ASSIGNMENT NO : 4

TITLE: PRODUCER-CONSUMER PROBLEM (USING SEMAPHORE OR MUTEX)

- **OBJECTIVE:**
  1. To study use of counting semaphore in linux.
  2. To study producer-consumer problem of operating system.
- **PROMBLEM STATEMENT:**

Thread synchronization using counting semaphores and mutual exclusion using mutex. Application to demonstrate: producer consumer problem with counting semaphores and mutex.

- **THEORY:**
  - **PRODUCER-CONSUMER PROBLEM:**

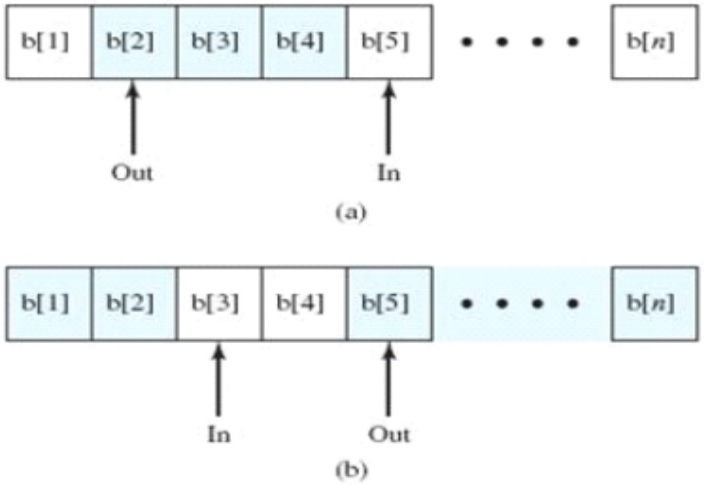
The general statement is this : there are one procedure generating some type of data(records,character) and placing these in a buffer. There is a single consumer that is talking items out of the buffer one at a time. The system is to be constrained to prevent the overlap of buffer operation. That is, only one agent(producer or consumer)may access the buffer at any one time.

The problem is to make sure that the producer won't try to add data in to the buffer if it is full and that the consumer won't try to remove data from an empty buffer. To begin, let us assume that the buffer is infinite and consist of a linear array elements.

We can define the producer and consumer functions as follows:

Bounded Buffer

Block on:	Unblock on:
Producer: insert in full buffer	Consumer: item inserted
Consumer: remove from empty buffer	Producer: item removed



➢ SEMAPHORE:

Semaphore is a variable or abstract data type that is used to control access to a common resource by multiple process in a concurrent system such as a multiprogramming operating system. to achive the desired effect, we can view the semaphore as a variable that has an integer value upon which three operations are defined:

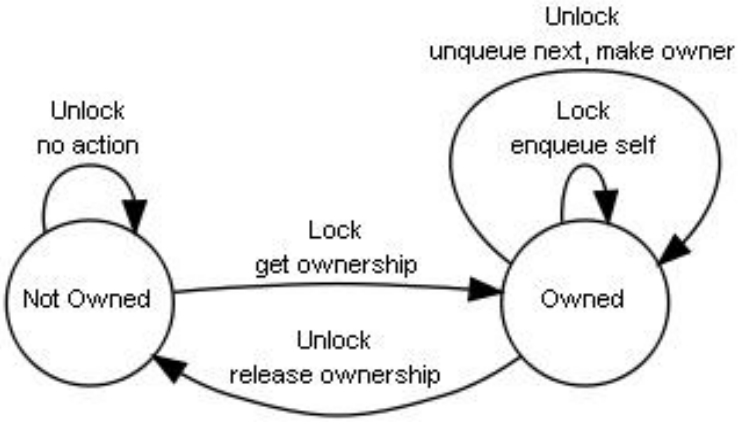
1. A semaphore may be initialized to a nonnegative value.
2. The wait operation decrements the semaphore value. If the value becomes negative, then the process executing the wait is blocked.
3. The signal operation increments the semaphore value. If the value is not positive, then a process blocked by await operation is unblocked.

There are two types of semaphore:

Binary semaphores: binary semaphore have 2 methods associated with it. (up, down/ lock, unlock).

Binary semaphore can take only two values(0/1). They are used to acquire locks. When a resource is available, the process in charge set the semaphore to 1 else 0.

Counting semaphores: counting semaphores represents the multiple resources.



➢ MUTEX FUNCTION:

Int pthread_mutex_init (pthread_mutex_t *mutex, const pthread_mutexattr_t *attr);

The pthread_mutex_init() function initializes the mutex referenced by mutex with attributes specified by attr. If attr is NULL ,the default mutex attributes are used; the effect is the same as passing the address of adefault mutex attribute object. Upon successful initialization,the state of the mutex becomes initialized and unlocked.

The mutex object referenced by mutex is locked by calling pthread_mutex_lock(). If the mutex is already locked, the calling thread blocks until the mutex becomes available. This operation returns with the mutex object referenced by mutex in the locked state with the calling thread as its owner. Int pthread_mutex_unlock (pthread_mutex_t *mutex).

• Header files:

1)#include<stdio.h> : This ANSI header file relates to "standard" input/output functions. Files, dvices and directories are referenced using pointers to objects of the type FILE . The header file contains declarations for these functions and macros, defines the FILE type, and contains various constants related to files.

2)include<stdlib.h> : This ANSI header file contains declarations for many standard functions, excluding those declared in other header files discussed in this section.

3)#include<sys/types.h> : This POSIX header file contains declarations for the types used by system-level calls to obtain file status or time information.

4)#include<pthread.h> : when we use pthread.h library in our program, program doesn't execute using simple compile command after including pthread.h header file, program doesn't compile and return error.

To compile c program with pthread.h library, you have to put –lpthread just after the compile command will tell to the compiler to execute program with pthread.h library.

The command is : gcc thread.c –o thread -lpthread

gcc is the compiler command.

Thread.c is the name of c program file.

-o is option to make header file.

Thread is the name of object file.

-lpthread is option to execute pthread.h library file.

5)#include<sys/stat.h> : The <sys/stat.h> header defines the structure of the data returned by the function fstat(),lstat(), and stat().

6)#include<semaphore.h> : The <semaphore.h> header shall define the sem_t type, used in performing semaphore operations. Inclusion of the <semaphore.h> header may make visible symbols defined in the header <sys/types.h>.

- **CONCLUSION:-**

  Thuse we have implemented producer-consumer problem using semaphore and mutex.