

# Process Scheduling Algorithms

- FCFS
- SJF
- Priority
- SRIF
- RR
- multi-level queue

← non-preemptive / preemptive

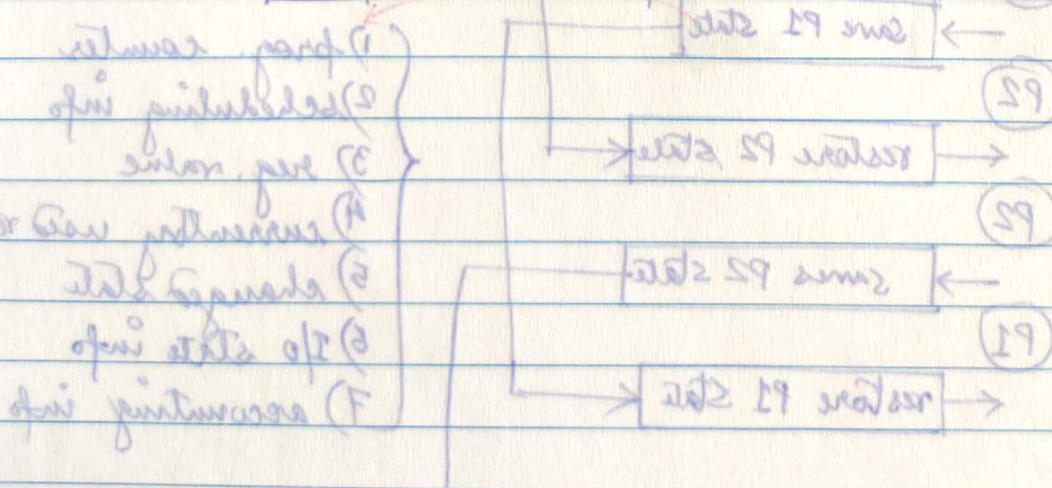
## FCFS

- non-preemptive
- easy to implement
- implemented with queue
- poor performance as avg. waiting time is high.

	AT	ET	ST	WT
P <sub>0</sub>	0	5	0	0
P <sub>1</sub>	1	3	5	(5-1)=4
P <sub>2</sub>	2	8	8	(8-2)=6
P <sub>3</sub>	3	6	16	(16-3)=13

$$\text{Avg WT.} = \frac{0+4+6+13}{4}$$

$$= 5.75$$





## SJF:

- non-preemptive, pre-emptive
- minimizes waiting time
- easy to implement in Batch system where CPU time is known.
- not for interactive system where CPU time is not known.

	AT.	ET	ST	WT	
P <sub>0</sub>	0	5	0	0	P <sub>0</sub>   P <sub>1</sub>   P <sub>3</sub>   P <sub>2</sub>
P <sub>1</sub>	1	3	5	4	0   5   8   14   22
P <sub>2</sub>	2	8	14	12	
P <sub>3</sub>	3	6	8	5	

$$\text{Avg. WT.} = \frac{0+4+12+5}{4}$$

$$= 5.25$$

## Priority Based:

- non-preemptive
- process with same priority follows FCFS.
- priority depends on memory requirement, time requirement, resource requirement.

	AT	ET	Prd.	ST.	WT	
P <sub>0</sub>	0	5	1	0	0	P <sub>0</sub>   P <sub>3</sub>   P <sub>1</sub>   P <sub>2</sub>
P <sub>1</sub>	1	3	2	11	10	0   5   11   14   22
P <sub>2</sub>	2	8	1	14	12	
P <sub>3</sub>	3	6	3	5	2	

$$\text{Avg WT.} = \frac{0+10+12+2}{4} = 6$$



## SRTF:

- pre-emptive version of SJF.
- processor is allocated to a job closest to completion.
- can be pre-empted by a newer ready job with shorter completion time.
- used in batch environment (process preference).

AT	ET	ST	WT
0	0	0	0
1	2	1	1
2	8	2	2
3	14	3	3

## RR:

- pre-emptive.
- given fixed time to execute; "quantum"
- context-switching is used to save process states when pre-empted.

	AT	ET	ST	WT
P <sub>0</sub>	0	5/2		0 + (12-3) = 9
P <sub>1</sub>	1	3		(3-1) = 2
P <sub>2</sub>	2	8/2		(6-2) + (14-9) + (20-17) = 12
P <sub>3</sub>	3	6/3		(9-3) + (14-12) = 11

AT	ET	ST	WT
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3

avg. WT. =  $(9 + 2 + 12 + 11) / 4 = 8.5$

$(2) = 0 + 10 + 15 + 5 = 30$



## Multi-level queue scheduling:

- not independent scheduling.
- make use of other existing algorithms.
- multiple queues are maintained for process with common characteristics.
- each queue can have its own scheduling algorithm.
- priorities are assigned to each queue.

P.S. // eg: CPU-bound jobs can be scheduled in one queue and I/O-bound " " " " " other " " .

Process Scheduler alternatively selects jobs from each queue and assign them to the CPU based on algo.

A resource can't be taken from a process unless the process releases it voluntarily.

A set of processes are waiting for each other in circular form.

[...]