B

Department of Computer Science and Engineering
Midterm Examination Spring 2025
CSE 321: Operating Systems

Duration: 1 Hour 30 Minutes **Total Marks:** 30

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

Name:	**ID:**	**Sec:**

1. CO1 **a) Write the answers in the question paper.**

[1*2 =2]

i) Let's consider a scenario where we have a total of 4 processes having total burst times like **10, 5, 8, 4**. Now, at first, the Round-Robin scheduling algorithm has a time quantum of **12**. Next, the time quantum has been reduced to **6**. Later, the time quantum is even reduced to 1. Based on the current situation, **identify** the best time quantum value in terms of providing minimum turnaround time.

Answer:

ii) Observe the following code. **Calculate** how many processes and how many threads will be created.

```
pid_t pid;
pid = fork();
if (pid == 0) { /* child process */
      fork();
      pthread_create( . . .);
}
fork();
```

Answer:

b) Find outputs of the given code: **[10* 0.5= 5]**

```
int main(){
    pid_t x,y;
    int z[]={15,21,33};
    x=fork();
    if(x<0){
        printf("error\n");
    }
    else if(x>0){
        z[0]=z[1]*2;
        z[1]=z[0]*4;
        z[2]=z[1]*6;
        wait();
        y=fork();
        if(y<0){
            printf("error\n");
        }
        else if(y==0){
            z[0]=z[1]+z[2];
            z[1]=z[0]+z[2];
            z[2]=z[1]+z[0];
        }
        else{
            printf("hello\n");
            wait();
            z[0]=z[1]-z[2];
            z[1]=z[0]-z[2];
            z[2]=z[1]-z[0];
        }
    }
    else{
        z[0]=z[1]-z[2];
        z[1]=z[0]-z[2];
        z[2]=z[1]-z[0];
    }

    for(int i=0;i<3;i++){
        printf("z[%d]= %d\n",i,z[i]);
    }
    return 0;
}
```

Write the outputs in the question paper. Output sequence should be exactly matched.

Outputs

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

c) Write the answers in the answer script.

Dipu is implementing his own OS. For his OS he has decided to develop **Multilevel Feedback Queue** as a scheduler. To design the scheduler, he decided to divide the ready queue into **three** levels. According to his preferences, in every level round-robin algorithm will be used and some constraints should be maintained in each level which are given below.

- **Q1:** time quantum = **2**, priority = **20**
- **Q2:** time quantum = **4**, priority = **33**
- **Q3:** time quantum = **8**, priority = **49**

Here, the lower priority value indicates the higher priority and in case of an I/O request, the scheduler will promote a process by one level upon I/O operation completion. At a certain moment, Dipu has collected information on burst times, arrival times and I/O times of four processes.

Process	Burst Time	Arrival Time	I/O Time
P1	13	3	5 [After **9s** of total CPU allocation]
P2	6	0	N/A
P3	7	7	3 [After **5s** of total CPU allocation]
P4	14	10	N/A

I. **Draw** a Gantt chart using multilevel feedback queue scheduling algorithm showing the states of the ready queues of different levels. **[12]**

II. **Calculate** the average waiting time and average turnaround time from the Gantt chart. **[1.5+1.5=3]**

2. CO2 **a) Write the answers in the question paper.** **[1*3=3]**

i) You are given an array of 100 integers. You are required to perform four operations: **1+2+3+..., -1-2-3-..., 1 x 2 x 3 x ..., 1/2/3/...** on the entire array using a **4-**core CPU. **Identify** what kind of parallelism you need to implement to ensure that the operation is performed most efficiently.

Answer:

ii) Imagine that BRACU Ayesha Abed Library is equipped with **20** Individual Study Pods, which are allocated to students from different departments based on their needs. The allocation is as follows:

- **7** pods are dedicated to CSE students
- **6** pods are dedicated to EEE students
- **7** pods are dedicated to BBA students

Each department has its dedicated study pods and a strict policy is enforced, ensuring that students can only use the pods allocated to their respective department. During finals week, **3** CSE students arrive at the library, hoping to access the study pods for their preparation. However, **6** out of **7** CSE pods are already booked for the whole day, leaving only **one** pod available for use. As a result, the students must take turns to access the study pod. **State** which synchronization method can be used here to ensure the synchronization.

Answer:

iii) You are booking a train ticket for your journey back home during Eid. When you visit the booking website, you find that there is only one seat left. You proceed to book it, and the booking is confirmed. The code used to book a seat is as follows:

```
lock = 1
book_seat(){
    while(compare_and_swap(&lock,1,0));
     if (available_seats > 0) {    // Check if seats are
available
          available_seats = available_seats - 1  // Deduct
seat
        print("Booking confirmed!")  // Confirm booking
    }
    else {
        print("No seats available.")
    }
    lock = 0;
}
```

Figure out the error from the above code.

Answer:

b) Find outputs of the given code: **[10* 0.5= 5]**

```
int t_id[]={1,2,3,4};
void *func(int *id);
int p=45;
int q=78;
sem_t s1,s2,s3,s4;

int main(){
        pthread_t t[4];
        sem_init(&s1,0,0);
        sem_init(&s2,0,0);
        sem_init(&s3,0,0);
        sem_init(&s4,0,1);
        for(int i=0;i<4;i++){
                pthread_create(&t[i],NULL,(void *)func,&t_id[i]);
        }
        for(int i=0;i<4;i++){
                pthread_join(t[i],NULL);
        }
        sem_destroy(&s1);
        sem_destroy(&s2);
        sem_destroy(&s3);
        sem_destroy(&s4);
        printf("Final p: %d\nFinal q: %d\n",p,q);
        return 0;
}
void *func(int *id){
    if(*id==1){
        sem_wait(&s1);
        printf("p: %d\nq: %d\n",p,q);
        p=p-15;
        q=q-p;
        sem_post(&s1);
    }
    else if(*id==2){
        sem_wait(&s3);
        printf("p: %d\nq: %d\n",p,q);
        p=p+q;
        q=p-7;
        sem_post(&s3);
        sem_post(&s1);
    }
    else if(*id==3){
        sem_wait(&s4);
        printf("p: %d\nq: %d\n",p,q);
        p=p*9;
        q=q*11;
        sem_post(&s4);
        sem_post(&s2);
    }
    else{
        sem_wait(&s2);
        printf("p: %d\nq: %d\n",p,q);
        p=p+q;
        q=p+q;
        sem_post(&s2);
        sem_post(&s3);
  }

}
```

Write the outputs in the question paper. Output sequence should be exactly matched.

Outputs

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	