CS 471: Operating System Concepts
Fall 2005
Examination I
Points: 150
October 1, 2005
Time: 8:30-11:30 AM
CLOSED BOOK

Turning in this exam under your name confirms your continued support for the honor code of Old Dominion University and further indicates that you have neither received nor given assistance in completing it.

Name: ______________________________ UID: ____________________

CS Unix ID: __________@cs.odu.edu

Question #	Points	
	Maximum	Obtained
1	30	
2	30	
3	30	
4	30	
5	30	
Total	150	

Question 1.

(a) In a system, processes P1, P3, and P5 are waiting for I/O on disk 1. P2, P7, and P8 are waiting for the CPU. Process P4 has terminated. P6 is currently executing on the CPU. Show (diagrammatically) how the PCB’s of these processes are queued in the system.

(b) When and how are the PCB entries **program counter and registers** used by operating systems?

(c) Two processes P1 and P2 intend to coordinate in a task of sharing a single printer. Describe how the following two methods can be employed by processes in accomplishing this: (i) How Shared memory can be used? (ii) How message passing can be used?

Question 2.

(a) Given the following set of processes, with the length of the CPU burst, arrival time, and priority, answer the following assuming **Round-robin scheduling** with a **time quantum** of **10 units**, draw the Gantt chart and compute average waiting time.

Process ID	CPU Burst time	Arrival time	Priority
100	23	5	4
150	8	8	2
200	17	23	3
220	12	25	1

(b) Answer question (a) assuming **shortest-job-first scheduling with preemption**.

(c) Answer (a) above assuming **priority scheduling with preemption.**

Question 3.

(a) Given the following solution for the critical section problem (the code is for process Pi), give an example to show why it does not satisfy the mutual exclusion property for processes Pi and Pj.

```
do {

    flag[i]=TRUE;
    while (flag[j]) ; // Do nothing
    flag[i]=FALSE

            Critical section

    flag[i]= FALSE;

            Remainder section
}
```

(b) A programmer intends to use counting semaphores for mutual exclusion between processes Pi and Pj in sharing resources R1 and R2. There is only one instance of R1 and R2 in the system. Accordingly, he wrote the following code.

(i) Is there any problem if Pi and Pj execute only one at a time? Justify (with an example)

(ii) Is there any problem if both execute concurrently (at the same time)? Justify (with an example).

Pi	**Pj**
wait(R1);	wait (R2);
Read (R1);	Read (R2);
Update (R1);	Update (R2);
Signal (R2);	Signal (R1);
Wait(R2);	wait(R1);
Read (R2);	Read (R1);
Update (R2);	Update(R1);
Signal (R1);	Signal(R2);

(c) Given the following monitor solution for mutually exclusive access of a single instance resources R1 and R2, answer the following questions.

```
monitor MR12
{
    condition condR1, condR2;
    void get_R1() {
          condR1.wait();
          …………                    S1
                }
     void put_R1() {
          condR1.signal();
          ………………             S2
               }
      void get_R2() {
          condR2.wait();
          ……………                 S3
                }
     void put_R2() {
          condR2.signal();
          ………………              S4
               }
}
```

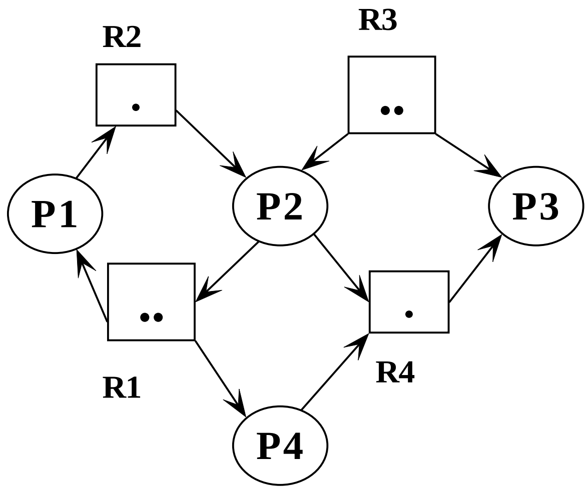
(i) What happens if process P2 executes the command get_R2() when process P1 is inside the monitor at S1 in get_R1()?

(ii) What happens if process P3 executed the command get_R1() and started using the resource when P4 executes get_R1()?

(iii) What if both P3 executes put_R1() and P2 executes put_R2() simultaneously?

Question 4.

(a) Given the following resource-allocation diagram, determine whether or not there is a deadlock. If yes, justify clearly indicating the reason. If no, explain why there is no deadlock.



(b) Given the following snapshot of a system, determine whether or not the system is in a deadlock. Including the ones already allocated, the system has 6 instances of of R1, 7 of R2, and 9 of R3. SHOW YOUR WORK justifying your answer.

Process ID	Max need			Current allocation		
	R1	R2	R3	R1	R2	R3
A	3	2	1	1	1	0
B	0	2	3	0	0	3
C	4	4	4	2	4	2
D	1	1	6	1	0	4

(c) Given the following resource-allocation state of a system at time T0, determine if the system is in deadlock or not at T0. Justify your answer.

Process ID	**New request**			**Current allocation**			**Currently available**		
	R1	**R2**	**R3**	**R3**	**R1**	**R2**	**R1**	**R2**	**R3**
A	3	2	0	1	1	1	2	2	2
B	0	2	2	3	0	0			
C	4	4	2	4	2	4			
D	1	1	4	6	1	0			

Question 5.

(a) Given the following page table, map the given logical addresses to physical addresses. Assume the page/frame table to be 1024 words. (All memory units are in words)

5
25
7
25
10

Page table

Logical address	Physical address
2500	
3956	
10000	
5000	
1000	

(b) A paging hardware uses TLB. The access time for TLB is 20 nanoseconds. The main memory access time is 500 nanoseconds.

(i) If a page entry is in the TLB, then what would be the memory system access time?

If a page entry is not in the TLB, then what would be the memory system access time?

(ii) If on the average 80% of the memory references are found in the TLB, what is the effective memory access time?

(c) Given the following hash function, show how a **hash page table** will be organized for a process with 4 pages (numbered 0,1,2,3).

$H(p) = p^2 - 3p + 2$

(d) Given the following **segment table**, map the logical addresses to physical addresses.

Segment #	Limit	Base
0	1000	2500
1	3000	30000
2	10500	5000
3	500	40000

Logical address <seg#, offset>	Physical address
<3, 200>	
<2, 4500>	
<1, 3500>	
<0, 250>	
<1, 22>	