more programs.

Process:

A process is a program in execution. Any program which is running in the computer is called process. Every process will have a unique ID called, '***Pid***'

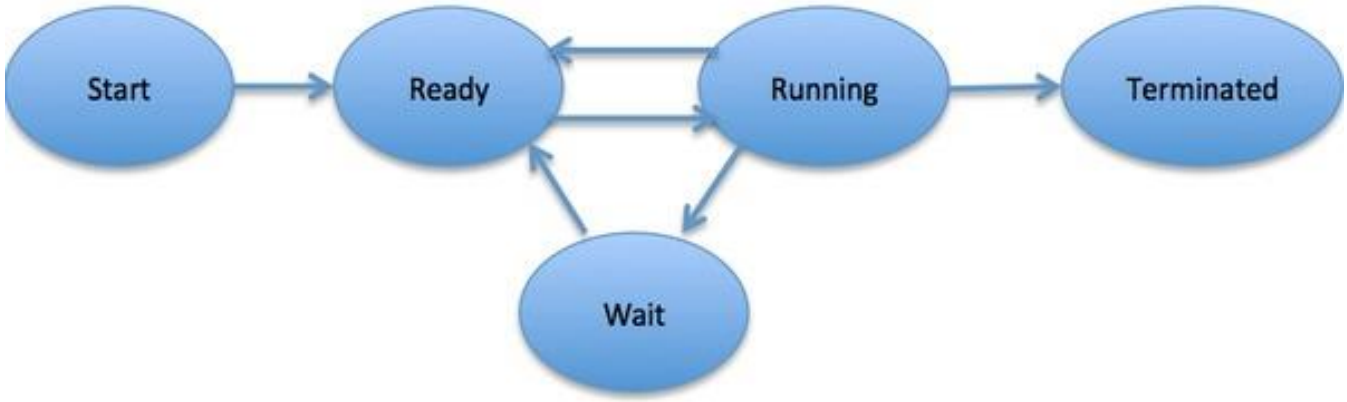
Life Cycle of a Process:

When a process executes, it passes through different states. These stages may differ in different operating systems, and the names of these states are also not standardized.

In general, a process can have one of the following five states at a time.

S.N.	State & Description
1	**Start** This is the initial state when a process is first started/created.
2	**Ready** The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run. Process may come into this state after Start state or while running it by but interrupted by the scheduler to assign CPU to some other process.
3	**Running**

	Once the process has been assigned to a processor by the OS scheduler, the process state is set to running and the processor executes its instructions.
4	**Waiting** Process moves into the waiting state if it needs to wait for a resource, such as waiting for user input, or waiting for a file to become available.
5	**Terminated or Exit** Once the process finishes its execution, or it is terminated by the operating system, it is moved to the terminated state where it waits to be removed from main memory.

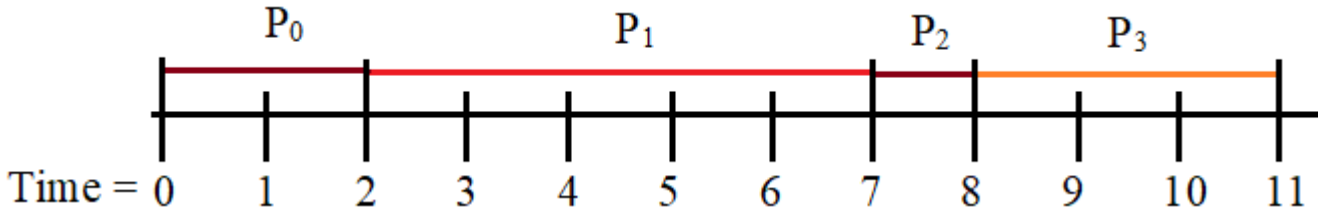


CPU Utilization:

P ST CT ST – Schedule Time CT – Completion Time

P_0	0	2
P_1	0	5
P_2	0	1
P_3	0	3

CT is completion time for a particular process to complete. In a CPU,



Total time CPU is getting used is 11 times and the total time is 11.

CPU Utilization = No.of times CPU is getting used / Total Time = (11 / 11) *100 = 100%. A healthy number for CPU utilization is 70%.

Turn Over Time:

It means, the amount of time taken to complete a process or fulfill a request

Now P_0, took 2 sec, so the turnover time is 2 sec.

Now P_1, took 7 sec, so the turnover time is 7 sec, but since P_1 started after P_0 was finished so the user will see the lag of 2 sec.

Now P_2, = 8 sec and for P_3 is 11 sec.

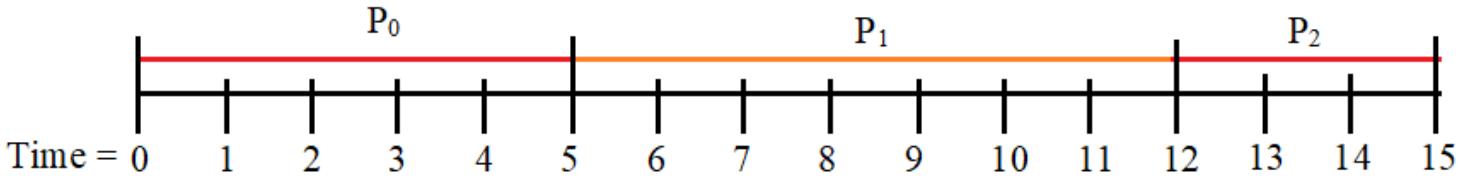
So, the average turnover time = **(2 + 7 + 8 + 11) / 4**

Response Time:

Response Time is the time spent between the ready state and getting the CPU for the first time. **Waiting Time** is the total time taken by the process in the ready state.

P	**ST**	**CT**
P_0	0	5
P_1	3	7
P_2	5	3

ST – Schedule Time CT – Completion Time



Waiting Time:

P_0 0 – 0 = 0.

P_1 5 – 3 = 2. It should start at t = 3, but it started at t = 5, so the waiting time = 2

P_2 12 – 5 = 7, It should start at t = 5, but it started at t = 12, so the waiting time = 7

So, Average waiting time is = **(0 + 2 + 7) / 3**

This is how we measure the efficiency of our OS. OS manages how a process should come to the CPU so that our waiting time is less. One algorithm is "**first come, first served**" and another algorithm, "**shortest job first**".

Shortest Job First:

At t = 0, 1 & 2, P0 is being executed. At t = 3, P0 is put on hold and P1 is being executed as it completion time is 1 sec. Once P1 is done, then at t = 4 P0 is being executed until t = 5. At t = 5, P0 is completed and P2 begins to execute.

This is called Pre-emptive Shortest Job First

In Preemptive Shortest Job First Scheduling, jobs are put into ready queue as they arrive, but as a process with short burst time arrives, the existing process is preempted or removed from execution, and the shorter job is executed first.

https://afteracademy.com/blog/what-is-burst-arrival-exit-response-waiting-turnaround-time-and-throughput