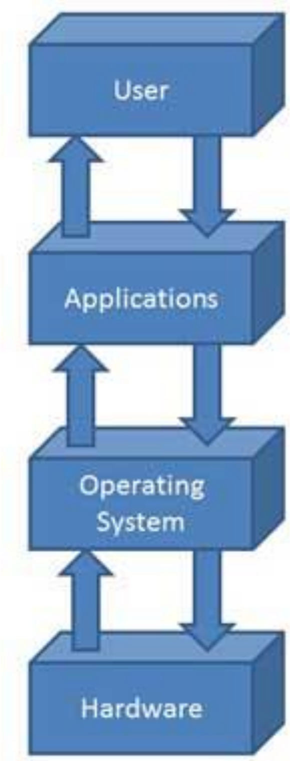
Unit – I Operating System Basics

Basic Concepts of Operating System - Services of Operating System-Classification of Operating System- Architecture and Design of an Operating System-Process Management - Introduction to Process-Process State -PCB - Process Scheduling - Interprocess Communication

The Operating System is a program with the following features −

- An operating system is a program that acts as an interface between the software and the computer hardware.
- It is an integrated set of specialized programs used to manage overall resources and operations of the computer.
- It is specialized software that controls and monitors the execution of all other programs that reside in the computer, including application programs and other system software.



Objectives of Operating System

The objectives of the operating system are –

- To make the computer system convenient to use in an efficient manner.
- To hide the details of the hardware resources from the users.
- To provide users a convenient interface to use the computer system.
- To act as an intermediary between the hardware and its users, making it easier for the users to access and use other resources.
- To manage the resources of a computer system.
- To keep track of who is using which resource, granting resource requests, and mediating conflicting requests from different programs and users.
- To provide efficient and fair sharing of resources among users and programs.

Characteristics of Operating System

Here is a list of some of the most prominent characteristic features of Operating Systems −

- **Memory Management** − Keeps track of the primary memory, i.e. what part of it is in use by whom, what part is not in use, etc. and allocates the memory when a process or program requests it.
- **Processor Management** − Allocates the processor (CPU) to a process and deallocates the processor when it is no longer required.
- **Device Management** − Keeps track of all the devices. This is also called I/O controller that decides which process gets the device, when, and for how much time.
- **File Management** − Allocates and de-allocates the resources and decides who gets the resources.
- **Security** − Prevents unauthorized access to programs and data by means of passwords and other similar techniques.
- **Job Accounting** − Keeps track of time and resources used by various jobs and/or users.
- **Control Over System Performance** − Records delays between the request for a service and from the system.

- **Interaction with the Operators** − Interaction may take place via the console of the computer in the form of instructions. The Operating System acknowledges the same, does the corresponding action, and informs the operation by a display screen.
- **Error-detecting Aids** − Production of dumps, traces, error messages, and other debugging and error-detecting methods.
- **Coordination Between Other Software and Users** − Coordination and assignment of compilers, interpreters, assemblers, and other software to the various users of the computer systems.

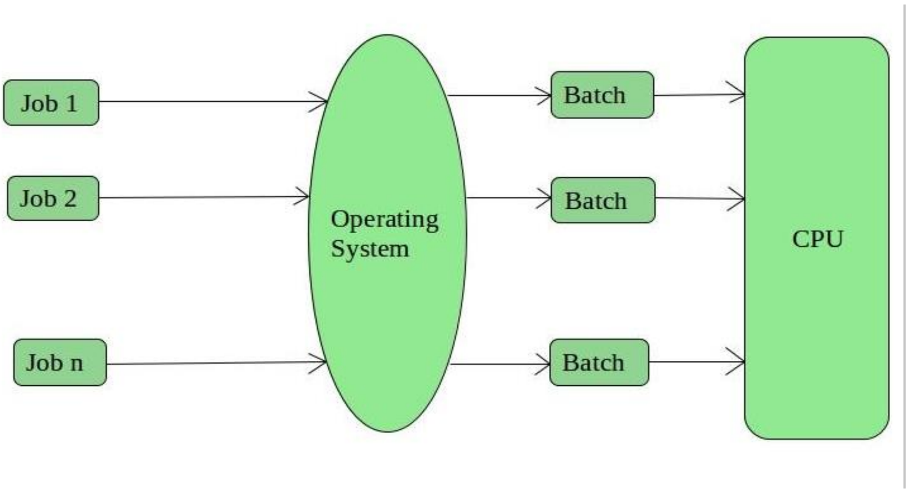
Types of Operating Systems

An Operating System performs all the basic tasks like managing file,process, and memory. Thus operating system acts as manager of all the resources, i.e. **resource manager**. Thus operating system becomes an interface between user and machine.

Types of Operating Systems: Some of the widely used operating systems are as follows-

1.BatchOperatingSystems

This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having same requirement and group them into batches. It is the responsibility of operator to sort the jobs with similar needs.



Advantages of Batch Operating System:

- It is very difficult to guess or know the time required by any job to complete. Processors of the batch systems know how long the job would be when it is in queue

- Multiple users can share the batch systems
- The idle time for batch system is very less
- It is easy to manage large work repeatedly in batch systems

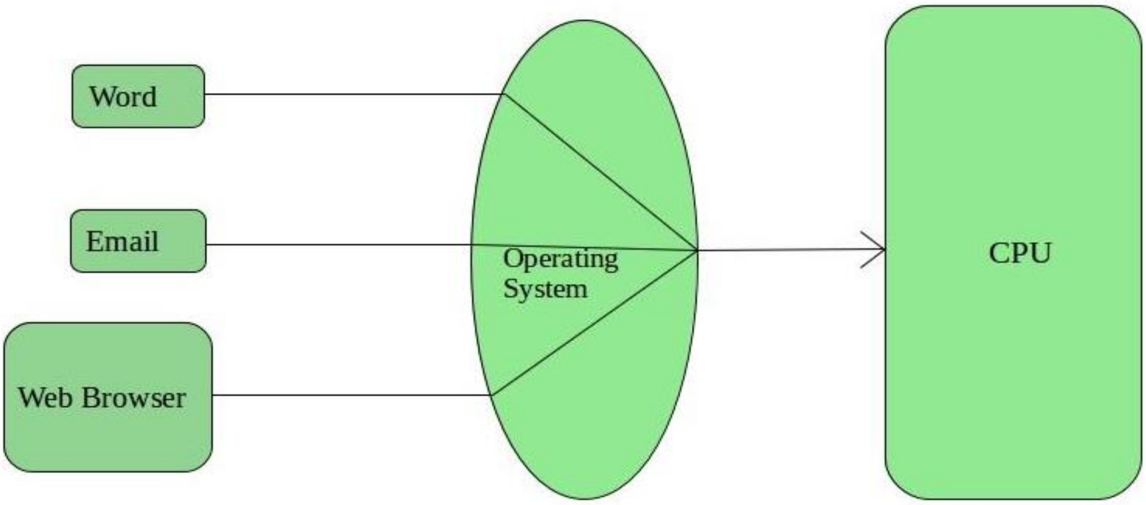
Disadvantages of Batch Operating System:

- The computer operators should be well known with batch systems
- Batch systems are hard to debug
- It is sometime costly
- The other jobs will have to wait for an unknown time if any job fails

Examples of Batch based Operating System: Payroll System, Bank Statements etc.

2. Time-Sharing Operating Systems –

Each task is given some time to execute, so that all the tasks work smoothly. Each user gets time of CPU as they use single system. These systems are also known as Multitasking Systems. The task can be from single user or from different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to next task.



Advantages of Time-Sharing OS:

- Each task gets an equal opportunity
- Less chances of duplication of software
- CPU idle time can be reduced

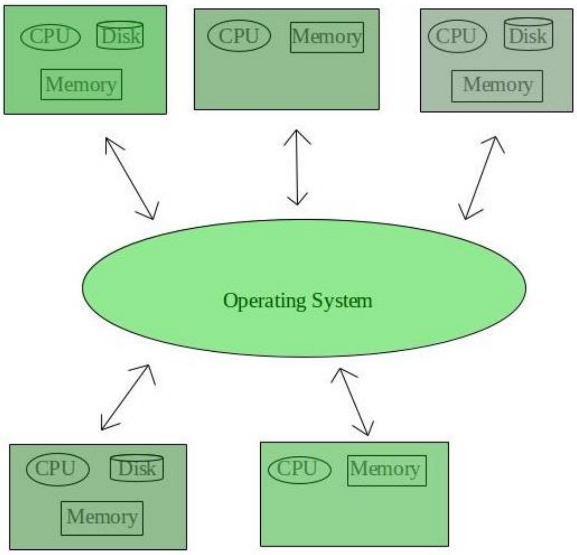
Disadvantages of Time-Sharing OS:

- Reliability problem
- One must have to take care of security and integrity of user programs and data
- Data communication problem

Examples of Time-Sharing OS are: Multics, Unix etc.

3. Distributed Operating System –

These types of operating system is a recent advancement in the world of computer technology and are being widely accepted all-over the world and, that too, with a great pace. Various autonomous interconnected computers communicate each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred as **loosely coupled systems** or distributed systems. These system's processors differ in size and function. The major benefit of working with these types of operating system is that it is always possible that one user can access the files or software which are not actually present on his system but on some other system connected within this network i.e., remote access is enabled within the devices connected in that network.



Advantages of Distributed Operating System:

- Failure of one will not affect the other network communication, as all systems are independent from each other
- Electronic mail increases the data exchange speed
- Since resources are being shared, computation is highly fast and durable

- Load on host computer reduces
- These systems are easily scalable as many systems can be easily added to the network
- Delay in data processing reduces

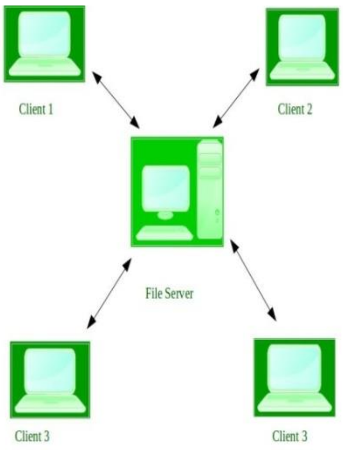
Disadvantages of Distributed Operating System:

- Failure of the main network will stop the entire communication
- To establish distributed systems the language which are used are not well defined yet
- These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet

Examples of Distributed Operating System are- LOCUS etc.

4. Network Operating System –

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These type of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network. One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections etc. and that’s why these computers are popularly known as **tightly coupled systems**.



Advantages of Network Operating System:

- Highly stable centralized servers
- Security concerns are handled through servers
- New technologies and hardware up-gradation are easily integrated to the system
- Server access are possible remotely from different locations and types of systems

Disadvantages of Network Operating System:

- Servers are costly
- User has to depend on central location for most operations
- Maintenance and updates are required regularly

Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD etc.

5. Real-Time Operating System –

These types of OS serve the real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called **response time**.

Real-time systems are used when there are time requirements are very strict like missile systems, air traffic control systems, robots etc.

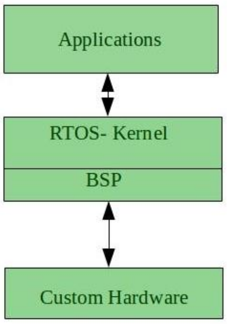
Two types of Real-Time Operating System which are as follows:

- **Hard Real-Time Systems:**

  These OS are meant for the applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or air bags which are required to be readily available in case of any accident. Virtual memory is almost never found in these systems.

- **Soft Real-Time Systems:**

  These OS are for applications where for time-constraint is less strict.



Advantages of RTOS:

- **Maximum Consumption:** Maximum utilization of devices and system,thus more output from all the resources

- **Task Shifting:** Time assigned for shifting tasks in these systems are very less. For example in older systems it takes about 10 micro seconds in shifting one task to another and in latest systems it takes 3 micro seconds.
- **Focus on Application:** Focus on running applications and less importance to applications which are in queue.
- **Real time operating system in embedded system:** Since size of programs are small, RTOS can also be used in embedded systems like in transport and others.
- **Error Free:** These types of systems are error free.
- **Memory Allocation:** Memory allocation is best managed in these type of systems.

Disadvantages of RTOS:

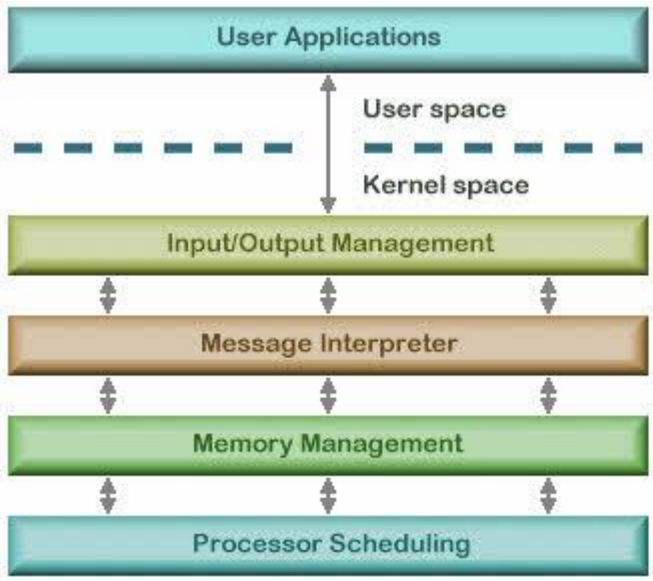
- **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on few applications to avoid errors.
- **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
- **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
- **Device driver and interrupt signals:** It needs specific device drivers and interrupt signals to response earliest to interrupts.
- **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

Examples of Real-Time Operating Systems are: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

Architecture of Operating systems :

The operating systems control the hardware resources of a computer. The kernel and shell are the parts of the operating system that perform essential operations.

When a user gives commands for performing any operation, the request goes to the shell part, which is also known as interpreter. The shell part then translates the human program into a machine code, and then transfers the request to the kernel part.



Architecture of operating system

When the kernel receives the request from the shell, it processes the request and displays the result on the screen. The kernel is also known as the heart of the operating system as every operation is performed by it.

Shell

The shell is a part of the software which is placed between the user and the kernel, and it provides services of the kernel. The shell thus acts as an interpreter to convert the commands from the user to a machine code. Shells present in different types of operating systems are of two types: command line shells and graphical shells.

The command line shells provide a command line interface while graphical line shells provide a graphical user interface. Though both shells perform operations, the graphical user interface shells perform slower than the command line interface shells.

Types of shells

- Korn shell
- Bourne shell
- C shell
- POSIX shell

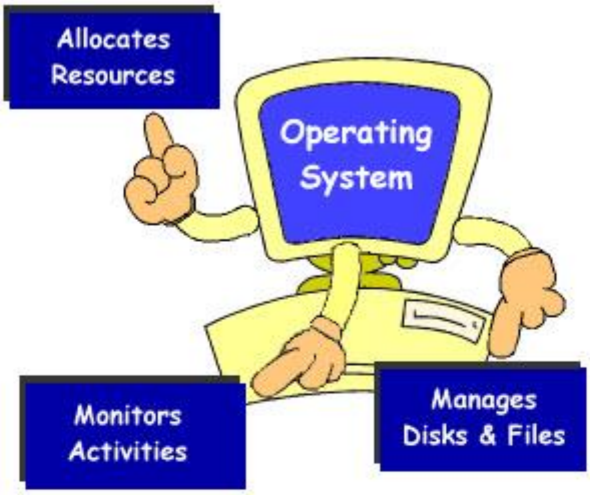
Kernel

The kernel is a part of software. It is like a bridge between the shell and hardware. It is responsible for running programs and providing secure access to the machine's hardware. The kernel is used for scheduling, i.e., it maintains a time table for all processes.

Types of Kernels

- Monolithic kernel
- Microkernels
- Exokernels
- Hybrid kernels

Operating System Functions



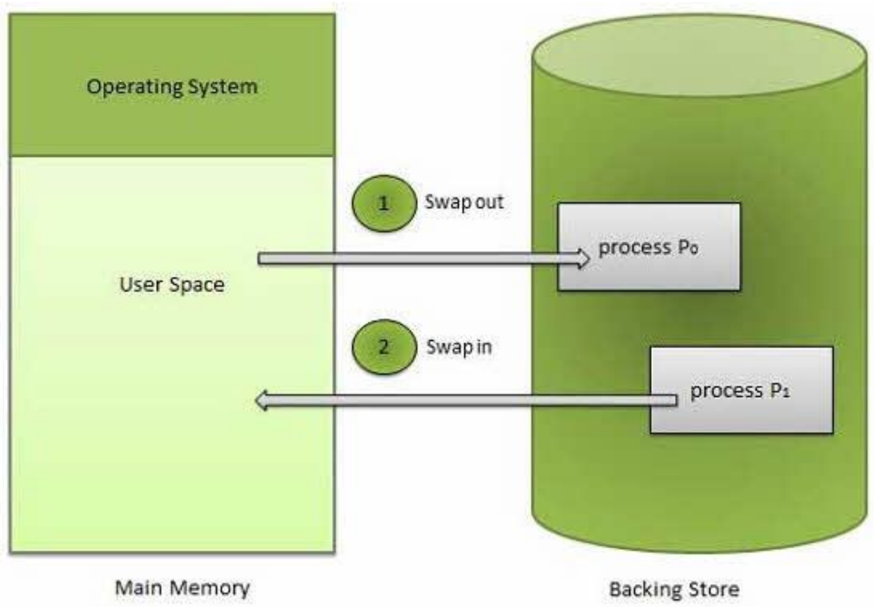
Operating System functions

An operating system performs the following functions:

- Memory management
- Task or process management
- Storage management
- Device or input/output management
- Kernel or scheduling

Memory Management

Memory management is the process of managing a computer memory. Computer memories are of two types: primary and secondary memory. The memory portion for programs and softwares is allocated after releasing the memory space.



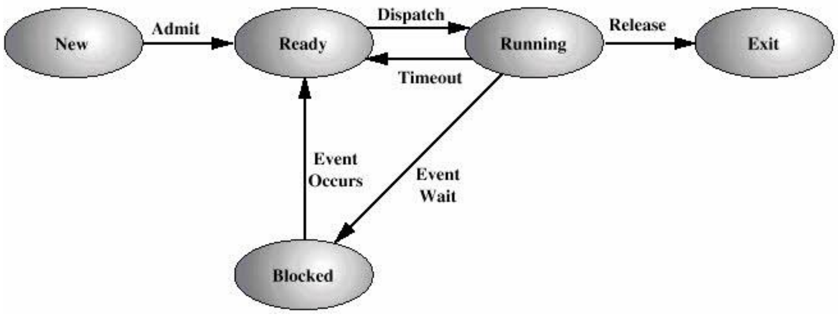
Memory Management

Memory management is important for the operating system involved in multitasking wherein the OS requires switching of memory space from one process to another. Every single program requires some memory space for its execution, which is provided by the memory management unit. A CPU consists of two types of memory modules: virtual memory and physical memory. The virtual memory is a RAM memory, and the physical memory is a hard disk memory. An operating system manages the virtual memory address spaces, and the assignment of real memory is followed by the virtual memory address.

Before executing instructions, the CPU sends the virtual address to the memory management unit. Subsequently, the MMU sends the physical address to the real memory, and then the real memory allocates space for the programs or data.

Task or Process Management

Process management is an instance of a program that is being executed. The process consists of a number of elements, such as identifier, program counter, memory pointer and context data, and so on. The Process is actually an execution of those instructions.



Process Management

There are two types of process methods: single process and multitasking method. The single process method deals with the single application running at a time. The multitasking method allows multiple processes at a time.

Storage Management

Storage management is a function of the operating system that handles memory allocation of the data. The system consists of different types of memory devices, such as primary storage memory (RAM), secondary storage memory, (Hard disk) and cache storage memory.

Scheduling

Instructions and data are placed in the primary storage or cache memory, which is referenced by the running program. However, the data is lost when power supply cut off. The secondary memory is a permanent storage device. The operating system allocates a storage place when new files are created and the request for memory access is scheduled.

Device or Input/output Management

In a computer architecture, the combination of CPU and main memory is the brain of the computer, and it is managed by the input and output resources. Humans interact with the machines by providing information through I/O devices.

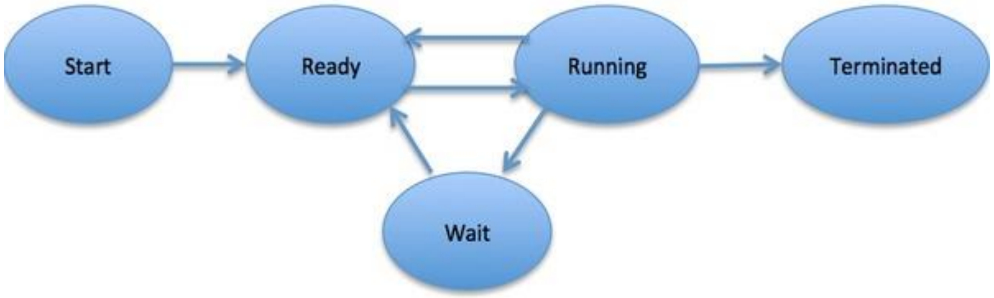
The display, keyboard, printer and mouse are I/O devices. The management of all these devices affects the throughput of a system; therefore, the input and output management of the system is a primary responsibility of the operating system.

What is a Process?

Process is the execution of a program that performs the actions specified in that program. It can be defined as an execution unit where a program runs. The OS helps you to create, schedule, and terminates the processes which is used by CPU. A process created by the main process is called a child process.

Process operations can be easily controlled with the help of PCB(Process Control Block). You can consider it as the brain of the process, which contains all the crucial information related to processing like process id, priority, state, CPU registers, etc.

Process Life Cycle:

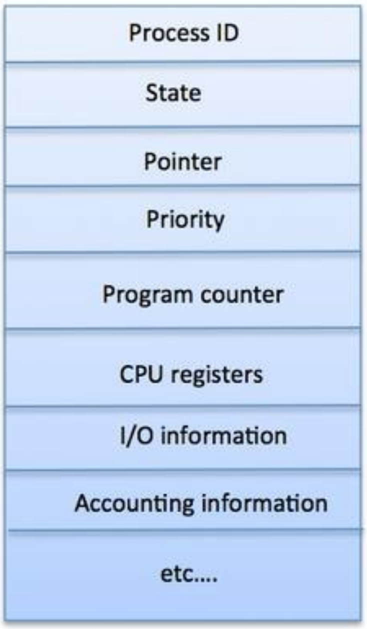


When a process executes, it passes through different states. These stages may differ in different operating systems, and the names of these states are also not standardized.

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.

Process Control Block (PCB)



A Process Control Block is a data structure maintained by the Operating System for every process. The PCB is identified by an integer process ID (PID).

A PCB keeps all the information needed to keep track of a process as listed below in the table –

S.N.	Information & Description
1	**Process State** The current state of the process i.e., whether it is ready, running, waiting, or whatever.
2	**Process privileges** This is required to allow/disallow access to system resources.
3	**Process ID**

	Unique identification for each of the process in the operating system.
4	**Pointer** A pointer to parent process.
5	**Program Counter** Program Counter is a pointer to the address of the next instruction to be executed for this process.
6	**CPU registers** Various CPU registers where process need to be stored for execution for running state.
7	**CPU Scheduling Information** Process priority and other scheduling information which is required to schedule the process.
8	**Memory management information** This includes the information of page table, memory limits, Segment table depending on memory used by the operating system.
9	**Accounting information** This includes the amount of CPU used for process execution, time limits, execution ID etc.
10	**IO status information** This includes a list of I/O devices allocated to the process.

The architecture of a PCB is completely dependent on Operating System and may contain different information in different operating systems. Here is a simplified diagram of a PCB –

The PCB is maintained for a process throughout its lifetime, and is deleted once the process terminates.

Schedulers

Schedulers are special system software which handle process scheduling in various ways. Their main task is to select the jobs to be submitted into the system and to decide which process to run. Schedulers are of three types −

- Long-Term Scheduler
- Short-Term Scheduler
- Medium-Term Scheduler

Long Term Scheduler

It is also called a **job scheduler**. A long-term scheduler determines which programs are admitted to the system for processing. It selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling.

The primary objective of the job scheduler is to provide a balanced mix of jobs, such as I/O bound and processor bound. It also controls the degree of multiprogramming. If the degree of multiprogramming is stable, then the average rate of process creation must be equal to the average departure rate of processes leaving the system.

On some systems, the long-term scheduler may not be available or minimal. Time-sharing operating systems have no long term scheduler. When a process changes the state from new to ready, then there is use of long-term scheduler.

Short Term Scheduler

It is also called as **CPU scheduler**. Its main objective is to increase system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process. CPU scheduler selects a process among the processes that are ready to execute and allocates CPU to one of them.

Short-term schedulers, also known as dispatchers, make the decision of which process to execute next. Short-term schedulers are faster than long-term schedulers.

Medium Term Scheduler

Medium-term scheduling is a part of **swapping**. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium-term scheduler is in-charge of handling the swapped out-processes.

A running process may become suspended if it makes an I/O request. A suspended processes cannot make any progress towards completion. In this condition, to remove the process from memory and make space for other processes, the suspended process is moved to the secondary storage. This process is called **swapping**, and the process is said to be swapped out or rolled out. Swapping may be necessary to improve the process mix.

Comparison among Scheduler

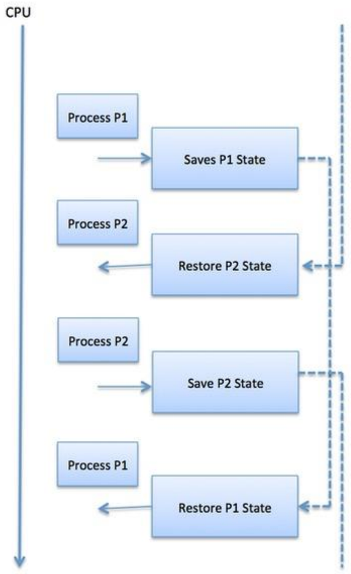
S.N.	Long-Term Scheduler	Short-Term Scheduler	Medium-Term Scheduler
1	It is a job scheduler	It is a CPU scheduler	It is a process swapping scheduler.
2	Speed is lesser than short term scheduler	Speed is fastest among other two	Speed is in between both short and long term scheduler.
3	It controls the degree of	It provides lesser control over degree of	It reduces the degree of

	multiprogramming	multiprogramming	multiprogramming.
4	It is almost absent or minimal in time sharing system	It is also minimal in time sharing system	It is a part of Time sharing systems.
5	It selects processes from pool and loads them into memory for execution	It selects those processes which are ready to execute	It can re-introduce the process into memory and execution can be continued.

Context Switch

A context switch is the mechanism to store and restore the state or context of a CPU in Process Control block so that a process execution can be resumed from the same point at a later time. Using this technique, a context switcher enables multiple processes to share a single CPU. Context switching is an essential part of a multitasking operating system features.

When the scheduler switches the CPU from executing one process to execute another, the state from the current running process is stored into the process control block. After this, the state for the process to run next is loaded from its own PCB and used to set the PC, registers, etc. At that point, the second process can start executing.



Context switches are computationally intensive since register and memory state must be saved and restored. To avoid the amount of context switching time, some hardware systems employ two or more sets of processor registers. When the process is switched, the following information is stored for later use.

- Program Counter
- Scheduling information
- Base and limit register value
- Currently used register
- Changed State
- I/O State information
- Accounting information

Inter process communication

Inter process communication (IPC) means the processes to communicate with each other while they are running. IPC allows processes to synchronize their action without sharing the same address space. Messages provide basic communication capabilities between two processes. Interprocess communication is best provided by a message passing system. IPC is especially useful in a distributed environment where communicating processes may dwell on different computers connected with a network.

Message Passing:

The key idea behind IPC is message passing. That means, one process sends a message and the other process receives it. So the message system should provide at least two operations as follow:

1. send message
2. receive message

For any two processes to communicate with each other a link must be established between them. This link may be unidirectional or bidirectional. Also, the messages can be of fixed size or variable size. The link between two processes can be implemented using direct communication or indirect communication.

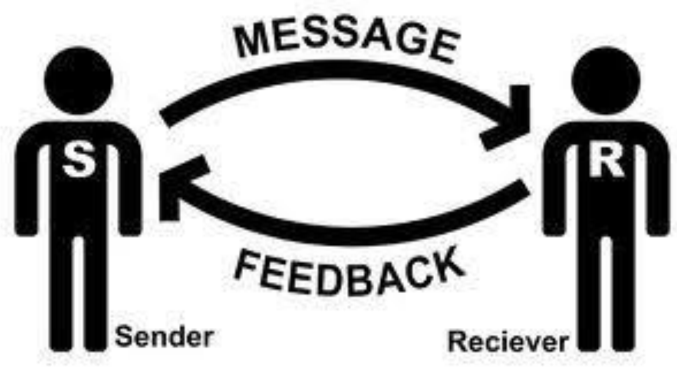
1. Direct Communication:

Here the process that wants to communicate with another process must explicitly state the name of the sender or the receiver. The send and receive primitives, in this case, are defined as follows:

- Send(a, message): send a message to process a
- Receive(b, message): receive a message from process b.

In direct communication,

- A pair of processes will have only one link associated with them.
- A link is automatically established between every pair of processes that want to communicate.
- The link is general, bi-directional.



2. Indirect Communication:

In this case, each pair of processes wants to communicate with each other needs to have a shared mailbox. The sender places the message in the mailbox also known as a port and the receiver removes the data or message from the box. In this case, the send and receive primitives are defined as follows:

- Send(p, message): send a message to mailbox p.
- Receive(q, message): receive a message from mailbox q.

- A link is established between a pair of processes only if those processes have a shared mailbox.
- A link is associated with more than one pair.
- A link may be either unidirectional or bi-directional in nature.

REFERENCES

Text Books:

Abraham Silberschatz Peter B. Galvin, G. Gagne, “Operating System Concepts”, Sixth Edition, Addison Wesley Publishing Co., 2003.

1. www.greeksforgreeks.com

2. www.elprocus.com

3. www.faceprep.com

4. www.tutorialspoint.com