Lecture 4: Task Scheduling

- **Schedulable**: a task always meets its timeliness constraints.

- **Deterministically schedulable**: a task can be guaranteed to always meet all deadlines (always schedulable)
  - Worst case response time $\leq$ task’s deadline

- All tasks are schedulable $\Rightarrow$ system is schedulable
L4: Task Scheduling

CPU Utilization:

• Ideally: CPU is close to 100% of time busy. (Why?)
• In practice, that won’t happen.

• For a periodic task $i$: $u_i = \frac{e_i}{p_i}$

  $u_i$ fraction of time task keeps CPU busy
  $e_i$ execution time
  $p_i$ period
L4: Task Scheduling

*When to schedule?*

1. A process goes from running to waiting (e.g. I/O request).
2. When a running process must go into ready state (e.g. an interrupt occurs).
3. A waiting process becomes ready (e.g. I/O activity completed).
4. A process ends.

- 1 and 4 do not require preemption. Non-preemptive scheduling: a process keeps CPU until termination or switching to waiting state.
- If 2 and 3 are also used: preemptive scheduling.
L4: Scheduling Criteria

Simple Priority

• The highest priority task among those ready will be selected to run.
• In preemptive systems: lower priority tasks running are preempted by higher priority tasks that may arrive.

i. Priority of Task A > Priority of Task B
   ➞ Task A is running until it requires a resource. If Task B “owns” such resource, ...
   ... it will still run despite the two priorities ➞ Blocking.

ii. Three tasks: A, B, and C (A > B > C)
If A and C require a resource, but B doesn’t, ...
B may end up running more than A or C ➞ Priority Inversion.
L4: Scheduling Criteria

Turnaround Time
Total interval from submission until completion (including switching time, waiting time, etc).

Throughput
# of processes that are completed per unit of time. It depends on the complexity of the task.
A high throughput with a long turnaround may lead to unattained time deadlines.

Waiting Time
Tasks spend time on different queues (entry, ready, I/O, device).

Response Time
For interactive systems, what counts is the first response, not the task completion (turnaround).
Next time...

Scheduling Algorithms (read sec. 12.3)

• Asynchronous interrupt event driven
• Polled
• Polled with timing element
• State Based
• Synchronous interrupt event driven
• Combined interrupt event driven
• Foreground-Background
• Time Shared
• Priority schedule