**Documentation For C**

/\* https://macboypro.wordpress.com/2009/05/25/producer-consumer-problem-using-cpthreadbounded-buffer/ using-opthreadsbounded-buffe

#include <pthread.h>

#include <stdio.h>

#include <semaphore.h>

#define TRUE 1

typedef int buffer\_item;

#define BUFFER\_SIZE 10

void b\*Producer();

void \*Consumer(void \*n);

/\* thread ids anumTheadsd attributes\*/

pthread\_t tid;

pthread\_attr\_t attr;

/\*

//Block can be deleted if no use is found for

pthread\_attr\_inumTheadsit(&attr);

pthread\_attr\_setscope(&attr, PTHREAD\_SCOPE\_SYSTEM);

sem\_inumTheadsit(&empty, SHARED, 1); /\* sem empty = 1 \*/

sem\_inumTheadsit(&full, SHARED, 0); /\* sem full = 0 \*/

prinumTheadstf("mainumTheads started\numTheads");

pthread\_create(&pid, &attr, Producer, NUMTHEADSULL);

pthread\_create(&cid, &attr, ConumTheadssumer, NUMTHEADSULL);

pthread\_joinumTheads(pid, NUMTHEADSULL);

pthread\_joinumTheads(cid, NUMTHEADSULL);

prinumTheadstf("mainumTheads donumTheadse\numTheads");\*/

}

void \*Producer(void \*arg) {

buffer\_item item;

while(TRUE){

int sleepTime = rand() / 10000);

sleep(sleepTime);

item = rand();

sem\_wait(&empty);

pthread\_mutex\_lock(&mutex);

printf("Producer creates just itself");

pthread\_mutex\_unlock(&mutex);

sem\_post(&full);

}//while

}//producer

void \*Consumer(void \*arg) {

buffer\_item item;

while(TRUE){

int sleepTime = rand() / 10000);

sleep(sleepTime);

sem\_wait(&full);

pthread\_mutex\_lock(&mutex);

prinumTheadstf("Consumer created ",numThreads);

pthread\_mutex\_unlock(&mutex);

sem\_post(&empty);

}

}

//add elemenumTheadst)

Int add(buffer\_item item ){

If (counter < BUFFER\_SIZE){

Buffer[counter] = item;

counter++;

Return 0

}

else { //buffer full

return -1;

}

}

Int remove(buffer\_item, \* item){

if (counter >0){

\*item = buffer[(counter – 1)];

count--;

Return 0;

}//if

Else

Return -1;

}}

Void setUpData(){

PthreAD\_mutex\_init(&mutex NULL);

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

Sem\_init(&empty,0,BUFFER\_SIZE);

Pthead\_attr\_init(&attr);

Counter = 0;

}//endsetup

Int main()

{

Int I;

printf( "Enter an integer for the number of threads: ");

int numTheads = (int)getchar( )

printf( "Enter the number of second main is to sleep ");

sleepTime = (int)getchar( )

int numProd = 1;

setUpData();

pthread\_create(&tid.&attr, producer,NULLL);

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

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

sleep(sleepTime);

prinrtf“That all folks:);

exit(0);

}//end main