SIES Graduate School of Technology Subject : OS

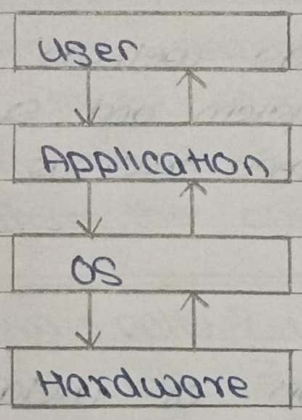
Roll No : 224A1134

Assignment No. A1 Date : 18 / 02 / 2025

ASSIGNMENT No 1

Q1 Define Operating System. Brief Functions of OS.

- Operating System refers to interface between computer & user.
- It is responsible for the management & co-ordination of activities & the sharing of the resources of the computer.
- Application program requires OS to function.
- It keeps track of files and folder on the disk.
- It also manages CPU time.
- The main purpose of an operating system is to provide an environment in which we can execute programs.



- Functions of Operating System

1. ~~Progra~~ Process Management

- The function of the Operating System in this regards is to keep track of all competing processes, schedule them, dispatch them one after the other, but while doing so, try to give each user an impression that he has the

full control of the CPU.

- A process needs certain resources - including CPU time, memory, files and I/O devices - to accomplish it's task.
- These resources are either given to the process when it is created, or allocated to it while it is running.
- In addition to the various physical & logical resources that a process obtains when it is created.

2. Main Memory Management

- Memory management are directed to keep track of memory and allocating / deallocating it to various process
- The Operating System keeps a list of free memory locations.
- It allocates primary as well as secondary memory to the user and system and system process.
- Once the block becomes free, OS allocates it again to the processes

3. File Management

- A file is a collection of related information defined by it's user.
- OS performs following activities :-

a. Creation & Deletion of files and directories
b. manipulating & updating files and directories
c. mapping files on to secondary storage
d. Backing up files on storage media.

4. I/O management
- One of the purposes of an OS is to hide the peculiarities of specific hardware devices from the user.
- The I/O subsystem consist of
  a. A memory management component that includes buffering, caching, & spooling.
  b. A general device-driver interface
  c. Drivers for specific hardware devices.

5. Security
- If a computer system has multiple users and allows the concurrent execution of multiple processes, then the various processes must be protected from one another's activities
- To have protection mechanism, we must ensure that files, memory segments, CPU and other resources are to be operated on by only those processes that have gained proper authorization from the operating system.

Q2 What is system calls? Explain different types of system calls with eg.

→ - System calls provide an interface between the process of an operating system
- It is typically written in a high-level language (C or C++)
- System calls allow user-level processes to request some services from an operating system which process itself is not allowed to do.
- There are different system calls made by an operating system :-

1. Process control
- mainly used for process.
- end, abort, load, execute, create process, terminate --
2. File management
- Used for manipulation of files
- create file, delete file, open, close, read, write, reposition
- get file attributes, set file attributes.
3. Device Management
- Used for managing the devices
- Request devices, release device
- Read, write, reposition.
- get device attributes, set device attributes
- Logically attach or detach devices
4. Information maintenance
- to do information maintenance
- get time or date, set time or date
- get system data, set system data
- get file, process or device attributes, set file, process or device attributes.
5. Communications
- used for communicating among calls.
- create, delete communication connection
- send, receive messages
- transfer status information.
- Attach or detach remote devices.

Q3 Discuss how the following pairs of scheduling criteria conflicts in certain settings.

a) CPU utilization and response time
- CPU utilization aims to keep the CPU as busy as possible, maximizing the time spent executing processes rather than idling.
- This is often achieved by favoring long-running processes and minimizing context switches
- Response time, on the other hand, is the time from when a request is submitted until the first response is produced.
- This is crucial for interactive system, where users expect quick feedback.

Conflicts
- High CPU utilization may lead to long response times, especially if the scheduler prioritizes CPU-bound processes over short interactive tasks.
- for eg, in FCFS scheduling, a long CPU-bound job can delay short interactive tasks, leading to poor response times
- Round Robin (RR) improves response time by frequently switching tasks, but excessive context switching reduces CPU utilization.

b) Average Turn around Time and maximum waiting time
- Turn around time is the total time a process takes from arrival to completion, including waiting and execution time.
- maximum waiting time refers to the longest time

any process spends in the ready queue before execution.

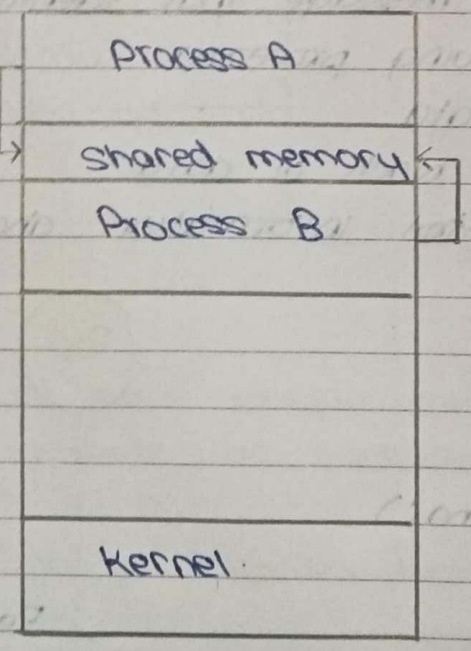
- Conflict
- Scheduling algorithms that minimize average turnaround time may increase the maximum waiting time for some processes.
- Shortest Job first (SJF) reduces average tat by prioritizing short jobs, but long processes may suffer starvation.
- RR ensures that no process waits too long but may increase overall tat due to frequent context switches

Q4 Explain the term inter-process communication.

→ - Inter-Process communications or IPC, is the mechanism whereby one process can communicate, that is exchange data, with another process.
- While these techniques are very useful for communicating with other programs, they do not provide the fine grained control that's sometimes needed for large scale application.
- In these applications it is quite common for several processes to be used, each performing a dedicated task and other processes requesting services from them.
- Various techniques can be used to implement the Inter Process Communication. There are two model of Inter-Process communication that are commonly used,
- They are:-

P.T.O →

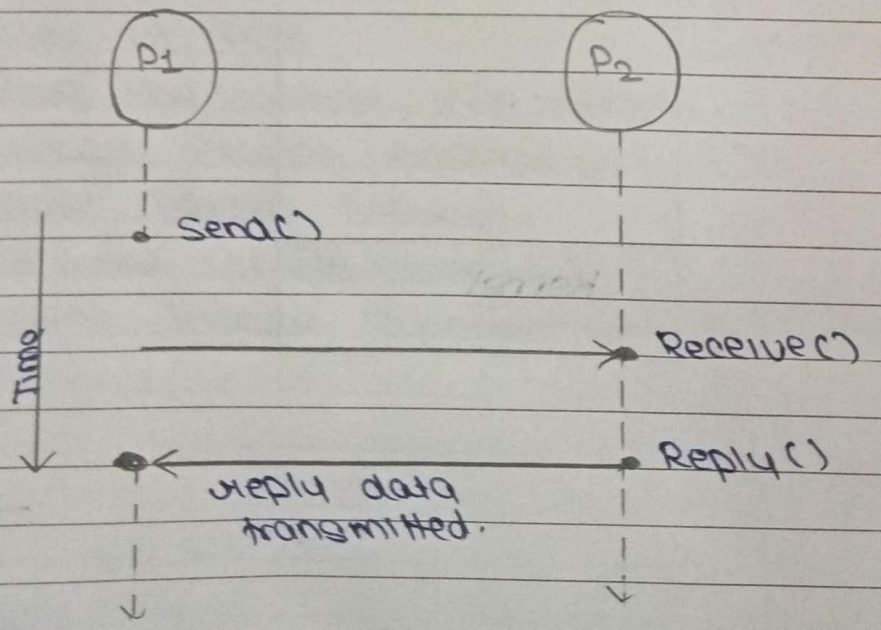
1. Shared memory model
- In shared memory model, the co-operating process shares a region of memory for sharing of information.
- Some operating systems use the supervisor call to create a share memory space.
- Similarly, some OS use file system to create RAM disk, which is a virtual disk created in RAM
- The shared files are stored in RAM disk to share the information between processes.
- The shared files in RAM disk are actually stored in the memory
- The process ~~the~~ can share information by writing and reading data to the shared memory location or RAM disk.



→

2 Message Passing Model

- In Message Passing model, data is shared between process by passing and receiving messages between co-operating process.
- Message Passing mechanism is easier to implement than shared memory but it is useful for exchanging smaller amount of data.
- In message passing mechanism data is exchanged between processes through kernel of operating system using system calls.
- Message passing mechanism is particularly useful in distributed environment where the communicating process may reside on different components connected by the network.
- A message contain the following information
  a. Header of message that identifies the sending and receiving processes
  b. block of data
  c. pointer to block of data
  d. Some control information about the process
-



Q5 Explain the role of PCB

- The OS maintains the information about each process in a record or a data structure called Process control Block
- Each user process has a PCB. It is created when user creates a process and it is removed from the system when the process is killed.
- All these PCBs are kept in the memory reserved for the OS.

Identifier
State
Priority
Program counter
Memory pointers
Contex data
I/O status info.
Accounting informat.
⋮

1. Identifier
   - Every process must have unique identifier
   - If there are n numbers of prosses, then Pid becomes 1,2,3 .... n
   - If Round Robin policy is followed, then after n, Pid is 1
2. State
   - It stores the current state of a process like new, running, blocked, terminate, etc.
3. Priority
   - The processes which are urgently required to be completed are set at higher priority while others are

set with lower priority.

- This priority can be set externally by the user/system manager, or it can be decided by the Operating System.

4. Program counter
- It gives us the address of next instruction which needs to be executed, after the current execution is completed.

5. Memory Pointers
- It stores the address of memory where process is stored

6. Context data
- It store the data which is currently required by the process

7. I/O status information
- It stores the outstanding i.e pending I/O request

8. Accounting Information
- This section stores the statistics like process start time, end time.
- Number of registers required, amount of memory, etc.

- Roles of PCB

1. Stores the process information
2. manage Process Scheduling
3. Tracks Process Resources
4. Facilitates Context Switching
5. Ensures Process Synchronization & Communication.

Q6 What is mutual Exclusion. Explain its significance

→ - While one process executes the shared variable, all other processes desiring to so at the same moment should be kept waiting.
- When that process has finished executing the shared variable, one of the processes waiting to do so should be allowed to proceed. In this fashion, each process executing the shared data excludes all others from doing so simultaneously.
- Only one process should be allowed to execute in critical section. This is called mutual exclusion.
- The mutual exclusion is put into effect only if processes access shared changeable data. If during execution, the operations of the processes do not conflict with each other, they should be permitted to proceed in parallel.

• Significance

1. Prevents Race around condition
- A race around condition occurs when multiple process access shared data simultaenously
- Mutual exclusion ensures that only one process modifies the resource at a time.
2. Data Integrity and Consistency
- Without mutual exclusion, concurrent acess can lead to inconsistent data
- For ea, in banking system, if two users withdraw money at same time, an incorrect balance might be recorded.

3. Deadlock and Starvation Prevention
- Proper implementation of mutual exclusion helps in avoiding deadlocks (where two or more processes wait indefinitely) and starvation
4. Synchronization in Multiprocessing Systems
- In multi-core processors, multiple threads may execute simultaneously. Synchronization mechanisms like mutual exclusion ensure orderly execution.
5. Efficient Resource Utilization
- Ensures that resources are allocated in controlled manner, improving system performance and stability.

Q7 Consider the following set of processes, assuming all are arriving at time 0. Calculate average waiting time and turn-around time for FCFS, SJF (Non-Premptive), Priority and RR (Quantum=2)

Process	AT	BT	Priority
P1	0	2	2
P2	0	1	1
P3	0	8	4
P4	0	4	5
P5	0	5	3

1. FCFS :-

	CT	TAT	WT
P1	2	2	0
P2	3	3	2
P3	11	11	3
P4	15	15	11
P5	20	20	15
	Average →	10.2	6.2

P1	P2	P3	P4	P5

0 2 3 11 15 20

2. SJF (Non-Pre-emptive)

	CT	TAT	WT
P1	3	3	1
P2	1	1	0
P3	20	20	12
P4	7	7	3
P5	12	12	7
	Average →	8.6	4.6

P2	P1	P4	P5	P3

0 1 3 7 12 20

3. Priority

	CT	TAT	WT
P1	3	3	1
P2	1	1	0
P3	16	16	8
P4	20	20	16
P5	8	8	3
	Average →	9.6	5.6

P2	P1	P5	P3	P4

0 1 3 8 16 20

4. Round Robin (Time Quantum = 2)

	CT	TAT	WT
P1	2	2	0
P2	3	3	2
P3	20	20	12
P4	13	13	9
P5	18	18	13
		11.2	~~35~~ 7.2

Ready Queue- ~~P1~~ ~~P2~~ ~~P3~~ ~~P4~~ ~~P5~~ ~~P3~~ ~~P4~~ ~~P5~~ ~~P3~~ ~~P5~~ ~~P3~~

↓

Running Queue – P1 P2 P3 P4 P5 P3 P4 P5 P3 P3 P3

0 2 3 5 7 9 11 13 15 17 18 20