Operating System: UNIT-I

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

- The purpose of an **operating system**is to provide an environment in which a user can execute programs in convenient and efficient manner.
- The **operating system**must ensure connect operation of the computer system.
- The **operating system**provides certain services to programs and to the user to make programming task easier.

1. **Multiprocessing system:** Computing system that employs more than one processer.
2. **Multiprogrmming:**

Ability to store multiple programs in memory at once so that they program in memory at once so that they can be executedconcurrently. (simultaneously)

3. **Multithreading:**

   Technique that incorporates multiple threads of execution within process to perform parallel activities, possibility simultaneously.

   Threads: An independently executables system of program instruction.

Types of operating system :

1. **Embedded operating system:** It is used for handheld of PDA. The **operating system** programs are permanently stored on ROM.

   example of **operating system**window CE.NET, palm**operating system**.
2. **Network operating system:** It is used to control and coordinate computers that are networked or linked together.

   e.g. Netware, window NT server, window XP server.
3. **Stand-alone operating system:** It is also called desktop **operating system** andcontrol a single desktop on network computer.

   e.g. Windows, MAC **operating system**, windows98.

Some another important **operating system:**

1. **Window NT workstation:** Client **operating system** designer to work with window NT server.
2. **Window 98 : It is** Stand alone **operating system**
    1. Easier to use
    2. Faster
    3. True web integration
    4. More entertaining

3. Window 2000 professional: Upgrade to window NT workstation.

4. Window ME: Upgrades to window 98 specific designed for home uses.

5. Window XP: One of microsoft's newest and most powerful **operating system**

6. UNIX: The UNIX **operating system** as originally designed to run minicomputers in network environment.

- **Interfaces:**
    1. Character based interface
    2. Graphical User Interface
- **Multiuser operating system:** Number of user worked under no of categories.

Characteristics of Operating system

1.Efficiency: An Efficient **operating system**achieves high through put and low average thousand time. An efficient **operating system** minimizes the time spent providing the services.

2. Robustness: A Robust **operating system** is fault tolerant and reliable system will not be failed due to isolated applications or hardware errors.

3. Scalability: Scalable **operating system** is able to use resources as they are added. A scalable **operating system** can readily adjust its degree of multi-programming. Scalability is a particularly important and attribute of multi-programing scalability is a particularly important attribute of multi-processor systems. As more processor is added to system, ideally processing capacity should increase in proportion to the number of processes.

4. Extensibility: An extensibility **operating system** will adapt well to new technologies and provide capabilities to extend the **operating system** to perform task beyond its original design.

5. Portability: A portable **operating system** is designed such that it can operate on many hardware configurations. Application portability is also important, because it is costly to develop applications, so that same application should run on a variety of hardware configuration to reduce development cost. The **operating system** is crucial to achieving this kind of portability.

6. Security: A secure **operating system** prevents the users and software from accessing services and resources without authorization.

7. Interactivity: Interactivity **operating system** allows applications to respond quickly to user actions, on event.

8. Usability: These **operating system** generally provide an easy-to-use user interface e.g: Linux, Windows XP, is called usability **operating system** because each supports a large set of applications and provides standard user interface.

Component of Operating System :

The following are component of Operating System;

1. Graphical User Interface(GUI)
2. Kernel
3. Interrupt
4. Memory Management

1.GUI: (Graphical User Interface)

It allows the user to point and click and drag and drop icons to request service from the Operating System. e.g, Microsoft Window XP provides GUI through which users can issue command alternatively, the user can open a command prompt window that accepts type command.

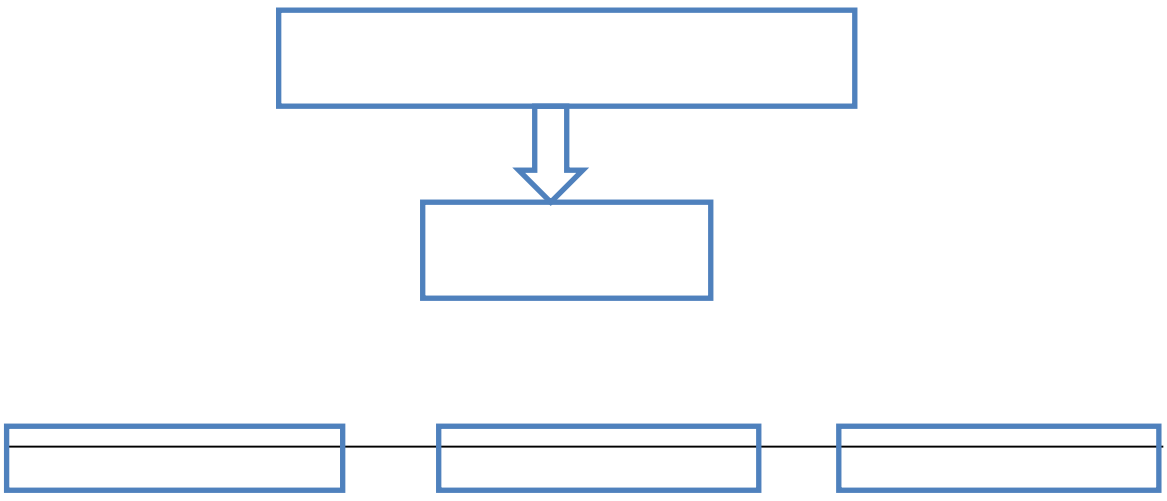
2.Kernel:

The software that contains the core components of **operating system** is referred to as the kernel.

The kernel sets control when :

1. An exceptional situation arises in the system.
2. An O.S policy module explicitly involve kernel mechanism.

The Operating System policy module typically transfers control to the kernel through a software interrupt.



3.Interrupt:

Interrupt enable to respond to signal from hardware. The **operating system**may specify set of instructions called an interrupt handles to be executed in response to each type of interrupt. This allows the **operating system** to gain control of the processes to manage system and resources. A processor may generate an interrupt as a result of executing a process instructions.

4.Memory Management:

The memory manager is an **operating system** component concerned with the system's organization scheme of memory management strategies.The memory space is allocated to process and how to respond to changes in a process's memory. It also interacts with special purpose memory management hardware to improve performance.

Operating system services:

The **operating system** provides certain services to program and to the user. It devices its services in the following three services.

1. **Information management**
2. **Process management**
3. **Memory management**

1. **Information Management:**

Information management the facility to store, retrieve, modify or remove the information on files/directories.

a. These system services manages organization of information into files and directories by allocating memory space to them.

b. It also ensure that connect programs have access to information, have occupied memory space & driving various services.

c. Some of system services(system call) as follows:

1. Create a file
2. Create a directory
3. Open a file foe read / write purpose

4. Explore directory
5. Close a file
6. Read data from file to buffer
7. Write data from file to buffer
8. Move file pointer
9. Create a work
10. Change working directory.

1. **Process Management:** To keep track of all running programs called process.

a. In multiuser **operating system** a number of uses, located at different terminals of network, may execute same or different program at a time.

b. But such a computer system has only one C.P.U. and it can execute only one instruction, belonging to any one of these programs at the same time.

c. The **Process management**of such **operating system** keeps track of all running programs called processes schedule them & dispatch them one after another, while doing so, it gives an impression to each user that it has the full control of C.P.U

d. The **Process management**module of single **operating system**are less compatible multiuser **operating system.**

e. The services of **Process management** (system call) as follows :

1. Running process
2. Ready a process
3. Block process
4. Resume process
5. Terminate process
6. Suspend a process
7. Delay a process
8. Change the priority of process
9. Dispatch a process
10. Fork process

3.Memory Management:

The services in memory management keeps track of all memory locations. They determine memory locations policy and uses various techniques and algorithm for it.

1. When job is to be executed the O.S. loads the job in main memory from disk.
2. But before loading it in memory, it should know that how much of free memory is available and how much of memory should be allocated to the job.
3. For this, the O.S. keeps list of all free memory locations. Before a program is loaded in memory the O.S. consults with this list.
4. When the program is executed completely it removes programs from main memory and again modify the list.
5. For this, O.S. determines memory allocation/deallocation policies and uses various techniques and algorithm for it

6. The system call: **a.** To allocate a chunk of memory to process**. b.** To free chunk of memory from a process.

1. **Ready:** Process state from which a process may be dispatched to the processor.
2. **Block:** The processor is waiting for the completion of some event, such as an I/O completion, and cannot use a processor even if one is available,
3. **Resume:** Remove a process from suspended state.
4. **Terminate:** Thread state that denotes thread has finished executing.
5. **Suspend:** Method of halting a process releasing its resources to other processes.
6. **Delay:** Technique where by a process spin a lock for fixed amount of time before it blocks.
7. **Awake:** Completion of some event it is waiting.
8. **Priority of process:** Important on urgency of a process relative to other processes.
9. **Generate a process:** To create a process.
10. **Dispatch a process: operating system** component that assigns the first process on the ready list to process.
11. **Fork a process:** A process creates child process and duplicate all the essential data structure.
12. **Running:** There is only one process, which is executed by C.P.U at any given time. This process is called as running process.

PROCESS CONCEPT:

The term "process" in the context of O.S was first used by and designs of the multics system in the 1960. Process is nothing but program in execution. Each process has its own address space which typically consists of text region, data region, stack region. The text region stores the code that the process executes. The data region stores variables and dynamically allocated memory that the process uses during execution. The stack region stores instruction and local variables for active procedure calls.

- **PROCESS STATE : (STATUS OF PROCESS)**

The **operating system** must ensure that each process receives a sufficient amount of process time. For any system,these can be only as many truly concurrently executing process as these are processors normally, these are many process than processors in a system. Thus at any

given time,some process can executeand some cannot. During its lifetime, a process moves through series at discrete process status various events can cause a processto change state a process is said to be running (it is the running state) if it is executing on a processor.

A process is said to be ready (i.e. in ready) if it could executeon processor if one were available.A process is said to be blocked (i.e. in the block state) if it is waiting for some event to happen (such an completion event) be use it can proceed. In uniprocessor system only one process may be running at a time,but several may be ready or blocked. The **operating system** maintains a ready list of ready processes and blocks list of blocked process.The ready list is maintained priority orders, so that the next process to executing a processor is the first one in the list (i.e. the process with the highest priority).

Process states and state transitions :

When a user runs a program, processes one created and inserted into ready list. A process mores towards the head of the list as other process complete their turns using a processor. When a process reaches the head of list, and when a processor becomes available, that process is given a processor and is said to make state transition from the ready state to the running state.

The acts of assigning a processor to first process on the ready list is called as dispatching and perform by system is called as dispatcher. Process that are is ready or running states are said to be awake, because they are activity contending for processes time. The **operating system** manages state transition to best serve professes in the system. To prevent any one process from monopolizing the system, either accidently or maliciously, **operating system** set hardware interrupting clock (interval timer) to allow a process to run specific time interval or quantum.

O**perating system** changes the states of previously running process to ready and dispatches the first process on ready list, changing its state from ready to running. It running process initiates an I/O operation before its quantum expires, it must wait for to complete before it can use process again. The running process voluntary set the processor. In this case, the process is said to be block, itself, pending the completion of I/O operation. Processes in the block state are said to be asleep because they cannot execute even it processor become available.

Four possible state transitions are:

Ready to run → Dispatch.

Running to ready → Process's quantum expires

Running to block → Process block

Block to ready → Process wakeup

1. **Awake :** They actively challenging process or time.
2. **Asleep :** They cannot execute even if processes become are set
3. **Wakeup:**Completion of some event it is awaiting.

Operations on process:

O**perating system** provides various processes, each perform some action, that action is called as operation on process.

The following main operations on process are as follows:

1. **Create process:** To create any process that is also called as generate a process.
1. **Destroy process:** When any destroy means that it is completely vanished on deleted from the memory of computer.
1. **Suspend process:**This process is use to handle a method of halting a process and releasing its resource to the other process.
1. **Resume Process:** This process id used to remove process from suspended state.
1. **Change process priority:**

This is used to decide the important on urgency of a process related to other processes. Generally there are two types of priorities.

- **High priority**
- **Low priority**

1. **Block a Process:**

The process is waiting for the completion of some event such as I/O operation and cannot use processes event if it is available is called block process.

1. **Wakeup process:** The complexation of some event which is waiting for to wake up the process in the blocked state.
1. **Dispatch process:**

O**perating system** components that assigned the first process on the ready list to the process is called dispatch process.

1. **Inter process communication:**

This process is used to enable a process. To create a process is called are process and created process is called child process. Each child process is created by exactly one parent process such as creation of hierarchical process structure as follows.

Interprocess communication:-

- **Interprocess communication (IPC) is a set of programming interfaces that allow a programmer to coordinate activities among different program processes that can run concurrently in an operating system. This allows a program to handle many user requests at the same time. The IPC interfaces make this possible.**

"To enable a process to communicate with another process"

1. It is common for process to communicate with one another in multiprogrammed & networked environment.

2. Many operating system provide mechanism for IPC e.g. enable a text editor to send a document to point spooler on web browser to retrieve data from distant server.

3. IPC is also essential for process that must coordinate (i.e. synchronize) activities to achive a common goal.

I. Signals:

1.Signals are software interupts that notify a process that an event has occured.

2.Signals don't support data exchange between processes.

3.A software signal's depend on the operating system and software generated interrupts supported by particular processor.

4.When signal occurs, o.s. first determines which process should receive the signal and how that process will respond to the signal.

5.Processes may catch,ignore or mask asigned.

6.A process catches a signal by specifying a routine that operating system calls when it delivers the signal.

7.Process may also ignorte the signal, process relies on the operating system's default action to handel the signal.

8.A common default action is to about, which causes the process to exit immedietly.

9.A process can also stack a signal by masking it when process masks a signal of specific type (e.g, suspended signal)

process clears the signal masks.

II. Message passing:-

1. Message can be passed in one direction at a time.

2. One process is sender & another is receiver.

3. Message passing may be bidirectional, meaning that process can act as either a sender or receiver while participating in IPC.

4. One model of message passing specifies that processes send and receive message by making system call such as

send (receiveprocess, message); e.g. send(P,message)

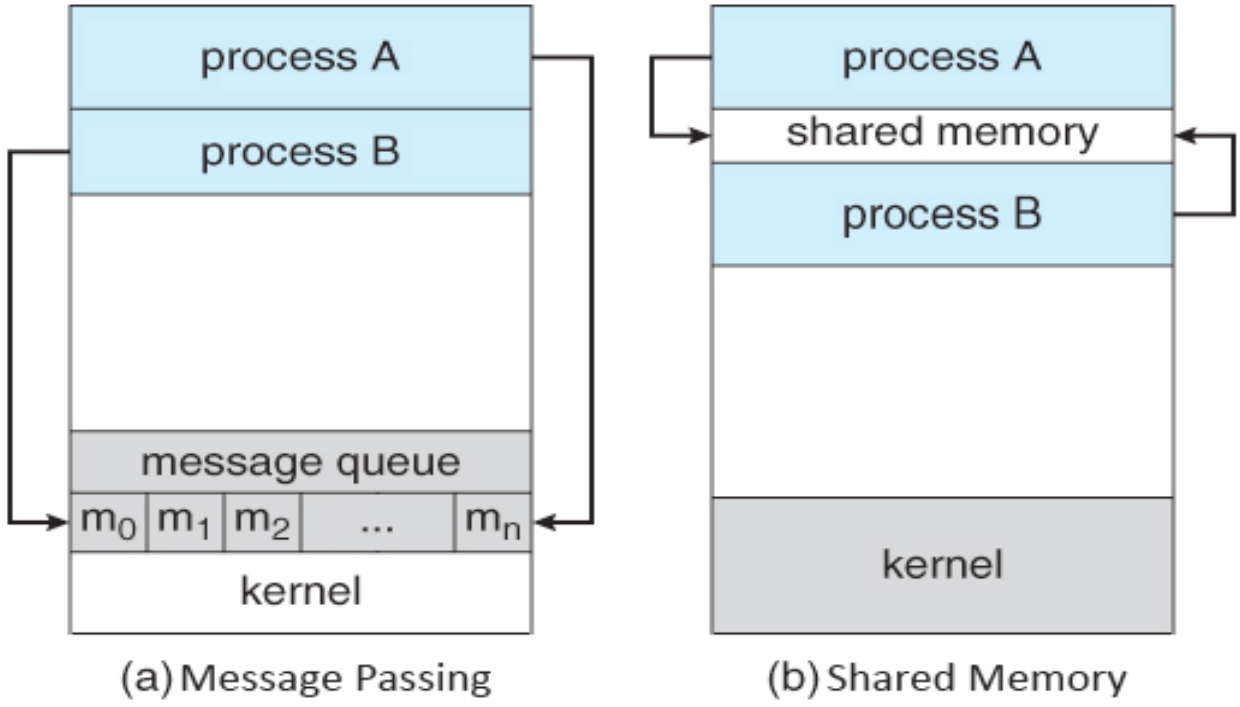
reveive (senderprocess, message);e.g. receive(Q,message).

5.The send and receive calls are normally implemented as system calls accessible from many programing language environments.

6.A blocking send must wait to the receiver to receive the message, requiring that reveiver not the sender when the message is received (acknowledgement) blocking send is example of synchronous communication.

7.A nonblocking send enables sender to continue with other processing even if the reveiver has not yet received (and acknowledged) the message; this requires a message bufferimg mechanish to hold message until receiver receives it. It is asynchronous communication.

8.A popular implementation of message passing is pipe a region of memory protected by o.s. that serves as buffer allowing two or more process to exchange data.



- **The two communications models are contrasted in the figure above:**
- **In the message-passing form, communication takes place by way of messages exchanged among the cooperating processes.**
- **In the shared-memory model, a region of memory which is shared by cooperating processes gets established. Processes can be then able to exchange information by reading and writing all the data to the shared region.**
- **IPC usually utilizes shared memory that requires communicating processes for establishing a region of shared memory. Typically a shared-memory region resides within the address space of any process creating the shared memory segment. Other processes that wish for communicating using this shared-memory segment must connect it to their address space.**

Process scheduling:-

1. To keep track of status of process.

2. It decide which process gets a processor when and for how long.

3. Process scheduler is traffic controller.

4.Allocation & deallocation of processor to process, boost process. Two objective of process scheduling:

A} The Objective of multiprogramming is to have some process running out all times to maximize CPU utilization.

B} The Objective time sharing is to switch the CPU among processes so frequently that users can interact with operating system.

(Job scheduling)	(Process scheduling)
1. Job is collection of task. E.g. Load, Compile, Execute.	1. process is program in execution. e.g. Ready, Block, running.
2. It is constructed with management of job.	2. It is the management of processor to processes.
3. Job schedules is "super manager".	3. Process scheduler is the traffic controller.
4. Job schedule select job to sum & create corresponding process.	4. process scheduler create a process.
5. keeping track of status of all jobs.	5.keeping the stack of the status of the process.
6. Choose policy By which job will "enter the system" (blocked to ready).	6. Decide which process get a processor.
7. Allocate the resources for scheduled job by use of memory device.	7. Allocation of processor to process.
8. Deallocate resources when job is done.	8. Deallocation of processor from process.
9. Job scheduled is like the coordination of contest.	9.It decides which participant will win the contest.
10. It supports dynamic process.	10.Itsupports static process.

➔ **Select a process from ready Queue for execution.**

Evolution sequences:

1) **CPU utilization.**
2) **System throughout.**
3) **Waiting time.**
4) **Response time.**

<u>Scheduling queue:-</u>

1. **Process enter the system they are put into job queue.**
2. **Ready & Waiting process to execute event or a list called ready queue.**
3. **The list of processes waiting for particular I/O device is called device survey (each device its own queue).**

A new process is initially put in the ready queue. It waits there until it is selected for execution on dispatched, once process is also called the CPU & is executing one of several events could occur.

- **The process could issue an I/O request and then be placed in an I/O queue.**
- **The process is could is create new sub process and wait for the sub process's termination.**
- **The process could be removed forcibly from the CPU as a result of an interrupt, and be put back in the ready queue.**

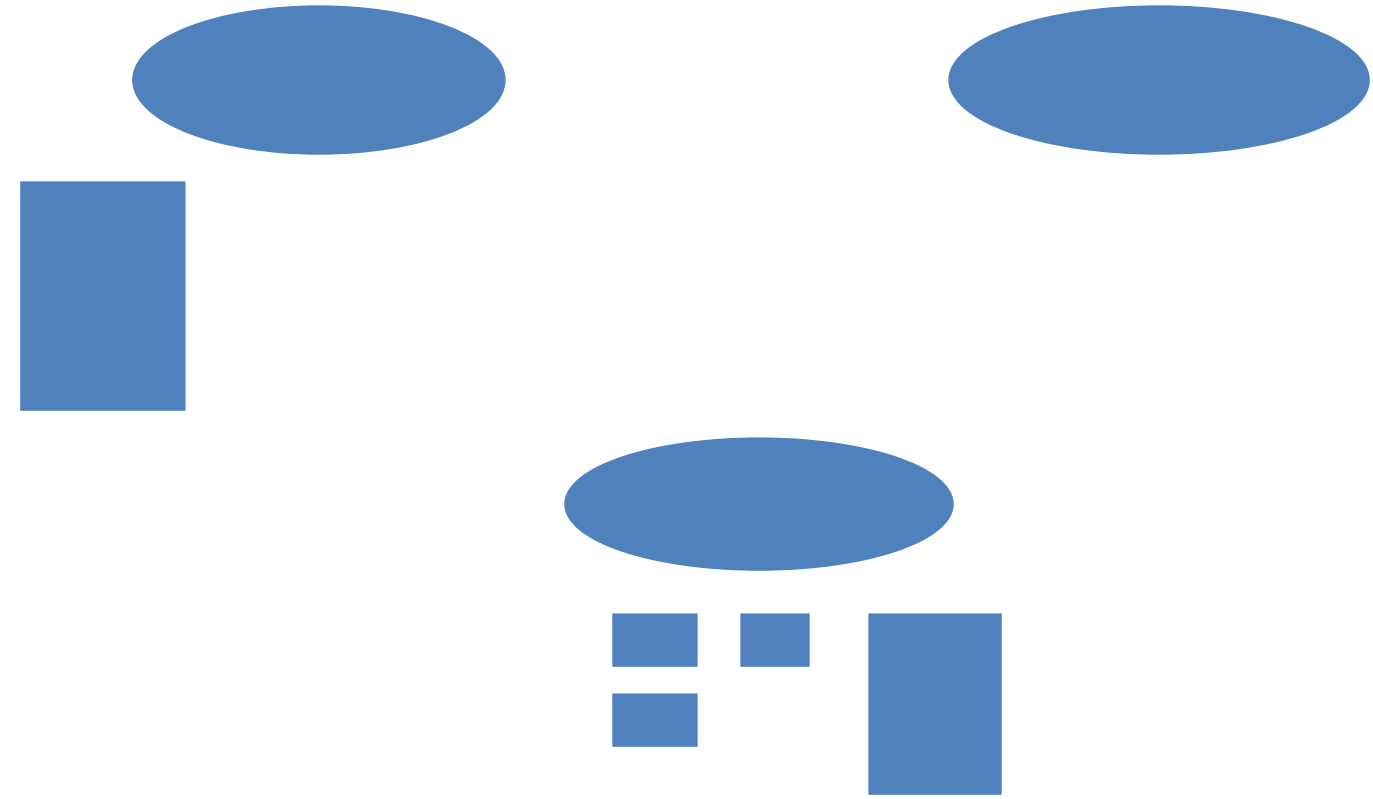


Fig. Process Synchronization