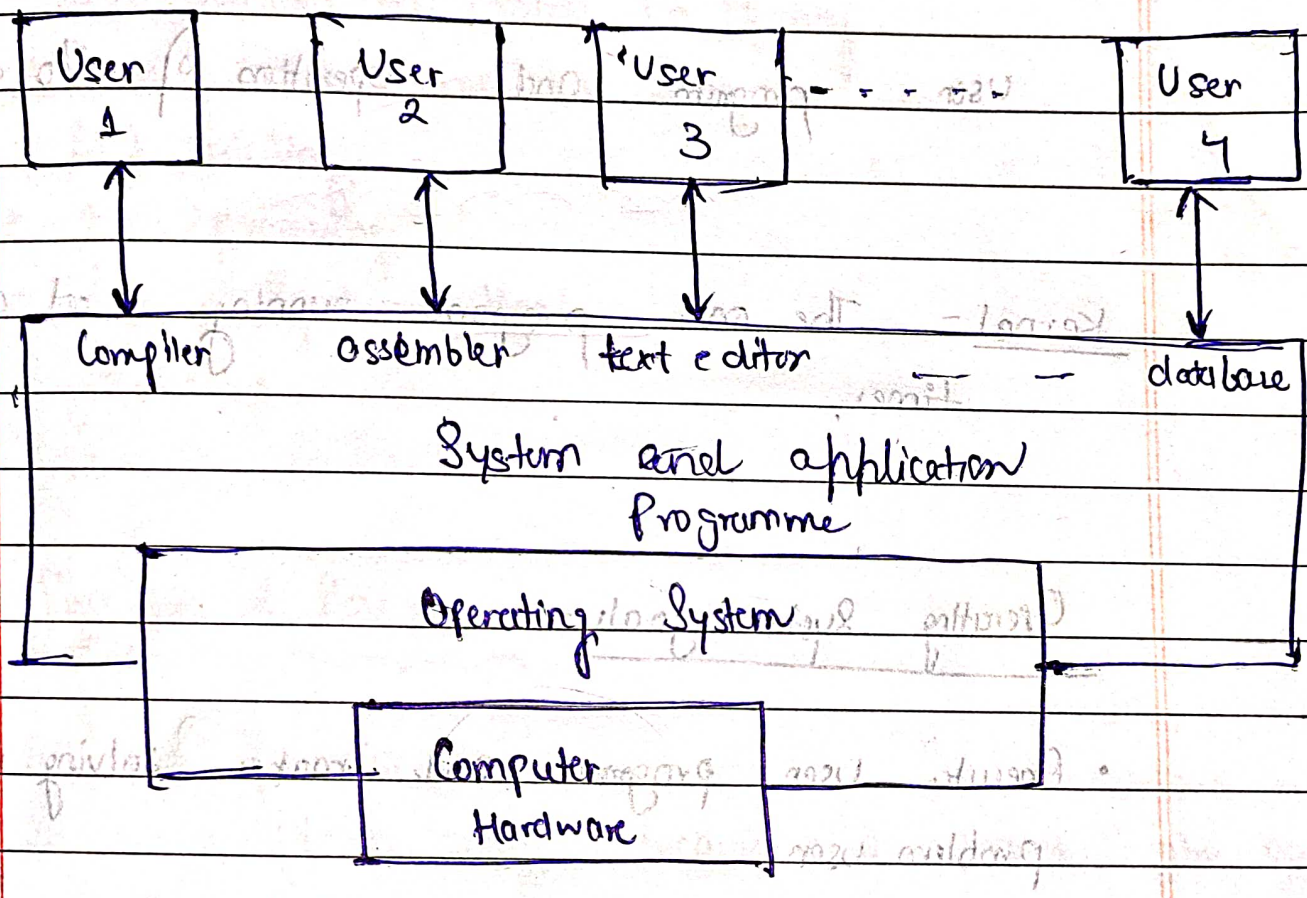
What is Operating System?



Operating system act as a interface between users and the hardware.

There are 2 views of operating system

- Top-down view — (view of User)
- Bottom up view — (view of Hardware)

2 modes of OS – • User Mode
• Kernel Mode

PAGE No
DATE

Hardware sees a operating system as resource manager, which manages and allocates resource.

Controls programs – OS control the execution of user program and operation of I/O devices.

Kernal – The one program running at all times.

Operating System goals –

- Execute user program and make solving user problem user.
- Make computer systems convinient.
- Use computer hardware in an efficient manner.

Primary goal – To make computer usage easy.

vibhu
PAGE No
DATE

PROCESS MANAGEMENT

- ✓ - Program and Process
- ✓ - Representation of Process in Memory
- ✓ - Process State transition Diagram.
- ✓ - CPU Scheduling
- Multiprocessing
- Process Synchronization
- Deadlock

Program & Process

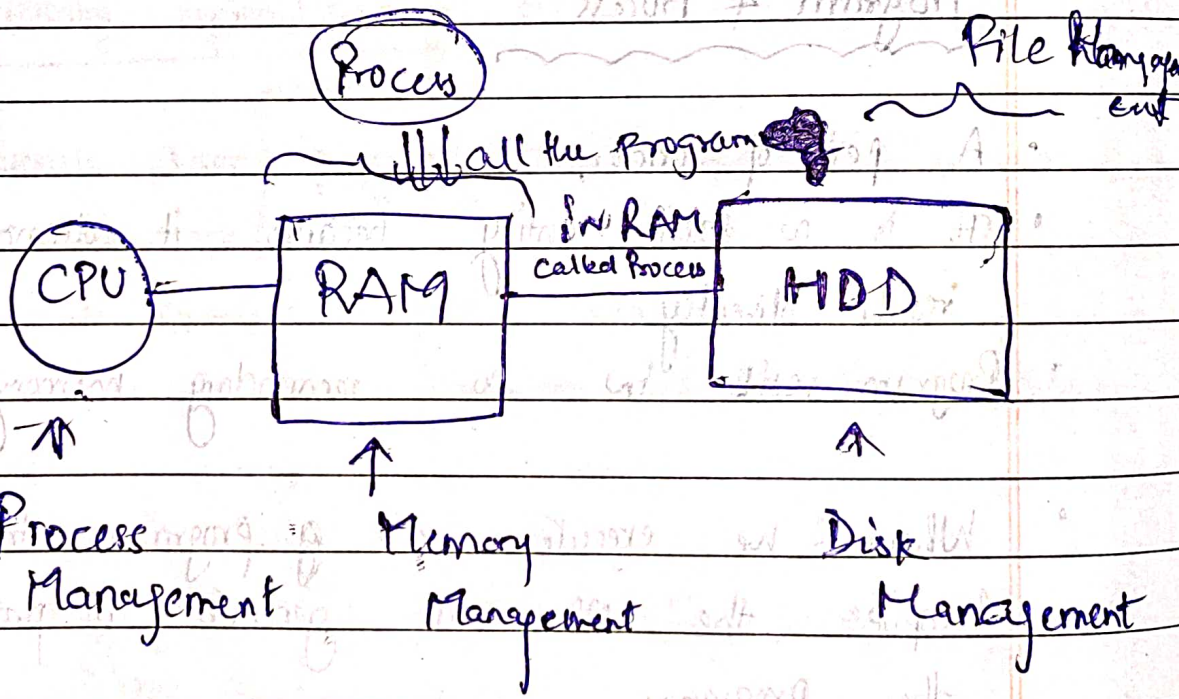
- A peice of code.
- It is a passive entity because it doesnot give a result directly.
- Program resits in a secoundary memory.
- When we execute a program that just compiles, the OS will generate a process to execute the program.

Process

Program in execution is called Process

1 Program can have multiple processes

Process can be defined as an instance of a program running on a computer or as an entity that can be assigned to and executed on a processor.



~~Multiprocessing &~~

Multiprogramming & Multi-tasking are same

TYPES OF SYSTEM (on the basis of Processor)

Uni-processing → 1 CPU

→ Uni-programming - one program at one time

→ Multi-programming

→ ~~Multiprocessing~~ Multitasking / Time sharing

Multi-processing → 2 or more CPU

Uniprogramming

Suppose 3 processes

P_1 P_2 P_3

30 30 30

	Fetch	Decode	Exe	
P_1	10	10	10	①
P_2	10	10	10	②
P_3	10	10	10	③

Total Time to complete all process = 30 + 30 + 30
= 90 units.

vibhu
PAGE No
DATE

Multi-Programming

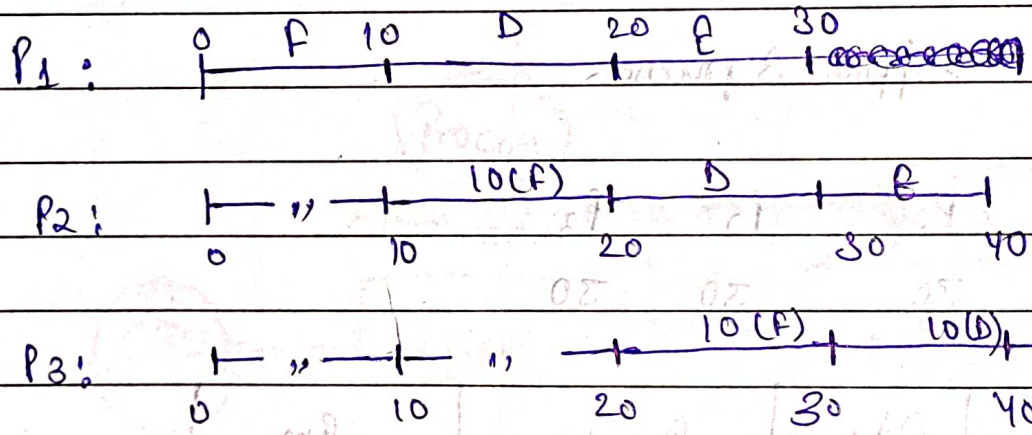
- Concept of pipeline come into the picture.

Suppose 3 Processes

P_1	P_2	P_3
30	30	30

More than 1 process can be done simultaneously ie with the help of pipelining is called multiprogramming.

10 → Fetch
10 → Decode
10 → Execute.



Total time to complete all process = 50 units

Instruction Cycle - fetch, Decode, Execute.

Time Sharing Systems

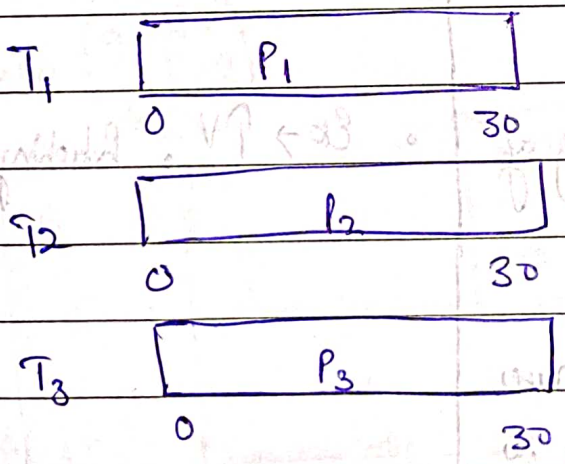
The CPU is multiplexed among several jobs that are kept in memory and on disk.

A job swapped in and out of memory to the disk.

Multiprocessing

- 2 or more CPU.

3 Process P_1, P_2, P_3



Total Time = 80 units.

Real Time System

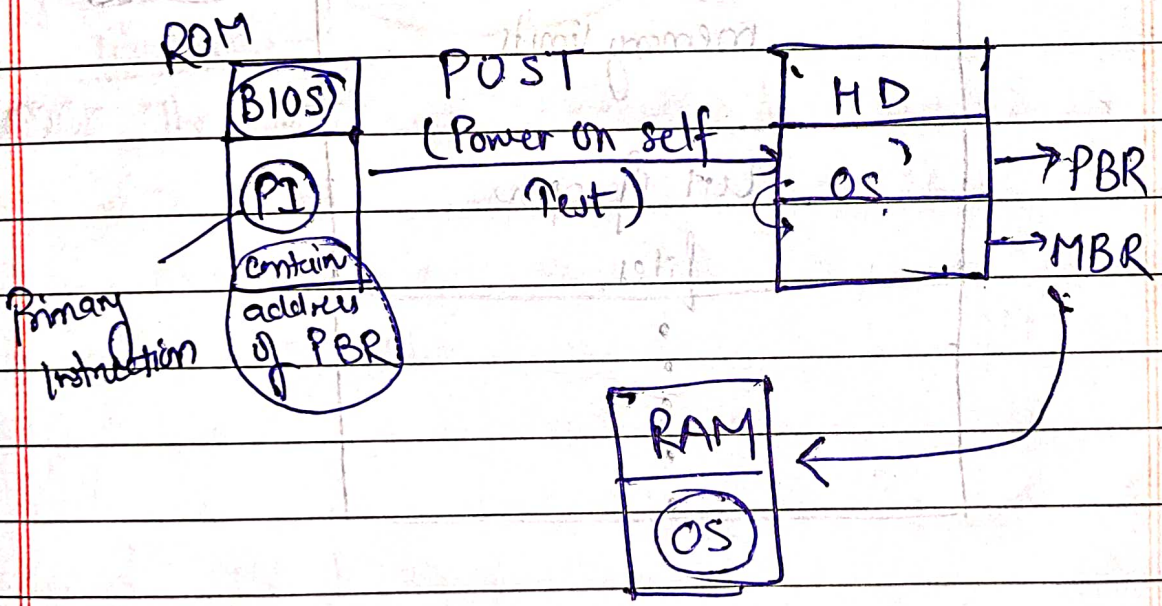
Real Time System (All system which is running practically

Hard Real Time System (Critical System)	Soft Real Time System.
We cannot afford any kind of error in those system.	In which loss is less.
• In which loss is very much	
• Ex → Heart Surgery Machine	• Ex → TV, Washing Machine.
• System which have to perform a task in a particular time.	

→ Difference b/w service and function.

- The working is called functions.
- The output provided is called service.

Booting

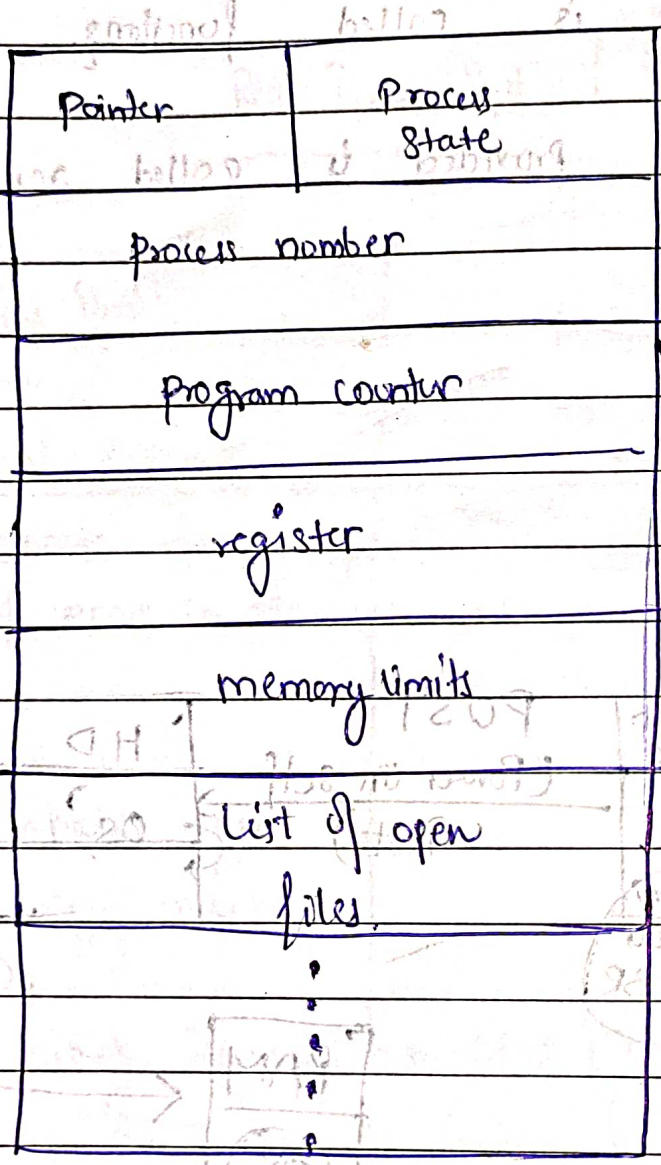


POST- checks all the hardware (Primary Hardware).

PBR- Primary Boot Record
MBR - Master Boot Record.

Vibhu
PAGE No
DATE

Data structure is used to store process in memory is called PCB (Process Control Block).



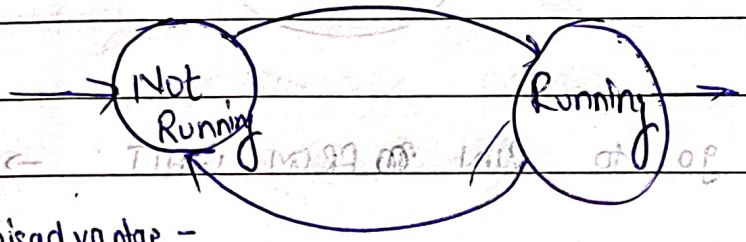
PCB

PCB Contain only the primary information.

Process State -

It is useful in knowing whether in which state the process is.

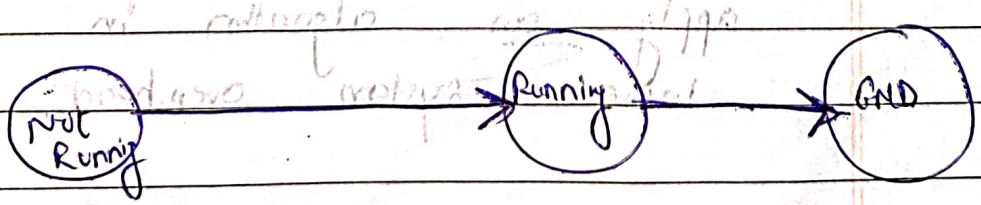
2 State Model



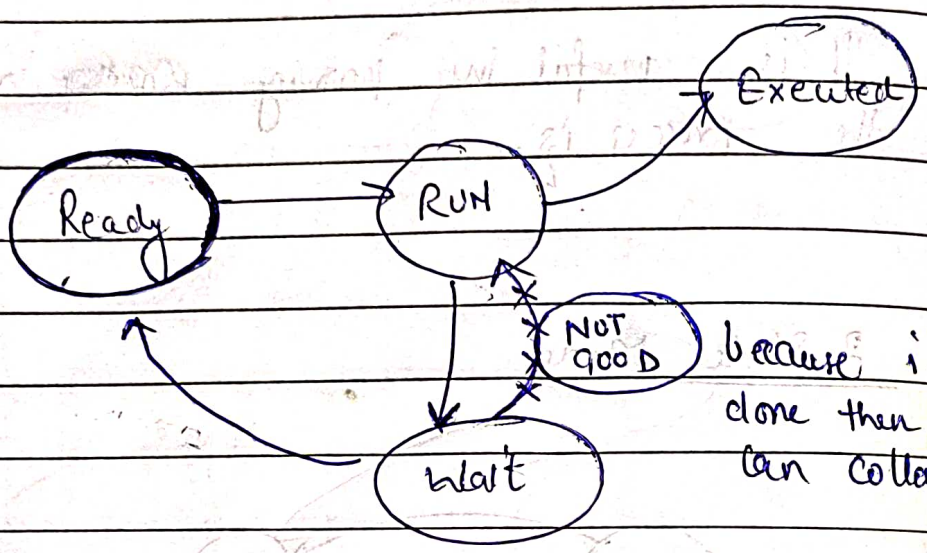
Disadvantage -

- There is no defination of Not Running state.
- → We are not able to identify whether the process is executed or not or waiting for its turn.

3 State Model



4 - State Model



because if it done then process can collapse.

- Can we go to RUN FROM WAIT → Yes,
- it is good to go TO RUN FROM WAIT

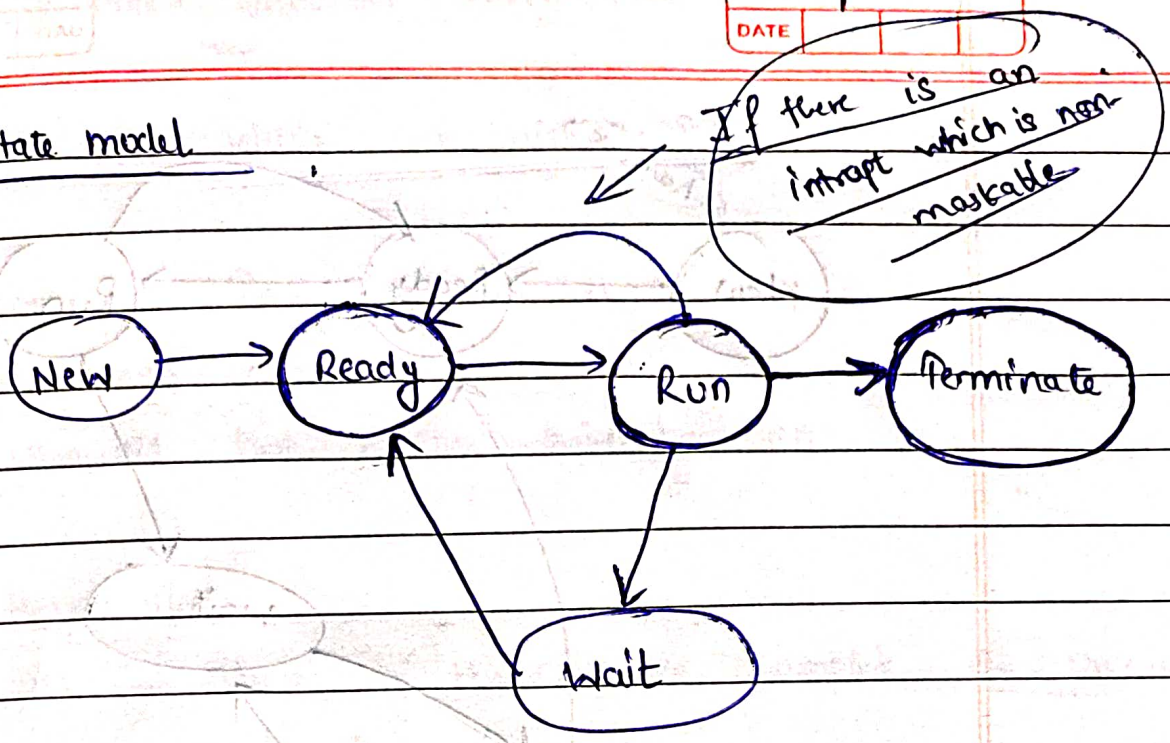
NO, because in ready there is an sheduling algorithm, for selection of process

For going from wait to RUN we have to apply an algorithm in wait also, which increase system overhead.

vibhu

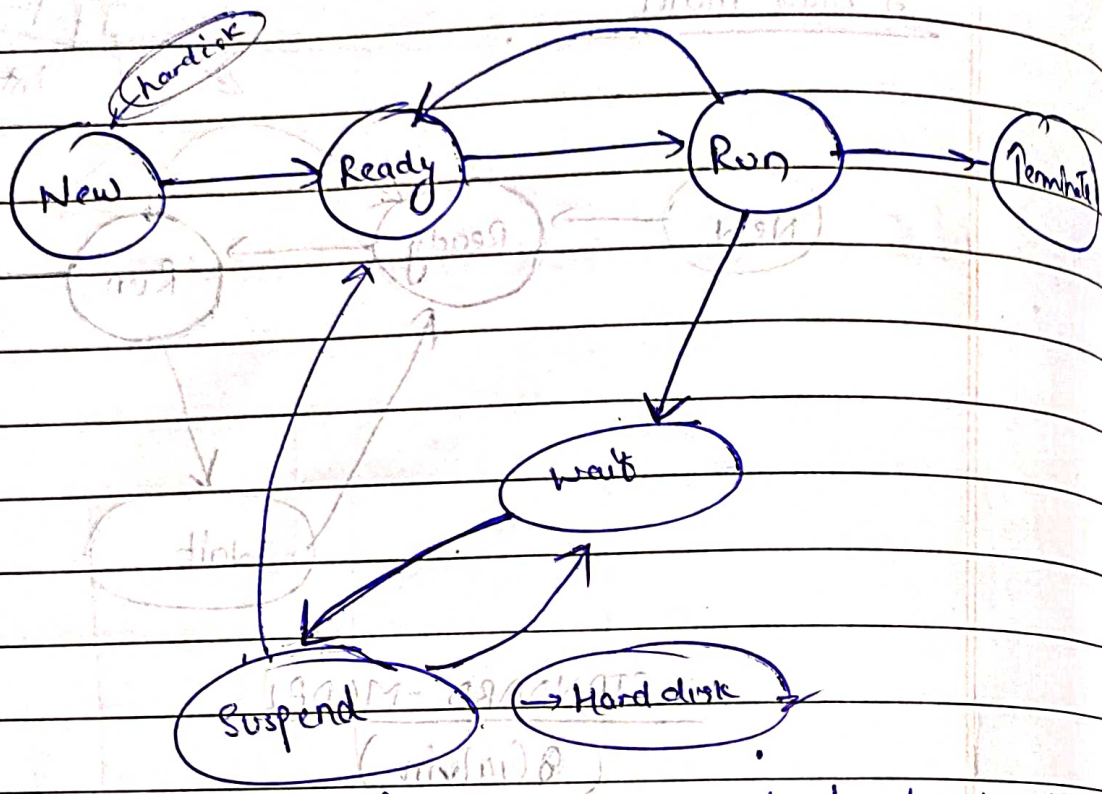
PAGE No
DATE

5 State model



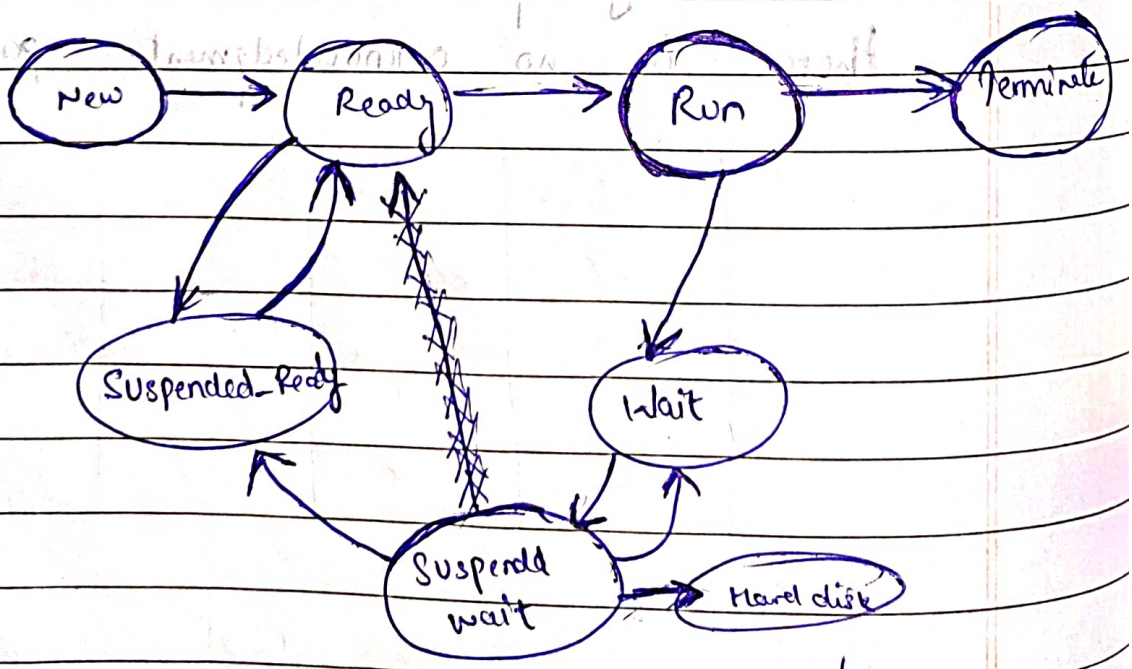
STANDARD - MODEL
(@Galvin)

- Termination meaning when process go out from RUN.
- We cannot go from wait to terminate directly because there is no acknowledgment path over there.



New & Suspende state is in hard disk.

6 - State Model



7 - State Model

TURN AROUND TIME = BT + WT

Wait time = Start Time - Arrival Time

vibhu

PAGE No

DATE

Criteria for selecting an algo

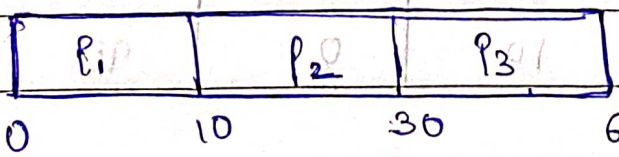
1) Waiting time → Total wait time.
2) Response time.
   ↳ When did first response given by CPU.
3) Turn Around Time
   ↳ Time taken by the CPU to complete the process

TAT = Waiting Time + Running Time

Algorithm

1) First come first Serve Algo.

	Brust Time / CPU time	Arrival Time	WT	TAT
P_1	10	0	0	10
P_2	20	0	10	30
P_3	30	0	30	60



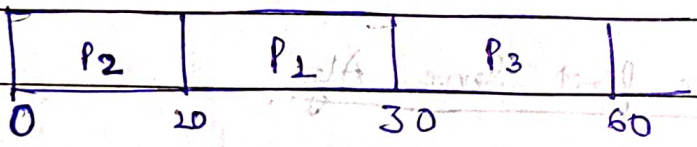
↳ Gannt Chart

Arrival Time - Arival time is a time when process is arrived at ready queue.

If arrival Time is not given, it is assumed all process arrived at same time.

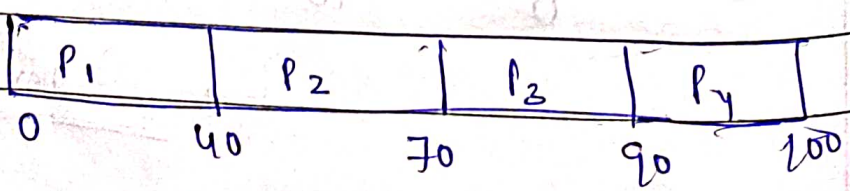
Ex-2

	Arrival Time	Burst Time	WT
P_1	10	40	10
P_2	0	20	0
P_3	20	30	10



Ex-

	BT	AT	WT
P_1	40	0	0
P_2	30	0	40
P_3	20	0	70
P_4	10	0	90



vibhu

PAGE No

DATE

In FCFS algorithm there is very high waiting time.

So we use SJF (Smallest Job first algo).

PROCESS IN MEMORY

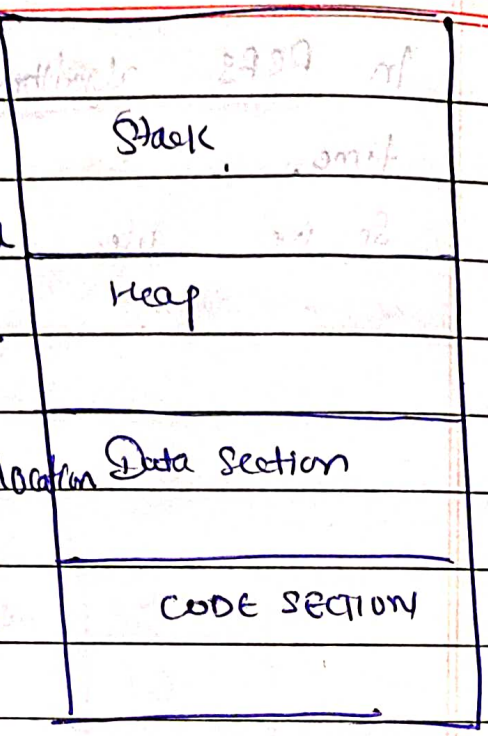
vibhu
PAGE No
DATE

Code Section - Code.

Data Section - contain static & ~~dynamic~~ global data variables.

Heap - Dynamic memory allocation

Stack - Store ~~the address of~~ addresses ~~return address and~~ function parameters, local variables and other temporary data, addr of next instruction during the function call.



~~PCB~~

Contains Information associated with each process

- Process id
- Process state
- Program counter
- CPU register.
- CPU scheduling info
- Memory - management information.
- Accounting information
- I/O status info.

PCB is a datastructure that maintains all the information pertaining to the process.
The lifetime of PCB is equal to its corresponding process.

SCHEDULAR AND DISPACHER.

Shedular - Which shedule something.
Dispacher - It will send something from one place to another.

Shedular is a program which is used to shedule processes
It is a primary program in the main memory.

Dispather is a program is used for dispatching process from one state to another.

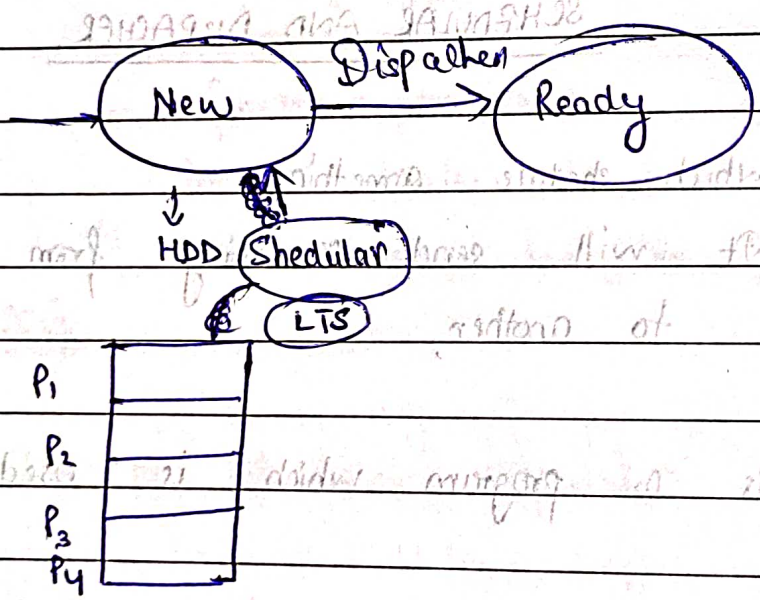
There are 3 type of Shedular :-

1) Long term
2) Mid term
3) Short Term

Long Term Shedular (LTS)

Sheduling process from new state to ready state is done by long term shedular

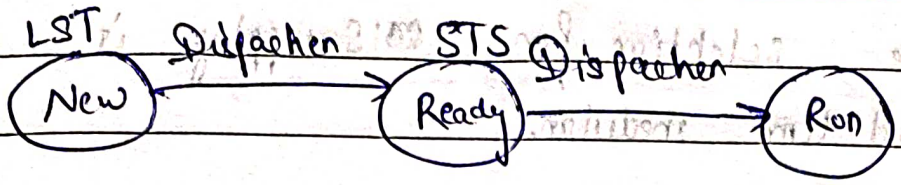
Long term shedular works on new states.



Short - Term Shedular (STS)

Short term shedular works on Redy queue.

It select processes from ready state to running state.

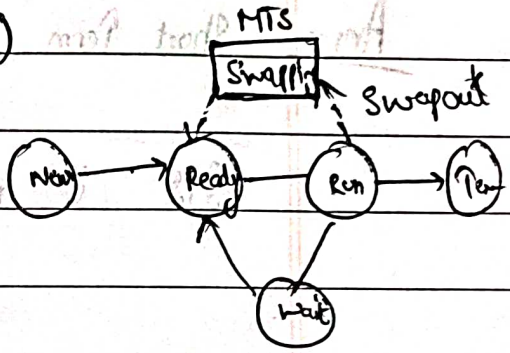


MTS (Mid Term Shedular)

When RAM is fulled

→ Swap proces out from memory (RAM)

→ Swap proces in Memory.



RAM → HDD
HDD → RAM

Def

MTS is used to select the proces which we want to send out from the memory and again brings in the memory.

- sending out of proces from memory is called swap out.
- sending in to the memory is called swapin.

The whole criteria is known as Swapping.

- The selection for swapping is done by mid term shedular.

Q1→ Which scheduler is most frequency invoked?

Ans- Short Term Shedular is most frequently invoked.

STS is also known as CPU Shedular.

Q2→ Which shedular controls the degree of multiprogramming?

Degree of multiprogramming = no. of process in ram.

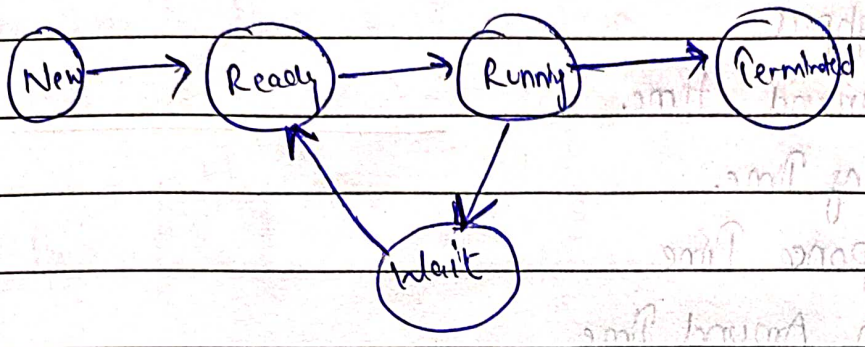
Ans- LONG TERM SHEDULAR.

Because Long Term shedular send the process from HDD to RAM.

STS is called CPU Shedular

PAGE No
DATE
vibhu

CPU Scheduling {STS}



1. Scheduling and performance criteria.
2. Algorithm for cpu scheduling.

Criteria of monitoring a performance.

1. Processor Utilization.
2. Throughput
   ↳ Completion of proces per unit of time.
   - If the ~~exceed~~ larger process at first and smaller at end then the throughput is reduced.
3. Turnaround time.

Interrupts < Maskable / Non-Maskable

PAGE No
DATE

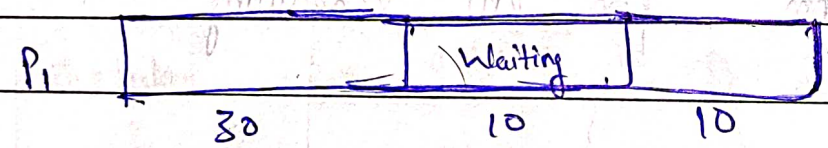
vibhu

Scheduling Criteria

1. Processor utilization.
2. Throughput
3. Turnaround time.
4. Waiting Time.
5. Response Time

Turn Around Time

- How much time a process take to complete its execution



TAT = Burst Time (B.T) + Waiting Time (WT)

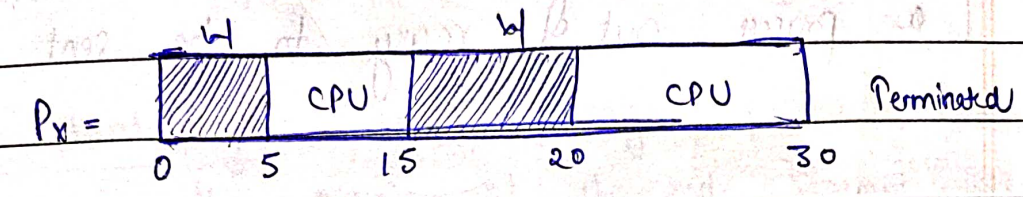
Waiting time.

The time for which process waited for its turn to be executed in CPU.

Response Time

It is a subset of waiting time.

Response Time = first waiting time.



BT (Burst Time) = (15-5) + (30-20)

10 + 10 = 20 units

WT (waiting Time) = (5-0) + (20-15)

5 + 5 = 10 unit.

TAT = BT + WT

10 + 20 = 30 unit

Response Time (RT) = 5 units.

PAGE No
DATE

CPU Scheduling

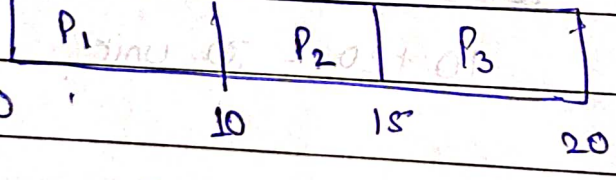
1. FCFS (First Come First Serve Algorithm)

Algorithm is a mechanism for selection of a process out of many to be sent to run state.

The process comes first executed first.

	BT	Wait Time	TAT
P_1	10	0	10
P_2	5	10	15
P_3	5	15	20

Imp



Gantt chart.

Overall Wait Time - $\left(\frac{0+10+15}{3}\right) = \frac{25}{3} = 8.33$

for fcfs.

Wait time = Starting Time - Arrival Time

TAT - { WT + BT
End Time - Arrival Time

PAGE NO
DATE

Total Turn Around Time = 10 + 15 + 20
= 45

FCFS has an advantage - It is very easy to implement.

FCFS disadvantage - If a longer process came, then the process at the end does not get the CPU i.e. waiting time of the another process become so longer.

To verify the gantt chart:

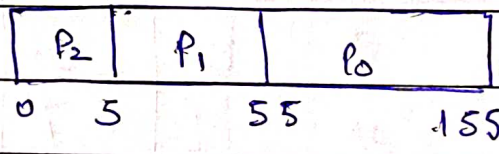
Total Time of chart = Last Time - First Time
= Total Burst Time. = Ideal Portion

Shortest Job First (SJF)

The process which require the minimum amount of time should execute first.

.	B.T.	A.T.	WT	WT (Acc to FCFS)
P_0	100	0	55	0
P_1	50	0	5	100
P_2	5	0	0	150

SJF



SJF → Total Waiting Time = 60

FCFS → Total Waiting Time = 250

Conclusion - In SJF we reduced the waiting time.

Advantage of SJF -

The waiting time is lesser than 1/4 of FCFS

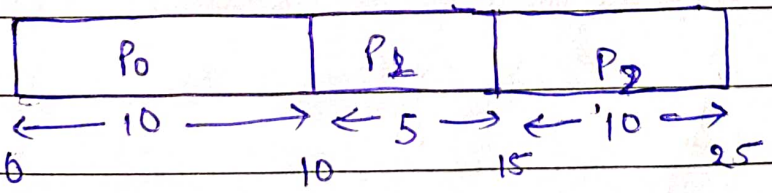
With respect to execution process has 2 variation

* 1) Premmitive → When a new job comes it checks for shortest one
* 2) Non-Premitive → When a job starts it should complete.

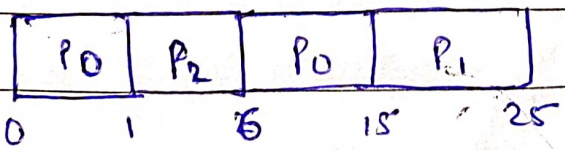
Imp: The use of SJF is not practically possible because no one knows the burst time of the process exact.
We can implement the process using some predictive analysis for finding the burst time.

	B.T.	A.T
P_0	10	0
P_1	10	01
P_2	5	02

SJF - Non Premitive



SJF (Premitive)



$P_0 = 9$
$P_1 = 10$
$P_2 = 5$

PAGE No
DATE

PRIORITY SCHEDULING

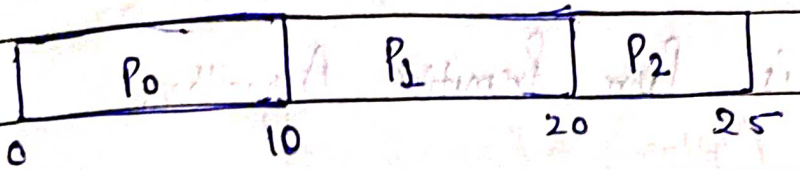
In SJF it might be possible that the smaller job is not so important as the larger one or Shortest Job is dependent on Largest job then SJF algorithm stuck.

Priority Scheduling algo came into the picture.

	B.T.	A.T.	Priority
P_0	10	0	3
P_1	10	1	1
P_2	5	2	2

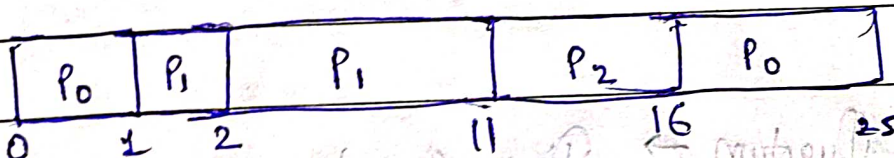
If Nothing mentioned about priority smaller number having higher priority.

Priority (Non-Premithive)



Priority (Premithie)

$P_0 = 9$
$P_1 = 9$
$P_2 = 5$



Advantage :-

- User can provide external priority and on the base of that priority process will get executed.

Disadvantage :-

- Intractivity is not there, (If I want to make time sharing system, using this algo it is not possible).

vibhu
PAGE No
DATE

4) Round Robin

for Time sharing system.

Round Robin is pure Premitive Algorithm.

When process will come according to the time processes will shedule, fixed time Quantum will given to every process.

Time Quantum → Time given to every process for execution.

P No	Burst Time	Arrival Time	Priority	WT
P_1	10	0	1	18
P_2	8	0	4	18
P_3	6	0	2	16
P_4	4	0	3	12

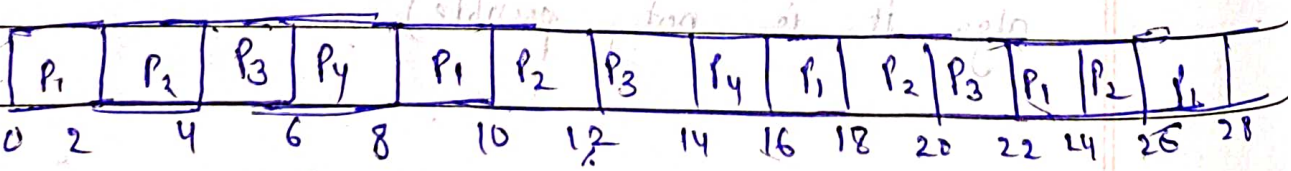
$P_1 = 8$ | 6 | 4 | 2 | 0

$P_2 = 6$ | 4 | 2 | 0

$P_3 = 4$ | 2 | 0

$P_4 = 2$ | 0 | 0

Time Slice = 2



$WT(P_1) = 0 + 6 + (16-10) + (22-18) + (26-24)$

$= 6 + 6 + 4 + 2 = 18$

PAGE NO. vibhm
DATE

$WT(P_2) = 2 + (10-4) + (18-12) + (24-20)$

$= 2 + 6 + 6 + 4$

$= 18$

$WT(P_3) = 4 + (12-6) + (20-14)$

$4 + 6 + 6$

$= 16$

$WT(P_4) = 6 + (14-8)$

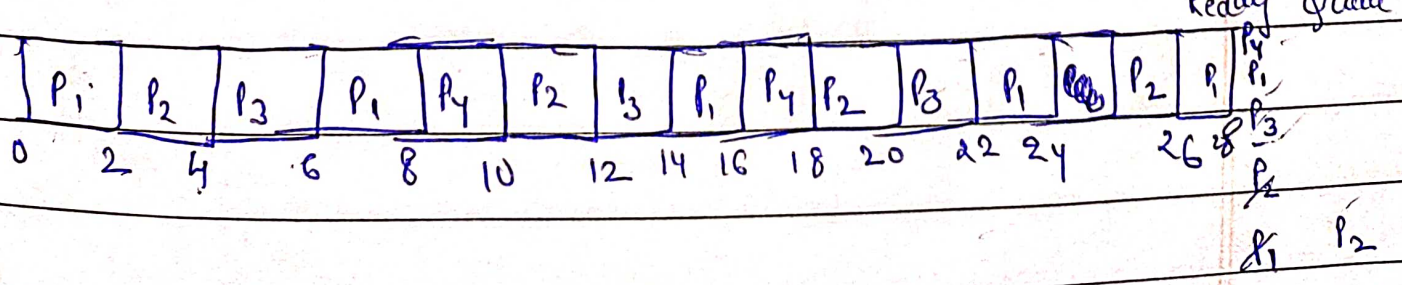
$= 6 + 6$

$= 12$

Find Weight Time

P. No.	Burst Time	Arrival Time	Priority	WT
P_1	~~10~~ ~~8~~ ~~4~~ ~~2~~ 0	0	1	
P_2	~~8~~ ~~6~~ ~~4~~ ~~2~~ 0	1	4	
P_3	~~8~~ ~~4~~ ~~2~~ 0	2	2	
P_4	~~4~~ ~~2~~ 0	3	3	

Time Quantum = 2



FCFS
Round Robin } No starvation

PAGE No
DATE

Advantage -

- It is most important algorithm for user interacting
- This algorithm is free from starvation.

** If any process in any specific circumstances have to wait for infinite time is called starvation

** FCFS suffer from convoy effect.

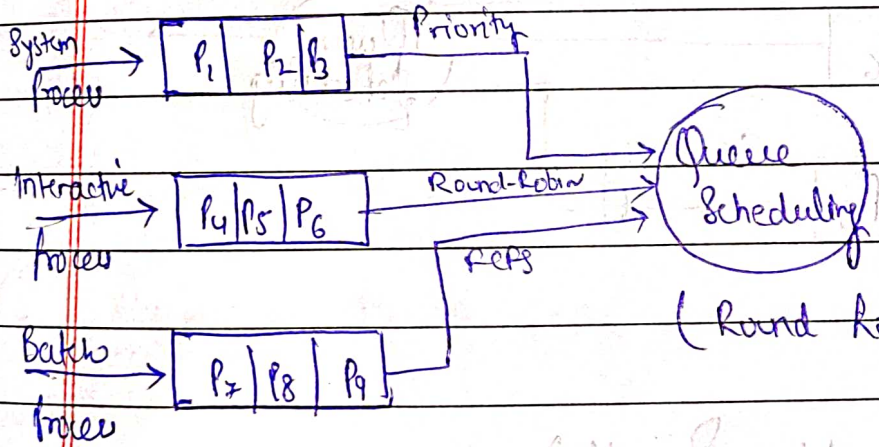
Convoy effect → If higher burst time process arrive first so it delay the last process.

5) Multiple level Queue (MLQ) Scheduling

There are 3 types of processes -

1) System Process → Priority base
2) Interactive Process → Round-Robin
3) Batch Process → FCFS

In MLQ the ready queue is divided into 3 parts for putting different (3) types of process.

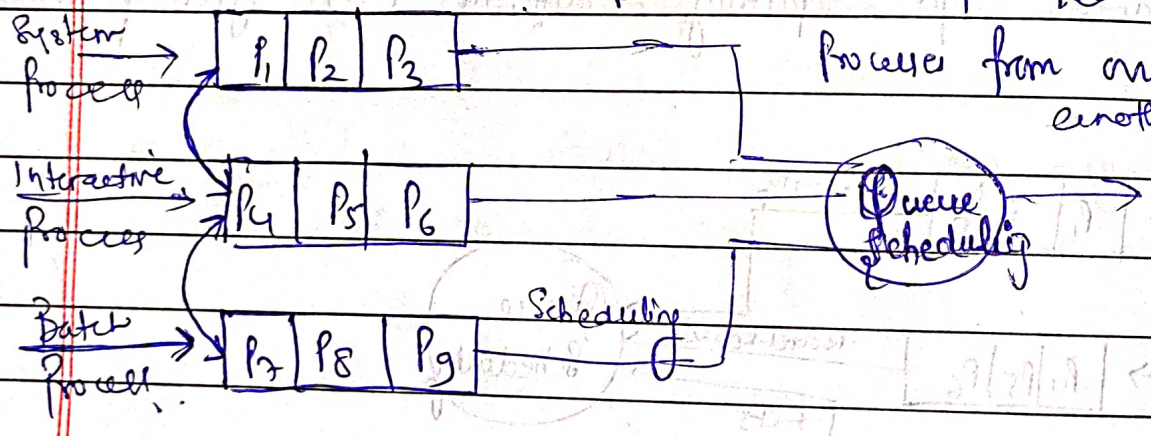


MLQ is 'combination of different fundamental Algorithm.

Dynamic Priority Algorithm → Doesn't have the problem of Starvation

Multilevel Feedback Queue Scheduling (MLFQ)

There are feedback path from one queue to another. If any process has very high priority or would execute first then it pushed to the first queue (or queue in execution).



FCFS - Purely Non Premittie

SJF - } Premittie and Non-Premittie both

Priority - }

Round Robin - Premittie Purely.

Multiprocessing

Multiple process involved.

Load Balancing - Dividing the load into different processors equally.

For achieving Load Balancing we have 2 type of structures.

1) Asymmetric Multiprocessing
2) Symmetric Multiprocessing.

1) Asymmetric Multiprocessing