1. Purpose

This homework builds understanding of scheduling algorithms by comparing the Linux 2.6 kernel method to a more simplistic approach.

2. Description

The goal of this homework is to build a simulator that will allow you to compare scheduling methods. Given a fixed number of threads, your simulation will run until all threads have terminated. You will use the statistics gathered to generate a report of your findings.

System Assumptions

1. Single CPU
2. Single I/O operation possible (e.g., keyboard input)
3. All I/O operations return in a fixed amount of time (e.g., 500ms)
4. Context switches are instantaneous
5. Threads (not processes) are scheduled and are independent
6. Preemption is possible
7. The base quantum is 5msec

Thread Assumptions

1. Total number of threads fixed and known when simulation starts
2. Threads terminate individually after uniformly randomly generated period of CPU time
3. Each thread has probability $p$ of completing a 5msec slice without I/O blocking

Desired Metrics

1. Total/average turnaround time
2. Ready percentage (time spent in ready queue divided by total time thread is alive)

Suggested Scheduling Infrastructure

- Use processor clicks (1ms = 1 click) to advance time during the simulation
• Maintain one structure of *ready* threads and one *wait* structure for those currently blocked for I/O.

• Algorithm
  - Pick thread *x* to run and assign it a quantum
  - Use its probability *p* to determine whether *x* will run its entire quantum
    - If not, uniformly randomly generate the amount of time before I/O block
  - Check *wait* to see if I/O will complete before *x* either blocks or finishes its time-slice.
  - Advance time until I/O preemption, block, or *x*’s time-slice expires
  - Repeat

**Scheduling Algorithms**

**Round-robin**
- No priorities
- Threads that come off the *wait* queue should be added to the end of the *ready* queue

**Simplified Linux**
- Use basic runqueue structure presented in class
- Priorities
  - Range from 0-39 and are available to all thread
  - Priority 0 (highest) has a quantum of 800ms
  - Priority 39 has quantum of 5ms
  - Each thread starts with priority 19
- Adjusting priorities
  - If a thread uses all of its assigned quantum, lower its priority by one
  - If a thread does not use its assigned quantum, raise its priority by one

3. **Report**

Your report (one per group) should contain a description of your simulator and anything interesting you learned while developing it. It should also contain:

1. Experiments that establish the performance of your simulator (e.g., number of threads it can schedule in a reasonable amount of time, time to completion, etc.)
2. Comparison of round-robin to simplified Linux under various conditions using the desired metrics.
   a. Equal parts I/O bound and CPU-bound processes
   b. Largely I/O bound processes
   c. Largely CPU-bound processes
   d. Any other of interest
3. Comparison of metrics for I/O bound (interactive) and CPU bound processes
4. Given your analysis, suggested improvements to either algorithm.