

Grading Rubric: COS 318 Project 3

Fall 2013

This project is worth a total of **10 points**, plus 1 point extra credit.

We have tests created for grading each section of the rubric. Please make sure to test your submitted files in the Dropbox to check if they compile without errors or warnings. You do **not** need to test your kernel by booting off a USB.

A: Preemptively scheduled processes. 2 pts.

- A01: Processes can be preempted. 1 pt.
- A02: Preemption preserves all registers. 1pt.

B. Blocking sleep. 2 pts.

- B01: Processes block on sleep. 0.5 pt.
- B02: Threads block on `do_sleep`. 0.5 pt.
- B03: Scheduler survives when all processes are sleeping. 0.5 pt.
- B04: Sleep counts time correctly, Specifically, if your code passes the `test_blocksleep` case with some acceptable error, it should be fine. 0.5 pt.

C. Condition Variables. 2 pts.

- C01: Condition signal wakes one. 0.5 pt.
- C02: Condition broadcast wakes all. 0.5 pt.
- C03: Producer/consumer works when building a queue using your `CV.signal`. 0.5 pt.
- C04: Producer/consumer works when building a queue using your `CV.broadcast`. 0.5 pt.

D. Semaphores. 2 pts.

- D01: Producer/consumer works when building a queue using your semaphores. 1 pt.
- D02: Semaphores are fair. 1 pt.

E. Barriers 2 pts.

- E01: Barriers successfully wait for all processes. 1 pt.
- E02: Barriers are automatically renewed. 1pt.

X. Extra Credit. Prioritized task scheduling 1 pt.

- X01: Priority with preemption. Tasks should be scheduled according to their priorities, and without starvation. Specially, if you have task A with priority 100, task B with priority 200, task B should be scheduled roughly twice as many as task A. We accept a fairly wide error here, say, $\pm 15\%$. Besides, if you have task A with priority 100000, task B with priority 1, you should make sure task B somehow will get a chance to run. 0.5 pt.
- X02: Priority with yield. Similar to X01, except that tasks would yield at some points. 0.5 pt.

Additionally, we reserve the right to remove as much as 1 point for submissions which are extremely confusing, obfuscated, or blatantly do not follow the given API (this does not include helper functions you added). Please write simple, readable code with comments, but don't leave unnecessary debugging print statements in your code. Don't forget to attach a brief README in text format.

Recall that this class has a firm late submission policy, which is detailed on the COS 318 website.