Notre Dame University

Computer Science Department

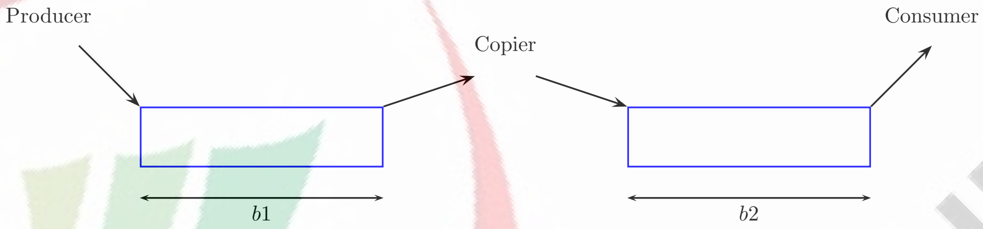
CSC 414 Applied Operating Systems
Exam 2
Spring 2008

Duration: 1 Hour
Name:
ID:

Instructions

1. Write your answers in the space provided on the question sheet.
2. Use the back pages for rough work.
3. **There is NO correlation between the space provided for each problem and the space needed. If you need more space write on the back and clearly indicate that you did so.**

1. (50pts) Consider the producer, copier and consumer problem as shown in the figure below. The produces inserts items into buffer $b1$, the copier removes items from buffer $b1$ and inserts them into buffer $b2$. The consumer removes items from buffer $b2$. Note that the copier has no internal storage thus while it is reading from $b1$ it should be writing simultaneously to $b2$. Write a solution to the problem using semaphores assuming that buffers $b1$ and $b2$ have size n.



```
semaphore mutex1=1,full1=0,empty1=n
semaphore mutex2=1,full2=0,empty2=n
Producer:                           copier
wait(empty1);                       wait(full1);
wait(mutex1);                       wait(empty2);
insert item into b1;                wait(mutex1);
signal(mutex1);                     wait(mutex2);
singal(full1);                      copy from b1 to b2;
                                    signal(mutex1);
Consumer:                           signal(mutex2);
wait(full2);                        signal(empty1);
wait(mutex2);                       signal(full2);
remove item from b2;
signal(mutex2);
signal(empty2);
```

2. (45pts) A system has 5 processes and 4 resource types. Resource types with 10 instances each. The initial state of the system is as shown in the table below.

	Alloc				Max			
	R_0	R_1	R_2	R_3	R_0	R_1	R_2	R_3
P_0	2	1	0	1	3	5	3	3
P_1	1	1	3	0	2	5	3	1
P_2	1	1	4	3	2	4	4	3
P_3	1	0	0	2	1	2	1	2
P_4	2	3	1	1	2	6	3	2

Use the Banker's algorithm to test whether the following requests are granted or denied, always starting from the initial state given above. Show all your work.

(a) Request (1,4,1,1) by P_0.

(b) Request (0,4,0,0) by P_1.

(c) Request (1,2,0,0) by P_0.

3. (15pts)Use semaphores to solve the barrier problem where N processes execute the same code shown below. No process should be able to execute the statement ”critical point” until all processes have executed the statement ”rendezvous”. (Hint: define semaphore mutex, semaphore barrier and a variable count).

Process
rendezvous critical point

Solution

```
semaphore barrier=0;
semaphore mutex=1;

wait(mutex);
count=count+1;
if ( count==n ) signal(barrier);
signal(mutex);

wait(barrier);
signal(barrier);
critical point;;
```