Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

Operating Systems (CS30002)

Class Test 1

Spring 2019-20 Semester **Full Marks: 40** **Time: 60 minutes**

Answer all the questions

1. While executing a *printf* function call execution on a typical processor, explain the steps of how the processor switches between: **[4 marks]**
   a) user mode to privileged mode
   b) privileged mode to user mode

   User to privileged: While executing **`printf`**, the **`printf`** call is translated into the **`write`** function which contains a "**`syscall`**" instruction. Upon receiving the instruction the kernel set the mode bit in processor to 0, switching from user mode to privileged mode, and the processor start executing the instructions for write.

   Privileged to user: At the end of code for **`write`**, processor encounter a special "**`return`**" instruction, upon receiving the instruction the mode bit is set to 1, switching from privileged mode to user mode.

2. With respect to process state transition diagram, explain under what circumstances the following state transitions take place? **[6 marks]**
   a) Running to Ready
   b) Running to Waiting
   c) Waiting to Ready

   Running to Ready: Timer interrupt or any other external interrupt arrives.

   Running to Waiting: An I/O instruction or any other system call is invoked that causes the process to wait until some event occurs.

   Waiting to Ready: A process is in the blocked or waiting state when it is waiting for some event to occur. When the event occurs, the process is immediately shifted to the Ready state.

3. Suppose a short-term scheduling algorithm favours those processes that have used little processor time in the recent past. **[4x2 = 8 marks]**

   a) Does the process favour I/O-bound processes or CPU-bound processes? Why or why not?

      The process favours I/O bound processes, since such processes generally use low CPU.

   b) Is the following statement True or False? : "This algorithm will continuously deny processor time to CPU-bound processes". Explain your answer.

      `False`. If CPU bound processes get processor time denied continuously, then naturally they will use little processor time and get executed.

4. Answer the following. **[4x2 = 8 marks]**
   a) Clearly explain how the concept of SPOOLing helps in improving CPU utilization in traditional batch processing systems.
   b) Your printer connected to your computer uses SPOOLing. Point out what problem will you face if you want to print two documents simultaneously if you disable SPOOLing? Why?

      a) In SPOOLing a dedicated processor is responsible for all I/O operations between the I/O devices and the disk. When the CPU executes a program, it reads input data directly from the disk, and also prints the output data to the disk.
      b) Without SPOOLing there is no spooling buffer, thus the jobs cannot be stored in the printer spool. Thus, the printer cannot queue multiple jobs waiting for printing. Consequently without SPOOLing we cannot print two documents simultaneously.

5. For the following code segment, how many output numbers (as a function of n) will be printed in the terminal if you execute this code? How many of these numbers are non-zero? **[3 + 3 = 6 marks]**

```
...
main ()
{
        ...
        int pid;
        int n;
        int i;
        for(i=0;i<n;i++){
                pid = fork();
                printf ("%d \n", pid);
        }
        ...
}
```

1st iteration = 2 processes → 2 numbers printed
2nd iteration = 4 processes → 4 numbers printed
3rd iteration = 8 processes → 8 numbers printed
…
nth iteration = 2^n processes → 2^n numbers printed
total numbers printed = $2 + 4 + 8 + \ldots + 2^n = 2^{(n+1)} - 2$
In each iteration half of the numbers are 0, so rest are non-zero numbers → $2^n - 1$ non zero numbers.

6. Consider the following five processes, with their arrival times and execution times given in milliseconds.

Process	Arrival Time	CPU Burst Time
P1	0	2
P2	0	4
P3	3	2
P4	4	13
P5	6	6

Draw the Gantt charts and estimate the average waiting times for the following scheduling algorithms:

i) Non-preemptive Shortest Job First
ii) Round-robin with time slice of 3 ms (assume P1 starts executing first in the beginning)

[4 + 4 = 8 marks]

SJF schedule → P1, P2, P3, P5, P4
Waiting time → 0 + 2 + (6 - 3) + (8 - 6) + (14 -4) = 17, avg. = 17/5 = 3.4 ms

RR schedule → P1 (2ms), P2(3ms), P3 (2ms), P4 (3ms), P2 (1ms), P5(3ms), P4(3ms), P5(3ms), P4(7ms)
Waiting time → 0 + (2 + 5) + (5 - 3) + ((7 – 4) + 4 + 3) + ((10- 6) + 4) = 27/5 = 5.4 ms