Process Management - Program under execution is called process.

(i) It should reside in main memory

(ii) It is occupied by the CPU to execution the instruction.

Process will have various attribute -

1. Process Id
2. Process state
3. Program counter
4. Priority
5. General purpose register
6. List of open devices
7. Protection information

Process Id - Process Id is the unique identification number of which is assign by the O.S at the time of process creation. No two process has same process Id.

Process State - It contains the current state of process where it is residing. Process have various state it is residing at a particular time. This information providing by process state.

Program Counter - It contains the address of next instruction to be executed.

Priority - It is the parameter which is assigned by O.S at time of process creation.

General Purpose Register- What are all the register used by the purpose that informati will be maintained in general purpose register.

List to open files - What are all the files used for open by the processes will be maintained in the list of open file attribute of process.

List of Open device - What are all the devices used or open by process that information will be maintained in the list of open devices.

NOTE :- All the attributes of the process is called as the context of process.
The context of the process in process ~~cl~~ control bl

Protection :-

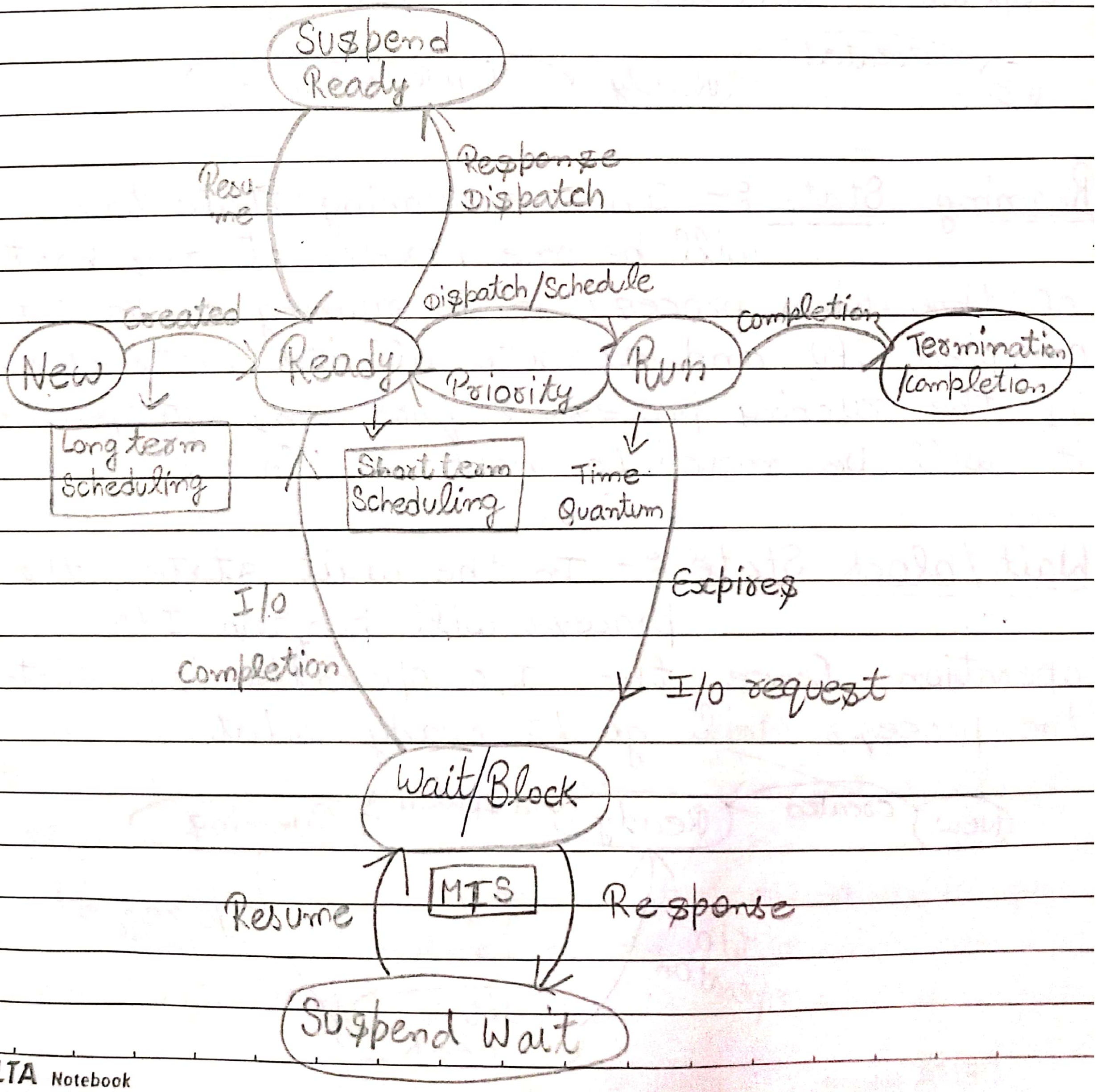
Pid	Process
Program Counter	Priority
GPR	LOF
LOD	Security

Every process will have ~~different state~~ of rotection by the process management ith

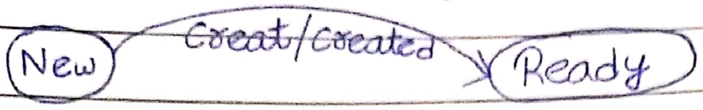
The Process have different state -

New
Ready
Running
Wait/Block
Termination/Completion
Suspend wait

Various operations perform on the process - creation, scheduling, execution, termination, killing, suspending.

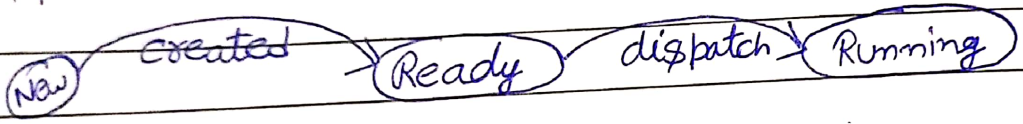


New State :- Initially the process will be in new state it means process is Under creation or being created.



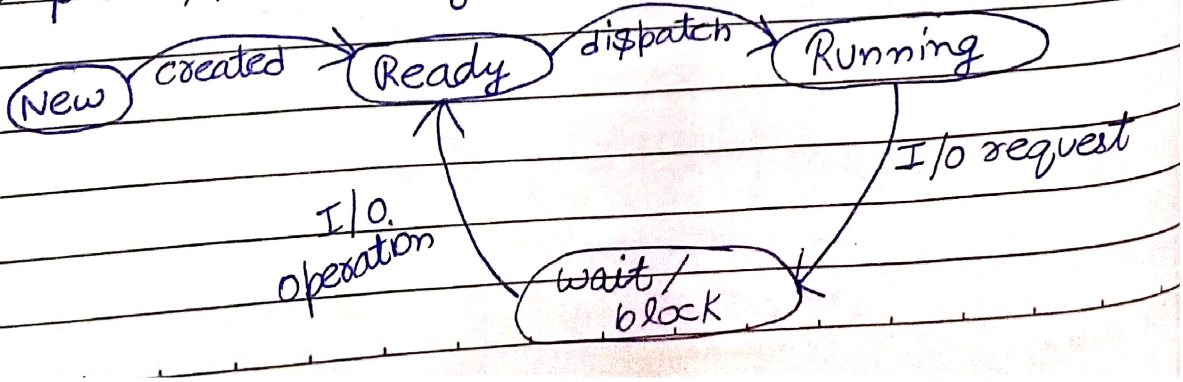
Once the process is ready it will be move to ready state.

Ready State :- In ready state there will be multiple of process. One of the proces is selected from ready state and it will be dispatch or schedule on the running state.

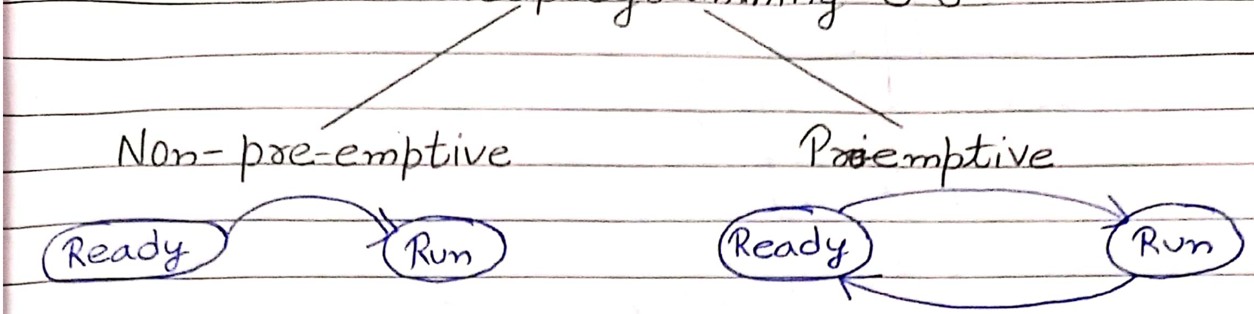


Running State :- In the running state there will be one process at any point of time when process is in running state. It occupied CPU and execute all its instruction. If the running process require any I/O operatio it will be moved to wait or block state.

Wait/block State :- In the wait state, the process will perform I/O operation. Once the I/O operation is complete the process will go to ready state



Multiprogramming O.S



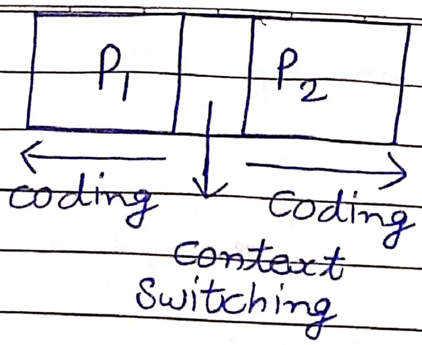
There is a case when main memory is enable to handle more than one process due to shortest of resources in the ready state. Then some of the process may be suspended the ready state and the suspended process will be move to suspend ready state.

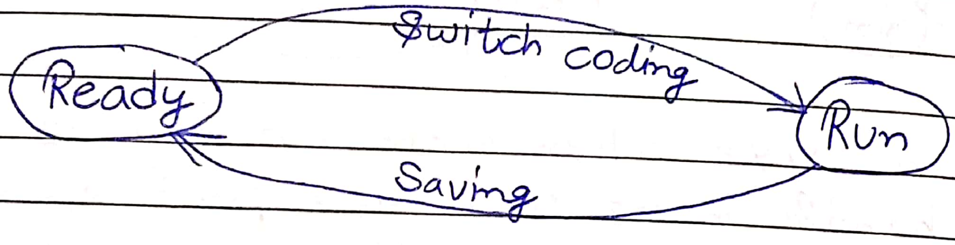
Whenever the resources are sufficient the process will resume back to the ~~th~~ ready state. When the process is in suspend, ready it is residing in the secondary memory.

Whenever the resources are not sufficient to manage the more than one process in the wait state then sub of process will be suspended and suspended the process will remove to suspend ready state. Whenever the resources are sufficient the process will be resumed back to wait state.

Each an every time when the process is moving from one state to another state, the context of the process will change it means context switching will happen.

Saving the context of one process and loading the process of another process is called as context Switching.





In the context of the process is ~~th~~ more than the ~~th~~ context switching time will also increase which is undesirable. The context switching time is considered as over head for the system. The process with respect for the system. The process with respect to there execution time are of two types :-

1. CPU bound process
2. I/O bound process

1. <u>CPU bound Process</u> :- The process which required the more CPU time are called CPU bound process. CPU bound process spend more time in the running state.

2. <u>I/O bound Process</u> :- The process which required more amount of I/O time are called as I/O bound process. I/O bound process will spend more time only in the waiting state.

Degree of Multiprogramming :- The number of process present in the main memory at any point of time is called degeree of multiprogramming.

Scheduler :- In O.S, there are three types of scheduler -

High level

(i) **Long term Scheduler ~~Or~~ or job scheduler -**

It is responsible for creating and bringing the new process into the system.

low level New $\xrightarrow{Created}$ Ready

(ii) **Short term Scheduler or CPU Scheduler -**

It is responsible of selecting one of the process in the ready state for scheduling onto the running State.

(iii) **Mid term Scheduler or Medium Scheduler -**

It is responsible of suspending and resuming the process. The job done by the mid term scheduler is called as swapping.

Dispatcher :-

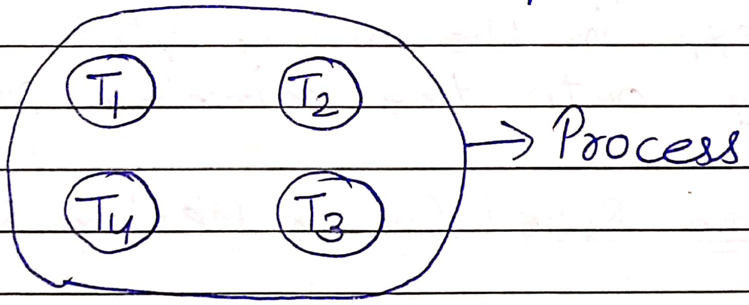
Dispatcher is responsible of saving the context of one process and loading the context of another process. Context switching will be done by dispatcher. The LDS should be select the combination of CPU bound and I/O bound process in order to get good throughput to the system.

If LDS Select only I/O bound process then all are in wait state and CPU become ideal and throughput decrease and ready queue will be always empty.

If LDS select only CPU bound process. CPU utilization will be more waiting queue will always be ~~more~~ empty and throughput decrease So the LDS control the degree of multi-programming

Thread Switch :-

Threads light weight process or less no. of instruction are called threads Thread is like a process having light weight Process is divided into multiple thread.



Big process (means containing more instructions)

Advantages :-

Responsiveness - If the process is divided into multiple threads then if one thread is completed its execution from the output will be response immediately.

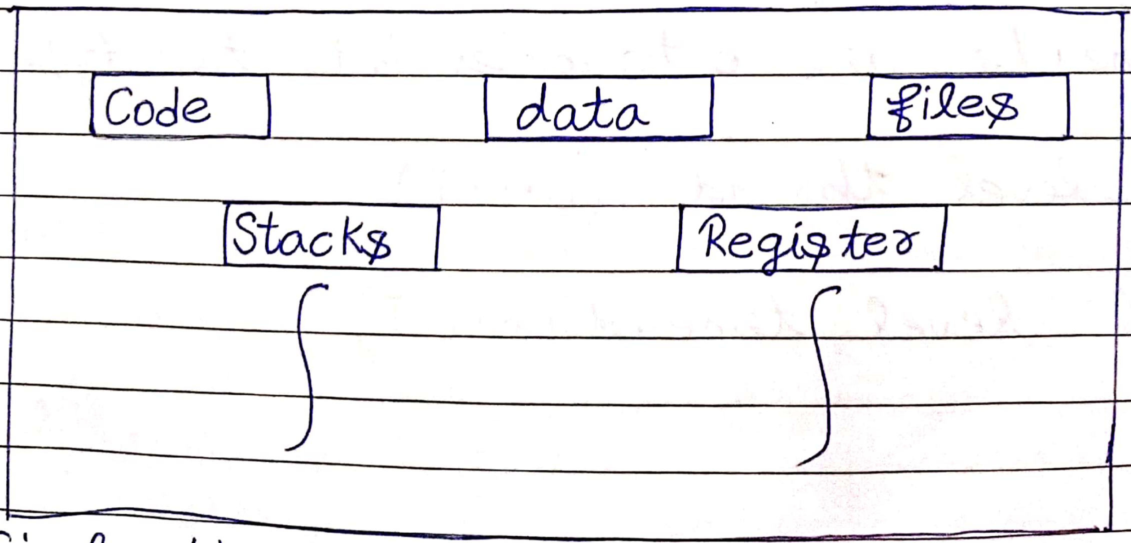
This response will faster compare to the response of process hence the thread will improve the responsiveness.

Faster Context Switching :- The context switching time between the process thread will be very less compare to the context switching time b/w the process because the thread will have less context compare to the context to the process.

Effective Utilization of multiprocessor system :-

If the process is divided into multiple thread then the different thread can be scheduled onto the different processor. So that process execution will be faster.

Resource Sharing :- The resource like a code data, files and memory with in shared among all the thread with in process the stack and register cannot be shared b/w the threads. Every thread will have it stacks and register.



Single thread process

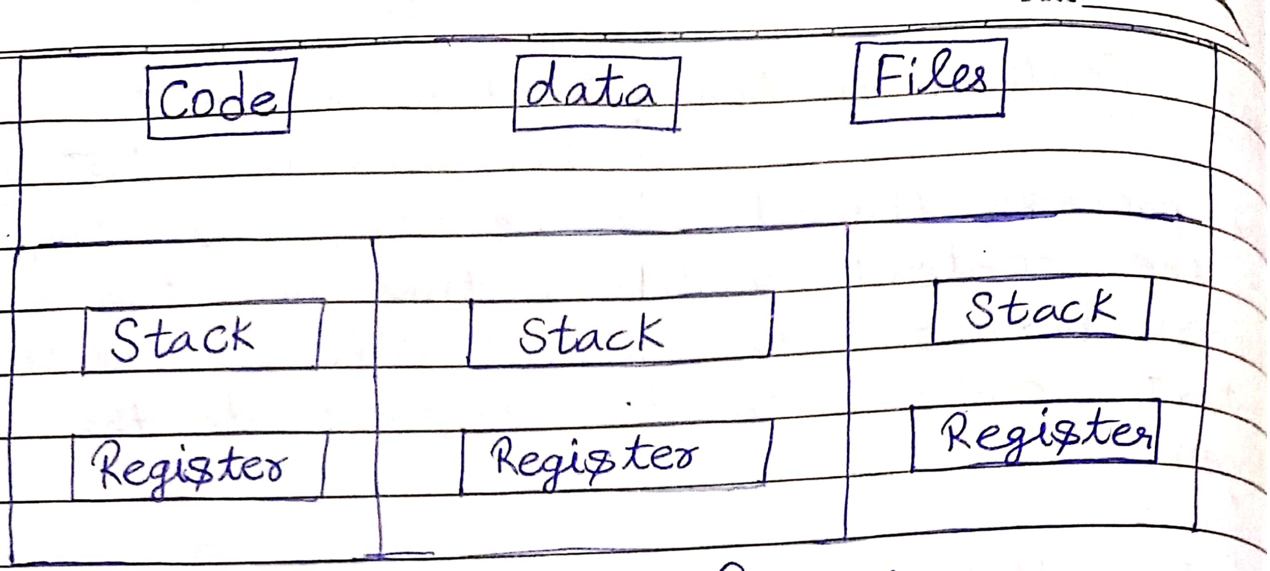


Fig :- Multi-thread Process

Economical :- The implementation threads doesn't required any cost. There are various programming language. API will support implementation of the thread.

Enhance throughtput of System :- If the process divided into multipl thread and if we consider one thread as on job than the no. of jobs completed for unit time will increase and hence the throughput of the system will be enhance.

The threads are categories into two types:

1. User level thread (U.L.T)

2. Kernal level thread (K.L.T)

User level thread	Kernal level thread
1. U.L.T are implemented by user or programmer.	1. K.L.T are implemented by O.S.
2. O.S does not Know about the user level threads. O.S can not recognised the U.L.T and O.S views the user level thread as a process only.	2. K.L.T are recognised by O.S.
3. If One U.l.T is performing blocking system call then the entire process will be blocked.	3. If One K.L.T is performing blocking system call then another threa will continue of executio
4. U.L.T are dependent.	4. K.L.T are independent.
5. The implementation of U.L.T is easy.	5. The implementation of K.L.T is complecated.
6. U.L.T have less context.	6. K.L.T have more contex
7. No hardware support is required for the U.L.T	7. Scheduling of K.L.T is required for hardware support.
8.	8. No.

NOTE :- The I/O of the process is categories into two types :-

1. Synchronous I/O
2. Asynchronous I/O

1. Synchronous I/O :- In synchronous I/O, the process is performing I/O operations will be placed in blocked state till the I/O operation is completed. At the point of time, the I/O operation will be completed and Input service routine will be be initiated which places the process from block state to ready state.

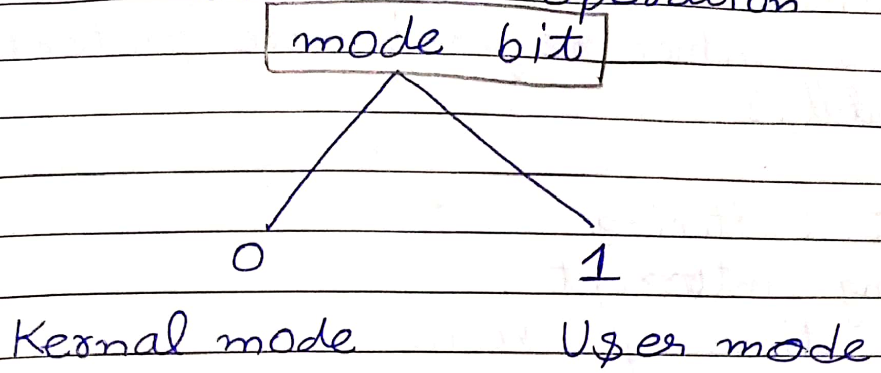
2. Asynchronous I/O :- In asynchronous I/O, by initiating the I/O request, a handler function will be registered. The process is not placed in the blocked state and it will continue to execute to remaining code. After initiate I/O request.

At the point of I/O ~~of~~ request is completed a signal mechanism is used to modify the process that the data is available and the register handler system will be asynchronously invoked. All instruction are not dependent on I/O operation those who are independent continue its execution.

Dual Mode Operation :-

User mode or Non- priviledge mode
Kernal mode or Priviledge mode

Dual mode operation



In the hardware level two different mode are used in order to execute the instruction.

1. User mode
2. Kernal mode.

Depending on the types of instruction the O.S will decide in which particular mode the instruction has to be executed. Generally, the priviledge instruction are executing in the Kernal mode and non-priviledge instruction are executing in the User mode.

Dual mode of operation is required to provide the security and protection to the O.S from the user program and to the O.S from the currupted user, In which particular mode the current instruction is executing will be identifying by using mode bit. At the boot time the system will always start in the Kernal mode. The O.S runs only in the Kernal mode.

Priviledge Mode :- (Some important instructions which required more protection and seculity).

1. Context Switching.
2. Disabiling interrupt.
3. Set the time of clock.
4. Changin the main memory map (changing the process from one main memory location to another).
5. I/o operation (Reading the data from the file of the hard disk).

Non- Priviledge Mode :-

1. Reading the time of the clock.
2. Reading the status of the process.
3. Sending the final ~~step~~ print output to the printer.

CO- Operating Processes :-

A process is said to be co-operating process if it can affect or be affected by the other processes executing in the system. On the other hand, if a process cannot affect or cannot be affected by the other processes in system, it is said to be an independent process.

Any process that shares data with other processes is a co-operating process.

There are several reasons why an environment for co-operating processes are provided :-

(i) Information Sharing :-

Several users may want to share same piece of information.

(ii) Computation speed - up :-

If we want a particular task run faster, we must divide it into sub-tasks, each of which will be executing in parallel with the others.

iii) Modularity :-

We may want to construct the system in a modular fashion, dividing the system functions in to separate processes or threads

(iv) Convenience :-

Even an individual user may have many task to work at one time.

For example :-

A user may be editing, printing and compiling in parallel.

CPU Scheduling Algorithm

Pre-emptive

- SRTF (Shortest Remaining time First)
- LRTF (Longest Remaining time first)
- Round Robin
- Priority based

Non-Pre-emptive

- FCFS
- SJF
- LJF(Longest Job Fir
- HRRN (Highest Response Ratio Next)
- Multilevel Queue

Parameters of CPU Scheduling :—

⇒ **Arrival Time** :— The time at which process enter the Ready Queue or state.

(Duration)
⇒ **Burst Time** :— Time required by a process to get execute on CPU.

⇒ **Completion Time** :— The time at which process complete its execute.

T.A.T (Turn Around Time) :-

Completion Time — Arrival Time

Waiting Time :-

Turn Around Time — Burst Time

Response time :-

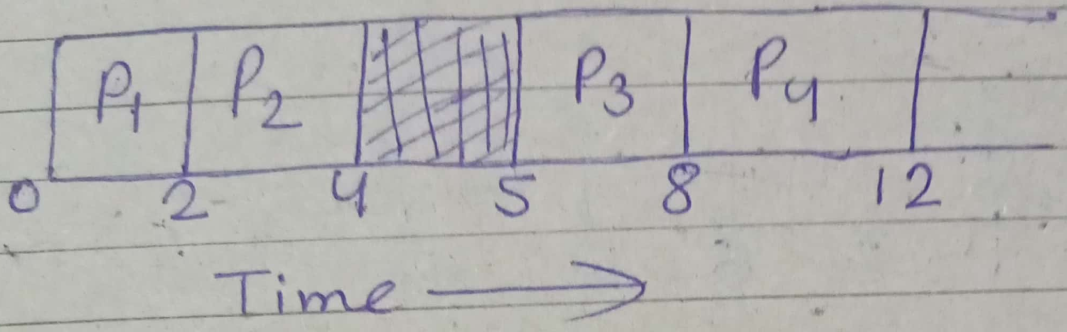
(The time at which a process get CPU first time) — (Arrival time)

☆ FCFS (First come First Serve) :-

R.T	Process No.	Arrival Time	Burst Time	Completion Time	TAT	W.T
0	P_1	0	2	2	2	0
1	P_2	1	2	4	3	1
0	P_3	5	3	8	3	0
2	P_4	6	4	12	6	2
					14	3

Criteria :- "Arrival Time"
Mode :- "Non-pre-emptive"

Giant Chart :



$$\text{Avg. T.A.T} = \frac{14}{4} = \frac{(\text{T.A.T})}{(\text{No. of Process})}$$

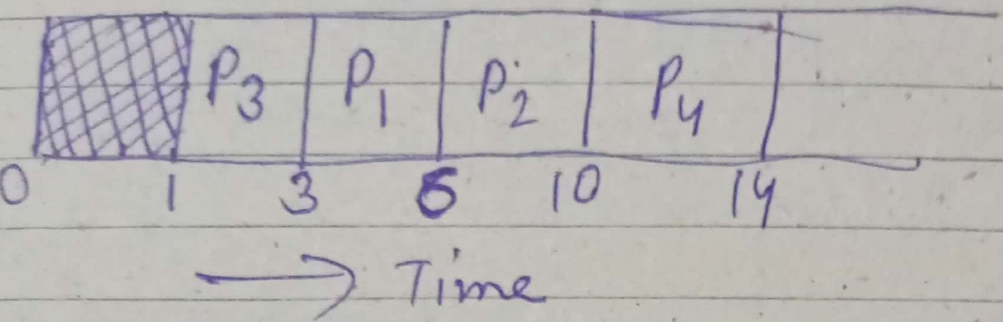
$$\text{Avg. Waiting Time} = \frac{3}{4} = \frac{(\text{Waiting Time})}{\text{No. of Process}}$$

SJF(Shortest Job First)

Criteria = "B.T"
Mode = "Non-Preemptive"

Process No.	A.T	B.T	Completion Time	T.A.T	W.T	R.T
P1	1	3	6	5	2	2
P2	2	4	10	8	4	4
P3	1	2	3	2	2	0
P4	4	4	14	10	6	6
				25	18	

Guantt Chart :



$$\text{Avg. T.A.T} = \frac{25}{4} = 6.25$$

$$\text{Avg. W.T} = \frac{12}{4} = 3.$$

✱ SRTF (Shortest Remaining Time First) Or (SJF with pre-emption)

P. No.	A.T	B.T	C.T	T.A.T	W.T	R.T
P_1	0	~~8~~ ~~4~~ ~~3~~ 2 ~~1~~ 0	9	9	4	0
P_2	1	~~3~~ ~~2~~ ~~1~~ 0	4	3	0	0
P_3	2	~~4~~ 0	13	11	7	7
P_4	4	~~1~~ 0	5	1	0	0
Total				24	14	7

Criteria
Burst Time

Mode = Pre-emption

T.A.T = CT − AT

W.T = TAT − BT

R.T = (CPU first time − AT)

P1	P2	P2	P2	P4	P1	P1	P1	P1	P3

0 1 2 3 4 5 6 7 8 9 13

→ Time

A·vg T·AT $= \frac{\cancel{24}\ 6}{\cancel{6}} = 6$, Avg. W·T $= \frac{\cancel{11}}{\cancel{4}} = 2.75$

Avg R·T $= \frac{7}{\cancel{4}} = 1.75$

☆ Round Robin ←

P.No.	A·T	B·T	C·T	T·AT	W·T	R·T
P1	0	~~5~~ ~~3~~ ~~1~~ 0	12	12	7	0
P2	1	~~4~~ 2 0	11	10	6	1
P3	2	~~2~~ 0	6	4	2	2
P4	4	~~1~~ 0	9	5	4	4

Criteria

~~Time~~-quantum

Mode

Pre-emptive

TAT = C·T − AT

W·T = T·AT − B·T

R·T = {CPU first time − AT

→ Context Switching means save the running process & bring new one Ready

Ready ⇄ Running

Given T.Q = 2

Ready Queue

~~P1~~	~~P2~~	~~P3~~	~~P1~~	~~P4~~	~~P2~~	~~P1~~

0

context switching

Running Queue (Gantt chart)

P1	P2	P3	P1	P4	P2	P1

0 2 4 6 8 9 11 12

→ Time

Context means PCB (Process control block)

No. of context switching ⇒ 6

Priority Scheduling Algorithm (Pre-emptive mode)

~~P. No~~

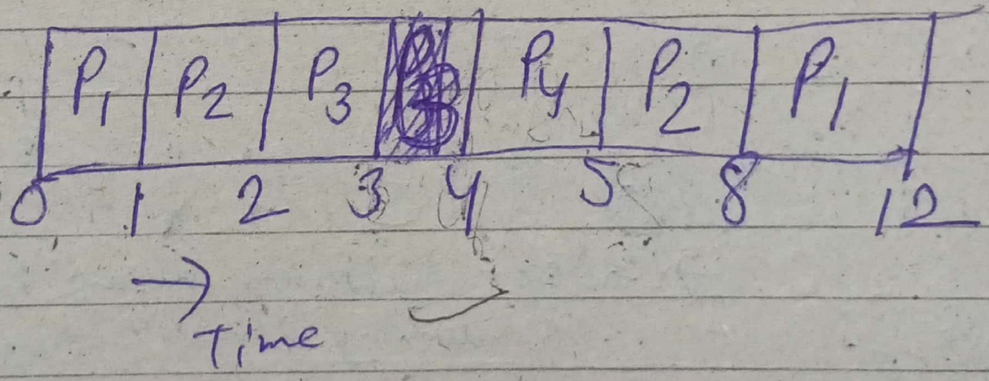
Priority	P.No	A.T	B.T	C.T	T.A.T	W.T
10	P_1	0	~~5~~ ~~4~~ 0	12	12	7
20	P_2	1	~~4~~ ~~3~~ 0	8	7	3
30	P_3	2	~~2~~ ~~1~~ 0	4	2	0
40	P_4	4	~~1~~ 0	5	1	0
Total					22	10

Criteria
"Priority"

Mode
"Pre-emptive"

T.A.T ⇒ C.T − A.T
W.T ⇒ T.A.T − B.T

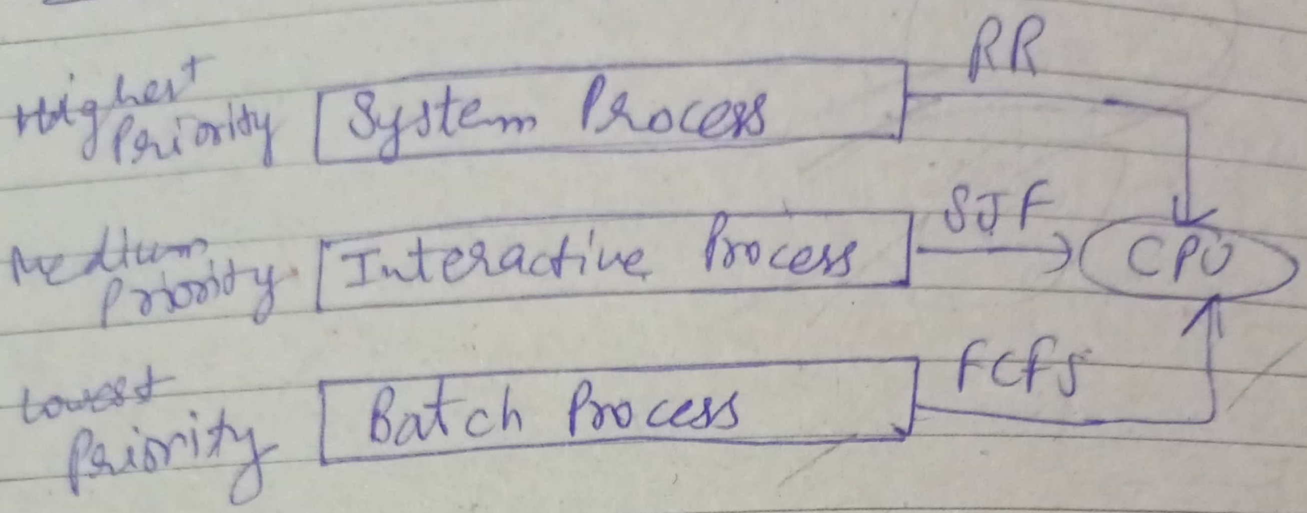
Higher the no. Higher the priority
Ex:- 40



$$\text{A.Vg A.T} = \frac{22}{4} = 5.5$$

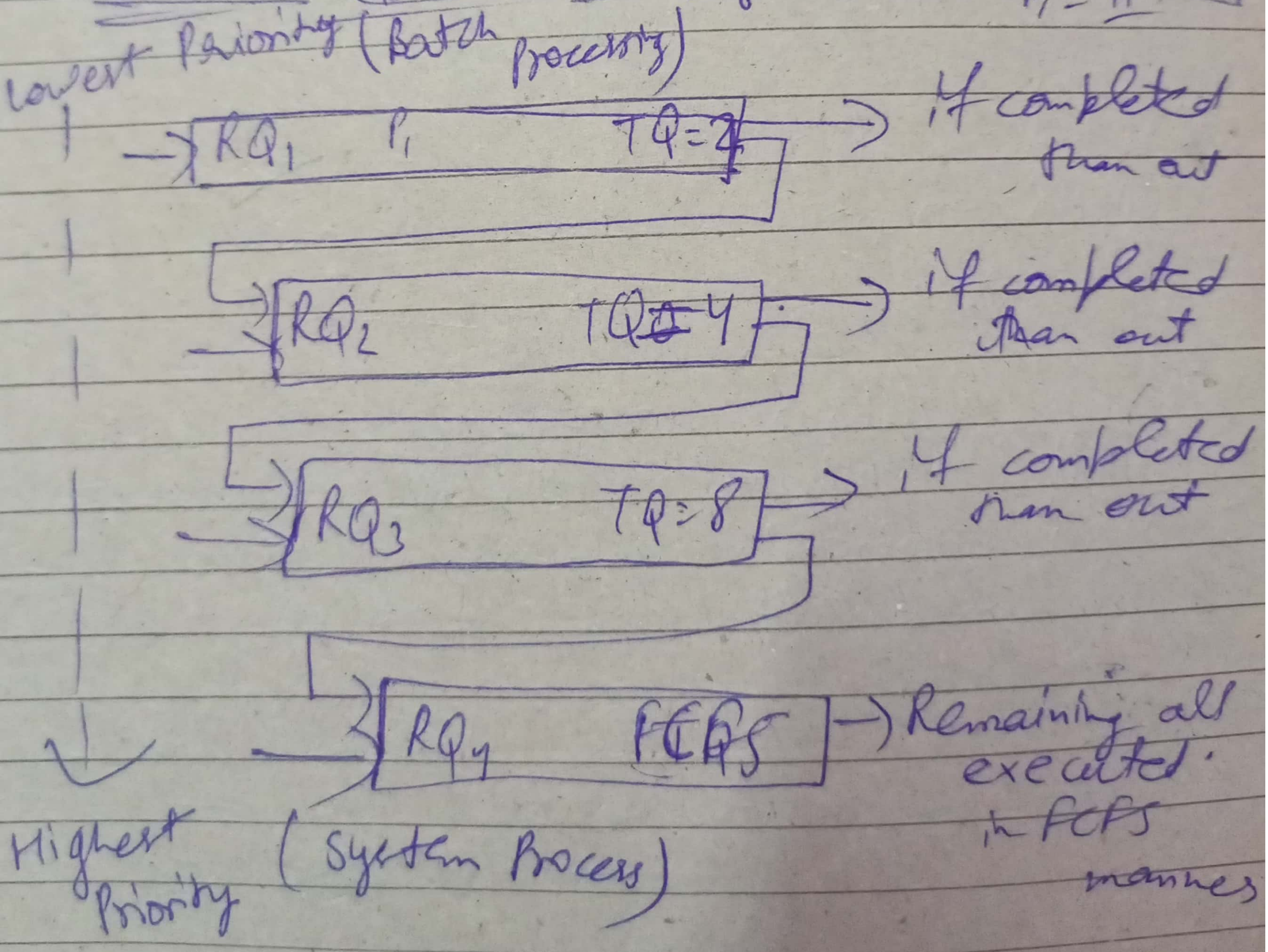
$$\text{Avg W.T} = \frac{10}{4} = 2.5$$

* Multilevel Queue Scheduling :-



* Multilevel Queue with feedback Scheduling

$P_1 = 19$


Lowest Priority (Batch processing)
RQ1
P1
TQ=2
If completed than out
RQ2
TQ=4
if completed than out
RQ3
TQ=8
If completed than out
RQ4
FCFS
Remaining all executed in FCFS manner
Highest Priority (System Process)