CPU Scheduling
Scheduling
Question

• What is the job of a CPU scheduler?

• What does preemption mean?

• Name a few scheduling algorithm?
Definitions

• Processes are also called jobs.

• Scheduling policy
  – A decision made by CPU as to which process to run next
Scheduling Metric

• Scheduling policies are compared with a metric
  – A way to measure something

• Turnaround time

\[ T_{\text{turnaround time}} = T_{\text{completion}} - T_{\text{arrival}} \]

\[ T_{\text{arrival}} = 0 \text{ (all jobs arrive at the same time)} \]
First In, First Out

Three jobs: A, B, C

• Assume each job runs for 10 seconds
  \[ \text{Average } T_{\text{turnaroundtime}} = \frac{10 + 20 + 30}{3} = 20 \]

• A Job runs for a different amount of time
  \[ \text{Average } T_{\text{turnaroundtime}} = \frac{100 + 110 + 120}{3} = 110 \]
Shortest Job First

The shortest job runs first.

Average $T_{\text{turnaround}} = \frac{(10 + 20 + 120)}{3} = 50$

Jobs do not arrive at the same time

Average $T_{\text{turnaround}} = \frac{(100 + (110 - 10) + (120 - 10))}{3} = 103.33$
Shortest Time-to-Completion First

• When a new job arrives, the remaining time for all jobs is computed.
• Job A is **preempted** to run jobs B and C.

\[
\text{Average } T_{\text{turnaroundtime}} = \frac{(120 - 0) + (20 - 10) + (30 - 10)}{3} = 50
\]
Response Time

\[ T_{\text{response}} = T_{\text{firstrun}} - T_{\text{arrival}} \]

• For the example shown in shortest time-to-completion first:
  Average \( T_{\text{response}} = (0 + 0 + 10) = 3.33 \)

• Response time needs to be short for interactive systems.
Round Robin (RR)

- RR runs jobs using a time slice.
  - Interrupts jobs every time slice units of time
- Too short time slice has high overhead (context switch)
- RR does not do well using the turnaround time.
  - Delays finishing jobs
- RR is fair (divides CPU time fairly among processes)

Average $T_{response} = \frac{0 + 5 + 10}{3} = 5$ sec  
Average $T_{response} = \frac{0 + 1 + 2}{3} = 1$ sec  
Average $T_{turnaround} = \frac{13 + 14 + 15}{3} = 14$ sec
Incorporating I/O

• When a job makes an I/O request, it is blocked.
• CPU can run another job.
• After completing I/O, CPU resumes the job.
• Example
  – Jobs A and B run for 50 ms.
  – Job A runs for 10ms and then makes an I/O request.

![Diagram showing CPU and Disk activity with timelines for CPU usage and I/O operation.](image)

CPU time is used poorly

Better use of CPU time through overlap
Process States

• Running: CPU executes process instructions.

• Ready: ready to run, but scheduler decides not to run it at a given time.

• Blocked: a process performs some operation (e.g., I/O) that makes it not ready to run. Other processes use CPU.
Process States

- **Running**
- **Ready**
- **Blocked**

States transition as follows:
- From **Running** to **Ready**: Descheduled
- From **Ready** to **Running**: Scheduled
- From **Running** to **Blocked**: I/O: initiate
- From **Blocked** to **Running**: I/O: done
### Process States

<table>
<thead>
<tr>
<th>Time</th>
<th>Process₀</th>
<th>Process₁</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Running</td>
<td>Ready</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Running</td>
<td>Ready</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Running</td>
<td>Ready</td>
<td>Process₀ initiates I/O</td>
</tr>
<tr>
<td>4</td>
<td>Blocked</td>
<td>Running</td>
<td>Process₀ is blocked, so Process₁ runs</td>
</tr>
<tr>
<td>5</td>
<td>Blocked</td>
<td>Running</td>
<td>I/O done</td>
</tr>
<tr>
<td>6</td>
<td>Blocked</td>
<td>Running</td>
<td>Process₁ now done</td>
</tr>
<tr>
<td>7</td>
<td>Ready</td>
<td>Running</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Ready</td>
<td>Running</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Running</td>
<td>–</td>
<td>Process₀ now done</td>
</tr>
<tr>
<td>10</td>
<td>Running</td>
<td>–</td>
<td></td>
</tr>
</tbody>
</table>
Question

• For the workload shown here, show the schedule timelines for
  – FIFO
  – SJF (non-preemptive)
  – STCF (preemptive)
  – RR (time slice = 1 second)

<table>
<thead>
<tr>
<th>Job</th>
<th>Arrival time</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>