#include<pthread.h>

#include<semaphore.h>

#include<stdio.h>

#include<stdlib.h>

#define maxitems 5

#define buffersize 5

sem\_t empty;

sem\_t full;

int in=0;

int out=0;

int buffer[buffersize];

pthread\_mutex\_t mutex;

void \*producer(void \*pno)

{

int item;

for(int i=0;i<maxitems;i++)

{

item=rand();

sem\_wait(&empty);

pthread\_mutex\_lock(&mutex);

buffer[in]=item;

printf("Producer %d:insert item %d at %d \n",((int)pno),buffer[in],in);

in=(in+1)%buffersize;

pthread\_mutex\_unlock(&mutex);

sem\_post(&full);

}

}

void \*consumer(void \*cno)

{

for(int i=0;i<maxitems;i++)

{

sem\_wait(&full);

pthread\_mutex\_lock(&mutex);

int item=buffer[out];

printf("Consumer %d:Remove item %d at %d \n",((int)cno),item,out);

out=(out+1)%buffersize;

pthread\_mutex\_unlock(&mutex);

sem\_post(&empty);

}

}

int main()

{

pthread\_t pro[5],con[5];

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

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

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

int a[5]={1,2,3,4,5};

for(int i=0;i<5;i++)

{

pthread\_create(&pro[i],NULL,(void\*)producer,(void\*)&a[i]);

}

for(int i=0;i<5;i++)

{

pthread\_create(&con[i],NULL,(void\*)consumer,(void\*)&a[i]);

}

for(int i=0;i<5;i++)

{

pthread\_join(pro[i],NULL);

}

for(int i=0;i<5;i++)

{

pthread\_join(con[i],NULL);

}

pthread\_mutex\_destroy(&mutex);

sem\_destroy(&empty);

sem\_destroy(&full);

return 0;

}