COSC 4330SECOND MIDTERMMARCH 31, 2003

This exam is **closed book**. You can have <u>one</u> page of notes. UH expels cheaters.

- 1. For each of the statements below, indicate in one sentence whether the statement is true or false (2 points), and why (3 points).
 - a) There are some problems that you can solve with *semaphores* but not with *monitors*.

FALSE, as semaphores could be implemented on the top of monitors.

b) The *notify* monitor primitive is easier to use than the *signal* primitive.

TRUE, as you can put it anywhere in your monitor procedures without fearing to have to release the monitor inside a critical section.

c) UNIX sockets are an example of *public mailboxes*.

FALSE, as sockets die with the process that created them.

d) You can create several semaphores through a single call to **semget()**.

TRUE, each semget call is capable of creating a whole semaphore array.

e) Peterson's algorithm assumes the existence of *shared variables*.

TRUE, it requires one shared integer and one shared array of integers.

f) The throughput of a computer using a round-robin scheduling policy often goes down when the number of users goes up.

TRUE, as the scheduler will reduce the CPU time slice to maintain a good interactive response time, which will unfortunately increase the time lost performing context switches.

2. Consider the following solution to the mutual exclusion problem:

```
shared int busy = 0; //shared variable
// enter critical section:
while (busy); // wait
busy = 1;
...
// leave critical section:
busy = 0;
```

a) What is the problem with this solution? (5 points)

Mutual exclusion will be denied.

b) When does this problem manifest itself? (5 points)

When two—or more—processes arrive in lockstep: they will all see busy equal to zero and enter the critical section.

3. Consider the following System V Release 4 scheduler:

#ts_quantum	ts_tqexp	ts_slpret	ts_maxwait	ts_lwait	LEVEL
2000	XXX	1	16000	1	# 0
1000	0	2	8000	2	# 1
YYY	1	3	4000	3	# 2
200	2	3	2000	3	# 3

a) Circle in the list below the most meaningful value for the **XXX** parameter. (5 points)

<u>0</u> 1 2 3 4 5

b) Circle in the list below the most meaningful value for the YYY parameter. (5 points)

100 200 <u>500</u> 1000 2000 4000

4. How would you pass a *linked list* to a remote procedure? (10 points)

As an array.

5. A parking can accommodate up to 20 cars. There is one gate through which all cars must go one at a time when they enter or leave the parking . Add the required semaphore calls to the following template to ensure that (a) the parking will never contain more than 20 cars, and (b) cars entering or leaving the parking will go though the gate one at a time. (20 points minus 2 points if you forget to initialize the semaphores)

<pre>semaphore spaces =20_;</pre>			
<pre>semaphore gate =1_;</pre>			
<pre>enter_parking(){</pre>	<pre>leave_parking(){</pre>		
_P(&spaces);	;		
_P(&gate);	_P(&gate);		
<pre>go_through_gate();</pre>	<pre>go_through_gate();</pre>		
_V(ågate);	_V(&gate);		
	_V(&spaces);		
<pre>} // enter_parking</pre>	<pre>} // leave_parking</pre>		

- 6. What is the difference between *datagrams* and *virtual circuits*? (10 points)
- 7. What is the difference between *blocking* and *non-blocking* receive primitives? (10 points)Answers to these two questions should be easy to find in the notes.