CS 4410
Operating Systems

CPU Scheduling

Summer 2016
Cornell University
Today

- Scheduling algorithms
- Constraints
- Optimization criteria
Scheduling algorithm

• The OS should create the illusion that all threads produce work “at the same time”.
• The OS employs a scheduling policy.
• The scheduling policy determines which “ready” thread will use the CPU next.
• Some task constraints:
  – arrival time, deadline, priority,
  – CPU bound (i.e. matrix multiply) or I/O bound (i.e. text editor)?

• Some resource constraints:
  – number of resources,
  – can we preempt resources?
Input of a scheduling algorithm for 1 CPU.

- The input usually does not satisfy all constraints.
  - Here, the constraint of 1 CPU is violated, because a vertical line may cut more than 1 tasks.
- The output of a scheduling algorithm should satisfy all constraints.
Many ways to schedule tasks and satisfy constraints

Input

Constraints:
• arrival time (cannot schedule a task before it is arrived),
• 1 CPU,
• CPU can be preempted (the algorithm is able to preempt the CPU),
First In First Out (FIFO)

Input

Output

Time
Shortest Job First (SJF)

Input

Output

Unrealistic Assumption:
Duration for each task is known!

Use of preemption!
Round Robin (RR)

Input

Output

Quantum:
Which algorithm is the best?

It depends on the criteria we want to optimize for!
Optimization Criteria

• CPU utilization
• Throughput
• Turnaround time
• Waiting time
• Response time
• Lateness
• Simplicity
• Energy consumption
• Starvation freedom
• Low overhead

Can we have it all?
Which algorithm is the best for ...?

- **FIFO**
  - Simplicity
  - Low overhead

- **SJF**
  - Lateness
  - Turnaround time

- **RR**
  - Response time
  - Starvation freedom
RR: Choice of Time Quantum

• Too short quantum:
  – Better responsiveness.
  – Preferred by I/O-bound tasks.
  – Increased overhead (due to context switching).

• Too long quantum:
  – Reduced overhead.
  – Preferred by CPU-bound tasks.
  – Worse responsiveness.

• Usually operating systems pick a quantum between 10 and 100 ms.

• It depends on the criteria we want to optimize for!
Turnaround Time w/ Time Quantum
Multi-level Feedback Queue (MFQ)

• Different quanta are suitable for different types of tasks, in order to achieve:
  – better responsiveness and
  – lower overhead,
on average.

• MSQ “learns” a suitable quantum for each task.
MFQ

<table>
<thead>
<tr>
<th>Priority</th>
<th>Time Slice (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>80</td>
</tr>
</tbody>
</table>

Round Robin Queues:
- New or I/O Bound Task
- Time Slice Expiration
A Multi-level System

- I/O bound jobs
- CPU bound jobs

Priority vs. Timeslice diagram
Multiprocessor Scheduling

• Additional constraints that may be considered:
  – N>1 CPUs
  – Affinity: each task is always scheduled at the same CPU.
  – Groups: all the tasks of a program should be scheduled together.
Today

- Scheduling algorithms
- Constraints
- Optimization criteria
Coming up…

• Next lecture: Synchronization