EAST WEST UNIVERSITY BANGLADESH
Department of Computer Science & Engineering

CSE325: Operating System

Midterm Examination **Spring 2015**

Total Marks: 40 | Instructor: Dr. Md. Shamim Akhter | Time: 90 minutes

Instruction: Answer eight (8) questions.

1. A **context-switch** is when the processor (CPU) stops executing one process and begins executing another. List the necessary steps for the operating system to perform a context-switch, including what information must (typically) be saved.

2. What is the difference between a **trap and an interrupt**? What does the CPU hardware do when a trap or interrupt occurs? Describe the specific steps that the hardware performs (but just the hardware - not the OS or any process).

3. What is the purpose of a **process control block** (PCB)? When it is updated and when it is read by the operating system? Define zombie process with example and explain- why zombie is not welcomed.

4. When a process executes a **fork ()** system call, a duplicate process (i.e. the child process) is created. How does the code in the processes know which process is the parent and which is the child– since the code is identical in both parent and child processes? Explain- the difference between child process execution with & without **execlp()** call.

5. The following code (in C) provides an example of a race condition, if it were executed by two processes roughly at the same time. Explain- what bug could arise for the race condition. Solve the bug using **Mutex/Binary Semaphore**.

```
int queue[SIZE], i = 0; //assume these are shared between processes
void insert(int x)
{
    queue[i] = x;
     i++;
}
```

6. Give an example showing that **multithreading** can achieve better performance than single threading. Describe whether user threads or kernel threads can provide better performance in your example.

7. Is **busy waiting** always less efficient than a blocking wait? Explain.
   Disabling interrupts is a common method for the operating system to ensure mutual exclusion on a uniprocessor. Is it appropriate for a shared memory (MIMD) multiprocessor? Why or why not?

8. Name three ways in which the processor can transition from user mode to **kernel mode**? How does the system keep track of which mode it's operating in? Can the user execute arbitrary code

after transition from user mode to kernel mode? Explain.

9. Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	Burst Time	Priority
P_1	7	3
P_2	4	1
P_3	5	3
P_4	10	4
P_5	5	2

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time zero. Also consider a **time quantum of 2**.

a. Draw two graphical representation of the schedule of tasks that illustrate the execution of these processes using the following scheduling algorithms: **Shortest Job First (SJF), Round Robin (RR)**.

b. What is the waiting time of each process for **SJF** and **RR** scheduling algorithms?