EAST WEST UNIVERSITY BANGLADESH
Department of Computer Science & Engineering

CSE325: Operating System (1)

Midterm I Exam **FALL 2016**

Total Marks: 25=5x5 Instructor: Dr. Md. Shamim Akhter Time: 80 minutes

1.
   a) Under what circumstances can a multithreaded program complete more quickly than a non-multithreaded program? Keep in mind that multithreading has context-switch overhead associated with it. (3)
   b) What performance costs are saved by the use of **thread pools**? (2)

2.
   a) What is the difference between **a mode switch** and **a context switch**? (2)
   b) Name three ways in which the processor makes transition from user mode to kernel mode?(3)

3.
   a) There are three types of schedulers. What are they and what are their assigned tasks? (2)
   b) Change the following code snippet so that the **execlp( ) function** can work only inside child program. After changing, assume **the parent is a multithreaded program**, and explain your reasonable understanding about the number of thread (s) should be in the child program. (3)

```
int main ( ){
   pid_t child_p;
   printf("Running ps with fork() " );

   child_p= fork( );

   execlp("ps", "ps", "-ax", 0);
   return 0;
}
```

4. Assume a uniprocessor system has three processes (P1, P2, and P3) and P1 needs eight (8), P2 needs three (3), and P3 needs seven (7) seconds CPU time. Each of which needs constant amount of input waiting time (3-seconds) and output waiting time (2-seconds). How long will it take to complete the processes in:
   a) Batch processing system? (1.5)
   b) Multiprogrammed processing system? (1.5)
   c) **Time-shared processing** system with variable time quantum (time slice)? (2)
      - During the execution of the two processes @ system, the time slice=3.
      - During the execution of the three processes @ system-the time slice= 2.

5.
   a) What is a **virtual processor system**? How does the processor virtuality work in multi-programming (or multi-tasking) system? (3)
   b) Explain the activity of the following code segment and make necessary changes to generate **positive thread id**. `printf("Parent ID is %d\n", getpid());` (2)