# IMP QUESTION of B OS (4330703)

## 1 . what is an os .?

An operating system (OS) handles your computer needs by finding resources, applying hardware management and providing necessary services. Operating systems are essential for computers to be able to do everything they need to do.

**How do operating systems work?**

An operating system communicates with the various parts of your computer. It sends information to and from your computer hardware and the desired application or action in order to carry out the tasks you request. The following pieces of equipment are in contact with your operating system:

Keyboard
Monitor
Printer
Mouse
Disk drive

## 2. Explain services of

**OS.**User Interface
Program Execution
File system manipulation
Input / Output Operations
Communication
Resource
AllocationError
Detection
Accounting
Security and protection

## 3. Explain the need of Operating System.

OS as a platform for Application programs: Operating system provides a platform, on top of which, other programs, called application programs can run. These application programs help the users to perform a specific task easily. It acts as an interface between the computer and the user. It is designed in such a manner that it operates, controls and executes various applications on the computer.

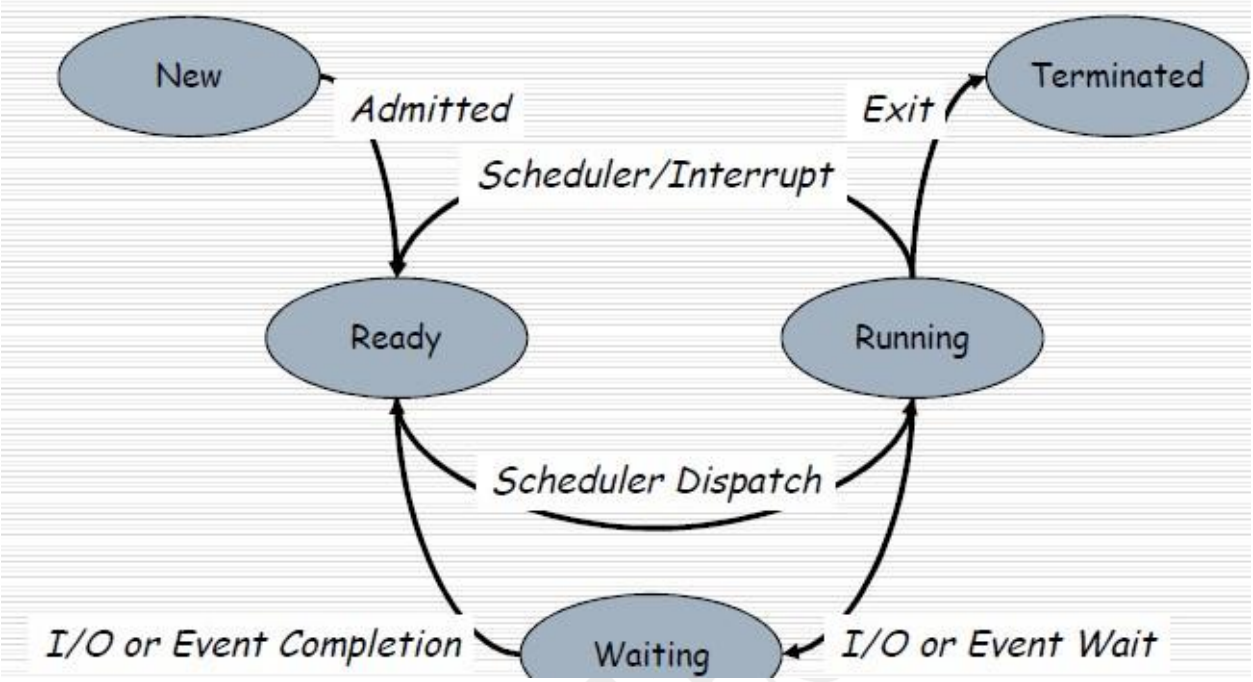
Managing Input-Output unit: Operating System also allows the computer to manage its own resources such as memory, monitor, keyboard, printer etc. Management of these resources is required for an effective utilization.The operating system controls the various system input-output resources and allocates them to the users or programs as per their requirement.

Consistent user interface: Operating System provides the user an easy-to-work user interface,so the user doesn't have to learn a different UI every time and can focus on the content and be productive as quickly as possible. Operating System provides templates, UI components to make the working of a computer, really easy for the user.

Multitasking: Operating System manages memory and allow multiple programs to run in their own space and even communicate with each other through shared memory. Multitasking gives users a good experience as they can perform several tasks on a computer at a time.

## 4. Explain process life cycle

| | |
|---|---|
| 1 | Start<br>This is the initial state when a process is first started/created. |
| 2 | Ready<br>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<br>Once the process has been assigned to a processor by the OS scheduler, the process state is setto running and the processor executes its instructions. |
| 4 | Waiting<br>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<br>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. |




## 5. Explain necessary conditions for deadlocks to occur.

A deadlock situation can arise if and only if the following four conditions hold simultaneously in a system-

**Mutual Exclusion**: At least one resource is held in a non-sharable mode that is only one process at a time can use the resource. If another process requests that resource, the requesting process must be delayed until the resource has been released.

**Hold and Wait**:There must exist a process that is holding at least one resource and is waiting to acquire additional resources that are currently being held by other processes.

**No Preemption**: Resouces cannot be preempted; that is, a resource can only be released voluntarily by the process holding it, after the process has completed its task.

**Circular Wait**: There must exist a set $\{p_0, p_1, . p_n\}$ of waiting processes such that $p_0$ is waiting for a resource which is held by $p_1$, $p_1$ is waiting for a resource which is held by $p_2$,..., $p_{n-1}$ is waiting for a resource which is held by $p_n$ and $p_n$ is waiting for a resource which is held by $p_0$.

## 6. Explain deadlocks and solution to remove deadlocks.

**Eliminate Mutual Exclusion**

It is not possible to dis-satisfy the mutual exclusion because some resources, such as the tapdrive and printer, are inherently non-shareable.

**Eliminate Hold and wait**

Allocate all required resources to the process before the start of its execution, this way hold and wait condition is eliminated but it will lead to low device utilization. for example, if a process requires printer at a later time and we have allocated printer before the start of its execution printer will remain blocked till it has completed its execution.

The process will make a new request for resources after releasing the current set of resources. This solution may lead to starvation.

**Eliminate No Preemption**

Preempt resources from the process when resources required by other high priority processes.

**Eliminate Circular Wait**

Each resource will be assigned with a numerical number. A process can request the resources only in increasing order of numbering.

For Example, if P1 process is allocated R5 resources, now next time if P1 ask for R4, R3 lesser than R5 such request will not be granted, only request for resources more than R5 will be granted.

## 7. Explain FCFS (first come first served) algorithem.

- Jobs are executed on first come, first serve basis.
- Easy to understand and implement.
- Poor in performance as average wait time is high.

| Process | Arrival Time | Execute Time | Service Time |
|---|---|---|---|
| P0 | 0 | 5 | 0 |
| P1 | 1 | 3 | 5 |
| P2 | 2 | 8 | 8 |
| P3 | 3 | 6 | 16 |


Wait time of each process is following

| Process | Wait Time : Service Time - Arrival Time |
|---|---|
| P0 | 0 - 0 = 0 |
| P1 | 5 - 1 = 4 |
| P2 | 8 - 2 = 6 |
| P3 | 16 - 3 = 13 |

Average Wait Time: (0+4+6+13) / 4 = 5.55

## 8. Explain IPC.

- In multiprogramming OS more than one process may be running simultaneously. Such processes can communicate with each other, such type of communication is called Inter process communication.
- IPC is useful in a distributed environment where the communication process may reside on different machine connected with a network.
- Example of IPC:
  - A shell pipeline in UNIX : ls | wc –l
  - Printing a paper in network
  - Chat or mail server

## 9. Explain Race condition / Racing problem.

It is situation where two or more process are reading / writing some shared data andfinal result depends on relative order of their execution ,is called race condition.

- A=1000,At the end of two process A should be 1100.but if P0 and P1 permitted to execute in any arbitrary fashion then output will be not same.

| Po | P1 |
|---|---|
| Read (A); | Read(A) |
| A: A-100 | A :A +200; |
| Write (A) | Write (A) |

**Possibility : 1**

Here, Answer will be 1200

| Po | P1 |
|---|---|
| Read (A); | |
| | Read(A) |
| A: A-100 | |
| Write (A) | |
| | A :A +200; |
| | Write (A) |

**Possibility : 2**

Here, Answer will be 900

| Po | P1 |
|---|---|
| Read (A); | |
| | Read(A) |
| | A :A +200; |
| | Write (A) |
| A : A-100 | |
| Write (A) | |

on relative execution order of P0 and P1,such situation is called race condition.

- This implies that concurrent process are racing with each other to access a shared resource in arbitrary order and procedure wrong final results ,so race condition must be avoided.

## 10. Difference between Logical Address & Physical Address

| No. | Logical Address | Physical Address |
|---|---|---|
| 1 | The process address space can be considered as a sequential list of bytes. Each byte has an address that is used to locate it, these addresses are called logical address. | The entire physical memory can be considered as a sequential list of bytes. Each byte has an address that is used to locate it; these addresses are called physical address. |
| 2 | It is generated by CPU, means CPU determines location of each instruction & data in process address space. | It is generated by Main Memory. |
| 3 | Logical Address Space is a set of all logical address that can be referenced by a process. | Physical Address Space is a set of physical address occupied by a process in main memory during its execution. |

## 11. Explain Multiprogramming with fixed (static) partition.

- This method allows multiple processes to execute simultaneously.
- Memory is shared among Operating System and various simultaneously running processes.
- Multiprogramming increases the CPU utilization, CPU can be kept busy almost all timeby keeping more than one processes simultaneously in memory.
- Memory is divided into fixed partition size can be equal or unequal for different partition. Generally unequal partitions are used for better utilization of memory.
- Each partition is accommodating exactly one process.
- Whenever program needs to be loaded in memory, a free partition big enough to hold program is found and allocated.
- If there is no free partition available of required size, which process needs to wait such process will be put in a queue.
- There are two possible ways to implement method with a queue.
  1. Using multiple input queues. – Fig (a)
  2. Using single input queue. – Fig (b)

Multiple input queue

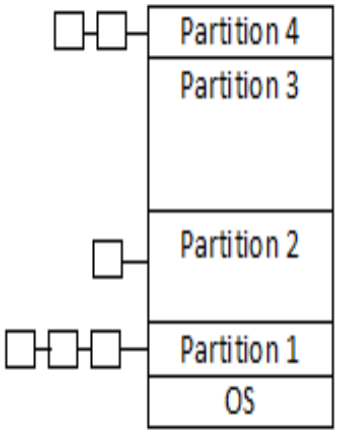



Fig (a)

single input queue

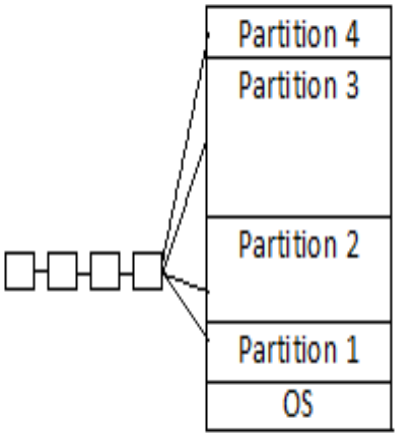
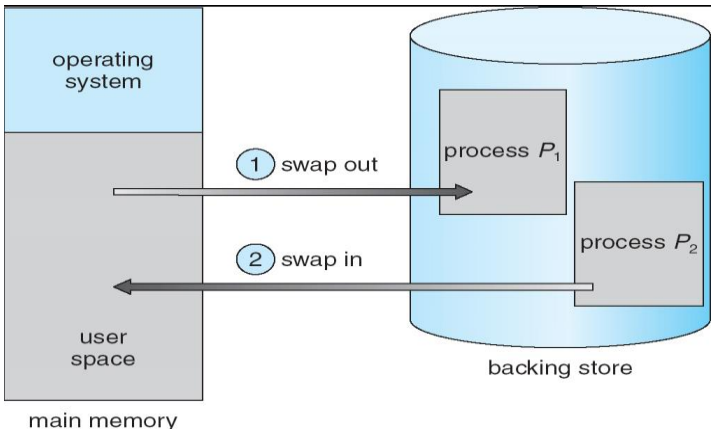



Fig (b)

## 12. Explain swapping.

- *Swapping* is a technique in which processes are moved between main memory and disk.
- Swapping uses some of the portion of secondary storage (disk) as a backing store; this area is called *swap area.*
- Operation of moving process from main memory to swap area is *swap-out.*
- Operation of moving processes from swap area to main memory is *swap-in.*
- When a process is brought back to memory, it may be loaded at some different location rather than its original one. So It needs memory relocation, There is problem of external fragmentation.

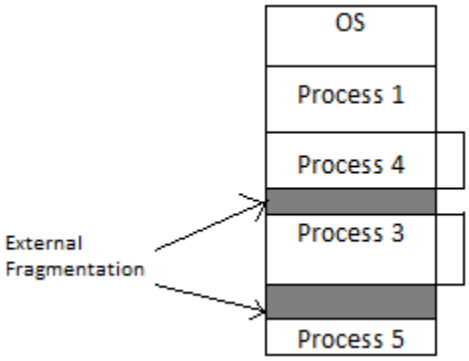


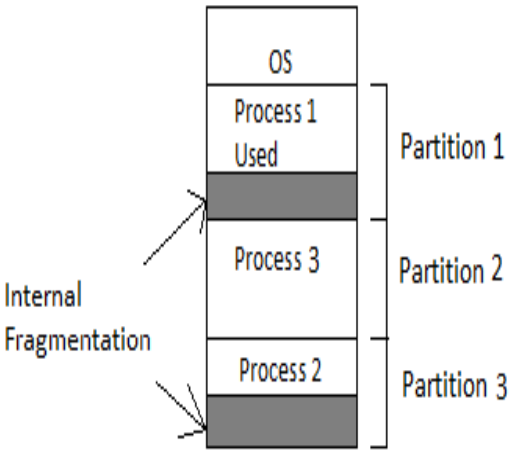
## 13. Explain memory relocation.

- A process can loaded in any partition in main memory.
- Address in Logical Address Space and Physical Address Space is not a same here.
- Logical Address Space specifies the location of instructions and data within process address space.
- Physical Address Space specifies actual location in main memory.
- Logical Address is required to actually fetch instruction. So whenever there is a reference to any Logical Address it should be converted to physical address this problem is called memory relocation.
- EX: suppose process is loaded at location 1000 in main memory & there is need to fetch instruction located at location 5 in Logical Address Space.
- So Logical Address (5) should be converted to actual Physical Address (1000 + 5) = 1005.

## 14. Explain Fragmentation.

- Memory is allocated when process enters in the system and released when it terminates.
- We can't utilize (100%) full memory due to some problems (fragmentation).
- Fragmentation refers to unused memory that cannot be allocated to any process. Means there is free memory available but it can't be used.
- Two types of fragmentations:
    - External Fragmentation
    - Internal Fragmentation






## 15. Explain file operation.

- Create :-A new file can be created by a system call embedded in a program or by an OS command issued by user.
- Delete:- when a file is no longer needed it has to be deleted to free up disk space.
- Open :- before using a file it must be opened.
- Close:- when use of file is finished it should be closed to free up main memory space
- Read:- Data are read the file. The system maintain a read pointer to specify the location in a file from where to read the data content.
- Write:- Data are written to the file. The system maintain a write pointer to specify the location in a file from where to write the data content
- Append:- It is restricted form of a write pointer. here data are only added to the end of the file.
- Seek:- For random access files, a location is needed to specify from where to start read/write operations.
- Get Attributes :- It is used to retrieve file attributes.
- Set Attributes :- It is used to write file attributes.
- Rename :- It is used to change the name of an existing file.

## 16. Explain Linux Architecture.

- It is also known as the layered structure of Linux.
- Linux is a UNIX-like OS, its architecture resembles to that of UNIX.
- 1. Hardware:
- Bottom layer is hard ware.
- It consist of physical devices such as CPU ,memory ,disks ,monitors ,printers etc.

- These devices provides various services.