CS314 - Operating Systems Homework 1 (Fall '23)

1 Administrivia

Homeworks should be done individually and turned in via Moodle. LaTeX version of this file is here: `~/Public/cs314/hw1.tex` To generate a pdf with your solutions copy the above file to someplace in your home directory and compile with `pdflatex hw1.tex`

2 Get On With It

1. (20 pts) Rewrite the example from the "Dining Philosophers" lecture slide to working C code. Write a main() that creates one for each philosopher. The number of philosophers should be passed in via command line argument. Demonstrate that your solution works by printing something when each philosopher changes state. Let the program exit when each philosopher has eaten at least twice. Turn in code and output.

2. (80 points) Implement semaphores using the pthread library. Consult the thread implementation in DLXOS for inspiration (semaphores are implemented in synch.h and synch.c).
   Use `pthread_mutex_lock` and `pthread_mutex_unlock` (see man pthread_mutex_lock) to make the semaphore wait and post operations atomic.
   Use `pthread_cond_wait` (see man pthread_cond_timedwait, but don't use pthread_cond_timedwait) and `pthread_cond_signal` (see man pthread_cond_signal).
   Test your solution by substituting the pthread semaphores in your first solution with your own semaphores.

   Note: Define your own sem_signal and sem_wait functions. **not the ones found in semaphore.h. DO NOT INCLUDE semaphore.h**.

   A general solution will include a semaphore struct like

```
struct Sem{
int count;
pthread_cond_t cond;
pthread_mutex_t mutex;
};
```

3. (practice) Five jobs (A, B, C, D, and E) arrive in alphabetical order, in the following quanta: 1, 3, 3, 6, 10, respectively. They have estimated running times of 4, 5, 5, 3, and 8 minutes. Their priorities are 5, 4, 3, 2, and 1, respectively, with 5 being the lowest priority. For each of the following scheduling algorithms, determine the mean process turnaround time. Ignore process switching overhead and assume a quantum of 1 minute. Show your work for partial credit.

    - Round robin.
    - Priority scheduling.
    - First-come, first-served
    - Shortest-job first.

4. (practice) The aging algorithm with $a = 1/2$ is being used to predict run times. The previous four runs, from oldest to most recent, are 8, 12, 10, and 14 msec. What is the sequence of predicted runtimes starting with the first of the observed runs (8), ending with the predicted runtime after the last observed run (14). (See Tanenbaum's section on Scheduling in Real-Time Systems, Ch. 2)

5. (practice) A real-time system needs to handle one voice call that runs every 10 msec and consumes 4 msec of CPU time per burst, plus a video at 32 frames/sec, with each frame requiring 20 msec of CPU time. Is this system schedulable? Explain.

3 What to Turn In

Turn in one file, preferrably a tgz or zip, containing the answers to the questions above, along with all working code and/or scripts, where applicable.