Midterm Exam

Write the answers to question 1 on this sheet. Write the other answers in the exam booklet. Give short answers where possible.

1. True/False. Circle the appropriate choice. (1 point each)
   
   (a) T F  “Device-independent I/O software” is software that doesn’t care what brand of CPU (Intel, IBM, Motorola, etc.) that it’s running on.
   
   (b) T F  On a spinning disk, the outer tracks are moving faster than the inner tracks.
   
   (c) T F  A printer daemon is generally responsible for spooling print jobs.
   
   (d) T F  Relocation isn’t necessary on monoprogramming systems.
   
   (e) T F  The dining philosophers problem is fundamentally a problem of mutual exclusion, preventing two philosophers from picking up the same chopstick at the same time.
   
   (f) T F  This course is primarily about Pascal programming.
   
   (g) T F  An OS provides libraries of interrupt handlers for the users to call.
   
   (h) T F  The problem with using lock variables (i.e. regular integer variables whose value, either 1 or 0, determines if a critical section is locked or not) to avoid race conditions is that they are themselves subject to race conditions.
   
   (i) T F  With shortest-job-first scheduling, the time required to finish all the jobs is less than for first-come-first-serve scheduling.
   
   (j) T F  A quantum (for round-robin scheduling) is one clock tick.

2. Processes (15 points)

   (a) Suppose you were implementing an OS that used a round-robin scheduler with a quantum of 50 milliseconds. Suppose also that the computer had a timer (i.e. clock) chip that used a crystal that sent a signal causing the counter on the chip to be decremented at 100 MHz. Describe (being specific about numbers) how the clock driver would support the desired quantum on this machine.

   (b) Why does a CPU need a kernel (aka supervisor) mode and a user mode? How does the CPU switch modes?

   (c) A potential problem with a multiprogramming system that supports swapping is that a context switch might waste CPU time because, when a process is scheduled to execute, the CPU might have to wait for the process’s image (i.e. the code, data, and stack sections) to be brought into memory from the disk. If you were writing an operating system, how would you design the scheduler and memory manager to avoid the CPU having to wait?

3. Race Conditions (10 points)

   (a) Give an example (in C or Pascal) of a race condition.

   (b) Fix the race condition in your example using a binary semaphore and then fix it using a TSL (test-and-set lock).

   (c) Describe in detail (including what the hardware does) how you would implement binary semaphores if you were writing an OS. Be sure to describe the implementation of the down() and up() operations.

4. Bankers Algorithm (15 points)

   (a) Describe the idea behind the banker’s algorithm for avoiding deadlock (for a single class of resources).

   (b) Give an example of a safe state and an example of an unsafe state.

   (c) In code, pseudocode, or detailed prose, give the algorithm itself.