11 YASHRAJ DEEPAK DEVRAT

#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<pthread.h>

#include<semaphore.h>

#define BUFFERSIZE 5

sem\_t empty,full;

pthread\_mutex\_t mutex; int

buffer[BUFFERSIZE];

long int num;

int in=0,out=0;

void \*producer\_process (void \*arg)

{

long int num = (long int) arg;

sem\_wait(&empty);

pthread\_mutex\_lock(&mutex);

int item =rand() % 10 ;

printf("\n Producer[%ld] has produced %d ",num+1,item);

sleep(2); buffer[in] =item ;

in =(in+1) % BUFFERSIZE;

pthread\_mutex\_unlock(&mutex);

sem\_post(&full);

return NULL;

}

void \*consumer\_process (void \*arg)

{

long int num = (long int) arg;

sem\_wait(&full);

pthread\_mutex\_lock(&mutex);

int item =rand() % 10 ;

printf("\n consumer[%ld] has consumed %d ",num+1,item);

sleep(2);

// buffer[in] =item ;

out =(out+1) % BUFFERSIZE;

pthread\_mutex\_unlock(&mutex);

sem\_post(&empty);

return NULL;

}

int main()

{

int numProducers, numConsumers;

pthread\_t producers[10],consumers[10];

long int i;

printf("\n Enter number of producers :");

scanf("%d", &numProducers);

printf("\n Enter number of consumers :");

scanf("%d", &numConsumers);

sem\_init(&empty,0,5);

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

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

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

pthread\_create(&producers[i], NULL, producer\_process,(void \*)i);

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

pthread\_create(&consumers[i], NULL, consumer\_process,(void \*)i);

for(i=0; i<numConsumers

; i++)

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

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

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

return 0;

}

