Task 6. [CO3]: Use of semaphores to control access to a shared device between processes**.**

AIM:

To write a program for Implement the Producer – Consumer problem using
semaphores(using
UNIX system calls).

ALGORITHM:

1. Start the process
2. Initialize buffer size
3. Consumer enters, before that producer buffer was not empty.
4. Producer enters, before check consumer consumes the buffer.
5. Stop the process.

PROGRAM:

```
#include<stdio.h>
int mutex=1,full=0,empty=3,x=0;
main()
{
int n;
void producer();
void consumer();
int wait(int);
int signal(int);
printf("\n1.PRODUCER\n2.CONSUMER\n3.EXIT\n");
while(1)
{
printf("\nENTER YOUR CHOICE\n");
scanf("%d",&n);
switch(n)
{
case 1:
if((mutex==1)&&(empty!=0))
producer();
else
printf("BUFFER IS FULL");
break;
case 2:
if((mutex==1)&&(full!=0))
consumer();
else
printf("BUFFER IS EMPTY");
break;
case 3:
exit(0);
break;
}
}
}
int wait(int s)
{
return(--s);
}
```

```
int signal(int s)
{
return(++s);
}
void producer()
{
mutex=wait(mutex);
full=signal(full);
empty=wait(empty);
x++;
printf("\nproducer produces the item%d",x);
mutex=signal(mutex);
}
void consumer()
{
mutex=wait(mutex);
full=wait(full);
empty=signal(empty);
printf("\n consumer consumes item%d",x);
x--;
mutex=signal(mutex);
}
```

OUTPUT:

```
[root@localhost ~]# ./a.out
1.PRODUCER
2.CONSUMER
3.EXIT
ENTER YOUR CHOICE
1producer produces the item1
ENTER YOUR CHOICE
1producer produces the item2
ENTER YOUR CHOICE
2consumer consumes item2
ENTER YOUR CHOICE
2consumer consumes item1
ENTER YOUR CHOICE
2BUFFER IS EMPTY
ENTER YOUR CHOICE
3
```