**Name**: Akshat Rajesh Andhale **Class**: TEIT

**Roll** **no**: 03

**Assignment no: 04(A):** Producer Consumer problem with counting semaphores and mutexes

**CODE**:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <string.h>

#include <unistd.h>

#include <pthread.h>

#include <semaphore.h>

pthread\_t \*producers;

pthread\_t \*consumers;

sem\_t buf\_mutex,empty\_count,fill\_count;

int \*buf,buf\_pos=-1,prod\_count,con\_count,buf\_len;

int produce(pthread\_t self)

{

    int i = 0;

    int p = 1 + rand()%40;

    while(!pthread\_equal(\*(producers+i),self) && i < prod\_count)

    {

        i++;

    }

    printf("Producer %d produced %d \n",i+1,p);

    return p;

}

void consume(int p,pthread\_t self)

{

    int i = 0;

    while(!pthread\_equal(\*(consumers+i),self) && i < con\_count)

    {

        i++;

    }

    printf("Buffer:");

    for(i=0;i<=buf\_pos;++i)

        printf("%d ",\*(buf+i));

    printf("\nConsumer %d consumed %d \nCurrent buffer len: %d\n",i+1,p,buf\_pos);

}

void\* producer(void \*args)

{

    while(1)

    {

        int p = produce(pthread\_self());

        sem\_wait(&empty\_count);

        sem\_wait(&buf\_mutex);

        ++buf\_pos;          // critical section

        \*(buf + buf\_pos) = p;

        sem\_post(&buf\_mutex);

        sem\_post(&fill\_count);

        sleep(1 + rand()%3);

    }

    return NULL;

}

void\* consumer(void \*args)

{

    int c;

    while(1)

    {

        sem\_wait(&fill\_count);

        sem\_wait(&buf\_mutex);

        c = \*(buf+buf\_pos);

        consume(c,pthread\_self());

        --buf\_pos;

        sem\_post(&buf\_mutex);

        sem\_post(&empty\_count);

        sleep(1+rand()%5);

    }

    return NULL;

}

int main(void)

{

    int i,err;

    srand(time(NULL));

    sem\_init(&buf\_mutex,0,1);

    sem\_init(&fill\_count,0,0);

    printf("Enter the number of Producers:");

    scanf("%d",&prod\_count);

    producers = (pthread\_t\*) malloc(prod\_count\*sizeof(pthread\_t));

    printf("Enter the number of Consumers:");

    scanf("%d",&con\_count);

    consumers = (pthread\_t\*) malloc(con\_count\*sizeof(pthread\_t));

    printf("Enter buffer capacity:");

    scanf("%d",&buf\_len);

    buf = (int\*) malloc(buf\_len\*sizeof(int));

    sem\_init(&empty\_count,0,buf\_len);

    for(i=0;i<prod\_count;i++)

    {

        err = pthread\_create(producers+i,NULL,&producer,NULL);

        if(err != 0)

        {

            printf("Error creating producer %d: %s\n",i+1,strerror(err));

        }

        Else

{

            printf("Successfully created producer %d\n",i+1);

        }

    }

    for(i=0;i<con\_count;i++)

    {

        err = pthread\_create(consumers+i,NULL,&consumer,NULL);

        if(err != 0)

        {

            printf("Error creating consumer %d: %s\n",i+1,strerror(err));

        }

        else

        {

            printf("Successfully created consumer %d\n",i+1);

        }

    }

    for(i=0;i<prod\_count;i++)

    {

        pthread\_join(\*(producers+i),NULL);

    }

    for(i=0;i<con\_count;i++)

    {

        pthread\_join(\*(consumers+i),NULL);

    }

    return 0;

}

**OUTPUT**:

Enter the number of Producers:2

Enter the number of Consumers:3

Enter buffer capacity:5

Successfully created producer 1

Producer 1 produced 26

Successfully created producer 2

Successfully created consumer 1

Producer 2 produced 22

Successfully created consumer 2

Successfully created consumer 3

Buffer:26 22

Consumer 3 consumed 22

Current buffer len: 1

Buffer:26

Consumer 2 consumed 26

Current buffer len: 0

...Program finished with exit code 0

Press ENTER to exit console.