

电子科技大学

计算机专业类课程

实验报告

课程名称：计算机操作系统
学 院：计算机科学与工程学院
专 业：计算机科学与技术
学生姓名：吴思扬
学 号：2013060105016
指导教师：薛瑞尼

日期： 2016 年 6 月 4 日

电子科技大学 实 验 报 告 实验一

学生姓名：吴思扬

学 号：2013060105016

一、 实验项目名称： 生产者消费者问题

共享缓冲区中放置一个数字，取值范围为 $[0, 10]$ ，初值为0。生产者将此值加1，消费者将此值减1。

1. 场景1

- * 同一进程内启动一组生产者线程和一组消费者线程
- * 缓冲区为本进程的全局变量

2. 场景2

- * 启动一组生产者进程和一组消费者进程
- * 同一个数据文件为缓冲区

* 输入

* p ：生产者数量

* c ：消费者数量

* 输出

打印当前共享缓冲区中的数值，或者生产者消费者的状态。

二、实验原理：

//

```

//  main.cpp
//  lab1
//
//  Created by 吴思扬 on 16/6/10.
//  Copyright © 2016年 吴思扬. All rights reserved.
//

#include<iostream>
#include <thread>
#include <mutex>
#include <unistd.h>
using namespace std;

int Buffer = 4;

mutex Lock_mutex;

int I = 0;

int for_num;

void Producer(int num_producer)
{
    int waiting = 0;
    while (I<for_num)
    {
        Lock_mutex.lock();
        if (Buffer < 10)
        {
            if (waiting==1)
            {
                waiting = 0;
                printf("# Producer %d: resumed\n",num_pro-
ducer);
            }
            Buffer++;
            printf("# Producer %d : %d -> %d\n", num_pro-
ducer,Buffer-1,Buffer);
        }
        else if(waiting==0)
        {
            printf ("# Producer %d :waiting\n",num_producer);
            waiting = 1;
        }
        Lock_mutex.unlock();
        if (waiting==0)
        {
            I++;
            sleep(10);
        }
    }
}

```

```

void Consumer(int num_consumer)
{
    int waiting = 0;
    while (I<for_num)
    {
        Lock_mutex.lock();
        if (Buffer >0)
        {
            if (waiting==1)
            {
                waiting = 0;
                printf("# Consumer %d: resumd\n",num_consumer);
            }
            Buffer--;
            printf("# Consumer %d : %d -> %d\n", num_consumer,Buffer+1,Buffer);
        }
        else if(waiting==0)
        {
            printf ("# Consumer %d :waiting\n",num_consumer);
            waiting = 1;
        }
        Lock_mutex.unlock();
        if (waiting==0)
        {
            I++;
            sleep(10);
        }
    }
}

```

```

int main()
{
    int num_producer, num_consumer;
    printf("请输入生产者数量\n" );
    scanf("%d",&num_producer);
    printf("请输入消费者数量\n");
    scanf("%d",&num_consumer);
    printf("运行次数\n");
    scanf("%d",&for_num);

    thread Tread_pro[10];
    thread Tread_con[10];

    for (int i = 0; i < num_producer; i++){

```

```

        Tread_pro[i] = thread(Producer,i);
    }

    for (int i = 0; i < num_consumer; i++){
        Tread_con[i] = thread(Consumer,i);
    }

    for (int i = 0; i < num_producer; i++){
        Tread_pro[i].join();
    }

    for (int i = 0; i < num_consumer; i++){
        Tread_con[i].join();
    }
    return 0;
}

```

七、实验器材（设备、元器件）：

PC Xcode

八、实验步骤及数据结果分析：

buffer 初始值为 4，输入消费者数量和生产者数量，输出结果如下图所示：

请输入生产者数量

4

请输入消费者数量

4

运行次数

5

Producer 0 : 4 -> 5

Producer 3 : 5 -> 6

Producer 1 : 6 -> 7

Consumer 0 : 7 -> 6

Consumer 1 : 6 -> 5

Consumer 2 : 5 -> 4

Consumer 3 : 4 -> 3

Producer 2 : 3 -> 4

Program ended with exit code: 0

All Output ↕



报告评分:

指导教师签字: