CSSE 332 – Operating Systems
Rose-Hulman Institute of Technology
Computer Science and Software Engineering Department

Exam 2 — Computer Part

Name:__________________________________ Section:______ CM:_________

Instructions: This exam is open book, open notes, open computer.

- You must disable all chat tools (IM, ICQ, IRC, etc.) before the exam starts.
- **Any communication with anyone other than an exam proctor during the exam could result in a failing grade for the course**.
- You may refer to programs you have written for the course-both those assigned and those you have written for practice. You may not, however, refer to programs written by others except those provided in your textbook or by the course staff.
- Regarding materials on the web, you may only use the course web site and pages linked directly from it. Of course, search engine use is not allowed.
- Write all answers on these pages.
- Submit all code and support files to your svn repository.
- Read the entire examination before starting, and then budget your time.
- If you are stuck on problems 2, 3, or 4, you may buy the answer to that question by forfeiting the points for that problem. In that case, your instructor will give you the requested answer and ask that you write your initials next to the question(s) for which you will lose the credit.

	Points available	Your Score
1	10	
2	5	
3	5	
4	5	
5	15	
6	5	
7	5	
Total	50	

Checkout the `Barbershop` project from your course svn repository into your csse332 directory. You should be able to do this by simply doing an "`svn update`" on your working copy of the course repository.

Barbershop problem: Consider a barbershop with b barbers and n chairs in the waiting room. Customers enter the barbershop and if there is an empty chair in the waiting room, they take a seat. If all the seats are filled, the customer leaves the barbershop. If there are customers in the barbershop waiting, a barber will select a customer (the order is not important) and cut his hair. Otherwise, the barber will take a nap.

The `Barbershop` project attempts to synchronize the barbers and customers. However, this solution results in deadlock whenever it is executed.

Your job is to complete the project with a solution that does not deadlock. You may not use more than the five semaphores that are declared in the shared memory region.

The following questions all refer to the code in the `Barbershop` project. The code consists of five C source files, two header files, a Makefile, and a txt file as follows.

`utility.c` `utility.h`	Implements utility functions that the customer and barber processes use to perform their various tasks.
`sharedMemory.c` `sharedMemory.h`	Implements utility functions for managing shared memory regions. Process can use these functions to get a shared memory region, detach a shared memory region, and remove a shared memory region.
`barber.c`	Implements barber process. The barber selects a customer, if any is waiting in the barbershop, and cuts his hair. Otherwise, the barber takes a nap.
`customer.c`	Implements the customer process. Customers enter the barbershop and take a seat if one is available. Otherwise, they leave.
`barbershopMain.c`	Implements the Barbershop problem. The process creates and attaches the shared memory region to hold the semaphores for the barber and customer processes. It then initializes the semaphores appropriately, spawns the barber and customer processes (`barber` and `customer`), wait form their completion, and detach and remove the shared memory region when they complete.
`SAMPLE_OUTPUT.txt`	Contains output from a sample run of the application without deadlock.

You do not need to understand all of the code provided. You should focus your energy understanding only what you need to answer the questions below.

Problem 1 (10 points) Write a Makefile to build the executables `barbershopMain`, `barber`, and `customer`. Be sure to structure your Makefile so that source files which have not changed are NOT rebuilt. Be sure to add and commit your files to your svn repository.

Problem 2 (5 points) None of the semaphores are initialized in the `barbershop` project. Write code to initialize the semaphores. Be sure to commit your changes to svn.

Problem 3 (5 points) Only one barber and one customer are created in the `barbershop` project. Write code to create the correct number of barbers and customers. Be sure to commit your changes to svn.

Problem 4 (5 points) Run your modified `barbershop` project. Does it work as intended? If not, describe why.

Problem 5 (15 points) Modify the `barbershop` project to correct the root cause of the problem. Be sure to commit your changes to svn.

Hint: You should assume that all functions in the code provided work as their names imply.

Problem 6 (5 points) The problem statement placed no requirements on the order in which customers are called. If the barber shop were to remain open continuously, what problem might this cause?

Problem 7 (5 points) Are the functions `printBarberSummary()` and `printCustomerSummary()` ever executed? Explain.

Have you committed your files and changes to svn?