BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Mid Term | Semester: Spring 2022
Duration: 1 Hour 10 minutes | Full Marks: 30

CSE 321: Operating Systems

Answer the following questions.
Figures in the right margin indicate marks.

---

1. **CO1**

a) **Explain** how the two modes of the hardware enable the operating system to securely control user processes. [2]

b) **Explain** the differences between Multiprogramming and Multiprocessing with examples. [2]

2. **CO2**

a) **Describe** what is the process control block, its contents, and how it is used. In particular, describe its role in context switching. [2+1]

b) **Find** the output of the following code snippet. [3]

```
int main()
{
    int pid1, pid2;
    pid1 = fork();
    if (pid1 == 0) {
        pid2=fork();
        if(pid2 == 0) printf("Hello!\n");
        else{
            wait(NULL);
            printf("World!\n");
        }
    } else {
        wait(NULL);
        printf("Missed Me?\n");
    }
    printf("Don't miss me!\n");
    return 0;
}
```

3. **CO4**

a) Suppose, in a system, you can use up to 4 processors for 40% of the applications, which means 40% of the applications can run in parallel. **Calculate** the speedup if you increase the number of processors from 1 to 4. [4]

b) Remember that *pthread_create(tid, NULL, fn, arg)* creates a new thread that executes the function *fn* with the argument *arg*, and *pthread_join(tid, NULL)* let the current thread wait for the thread with id = *tid* to complete execution. With this information in mind, **find** all possible outputs of the following program.

```
int[]  matrix = {4, 6, 9, 2, 5, 3, 0, 1, 11, 13, -1, 7};
void main() {
      pthread_t t1, t2;
      printf("Printing partial sums of the array");
      pthread_create(t1, NULL, sum, 0);
      pthread_create(t2, NULL, sum, 4);
      pthread_join(t1, NULL);
      sum(8);
}
void sum(int startIndex) {
     int partialSum = 0;
     for (int i = startIndex; i < startIndex+4; i++) {
            partialSum += matrix[i];
     }
     printf("For index %d to %d = %d", startIndex, startIndex+4, partialSum);
     pthread_exit(0);
}
```

[4]

4. **CO3** Consider the following processes with arrival time and burst time at a specific moment in the ready queue that needs to be scheduled.

Process	Arrival Time	Burst Time
P1	2	8
P2	7	3
P3	5	10
P4	5	6

a) Apply Shortest Remaining Time First (SRTF) scheduling algorithm and show the following - [2+2+1]
   i) Gantt Chart
   ii) Average Waiting Time & Average Turnaround Time
   iii) Number of Context Switching

b) Apply Round Robin (RR) scheduling algorithm with quantum = 3 and show the following - [2+2+1]
   i) Gantt Chart
   ii) Average Waiting Time & Average Turnaround Time
   iii) Number of Context Switching

c) Find the best-suited algorithm between these two and give your reasoning. [2]