(4) Security and Privacy → Processes should not interfere with each others data

L-1.2 : Batch operating System

Types of operating system

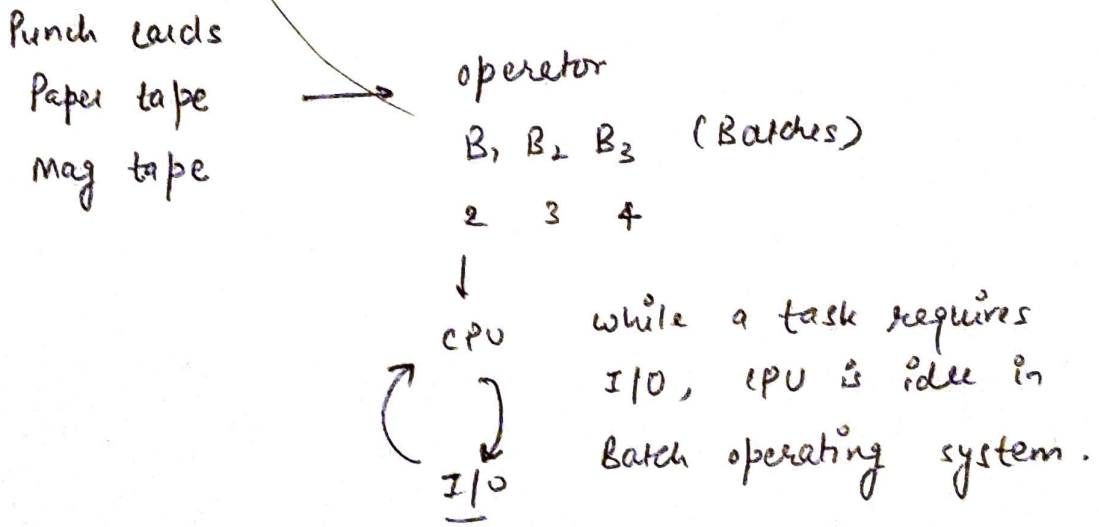
(i) Batch (ii) Multiprogrammed (iii) Multitasking

(iv) Real time OS (v) Distributed (vi) Clustered.

(vii) Embedded

→ Batch operating system

↳ Batch of similar tasks clubbed together.



1-1.3 Multiprogramming and Multitasking OS

↓ came first

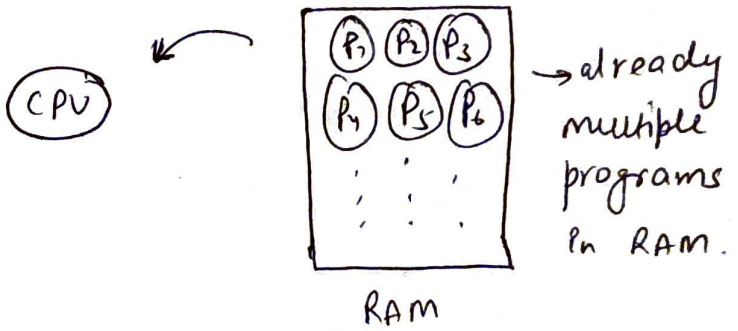
↳ bring as many processes as possible into the RAM.

→ Non preemptive scheduling is done in Multiprogramming

↳ CPU will completely execute one task before moving to another. unless the process itself moves to let's say input/output operation.

In that case, CPU will start executing the next process.

10 students
5-5 questions



→ Multitasking / Time sharing OS (preemptive)

↳ predefined time is given to a task, if it completes, well in good, if not CPU moves to next task. Idleness is minimum in this too but advantage over Multiprogramming is quicker response time.

Responsiveness ✓

→ We use Time sharing in laptops.

(only 2 questions at a time for any student)

1.1.4 Types of OS

→ Real time OS → Immediate output is required.
→ time constraint is there.

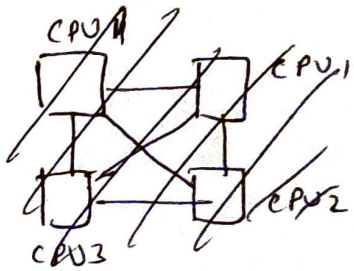
(i) Hard real time OS → missile systems
(strict time constraint, no delay) → airplane navigation

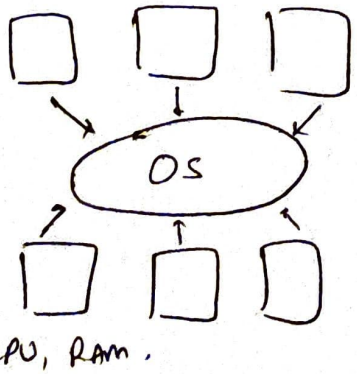
(ii) Soft real time OS → gaming
(not very strict) → youtube livestreaming.

→ Distributed environment

↳ Different processing units are connected by a network.

→ also called loosely coupled systems.





CPU, RAM.

→ Clustered OS

↳ Local form of distributed systems

→ example, OS shared by a company's computers.

→ Embedded

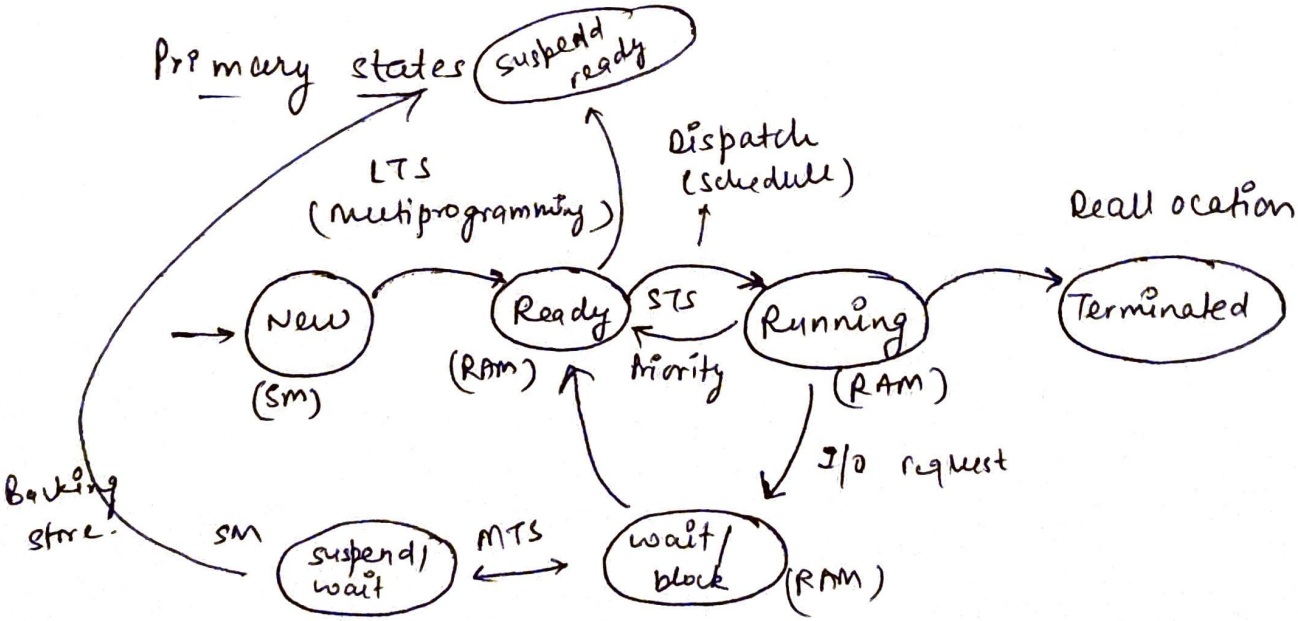
↳ fixed over your functionality

→ cannot be reprogrammed

→ used in devices like microwave, washing machines

L-1.5 Process states | schedulers

↳ states of a program from start to end.



New - program is created and stored in secondary memory

Ready queue - Now, the process has entered RAM.

→ Long Term scheduler brings the processes from New to Ready state.

LTS wants to convert maximum processes into ready state.

→ In uniprocessor systems, processes from Ready Queue are dispatched one by one to CPU.

→ Running state → also in RAM, but address changes from Ready state

→ Multitasking → Suppose if a higher priority process comes, then CPU pushes the current process in ready queue.

Time Quantum → CPU gives fixed amount of time.

→ Short term scheduler

↳ Between Ready and Running states

→ I/O process goes to ready queue

→ If all processes are I/O, then wait block fill get filled, if it overflows, MTS (medium term scheduler) will send some processes in secondary memory.

suspend ready - Let is say ready queue is full and then a high priority task comes, Medium term scheduler (MTS) transfers some tasks to suspend ready.

If suspend I/o block has read the file but wait block is full, then the process transfers to suspend ready / Ready queue. This is called backing store