Grading Rubric: COS 318 Project 3
Fall 2015

This project is worth a total of **12 points**, plus 1 point extra credit.

We have tests created for grading each section of the rubric. Please, make sure that your project compiles without errors or warnings. You do not need to test your kernel by booting off a USB.

DR: Correct descriptions of the main concepts of the project. **2pts.**

A: Preemptively scheduled processes. **2pts.**
1. A01: Processes can be preempted. **1pt.**
2. A02: Preemption preserves all registers. **1pt.**

B: Blocking sleep. **2pts.**
1. B01: Processes block on sleep. **0.5 pt.**
2. B02: Threads block on do_sleep. **0.5pt.**
3. B03: Scheduler survives when all processes are sleeping. **0.5pt.**
4. B04: Sleep counts time correctly. Specifically, if your code passes the test_blocksleep case with some acceptable error, it should be fine. **0.5pt.**

C: Condition variables. **2pts.**
1. C01: Conditional signal wakes one. **0.5pt.**
2. C02: Conditional broadcast wakes all. **0.5pt.**
3. C03: Producer/consumer works when building a queue using your CV.signal. **0.5pt.**
4. C04: Producer/consumer works when building a queue using your CV.broadcast. **0.5pt.**

D: Semaphores. **2pts.**
1. D01: Producer/consumer works when building a queue using your semaphores. **1pt.**
2. D02: Semaphores are fair. **1pt.**

E: Barriers. **2pts.**
1. E01: Barriers successfully wait for all processes. **1pt.**
2. E02: Barriers are automatically renewed. **1pt.**

X: Extra Credit. Prioritized task scheduling. **1pt.**
1. X01: Priority with preemption. Tasks should be scheduled according to their priorities, and without starvation. Specifically, if you have task A with priority 100, task B with priority 200, task B should scheduled roughly twice as many as task A. We accept a fairly wide error here, say, ±15%. Moreover, 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. **0.5pt.**
2. X02: Priority with yield. Similar to X01, except that tasks would yield at some points. **0.5pt.**

Additionally, we reserve the right to **remove as much as 1 point** for submissions which are extremely confusing, obfuscated or overcomplicated. Please write simple, readable code with comments, but don't leave unnecessary debugging print statements in your code. Dont 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.