Introduction to Operating Systems
Midterm Examination

Name:____________________________ E-mail:____________________________

Spring 1999

There are fifteen questions in the following examination. All have equal weight (five points). Answer them clearly and *concisely* (hot air is worth *zero* points). Some questions require only a short answer (a few sentences), others require a little more thought.

1. In class it was asserted that operating systems provide pleasant illusions to the programmer. One of the most important is the process abstraction. Define a *process* and (briefly) say why is useful to provide the illusion to every program that it is the only program running on the computer.

   Extra credit: For an extra three points, contrast the concept of *process* with that of a *virtual machine.*

2. A *microkernel* is an operating system technique that places minimal functionality in the kernel, but instead provides that functionality through user-state servers. What is a major performance obstacle that microkernel designers must face?

3. A *context-switch* is when the processor (CPU) stops executing one process and begins executing another. List the necessary steps for the operating system to perform a context-switch, including what information must (typically) be saved.

4. Briefly describe why *round robin* scheduling is appropriate for an interactive environment, while *non-preëmptive priority* scheduling is not.

5. Disabling interrupts is a common method for the operating system to ensure mutual exclusion on a uniprocessor. Is it appropriate for a shared memory (MIMD) multiprocessor? Why or why not?

6. Disabling interrupts is inappropriate for *user mode* and is only acceptable for brief periods in kernel mode on uniprocessors. Suppose you have a processor (CPU) like the MIPS R3000 that has no *test-and-set* instruction. How would you go about providing for mutual exclusion?

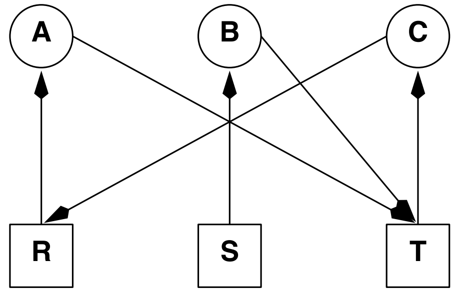
   Hint: You do not need to detail an algorithm, only mention it by name.

7. It can be shown that with a shared memory model *semaphores* and *monitors* are equivalent (one can implement the other). Using a high-level pseudo-code, show how semaphores can be used to implement monitors.

   For the purposes of this question, you may ignore *condition variables*.

8. Deadlock can be prevented by negating any one (or more) of four necessary and sufficient conditions (Coffman 1971). List the four conditions, and briefly describe the consequences of negating each of them.

9. Consider the following resource graph. Which processes (A, B, C) and which resources (R, S, T) are deadlocked? What needs to be done to break the deadlock? Briefly explain how you knew that.



10. In terms of deadlock avoidance, there are three states: *safe*, *unsafe* and *deadlocked*. Briefly define each of these states, and say why it is unwise to enter the *unsafe* state.

11. There are many policies for variable-sized partition memory management. In a sentence or two each, describe and rank (in terms of wasted memory): *first-fit*, *worst-fit* and *best-fit*.

12. The five (basic) states that a process can be in are: *new*, *ready*, *blocked*, *running* and *terminated*. Describe the transitions among these states, and state when they occur.

    Hint: The best way to do this is to draw a diagram.

13. Using the reference string ⟨2 1 0 3 2 1 4 2 1 0 3 4⟩, fill in the two tables below (representing systems with *three* and *four* page frames, respectively) using the FIFO page replacement policy. Indicate when a page fault occurs, and note how many page faults occur. Do you notice any anomalies?

14. Using the same reference string as in the previous question, fill in the following two tables as before using the LRU page replacement policy. Indicate when a page fault occurs, and note how many page faults occur.

15. The *inclusion property*, written $\mathcal{M}(m, r) \subset \mathcal{M}(m+1, r)$ states that any page in a memory of size m will also be in a memory of size $m+1$. Give one page replacement policy which *does* have this property, and one page replacement policy that *does not* have it (approximately is *not* good enough). What can happen if it is does not apply to a given page replacement policy?

Mid-Course Correction

It's time to consider a mid-course correction. I would like to know how you feel about the course. Please take a few minutes to tell me what you think. Don't put your name, just tear this sheet off and turn it in with the test.

I'm afraid that dropping *Nachos* is not possible this quarter, but if you have an alternative suggestion for next time, I would like to hear about it.

1. Does the instructor tell too many jokes?
2. Is the course going too fast? Too slow? Just right?
3. Should the instructor stick closer to the book?
4. Does the instructor cover material in enough detail? Too much?
5. Should the instructor talk more about current research?
6. What should the instructor do to make this course more useful to you?
7. Feel free to let the instructor know how you feel he should change the course. Say as much (or as little) as you like.