BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
First Semester 2003-2004

Course Title : OPERATING SYSTEMS **Course No *CS C372***
Component : Test I (Regular) ***Closed Book Component***
Weightage : 15% **Max Marks: 15** **Date : 15-09-2003**

QI (10 X 0.5 = 5 Marks): ***Please write only one-word answers in the FIRST PAGE of the answer book and the answers to the parts should be in order***

1. With _____the dual mode operation protect the OS from errant users and errant users from one another.
2. By using _____ and _____ we are implementing memory protection in dual mode operation.
3. _____ And _____ are the two communication methods used.
4. _____ Is the main advantage of layered approach
5. For each of the following statements, is the statement true or false?
   a. Multiple threads that are part of the same process have their own program counter and stack pointer values.
   b. Multiple threads that are part of the same process have their own copies of global variables

(A) True, True (B) True, False (C) False, True (D) False, False

6. For each of the following statements, is the statement true or false?
   a. Multiple threads that are part of the same process share any open files.
   b. It is generally faster to switch between threads that are part of the same process than it is to switch between processes.

(A) True, True (B) True, False (C) False, True (D) False, False

7.
```
Void main()
{
        if(!fork())   if(!fork())    if(!fork());
}
```
how many new processes are created by fork calls?

8. A process is about to execute the following code {fork();exec(…); fork();} what is the maximum number of new processes that should be created from this code?
9. A program containing race condition will _____result in data corruption or some other incorrect behavior?

(A) Always (B) Sometimes (C) Never (D) None

10 Suppose you try to enforce mutual exclusion as follows:

PROCESS 1:	PROCESS 2:

while (true) {
  while (p2_in_crit_sect == true)
  ; /* do nothing */
  p1_in_crit_sect = true;
  … *critical section* …
  p1_in_crit_sect = false;
}
	while (true) {
	  while (p1_in_crit_sect == true)
	  ; /* do nothing */
	  p2_in_crit_sect = true;
	  … *critical section* …
	  p2_in_crit_sect = false;
	}

Does this code always enforce mutual exclusion?

(A) Never (B) Sometimes (C) Always (D) Information not sufficient

Q II. (1 X 5 = 5 Marks)

Assume that a context switching may occur between any two-machine instructions. Initially, address x contains 2 and address y contains 10, Assembly-language code is given for two threads, T1 and T2 which execute concurrently. Basically, T1 does "y:= x + y" and Y2 does "x := y + 1".

T1	T2
LOAD x, r1 LOAD y, r2 ADD r1, r2 STORE r1, y	LOAD y, r1 INC r1 STORE r1, x

What are the possible pairs of final values of x and y (i.e.,the values after both threads have finished)? For example, your answer could be something like: x = {1, 2, 3 . . .} and y ={1, 2,. . . }.

Q III. (1 X 5 = 5 Marks)

For each of the following code segments, where the variables are initialized as shown and thread A and B can run concurrently, what are the possible ending values of x after both threads A and B are completed.

(a)	**(b)**	**(c)**

Int x=0;
Int y=23;
Semaphore m=1;
Semaphore s=0;
Thread A
{
x=x+1;
}
 	
Thread B
{
x=y+1;
}
	Int x=0;
	Int y=23;
	Semaphore m=1;
	Semaphore s=0;
	Thread A
	{
	wait(m);
	x=x+1;
	signal(m);
	}
	
	Thread B
	{
	wait(m);
	x=y+1;
	signal(m);
	}
		Int x=0;
		Int y=23;
		Semaphore m=1;
		Semaphore s=0;
		Thread A
		{
		wait(s);
		x=x+1;
		signal(m);
		}
		
		Thread B
		{
		wait(m);
		x=y+1;
		signal(s);
		}

Possible ending values of x	Possible ending values of x	Possible ending values of x

Please write the answer in the following format:

a) Possible ending values of x : _____
b) Possible ending values of x : _____
c) Possible ending values of x : _____