#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<pthread.h>

#include<semaphore.h>

#define TRUE 1

pthread\_mutex\_t mutex;

pthread\_t tid;

pthread\_attr\_t attr;

sem\_t full,empty;

int buffer;

void \*producer(void \*param);

void \*consumer(void \*param);

void initialize\_data()

{

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

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

sem\_init(&empty,0,1);

pthread\_attr\_init(&attr);

}

void \*producer(void \*param)

{

int item;

while(TRUE)

{

sleep(2);

printf("Enter item into buffer : ");

scanf("%d",&item);

sem\_wait(&empty);

pthread\_mutex\_lock(&mutex);

if(insert\_item(item))

{

printf("Producer finds an error");

}

else

{

printf("The item produced is : %d \n",item);

}

pthread\_mutex\_unlock(&mutex);

sem\_post(&full);

}

}

void \*consumer(void \*param)

{

int item;

while(TRUE)

{

sleep(2);

sem\_wait(&full);

pthread\_mutex\_lock(&mutex);

if(remove\_item(&item))

{

printf("Consumer finds an error");

}

else

{

printf("The item consumed is: %d \n", item);

}

pthread\_mutex\_unlock(&mutex);

sem\_post(&empty);

}

}

int insert\_item(int item)

{

buffer=item;

return(0);

}

int remove\_item(int \*item)

{

\*item=buffer;

return(0);

}

int main()

{

int x;

printf("\nEnter the main sleep time: ");

scanf("%d",&x);

int main\_sleep\_time=x;

initialize\_data();

pthread\_create(&tid,&attr,producer,NULL);

pthread\_create(&tid,&attr,consumer,NULL);

sleep(main\_sleep\_time);

return (0);

}

/\*

Enter the main sleep time: 20

Enter item into buffer : 2

The item produced is : 2

The item consumed is: 2

Enter item into buffer : 4

The item produced is : 4

The item consumed is: 4

Enter item into buffer : 6

The item produced is : 6

The item consumed is: 6

Enter item into buffer : 8

The item produced is : 8

The item consumed is: 8

Enter item into buffer : 10

The item produced is : 10

The item consumed is: 10

\*/