CS 471: Operating System Concepts
Fall 2006
Examination I
Points: 150
September 30, 2006
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: _____

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) List the sequence of activities that occur when a currently running process P1 has executed an I/O instruction. Currently, three processes---P2,P3,P4---are in the ready queue.

(b) What is printed by the following program?

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int value=20;
int main (int argvc, char **argv)
{
    pid_t pid;
    value++;
    pid = fork();

    if (pid == 0) {
        value++;
        printf("I am the child  %d\n", value);
    }
    else {
        value +=5;
        printf("I am the parent  %d\n", value);
    }
    printf("exiting\n");
    exit(0);
}
```

(c) Suppose processes P1-P5 have **terminated**, P6 is **running**, P7-P15 are **waiting** at different I/O queues, P16-P20 are in the **ready** queue, and P21-P25 are in the **new** state, determine the degree of multiprogramming of the system.

(d) What is the role of a medium-term scheduler?

Question 2.

(a) Given the following set of processes---with the specified length of the CPU burst, arrival time, and priority---compute **average turnaround time** assuming **preemptive priority** scheduling Show the Gantt chart and other working details.

Process ID	CPU Burst time	Arrival time	Priority
1	13	5	4
2	10	10	3
3	7	15	2
4	12	20	1

(b) Answer question (a) assuming **shortest-job first scheduling** without preemption.

(c) Consider the exponential average formula used to predict the length of the next CPU burst of a process. The initial estimate of the CPU burst time is $\tau 0 = 150$ milliseconds and $\alpha = 0.7$. The following are the actual CPU burst observed. t_0=60 msec; t1=80 msec; t2=100 msec; t3=20 msec. Compute τ_1, τ_2, τ_3, and τ_4.
(Hint: $\boldsymbol{\tau_{n+1} = \alpha\, t_n + (1-\alpha)\, \tau_n}$)

Question 3.

(a) Given the following solution for the critical section problem (the code is for processes P1 and P2), show why it does not satisfy the mutual exclusion property for processes P1 and P2. Here, **key** is a **local** variable and **unlock** is a **shared** variable (initialized to FALSE). Swap is an atomic operation. (Hint: Indicate one scenario where the mutual exclusion property is violated.)

```
do {

        key=FALSE;
        unlock= TRUE;
        while (key = =FALSE)  Swap(&unlock, &key);

        Critical section

        unlock= FALSE;

        Remainder section
}
```

(b) Suppose TestAndSet instruction is implemented in software instead of as a hardware instruction, what would be the impact on the following solution to the critical section? (Hint: lock is a shared variable initially set to FALSE; consider P1 and P2 as the concurrent processes using TestAndSetLock to enforce critical section on a resource.)

Critical section solution:

```
do {
        while (TestAndSetLock(&lock);  //do nothing
             //critical section

          lock = FALSE;
             //reminder section
       } while (TRUE);
```

Code for TestAndSet:

```
boolean TestAndSet(Boolean *target) {
            boolean rv = *target;
            *target = TRUE;
            return rv;
           }
```

(c) Given the following monitor solution for mutually exclusive access of a single instance resource R, answer the following questions.

```
monitor RA
{
    boolean busy;
    condition x;

    void acquire() {
          if (busy)
             x.wait();
          busy = TRUE;
    }

     void release() {
            busy = FALSE;
            x.signal();
      }

      Initialization_code() {
            busy = FALSE;
        }
}
```

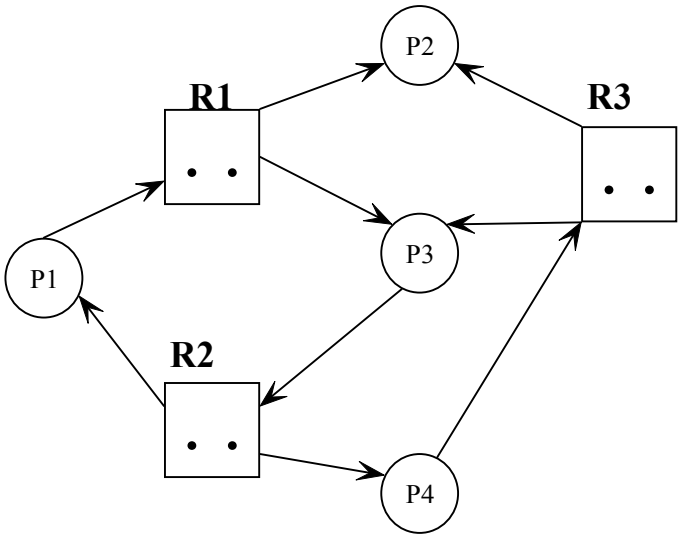
Answer each of the following questions independently.

(i) What are the consequences of two processes P1 and P2 simultaneously executing the statement RA.acquire() in their own code? Assume that prior to this, the resource is not acquired by any one.

(ii) What are the consequences, if any, of a process P3 incorrectly using the sequence: RA.release(); critical section; RA.acquire();?

Question 4.

(a) Given the following resource-allocation diagram, (i) Draw the wait-for graph (ii) 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 safety state. SHOW YOUR WORK justifying your answer. (Use Banker's algorithm)

Process	Maximum need			Current allocation			Available		
ID	R1	R2	R3	R1	R2	R3	R1	R2	R3
P1	3	2	3	2	2	2	1	1	2
P2	1	3	2	0	1	2			
P3	4	4	4	0	2	2			
P4	1	6	1	1	0	1			

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

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

Question 5.

(a) Given the following page table (page size = 512 bytes) of a process with size 2724 bytes, answer the following:

4
20
9
11
15
27

Page table

(i) How much memory was wasted due to internal fragmentation?
(ii) How much memory is wasted due to external fragmentation?
(iii) What steps are taken when the CPU is presented with logical address 2120?

(iv) What is the physical address (in memory) corresponding to the logical address 1265?

(b) A paging hardware uses TLB. The access time for TLB is 30 nanoseconds. The main memory access time is 100 nanoseconds. Currently TLB has the following entries:

Page#	Frame #
2	10
5	25
3	15

Suppose the process in (a) above is currently executing. Answer the following:

(i) What is the memory access time if the logical address is 1433.

(ii) What is the memory access time if the logical address is 1000.

(iii) If on the average 75% of the memory references are found in the TLB, what is the effective memory access time?

(c) 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, 122>	
<2, 4500>	
<1, 2500>	
<0, 750>	
<4,50>	