#include <stdio.h>

#include <windows.h>

#define N 10

typedef int semaphore; /\* 信号量是一种特殊的整型变量 \*/

semaphore mutex=1; /\* 互斥访问 \*/

semaphore empty=N; /\* 记录缓冲区中空的槽数 \*/

semaphore full=0; /\* 记录缓冲区中满的槽数\*/

semaphore buf[N]; /\* 有N个槽数的缓冲区buf[N]，并实现循环缓冲队列 \*/

semaphore front=0, rear=0;

void p(semaphore \*x) /\* p操作 \*/

{

\*x=(\*x)-1;

}

void v(semaphore \*y) /\* v操作 \*/

{

\*y=(\*y)+1;

}

void produce\_item(int \*item\_ptr)

{

/\*printf("produce an item\n");\*/

\*item\_ptr='m'; /\* 'm' is "man满" \*/

}

void enter\_item(int x)

{

buf[front]=x;

printf("输入了 %c 到缓冲区 buf[%d]\n", buf[front], front);

front=(front+1)%N;

}

void remove\_item(int \*yy)

{

printf("从缓冲区 buf[%d] 取走 %c", rear,buf[rear]);

\*yy=buf[rear];

buf[rear]='k'; /\* 'k' is "kong空" \*/

printf(" 因此缓冲区 buf[%d] 变成空了，用%c表示\n", rear, buf[rear]);

rear=(rear+1)%N;

}

void consume\_item(int y)

{

printf("消费从缓冲区取出的产品 %c\n", y);

}

void producer();

void consumer();

/\* 生产者 \*/

void producer()

{

int item;

while(1){

Sleep(1000);

produce\_item(&item);

p(&empty); /\* 递减空槽数 \*/

p(&mutex); /\* 进入临界区 \*/

enter\_item(item); /\* 将一个新的数据项放入缓冲区 \*/

v(&mutex); /\* 离开临界区 \*/

v(&full); /\* 递增满槽数 \*/

if(full==N) /\* 若缓冲区满的话，唤醒消费者进程 \*/

consumer();

}

}

/\* 消费者 \*/

void consumer()

{

int get\_item;

while(1){

Sleep(1000);

p(&full); /\* 递减满槽数 \*/

p(&mutex); /\* 进入临界区 \*/

remove\_item(&get\_item); /\* 从缓冲区中取走一个数据项 \*/

v(&mutex); /\* 离开临界区 \*/

v(&empty); /\* 递增空槽数 \*/

consume\_item(get\_item); /\* 对数据项进行操作（消费）\*/

if(empty==N) /\* 若缓冲区全空的话，唤生产者进程 \*/

producer();

}

}

/\* 调用生产者－消费者进程实现进程间同步 \*/

int main()

{

producer();

return 0;

}