Week 3 – Process Synchronization.

SOLUTIONS

Problem-01:

A counting semaphore S is initialized to **10**. Then, 6 **wait**() operations and 4 **signal**() operations are performed on S. What is the final value of S?

Solution

We know:

- wait() operation decrements the value of semaphore variable by 1.
- signal() operation increments the value of semaphore variable by 1.
- **P** operation also called as **wait** operation decrements the value of semaphore variable by 1.
- **V** operation also called as **signal** operation increments the value of semaphore variable by 1.

Thus,

Final value of semaphore variable S = 10 — (6 x 1) + (4 x 1) = 10 — 6 + 4 = 8

Problem-02:

The following program consists of 3 concurrent processes and 3 binary semaphores. The semaphores are initialized as S0 = 1, S1 = 0 and S2 = 0.

Process PO	Process P1	Process P2

while (true)
{
wait (S0);
print '0'
signal (S1);
signal (S2);
}
	wait (S1);
	signal (S0);
		wait (S2);
		signal (S0);

What is the **maximum** number of times process **P0** can print '0'?

a) At least twice
b) Exactly twice
c) Exactly thrice
d) Exactly once

Solution

Maximum Number Of Times Process P0 Can Print '0'-

Maximum number of times process P0 can print '0' = 3

The occurrence of following scenes will cause process P0 to print '0' three times-

Scene-01:

- Process **P0 runs** first.
- It executes wait operation on semaphore S0 successfully. Now, **S0 = 0**.
- It then **prints '0'. (1st time)**
- It executes signal operation on semaphore S1. Now, **S1 = 1**.
- It executes **signal** operation on semaphore S2. Now, **S2 = 1**.
- While loop causes process P0 to run again.
- It executes **wait** operation on semaphore S0 unsuccessfully and gets blocked.

Scene-02:

- Process **P1** runs.
- It executes **wait** operation on semaphore S1 successfully. Now, **S1 = 0**.
- It executes **signal** operation on semaphore S0 which wakes up the process P0.
- The execution of process **P1 is completed.**

Scene-03:

- **Process P0 runs again.**
- **It prints '0'. (**2nd time)
- It executes **signal** operation on semaphore S1. Now, **S1 = 1**.
- It executes **signal** operation on semaphore S2. Now, **S2 = 1**.
- While loop causes process P0 to execute again.
- It executes wait operation on semaphore S0 unsuccessfully and gets blocked.

Scene-04:

- **Process P2 runs**.
- It executes **wait** operation on semaphore S2 successfully. Now, **S2 = 0**.
- It executes **signal** operation on semaphore S0 which wakes up the process P0. Now, **S0 = 1**.

- The execution of process P2 is completed.

Scene-05:

- **Process P0 runs again (because S0 = 1)**.
- It **prints '0'.** (3rd time)
- It executes **signal** operation on semaphore S1. Now, **S1 = 1**.
- It executes **signal** operation on semaphore S2. Now, **S2 = 1**.
- While loop causes process P0 to execute again.
- It executes wait operation on semaphore S0 unsuccessfully and gets blocked.

Now,

- The execution of processes P1 and P2 is already completed.
- There is no other process in the system which can perform signal operation on semaphore S0.
- Thus, process P0 cannot execute any more.

Thus, maximum number of times process P0 can print '0' = **3 times**

Thus, Option (C) is correct.

Problem-03:

A shared variable x, initialized to zero, is operated on by four concurrent processes **W, X, Y, Z** as follows.

Each of the processes **W** and **X** reads **x** from memory, increments by one, stores it to memory and then terminates.

Each of the processes **Y** and **Z** reads **x** from memory, decrements by two, stores it to memory, and then terminates.

Each process before reading x invokes the **wait()** operation on a counting semaphore S and invokes the **signal**() operation on the semaphore S after storing **x** to memory.

Semaphore **S is initialized to two**.

Find the maximum possible value of x after all process complete execution.

A. -2
B. -1
C. 1
D. 2

Process W	Process X	Process Y	Process Z
Wait (S)	Wait (S)	Wait (S)	Wait (S)
Read (x)	Read (x)	Read (x)	Read (x)
x = x + 1;	x = x + 1;	x = x - 2;	x = x - 2;
Write (x)	Write (x)	Write (x)	Write (x)
Signal (S)	Signal (S)	Signal (S)	Signal (S)

Solution

Initially, counting semaphore **S is initialized with value 2**.
Now, we have been asked the maximum possible value of x after all the processes complete execution.

Clearly,

- Processes W and X increments the value of x.
- Processes Y and Z decrements the value of x.

To obtain the maximum value of x, the processes must execute in such a way that-

- Only the impact of the processes W and X remains on the value of x.
- The impact of processes Y and Z gets lost on the value of x.

STRATEGY

- First of all, make the process W read the value of x = 0.
- Then, preempt the process W (stop its execution).
- Now, schedule process Y and process Z to execute one by one.
- After executing them, reschedule process W.
- Now, when process W gets scheduled again, it starts with value x = 0 and increments this value.
- This is because before preemption it had read the value x = 0.
- The updates from the processes Y and Z gets lost.
- Later, execute process X which again increments the value of x.

Step-01:

- Process W runs first.
- It executes the wait(S) operation and the value of S decrements by 1. Now, S = 1.
- It reads the value x = 0.
- It gets preempted (the O.S. stops its execution).

Step-02:

- Process Y runs.
- It executes the wait(S) operation and the value of S decrements by 1. Now, S = 0.
- It reads the value x = 0.
- It decrements the value of x by 2. Now, x = 0 - 2 = -2.
    - It writes the value x = -2 in the memory.

- It executes the signal(S) operation and the value of S increments by 1. Now, S = 1.
- Now, execution of process Y is completed.

Step-03:

- Process Z runs.
- It executes the wait(S) operation and the value of S decrements by 1. Now, S = 0.
- It reads the value x = -2.
- It decrements the value of x by 2. Now, x = - 2 - 2 = -4.
- It writes the value x = -4 in the memory.
- It executes the signal(S) operation and the value of S increments by 1. Now, S = 1.
- Now, execution of process Z is completed.

Step-04:

- Process W runs again.
- It resumes its execution from where it left.
- Before preemption it had already read the value x = 0.
- Now, it increments the value of x by 1. Now, x = 0 + 1 = 1.
- It writes the value x = 1 in the memory.
- It executes the signal(S) operation and the value of S increments by 1. Now, S = 2.
- Now, execution of process W is completed.

Step-05:

- Process X runs.
- It executes the wait(S) operation and the value of S decrements by 1. Now, S = 1.
- It reads the value x = 1.
- It increments the value of x by 1. Now, x = 1 + 1 = 2.
- It writes the value x = 2 in the memory.
- It executes the signal(S) operation and the value of S increments by 1. Now, S = 2.
- Now, execution of process X is completed.

Thus,

- Final value of x = 2.
- This is the maximum possible value of x that can be achieved after executing all the 4 processes.

Option (D) is correct.