# BRAC UNIVERSITY
## Department of Computer Science and Engineering

Examination: Mid Term
Duration: 1 Hour 10 minutes

Semester: Spring 2022
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.

[4]

```
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. **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]

# BRAC UNIVERSITY
## Department of Computer Science and Engineering

Examination: Final
Duration: 2 Hours

Semester: Spring 2022
Full Marks: 40

## CSE 321: Operating Systems

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

---

1. **CO5**

a) **Explain** Race conditions with an example. **Mention** how we protect the system from this phenomenon. [2+1]

b) **Explain** with a code example how a careless ordering of semaphore operations can lead to a deadlock situation among two processes. [2]

c) Suppose, you have to design an online consultation system for the teachers and students of your university. There are certain constraints that you have to keep in mind while designing the system - [5]

i. the teachers can set their status whether they are available to give consultation or not. A teacher will set him unavailable after the giveConsultation() function.
ii. the students will enter a voice channel if the teacher is available for consultation.
iii. the students will wait if one student is in consultation with the teacher

Now, you have to **design** the teacher and student function using semaphores so that synchronization can be achieved among them maintaining the constraints mentioned above. You can use the following code template given below and complete it. **Mention the initial semaphore values before writing the functions.**

```
//initialize the semaphore values here
teacher(){
        // write semaphore code here
        giveConsultation();
        // write semaphore code here
}

student(){
        // write semaphore code here
        takeConsultation();
        // write semaphore code here
}
```

2. **CO5**

a) **Explain** how Banker's algorithm can help to find the processes that are causing a deadlock in a system. [2]

b) **Describe** some strategies for deadlock prevention that can break the hold-and-wait condition. [2]

c) Suppose, in an office, we have a set of resource types, R = {R1, R2, R3} and a set of processes, P = {P1, P2, P3, P4}. R1, R2, and R3 have 4, 2, and 2 instances respectively.

i) P1 is holding 2 instances of R1,

ii) P2 requests 1 instance of R3

iii) P3 requests 2 instances of R2

iv) P2 requests 1 instance of R1

v) P2 is holding 1 instance of R2

vi) P3 is holding 1 instance of R3 [3]

**Construct** a resource allocation graph for the above scenario and **identify** whether there is a deadlock or not.

d) Consider the following snapshot of a system:

| | | Allocation | | | | | Max | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **A** | **B** | **C** | **D** | | **A** | **B** | **C** | **D** |
| **P1** | | 0 | 0 | 1 | 2 | | 0 | 0 | 2 | 3 |
| **P2** | | 1 | 0 | 0 | 0 | | 1 | 2 | 2 | 0 |
| **P3** | | 1 | 3 | 5 | 4 | | 2 | 3 | 5 | 6 |
| **P4** | | 0 | 0 | 0 | 1 | | 2 | 2 | 0 | 1 |

| Available | | | |
|---|---|---|---|
| 1 | 2 | 2 | 0 |

[3]

i. Is the system in a safe state? [2]

ii. Can P3's request (1 0 0 0) be safely granted immediately? [3]

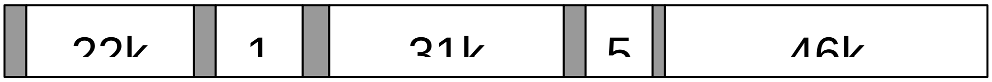
iii. If P3's request is granted immediately, does the system enter a deadlock?

3. **CO6**

a) **Explain** the disadvantage of using Contiguous allocation and how Paging is more beneficial than Contiguous allocation. [3]

b) **Explain** how the operating system's behavior and hardware mechanism for logical to physical address translation ensure that one process cannot access the memory allocated for another process. [2]

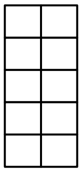
c) At a particular time, the snapshot of the Main memory is given below for dynamic partition. Gray portions of the memory are occupied space.



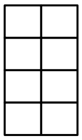
i) **Apply** worst-fit and best-fit algorithms to allocate processes with the space requirement of P1(26k), P2(30k), P3(15k), P4(20k), and P5(6k). [4]

ii) **Explain** which algorithm makes the most effective use of memory? [2]

d) Suppose, in a system, there are two processes - P1 (16 bytes) and P2 (12 bytes) with a page size of 4 bytes. The main memory size of the system is 32 bytes. Page tables of both processes are given below.



Page Table of P1



Page Table of P2

Find the corresponding physical address of the following logical addresses - [4]

i. address 1011 of P1
ii. address 0100 of P1
iii. address 0111 of P2
iv. address 1010 of P2