Exam 1: Processes, CPU Scheduling, & Synchronization

Exam Instructions:

- Put your name and student number on the exam books NOW!
- The exam is closed book and closed notes.
- You have 75 minutes to complete the exam. Be a smart test taker, if you get stuck on one problem go on to the next. Don’t waste your time giving details that the question does not request.
- Show your work. Partial credit is possible, but only if you show intermediate steps.
- Good luck.

1. (25 pts) Short answer questions.

   (a) (5 pts) You have a threaded application that would benefit from a different scheduling algorithm than the one your operating system implements. Would you prefer to use kernel or user threads to implement your scheduling algorithm and why?

   (b) (5 pts) Which of the scheduling algorithms we covered in class is the fairest? What effect does fairness have on response time?

   (c) (5 pts) Compare using the Test&Set instruction versus disabling and enabling interrupts to implement locks (or any other high-level synchronization construct).

   (d) (5 pts) Will a monitor that executes correctly using Mesa-style semantics execute correctly if Hoare-style semantics are used instead. Why or why not?

   (e) (5 pts) If we want to use semaphores within a monitor instead of condition variables, what problems might this cause and how could we solve them?

2. (25 pts) Scheduling.

   (a) (5 pts) Name the five scheduling algorithms we considered in class.

   (b) (10 pts) Pick 2 scheduling algorithms and a job mix with 2 or more jobs that have the same average response time.

   (c) (10 pts) Pick another 2 scheduling algorithms and a job mix with 2 or more jobs that yield the same average response time. (Note, you can only overlap at most one scheduling algorithm with part b.)
3. (25 pts) **Processes & Threads.**

   (a) (5 pts) Why is a context switch faster for user threads than for kernel threads?

   (b) (10 pts) What happens on a context switch between independent processes? (In your solution, discuss all the affected OS data structures and include a drawing of the process state diagram and discuss the possible transitions.)

   (c) (10 pts) Is there any reason to use threads if the scheduler is non-preemptive (i.e., context switches may occur only inside blocking synchronization primitives, in response to an explicit request to yield the processor, or in response to an I/O request)?

4. (25 pts) **Semaphores & Monitors.** You are given a game with many players and 3 colors (red, blue, green). You must ensure that the players only move pieces in the order: red, blue, green, red, blue, green, etc. Write three routines for the players to call: MoveRed, MoveBlue, and MoveGreen. Remember to write the initialization routines, and you may assume red always starts.

   (a) (10 pts) Write a semaphore solution.

   (b) (15 pts) Write a monitor solution.