#include <stdio.h>

#include <pthread.h>

#include <windows.h>

#define N 100

#define true 1

#define producerNum 10

#define consumerNum 5

#define sleepTime 1000

typedef int semaphore;

typedef int item;

item buffer[N] = {0};

int in = 0;

int out = 0;

int proCount = 0;

semaphore mutex = 1, empty = N, full = 0, proCmutex = 1;

void \* producer(void \* a){

while(true){

while(proCmutex <= 0);

proCmutex--;

proCount++;

printf

proCmutex++;

while(empty <= 0){

printf("缓冲区已满！\n");

}

empty--;

while(mutex <= 0);

mutex--;

buffer[in] = proCount;

in = (in + 1) % N;

mutex++;

full++;

Sleep(sleepTime);

}

}

void \* consumer(void \*b){

while(true){

while(full <= 0){

printf("缓冲区为空！\n");

}

full--;

while(mutex <= 0);

mutex--;

int nextc = buffer[out];

buffer[out] = 0;//消费完将缓冲区设置为0

out = (out + 1) % N;

mutex++;

empty++;

printf

Sleep(sleepTime);

}

}

int main()

{

pthread\_t threadPool[producerNum+consumerNum];

int i;

for(i = 0; i < producerNum; i++){

pthread\_t temp;

if(pthread\_create(&temp, NULL, producer, NULL) == -1){

printf

exit(1);

}

threadPool[i] = temp;

}//创建生产者进程放入线程池

for(i = 0; i < consumerNum; i++){

pthread\_t temp;

if(pthread\_create(&temp, NULL, consumer, NULL) == -1){

printf("ERROR, fail to create consumer%d\n", i);

exit(1);

}

threadPool[i+producerNum] = temp;

}//创建消费者进程放入线程池

void \* result;

for(i = 0; i < producerNum+consumerNum; i++){

if(pthread\_join(threadPool[i], &result) == -1){

printf("fail to recollect\n");

exit(1);

}

}//运行线程池

return 0;

}