CMSC 412 Midterm #1 (Fall 2011)

Name _____ Signature _____

(1) This exam is closed book, closed notes, and closed neighbor. No calculators are permitted. Violation of any of these rules will be considered academic dishonesty.

(2) You have 70 minutes to complete this exam. If you finish early, you may turn in your exam at the front of the room and leave. However if you finish during the last ten minutes of the exam please remain seated until the end of the exam so you don't disturb others. Failure to follow this direction will result in points being deducted from your exam.

(3) Write all answers on the exam. If you need additional paper, I will provide it. Make sure your name is on any additional sheets.

(4) Partial credit will be given for most questions assuming I can figure out what you were doing.

(5) Please write neatly. Print your answers if your handwriting is hard to read. If you write something, and wish to cross it out, simply put an X through it. Please indicate if your answer continues onto another page.

Question	Possible	Score
1	**20**	
2	**20**	
3	**16**	
4	**12**	
5	**12**	
6	**20**	
Total	**100**	

1.) (20 points) Define and explain the following terms:

a) Starvation

b) Interrupt

c) Deadlock

d) Layering

2.) (20 points) Synchronization: Give an implementation of the readers/writers problem using semaphores that allows at most 5 readers at once. There can be at most one writer at once. Readers and writers may not be active at the same time. It should use no busy waiting (i.e. blocking on semaphore operations). Full credit requires a starvation free solution.

Semaphore and variable declarations:

Reader():

Writer():

3.) (16 Points) Scheduling

a) Given a round robin scheduler (with a quantum of 1 unit), and the following jobs, indicate when each job finishes:

Arrival Time	Required Time	Completion Time
0	10	
3	2	
6	6	
20	2	

b) Why have scheduling quantum remained relatively the same for the past 30 years (around 3-10 ms) despite many orders of magnitude increase in processor speed?

4.) (12 points) Explain the difference between processes and threads. How do kernel only threads (i.e. in the project) differ from user threads scheduled by the kernel?

5.) (12 points) Can disabling interrupts be used to provide mutual exclusion in a kernel? If so, under what conditions?

6.) (20 points) Project

a) In project #2, setup Frame copies the interrupt state onto the user stack even though there is a copy on the kernel stack. Why does it do that?

b) When implementing PS, what is the purpose of the len parameter passed from user space and how should it be used in the kernel?

c) Given that all memory in the system is mapped into the kernel's address space, why is there a function in the kernel called Copy_From_User?