CS & IT ENGINEERING

Operating System

REVISION

CPU Scheduling- Part-01

Lecture No.- 03

By- Dr. Khaleel Khan Sir

Recap of Previous Lecture

Topic

Process Concepts

Threads and Multithreading

Topics to be Covered

- **Topic** CPU Scheduler
- **Topic** Round Robin (RR)
- **Topic** Priority Scheduling
- **Topic** Multilevel Queue

Topic : Basic Concepts

- Maximum CPU utilization obtained with multiprogramming
- CPU–I/O Burst Cycle – Process execution consists of a cycle of CPU execution and I/O wait
- CPU burst followed by I/O burst
- CPU burst distribution is of main concern

→ If Process has very less or almost no IO Burst then Such Process is known as CPU Bound ;

Lifecycle

⋮	
Load store add store read from file	CPU burst
wait for I/O	I/O burst
Store increment Index Write to file	CPU burst
wait for I/O	I/O burst
Load store And store Read from file	CPU burst
wait for I/O	I/O burst
⋮	

Topic : CPU Scheduler

(STS)

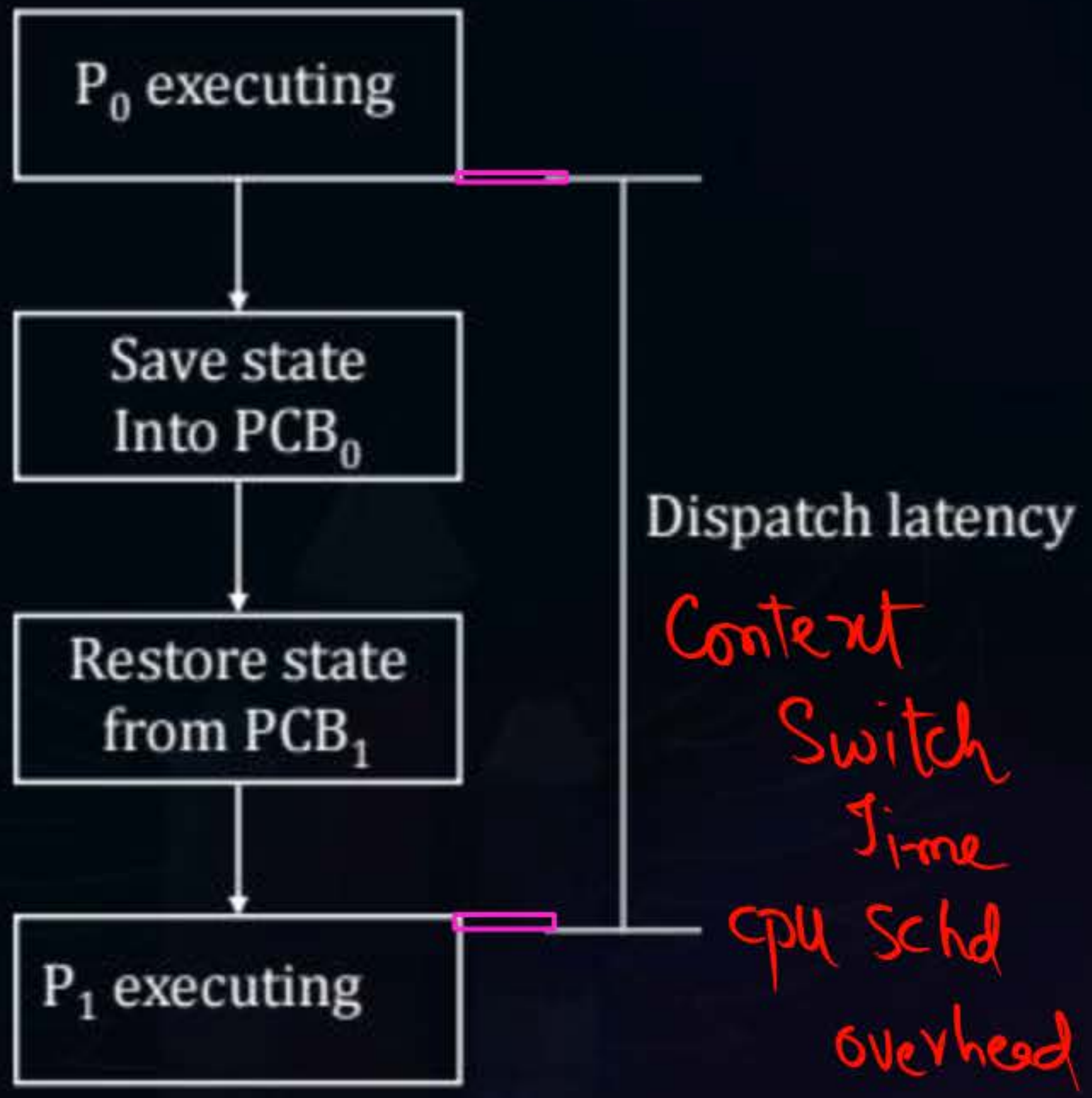
- The CPU scheduler selects from among the processes in ready queue, and allocates a CPU core to one of them
  - Queue may be ordered in various ways < FIFO + LIFO + Priority + - - - >
- CPU scheduling decisions may take place when a process:
  1. Switches from running to waiting state
  2. Switches from running to ready state
  3. Switches from waiting to ready
  4. Terminates
- For situations 1 and 4, there is no choice in terms of scheduling. A new process (if one exists in the ready queue) must be selected for execution.
- For situations 2 and 3, however, there is a choice.

Topic : Preemptive and Nonpreemptive Scheduling

- When scheduling takes place only under circumstances 1 and 4, the scheduling scheme is nonpreemptive.
- Otherwise, it is preemptive.
- Under Nonpreemptive scheduling, once the CPU has been allocated to a process, the process keeps the CPU until it releases it either by terminating or by switching to the waiting state.
- Virtually all modern operating systems including Windows, MacOS, Linux, and UNIX use preemptive scheduling algorithms.

Topic : Dispatcher

- Dispatcher module gives control of the CPU to the process selected by the CPU scheduler; this involves:
  - Switching context
  - Switching to user mode
  - Jumping to the proper location in the user program to restart that program
- **Dispatch latency** – time it takes for the dispatcher to stop one process and start another running



Topic : Scheduling Criteria

$= \frac{\text{No. of Processes Completed}}{\text{Unit Time}}$

- **CPU utilization** – keep the CPU as busy as possible ⇒ Max. of Thruput:
- **Throughput** – # of processes that complete their execution per time unit
- **Turnaround time** (TAT) – amount of time to execute a particular process
- **Waiting time** (WT) – amount of time a process has been waiting in the ready queue
- **Response time** (R.T) – amount of time it takes from when a request was submitted until the first response is produced.

CPU Scheduling (Problem)

FCFS	SJF	SRTF	R.R	F.S
S_1	S_2	S_3	S_4	S_5

Process Times

T.A.T

P_{r_i}							
	$W.T_1$	$B.T_1$	$IOBT_1$	$W.T_2$	$B.T_2$	$W.T_3$	$B.T_3$
	R.Q	CPU	IO	R.Q	CPU	R.Q	CPU

A.T $\qquad$ R_t $\qquad$ C.T

$\boxed{\text{(i) } TAT = CT - A.T}$

(ii) $W.T = TAT - (BT + IOBT)$

if $IOBT = 0$

$WT = TAT - BT$

For a set of n-processes, n! schedules

n=3

L

P_1	P_2	P_3
P_2	P_3	P_1

$\boxed{\text{Schedule length } (L) = Max(CT) - Min(AT)}$

$\boxed{\text{Thruput } (\mu) = \frac{n}{L}}$

$n \longrightarrow L \text{ units}$

$? \longleftarrow 1$

Response Time (R.T)

1) for Process

: The Time @ which the Process gets onto CPU for the First Time

$$\boxed{R.T(Pr_i) = \text{First Scheduling Time on CPU} - A.T}$$

2) for Request of a Process ✓

→ A process during its lifetime may be associated with several Requests;

→ Each Request will have its Response Time

→ R.T of the Req_i of Pr_K is the Time @ which the Req_i is Submitted into R.Q & the Time @ which its result (response) is generated by executing on CPU

Topic : Scheduling Algorithm Optimization Criteria

- Max CPU utilization
- Max throughput
- Min turnaround time
- Min waiting time
- Min response time

Scheduling Techniques (Algo's)

- Non-PreEmptive
  - FCFS
  - SJF
  - Priority
  - HRRN
- PreEmptive
  - SRTF;
  - LRTF;
  - Priority;
  - Round Robin

Topic : First- Come, First-Served (FCFS) Scheduling

Sel. Parameter : A.T

Mode : N.Pr.

Process		Burst Time
P_1	0	24
P_2	0	3
P_3	0	3

A.T = 0

$$Av.R.T = \frac{0+24+27}{3}$$

- Suppose that the processes arrive in the order: P_1 , P_2 , P_3

  The Gantt Chart for the schedule is:

P_1	P_2	P_3

0 24 27 30

- Waiting time for $P_1 = 0$; $P_2 = 24$; $P_3 = 27$
- Average waiting time: $(0 + 24 + 27)/3 = 17$

FCFS

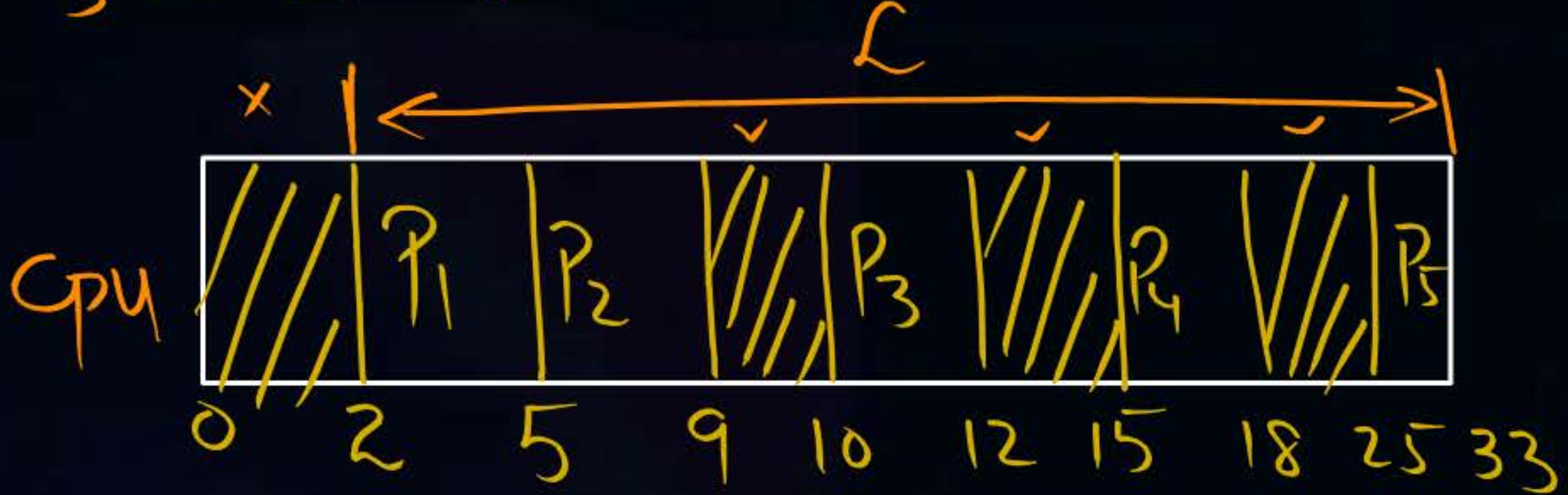
P.No	A.T	B.T
1	2	3
2	3	4
3	10	2
4	15	3
5	25	8

1) $L = 33 - 2 = \underline{\underline{31}}$

2) $Av.\ TAT = (3+6+2+3+8)/5 = 22/5 = 4.4$

3) $Av.\ W.T = (0+2+0+0+0)/5 = 0.4$

4) $Av.\ R.T = \dfrac{0+2+0+0+0}{5} = 0.4$



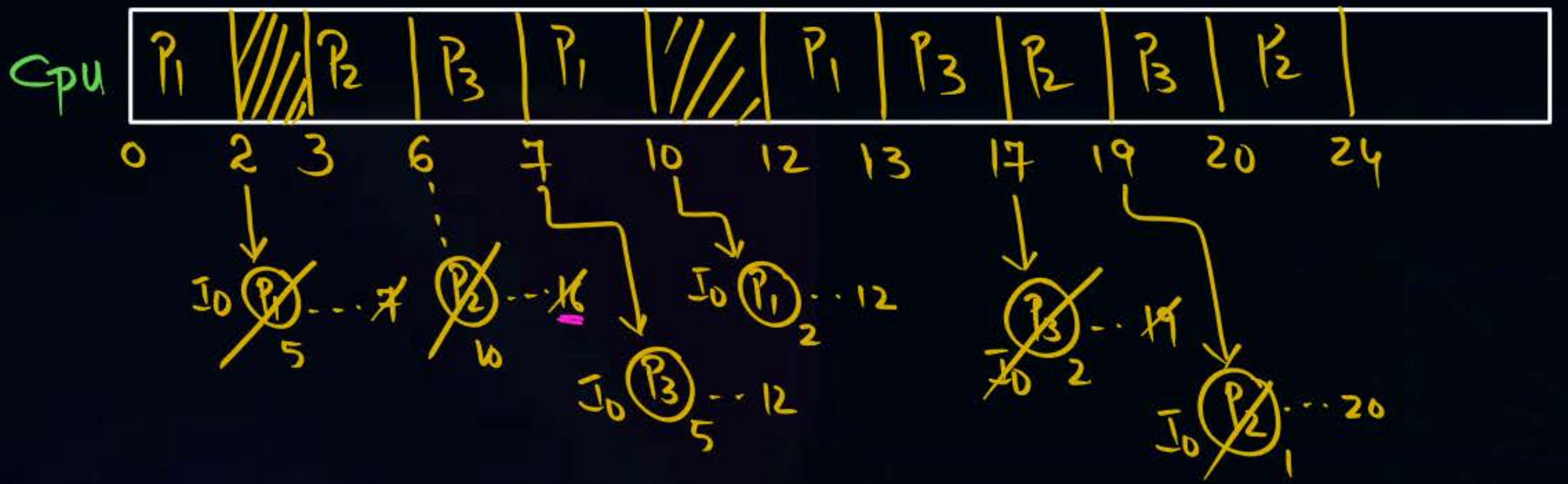
$Av.\ R.T = Av.\ W.T$

iff Schdlng in N.Pr

Concurrent IO ✓

P.No	A.T	⟨BT ; IOBT ; BT ; IOBT ; B.T⟩
1	0	⟨2 ; 5 ; 3 ; 2 ; 1⟩
2	3	⟨3 ; 10 ; 2 ; 1 ; 4⟩
3	5	⟨1 ; 5 ; 4 ; 2 ; 1⟩

R.Q $\boxed{\cancel{P_1}\ \cancel{P_2}\ \cancel{P_3}\ \cancel{P_1}\ \cancel{P_1}\ \cancel{P_3}\ \cancel{P_2}\ \cancel{P_3}\ P_2}$



$L = 24 - 0 = 24$

$\frac{\% \text{Cpu Idleness}}{L} = 3/24 = 1/8$

$\text{Av.TAT} = \frac{13+21+15}{3} = \frac{49}{3}$

$\text{Av.W.T} = \frac{0+1+1+1+0}{3} = 1$

$\text{Av. R.T} = \frac{0+0+1}{3} = 1/3 = 0.33$

$R.T(P_1) = 0$

$R.T(P_2) = 0$

$R.T(P_3) = 1$

$\delta = 1$

P.No	A.T	⟨IOBT ; B.T ; IOBT⟩
1	0	⟨3 ; 5 ; 2⟩
2	3	⟨2 ; 6 ; 4⟩

RQ: ~~P1~~ ~~P2~~ ~~P1~~ ~~P2~~ ~~P1~~ P2

CPU

δ	P1	///	δ	P2	δ	P1	δ	P2	δ	P1	///	δ	P2
0–1	1–1	1–3	3–4	4–4	4–5	5–10	10–11	11–17	17–18	18–18	18–21	21–22	22–22

0 1 1 3 4 4 5 10 11 17 18 18 21 22 22

IO ~~P1~~ 3 ... 4

IO ~~P2~~ 2 ... 6

~~IO~~ P1 2 ... ~~12~~

IO ~~P2~~ 4 ... 21

Topic : Shortest-Job-First (SJF) Scheduling

- Associate with each process the length of its next CPU burst
  - Use these lengths to schedule the process with the shortest time
- SJF is optimal – gives minimum average waiting time for a given set of processes
- Preemptive version called shortest-remaining-time-first
- How do we determine the length of the next CPU burst?
  - Could ask the user
  - Estimate

Topic : Example of SJF

N. Pr

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart

P_4	P_1	P_3	P_2

0 3 9 16 24

- Average waiting time = (3 + 16 + 9 + 0) / 4 = 7

P.No	A.T	B.T
1	3	4
2	6	2
3	7	1
4	8	2
5	10	1

CPU

///	P_1	P_3	P_2	P_5	P_4	

0 3 7 8 10 11 13

$$Av.TAT = \frac{4+4+1+5+1}{5} = \frac{15}{5} = 3$$

$$Av.W.T = \frac{0+2+0+3+0}{5} = 1$$

$$Av.R.T = 0+2+0+3+0 = 1$$

Topic : Determining Length of Next CPU Burst

- Can only estimate the length – should be similar to the previous one
- Then pick process with shortest predicted next CPU burst
- Can be done by using the length of previous CPU bursts, using exponential averaging
    1. t_n = actual length of n^{th} CPU burst
    2. τ_{n+1} = predicted value for the next CU burst
    3. α, $0 \le \alpha \le 1$
    4. Define:

(Aging Algo.)

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

- Commonly, α set to ½

S.J.F
↳ optimal Algo
→ Least Av. TATs & Av. W.T's
→ Max. Thruput
→ Starvation to long Processes
→ Implementation

Topic : Examples of Exponential Averaging

- $\alpha = 0$
  - $\tau_{n+1} = \tau_n$
  - Recent history does not count
- $\alpha = 1$
  - $\tau_{n+1} = \alpha t_n$
  - Only the actual last CPU burst counts
- If we expand the formula, we get:
  - $\tau_{n+1} = \alpha t_n + (1 - \alpha)\alpha t_{n-1} + \ldots$
  - $+ (1 - \alpha)^j \alpha t_{n-j} + \ldots$
  - $+ (1 - \alpha)^{n+1} \tau_0$
- Since both α and $(1 - \alpha)$ are less than or equal to 1, each successor predecessor term has less weight than its predecessor

Topic : Shortest Remaining Time First Scheduling

- Preemptive version of SJN
- Whenever a new process arrives in the ready queue, the decision on which process to schedule next is redone using the SJN algorithm.
- Is SRT more “optimal” than SJN in terms of the minimum average waiting time for a given set of processes?

Topic : Example of Shortest-Remaining-Time-First

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	Arrival Time	Burst Time
P_1	0	~~8~~ 7
P_2	1	× 4
P_3	2	9
P_4	3	5

CPU | P_1 | P_2 | P_4 | P_1 | P_3 |
0 1 5 10 17 26

Av. W.T $= \frac{9+15+2}{4} = \frac{26}{4} = 6.5$

Av. R.T $= \frac{0+0+15+2}{4} = \frac{17}{4} = 4.25$

P_1	P_2	P_4	P_1	P_3

0 1 5 10 14 26

×

- Preemptive SJF Gantt Chart

- ~~Average waiting time = [(10 – 1) + (1 – 1) + (17 – 2) + (5 – 3)]/4 = 26/4 = 6.5~~

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

Pr

O

CPU

P1	P2	P2	P3	P3	P4	P1	

0 2 4 6 7 8 9 14

Pr

$$Av.\ W.T = 7+0+2+1 = 10/4 = 2.5$$

$$Av.\ R.T = 0+0+2+1 = 3/4 = 0.75$$

2 mins Summary

Topic One : Scheduling Criteria

Topic Two : classification

Topic Three : FCFS

Topic Four : S.J.F

33

Topic Five : S.R.T.F

THANK - YOU