CS & IT ENGINEERING

Operating System

Process Synchronization / Coordination

DPP 04 Discussion Notes

By- Anjnee Bhatnagar ma'am

01 Question

02 Discussion

Q.1

[MCQ]

Consider the following program:

```
Bool Lock = False;
void Process (int i)
{
        while(1)
        {
        < Non-critical section >
        while(TSL(&Lock)= = True);
        < Critical Section >
        Lock = False;
        }
}
```

Lock = True

1. M.E ✓
2. Progress ✓
3. Bounded waiting.

⇒ Spin lock

B

Which of the following are correct regarding given code?

A. It guarantees mutual exclusion but not progress.

B. It gurantees mutual exclusion and progress but not bounded waiting.

C. It guarantees progress but not mutual exclusion.

D It guarantees progress and bounded waiting but not guarantees mutual exclusion.

Q.2

Which of the following is/are correct regarding "SWAP"? [MSQ]

A. It is a software-based synchronization mechanism. X

B. It is atomic instruction.

C. It is based on "Lock Key" mechanism.

D. It is non-privileged instruction. X

B, C

Q.3

[MCQ]

Match the following:

1) TSL	(i) Mutual exclusion
2) SWAP	(ii) Progress
	(iii) Bounded waiting

Which of the following option is correct?

A. 1 – (i); 2 – (ii); 2 – (iii)

B. 1 – (ii); 1 – (iii); 2 – (ii)

C. 1 – (i); 2 – (i); 2 – (ii)

D. 1 – (iii); 2 – (i); 1 – (i)

Q.4

Which of the following is/are correct?

[MSQ]

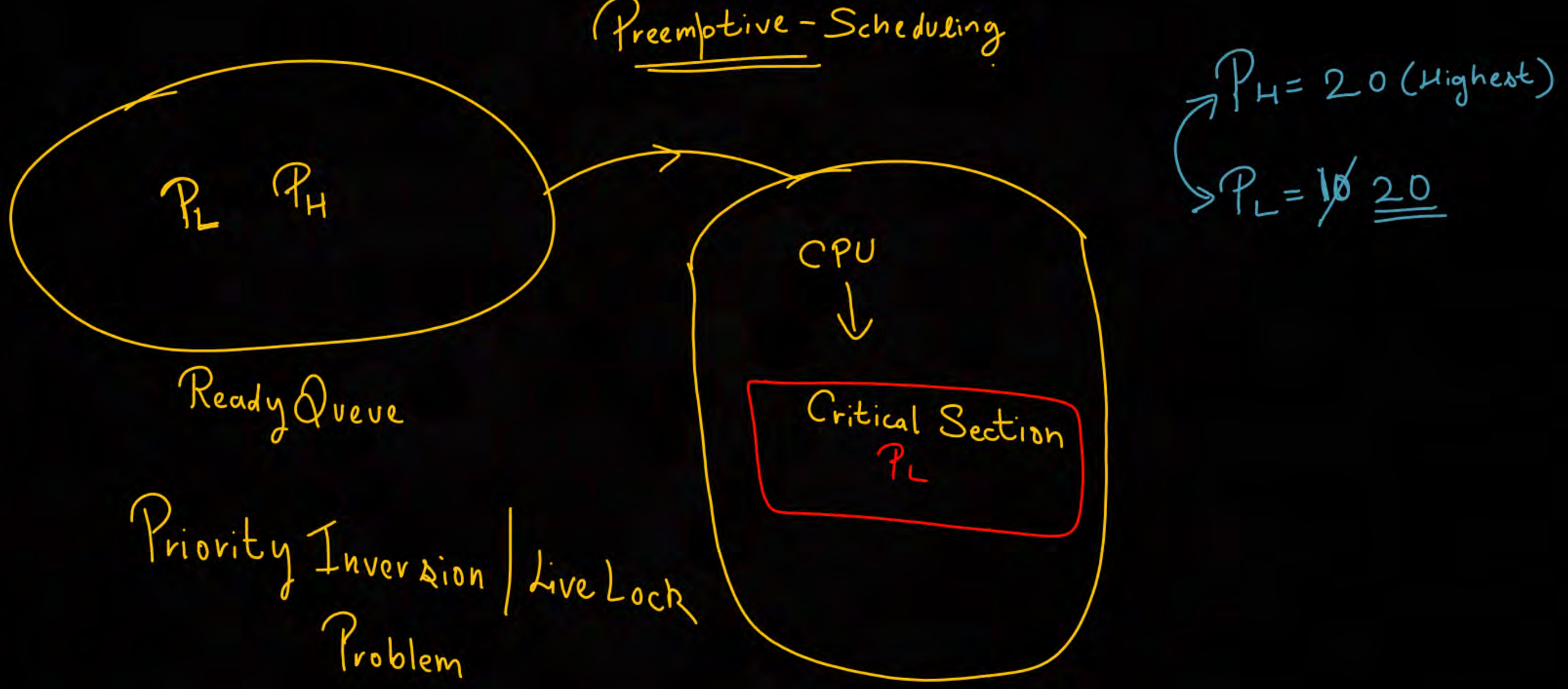
A. TSL has busy waiting and wastes CPU cycle.

B. TSL suffers from priority - inversion.

C. SWAP has busy waiting and utilizes CPU cycle efficiently.

D. SWAP suffer from priority-inversion.

A, B, D


Preemptive - Scheduling
$P_H = 20$ (Highest)
$P_L = \cancel{10}$ 20
P_L P_H
Ready Queue
CPU
Critical Section
P_L
Priority Inversion | Live Lock
Problem

Q.5

Priority inversion problem can be solved by _____.

[MCQ]

A. Preemption.

B. Priority Inheritance

C. Preemption followed by priority exchange.

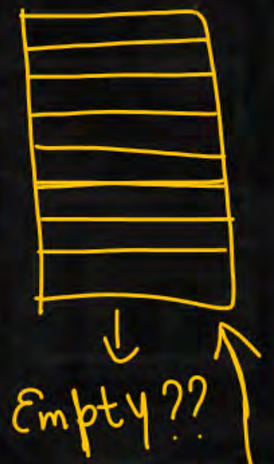
D. No solution, and priority inversion problem can cause deadlock.

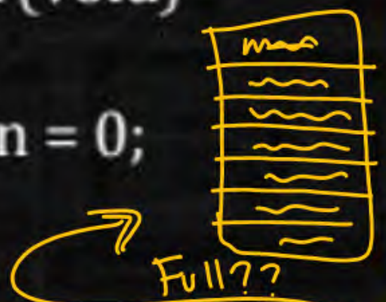
Q.6

[MCQ]

Consider the following program of producer from producer-consumer problem implemented using sleep( ) and wakeup ( ).

```
void Producer(void)
{
    int itemp, in = 0;
    while(1)
    {
     itemp = Produce.item();
    X;
    Buffer [in] = itemp;
    in = (in + 1) % N;
    Count = Count + 1;
    Y if(Count == 1) wakeup Consumer
    }
}
```





A. X = if(count = = 1) sleep( );
Y = if(count = = N) wakeup(consumer);

B. X = if(count = = N) sleep( );
Y = if(count = = N -1) wakeup(consumer);

C. X = if(count = = N) sleep( );
Y = if(count = = 1) wakeup(consumer);

D. X = if(count = = N) wakeup(consumer);
Y = if(count = = 1) sleep( );

What will be the value of X and Y respectively?

Q.7

Operations performed on semaphores are _____.

[MCQ]

A. Wait and signal

B. Sleep and wakeup

C. Increment and decrement

D. All of the above.

Wait , Signal

Up() , down()

P , V

Q.8

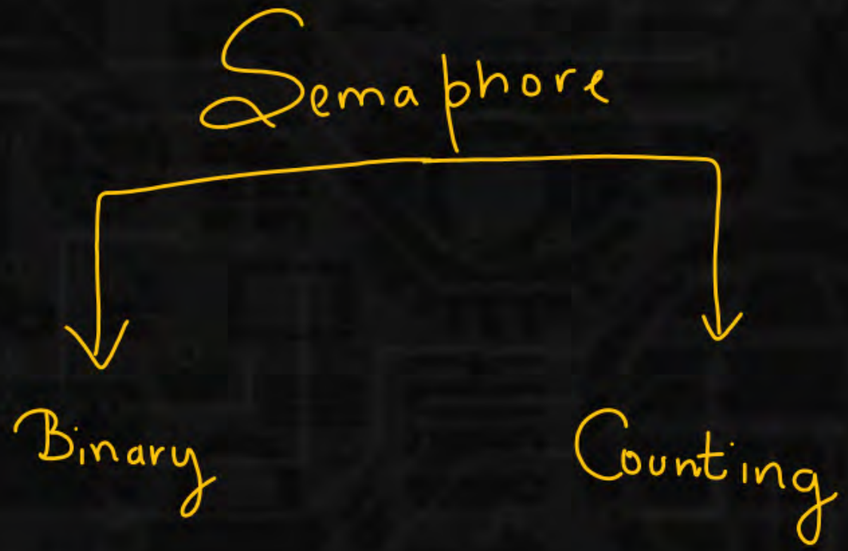
Which of the following are the types of semaphore? [MSQ]

A. Binary

B. Bounded

C. Counting

D. Incremental



A, C


THANK YOU GW
SOLDIERS !