

Operating Systems

Process Synchronization/Coordination

DPP 08

[MCQ]

1. In dining philosopher's problem, deadlock can occur when ____
- Two philosopher's pick there left fork and get preempted before picking right fork.
 - Two philosopher's sitting in front of each other pick there left fork and then right fork.
 - Four philosophers are sitting and doing nothing.
 - 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?

- Prevents Deadlock but has starvation.
- Has deadlock but do not have starvation.
- Has deadlock and starvation both.
- Do not have deadlock and starvation.

[NAT]

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

I₁: p = q * r;
 I₂: q = p + z;
 I₃: l = m + k;
 I₄: n = a + b;
 I₅: m = z + s;

[MCQ]

4. Consider the following instructions:

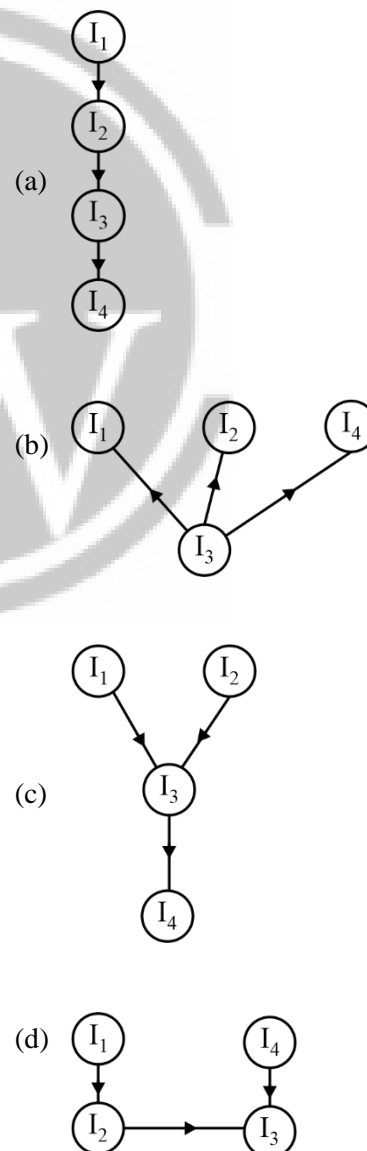
I₁: a = b + c;

I₂: d = e * f;

I₃: k = a + d;

I₄: l = k + m;

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



[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?

```
S1:
Parbegin
  begin;
    S2; S3;
  end;
begin
  S4;
  begin
    S5; S6;
  end;
Parbegin
  S7
begin
  S8; S9
end;
Parend;
end;
Parend;
S10
```

Answer Key

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



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)

$I_1: p = q * r;$

$I_3: l = m + k;$

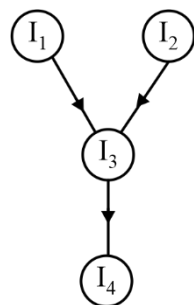
$I_4: n = a + b;$

$I_1, I_3,$ 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 $x = 15$.

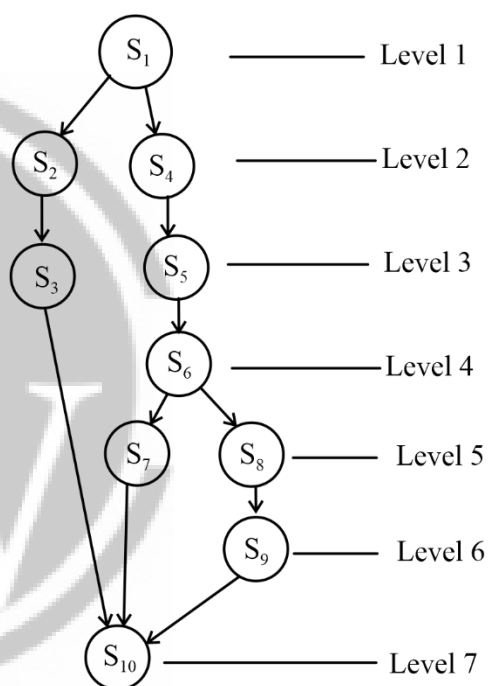
So, possible values are 9, 10 and 15.

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.



Any issue with DPP, please report by clicking here:- <https://forms.gle/t2SzQVvQcs638c4r5>

For more questions, kindly visit the library section: Link for web: <https://smart.link/sdfez8ejd80if>



PW Mobile APP: <https://smart.link/7wwosivoicgd4>