CompSci 2SD3 Tutorial #8

TA: Jatin Chowdhary

DATE: March 21st, 2022

Announcements (1)

- Midterm #2
  - How was it?
    - Joke, Easy, Okay, Hard, Impossible?
    - Did you even study?
  - How was the review material?
    - Do you want more questions? Less?
    - Did the material help?
  - Is there anything you struggled with?
    - Any question in particular you want to takeup?

Announcements (2)

- Marks
  - Now that midterm #2 is over, I don't need to prepare questions, slides, etc.
    - Meaning, I can mark the assignments.
    - Hopefully the time spent into making midterm review questions was worth it
    - Do not want you going into the exam thinking, "*Am I going to pass this course*?"
- ETA:
  - Assignment #1: Thursday
  - Assignment #2: Saturday
  - Midterm #2: Sometime next week
    - But y'all already know how you did

Announcements (3)

- Midterm Marking
  - I'll be marking Question #9
    - Can't wait to read the same answer, 250 times
    - Rather, can't wait to read my answer, 250 times
  - Question #10 will be marked by Haotian
  - Multiple choice questions are marked by Dr. Franek
    - Same setup as last time

Announcements (4)

- Final Exam
  - Will be much harder
    - Expect something on the magnitude of midterm #1
  - Right after classes end
    - April ~14th, 2022
  - In-person!
  - What is the best way to motivate y'all?

Announcements (5)

- Assignment #3
  - About threads
    - More *pthreads*
  - Skeleton file provided by Dr. Franek
    - Pseudocode provided by Zhonguyan and Haotian
  - Due on Monday, March 28th, 2022
    - Next week

Any Questions?

Outline

- Review Midterm #2
  - Only interesting questions
- Review Assignment #3
  - Basics
  - How-to
  - Getting started
  - Other stuff
- *End tutorial at 3:15*

Midterm #2 Question #2

- **Question:** Consider a round-robin scheduling of jobs on a CPU with a simplification that each job is assigned quantum with the same length (ie., all jobs get assigned the same quantum). Which of the following statements is most appropriate?
- **Options:**

  A. The shorter the quantum, the faster the jobs are executed.

  B. The shorter the quantum, the more jobs can be executed.

  C. The shorter the quantum, the more responsive the jobs are.

  D. All of the above.

  E. None of the above.

Midterm #2 Answer #2

- **Question:** Consider a round-robin scheduling of jobs on a CPU with a simplification that each job is assigned quantum with the same length (ie., all jobs get assigned the same quantum). Which of the following statements is most appropriate?
- **Options:**

  A. Incorrect; The scheduling algorithm has nothing to do with how fast jobs are executed. *What determines the speed of execution?*

  B. Incorrect; Shortening the time quantum does not allow more jobs to be executed. However, it does allow jobs to quickly receive CPU time. Which makes the jobs more responsive.

  C. Correct; The shorter the quantum, the more responsive the jobs are. If the time quantum is small, then jobs are quickly cycled through which increases responsiveness.

  D. Incorrect

  E. Incorrect

Any Questions?

Midterm #2 Question #3

- **Question:** Consider the the program *question3.c* (we omitted the header files and stuff related to using log functions for simplification):
- **Options:**

  A. got 2; got 2; terminating the process.

  B. got 2; got 3; terminating the process.

  C. got 3; got 3; terminating the process.

  D. It cannot be determined, but it will be either A or C.

  E. It cannot be determined, but it will be either B or C.

- **Answer:**

Any Questions?

Midterm #2 Question #5

- **Question:** Consider the the program *question5.c* (we omitted the header files and stuff related to using log functions for simplification). Which scenario, given below, will we see on the screen?
- **Options:**

  A. Scenario A, because the critical section of *doit()* starts with *pthread_mutex_lock(&flock);* and ends with *pthread_exit(NULL);*. Thus each thread is "protected" during the "conversation" with the user.

  B. Scenario B, since the critical section of *doit()* starts with *pthread_mutex_lock(&flock);* and ends with *pthread_exit(NULL);* and thus the mutex *flock* is never unlocked. Thus the first thread will lock *flock* and the second thread will wait forever.

  C. It cannot be determined, it depends on the timing of events. If the first thread terminates before the second thread tries to lock *flock*, everything will be OK and we will see Scenario A, otherwise we will see Scenario B.

  D. We will see neither of the given scenarios.

- **Answer:**

Any Questions?

Midterm #2 Question #7

- **Question:** Consider the HRRN (Highest-Response-Ratio-Next) scheduling system. It is based on the simple SJF (Shortest-Job-First) scheduling model, but it is designed to improve it. What does it improve?
- **Options:**

  A. In comparison to SJF, HRRN improves the responsivness of the jobs.

  B. In comparison to SJF, HRRN improves the turn-around of the jobs (so more jobs can be run).

  C. In comparison to SJF, HRRN improves the chances of longer jobs to get a turn and thus prevent so-called “starvation”.

  D. All of the above.

  E. None of the above.

- **Answer:**

THE END

(Stop Recording & Start New One)