SET: A

EAST WEST UNIVERSITY BANGLADESH

Department of Computer Science & Engineering

CSE325: Operating System (1, 2)

Term-I Examination **Summer 2015**

Total Marks: 25 Instructor: Dr. Md. Shamim Akhter Time: 60 minutes

1. Most hardware provides user mode and kernel mode. Applications are run in user mode, while the OS runs in kernel mode. The tricky part is **transitioning from one mode to the other**, which is done via system call. What takes place during the system call? (5)

2. When a process executes a **fork ()** system call, a duplicate process (i.e. the child process) is created. Explain the difference between - child execution with & without **execlp()** call, and parent execution with & without **wait( )** call. (5)

```
pid t pid;
pid = fork();
if ( pid == 0 ) {
    execlp("ps", "ps", "-ax", 0);
}
else if (pid < 0) {
    printf("fork failed.\n");
    exit(1);
}
else {
    wait(NULL);
}
exit(0);
```

3. Explain the difference between **processes and threads**. What are the advantages of using threads (rather than processes) in implementing a complex application? What are the unique components that each thread has and does not share with other threads and/or processes? (5)

4. **Disabling interrupts** is inappropriate for user mode and is only acceptable for brief periods in kernel mode on uniprocessors. Suppose you have a processor (CPU) that has no test-and-set instruction. How would you go about providing for mutual exclusion? Be brief & explicit. (5)

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 **Lock variable and Busy waiting**. Note: assume both process has same execution speed. (5)

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