BRAC UNIVERSITY
Department of Computer Science and Engineering
CSE321: Operating Systems

Midterm Exam **Fall 2021**
Duration: 1 Hour 10 minutes **Total Marks: 35**

1. a) How is the kernel different from the other parts of an operating system code? [2]

   b) What is a system call? Briefly explain the role of system call interface. [3]

   c) Explain with example why a long term scheduler should wisely select processes to load into memory. [4]

   d) What is the difference between creating multiple processes and creating multiple threads for distributing the work (computation/io) a program does? When should we prefer one against another? Explain with examples. [4]

   e) Explain if maximizing the CPU utilization without any concern about throughput will lead to a good scheduling strategy. [2]

2. What are the possible outputs of the following code snippet? [Hint: There can be multiple set of outputs] [4]

```
int main()
{
    int x = 1;
    pid_t pid = fork();
    if (pid < 0) {
        print("Fork failed");
        exit(1);
    } else if (pid > 0) {
        x++;
        printf("Value of x = " + x);
    } else {
        x--;
        print("Value of x = " + x);
    }
   return 0;
}
```

3. You have the money to buy either a 4 core or an 8-core processor. However, an 8-core processor costs 10 thousand taka more than a 4-core processor. You decide to only buy an 8-core processor if the programs you execute run at least 20% faster in it than what they do in a 4-core processor. Now, if all your programs are 40% serial/sequential then would you buy the 8-core processor? Justify your answer by showing proper computation. [5]

4. Consider the information of the following processes.

Process	Arrival Time	Burst Time
P0	0	12
P1	1	6
P2	3	3

Now, Compare the Preemptive and Non-preemptive shortest job first algorithm to schedule these processes.

a) Draw Gantt charts for both algorithms. [2]
b) Find the average waiting time in both cases. [2]
c) Calculate the percentage of performance increase in waiting time if you pick the better algorithm. [1]

5. The Round Robin (RR) CPU scheduling policy preempts an executing process via a system parameter called time quantum. Consider the information of the processes in a system given below.

Process	Burst Time	Arrival Time
P1	135	5
P2	102	150
P3	56	200
P4	48	300
P5	55	308

Using the RR policy with a time quantum of 20 time units,

a) Construct a Gantt chart of the execution [3]
b) Calculate average waiting time, and average response time for all processes. [3]