

**Question:** Explain semaphores and write a short note on it.

**Answer:** Dijkstra proposed a significant technique for managing concurrent processes for complex mutual exclusion problems. He introduced a new synchronization tool called Semaphore.

Semaphores are of two types –

1. Binary semaphore
2. Counting semaphore

Binary semaphore can take the value 0 & 1 only. Counting semaphore can take nonnegative integer values.

Two standard operations, wait and signal are defined on the semaphore. Entry to the critical section is controlled by the wait operation and exit from a critical region is taken care by signal operation. The wait, signal operations are also called P and V operations. The manipulation of semaphore (S) takes place as following:

1. The wait command P(S) decrements the semaphore value by 1. If the resulting value becomes negative then P command is delayed until the condition is satisfied.
2. The V(S) i.e. signals operation increments the semaphore value by 1.

Mutual exclusion on the semaphore is enforced within P(S) and V(S). If a number of processes attempt P(S) simultaneously, only one process will be allowed to proceed & the other processes will be waiting. These operations are defined as under –

P(S) or wait(S):

If  $S > 0$  then

Set S to S-1

Else

Block the calling process (i.e. Wait on S)

V(S) or signal(S):

If any processes are waiting on S

Start one of these processes

Else

Set S to S+1

The semaphore operation are implemented as operating system services and so wait and signal are atomic in nature i.e. once started, execution of these operations cannot be interrupted.

Thus semaphore is a simple yet powerful mechanism to ensure mutual exclusion among concurrent processes.