Interactive Scheduling Algorithms

- Usually preemptive
  - Time is **sliced** into quantum (time intervals)
  - Scheduling decision is also made at the beginning of each quantum

- Performance Criteria
  - Min Response time
  - best proportionality

- Representative algorithms:
  - Priority-based
  - Round-robin
  - Multi Queue & Multi-level Feedback
  - Shortest process time
  - Guaranteed Scheduling
  - Lottery Scheduling
  - Fair Sharing Scheduling

Priority Scheduling

- Each job is assigned a priority.
- FCFS within each priority level.
- Select highest priority job over lower ones.

- Rationale: higher priority jobs are more mission-critical
  - Example: DVD movie player vs. send email

- Problems:
  - May not give the best AWT
  - Starvation
Set Priority

- Two approaches
  - Static (for system with well known and regular application behaviors)
  - Dynamic (otherwise)
- Priority may be based on:
  - Cost to user.
  - Importance of user.
  - Aging
  - Percentage of CPU time used in last $X$ hours.

Round-Robin (RR)

- One of the oldest, simple, commonly used scheduling algorithms
- Select process/thread from ready queue in a round-robin fashion (take turns)

- Problems:
  - Do not consider priority
  - More context switch overhead
Round-robin: Example

<table>
<thead>
<tr>
<th>Process</th>
<th>Duration</th>
<th>Order</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>P2</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>P3</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Suppose time quantum is: 1 unit, P1, P2 & P3 never block

Do it yourself

P1waiting time: 4
P2waiting time: 6
P3 waiting time: 6

The average waiting time (AWT):

\[
\frac{(4+6+6)}{3} = 5.33
\]

Time Quantum

- **Time slice too large**
  - FIFO behavior
  - Poor response time
- **Time slice too small**
  - Too many context switches (overheads)
  - Inefficient CPU utilization
- **Heuristic: (Eliminating preemption)**
  - 70-80% of jobs block within time-slice
- **Typical time-slice**
  - 10 to 100 ms
- **Time spent in system depends on size of job**
Multi-Queue Scheduling

- Hybrid between priority and round-robin
- Processes assigned to one queue
- Scheduling between queues
  - Fixed Priorities
  - Dynamic priorities based on CPU % spent on queue
- Example
  - System processes
  - Interactive programs
  - Background Processes
- Address the starvation problem

Multi-Queue Scheduling: Example

Highest priority

system processes

interactive processes

interactive editing processes

batch processes

student processes

lowest priority