DPP - 01

CS & IT

Operating system

Process Synchronization

Q1 Consider a process scenario in which each process executes first in CPU then goes for IO operation, then once again process needs a CPU burst and then terminates. Following is given a process scenario in which for CPU execution system uses preemptive SJF algorithm. Consider system has enough number of resources to carry out IO operations for all processes in parallel at a time.

Process	Arrival Time	CPU Burst Time	IO Burst Time	CPU Burst Time
P1	0	6	7	2
P2	1	4	2	1
P3	2	1	4	3

The completion times for the processes P1, P2 and P3 are respectively?

(A) 24, 11, 9 (B) 24, 9, 11

(C) 17, 11, 9 (D) 17, 9, 10

Q2 The following two functions P1 and P2 that share a variable D with an initial value of 4 execute concurrently.

```
P1() {              P2() {
    X = D – 2;          Y = D * 3;
    D = X * 2;          D = Y – 2;
   }                   }
```

The number of distinct possible values of D after a successful execution of P1 and P2 are ____?

Q3 Consider 5 concurrent processes P1, P2, P3, P4 and P5 as shown below, which access a shared variable B. Variable B is initialized to 35.

```
P1          P2          P3          P4          P5
{           {           {           {           {
 B = B +     B = B –     B = B +     B = B +     B = B –
10          8           3           6           5
}           }           }           }           }
```

The processes are executed on single CPU in time-shared environment. The minimum possible value of B after all 5 processes completed is M and maximum possible value of B after all 5 processes completed is N then the value of M – N is _____?

Q4 Consider the following solution for synchronization of 2 processes P1 and P2. Consider here the variable *lock* is Boolean type and is shared between both the processes.

```
P1()                        P2()
while(true)                 while(true)
{                           {
  while(lock!=True);          while(lock!=False);
    //critical section;         //critical section;
  lock = False;               lock = True;
 }                           }
```

Which of the following is correct if *lock* variable is initialized to False?

(A) Mutual exclusion is satisfied

(B) Progress is satisfied

(C) Bounded waiting is satisfied

(D) There is starvation

Q5 A shared variable x, initialized to 3, is operated on by four concurrent processes , , , as follows. Each of the process and reads x from memory, increments by 2, stores it to memory and then terminates. Each of the processes and reads x from memory , decrements by 3, stores it to memory and

then terminates. Each processes before reading x invokes the operation (i.e., wait) on a counting semaphore and invokes the operation (i.e., signal) on the semaphore after storing x to memory. Semaphore is initialized to two. The minimum and maximum possible values of x after all processes complete execution are A and B respectively, then value of B – A is _____?

Q6 Which of the following statements is/are not incorrect for semaphores?

(A) Synchronization solutions using semaphore can have busy waiting
(B) Synchronization solutions using semaphore may have deadlock
(C) Synchronization solutions using semaphore may suffer from priority inversion
(D) Synchronization solutions using semaphore may not have mutual exclusion

Q7 A non-negative counting semaphore S is initiated with value x. After performing 13 P() and 4 V() functions values of semaphore S becomes 27. Values of x is _____?

Answer Key

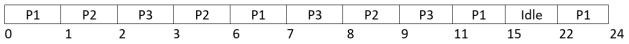
Q1	B
Q2	3
Q3	-32
Q4	A, C, D
Q5	10
Q6	A, B, C, D
Q7	36

Hints & Solutions

Note: scan the QR code to watch video solution

Q1 Text Solution:

The gantt chart for the execution of the processes will be as follows:



The completion times of P1, P2 and P3 are 24, 9, 11

Q2 Text Solution:

Case 1: When P1 executes completely and then P2 executes.

Hence after P1, D = 4

After P2 executes D = 10

Case 2: When P2 executes completely and then P1 executes.

Hence after P2, D = 10

After P2 executes D = 16

Case 3: When P1 and P2 reads D=4 concurrently and P1 writes last.

Hence after P1, D = 4

After P2 executes D = 10

And P1 writes at the end D = 4

Case 4: When P1 and P2 reads D=4 concurrently and P2 writes last.

Hence after P1, D = 4

After P2 executes D = 10

And P1 writes at the end D = 10

Hence there are 3 distinct values of D: 10, 16, 4

Q3 Text Solution:

For minimum possible value all the subtractions will be only in impact one after another, hence value of B will be:

B = 35 – 8 – 5

= 22

For minimum possible value all the additions will be only in impact one after another, hence value of B will be:

B = 35 + 10 + 3 + 6 = 54

M = 22, N = 54

M – N = 22 – 54 = –32

Q4 Text Solution:

If lock variable is true then only process P1 can enter into critical section,

And when lock variable is false then only process P2 can enter into critical section.

Hence There is mutual exclusion because lock can be either true or false as time and it is changed only after critical sections of processes.

When lock is initiated to true then P2 can not enter into critical section until P1 comes and makes lock to False. Hence there is no progress and also process p2 can starve here.

When lock is initiated to false then P1 can not enter into critical section until P2 comes and makes lock to True. Hence there is no progress and also process p1 can starve here.

There is strict alternation in process execution here as when P1 makes lock to False then only P2 will run, and after P2 makes lock to True then only P1 will run. Hence one process can not run 2 times while keeping other process in waiting. Hence bounded waiting is satisfied.

Q5 Text Solution:

Semaphore S is initialized with 2 hence 2 processes can enter into critical section, and hence can read variable x together. Hence there will be race condition here.

For minimum possible value all the subtractions will be only in impact one after another, hence value of x will be:

x = 3 – 3 – 3 = – 3

A = – 3

For maximum possible value all the additions will be only in impact one after another, hence value of x will be:

x = 3 + 2 + 2 = 7

B = 7

B – A = 7 – (– 3) = 10

Q6 Text Solution:

(a) Semaphore can cause busy waiting because of wait functions.

(b) Wrongly used semaphore can cause deadlock among processes.

(c) It might possible that a high priority process is stuck in wait due to unavailability of critical section, and other low priority process later can use critical section as soon as it is available. Hence high priority process may have to wait for low priority process here. It is called as priority inversion.

(d) If counting semaphores are used and are initialized with value greater than 1 hence it is possible to not satisfy mutual exclusion here.

Q7 Text Solution:

Every P() function decrements value of S

And every V() function increments value of S

27 = x -13 + 4

x = 27 +13 -4

x = 36