Lecture : 1

Introduction to OS

Operating system

* It is an interface between user and hardware
* Resource Allocator
* Manager → Memory, Processer, files, security etc.

Goals of OS:
Primary: Convinience
Secondary: efficiency

Type of OS :→
1. Batch OS
2. Multi-programming OS
3. Multi-Tasking OS
4. Multi-Processing OS
5. Real time OS

Batch OS:
- Only one job is running at a time.
- Until it is not finished completely and generate its output, all the remaining jobs has to wait.
- CPU may be ideal for a long time whether CPU is free and job is using any input/output device.
- starvation is created by jobs
- It is not efficient
- It is not interactive.

Ex. J_1, J_2, J_3 → Computer → output of J_1

When J_1 is finished and generated its output, after that job J_2 will be started and so on.

Multi - Programming OS:
- It is an Extension to the batch operating system.
- When a job goes to I/O device and CPU is free then other job is allocated to CPU.
- In that sense CPU is not ideal and multiple jobs may ruen at a time.
- Efficient CPU utilization.
- CPU will be busy all the time.

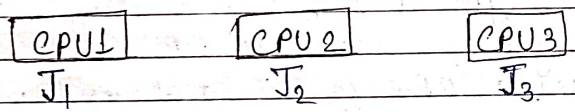
Multi - Tasking OS

- Extension to multiprogramming.
- Let CPU has a 4 Jobs J_1, J_2, J_3 and J_4. Idence CPU will execute J_1, for some time, then J_2 for some time, then J_3 for some time and then J_4 for some time.
That means CPU will perform multiplexing amo all the jobs without even completing any job.

- In that way interactiveness can be improved.
- In multi tasking preemption may takes place but in Batch OS and multi-programming OS there is no preemption.
- Multi-Tasking and multi-Programming both are nearly same.
- Here also CPU is not ideal.

Multi-Processing OS:

Instead of having only one CPU it has more than one CPUs.



- Many jobs can be handled simultaneously i.e. Parallelism is included in multi programming.
- Throughput (no. of jobs running per unit time) can be improved. Reliability is also increased i.e. if one CPU fails then jobs can be allocated to other CPUs.

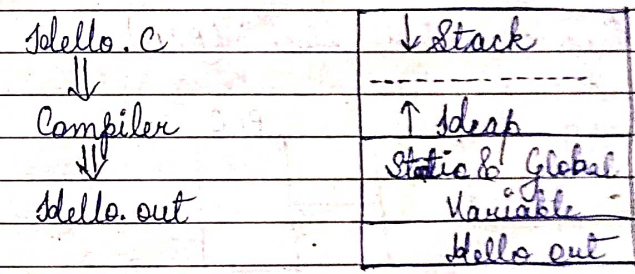
Real time OS:-

* Here we will be given some jobs and jobs are having some deadlines.
* And you have to finish the job in given deadline.
* Here timing is everything.

Ex: In testing of Missiles Exact timing is needed. In that case real time OS is best.

Lecture - 2
Process Management

Process:
Process is real entity and program is vertical entity.
Ex. Body is process and soul is program.



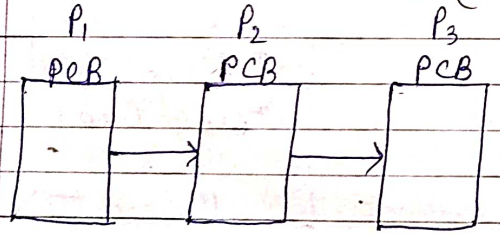
Attributes of Process:

1. Process Id.
2. Program counter
3. Process state
4. Priority
5. General purpose Registers.
6. List of open files

7. List of open devices
8. Protection.

Program Counter:
Program counter keeps the record of next instruction that has to be executed. Let a process P_1 contains 10 instructions $I_1, I_2, I_3, \ldots I_9, I_{10}$. After executing instruction I_6 P_1 is preempted and CPU is allocated to process P_2 then program counter of process P_1 keeps the record of how many instruction has been executed and which is next instruction to execute.

Process control Block: (PCB)



- For Every process, there is one PCB created.
- In PCB the entire information of process will be present.
- For every process there will be only one PCB.
- All the PCB are connected to each other in linked list Manner.
- So by looking at the list of PCB we can know how many processes are there in the system and which one is doing what by looking at the state of process.
- To know the information about a process, look at only their PCB and to know the information about the entire processer, look at the list.
- Everything is maintained by O.S in a linked list.

Lecture-3
Process states

Context:→
- Everything about a process is context
- That means, process states, program counter, general purpose registers everything is called context.
- PCB works of all as context.

States of a process
1. New
2. Ready
3. Run
4. Block or wait
5. Termination or completion
6. Suspend Ready
7. Suspend wait for or suspend block

1. New: (Secondary Memory)
- When a process is about to be created, it will be in the secondary memory and whenever we are going to pickup a process from secondary memory it will be called as new.
- If a process is about to be created, it is a program and if it is created, it is called as a process.

2. Ready: (Main Memory)
- Whenever a process is newly created that will be ready to run.
- Whenever a process is created that will be directly goes in to the ready state.
- In the ready state process is in the main Memory.
- And in the new state process/program is in the secondary memory.
- In main Memory there can be many processes which are ready to run and it is up to you which one you are going to choose and Execute.

3. Run: (Main Memory)
- From number of processes in the ready state one of them is choosen and given to the CPU to be Execute.
- If you have only one CPU then at a time only one process can be running.
- When a process is running it will be in the main memory.

4. Block or wait (Main Memory)
When a process is running sometimes it needs to read a file or write a file i.e at that time it does not need only help from CPU. It has to simply read or write has to simply read or write ~~also~~ then process is pull it out from the CPU and put it in the Block state or wait state, untill it finishes the i/o it will be again bring to the ready state

5. Termination or completion:
- Once a process in running state completed all its job, then it is terminated (like killing).
- Even its context will be deleted.
- Once a process is completed and terminated then its PCB is also deleted.
- Termination or completion is going to be happen after everything is over and entry trace about the process will be deleted.

6. Suspend Ready: (Secondary memory)
Let main memory is able to hold 10 processes

and all the IO processes are there suddenly a new process come into existence and its priority is very very high compared to all other IO existing processes then we should definately move one of the IO processes from main memory to secondary memory to make the room for new one it is suspend ready

7. Suspend wait or suspend block : SM)
1. It is similar to the suspend ready.
2. But instead of suspending the processes which are ready, it is better that you suspend the processes which are blocked.
8. if a process is waiting for i/o rather than CPU then better is to wait in secondary memory.
4. If you send a process into secondary memory which is already in block or wait state then it is called as suspend wait or suspend block.

Important note:

1. If a process is in suspend wait state/ suspend block state and suddenly it has finished its I/O then it could make a transaction to the suspend ready.

2. If a process is in the suspend wait state and then it could be back to the main memory then it will come back to block or wait state
3. If a process is in suspend ready state and if the memory is available then it will come back to ready state

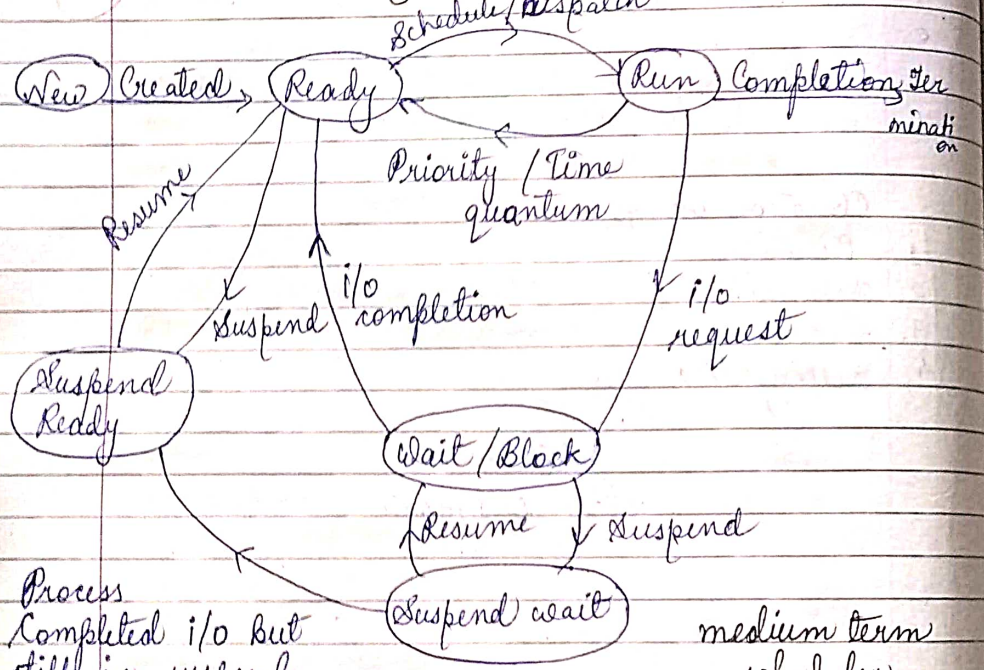
Operation on process
1. Creation
2. Scheduling
3. Executing
4. Killing / Delete

~~Lecture 4~~
~~State Diagram~~

Lecture → 4

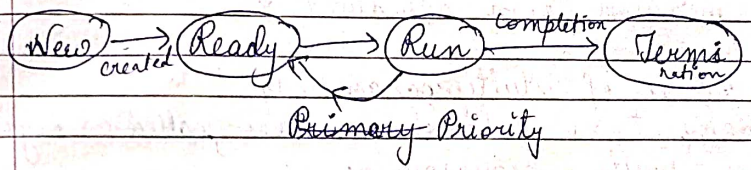
State Diagram

State Diagram of a Process.



Note:

- At any time many no. of processes may be in ready state.
- Depending upon the resource it is decided that how many processes can be in the ready state.
- Minimum number of states a process can be go though is 4. i.e. new, ready, run termination



- If a process is having lots of CPU time, it is called CPU bound and if it is having lots of i/o time, it is called i/o bound.

④ Types of Scheduler :→

1. Long term scheduler
2. Short term scheduler
3. Medium term scheduler.

Long term scheduler :→ It makes the decision about how many processer should be made.

Short term scheduler :→ (Dispatcher) choosing a process from ready state for execution is a short term decision and this decision is taken by short term scheduler or dispatcher.

Medium term Scheduler :→ Suspension decisions, are made by medium term scheduler because

once a process is suspended it is not going to there for a very long time or very short time that's why it is called as medium term scheduler.

Degree of Multiprogramming:

- Long term Scheduler is also called as degree of multi - programming
- Degree of multi - programming is the maximum number of processes that can be present in the ready state.
- The main decision is taken by long term scheduler.
- Long term scheduler is going to affect the performance of the system.
- Short term Scheduler will decide which process to be executed next and then after deciding, it will call dispatcher.

Hence dispatcher is the software which is responsible for moving a process from run to ready state and from ready to run state.

Therefore whenever you try to change any process from the CPU you will first call the short term scheduler which will choose a process to be scheduled and then short term scheduler will call the dispatcher and dispatcher will do the remaining work

i.e. it will pull out one process and put other process back. It is called as context switching.

- Context switching means, it will completely change the status of CPU with a new process by changing old process.
- Context switching is saving the context and updating the context.
- Time taken to do this context switching depends on the size of the context.
- So context size should be as minimum as possible so that context switching time to be very less

Swapping :-

- Swapping is moving a process from main memory to secondary memory and then getting it back.
- So we need an efficient medium term scheduler so that swapping is going to be less because swapping is going to take considerable amount of time.

Note :-

1. Long Term Scheduler is related to overall perform

(Degree of multiprogramming)

2. Short Term Scheduler :→ is related to context switching.
3. Medium Term Scheduler :→ is related to swapping

Lecture - 5
Question on Process state

Q-) Consider a system with 'N' CPU processors and 'M' processes. Then how many processes can be present in ready state, in running state, in block state at minimum and at maximum?

Sol :-

	Min	Max
Ready	0	M
Running	0	N
Block	0	M

Lecture - 6
Various Times related to Processes

Important Parameters of a Process :→

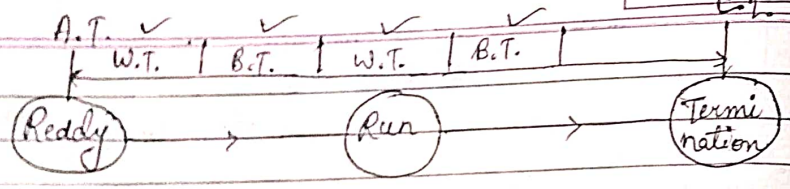
1. Arrival time
2. Burst time
3. Completion time
4. Turn arround time
5. Waiting time
6. Responsed time

1. Arrival Time :
- Whenever any process gets into ready queue, that time is arrival time.
- Arrival Time is the time at which a process gets ready to be executed.

2. Burst Time :
- The amount of time required by a process to finish.

3. Completion Time :
- The time at which a process finishes its execution

$$CT - AT = BT + WT$$



4. Turn Around Time:
   - Difference between completion time and arrival time.

$$T.A.T. = C.T. - A.T.$$

5. Waiting Time:

$$W.T. = T.A.T. - B.T.$$

6. Response Time:
   It is the first time at which a process get processed after its arrival.

Lecture - 7
CPU Scheduling

CPU scheduling: Picking up a process from a ready state and giving it to CPU is called as CPU scheduling.

- Who will do it → Short term scheduler along with Dispatcher.
- Where → Ready state to running state.
- When → when a process moves from
  1) Run → Termination
     Run → Wait
     Run → Ready.
  2) New → Ready i.e. when a new process is just created.
  3) Wait → Ready

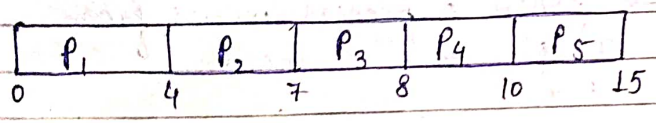
Note: In point 2 and point 3, CPU scheduling is only needed, if higher priority process come into existance otherwise CPU scheduling will not be performed.

Lecture - 8
First come First serve.

Que. Criteria : Arrival time
Mode : Non-preemptive

P.No.	A.T.	B.T.
1.	0	4
2.	1	3
3.	2	1
4.	3	2
5.	4	5

Sol:- Gantt chart:



P. No.	A.T.	B.T.	C.T.	T.A.T.	W.T.	R.T.
1	0	4	4	4	0	0
2	1	3	7	6	3	3
3	2	1	8	6	5	5
4	3	2	10	7	5	5
5	4	5	15	11	6	6

Property of FCFS:
It has convay Effect (Disadvantage)

Lecture - 9

Convay Effect:→

Convay Effect:→ • It is the disadvantage of FCFS.

Ex:→ if a VIP is passing through a road function, then every one is blocked and has to wait. It is convey effect.

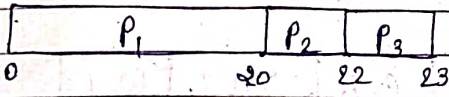
• Similarly in FCFS when a process is having a long Burst time then all the processes which are after it are going to stop (wait).

That is also called as starvation or convay effect.

Ex:→

P. No.	A.T.	B.T
1	0	20
2	1	2
3	1	1

its Gantt chart is:



P. No.	A.T.	B.T.	C.T.	T.A.T.	W.T.
1	0	20	20	20	0
2	1	2	22	21	19
3	1	1	23	22	21

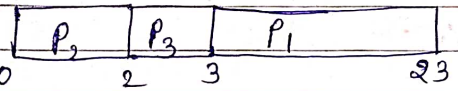
Now Average waiting time = $\frac{0+19+21}{3}$

Average W.T. = $\frac{40}{3}$

Ex:→2. If the same processes arrives in following manner then -

P. No.	A.T.	B.T.
1	1	20
2	0	2
3	0	1

then its Gantt chart will be as-



P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T
1	1	20	23	22	2
2	0	2	2	2	0
3	0	1	3	3	2

Average waiting time = $\frac{4}{3}$

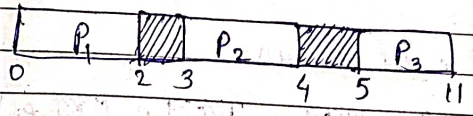
Lecture - 10.

FCFS Example

i.→

P.No.	A.T.	B.T.
10	0	2
2	3	1
3	5	6

→ its Gantt chart will be as:



P.No	A.T.	B.T.	C.T.	T.A.T.	W.T
10	0	2	2	2	0
2	3	1	4	1	0
3	5	6	11	6	0

How to Implement FCFS:

The simplest data structure to implement FCFS is Queue.

time complexity = O(n)

where n is the number of processes.

Lecture - 11.

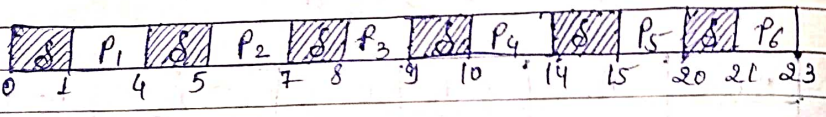
FCFS with Overhead

δ = 1 unit

Ex:-

P. No.	A.T.	B.T.
1	0	3
2	1	2
3	2	1
4	3	4
5	4	5
6	5	2

Soln:- Its Grantt short will be as -



Hence total schedule length = 23
and total useless time = 6 units.

$$\text{inefficiency} = \frac{6}{23} \times 100$$

$$= 26.07\%$$

$$\text{and efficiency} = \left(1 - \frac{6}{23}\right) \times 100$$

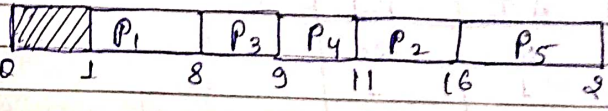
$$= 73.91\%$$

Lecture - 12
Shortest Job First

Criteria : Burst time
Mode : Non-preemptive

P.No.	A.T.	B.T.
1	1	7
2	2	5
3	3	1
4	4	2
5	5	8

Soln:- its Grantt chart will be as -



P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.
1	1	7	8	7	0
2	2	5	16	14	9
3	3	1	9	6	5
4	4	2	11	7	5
5	5	8	24	19	11

- Min Ideap is the best suitable data structure to implement shortest job first scheduling algorithm.
- Time complexity = $n \log n$

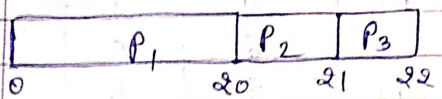
Lecture - 13
Analysis of SJF

It is not mensioned in any text book that SJF is having any convoy effect but with the following example it is very clear that SJF is also having convoy effect.

Ex.

P.No.	A.T.	B.T.
1	0	20
2	1	1
3	2	1

Sol:→ its Grantt chart will be as-



P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.
1	0	20	20	20	0
2	1	1	21	20	19
3	2	1	22	20	19

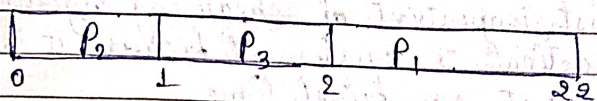
its average waiting time = $\frac{0+19+19}{3}$

$= \frac{38}{3}$

Ex:→ The some above example in a different way-

P.No.	A.T.	B.T.
1	2	20
2	0	1
3	0	1

its Grantt chart will be as-



P.No.	A.T.	B.T.	C.T.	T.R.T.	W.T.
1	2	20	22	20	0
2	0	1	1	1	0
3	10	1	2	1	0

here average waiting time = $\frac{0+0+0}{3} = 0$

Note: Completion time will always be greater than arrival time and burst time.
It is practically impossible to implement SJF.

In the above example.

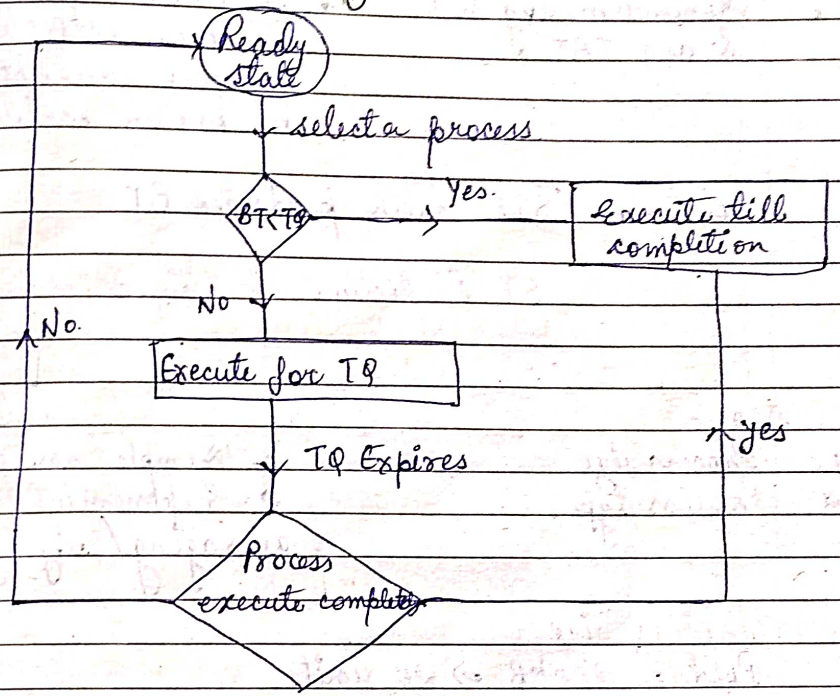
Throughput = $\frac{\text{no. of process}}{\text{total execution time}} = \frac{3}{22}$

Lecture → 14.

Round Robin Algorithm

- It is most important of scheduling algorithm.
- It is practical to implement because it is not dependent on burst time
- It does not require lots of data structure, like heap and all
- Using a Queue we can implement it
- There is No starvation here.
- It will take a process for execution and it will not execute it till completion or till next process arrives nothing like that.
- Whatever a process, it is going to execute it only for a some particular amount of time that is called time quantum.
- And after the execution of a process for a time quantum, it will be preempted and next process will get executed for some time quantum.
- So every process will get executed only for a small amount of time i.e. time quantum.
- Therefore no process is going to wait forever for the CPU.
- Therefore it has very less starvation problem.

Round Robin Algorithm:



Lecture → 15

SJF with Prediction of BT

SJF

Advantages
- Maximum throughput

Disadvantages
- Starvation to longer Jobs

Minimum avg WT & avg TAT

- It is not implementable because BT of a process can not be known earlier.

Solution :- SJF with predicted BT

BT Prediction Technique

Static
1. Process size
2. Process type

Dynamic
1. Simple averaging
2. Exponential averaging / Aging

Process size:
Pold = 200 KB ⇒ 20 units
∴ Pnew = 201 KB ⇒ 20 units

Process Type:

Process

Operating System Process

User Process

OS Process (3-5 units)

User process

Interactive (5-8 unit)

Foreground (10-15 unit)

Background (15-20 unit)

Simple Average Technique:

- Given n processes ($P_1, P_2, \ldots P_n$)
- Let t_i be the actual BT
- Let T_p denote the predicted B.T.

$$T_{n+1} = \frac{1}{n} \sum_{i=1}^{n} t_i$$

Exponential Average / Aging

$$T_{n+1} = \alpha t_n + (1-\alpha) T_n \quad \text{---} (I)$$

here $0 \le \alpha \le 1$
α : smoothing Factor

$$T_n = \alpha t_{n-1} + (1-\alpha) T_{n-1} \quad \text{---} (II)$$

$\alpha = 0.5$, $T_1 = 10$

Actual BT $(t_1, t_2, t_3, t_4) = (4, 8, 6, 7)$ then
$T_5 = ?$

Soln:→ $T_2 = \alpha t_1 + (1-\alpha) T_1$

$T_2 = 0.5(4) + (1-0.5)10$
$= 7$

$T_3 = \alpha t_2 + (1-\alpha).T_2$
$= 0.5(8) + (1-0.5)7$
$= 7.5$

$T_4 = \alpha t_3 + (1-\alpha) T_3$
$= 0.5(6) + (1-0.5)7.5$
$= 6.75$

$T_5 = \alpha t_4 + (1-\alpha) T_4$
$= 0.5(7) + (1-0.5)6.75$
$= 6.875$ Ans

Lecture - 16
Round Robin Algorithm 1

Ex:→

P.No.	A.T.	B.T.
1	0	4
2	1	5
3	2	2
4	3	1
5	4	6
6	6	3

Time Quantum = 2

Criteria : TQ + AT

Soln:→ its Grantt chart will be as -

P_1	P_2	P_3	P_1	P_4	P_5	P_2	P_6	P_5	P_2	P_6	P_5
0	2	4	6	8	9	11	13	15	17	18	19	21

Queue: P_1 P_2 P_3 P_1 P_4 P_5 P_2 P_6 P_5 P_2 P_6 P_5

P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.
1	0	4	8	8	4
2	1	5	18	17	12
3	2	2	6	4	2
4	3	1	9	6	5
5	4	6	21	17	11
6	6	3	19	13	10

- If time Quantum is large then contex switches will be very less and if time Quantum is small then there will be more

context switches.

Ex: Solve above example with time quantum (TQ) = 4.

Soln: Queue: P1 P2 P3 P4 P5 P6 P2 P5

Gantt Chart:

P1	P2	P3	P4	P5	P6	P2	P5
0	4	8	10	11	15	18	19	21

P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.
1	0	4	4	4	0
2	1	5	19	18	13
3	2	2	10	8	6
4	3	1	11	8	7
5	4	6	21	17	11
6	6	3	18	12	9

Lecture -17
RR Example 2

Que:

P.No.	A.T.	B.T.
1	5	5
2	4	6
3	3	7
4	1	9
5	2	2
6	6	3

Time Quantum = 3

Soln: Queue: P4 P5 P3 P2 P4 P1 P6 P3 P2 P4 P1 P3

Gantt chart:

P4	P5	P3	P2	P4	P1	P6	P3	P2	P4	P1	P3
0	1	4	6	9	12	15	18	21	24	27	30	32	33

P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.	R.T.
1	5	5	32	27	22	10
2	4	6	27	23	17	5
3	3	7	33	30	23	3
4	1	9	30	29	20	0
5	2	2	6	4	2	2
6	6	3	21	15	12	12

Note:

TQ	CS	RT	TQ	CS	RT
↑	↓	↑	↓	↑	↓

Note: if TQ = ∞ then RR ⟹ FCFS

Lecture - 18
RR Example 3

Ques. Consider Four jobs P_1, P_2, P_3 and P_4 arriving in ready queue in the same order at time $t=0$. If Burst time requirements of these jobs are 4,1,8,1 respectively then what will be the completion time of P_1, assuming Round Robin with TQ = 1.

Solu:→

P.No.	A.T.	B.T.
P_1	0	4
P_2	0	1
P_3	0	8
P_4	0	1

Time Quantum
TQ = 1

Queue: $P_1\ P_2\ P_3\ P_4\ P_1\ P_3\ P_1\ P_3\ P_1\ P_3\ P_3\ P_3\ P_3$

Gantt chart:

P_1	P_2	P_3	P_4	P_1	P_3	P_1	P_3	P_1	P_3	P_3	P_3	P_3

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

completion time of P_1 = 9

Lecture - 19
RR Example - 4

Q-) Consider n processes sharing the CPU in RR fashion. If the context switching time is s units. what must be the time quantum 'q', such that the number of context switches are reduced, but at the same time each process is guaranteed to get the turn at the CPU for every t seconds.

Solu:→ Let the same scenario for only 4 processes, then —

$P_1 \| P_2 \| P_3 \| P_4 \| P_1$
5 5 5 5

total context switching time = 4s
and total quantum time = 3q

Hence For 'n' processes

$$n(s) + (n-1)q \le t$$

$$\boxed{q = \frac{t - ns}{(n-1)}}$$ Ans

Lecture - 20
Longest Job First Algorithm

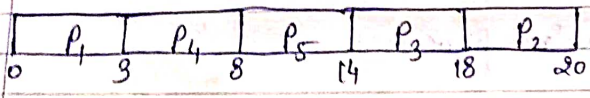
Process having largest Burst Time gets scheduled first

Criteria : BT
Mode : Non-Preemptive

Ex:

P.No.	A.T.	B.T.
1	0	3
2	1	2
3	2	4
4	3	5
5	4	6

Solu: its Gantt chart will be as -



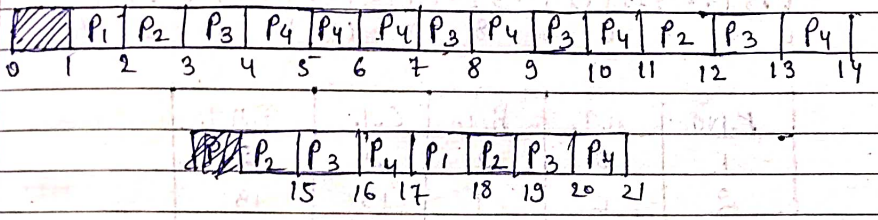
P.No.	A.T.	B.T.	C.T.	T.AT	W.T.	B.T.
1	0	3	3	3	0	0
2	1	2	20	19	17	17
3	2	4	18	16	12	12
4	3	5	8	5	0	0
5	4	6	14	10	4	4

Lecture - 21
Longest Remaining Time First

Ex:

P.No	A.T.	B.T.
1	1	2
2	2	4
3	3	6
4	4	8

Solu: its Gantt chart will be as -



P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.	R.T.
1	1	2	18	17	15	0
2	2	4	19	17	13	0
3	3	6	20	17	11	0
4	4	8	21	17	9	0

Lecture - 22
Longest Remaining time First Gate 2006 Question

Q) What is the average TAT using LRTF

P.No.	A.T.	B.T.
1	0	2
2	0	4
3	0	8

Soln: its Gantt chart will be as -

P_3	P_2	P_3	P_2	P_3	P_1	P_2	P_3	P_1	P_2	P_3

0 4 5 6 7 8 9 10 11 12 13 14

P.No.	A.T.	B.T.	C.T.	T.A.T.
1	0	2	12	12
2	0	4	13	13
3	0	8	14	14

Hence Average TAT = $\frac{12+13+14}{3}$

= 13 Ans

Lecture-23

Highest Response Ratio Next

Criteria: Response ratio (RR) = $\frac{W+S}{S}$

W: waiting time for a process so far

S = service time of a process or Burst time.

HRRN → it does not only favours shorter jobs but also limits the waiting time of longer jobs.

Mode → Non-preemptive

Ex:

P.No.	A.T.	B.T.
0	0	3
1	2	6
2	4	4
3	6	5
4	8	2

Soln: its Gantt chart for HRRN will be as -

P_0	P_1	P_2	P_4	P_3

0 3 9 13 15 20

$RR_2 = \frac{5+4}{4} = 2.25$ $RR_4 = \frac{1+2}{2} = 1.5$

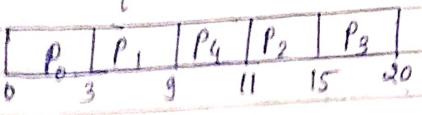
$RR_3 = \frac{3+5}{5} = 1.6$

Now again compute Response Ratios -

$RR_3 = \frac{7+5}{5} = \frac{12}{5} = 2.4$

$RR_4 = \frac{5+2}{2} = \frac{7}{2} = 3.5$

Gantt chart using shortest job first for some problem will be as -



Hence we can see the order of processes have been changed.

Note: HRRN is impossible to implement practically because we can not know the waiting time and burst time of a process before the execution.

P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.	R.T.
0	0	3	3	3	0	0
1	2	6	9	7	1	1
2	4	4	13	9	5	5
3	6	5	20	14	9	9
4	8	2	15	7	5	5

Lecture - 24
Priority scheduling

Priority

- Static
- Dynamic

Does not change throughout the execution of the process

changes at Regular intervals of time

Priority

- Preemptive
- Non-preemptive

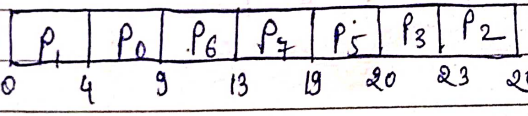
Note: We have to study only static Priority not Dynam

Lecture - 25
Non-preemptive priority scheduling

Ex.

P.No.	Priority	A.T.	B.T.
1	(L) 2	0	4
2	4	1	2
3	6	2	3
4	10	3	5
5	8	4	1
6	12 (H)	5	4
7	9	6	6

Soln: its Gantt chart will be as -



P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.	R.T.
1	0	4	4	4	0	0
2	1	2	25	24	22	22
3	2	3	23	21	18	18
4	3	5	9	6	1	1
5	4	1	20	16	15	15
6	5	4	13	8	4	4
7	6	6	19	13	7	7

Note :- In the case of Non-preemptive scheduling always Response Time and waiting time will be same. In the case of preemptive scheduling algorithm waiting time might be more compared to response time.

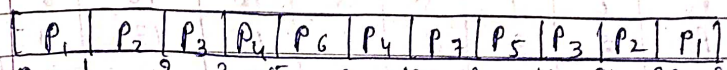
Lecture - 26

Preemptive Priority scheduling

Ex.

P.No.	Priority	A.T.	B.T.
1	2(L)	0	4
2	4	1	2
3	6	2	3
4	10	3	5
5	8	4	1
6	12(H)	5	4
7	9	6	6

Soln :- its Grantt chart will be as -



P.No.	A.T.	B.T	C.T.	T.A.T	W.T.	R.T.
1	0	4	25	25	21	0
2	1	2	22	21	19	0
3	2	3	21	19	16	0
4	3	5	12	9	4	0
5	4	1	19	15	14	14
6	5	4	9	4	0	0
7	6	6	18	12	6	6

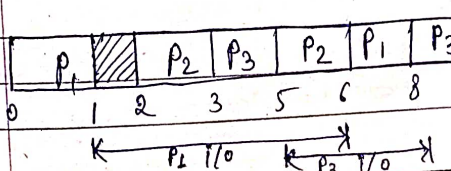
Lecture - 27

Preemptive priority with processes contains CPU and I/O time.

Ex.

P.No.	A.T.	Priority	CPU	I/O	CPU
P_1	0	2	1	5	3
P_2	2	3 (L)	3	3	1
P_3	3	1 (H)	2	3	1

Soln :- its Grantt chart will be as -



P.No.	A.T.	CPU	i/o	CPU	C.T.	T.A.T.	W.T.
P_1	0	1	5	3	10	10	6
P_2	2	3	3	1	15	13	9
P_3	3	2	3	1	9	6	3

Idere, CPU efficiency $= \frac{4}{15} \times 100$

CPU efficiency $= \left(1 - \frac{4}{15}\right) \times 100$

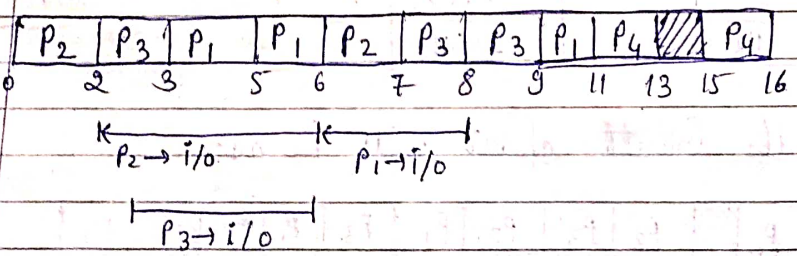
Lecture - 28

SRTF with processes contains CPU and i/o time example 1.

Ex.

P.No.	A.T.	B.T.	IOBT	B.T.	
1	0	3	2	2	→ 5
2	0	2	4	1	→ 3
3	2	1	3	2	→ 3
4	5	2	2	1	→ 3

Sol:) its Gantt chart will be as -

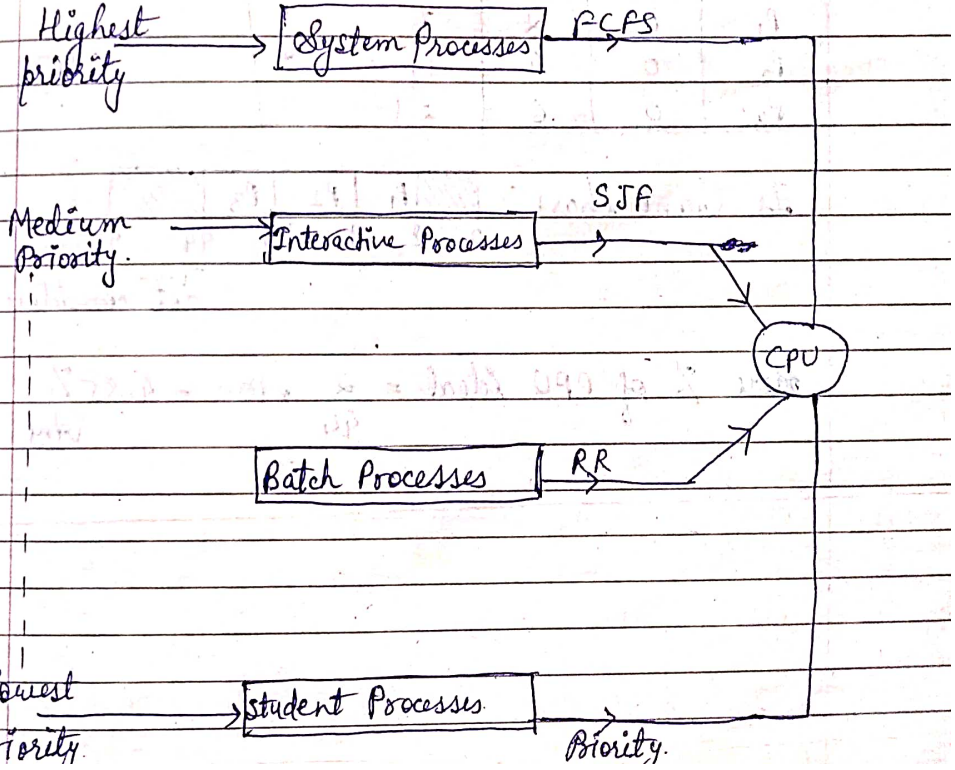


P.No.	A.T.	B.T.	IOBT	B.T.	C.T.	T.A.T.	W.T.
1	0	3	2	2	11	11	6
2	0	2	4	1	7	7	4
3	2	1	3	2	9	7	4
4	5	2	2	1	16	11	8

Lecture - 29

Multi level Queues and Multi level feedback queues.

Multiple Queue scheduling:



Lecture → 30.

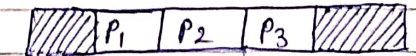
SRTF with Processes contains CPU and I/O time example 2.

Q-) Three Processes arriving at time zero with total execution time of 40, 20 and 30 units respectively. Each process spends the first 20% of execution time doing I/O, 70% of time doing computation and last 10% time doing again I/O. Compute % of CPU, Ideal time for SRTF.

Solu:-

P.No.	A.T.	I/O	CPU	I/O
P_1	0	2	7	1
P_2	0	4	14	2
P_3	0	6	21	3

its Gant chart:



0 2 9 23 44 45

not considered

Idere % of CPU ideal = $\frac{2}{44} \times 100 = 4.55\%$ Ans

Lecture - 31.

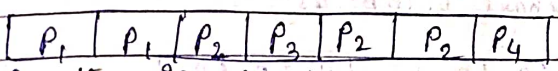
Example on SRTF Gate 2007.

Que:

P.No.	A.T.	B.T.
1	0	20
2	15	25
3	30	10
4	45	15

Find the waiting time of P_2 using SRTF.

Solu:- its Gantt chart will be as-



0 15 20 30 40 45 55 70

P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.
1	0	20	20	20	0
2	15	25	55	40	15
3	30	10	40	10	0
4	45	15	70	25	10

Hence waiting time of Process $P_2 = 15$ Ans

Lecture -

Smallest Remaining Time First

DATE:

Ex:-

P.No.	A.T.	B.T.	
1	0	7	6
2	1	5	4
3	2	3	2 1 0
4	3	1	0
5	4	2	0
6	5	1	0

Sol:- Its Gantt chart will be as-

P_1	P_2	P_3	P_4	P_3	P_3	P_5	P_5	P_2	P_1

0 1 2 3 4 5 6 7 9 13 19

P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.
1	0	7	19	19	12
2	1	5	13	12	7
3	2	3	6	4	1
4	3	1	4	1	0
5	4	2	9	5	3
6	5	1	7	2	1

Note:- It is also impractical to implement because we can never guess what will be the burst time for a process.

Lecture-33.

Example on SRTF Gate 2011

PAGE NO.:
DATE:

Ex:-

P.No	A.T.	B.T.
1	0	9
2	1	4
3	2	9

what is the average waiting time using SRTF?

Sol:- its Gantt chart will be as-

P_1	P_2	P_2	P_1	P_3

0 1 2 5 13 22

P.No.	A.T.	B.T.	C.T.	T.A.T.	W.T.
1	0	9	13	13	4
2	1	4	5	4	0
3	2	9	22	20	11

So. average waiting time = $\frac{4+0+11}{3} = 5$ units