Operating System

Unit-1

BASIC CONCEPTS OF OPERATING SYSTEM

INTRODUCTION

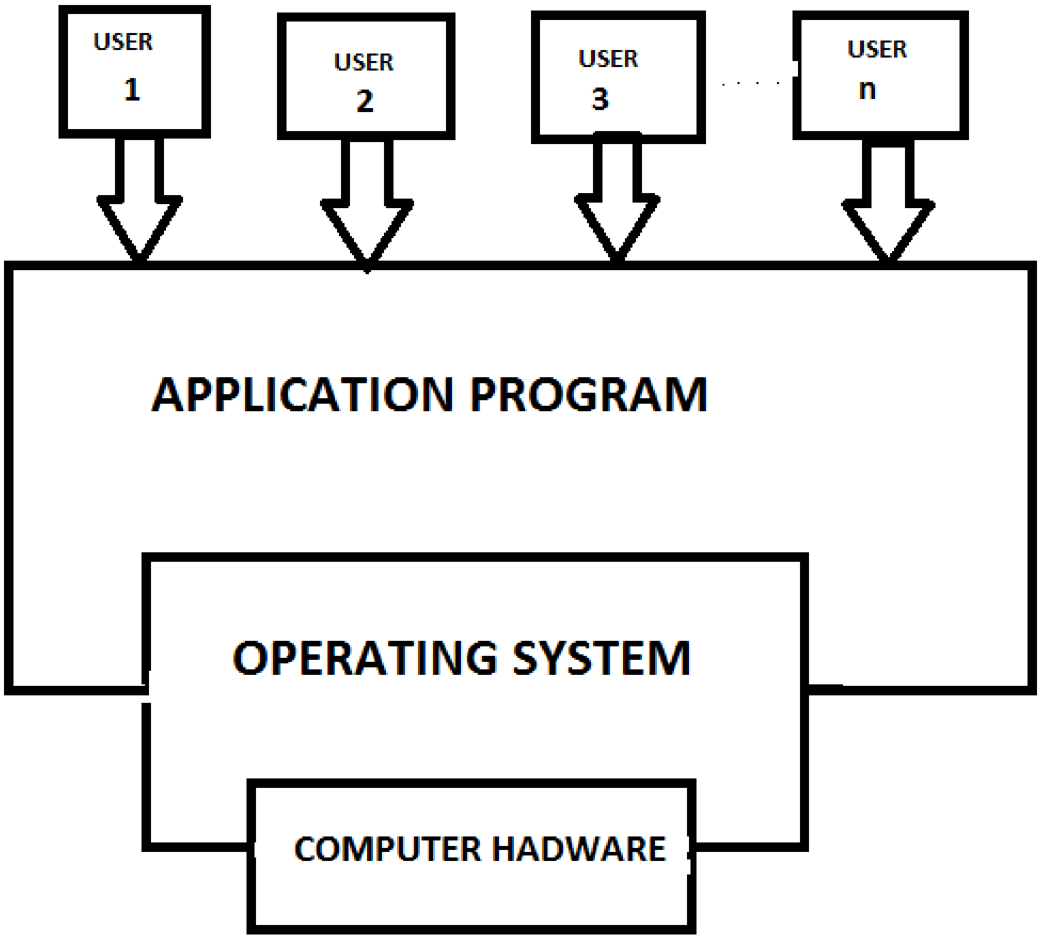
An operating system is a program that acts as a intermediary between a user of a computer and the computer hardware.

COMPONENTS OF A COMPUTER SYSTEM

1. Hardware
2. Operating system
3. Application Programs
4. Users.

- ❖ **Hardware**
  - CPU,Memory,I/O device.
  - Provides the computing resources.
- ❖ **Application Programs**
  - Compiler, Assembler etc.
  - Define the ways to solve the problems of the users.
- ❖ **Users**
  - People, machine.
  - Trying to solve different problems.

- **Operating System**
  - An operating system is a program that acts a intermediary between a user of a computer and the computer hardware.
  - The operating system controls and coordinates the use of the hardware.
  - **Government**
    - An operating system is similar to a government.
  - **Resource Allocator**
    - Operating system acts as the manager of resources and allocates them to specific programs and users.
  - **Control Program**
    - Controls the execution of user programs to prevent errors and improper use of the computer.
  - **Goal**
    - **Convenience**
      - Make computer convenient to use.
    - **Efficiency**
      - To use the computer hardware in efficient manner.



****

<u>**OPERATING SYSTEM SERVICES**</u>

- ❖ An operating system provide an environment for the execution of programs.
- ❖ The operating system provide the following services.
    1. Program execution
    2. I/O operations
    3. File system manipulations

4. Communications
5. Error detection
6. Resource Allocation
7. Accounting
8. Protection

- **Program Execution**
  - ✓ The Operating system loads a program into memory and runs it.
  - ✓ After execution the program must end either normally or abnormally.
- **I/O Operations**
  - ✓ A running program may require I/O. This I/O may involve a file or an I/O device.
  - ✓ For efficiency and protection, users cannot control I/O devices directly. Therefore the operating system must provide some means to do I/O.
- **File system Manipulation**
  - ✓ Operating system performs various file related operations such as create, Delete, Read & Write.
- **Communications**
  - ✓ There are many situations in which one process needs to exchange information with another process.
  - ✓ There are two ways in which communication can occur.

- ✓ The first takes place between processes executing on the same computer.
- ✓ The second takes place between processes executing on different computer systems that are connected by a network.
- ✓ Communication may be implemented into two ways. ie., by shared memory or message passing technique.

➢ **Error Detection**

- ✓ The operating system must be aware of possible errors.
- ✓ Error may occur in the CPU and memory hardware, I/O devices, or in the user program.
- ✓ For each type of Error the OS should take appropriate action to correct the error.

➢ **Resource Allocation**

- ✓ When there are multiple users or multiple jobs running at the same time resources must be allocated to each of them.
- ✓ The different types of resources are managed by the operating system.
- ✓ Some resources may have special allocation code and others may have general code.
- ✓ The various resources may be plotters models and other peripheral devices.

- **Accounting**
  - ✓ OS keeps track of which users use how much and what kind of computer resources.
  - ✓ This record keeping may be used for accounting or usage statistics.
- **Protection**
  - ✓ When several processes execute simultaneously it should not be possible for one process to interfere with others. So, protection and security is important. Such security is given using password by OS.

****

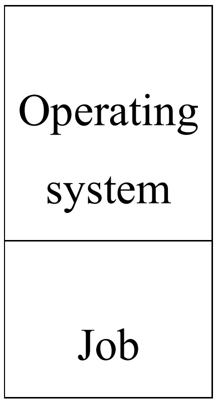
Types of Operating system

1. Simple Batch Systems.
2. Multi programmed Batched Systems.
3. Time-Sharing Systems.
4. Personal-Computer Systems.
5. Parallel Systems.
6. Distributed System.
7. Real-Time System.

1.Simple Batch System

- Early computers were large machines.

- The common input devices were card reader and the output devices were line printers. The user prepared a job which consisted of the program, data and some control information and submitted it to the computers. At later time the output appeared.
- In this early computers, to speed up processing, jobs with similar needs were batched together and were run through the computers as a group.
- A batch operating system reads the jobs and when the job is completed its output is displayed in printer.



❋ **Drawback of Simple batch system**

- ✓ There is lack of interaction between the user and the job that is executing.
- ✓ The delay between job submission and job completion is called turnaround time and it is more.
- ✓ The CPU is often idle due to the speed variation between I/O devices and CPU.

✵ **Spooling**

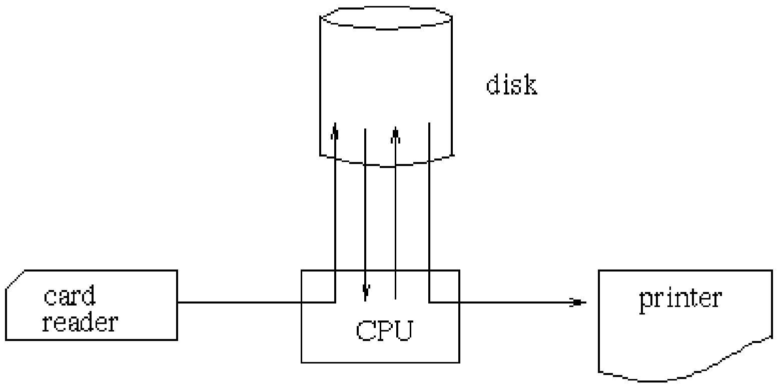
✓ To overcome this problem spooling technique is used.

✓ The expansion of spooling is **Simultaneous Peripheral Operation On-line.**

✓ Spooling uses the disk as a huge buffer, for reading from input devices and for storing output files until the output device are able to accept them.

✓ Advantage

- Spooling is used for processing data at remote sites.
- Spooling can keep both the CPU and the I/O devices working at higher rates.

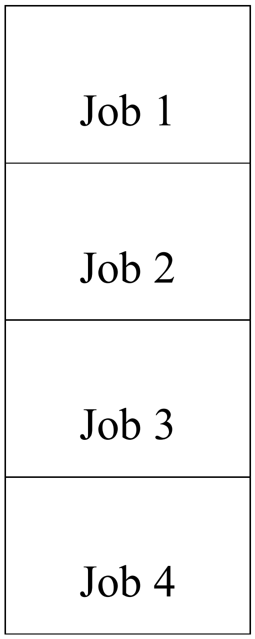


2. **Multi programmed Batched Systems**

In multi batched system execution of many process can be done at the same time by not allowing the CPU to be idle at any time. One important characteristic of job scheduling is multi programming.

❖ **Multi programming**

- Consider there are several jobs in the memory for execution. These jobs are only a subset of the job pool.
- Now the CPU pick up one job for execution and the job is being executed. Eventually the job has to wait in between for the tape to be mounted or for some I/O operation to take place.
- In a non-multi programmed system the CPU will remain idle during the process waiting.



❖ **Advantage**

- ➢ In multi programming System the CPU switches on to the next job and when that job needs to wait, CPU will move to the next job and so on.
- ➢ When the first job has completed its I/O work it will request for the CPU and finishes its work.
- ➢ Thus in this system the CPU is never allowed to be idle.

❖ **Job pool**

- ➢ It is a part of the memory where the jobs that are to be executed are held. The process that are entering the system will be placed in the job pool.

❖ **Job Scheduling**

- ➢ There may be many jobs available to enter into the memory. The process of deciding which job has to be brought into the memory is called job scheduling.

❖ **CPU Scheduling**

- ➢ If several jobs are ready to run at the same time, the system must choose among them. The process of deciding which job has to be executed by the CPU is called CPU Scheduling.

3.**Time-Sharing System**

There are various difficulties with batched system. They are

- ♦ The user cannot interact with the job while executing.
- ♦ The subsequent steps depend on the result or previous ones.
- ♦ The program has to be debugged statically.
- ♦ More turnaround time is required.

❖ **Time-Sharing (or) Multitasking**

- ➢ Time sharing or multitasking is a logical extension of multiprogramming.
- ➢ Multiple jobs are executed by CPU switching between them, but the switches occur so frequently so that the user can interact with each program during execution.

❖ **Interactive System**

- ➢ Interactive system or hands on system provides online communication between the user and the system.
- ➢ The user gives the command and receive immediate response.
- ➢ The user provide the commands to the OS, wait for the response and decides the next command based to the result of previous one.

❖ **On-line File System**

- To access both data and code conveniently online file system is needed.
- A file is a collection of related information. It represent both data and program.
- Files are organized into clusters or directories to make easier to locate and access by the authorized users.

❖ **Batch system Vs Interactive system**

- **Batch system**
  - ✓ It is used for large jobs with little interaction.
  - ✓ The user submits the job and returns later for result. That is the user does not wait for the immediate response.
  - ✓ The response time required is large.
- **Interactive system**
  - ✓ It is composed of small actions in which the result of next command is unpredictable.
  - ✓ The user submit the job and wait for the response.
  - ✓ The response time is short.
- **Single user interactive system**
  - ✓ In this the entire system is handled by a single user. The I/O process time normally depends upon the users speed and so the CPU is often idle.
- **Multi user interactive system**

- ✓ A time shared operating system allows many users to share the computer simultaneously by switching from one user to the next.

❖ **Virtual Memory**

- ➢ In time sharing system we use a technique called virtual memory.
- ➢ Virtual memory is a technique that allows the execution of a job that may not be completely in memory.

4.**Personal Computer Systems**

- ✓ A computer system dedicated to a single user.
- ✓ In personal computer system the earlier I/O devices where changed to keyboard, mouse, display screens and small,fast printers.
- ✓ Personal computers are smaller and less expensive.
- ✓ PC operating systems were neither multiuser nor multitasking.
- ✓ The PC system were developed to maximize user convenience and responsiveness and not for maximizing CPU and peripheral utilization.
- ✓ PC’s were more powerful, faster and more sophisticated.

5.Parallel Systems

System may have more than one processor, sharing the computer bus, memory and peripheral devices. These systems are referred as tightly coupled system.

❖ **Advantage**

- ✓ One advantage is increased throughput. That is by increasing the number of processors we get more work done in a shorter period of time.
- ✓ Multiprocessors can also save money compared to multiple single systems because the processors can share peripherals.
- ✓ Multiprocessor system increase reliability if functions can be distributed properly among several processors then the failure of one processor will not halt the system. The ability to continue providing service with surviving hardware is called graceful degradation.
- ✓ System that are designed for graceful degradation are called fault tolerant.

❖ **Symmetric and Asymmetric multiprocessing**

- ✓ Multiprocessor system may use symmetric multiprocessing model or Asymmetric multiprocessing.
- ✓ In symmetric multiprocessing each processor runs an identical copy of the operating system.

- ✓ In Asymmetric multiprocessing each processor is assigned a specific task. A master processor controls the system by providing instructions to other processors. This scheme defines master slave relationship.

6.**Distributed Systems**

- ❋ In distributed systems large computations can be distributed among several processors.
- ❋ The processes communicate with one another through communication lines.
- ❋ Distributed systems are referred as loosely coupled systems.

Loosely vsTightly coupled systems

- ❖ In loosely coupled systems each processor will have its own local memory and are connected using communication lines.
- ❖ In tightly coupled systems the processors share a common memory.

- The process in a distributed system may vary in size and function. These processors are referred by different names such as sites, nodes, computers and so on.

Advantages of distributed systems

- **Resource sharing**
  - If a number of different sites are connected to one another then a user at one site may be able to use the resources available at another site.
    Eg:
    A user at site “A”may use the printer available at site “B”.
  - Resource sharing in a distributed system provides mechanisms for sharing files,processing information in a data base, printing files at remote site and other operations.

- **Computation Speedup**
  - If a particular computation can be partitioned into a number of sub computations then a distributed system allow to distribute the computation among various sites and are run concurrently.

- If a particular site is overloaded with jobs them some of them may be moved to lightly loaded sites. This moment of job is called load sharing.

- **Reliability**
  - If one site fails in a distributed system the remaining sites can continue operating.That is if the system is composed of number of large processors then the failure of one processor will not affect the rest.
- **Communication**
  - When many sites are connected to one another by a communication network the processors at different site have the opportunity to exchange information. Users may communicate with one another through electronic mail.

7.**Real-Time Systems**

- ✓ Special purpose operating system are called as real-time systems. A real-time system is used when there are rigid time requirements on the operation of a processor.
- ✓ It is also used as a control device in a dedicated application.
- ✓ A real-time operation system has well defined, fixed time constrains and processing must be done with In the constraints.

Types

There are two types of real-time operating systems

1. Hard real-time system.
2. Soft real-time system.

❖ **Hard real-time system**

- A hard real-time system completes the Critical task on time.

❖ **Soft real-time system**

- A soft real-time system completes the task based on priority.
- Soft real-time system are useful in several areas mainly in advanced scientific projects such as undersea exploration.

Classification of operating system

❖ Operating systems can be classified as follows:

❖ **Multi-user:** is the one that concede two or more users to use their programs at the same time. Some of OS permits hundreds or even thousands of users simultaneously.

❖ **Single-User:** just allows one user to use the programs at one time.

❖ **Multiprocessor:** Supports opening the same program more than just in one CPU.

❖ **Multitasking:** Allows multiple programs running at the same time.

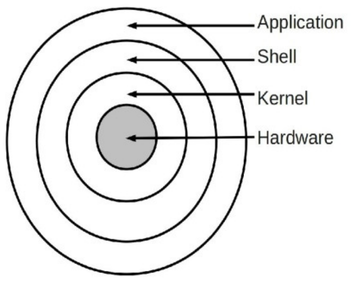
❖ **Single-tasking:** Allows different parts of a single program running at any one time.

❖ **Real time:** Responds to input instantly. Operating systems such as DOS and UNIX, do not work in real time.

****

Architecture and Design of an Operating System

❖ General Architecture of an Operating System



❖ An operating system is a program that acts as an interface between a user of a computer and the computer resources. The purpose of an operating system is to provide an environment in which a user may execute programs.

- **Hardware**
- The hardware consists of the memory, CPU, arithmetic-logic unit, various bulk storage devices, I/O, peripheral devices and other physical devices.
- **Kernel**
- In computing, the kernel is the central component of most computer operating systems; it is a bridge between applications and the actual data processing done at the hardware level. The kernel's responsibilities include managing the system's resources (the communication between hardware and software components). Usually as a basic component of an operating system, a kernel can provide the lowest-level abstraction layer for the resources (especially processors and I/O devices) that application software must control to perform its function. It typically makes these facilities available to application processes through inter-process communication mechanisms and system calls.
- **Shell**
- A shell is a piece of software that provides an interface for users to an operating system which provides access to the services of a kernel. The name shell originates from shells

being an outer layer of interface between the user and the innards of the operating system (the kernel).

- Operating system shells generally fall into one of two categories: command-line and graphical. Command-line shells provide a command-line interface (CLI) to the operating system, while graphical shells provide a graphical user interface (GUI). In either category the primary purpose of the shell is to invoke or "launch" another program; however, shells frequently have additional capabilities such as viewing the contents of directories.

Process concepts

- Process
- Process state
- Process control block

Process

- A program in execution.
- Program is passive entity, the contents of file stored on disk.
- Process is active entity with program counter and set of resources such as processor registers, stack and data section with variables.

Process state

- The state of a process is defined by the current activity of that process.

1. New

- The process is being created.

2. Running

- Instructions are being executed.

3. Waiting

- Process waiting for some event to occur. (I/O completion).

4. Ready

- Process is waiting to be assigned to a processor.

5. Terminated

- Process has finished execution.

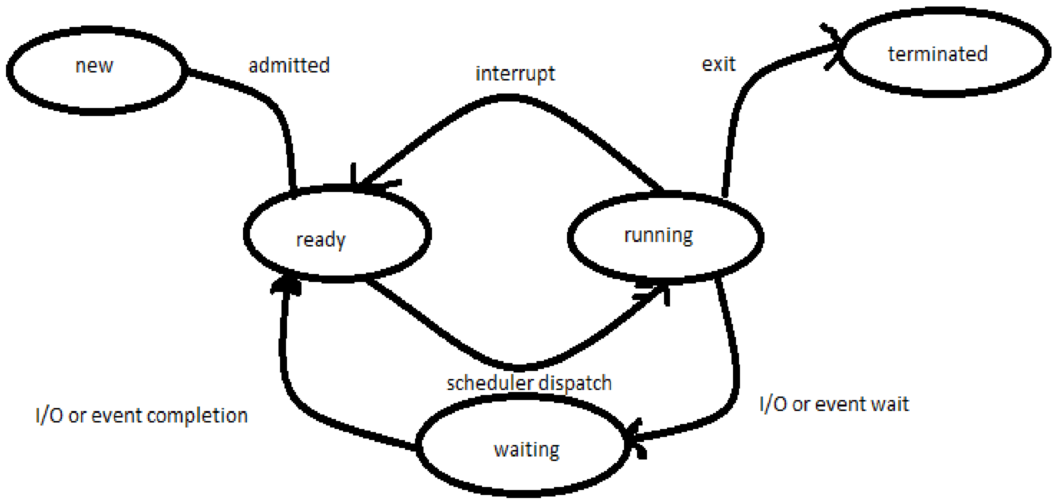


Diagram of Process state

Process control Block

Process control block contains the following information.

Process State

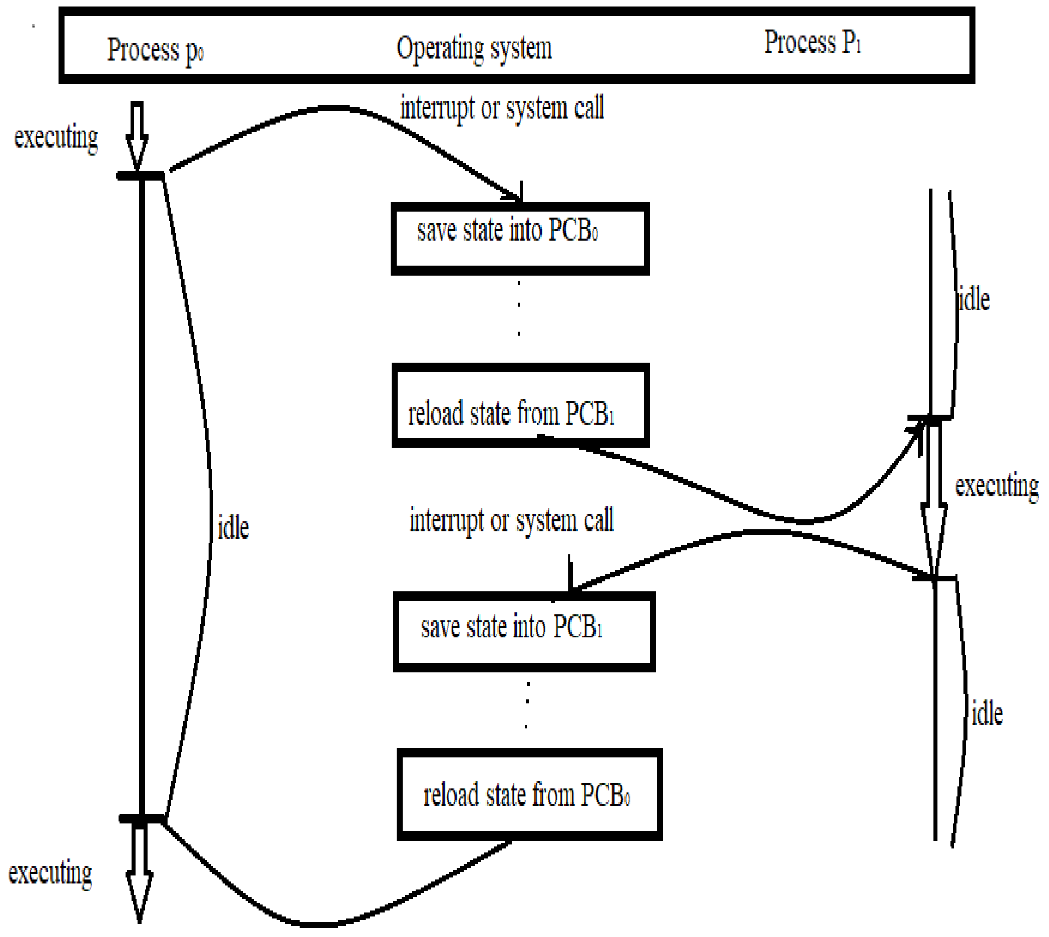
State of the process such as new, Running, etc.

Program Counter

Address of the next instruction to be executed.

CPU Registers

- ➢ Accumulators, general purpose registers and so on.
- ➢ To allow the process to continue after an interrupt.



CPU Scheduling information

- Process priority, pointers to scheduling queues and other scheduling parameters.

Memory Management information

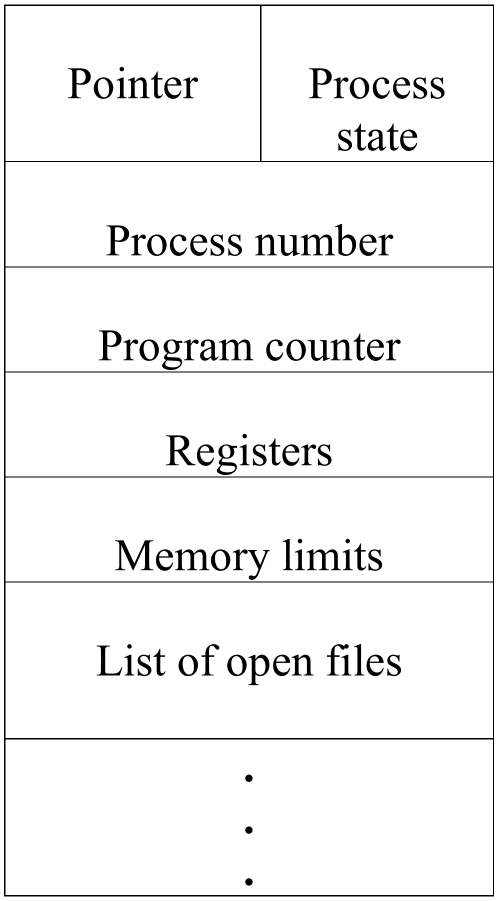
- Base and limit registers value, page tables or segment tables.

Accounting information

- Amount of time CPU used, process numbers time limit etc.

I/O Status information

- List of I/O devices allocated, list of open files..



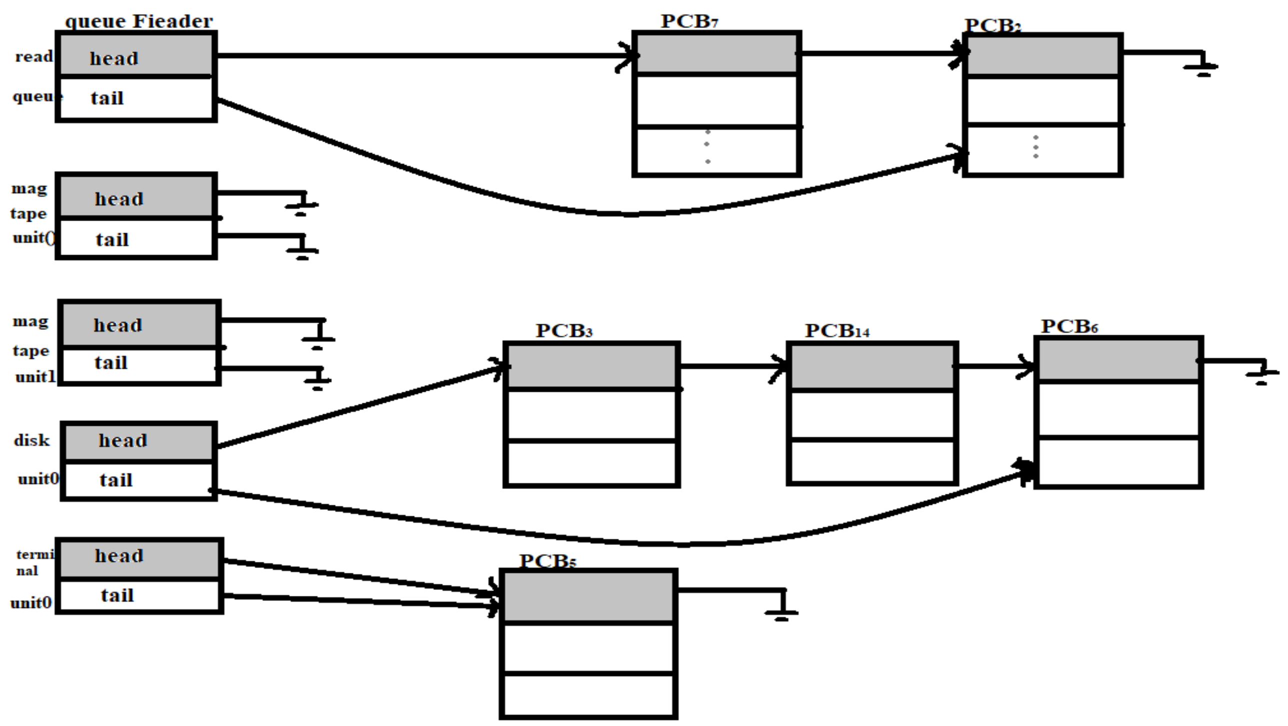
Process controls block

Process Scheduling

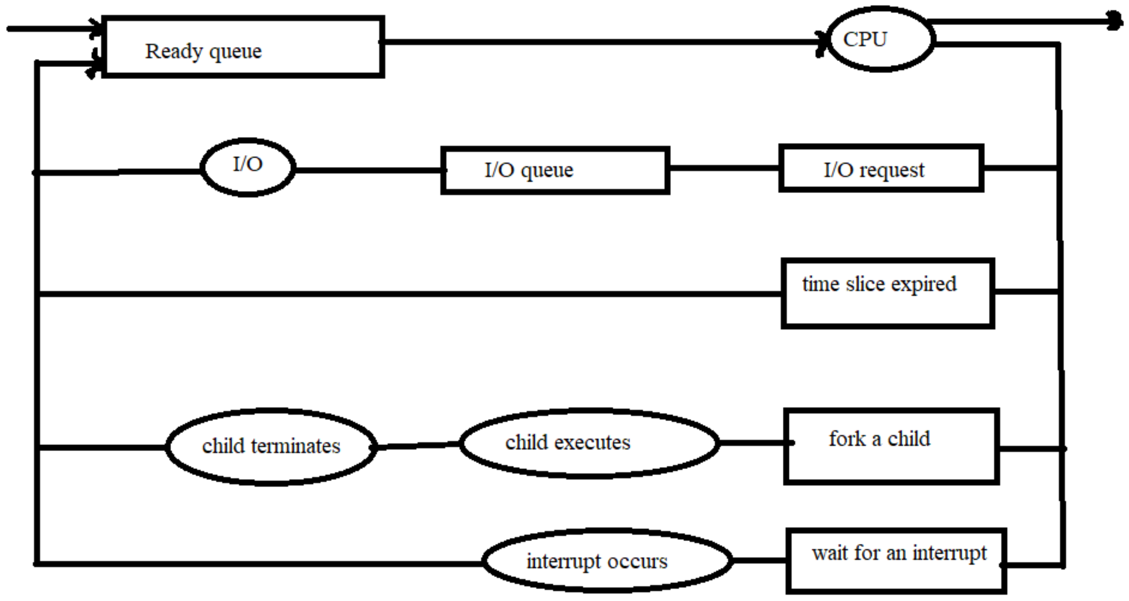
- **Multiprogramming**
  - Process running at all times to maximize CPU utilization.
- **Time sharing**
  - To switch the CPU among process so frequently.
    - Scheduling Queues
    - Schedulers
    - Context Switch

Scheduling Queues

- **Job Queue**
  - It consists of all processes in the system.
- **Ready Queue**
  - The processes that are residing in main memory and are ready and waiting to execute are kept in ready queue.
  - Stored as linked list.
  - Ready queue header will contain pointers to the first and last PCB.
  - Each PCB pointers points to the next process.
- **Device Queue**
  - List of processes waiting for a I/O device is called device queue.



Queue diagram representation of process scheduling.



Queueing-diagram representation of process scheduling

- Once the process is allocated with the CPU and is executing then one of the several events may occurs.

- ➢ Issue an I/O request and then placed in I/O queue.
- ➢ Create a new sub process and wait for termination.
- ➢ Process removed forcibly from the CPU due to interrupt.

Schedulers

- ❖ The selection of process from queue is carried out by the appropriate scheduler.
- ❖ Two types of scheduler.
  - ➢ Short term scheduler or CPU scheduler.
  - ➢ Long term scheduler or job scheduler.

Short term scheduler

- Selects from the processes that are ready to execute and allocates the CPU to one of them.
- Executes more frequently.
- Due to shorter interval it must be very fast.

Long term scheduler

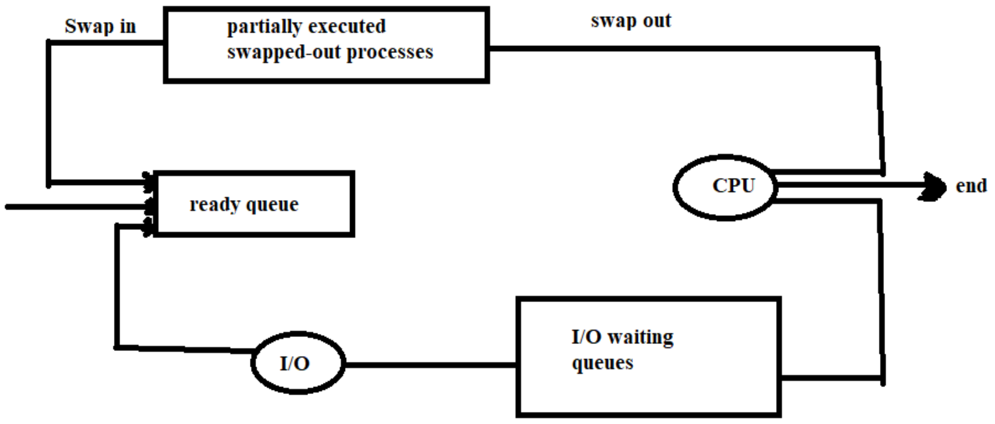
- Selects the processes from disk pool and loads into memory for execution.
- Executes much less frequently.
- Due to longer interval it takes more to decide.

I/O bound or CPU bound

- ✓ Process may be CPU bound or I/O bound.
- ✓ I/O bound process spends more time in doing I/O
- ✓ CPU bound process spends more time in doing computations.
- ✓ Long terms scheduler should a good process mix of I/O bound and CPU bound process the best performance of the system.

Medium term scheduler swapping

- ❋ At some case the process may be removed from memory and after some time the process is reintroduced into memory and its execution can be continued from where it left. This scheme is called swapping.
- ❋ The process is swapped out and swapped in later by medium term scheduler.



Addition of medium-term scheduling to the queueing diagram

Context Switch

- ❋ Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. The is task is known as context switch.
- ❋ Context switch time depends on two ways.
  - Changing the pointer to the current register set.
  - Coping register data to memory.

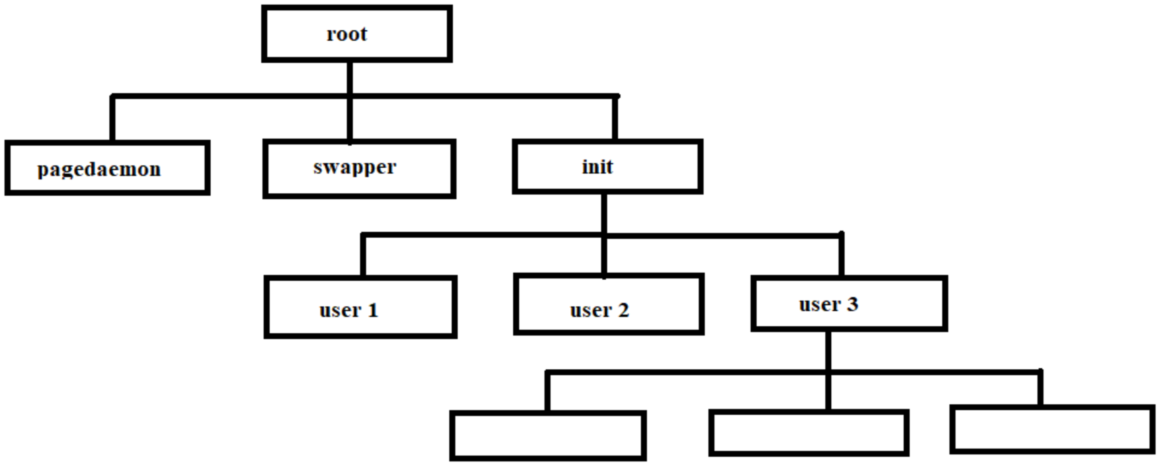
****

Operation on Process

- ♥ Process Creation
- ♥ Process Termination

Process Creation

- ❖ A process may create several new processes (sub process).
- ❖ Creating process is called parent.
- ❖ New process are called children.
- ❖ New process in turn creates other process forming a tree.



- ➢ **Resources**
  - Sub process may obtain resources from 1.OS , 2.Parent Process
  - The parent may partition its resources among its children or share some resources.
  - Logical resources may also be used by used by child process.
- ➢ **Execution**

Execution of the process can be done in two ways

- ♦ Parent continues to execute concurrently with its children.
- ♦ Parent wits until some or all of its children have terminated.

➢ **Address Space(program)**

Address space of the new process may be in two ways.

- ▪ Child process is a duplicate of the parent process.
- ▪ Child process has a program loaded into it.

➢ **UNIX System calls**

- ▪ Fork system call
- ▪ A new process is created by the fork system call and each process is identified by identifier (a unique integer).

➢ **Wait system call**

- ▪ A process may terminate by using the exit system call, and its parent process may wait for that event by using the system call.

❖ **Process Termination**

➢ A process terminates when it finishes executing by asking the OS to delete it by using exit system call.

➢ A process may terminate another process using abort system call (Parent process).

➢ Parent process may terminate the execution of one of its children sue to following reasons.

I. The child exceeds the usage of some of the resource allocated to it.

II. The task assigned to the child is no longer required.

III. The parent is exiting, and the OS does not allow a child to continue if its parent terminates.

- ➢ Cascading termination.
  If the parent terminates then all its children must be terminated.

****

Inter process communication (IPC)

- ❖ The operating system provide cooperating processes to communicate with each other via an inter process-communication (IPC) facility.
- ❖ IPC provide a mechanism to allow processes to communicate and to synchronize their actions. Inter process-communication is best provided by a message system.

Basic Structure

- ❈ An IPC facility provides at least the two operations: send(Message) and receive (message)
- ❈ Messages sent by a process can be of either fixed or variable size. If only fixed-sized messages can be sent, the physical implementation is straightforward.
- ❈ The task of programming is more difficult.
- ❈ Variable size messages require a more complex physical implementation, but the programming task becomes simpler.

- ❋ If processes P and Q want to communicate, they must sent messages to and receive messages from each other; a communication link must exist between them. This link can be implemented in a variety of ways.
- ❋ Some basic implementation questions are these:
  - How are links established?
  - Can a link be associated with more than two processes?
  - How many links can there be between every pair of processes?
  - What is the capacity of a link? That is, does the link have some buffer space? If it does, how much?
  - What is the size of messages? Can the link accommodate variable-sized or only fixed sized messages?
  - Is a link unidirectional or bidirectional?

There are several methods for logically implementing a link and the send/receive operations:

- o Direct or indirect communication
- o Symmetric or asymmetric communication
- o Automatic or explicit buffering
- o Send by copy or send by reference
- o Fixed sized or variable sized messages

Naming

Processes that want to communicate can use either direct communication or indirect communication.

Direct Communication

- ✓ In the direct communication, each process that wants to communicate must explicitly name the recipient or sender or the communication.
- ✓ The send and receive primitives are defined as follows:
  - ▪ Send(P, message)
    - Send a message to process P.
  - ▪ Receive (Q, message)
    - Receive a message from process Q
- ✓ A communication link in this scheme has the following properties:
  - ▪ A link is established automatically between every pair of processes that want to communicate. The processes need to know only each other's identity to communicate.
  - ▪ A link is associated with exactly two processes.
  - ▪ Between each pair of processes, there exists exactly one link.
  - ▪ The link may be unidirectional, but is usually bidirectional.

To allow the producer and consumer processes to run concurrently, we allow the producer to produce one item while the consumer is consuming another item. When the producer finishes generating an item, it sends that item to the consumer. The consumer gets that item via the receive operation. If an item has not been produced yet, the consumer process must wait until an item is produced.

- ✓ The producer process is defined as

```
repeat
……….
```

```
Produce an item in next P
……….
Send (consumer, next P);
until false;
```

The consumer process s defined as

```
repeat
    receive (producer, next C);
    ……………
    consumer the item in next  C
    …………….
Until false;
```

This scheme exhibits a symmetry in addressing; that is, both the sender and the receiver processes have to name each other to communicate. A variant of this scheme employs asymmetry in addressing. Only the sender names the recipient; the recipient is not required to name the sender. In this scheme, the send and receive primitives are defined as follows:

- ✓ Send (P, message)
  - Send a message to process P
- receive (ID, message)
  - receive a message from any process, the variable id is set to the name of the process with which communication take place.

The disadvantage in both of these schemes (Symmetric and asymmetric) is the limited modularity of the resulting process definitions.

Indirect Communication

With indirect communication, the messages are sent to and received from mailboxes (also referred to as ports). A mailbox can be viewed abstractly as an object into which messages can be placed by processes and from which messages can be removed.

- Send (A, messages)
    - ♦ Send a message to mailbox A.
- receive (A, message)
    - ♦ receive a message from mailbox A.

In this scheme, a communication link has the following properties:

- ✓ A link is established between a pair of processes only if they have a shared mailbox.
- ✓ A link may be associated with more two processes.
- ✓ Between each pair of communicating processes, there may be a number of different links. Each link corresponding to one mailbox.
- ✓ A link may be either unidirectional or bidirectional.

The are various ways to designate the owner and users of a particular mailbox. One possibility is to allow a process to declare variables of type mailbox. The process that declares a mailbox is that mailbox's owner. Any other process that knows the name of this mailbox can use this mailbox.

A mailbox that is owned by the operating system has an existence of its own. It is independent, and is not attached to any particular process. The operating system provides a mechanism that allows a process:

- ✓ To create a new mailbox.
- ✓ To send and receive message through the mailbox.
- ✓ To destroy a mailbox.

The process that creates a new mailbox is that mailbox's owner by default.

Exception conditions

- ✓ While performing communication several errors (exception condition) may occur and it must be recovered.
- ✓ Some of the exception conditions are given below.
  1. Process Terminates
  2. Lost Messages
  3. Scrambled messages

Process terminates

- ✓ Either a sender or receiver may terminate before a message is processed.
- ✓ We consider two cases.

i. A receiver process P may wait for a message from process Q that has terminated.
ii. Process P may send a message to a process Q that has terminated.

✓ In both case the system may either terminate P or inform P that Q has terminated.

Lost Messages

✓ A message from process P to Process Q may be lost somewhere in the communication network.

✓ There are three methods to deal this.

1. OS may detect the event and resend the message.
2. The sending process may detect the event and retransmit the message.
3. OS may detect the event and inform the sending process about the message lost.

✓ Common detection method is to use timeouts.

Scrambled Messages

✓ Message may be delivered to its destination but it would be scrambled.

✓ OS will retransmit the message.

✓ Error checking codes are used to detect this type of error.

****