

Operating Systems

| Round Robin – Priority Algorithms

Prof. Dr. Mostafa Elhosseini
Professor of Smart Systems Engineering

<https://elhosseini.com/>

| Agenda

- Round Robin (RR)
- Priority Scheduling
- ~~Multilevel Queue Scheduling (MLQ)~~
- ~~Multilevel Feedback Queue (MLFQ)~~

Round Robin (RR) Scheduling

-  Round Robin is a preemptive scheduling algorithm designed for time-sharing systems.
- Each process gets an equal time slice (quantum)  of CPU time.
- When its time expires, the CPU switches to the next process in the ready queue .
- The preempted process goes to the back of the queue to wait for its next turn..

Round Robin (RR) Scheduling

How It Works

- 1 All ready processes are placed in a FIFO queue.
- 2 The CPU picks the first process and allows it to run for 1 quantum.
- 3 If the process finishes → remove it from the queue.
- 4 If it doesn't → move it to the end of the queue, and pick the next one.

Round Robin (RR) Scheduling

Key Features

-  Preemptive by nature.
-  Ensures fairness — every process gets CPU time.
-  Time Quantum (q): must be chosen carefully!
 - Too small $\rightarrow$ too many context switches (inefficient).
 - Too large $\rightarrow$ behaves like FCFS.

Round Robin (RR) Scheduling

Analogy

- Think of a classroom discussion  :
- Each student gets 30 seconds to talk.
- When time's up, the teacher moves to the next student —
- but they'll get another turn later.

 Everyone gets a voice, no one dominates the mic!

Round Robin (RR) Scheduling

- Consider the following set of processes that arrive at time 0, with the length of the CPU burst given in milliseconds: If we use a time quantum of 4 milliseconds

Required

- Draw the Gantt Char
- Calculate the following:
 - Average Waiting Time
 - Average Turnaround Time

<u>Process</u>	<u>Burst Time</u>
P_1	24
P_2	3
P_3	3

Round Robin (RR) Scheduling

<u>Process</u>	<u>Burst Time</u>
P_1	24
P_2	3
P_3	3



Priority Scheduling Algorithm

- Each process is assigned a priority value $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$.
- The CPU is allocated to the process with the highest priority (lowest priority number or highest value — depending on the system's design).
- Can be Preemptive or Non-Preemptive depending on implementation.



Priority Scheduling Algorithm



How It Works

- When the CPU becomes free, the scheduler chooses the process with the highest priority.
- In preemptive mode, if a new process arrives with a higher priority than the current one → it interrupts and takes the CPU.
- Continue until all processes are finished.



Priority Scheduling Algorithm



Key Features

-  Ensures important or time-critical tasks are executed first.
-  Works well in real-time systems or when some jobs are more urgent.
-  Problem: Starvation — low-priority processes may never get CPU time.
-  Solution: Aging — gradually increase the priority of waiting processes..



Priority Scheduling Algorithm

Analogy

- Think of an emergency room  :
- patients are not treated in the order they arrive,
- but based on how critical their condition is  .
- Minor cases may wait, but serious ones get immediate attention.



Priority Scheduling Algorithm

- Consider the following set of processes, assumed to have arrived at time 0 in the order $P_1, P_2, \dots, P_5$, with the length of the CPU burst given in milliseconds

Required

- Draw the Gantt Chart
- Calculate the following:
 - Average Waiting Time
 - Average Turnaround Time

<u>Process</u>	<u>Burst Time</u>	<u>Priority</u>
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2



Priority Scheduling Algorithm

<u>Process</u>	<u>Burst Time</u>	<u>Priority</u>
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2