CPU Scheduling, continued

CSC 213 – October 7, 2022
Agenda for Today

1. Department Events
2. Context Demo
3. Q&A
4. Real Schedulers: MLFQ, Lottery, Stride, and CFS
5. Wrap Up
Department Events

CS Table 12pm on Tuesday, October 11 in JRC 224A
Surfing the Web with Netscape Navigator

CS Extra 4:15pm on Thursday, October 13 in Noyce 3821
Spatial AI and its Applications
Professor Yao-Yi Chiang from the University of Minnesota
(snacks available in the commons at 4pm)
Lab Questions

When a task is finished, what do we do?
   Record that the task has exited and do not run it again.
   The space in the tasks array that holds that task won't be reused.
   You need to find another task to run and swap to it.

How do we deal with exit_context?
   Don't. It has to be there to ensure task_exit() runs when a task finishes, but you
   shouldn't do anything with it directly.

Do we have to worry about task priority?
   No. Make sure your scheduler code checks all other tasks as candidates to
   run. There's no priority. Just don't miss a task.
Lab Questions

When the `generate_apples` task creates an apple, is it done?
No, it just blocks by calling `task_sleep` for two seconds.

What do we do if there's no task to switch to?
You have to just waste time until a task becomes runnable. You can do that by looping over the array of tasks, or you could be "polite" and call `sleep_ms` to pause briefly.
Real Schedulers
Multi-Level Feedback Queue

How would you describe MLFQ to a 161 student?
A bunch of queues with weighted priorities.
New tasks are placed in the highest priority queue, which has a short scheduler quantum.
Any task that uses up its quantum is demoted to a lower priority queue.
Periodically, we boost all tasks up to the top priority queue.
Lottery Scheduling

How would you describe lottery scheduling to a 161 student?
Each job has a number of lottery tickets. The more tickets a job has, the higher priority it is. Each job can also assign its own tickets to a sub-task to share its time. When the scheduler needs to choose a task to run, it picks a random lottery ticket.
Stride Scheduling

How would you describe stride scheduling to a 161 student?

Proportional share scheduler like lottery, but it is not randomized.
Just like with lottery, giving a task double priority gives it double time.
Because the scheduler isn't randomized, a task can't end up being "unlucky". In other words, a stride scheduler is guaranteed to be fair over a shorter time scale, unlike a lottery scheduler.
Aesthetics

Which scheduler (MLFQ, Lottery, Stride) do you like? Why?

MLFQ: 8
Lottery: 2
Stride: 7
Wrap Up
Reminders

Reading
Read about *Concurrency* and the *Threads API* for Monday

Lab
The worm lab is due next Wednesday, or with a late penalty on October 26. You can still turn in the malloc lab with a late penalty by October 12.

Mentor Office Hours
Sunday 1–4pm
Monday 8–9pm
Tuesday 8–9pm