OPERATING SYSTEM

SYSTEM SOFTWARE

Software that performs basic tasks such as running applications managing files and correcting errors.
It includes Bios, OS and utility programs.

BIOS : Basic Input Output System is the built in software on motherboard that :

1. Starts computer
2. performs Power on Self test (POST) to ensure critical systems are working.

Operating system : It provides the user interface that humans use to communicate commands and recieve feedback

- It communicates with hardware, instructing it to accomplish tasks
- It runs applications and enables humans to interact with them
- It controls and manages the file storage system.

EVOLUTION OF OPERATING SYSTEM

1. Early Systems (1940's - 50s) : In beginning there were no operating system and each task had to be executed manually The programs were run sequentially, and computers could only handle one job at a time.
2. Mainframe era (1960's) : As computing demand grew, multiprogramming was introduced where multiple jobs could be loded into memory and executed, improving efficiency. These were large systems used in enterprises.
3. Mini computers (1970's) : The rise of smaller, less expensive computers required more versatile OS'es. These system also introduced multi user capabilities allowing several users to access the machine concurrently.
4. Personal Computers (1980's) : With the advent of personal computers, user friendly operating systems like MS-DOS and Early windows made computers more accessible. Graphic User. Interface GUI replace text base GUI.
5. Modern OS (1990 - Present) : Operating system today manage multi-core processors, handle complex networks and offer ~~exclusive~~ extensive user-friendly features such as GUI's, multitasking and advance settings and better security.

Operating System Services & Functions:

User Interface (UI) : The OS provides an interface for the user to interact with the system, which can be a command line interface (CLI) or graphic user interface (GUI)

Process Management : The OS manages processes including their execution scheduling and termination. It ensures processes are executed without interfering with each other.

Memory Management: The OS manages the computers Memory ensuring that each process has enough space to execute and memory is efficiently utilized.

File System Management : The OS manages, organizes and maintains files on storage devices, providing a way to create, store, access, and delete files.

Device Management : OS co-ordinates hardware devices like printers ~~had~~ hard drives and displays providing an interface for software to communicate with hardware

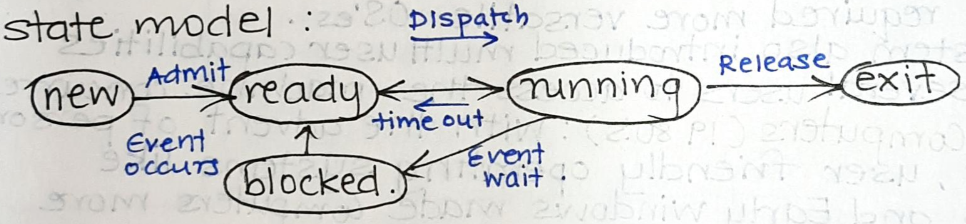
Security and Access Control : OS ensures the systems security by enforcing access control and authentication method, protecting data from unauthorized access.

PROCESS MANAGEMENT

PROCESS : A process is a program that is currently being executed. It includes the program counter, registers and variables in memory. Each process is isolated, and the OS controls its execution.

PROCESS STATES :

5-state model :



new: process being created.

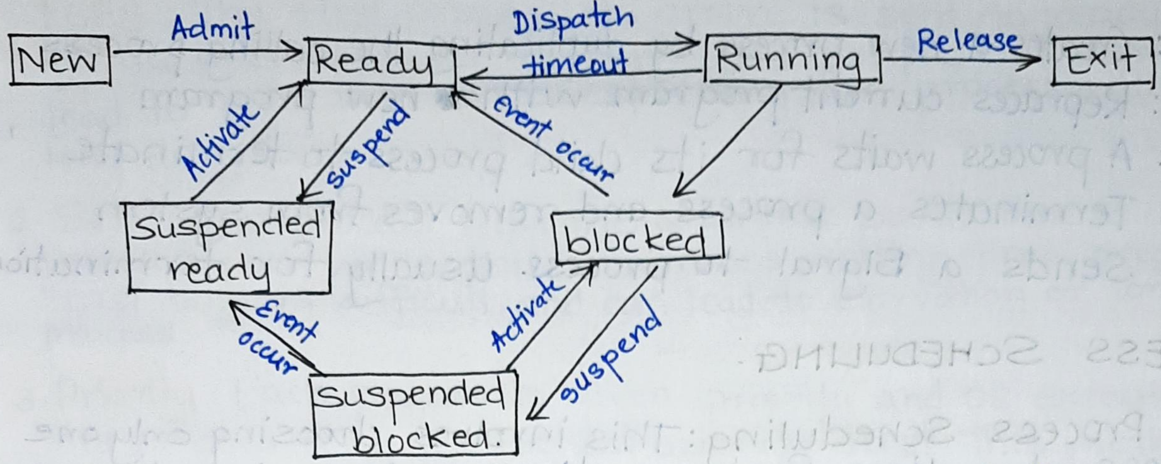
ready : The process is ready for execution, waiting for CPU time

running: The process is currently being executed on the CPU.

waiting (blocked): The process cannot continue until specific event occurs.

terminated : The process has finished its task.

7 state Model



6. Suspended ready : Process is in memory but not executing. It could be ~~stopped~~ swapped to disk when needed

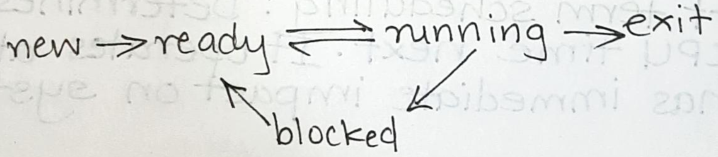
7. Suspended block/waiting : Process is waiting for an event but has been moved to disk

PROCESS CONTROL BLOCK

Process ID	: Uniquely identifies each process
Process State	: Current state of process
CPU register	: Holds value for context switching
Program Counter	: Points to next instruction to be executed
Memory Management Info	: Tracks memory allocation for the process
Scheduling Info	: Stores priority & scheduling info of process

Thread : A thread is the smallest unit of execution within a process.
Multiple threads can run within same process parallely
This makes thread management more efficient than process management.

Thread life cycle.



Multithreading.

One to one	One-to-Many	Many to Many
each user level thread is mapped with one kernel thread. Offers better performance but consumes more resources.	One kernel thread is mapped to many user level threads. Its simple but not efficient coz if one thread is blocked the rest are also	Many user threads are mapped with many kernel threads. This strikes balance between efficiency & resources

PROCESS CONTROL SYSTEM CALLS

Fork : Creates a new process by duplicating the calling process

Exec : Repraces current program with new program

Wait : A process waits for its child process to terminate

Exit : Terminates a process and removes from system

Kill : Sends a Signal to process. usually for termination

PROCESS SCHEDULING.

- Uni Process Scheduling : This involves choosing only one process at a time for execution as there is only one CPU.

SCHEDULING TYPES :

Preemptive : The OS can interrupt a running process and assign a different process. It ensures responsiveness and fairness.

Non Preemptive Scheduling : Once a process starts execution it continues until it completes or voluntarily yields to the CPU. This method reduces the overhead of context switching but can cause delays if a long process monopolizes the CPU.

SCHEDULING LEVELS :

1. Long term scheduling : Decides which process are admitted into the system. It controls the degree of multiprogramming.
2. Medium term scheduling : Moves process between memory and disk to manage load. Often used in systems with virtual memory
3. Short term scheduling : Determines which processes get CPU time next. It operates at higher frequency and has immediate impact on system responsiveness

SCHEDULING ALGORITHM :

1. FCFS : The first process to arrive is sent to ready queue and is executed first. Its simple but can lead to long wait time, espicially for process with long durations.

2. SJF : The process with the shortest burst time is selected next. while this minimizes waiting time predicting Burst time is difficult and can lead to starvation of longer process.

3. Priority : Each process is given priority and os executes the process with the highest priority first. This method can lead to starvation of lower priority process so some times aging mechanism is used to increase the priority of process.

4. Round Robin :

Each process is assigned a fixed time slice (quantum). After each time quantum the process is sent to the end of ready queue and next process of ready queue is taken up for execution. Its effective for time sharing system but can lead to longer wait time for processes if time quantum is too large.