# Assignment 4

## Thread synchronization using counting semaphores

→ **Aim:**

Application to demonstrate producer-consumer problem with counting semaphores.

---

→ **Objective:**

Implement C program to demonstrate producer-consumer problem with counting semaphores and mutex.

---

→ **Theory:**

Three operations can be performed on a semaphore: initialize, decrement & increment. The decrement results in blocking of process and increment results in unblocking.

Two Types.

1. Binary Semaphores &
2. Counting Semaphores.

→ Counting Semaphores :

Counting semaphores are free from the limitations of binary semaphores.

An integer variable, initialized to K (K>=0) during operation assume a value <=K to process a queue. It will hold all the PCBs of all processes, waiting to enter CR. Queue is implemented to allow process as per FCFS order.

---

→ Operation on Counting Semaphores :

1. Let initial value of semaphore = 1
2. When count of semaphore = 1 it means no process is in critical section & no process is waiting.
3. When count = 0 then one process is in critical section but no process is waiting in queue.
4. When count = N then one process in critical section & N processes are waiting
5. When a process is in queue, then they are in waiting or blocked state.
6. When a waiting process is selected for entry in CR, it is tranferred from block state to ready state.

→ ## The producer/consumer problem :

There are one or more producers, generating some type of data & placing it in buffer. There is a single consumer that takes out item from buffer one at a time.

The system is to be constrained to prevent overlap of buffer operations. i.e only one should access buffer at one time.

The problem is to make sure that the producer won't try to add data when buffer is full & consumer won't try to remove from it when it is empty..

Ex.

```
producer :

while (true) {

    b[in] = v;
    in++;
}
```

```
consumer :

while (true) {

    while (in<=out)
        /* do nothing */
    w = b[out];
    out++;
    /* consume */
```

→ Conclusion:

Topics Covered:

1. Semaphores
2. Types of Semaphores
3. Producer Consumer Problem.