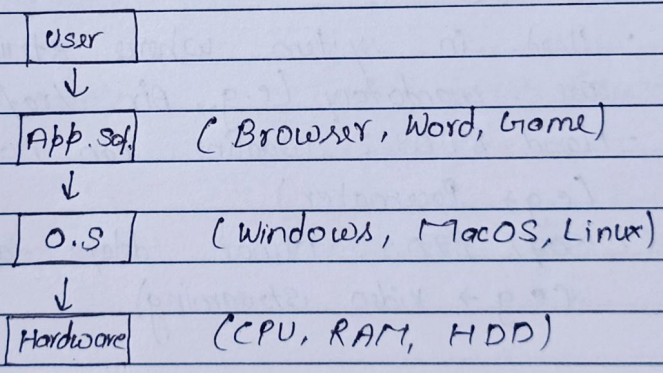
Date 21-01-26...

Operating System

An operating system is a system software that acts as an interface b/w the computer hardware and the User. It is the first program loaded into the computer's memory after booting.

- Main objects of O.S:
  - Convenience:- Makes the computer system easy to use (eg. windows GUI vs writing binary code.
  - Efficiency:- Ensures efficient use of computer resources like CPU, Memory and I/O devices.



- Types of operating System:

1. Batch operating system:
   - Users do not interact directly with the computer.
   - Similar jobs are grouped together into "batches" and execute one by one.
   - It's disadvantage is CPU remains idle during I/O operation.

Date.....

2. Multiprogramming OS :
- Keeps multiple jobs in memory simultaneously
- When a job waits for a I/O, the CPU switch to another job.
- It's goal to maximize CPU utilization.

3. Time-sharing OS (Multitasking)
- It is a logic extension of multiple programming.
- The CPU executes multiple jobs by switching among them ~~useful~~ using a fixed time slot (Time Quantum).
- It's advantage to provides quick response time to users.

4. Real-time OS :
- Used in system where strict time requirements are mandatory (e.g., Air traffic control, Robots).
- Hard RTOS : Deadline can not be missed. (e.g→ Pacemaker)
- Soft RTOS : Minor delays are acceptable (e.g → Video streaming)

- Operating System's Service

- Service provided by OS :

1. Process Management : Creating, shedduling and terminating process (programs).
2. Memory management : Allocation and de-allocation of RAM to different program.
3. File management : Reading / Writing data to Hard disk/ssd.
4. I/O " : Managing devices like printer, keyboard

APCO

Teacher's Sign.....

5. Secu

6. Error

P

- D

Process Management

- Difference b/w Program and Process

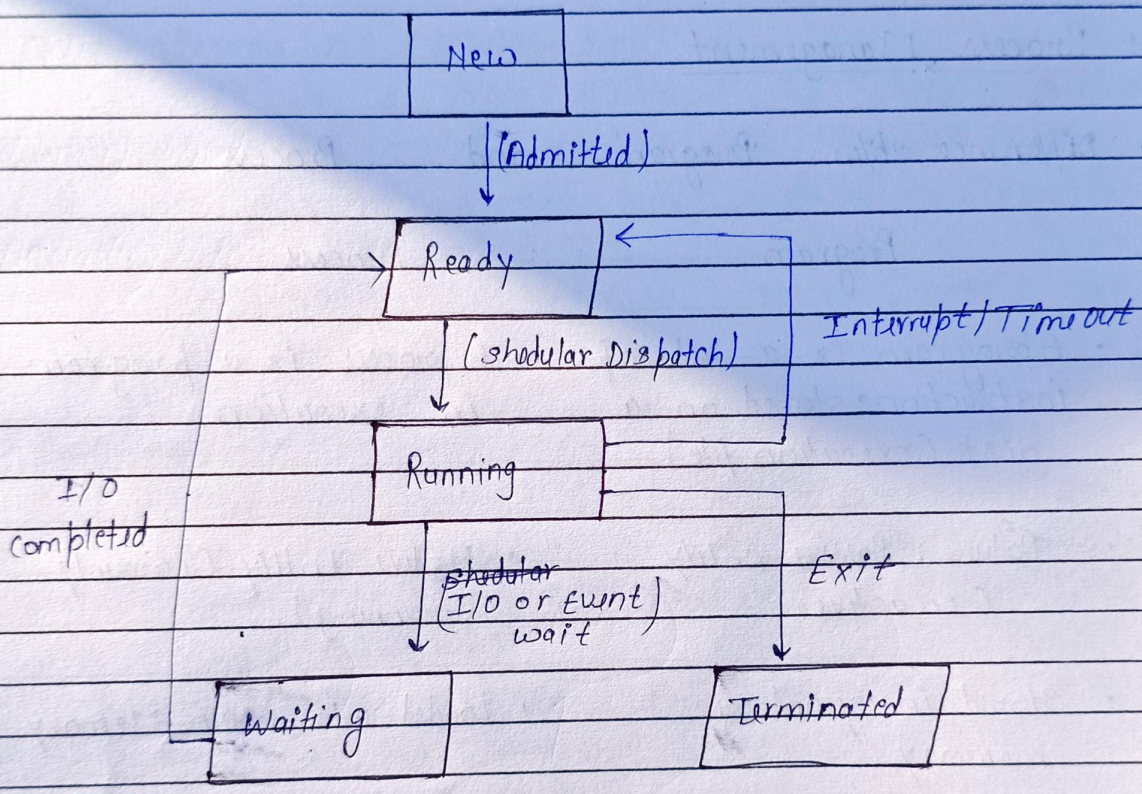
Program	Process
• A program is a set of instructions stored on a disk (execution file)	• A process is a program in execution.
• Nature: Passive Entity (Inactive).	• Active Entity (currently running).
• Stored in secondary memory	• Loaded into main Memory
• Does not require resources like CPU, Registers.	• Require resources like CPU, Memory, I/O and register.
• game.exe file on desktop	• Playing the game (running instance).

Teacher's Sign.....

Date.....

Process state Transition Diagram

A process change its state during its execution. A process can be in one of the following 5 states:



- Explanation:

1. New: The process is being created.
2. Ready: The process is loaded into Main Memory (RAM) and is waiting to be assigned to the CPU.
3. Running: Instructions are being executed by the CPU.
   In a single-processor system, only one process can be in the Running state.

Teacher's Sign.....

Date.....

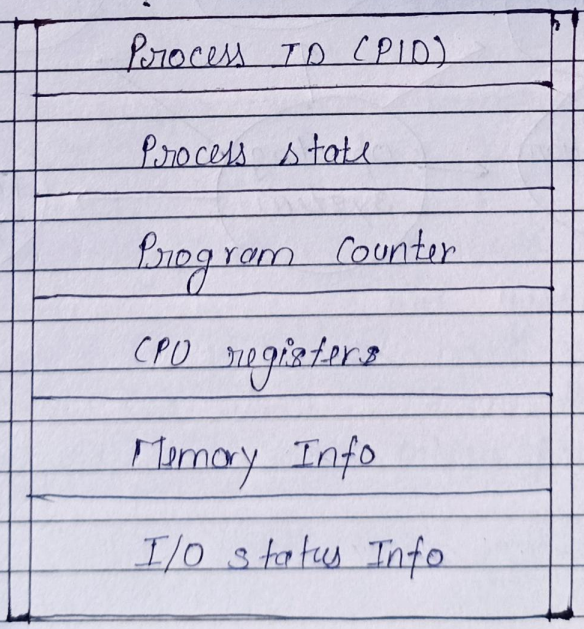
4. Waiting (Blocked) : The process is waiting for some event to occur (like I/O completion or user input). It cannot run even if CPU is free.

5. Terminated : The process has finished execution.

- Process Control Block (PCB)

Each process is represented in the operating system by a data structure called the process control Block or Task control Block. It ~~serves~~ serves as the repository for any information that may vary from process to process.

Process Control Block

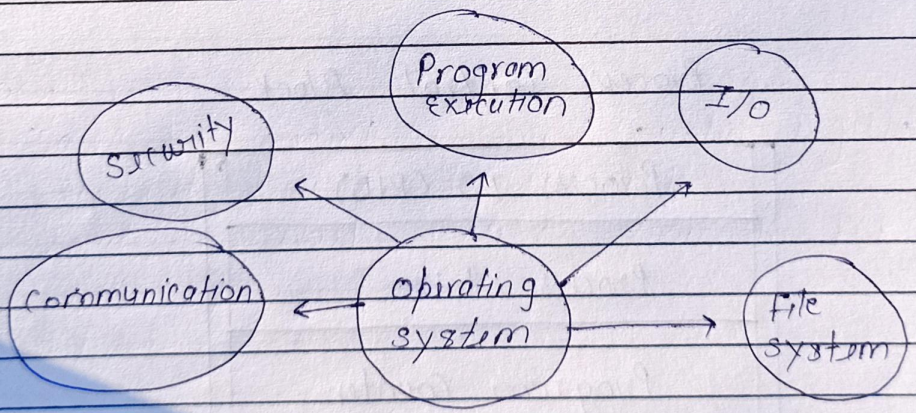


Date.....

- Key components :

1. Process ID : A unique number to identify the process
2. Process state : Current status (New, Ready, Running).
3. Program counter : Indicates the address of the next instruction to be executed.
4. CPU registers : Stores temporary data/calculations when the process is paused.
5. Memory Limits : Information about memory allocated (Base and Limit registers).
6. Accounting info : CPU usage time, time limits, etc.

- Diagram of OS's service



Teacher's Sign.....

Date.....

Threads & Multithreading

A Thread (Lightweight Process) is a basic unit of CPU Utilization. It constitutes a Program Counter (PC), register set and stack space.

- Key features: Threads belonging to the same process share common resources like Code section, Data section and Open Files.

- Difference b/w Process and Thread

Process (Heavyweight)	Thread (Lightweight)
• A program in execution	• A segment/component of a program
• Has its own independent memory space	• Shares memory with other threads of the same process.
• Communication b/w processes is slow (IPC)	• Communication b/w threads is very fast.
• Context switching is slow and heavy.	• Context switching is fast and lightweight.
• If one process fails, others are unaffected.	• If one thread crashes, the entire process may crash

Teacher's Sign.....

Date.....

- Types of Threads :

- User Level Threads : Managed by user-level libraries (without kernel support). Fast but if one threads blocks, the whole process blocks.

- Kernal Level Threads : Managed by the OS (kernal). slower to create but independent (if one blocks, crash, other can run).

Context switching

Context switching is the process of storing the state (PCB) of the currently running process and restoring the state of the next process to be executed

- Process

1. Save the context (Program counter, Registers) of the old process into its PCB.
2. Load the saved context from the PCB of the new process.
3. Resume execution of new process.

Context switching is considered overhead because the system does not useful work while switching.

APCO

Teacher's Sign.....

Date.....

Types of shedulers

Long term shedulers (Job shedulers)	short term sh. (CPU sh.)	Medium sh. (Swapper).
• It selects process from Disk and brings them to RAM	• Selects process from RAM and assigns them to CPU	• Moves processes b/w RAM and Disk (swapping)
• Less frequent (Sec/Min)	• Very frequent (Milliseconds)	• Medium frequency
• Controls Degree of Multiprogramming (How many processes are in Memory)	• Provides efficient CPU utilization	• Reduces degree of multiprogramming to free up RAM.
• New → Ready	• Ready → Running	• Ready Blocked ↔ suspende

APCO

Teacher's Sign.....

Date. 27-01-26.

Sheduling Criteria & Formulas

To Compare sheduling algorithms, we use the following criteria -

1. Arrival Time : The time at which the process enters the ready Queue.
2. Burst Time : The Total time required by the process for execution on the CPU.
3. Completion Time : The Time at which the process completes its execution.
4. Turnarround Time : The Total Time spent by the process in the system.
   - Formula

$$TAT = CT - AT$$

5. Waiting Time : The Total Time the process waits in the ready queue.
   - Formula

$$WT = TAT - BT$$

Teacher's Sign.....

Date.....

First ~~Time~~ Come, First-Serval (FCFS) Scheduling

FCFS is simplest scheduling algorithm. The process that requests the CPU first is allocated the CPU first is ~~allocated~~ allocated the CPU first. It is implemented using a fifo queue.

- Characteristics

1. Non-~~Prem~~ Preemptive: Once the CPU is allocated to a process, it keeps the CPU until it releases it (either by terminating or requesting I/O).

2. Convoy Effects: If a process with a large Burst Time arrives first, all other processes wait for a long Time, leading to low CPU utilization.

Ex→ Calculate Avrg Waiting Time

Process	AT	BT
P1	0	4
P2	1	3
P3	2	1

Sol→

P1	P2	P3
0	4	7	8

Process	AT	BT	CT	TAT	WT
P1	0	4	4	4	0
P2	1	3	7	6	3
P3	2	1	8	6	5

APCO

Teacher's Sign.....

Date.....

Avrg Waiting Time = $\frac{0+3+5}{3}$ = 2.66 units

Shortest Job First Scheduling

SJF associates with each process the length of its next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst.

- Characteristics -

- optimal : SJF gives the minimum Avrg Waiting Time for a given set of Processes.
- Models : It can be premptive (Shortest Remaing Time first) or Non-Preemptive. (We are discussing Non-Preeptive here.)
- Disadvantages: It is difficult to know the length of the next CPU burst in advance.

- Numericals Example (Non-Preemptive)

Consider the following processes

Processes	AT	BT
P1	0	4
P2	1	3
P3	2	1

Teacher's Sign.....

Date.....

Sol → Step 1 : Grantt chart logic

At time 0: Only P_1 is present. P_1 runs for 4 units (0 to 4)
" " 4: P_1 finishes. P_2 and P_3 are waiting
Decision : P_3 is smaller than P_2 ($1<3$), so P_3 runs next
At time 5: P_3 finishes, Now P_2 runs.

P_1	P_3	P_2
0	4	5	8

Processes	AT	BT	CT	TAT	WT
P_1	0	4	4	4	0
P_2	1	3	8	7	4
P_3	2	1	5	3	2

$$\text{Avrg WT} = \frac{0+4+2}{3} = \frac{6}{3} = 2 \text{ units}$$

Teacher's Sign.....

Date.....

Round Robin (RR) sheduling

Round Robin is a preemptive sheduling algorithm designed specifically for time-sharing systems. It is similar to FCFS, but preemption is added to switch b/w processes.

- Mechanism (Time Quantum)
  - A small unit of time, called a Time Quantum (or Time slice) is defined (e.g. 2ms to 100ms)
  - The Ready Queue is treated as a circular queue. The CPU sheduler goes around the queue, allocating the CPU to each process for a time interval of up to 1 Time Quantum.
- Working principle
  - If Burst Time < Time Quantum : The process will finish its execution and release the CPU voluntarily.
  - If Burst Time > Time Quantum : The Timer will go off, the process will be interrupted (context switch) and it will be placed at the tail (end) of the ready Queue.
- Advantages -
  - Fairness. Every process gets a share of CPU. No processes waits indefinitely.
- Disadvantages -
  - If time Quantum is too small, context switching overhead increases (system become slow).

APCO

Teacher's Sign.....

Date.....

If Time Quantum is too large, it behaves like FCFS.

- Round Robin Numerical

Time Quantum = 2 Units

Process	AT	BT
P_1	0	5
P_2	0	3
P_3	0	1

Soln → 1. start (Time = 0)
Ready Queue = P_1, P_2, P_3

2. Turn 1 (TQ = 2)
0 → 2
$P_1 \Rightarrow 5 - 2 = 3$
R.Q = P_2, P_3, P_1

3. Turn 2 (TQ = 2)
0 → 2 → 4
$P_2 \Rightarrow 3 - 2 = 1$
R.Q = P_3, P_1, P_2

4. Turn 3 (TQ = 2)
(4 → 6)
but $P_3 = 1$
so → 4 → 5) P_3 finished
R.Q = P_1, P_2

5. Turn 4 (TQ = 2)
P_1 (1), 5 → 7
P_2 finished, 5 → 7
R.Q = P_1

5. Turn 4 (TQ = 2)
5 → 7
$P_1 \Rightarrow 3 - 2 = 1$
R.Q = P_2, P_1

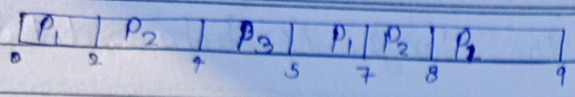
6. Turn 5
P_2 (1)
7 − 8 , P_2 finished
R.Q = P_1

7. Turn 6
P_1 (1)
8 − 9 . P_1 finished.

APCO

Teacher's Sign.....

Date.....



Calculation table

Process	BT	CT	TAT	WT
P_1	5	9	9	4
P_2	3	8	8	5
P_3	1	5	5	4

Avrg waiting Time = $\frac{4+5+4}{3}$

$= \frac{13}{3} = 4.33$ units

Teacher's Sign.....

Priority Scheduling

A priority ~~scheduling~~ number is associated with each process. The CPU is allocated to the process with the highest priority.

- Preemptive: CPU is taken away if a higher priority process arrives.
- Non-preemptive: CPU is not taken away until the current process completes.

- The problem: Starvation (Indefinite Blocking) Low-priority processes may be left waiting indefinitely if high-priority processes keep arriving. This situation is called starvation.

- The solution: Aging is a technique of gradually increasing the priority of processes that wait in the system for a long time.

Ex → Increasing priority by 1 every 15 minutes. Eventually, a low-priority process will become the highest priority process.

Teacher's Sign.....

Date.....

**Comparison of Scheduling Algorithm

Algorithm	Type	Description	Best Application
FCFS	Non-preemptive	Processes executed in Arrival order. (convoy effect)	Batch systems
SJFS	"	Shortest burst time executed first. (Minimizes Waiting Time)	General purpose (Difficult to implement)
Round Robin	Preemptive	Fixed Time Quantum for each process. (fairness)	Time-sharing systems (PC, Mobile)
Priority	Both	Highest priority executed first. (Risk of starvation).	Real-Time System

Date.....

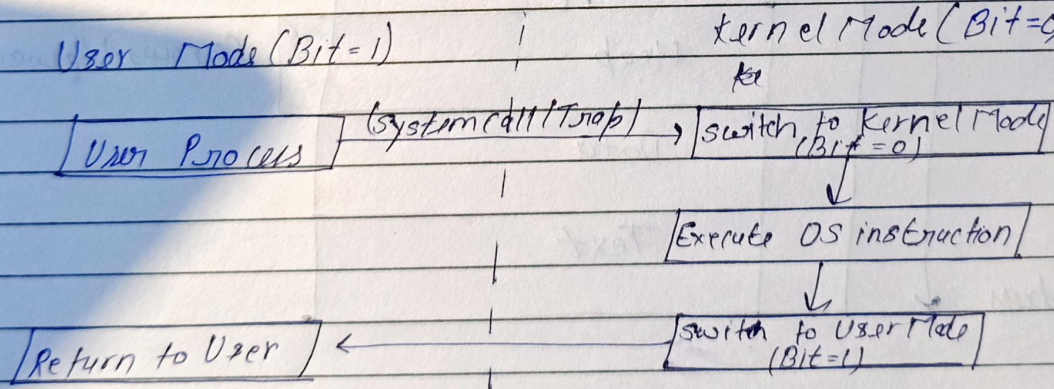
Dual Mode operation

- To Ensure proper execution of the operating system, We must be able to distiguish b/w the execution of operating-system code and user-defined code. Hardware provides at least two modes of operation:

- User Mode: When the Computer system is running user application (like word, Browser)

- Kernel Mode: When the operating system is executing specific tasks (like accessing hardware or memory). Also called system Mode or Privileged Mode.

* Mode Bit: A Bit called the mode bit, is added to the hardware of the computer to indicate the current mode:
  - 0 implies Kernel Mode
  - 1 implies User Mode



System call and Mode switching flow

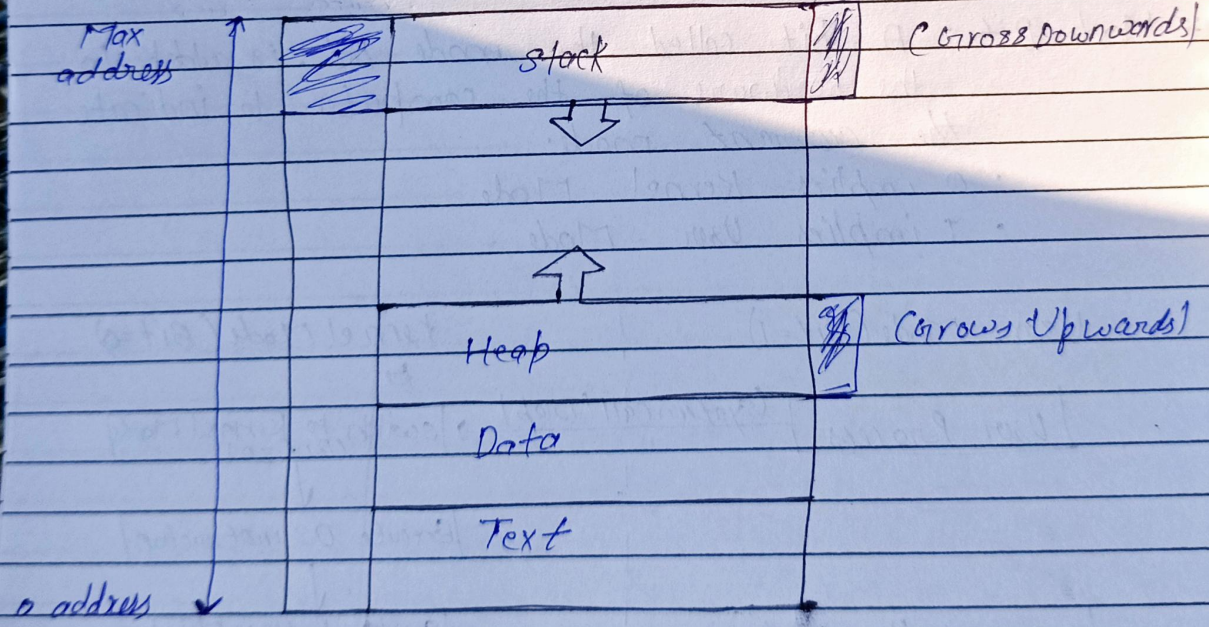
Teacher's Sign.....

Date.....

Process Memory Layout

A process in Memory is divided into four sections:

1. Text section : Contains the executable code (program instructions). It is read only.
2. Data section : Contains global and static variables
3. Heap : Used for dynamic memory allocation during run time (e.g., using malloc() in C). It grows upward.
4. Stack : Used for temporary data storage (function parameters, return addresses and local variables). It grows downward.



Date.....

Distributed & Network Operating System.

1. Distributed operating System (DOS):
   - Multiple independent computers (nodes) communicate via a network to appear as a single system to the user.
   - The user does not know where their file is stored or which CPU is processing their task.
   - Computation is fast and reliable.

2. Network operating system (NOS)
   - An OS that run on a server and manages data, users, groups and security for networked computers.
   - Users are aware of the existense of other computers and must explicitly log in to remote machines to transfers files.
   - Windows server, Linux.

APCO

Teacher's Sign.....

Date.....

Multilevel Queue scheduling

- In this algorithm, the Ready Queue is partitioned into several seperate Queues. processes are permanently assigned to one queue based on some property (e.g. memory size, process type).

- Example Structure:

- Queue 1: System Processes (Highest Priority) - Uses Round Robin.
- Queue 2: Interactive Processes (Medium Priority) - Uses SJF
- Queue 3: Batch Processes (Lowest Priority) - Uses FCFS

- Rule ⇒ No process in a lower-priority queue (e.g Queue 3) can execute unless all higher-priority queues (1 & 2) are empty.