Branch : CSE & IT Batch : Hinglish

Operating Systems

Process Synchronization/Coordination

DPP 08

[MCQ]

1. In dining philosopher's problem, deadlock can occur when ____

(a) Two philosopher's pick there left fork and get preempted before picking right fork.

(b) Two philosopher's sitting in front of each other pick there left fork and then right fork.

(c) Four philosophers are sitting and doing nothing.

(d) All philosopher's pick there left fork and preempted before picking right fork.

[MCQ]

2. Consider the following code:

```
# define N 6
void p (int i)
{
    while (1)
    {
        T(i);
        T_f(i);
        T_f((i + 1)%N);
        x(i)
        p_f(i);
        p_f((i + 1)%N);
    }
}
```

T is a think function, T_f is a take fork function, x is a eat function, p_f is a put fork function, and i represents the philosopher. Which of the following is true about above code?

(a) Prevents Deadlock but has starvation.

(b) Has deadlock but do not have starvation.

(c) Has deadlock and starvation both.

(d) Do not have deadlock and starvation.

[NAT]

3. How many of the following instructions can run independently?

I_1: p = q * r;

I_2: q = p + z;

I_3: l = m + k;

I_4: n = a + b;

I_5: m = z + s;

[MCQ]

4. Consider the following instructions:

I_1: a = b + c;

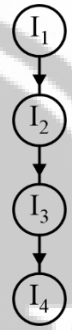
I_2: d = e * f;

I_3: k = a + d;

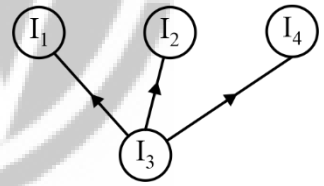
I_4: l = k + m;

Which of the following is correct precedence graph for the above instructions?

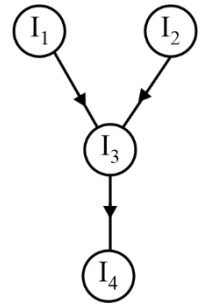
(a)



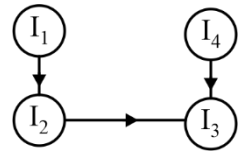
(b)



(c)



(d)



[MSQ]

5. Solve the following code by co-begin and co-end

x = 4;
co-begin
x = x + 5;
x = x + 6;
co-end;
What could be the possible final value of x?

(a) 5 (b) 9
(c) 10 (d) 15

[MCQ]

6. Busy waiting is _____.

(a) When a process polling on a variable.
(b) When a process periodically checks a variable.
(c) When a process issues an interrupt.
(d) When a process continuously checks a variable.

[NAT]

7. Consider the following concurrent program, how many levels are there in it's precedence graph?

S_1:
Parbegin
begin;
S_2; S_3;
end;
begin
S_4;
begin
S_5; S_6;
end;
Parbegin
S_7
begin
S_8; S_9
end;
Parend;
end;
Parend;
S_{10}

Answer Key

1. **(d)**
2. **(c)**
3. **(3)**
4. **(c)**
5. **(b, c, d)**
6. **(d)**
7. **(7)**

Hints & Solutions

1. (d)

In dining philosopher's problem if all philosophers get hungry and take their left fork and gets pre-empted before picking its right fork. Then, deadlock can occur.

2. (c)

The given code is example of classical IPC problem dining philosopher and it can suffer from deadlock and/or starvation.

Therefore, option (c) is correct.

3. (3)

I_1: p = q * r;

I_3: l = m + k;

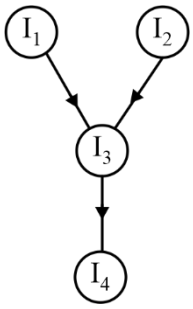
I_4 : n = a + b;

I_1, I_3, and I_4 can execute / run independently.

4. (c)

As, I_3 has 'a' and 'd' so it depends on I_1 and I_2, and I_4 has 'k' so it depends on I_3.

Therefore,



is the correct precedence graph.

5. (b, c, d)

If first we execute x = x + 5 $\Rightarrow$ 4 + 5 = 9

If it executes, x = x + 6 $\Rightarrow$ 4 + 6 = 10

If it executes x + 5 then x + 3 then x = 15.

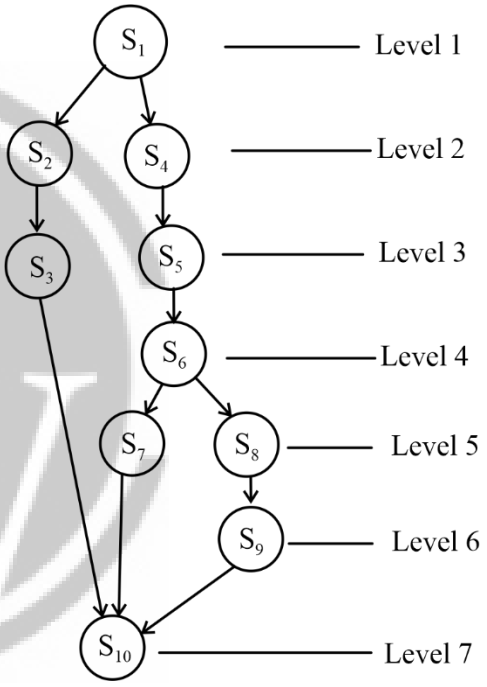
So, possible values are 9, 10 and 15.

6. (d)

Busy waiting means that the process continuously checks for the required value in a variables or an event, thereby waiting CPU cycle, as no useful work is being done.

So, option (d) is correct.

7. (7)



So, there are total 7 levels.