# Ex.No: 08 Bounded Buffer Problem using semaphore

**Date : 10.04.2021**

**Aim:**

# To write a program on bounded buffer problem using semaphore and to execute it.

**1.Write a bounded buffer problem using semaphore.**

**Algorithm:**

* Start
* Create a file with extension filename.c
* Enter the needed header files.
* It takes random numbers for execution.
* Using for loop,number of iterations are specified and executed.
* It displays the buffer content in it.
* As the for loop is infinite we should terminate using ctrl+c while output is displayed
* Stop.

**Program:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<pthread.h>

#include<unistd.h>

#include<time.h>

#include<semaphore.h>

#define THREAD\_NUM 8

sem\_t semEmpty;

sem\_t semFull;

pthread\_mutex\_t mutexBuffer;

int buffer[10];

int count=0;

int i;

void\* producer()

{

while(1)

{

int x=rand()%100;

sleep(1);

sem\_wait(&semEmpty);

pthread\_mutex\_lock(&mutexBuffer);

buffer[count]=x;

printf("Buffer Content:\n");

for(i=0;i<10;i++)

printf("%d\t",buffer[i]);

printf("\n");

count++;

pthread\_mutex\_unlock(&mutexBuffer);

sem\_post(&semFull);

}

}

void\* consumer()

{

while(1)

{

int y;

sem\_wait(&semFull);

pthread\_mutex\_lock(&mutexBuffer);

y=buffer[count-1];

count--;

pthread\_mutex\_unlock(&mutexBuffer);

sem\_post(&semEmpty);

printf("Got %d\n",y);

sleep(1);

}

}

int main(int argc,char\* argv[])

{

srand(time(NULL));

pthread\_t th[THREAD\_NUM];

pthread\_mutex\_init(&mutexBuffer,NULL);

sem\_init(&semEmpty,0,10);

sem\_init(&semFull,0,0);

int i;

for(i=0;i<THREAD\_NUM;i++)

{

if(i>0)

{

if(pthread\_create(&th[i],NULL,&producer,NULL)!=0)

{

perror('Failed to create thread');

}

}

else

{

if(pthread\_create(&th[i],NULL,&consumer,NULL)!=0)

{

perror('Failed to create thread');

}

}

}

for(i=0;i<THREAD\_NUM;i++)

{

if(pthread\_join(th[i],NULL)!=0)

{

perror('Failed to join thread');

}

}

sem\_destroy(&semEmpty);

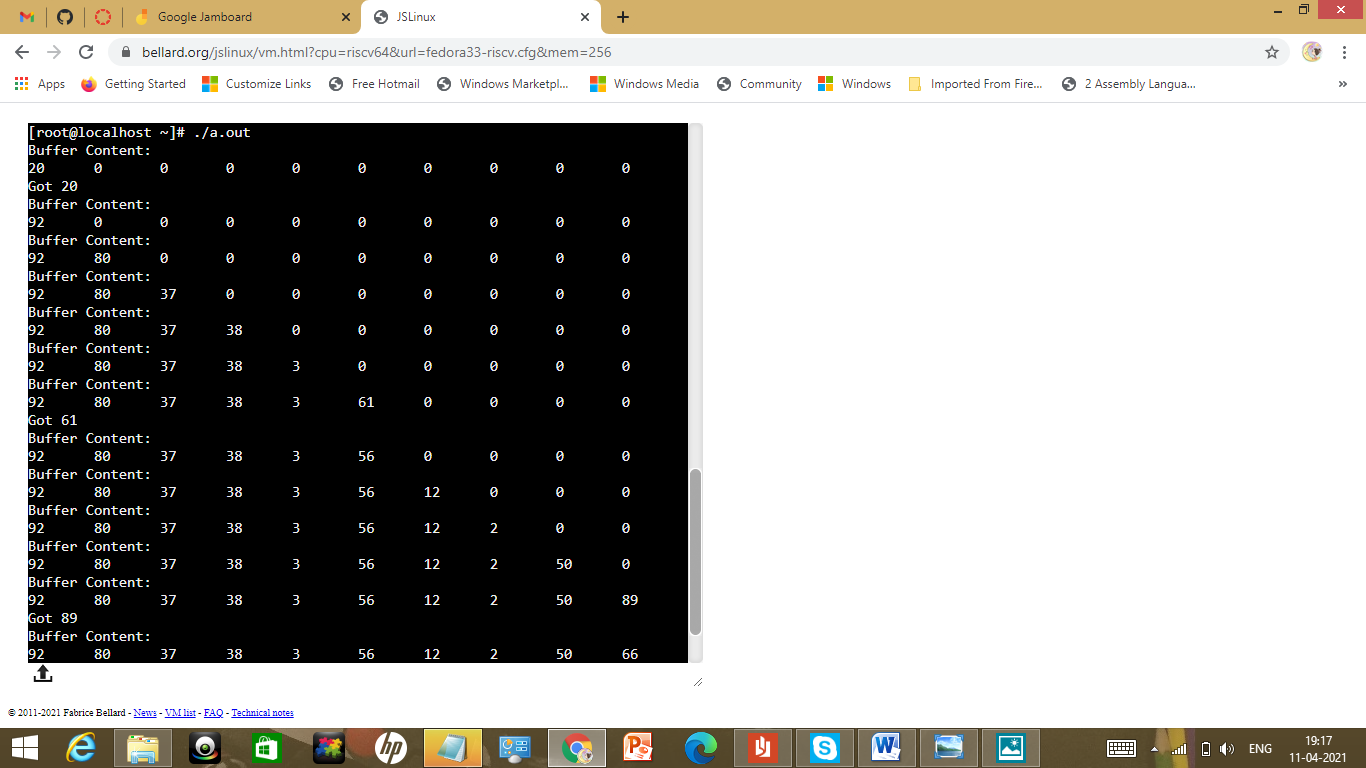
sem\_destroy(&semFull);

pthread\_mutex\_destroy(&mutexBuffer);

return 0;

}

**Output:**



|  |  |
| --- | --- |
| Observation(20) |  |
| Record(5) |  |
| Total(25) |  |
| Initial |  |

**Result:**

Thus the bounded buffer problem using semaphore were executed and outputs were noted.