University of California, Berkeley
College of Engineering
Computer Science Division – EECS

Spring 2008 Anthony D. Joseph

Midterm Exam

February 27, 2008
CS162 Operating Systems

Your Name:	
SID AND 162 Login:	
TA Name:	
Discussion Section Time:	

General Information:
This is a **closed book and notes** examination. You have 90 minutes to answer as many questions as possible. The number in parentheses at the beginning of each question indicates the number of points given to the question; there are 100 points in all. You should read **all** of the questions before starting the exam, as some of the questions are substantially more time consuming.

Write all of your answers directly on this paper. ***Make your answers as concise as possible.*** If there is something in a question that you believe is open to interpretation, then please ask us about it!

Good Luck!!

Problem	Possible	Score
1	33	
2	23	
3	17	
4	27	
Total	**100**	

1. (33 points total) Short answer questions:

a. (4 points) True/False and Why?

In the operating system environments that we've studied so far, an application programmer has to worry about deadlock occurring involving the CPU as a resource.

TRUE FALSE

Why?

b. (4 points) True/False and Why?

In the operating system environments that we've studied so far (i.e., no virtual memory), the operating system has to worry about deadlock occurring involving physical memory as a resource.

TRUE FALSE

Why?

c. (5 points) Disabling interrupts:

i) (3 points) Give a two to three sentence description of how interrupts can be used to implement critical sections.

ii) (2 points) Briefly (2-3 sentences) state the problems with implementing critical sections using disabling of interrupts.

d. (8 points) List the four requirements for deadlock.

e. (4 points) Briefly explain why when using a Hoare semantic monitors it is sufficient to use “if” statements when checking state variables, while most OS monitor implementations require the use of “while” statements.

f. (8 points) In lecture, we discussed using atomic instructions (e.g., test-and-set and swap) to implement spinlocks. With spinlocks, threads spin in a loop (busy waiting) until the lock is freed. We motivated the use of blocking locks and semaphores as an improvement over spinlocks because having threads spin can be very inefficient in terms of processor utilization. However, spinlocks are not always less efficient than blocking locks. In two to three sentences, briefly describe a scenario where spinlocks would be more efficient than blocking locks.

2. (23 points total) Implementing and Using Synchronization Primitives.

a. (6 points) While working on the first project, one of your project partners proposes to use a modified version of Semaphore to implement the condition variable class as follows:

```
Semaphore.set(0);         // Initialize semaphore to 0
Wait(Lock lock) {
     Semaphore.P(lock);   // Special semaphore that
                          // releases lock if it waits
}

Signal() {
     Semaphore.V();
}
```

Decide whether you think their proposal will work or not, and briefly, in two to three sentences, explain your decision

b. (9 points) Another one of your project partners proposes the opposite – to implement semaphores using an unmodified Mesa-style Monitor class. Show a possible implementation of a Semaphore class using monitors. You should provide the initialize(value), P(), and V() operations.

c. (8 points) Nested Monitors.

i) (5 points) In this question, we'll consider the effects of nesting two monitors, M1 and M2:

- Inside the code in Monitor M1 that manipulates M1's state variables, there is a procedure call into Monitor M2
- M2 then waits on one of its condition variables (releasing the lock for Monitor M2).

Assuming that M2 does not call any other monitors, including M1, are there any problems with nesting monitors in this manner? Use a simple example to explain your answer.

ii) (3 points) Suppose that the wait in M2 also releases the lock for Monitor M1. Does your answer for part (i) change or not? Briefly explain your answer.

3. (17 points total) Concurrency problem: Dining Graduate Students.

The Dining Graduate Students problem is as follows. Six graduate students are seated around a table with a large deep dish pizza in the middle. Graduate students are very refined and so they eat pizza with forks and knives. But they don't have a lot of money, so they have the utensils. There are three forks and three knives in a pile next to the pizza. Each student uses the following algorithm to eat:

(1) Pick up a knife
(2) Pick a fork
(3) Cut out a slice of pizza and eat it
(4) Return the knife and fork to the pile

a. (4 points) Specify the correctness constraints. Be succinct and explicit in your answer.

b. (5 points) Can deadlock ever occur? Explain your answer in terms of the conditions for deadlock to occur. If deadlock could occur, describe a reasonable deadlock avoidance algorithm. If deadlock cannot occur, explain why not.

c. (8 points) Implement the `Dine()` method using *only* semaphores (your solution may not use locks, monitors, or other synchronization primitives). Create a method `Dine()`, which waits until a student has fork and knife and can eat, then calls `Eat()`, and then releases the utensils before returning. Your solution should allow multiple students to eat at the same time (as long as there are sufficient utensils in a pile in the middle of the table). Assume you are given the variables, `forks` and `knifes`, which each starts off initialized to the total number of forks and knives available, respectively. *Based on your answer from part 3.b, your solution should avoid deadlock.*

4. (27 points total) Operating Systems.

a. (6 points) What are the two roles of an Operating System. For each role, briefly (one to two sentences) explain the role.

i)

ii)

b. (5 points) What private (thread-private) and public (shared between cooperating threads) resources are allocated when a thread is created? How do they differ from those allocated when a process is created?

c. (10 points) Draw the life cycle of a thread and label all the transitions:

d. (6 points) We discussed the Therac-25 in lecture and in an assigned reading.

i) What is the Therac-25, and how does it differ from the Therac-20?

ii) Explain the Therac-25's problems and the underlying causes.

No Credit – **Problem X** (000000000000 points)

A 2007 Darwin Award Runner Up

> "Named in honor of Charles Darwin, the father of evolution, the Darwin Awards commemorate those who improve our gene pool by accidentally removing themselves from it." (http://www.darwinawards.com/)

The Laptop Still Works!

[Confirmed True]

(26 February 2007, California) 29-year-old Oscar was driving on Highway 99 near Yuba City, when his Honda Accord crossed into oncoming traffic and collided with a Hummer. The occupants of the Hummer were not seriously injured. California Highway Patrol officers found Oscar's laptop still running, and plugged into the car's cigarette lighter. Investigators believe that he was using it when his car crossed the center line.

"Driving is not a time to be practicing your multitasking skills," remarked CHP spokesman Tom Marshall.

Oscar is not alone. Last year, 510 California drivers were charged with reckless driving because they were using a TV, video, or computer monitor. A 2001 CHP study cites cell phone use as the top cause of crashes involving distracted drivers, followed by fiddling with music. "Anything that distracts you can kill you, whether it's eating lunch or working on a computer," an AAA spokesman said.

Oscar was a computer tutor. Hopefully his fatal lesson will teach others to surf on the information superhighway, not the asphalt superhighway.

This page intentionally left blank