HRRN

ROUND ROBIN

30|10|2022

Session - I

CPU Scheduling

Idle_time = 1 ✓

2) Shortest Job First (SJF):

Sel. Criteria: Burst Time (B.T)

Mode of op'n : Non-PreEmptive

Conflict Resolution: Favor Lower Pid

Working Principle

Among the Processes present in Ready 'Q' Select the one having Least 'BT'; Schedule it & Complete;

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

CPU	P_3	P_5	P_2	P_1	P_4	
	0	1	2	5	9	14

$\mathcal{L} = 14$

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

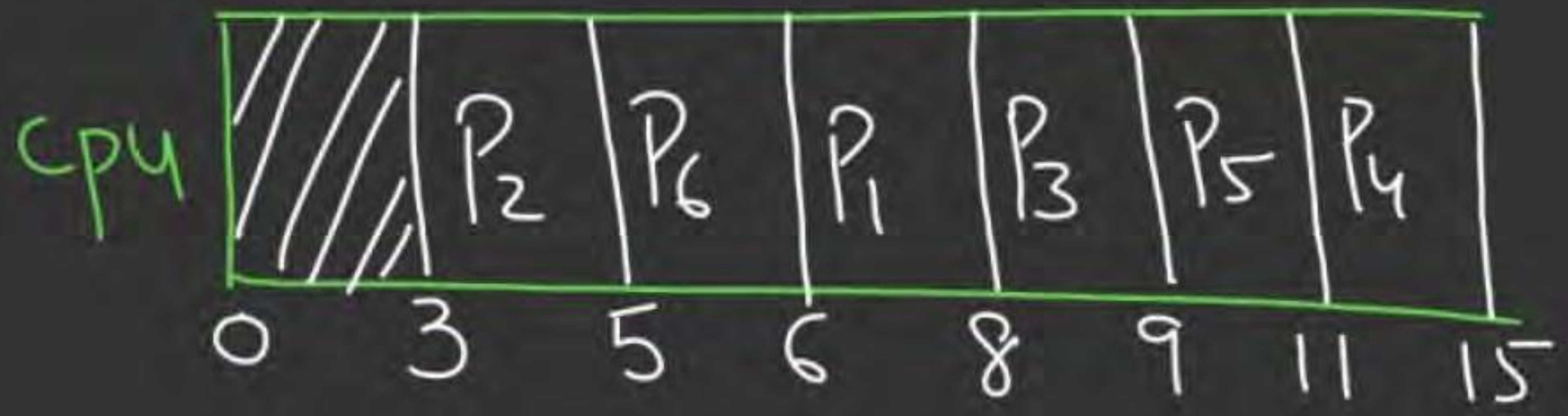
$\mathcal{L} =$

Av. TAT =

CPU		P_2	P_5	P_1	P_3	P_4	
	0	2	4	7	8	10	14

P.No	A.T	B.T
x1	5	2
x2	3	2
x3	8	1
4	6	4
x5	3	2
x6	4	1

$L = 15 - 3 = 12$



3. Shortest Remaining Time First (SRTF)

Pre Emptive SJF

Sel. Criteria: B.T
Mode of opn: Pre Emptive
Conflict resoln:
Lower Pid

"Pre Emption of running Process is based on arrival/availability of a strictly shorter process"

<

P.NO	A.T	B.T
1	0	~~7~~ 5
2	2	3

SJF: N.Pr

CPU	P_1	P_2
	0	7	10

SRTF/Pr-SJF

CPU	P_1	P_2	P_1
	0	2 (Pr)	5	10

P.NO	A.T	B.T
1	2	~~8~~ 6
2	3	7
×3	4	4

$\delta = 0$

S.J.F

CPU	///	P_1	P_3	P_2
	0 – 2	2 – 10	10 – 14	14 – 21

SRTF

CPU	///	P_1	P_3	P_1	P_2
	0 – 2	2 – 4	4 – 8	8 – 14	14 – 21

Pr

$L = 21 - 2$

P.NO	A.T	B.T
1	0	~~5~~ ~~4~~ 3
2	1	4
×3	2	2

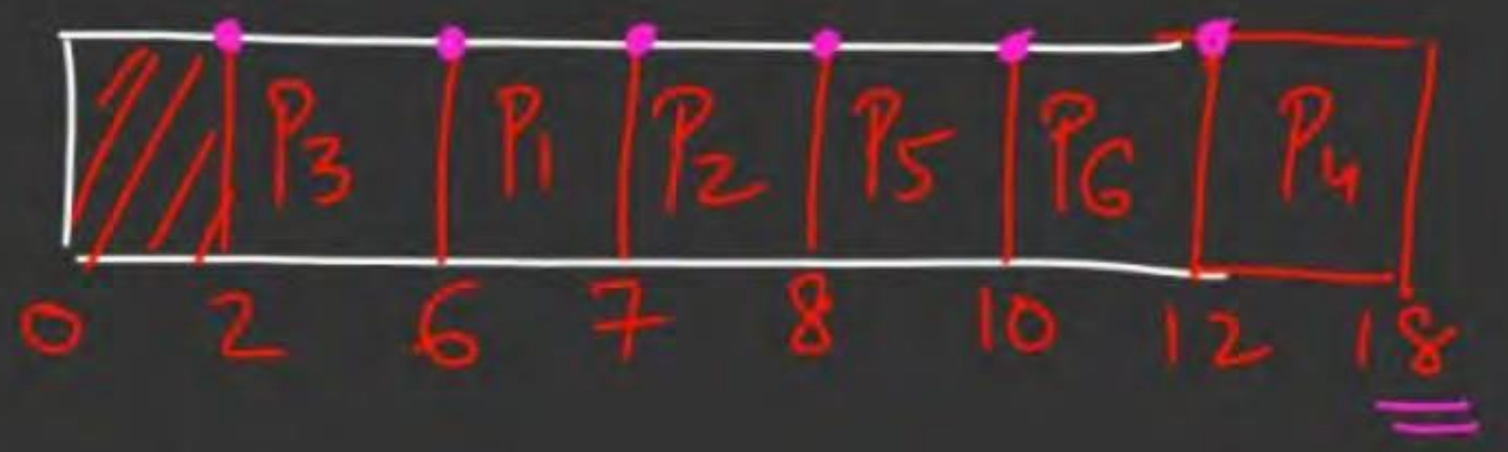
SRTF:

CPU	P_1	P_3	P_1	P_2
	0 – 2	2 – 4	4 – 7	7 – 11

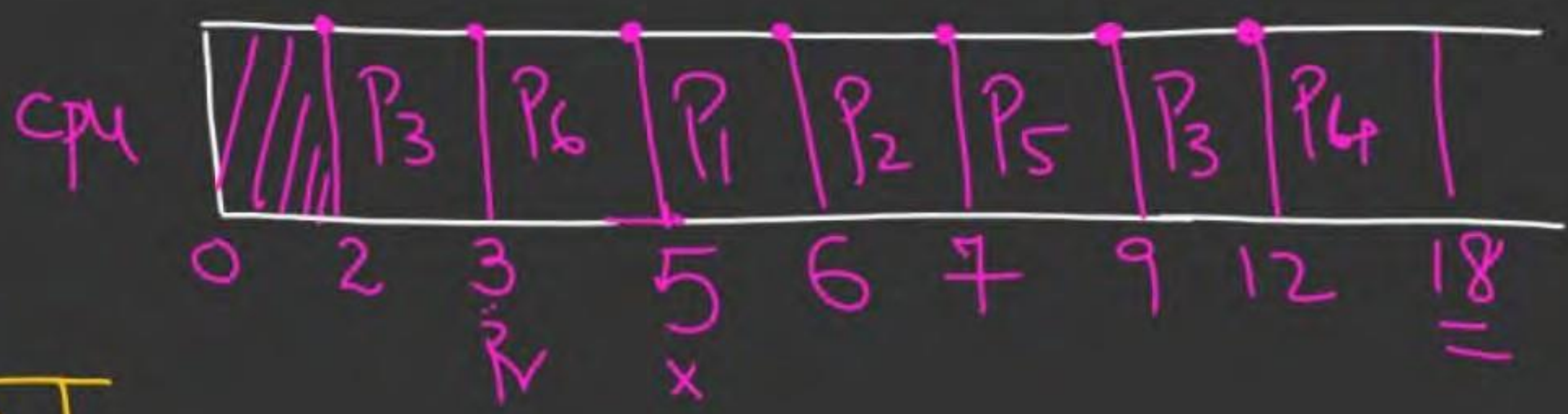
Pr

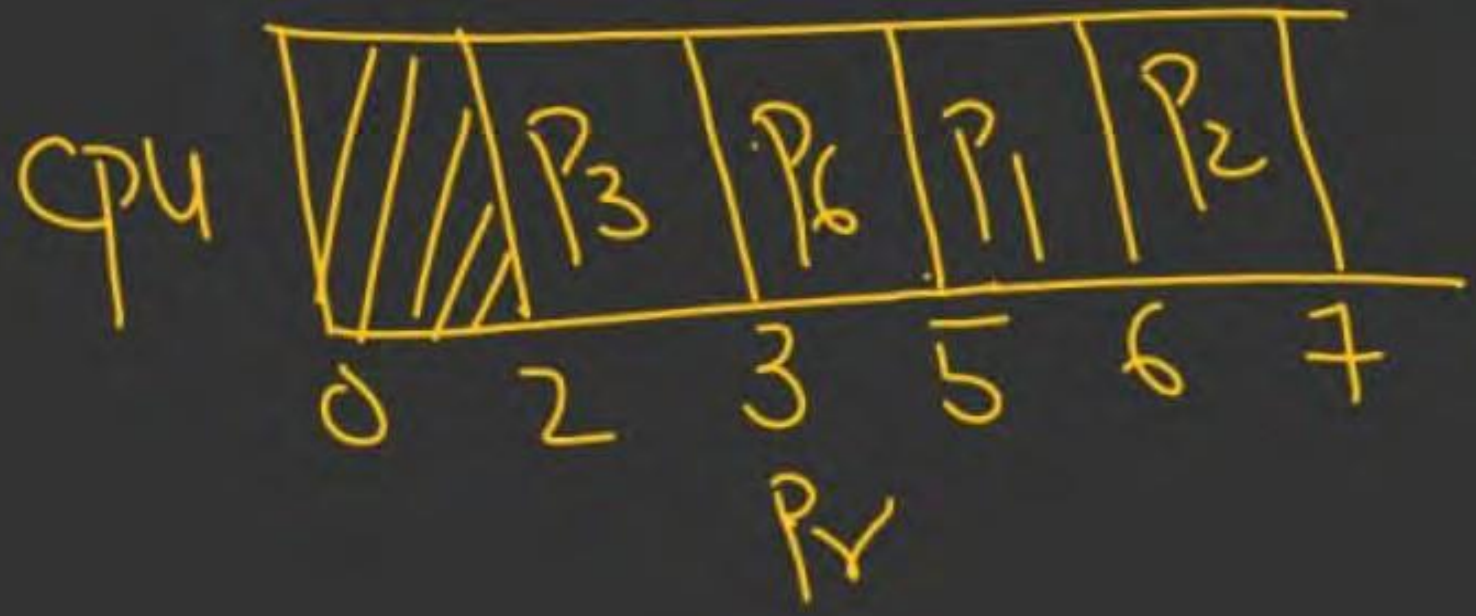
P.No	A.T	B.T
1	4	1
2	6	1
3	2	~~4~~ 3
4	2	6
5	5	2
6	3	~~2~~ ~~1~~

SJF



SRTF:





→ Performance of S.J.F / S.R.T.F

→ " Favors Shorter Process "

optimal Algo

{Advantage}

→ T

{→ Complete more # of Processes
→ Increased Thruput (Max.)
→ Minimizing Avg TAT & W.Ts ;}

{Drawback}

→ Cause Starvation to Longer Process

{Practical Limitation of SJF}

→ SJF Cannot be implemented with Real B.T's, but Can be impl. with Predicted B.T's

Since Burst times of Processes are not known a priori,

SJF | SRTF is practically non-Implementable ;

Optimal Algo

↓

Benchmark

R.Q | P_1 P_2 P_3

CPU

Can we predict the B.T's of P's

Exponential Averaging Technique to Predict NEXT CPU B.T of Process

→ Let P_i be Process

→ Let 't_i' Completed B.T

→ Let 'τ_i' the Predicted B.T

Let τ_{n+1} : Next cpu B.T of P_i

→ we already completed n-cpu Bursts

Initial value

Next-cpu B.T

Pr_i

τ_1	t_1	×	τ_2	t_2		
$W.T_1$	$B.T_1$	IOBT	WT_2	$B.T_2$	$W.T_3$	?
R.Q	cpu	IO	R.Q	cpu	R.Q	cpu

A.T

Pr t

τ_3

$$\boxed{\tau_{n+1} = \alpha t_n + (1-\alpha)\tau_n}$$

$$\boxed{0 \le \alpha \le 1}$$

Recurrence

Recursive Eqn

$$f(n) = n * f(n-1)$$
$$\rightarrow f(0) = 1$$

$$f(n+1) = \alpha f(n)$$

$$\tilde{T}_{n+1} = \alpha t_n + (1-\alpha)\tilde{T}_n - (1)$$

α : a Constant

$0 \le \alpha \le 1$

$$T_n = \alpha t_{n-1} + (1-\alpha) T_{n-1} - (2)$$

Back Substitution

$$T_{n-1} = \alpha t_{n-2} + (1-\alpha) T_{n-2} -$$

$$\tilde{T}_{n+1} = \alpha t_n + (1-\alpha)\left[\alpha t_{n-1} + (1-\alpha) T_{n-1}\right]$$

$$= \alpha t_n + \alpha(1-\alpha) t_{n-1} + (1-\alpha)^2 T_{n-1} - (3)$$

$$= \alpha t_n + \alpha(1-\alpha) t_{n-1} + \alpha(1-\alpha)^2 t_{n-2} + (1-\alpha)^3 T_{n-2} - (4)$$

$$\vdots$$

$$\tilde{T}_1$$

Even if, we Implement SJF with Predicted B.T, then still, it will Suffer from the Problem of STARVATION.

⟹ To overcome the problem of Starvation in SJF, we use "HRRN"

Consider a System using exponential Avg to Predict next CPU B.T. Assume $\alpha = 0.5$, $\tau_1 = 10$;
Previous Runs of a Process generates B.Ts of 8, 12, 14, 10
Predict the Next CPU Burst?

$\langle t_1 \; t_2 \; t_3 \; t_4 \rangle \; \tau_5$

$$\tau_5 = \alpha t_4 + (1-\alpha)\tau_4$$
$$= \frac{1}{2}(t_4 + \tau_4) = \frac{1}{2}(10 + \tau_4) = \frac{1}{2}(10 + 12.25) = \frac{22.25}{2} = 11.125$$

$$\tau_4 = \frac{1}{2}(t_3 + \tau_3) = \frac{1}{2}(14 + \tau_3) = \frac{1}{2}(14 + 10.5) = \frac{24.5}{2} = 12.25$$

$$\tau_3 = \frac{1}{2}(t_2 + \tau_2) = \frac{1}{2}(12 + \tau_2^{9}) = \frac{21}{2} = 10.5$$

$$\tau_2 = \frac{1}{2}(t_1 + \tau_1) = \frac{1}{2}(8 + 10) = 9$$

4) Longest Remaining Time First (LRTF): (*)

(*) G

Sel. Criteria : B.T
Mode of opn : PreEmptive

Tie breaking Rule : { Given ?

In Case of a tie b/w Processes, favor the process having lower Pid"

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

Q) Compute Avg. TAT using LRTF: $\frac{12+13+14}{3} = \frac{39}{3} = 13$ (2m)

CPU

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

Pr Pr Pr Pr Pr Pr Pr Pr

CPU

P_3	

0 4

Q.3 Consider the following processes, with the arrival time and the length of the CPU burst given in milliseconds. The scheduling algorithm used is preemptive Shortest Remaining-Time First (SRTF).

Process	Arrival Time	Burst Time
P1	0	10
P2	3	6
P3	7	1
P4	8	3

The average turnaround time of these processes is ________ milliseconds.

NAT

2m G

Q.4 Consider the following four processes with arrival times (in milliseconds) and their length of CPU bursts (in milliseconds) as shown below:

11:45 am

Process	P_1	P_2	P_3	P_4
Arrival time	0	1	3	4
CPU burst time	3	1	3	Z

These processes are run on a single processor using preemptive Shortest Remaining Time First (SRTF) Scheduling Algorithm. If the average waiting time of the processes is 1 millisecond, then the value of Z is