



United International University
Course: Operating Systems (CSE 4509), Spring 2025
Class Test 2, Set A
Total Marks: 20, Time: 30 minutes

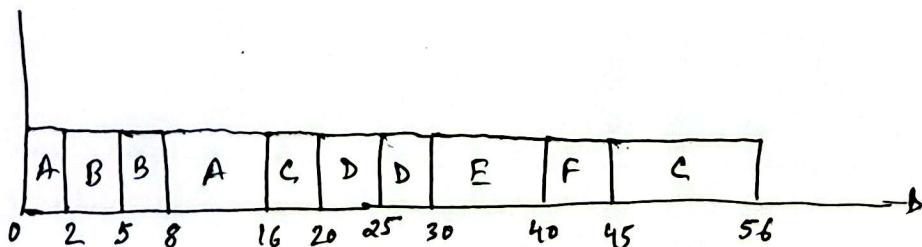
Name		ID		Section	
------	--	----	--	---------	--

1. What is the convoy effect? Show an example where Shortest Job First (SJF) can suffer from this effect. [4]

See slide

2. Draw the Gantt chart using STCF algorithm and find the average turnaround time and average response time from the following data: [6]

Process	A	B	C	D	E	F
Arrival (ms)	0	2	5	20	25	40
Duration (ms)	10	6	15	10	10	5



~~A=8~~

~~B=5~~

~~C=15~~

~~D=10~~

~~F=10~~

$$\text{Avg turn. time} = \frac{(16-0) + (8-2) + (56-5) + (30-20) + (40-25) + (15-40)}{6}$$
$$= \frac{103}{6} = 17.167 \text{ ms}$$

□

3. If there are two producers, one consumer and buffer size is 1, then show what problem may arise from the following implementation of Producer-Consumer problem: [6]

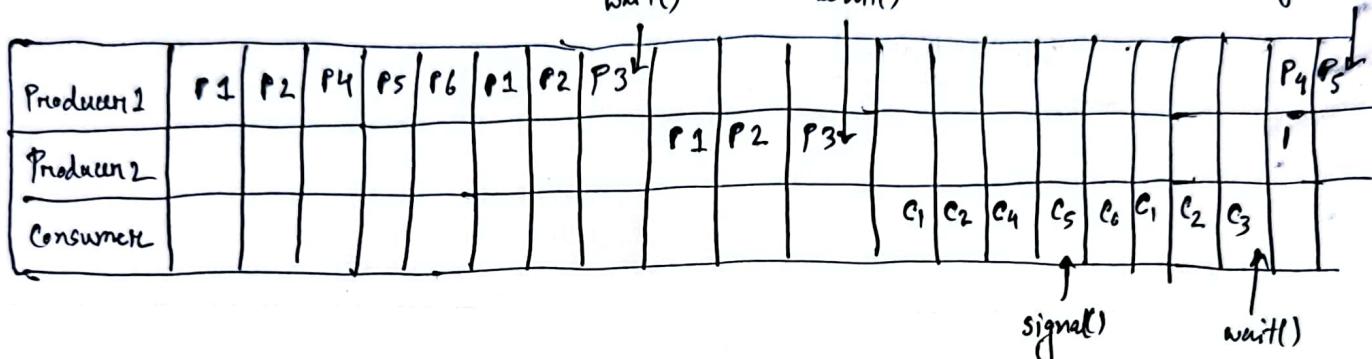
```

void *producer(void *arg) {
    for (int i=0; i<loops; i++) {
        Mutex_lock(&m); // p1
        if(numfull == max) //p2
            Cond_wait(&cond, &m); //p3
        do_fill(i); // p4
        Cond_signal(&cond); //p5
        Mutex_unlock(&m); //p6
    }
}

```

```
void *consumer(void *arg) {
    while(1) {
        Mutex_lock(&m); // c1
        if(numfull == 0) // c2
            Cond_wait(&cond, &m); // c3
        int tmp = do_get(); // c4
        Cond_signal(&cond); // c5
        Mutex_unlock(&m); // c6
        printf("%d\n", tmp); // c7
    }
}
```

c3 who will get
the signal?
producer 2 or consumer
signal()



4. Suppose you're in a project of two members. Your partner is working on the front end and when he completes, you'll start working on the backend. You're represented by thread 1 and your partner is represented by thread 2. Now consider the following code snippet:

thread_1() { mutex_lock(&lock); cond_wait(&cond, &lock); back_end(); mutex_unlock(&lock); }	thread_2() { mutex_lock(&lock); front_end(); signal(&cond); mutex_unlock(&lock); }
--	---

If we run this, then thread 1 gets frozen(Goes to waiting and never returns though thread 2 has given the signal)! Explain why this may happen. [4]

Say, `thread-2()` has completed running before `thread-1()` has been scheduled for first time. Then when `thread 1` runs, it goes to waiting without checking if it needs to wait or not. Now there is no one to signal `thread 1`. Thus it gets frozen.