
云南大学数学与统计学实验教学中心

实验报告

课程名称：操作系统实验	学期：2015~2016 学年上学期	成绩：
指导教师： 彭程	学生姓名：金洋	学号：20131910023
实验名称：生产者和消费者		
实验编号：三	实验日期： 11 月 11 日	实验学时： 2
学院： 数学与统计学院	专业： 信息与计算科学	年级： 2013 级

一、实验目的

模拟生产者、消费者进程。

二、实验内容

3 个生产者，2 个消费者，它们共享一个有界缓冲池，缓冲池有四个缓冲区，生产者向其中投放消息，消费者从中取得消息。只要缓冲池未空，生产者便可将消息送入缓冲池；只要缓冲池未满，消费者便可从缓冲池取走一个消息。

1. 有一临界资源，有三个缓冲区；
2. 有 3 个生产者，2 个消费者；
3. 要求
 - ①写出流程图；
 - ②用 VC 实现；
 - ③试把程序改为 3 个消费者，2 个生产者，结果与前面实现的进行比较；

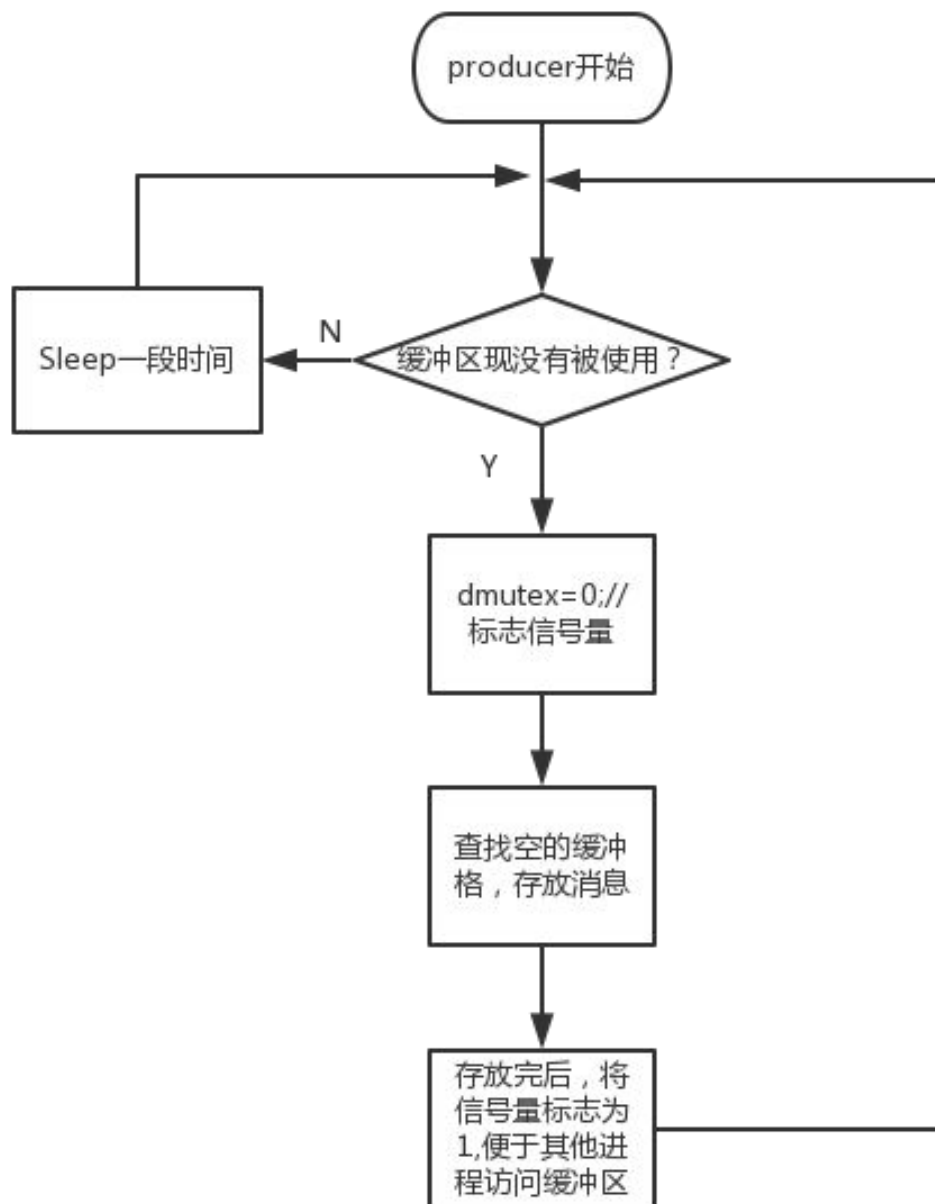
三、实验环境

平台：Visual C++ 6.0

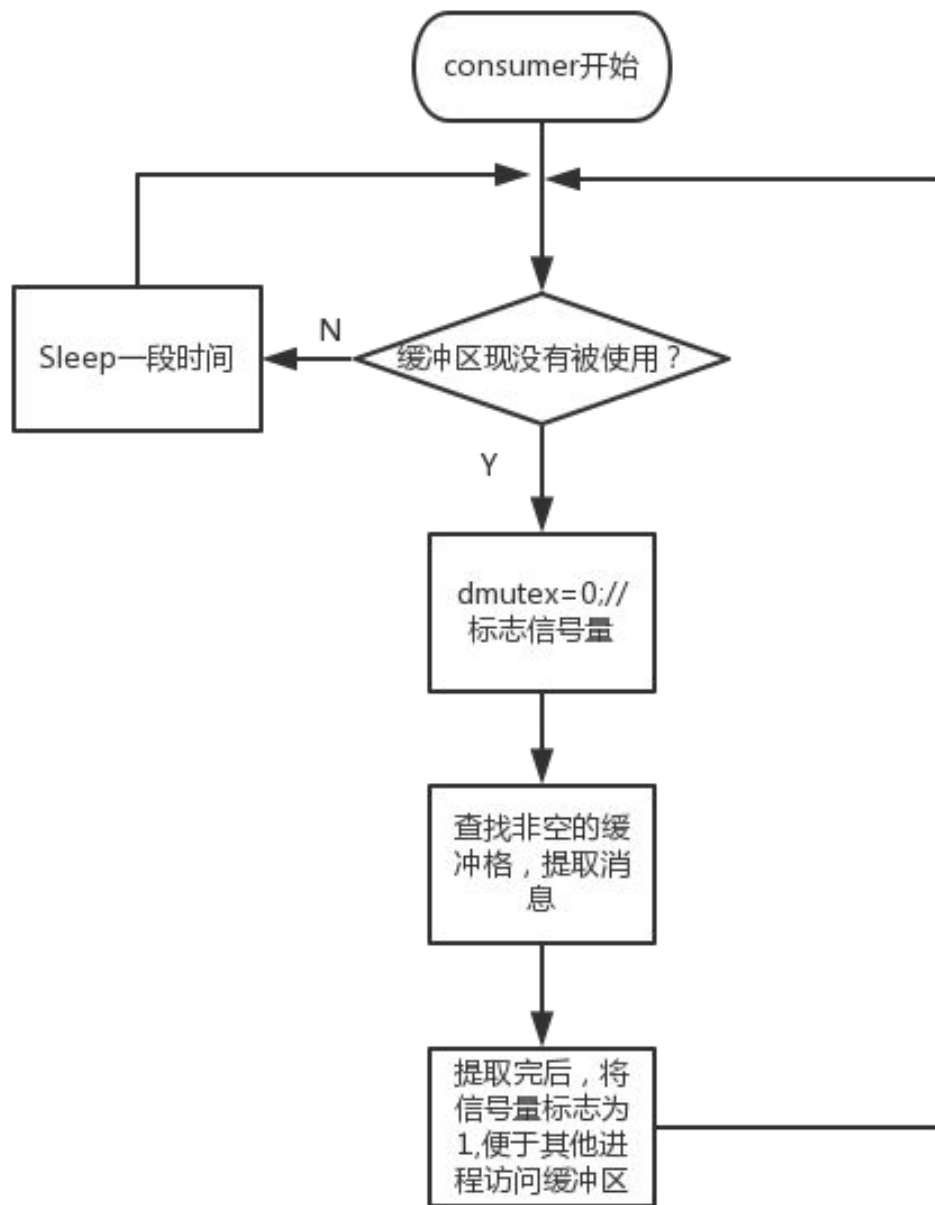
语言：C 语言

四、算法介绍

生产者进程流程图：



消费者进程流程图：



五、调试过程

1、程序代码

```
#include <stdio.h>
#include <windows.h>
#include <time.h>
#include <iostream.h>

int dmutex=1;//互斥信号量
int buffer[3]={0,0,0};//缓冲区状态，1 表示有消息，0 表示无消息

/*打印缓冲区状态*/
void display()
{
    printf("buffers
statu:%d,%d,%d;\n",buffer[0],buffer[1],buffer[2]);
}

/*生产者进程*/
DWORD WINAPI producer()
{
    int i;
    while (1)
    {

        /*若没有生产者正在向缓冲区存放消息，且没有消费者正向缓冲区提取
        消息，则此时生产者可以向缓冲区存放消息*/
        if (dmutex==1)
        {
            dmutex=0;//标志信号量

            /*查找空的缓冲格，存放消息*/
            while (1)
            {
```

```

        for (i=0;i<3;i++)
            if (buffer[i]==0){
                buffer[i]++;
                printf("Produce an item to buffer %d",i);
                display();
                break;
            }
        if (i<3)
        {
            /*存放完后，将信号量标志为 1,便于其他进程访问缓冲区
*/
            dmutex=1;
            break;
        }
        /*未找到空的缓冲格，重新查找*/
        else Sleep(rand()%1000+1);
    }

}

/*否则继续等待*/
else Sleep(rand()%1000+1);
}
}

/*消费者进程*/
DWORD WINAPI consumer()
{
    int i;

    while (1)
    {
        /*若没有生产者正在向缓冲区存放消息，且没有消费者正向缓冲区提取
        消息，则此时消费者可以从缓冲区提取消息*/

```

```

    if (dmutex==1)
    {
        dmutex=0;

        /*查找非空的缓冲格，提取消息*/
        while (1)
        {
            for (i=0;i<3;i++)
                if (buffer[i]==1){
                    buffer[i]--;
                    printf("Consume an item from buffer %d,",i);
                    display();
                    break;
                }

            if (i<3)
            {
                /*存放完后，将信号量标志为 1,便于其他进程访问缓冲区
*/
                dmutex=1;
                break;
            }
            /*未找到非空的缓冲格，重新查找*/
            else Sleep(rand()%1000+1);

        }
    }

    /*否则继续等待*/
    else Sleep(rand()%1000+1);

}

}

```

```
int main(){

    HANDLE handle[5];
    DWORD dw1,dw2,dw3,dw4,dw5;
    srand((unsigned) time(NULL));

    handle[0]=CreateThread(NULL,0,producer,NULL,0,&dw1);
    producer();

    Sleep(rand()%1000+1);
    handle[1]=CreateThread(NULL,0,consumer,NULL,0,&dw4);
    consumer();

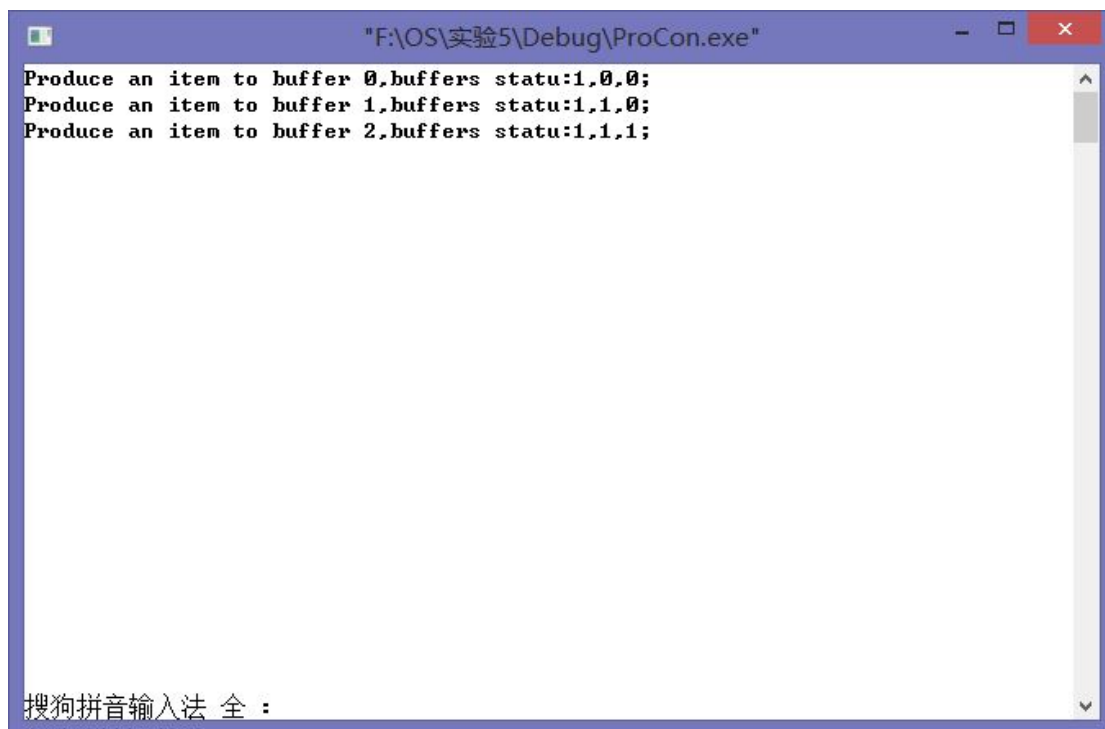
    Sleep(rand()%1000+1);
    handle[2]=CreateThread(NULL,0,producer,NULL,0,&dw2);
    producer();

    Sleep(rand()%1000+1);
    handle[3]=CreateThread(NULL,0,consumer,NULL,0,&dw5);
    producer();

    Sleep(rand()%1000+1);
    handle[4]=CreateThread(NULL,0,producer,NULL,0,&dw3);
    consumer();

}
```

2. 运行结果



```
"F:\OS\实验5\Debug\ProCon.exe"
Produce an item to buffer 0,buffers statu:1,0,0;
Produce an item to buffer 1,buffers statu:1,1,0;
Produce an item to buffer 2,buffers statu:1,1,1;

搜狗拼音输入法 全 :
```

六、总结

1. 首先要考虑生产者与消费者对缓冲区操作时的互斥。
2. 不管生产者与消费者有多少个，缓冲池有多少个缓冲区。都只有二个同步过程——分别是生产者要等待有空缓冲区才能投放产品，消费者要等待有非空缓冲区才能去取产品。
3. 线程（进程）同步的主要任务：在引入多线程后，由于线程执行的异步性，会给系统造成混乱，特别是在急用临界资源时，如多个线程急用同一台打印机，会使打印结果交织在一起，难于区分。当多个线程急用共享变量，表格，链表时，可能会导致数据处理出错，因此线程同步的主要任务是使并发执行的各线程之间能够有效的共享资源和相互合作，从而使程序的执行具有可再现性。

七、参考文献

- [1]汤小丹，梁红兵，哲凤屏，汤子瀛. 计算机操作系统[M]（第三版）. 西安：西安电子科技大学出版社，2007 年 5 月；
- [2]谭浩强著. c 程序设计[M]（第三版）. 北京：清华大学出版社. 2005. 7；

八、教师评语