SCHEDULING

DCS4103 Operating System

Process Scheduling

- ❑ Process of determining which process will own CPU for execution while another process is on hold
- ❑ Select at least one of the processes available in the ready queue for execution
- ❑ Selection process carried out by the CPU scheduler

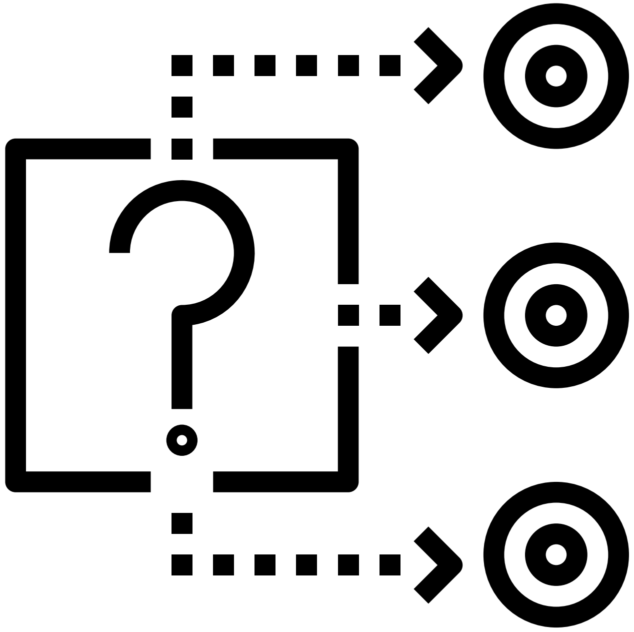
Criteria

Minimize

- ❑ CPU utilization
  - ▪ CPU remains busy
- ❑ Throughput
  - ▪ Number of processes that finish their execution per unit time

Maximize

- ❑ Waiting time
  - ▪ Process needs to wait in the ready queue.
- ❑ Response time
  - ▪ Request was submitted until the first response is produced.



Uniprocessor Scheduling

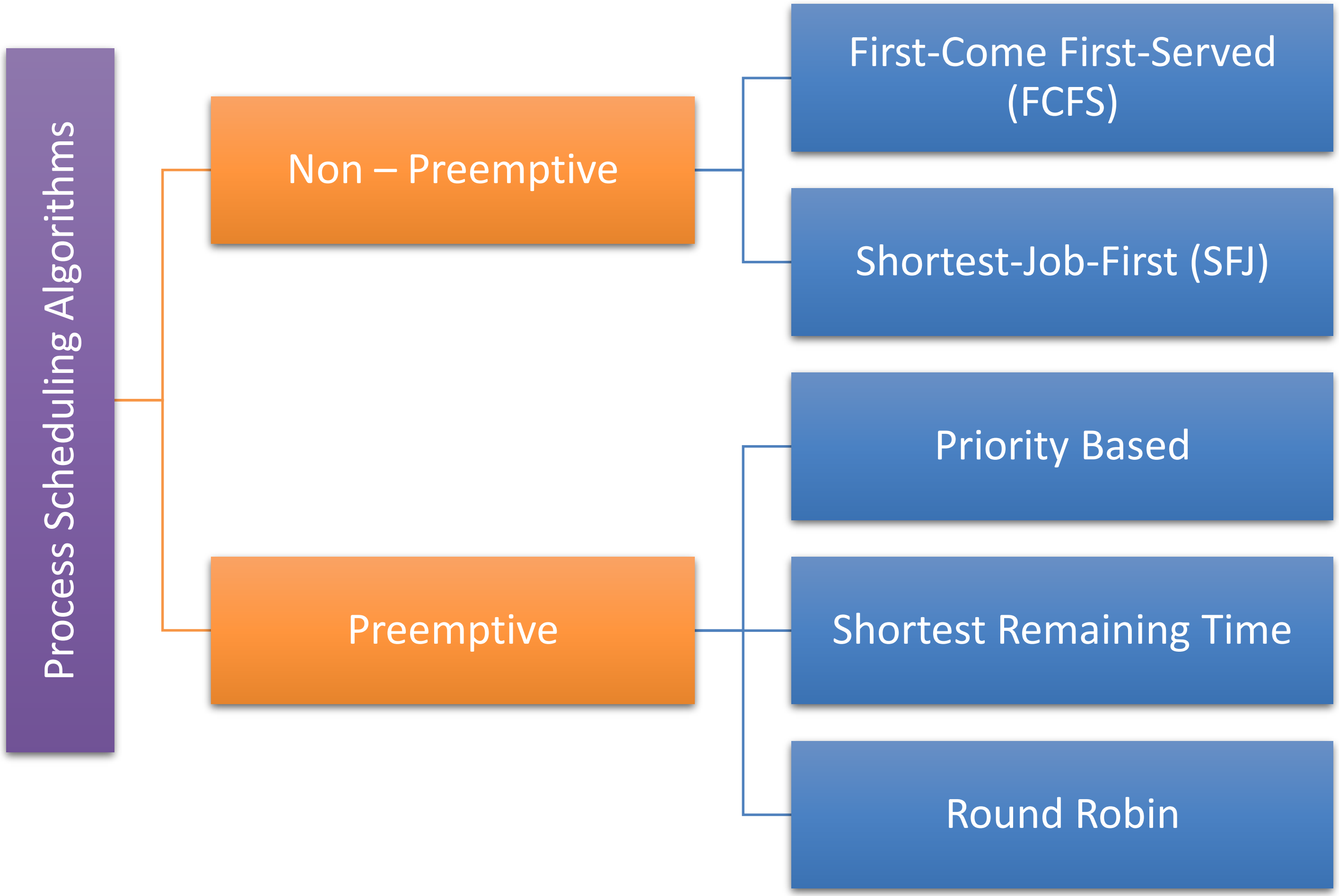
Process Scheduling Algorithms

Non - Preemptive

Process cannot be preempted until completes the allotted time

Preemptive

Based on priority where a scheduler may preempt a low priority running process anytime when a high priority process enters a ready state



Process Scheduling Algorithms

Arrival time	❑ Time at which the process arrives in the ready queue.
Completion Time	❑ Time at which process completes its execution.
Burst Time	❑ Time required by a process for CPU execution.
Turn Around Time	❑ Time Difference between completion time and arrival time. $\textit{Completion Time} - \textit{Arrival Time}$
Waiting Time (W.T)	❑ Time Difference between turn around time and burst time. $\textit{Turnaround Time} - \textit{Burst Time}$

Process Scheduling Algorithms

First-Come First Served (FSFC)

- ❑ Implemented non-preemptive
- ❑ Executed on a first-come, first-serve basis
- ❑ Easy to implement and use.
- ❑ Poor in performance
- ❑ High waiting time
- ❑ Example
  - ▪ Buying movie ticket at counter

Disadvantages

- ❑ In Non-preemptive, CPU will be release after the process finish its execution
- ❑ Short processes that are at the back of the queue must wait for the long process at the front to finish.
- ❑ Not suitable for time-sharing systems
- ❑ Not very efficient.

Shortest-Job-First (SJF)

- ❑ Implemented non-preemptive
- ❑ The next process or job with the shortest completion time will be executed first.
- ❑ Useful for batch-type processing
- ❑ Improves job output by offering shorter jobs
- ❑ Reduces the average waiting time

Disadvantages

- ❑ Job completion time must be known earlier
- ❑ Long turnaround times or starvation
- ❑ Elapsed time should be recorded, that results in more overhead on the processor.
- ❑ Hard to know the length of the upcoming CPU request.

Process Scheduling Algorithms

Priority Based

- The scheduler selects the tasks to work as per the priority.
- Preemptive
    - Run a task with a higher priority
- Lower the number, higher is the priority.
- If two jobs having the same priority are READY, it works on a FIRST COME, FIRST SERVED basis.
- Easy to use
- Suitable for applications with fluctuating time and resource requirements.

Disadvantages

- If the system eventually crashes, all low priority processes get lost.
- If high priority processes take lots of CPU time,
    - the lower priority processes may starve and will be postponed for an indefinite time.
- If a new higher priority process keeps on coming in the ready queue,
    - the process which is in the waiting state may need to wait for a long duration of time.

Process Scheduling Algorithms

Shortest Remaining Time

- ❑ SJF preemptive scheduling
  - Jobs are put into the ready queue
  - Shortest burst time begins execution
  - If a process with even a shorter burst time arrives, the current process is removed or preempted from execution, and the shorter job is allocated CPU cycle
- ❑ Prevents a newer ready state process from holding the completion of an older process.
- ❑ Applied in batch environments
- ❑ Helps to schedule the process with the shortest possible time

Disadvantages

- ❑ Job completion time must be known earlier
- ❑ Long turnaround times or starvation
- ❑ Elapsed time should be recorded, that results in more overhead on the processor.
- ❑ Hard to know the length of the upcoming CPU request.

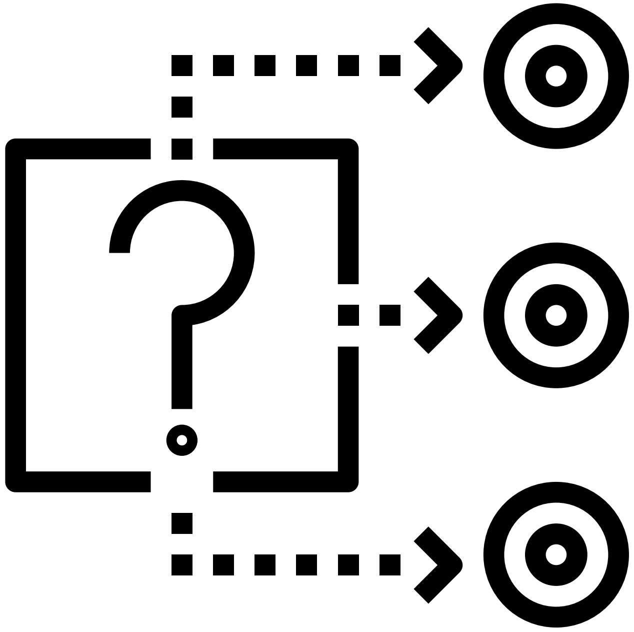
Process Scheduling Algorithms

Round Robin

- ❑ Implemented pre-emptive
- ❑ The oldest, easiest, fairest and simplest
- ❑ Each ready task runs turn by turn only in a cyclic queue for a limited time slice
- ❑ Preempted process added to the end of the queue.
- ❑ Clock-driven
- ❑ Used for multitasking
- ❑ Starvation free
- ❑ Without priority
- ❑ Performance in average response time

Disadvantages

- ❑ If slicing time of OS is low, the processor output will be reduced.
- ❑ Spends more time on context switching
- ❑ Performance depends on time quantum.
- ❑ Lower time quantum results in higher the context switching overhead in the system.
- ❑ Difficult to find correct time quantum



Multiprocessor Scheduling

Introduction

- ❑ Multiple CPUs share the load
- ❑ Various processes run simultaneously
- ❑ More complex
- ❑ Tightly coupled
  - ▪ Shares common bus, memory, and other peripheral devices
- ❑ Heterogeneous
  - ▪ Different kinds of CPUs
- ❑ Homogenous
  - ▪ The same CPU

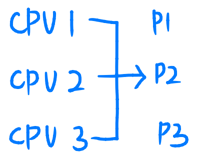
Scheduling Algorithms

- Processor Affinity
- Load Balancing
- Multi-core Processors
- Symmetric Multiprocessor
- Master-Slave Multiprocessor
- Virtualization and Threading

Approaches

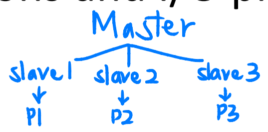
Symmetric

- ❑ Self-scheduling
- ❑ Each processor have its private queue



Asymmetric

- ❑ Master Server (handle scheduling decisions and I/O processing)
- ❑ User code (other processors)
- ❑ Reduce need data sharing



Revision Questions

Consider the set of five processes whose arrival time and burst time are given below:

Process	Arrival Time	Burst Time
P1	2 3	6
P2	5 5	2
P3	1 2	8
P4	0 1	3
P5	4 4	4

Calculate the Average Turnaround Time (TAT) and Average Waiting Time (AWT), if First-Come First-Serve (FCFS) algorithm is followed.

Gantt Chart

0–3	3–11	11–17	17–21	21–23
P4	P3	P1	P5	P2

Process ID	Completion Time		Turnaround Time		Waiting Time
P1	17	-2	15	-6	9
P2	23	-5	18	-2	16
P3	11	-1	10	-8	2
P4	3	-0	3	-3	0
P5	21	-4	17	-4	13

Average Turnaround Time $= 63 \div 5$
$= 12.6$ units

Average Waiting Time $= 40 \div 5$
$= 8$ units

Revision Questions

Consider the set of five processes whose arrival time and burst time are given below:

Process	Arrival Time	Burst Time
P1	0 1	21
P2	1	3 2
P3	2	6 3
P4	3	7 4

Calculate the Average Turnaround Time (TAT) and Average Waiting Time (AWT), if Shortest-Job-First(SJF) algorithm is followed.

Gantt chart

0	21	24	30	37
	P1	P2	P3	P4

Process ID	Completion Time	Turnaround Time	Waiting Time
P1	21 -0	21 -21	0
P2	24 -1	23 -3	20
P3	30 -2	28 -6	22
P4	37 -3	34 -7	27

Average Turnaround Time $= 106 \div 4$
$= 26.5$ units

Average Waiting Time $= 69 \div 4$
$= 17.25$ units

Revision Questions

Consider the set of five processes whose arrival time and burst time are given below:

Process	Arrival Time	Burst Time	Priority
P1	0 1	4	2
P2	1 2	3	3
P3	2 3	1	4
P4	3 4	5	5
P5	4 5	2	5

Calculate the Average Turnaround Time (TAT) and Average Waiting Time (AWT), if Priority Based algorithm is followed.

Gantt Chart

0	4	7	8	13	15
P1	P2	P3	P4	P5	

Process ID	Completion Time		Turnaround Time		Waiting Time
P1	4	0	4	4	0
P2	7	1	6	3	3
P3	8	2	6	1	5
P4	13	3	10	5	5
P5	15	4	11	2	9

Average TT = 37 ÷ 5
= 7.4 units

Average WT = 22 ÷ 5
= 4.4 units

Revision Questions

Consider the set of five processes whose arrival time and burst time are given below:

Process	Arrival Time	Burst Time
P1	2	6
P2	5	2
P3	1	8
P4	0	3

Calculate the Average Turnaround Time (TAT) and Average Waiting Time (AWT), if Shortest Remaining Time algorithm is followed.

Revision Questions

Consider the set of five processes whose arrival time and burst time are given below:

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	1
P4	3	2
P5	4	3

If the CPU scheduling policy is Round Robin with time quantum = 2, calculate the Average Turnaround Time and Average Waiting Time