



大连海事大学

操作系统课程实践报告

专业班级：计科一班

姓 名：王郁博

选课序号：31

学 号：2220183524

指导老师：米泽田

实验日期：2020 年 1 月 4 日到 1 月 15 日

目录

任务分配.....	3
1. 题目名称.....	3
实验选题.....	3
实验环境.....	3
问题描述.....	3
1.1.生产者消费者问题.....	3
1.2.线程并发执行.....	4
1.2.1.数独解决方案验证器.....	4
1.2.2.多线程排序.....	4
2. 问题分析.....	4
1.1.生产者消费者问题.....	4
1.2.线程并发执行问题.....	5
1.2.1.数独解决方案验证器.....	5
1.2.2.多线程排序.....	6
3. 解决方案.....	6
生产者消费者问题的 Linux 下 pthread 实现.....	6
Windows 下的图形化界面实现.....	11
数独解决方案验证器辅助.....	13
4. 程序运行结果展示.....	14
1) Linux 系统下生产者消费者.....	14
2) Windows 系统下生产者消费者.....	16
3) Linux 下数独解决方案验证器.....	17
4) Windows 下数独解决方案验证器.....	18
5) 多线程排序（6 线程）.....	19
5.重点、难点分析与课程设计体会.....	21
重点难点分析.....	21
课程设计体会.....	21
pthread 与 Windows API 区别.....	22

操作系统课程实践

任务分配

姓名	任务分配	百分比
廖泰宇	数独问题的 Windows 下实现, 生产者消费者问题辅助	25%
万文晓	生产者消费者问题的 Windows API 实现、数独问题的 Linux 下实现、辅助完成多线程排序	25%
王郁博	生产者消费者问题的 Linux 下 pthread 实现、Windows 下的图形化界面实现、数独问题辅助	25%
王晗	生产者消费者问题的 Linux 下 pthread 辅助实现、多线程排序问题 Windows 实现	25%

1. 题目名称

实验选题

1.1 生产者消费者问题、1.2 线程并发执行

实验环境

生产者消费者问题:

Linux: VMware Workstation Pro 虚拟机下的 Ubuntu32 位系统

Windows: Virtual C++ 6.0 (加 easyx 插件)

数独解决方案验证器:

Linux: VMware Workstation Pro 虚拟机下的 Ubuntu32 位系统

Windows: Code Blocks

多线程排序:

Code Blocks

问题描述

1.1.生产者消费者问题

预期效果:

1. 多个生产者、消费者线程并发执行
2. 用队列进行数据存储

3. 写入、读出字母
4. 在 Linux 下使用 pthread 实现代码
5. 在 Windows 下使用 Windows API 实现代码
6. 对单步操作与整体程序的计时计数
7. Windows 系统下的图形化界面采用三个信号量：empty(以记录有多少空的缓冲区)、full(以记录有多少满的缓冲区)及 mutex(二进制信号量或互斥信号量，以保护对缓冲区插入与删除的操作)
8. 写入内容为随机字母

1.2.线程并发执行

1.2.1.数独解决方案验证器

Linux 预期效果：

1. 9 个线程并发执行，分别完成行、列、3*3 九宫格的验证
2. 对于验证的结果进行输出
3. 报错、报出具体的行列以及九宫格出错

Windows 预期效果：

1. 27 个线程并发执行，分别完成 9 行、9 列、3*3 个九宫格的验证
2. 判断数独是否有效
3. 对数独元素是否越界，重复进行判断，并输出元素值及行列值

1.2.2.多线程排序

windows 预计效果（3,5,6 组线程）：

- 1.生成 90000 个随机数并分为 6 组，提供给 6 个线程
2. 6 个线程并发执行，分别完成冒泡排序、选择排序、插入排序
- 3.显示 6 个线程对应时间和排序后数组内容

2. 问题分析

1.1.生产者消费者问题

设计思想：

数据结构：采用队列管理字符数组。在 Linux 中通过 typedef 定义 buffer_item 型变量，之后利用 buffer_item 变量定义队列 Queue。

队列有关函数：进队函数：通过 int 型参数在字母表中查找字母，写入数组

并更改尾指针。出队函数：为 `char` 型参数赋值为头指针指向的字母，并更改头指针。队列空查询：比较头尾指针，若头尾指针相等则队列为空，返回 `TRUE`，否则返回 `FALSE`。队列满查询：比较头尾指针，若尾指针的值加一并用队列长度取余后与头指针相等则队满，返回 `TRUE`，否则返回 `FALSE`。队列内容输出：将队列头尾指针取出，根据取出的头指针顺序读取输出队列内容，不足 5 个的部分用 * 填充。

主要函数：主函数：设置截止时间等参数，设置信号量，创建 3 个生产者线程和 4 个消费者线程，最后将线程阻塞等待线程结束。生产者线程：首先对 `empty`、`mutex` 进行 `p` 语句操作，随后取随机数并根据该随机数从字母数组中取值存入缓冲区堆栈，最后对 `full`、`mutex` 进行 `v` 语句操作，并睡眠等待下一次循环。生产者线程的结束条件为到达截止时间。消费者线程：首先对 `full`、`mutex` 进行 `p` 语句操作，随后从缓冲区堆栈中取字母，最后对 `empty`、`mutex` 进行 `v` 语句操作，并睡眠等待下一次循环。生产者线程的结束条件为到达截止时间且缓冲区队列为空。

重点难点：

数据结构的设计，线程函数的传参，输出内容的调试，图形化界面的实现，Linux 下模糊 bug 的调试。

1.2.线程并发执行问题

1.2.1.数独解决方案验证器

Linux 下

1. 设计三个函数分别验证所有行、所有列、每个 3*3 的九宫格
2. 创建 9 个线程
线程 1：验证所有行
线程 2：验证所有列
线程 3-11：验证每个 3*3 的九宫格
3. 读取文件来进行数独的输入

Linux 下难点：

1. 对于验证 3*3 的九宫格的不同线程的传参问题，定位到相应要验证的区域
2. 验证函数的设计

Windows 下：

1. 设计 27 个线程来调用函数分别验证 9 行、9 列、9 个 3*3 的九宫格
2. 通过读取文件内容初始化待验证数组
3. 为待验证的数组元素设计一个坐标，再定义一个边缘左上角，一个边缘右下角坐标来固定每行、每列、每个 3*3 的九宫格，将这两个边缘坐标包含的 9 个元素传递给线程调用函数识别。

Windows 下难点：

1. 如何用线程调用函数
2. 如何用函数来验证数独

1.2.2.多线程排序

- 1.设计排序算法：冒泡算法、选择算法、排序算法
- 2.设计随机数生成（包括对于 for 循环中随机数种子的替换）
- 3.对于线程进行计时
- 4.显示每个线程中数组的内容

难点：

- 1.计时函数的使用
- 2.创造伪随机数的函数 rand()在 for 循环中由于间隔太短，只能随机到同一个随机数表的同一个值，所以要想有较随机的随机数：

（1）需要在 for 循环中插入一个随机数种子函数 srand()，用随机数加上时钟函数 time()作为新的随机数种子，刷新随机数的生成

（2）在生成一个随机数之后加入 Sleep(1000)函数进行等待，在等待 1 秒后能稳定读取下一个随机数表的数值。显然，这在生成大量随机数的时候并不适用。

3. 解决方案

我负责的部分是生产者消费者问题的 Linux 下 pthread 实现与 Windows 下的图形化界面实现，并辅助修改完成了数独解决方案验证器。

生产者消费者问题的 Linux 下 pthread 实现

在生产者消费者问题的 Linux 下 pthread 实现方面，我参考了《计算机系统实验与课程设计指导书》的“实验内容 11 生产者与消费者问题的实现”的代码示例，在模仿学习了代码的创建阻塞线程方法，信号量的使用，线程处理函数的写法后完成基础代码。

在线程创建中，使用 pthread_create 函数创建进程。通过该函数的第四个参数向线程的服务函数传参。

```
int Numflag1=1;
printf("[CREATED] Producer ID:1\n");
ret=pthread_create(&id1,NULL,(void *)productor,(void *)&Numflag1);//生产者线程
if(ret!=0)
```

```
    perror("pthread cread1");
```

在创建线程后，使用：

```
pthread_join(id1,NULL);
```

进行线程阻塞，该函数用于一直阻塞调用线程，直到指定的线程终止。当 pthread_join()返回之后，应用程序可回收与已终止线程关联的任何数据存储空间。

在对于信号量的使用上，使用如下代码实现：

```
sem_wait(&empty); //p(empty)
sem_wait(&mutex); //p(mutex)
```

.....

```
sem_post(&full); //v(full)
sem_post(&mutex); //v(mutex)
```

其中 sem_wait 作用相当于 p 语句, sem_post 作用相当于 v 语句, 来实现线程间的互斥操作与限制。

在基础代码基础上, 首先加入了计数语句:

```
start0=clock(); //计时
```

.....

```
end0=clock();
double seconds =(double)(end0 - start0)/CLOCKS_PER_SEC;
printf("All time used is: %.8f\n", seconds);
```

类似语句加入到了主函数的头尾与线程函数中每步操作的头尾来记录所用时间并输出到控制台。

加入了取随机数来使得生产者线程中写入缓冲区的字母为随机字母的代码如下:

```
srand((unsigned)time(NULL));
p = rand() % 52;
EnQueue(qt,p);
```

在这段代码中, 首先使用 srand 为随机数生成重新设置种子, 使用 rand 函数取得随机数, 之后将随机数作为参数传递给入队函数 EnQueue(qt,p), 来在字母表中查询并写入队列。

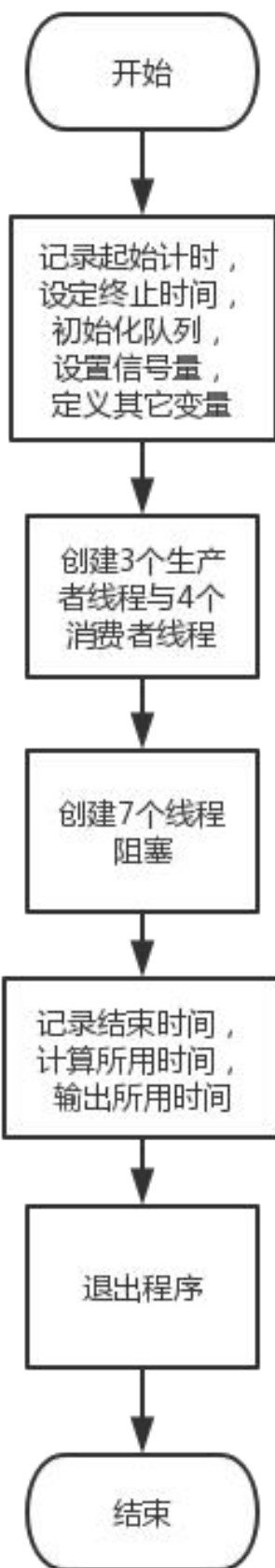
在数据结构方面,

```
typedef struct
{
    char q;
}buffer_item;
typedef struct //队列的基本结构
{
    buffer_item buffer[MaxSize]; //存放队列元素
    int front; //头指针
    int rear; //尾指针(尾元素的下一个位置)
}Queue;
```

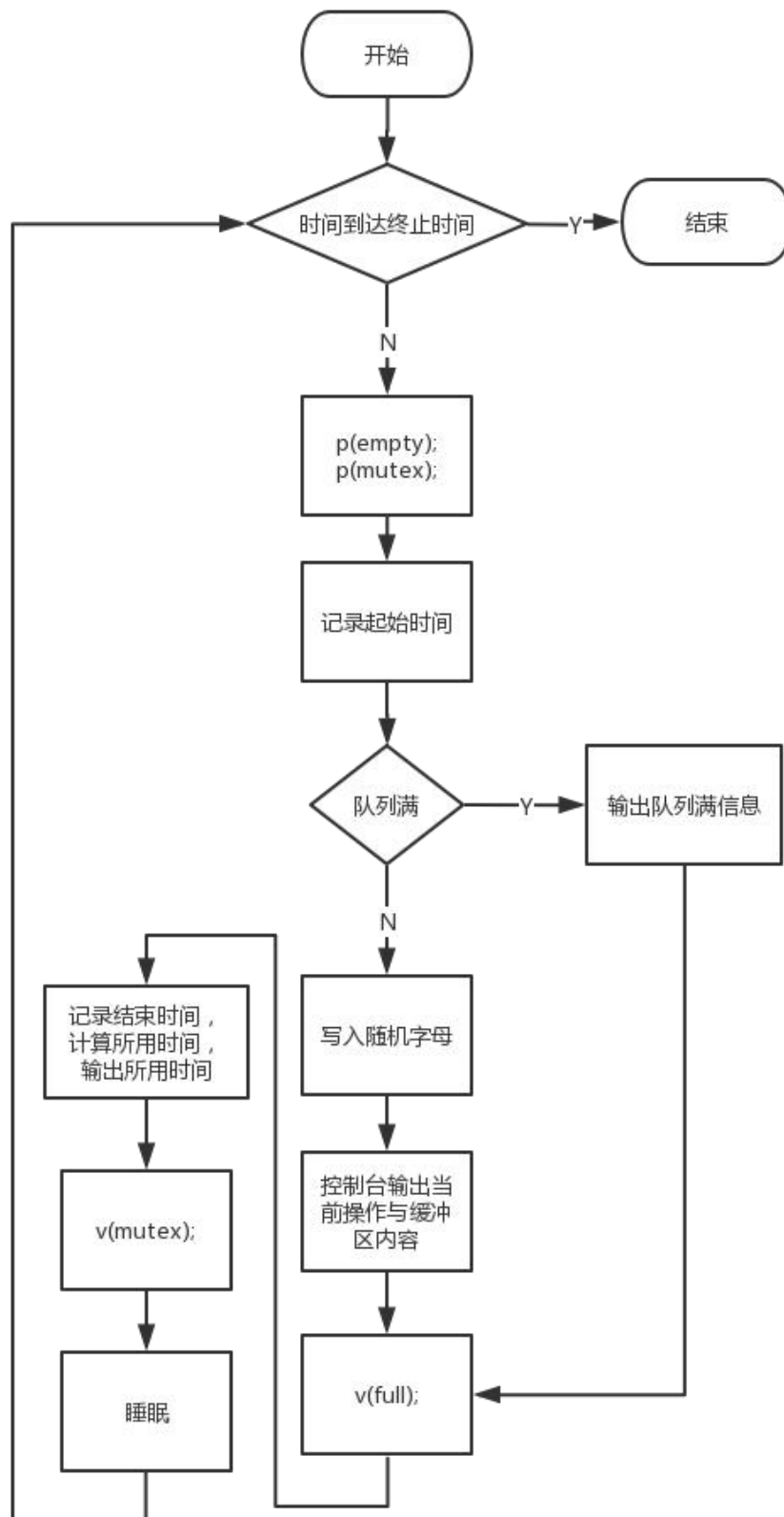
通过 typedef 定义 buffer_item 型变量, 之后利用 buffer_item 变量定义数组, 加入队列 Queue。

程序流程图如下:

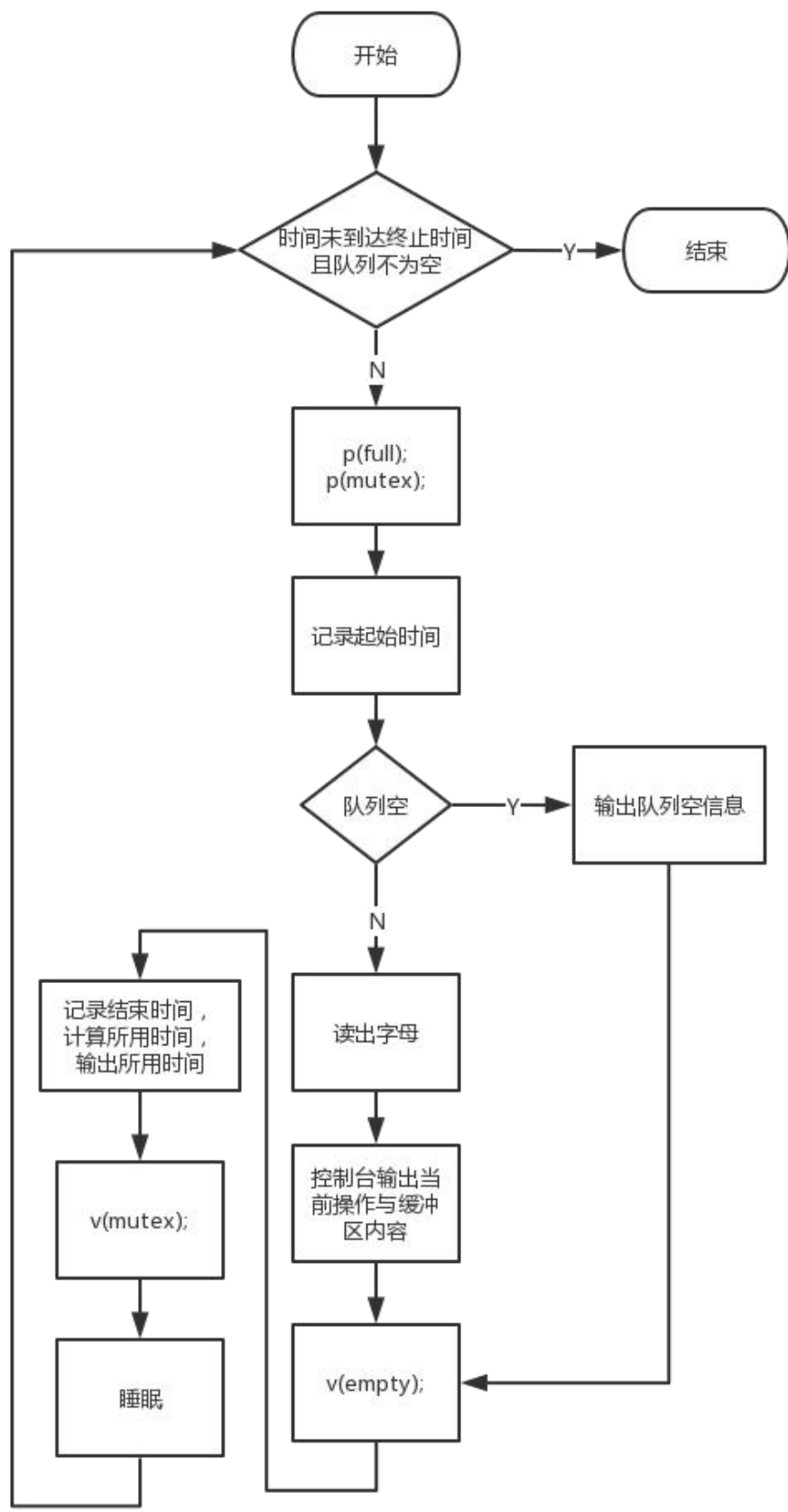
mian 函数流程图:



producer 函数:



consumer 函数:



Windows 下的图形化界面实现

在 Windows 图形化实现方面，首先选择使用 Windows API 进行实现。在 Windows 系统下的代码基础上，尝试使用 Windows API 进行图形化界面绘制，在大量阅读 CSDN 上教程，观看哔哩哔哩网站上的教学视频后进行到创建窗口与输出图形化组件后，在消息传递获取与处理时遇到问题。在搜索解决办法时，搜索到 c 语言绘制图形化界面设计的 easyx 插件，在下载插件后，利用 easyx 插件在代码基础上成功绘制图形化窗口。

图形化界面的基础思想是：在每个生产者或消费线程后读或写过程完成后，刷新窗口并重新向窗口写入当前缓冲区内容与所使用的时间。在生产者消费者进程全部结束后，在窗口中绘制一个按钮，若不点击按钮则长久不留在该界面，展现空的缓冲区与总的运行时间。点击按钮后，则退出图形化界面，显示控制台界面，可以详细查看每步的运行结果。

窗口文字输出部分代码如下：

```
cleardevice();//清屏（取决于背景色）
//窗口输出
char s[] = "当前缓冲区内容: ";
char s2[] = "当前时间: ";
char s3[] = "当前操作者: ";
char s4[] = "当前操作字母: ";
outtextxy(10, 0, s3); //字符串输出
outtextxy(10, 20, s4); //字符串输出
outtextxy(120, 20, buff.p); //字符串输出
outtextxy(120, 0, 'P'); //字符串输出
outtextxy(10, 40, s); //字符串输出
outtextxy(200, 80, s2); //字符串输出
outtextxy(10, 60, print[0]); //字符串输出
outtextxy(30, 60, print[1]); //字符串输出
outtextxy(50, 60, print[2]); //字符串输出
outtextxy(70, 60, print[3]); //字符串输出
outtextxy(90, 60, print[4]); //字符串输出
outtextxy(200, 100, t); //字符串输出
```

按钮输出部分代码如下：

```
int r3[]={300,150,350,190}; //退出按钮的矩形参数：左上x，左上y，右下x，右下y
RECT R3={r3[0],r3[1],r3[2],r3[3]}; //按钮2的矩形区域
LOGFONT f; //字体样式指针
gettextstyle(&f); //获取字体样式
_tcscpy(f.lfFaceName, T("宋体")); //设置字体为宋体
f.lfQuality = ANTIALIASED_QUALITY; //设置输出效果为抗锯齿
settextstyle(&f); //设置字体样式
settextcolor(BLACK); //BLACK在graphic.h头文件里面被定义为黑色的颜色常量
```

drawtext("退出",&R3,DT_CENTER | DT_VCENTER | DT_SINGLELINE);// 在矩形区域R3 内输入文字, 水平居中, 垂直居中, 单行显示

setlinecolor(BLACK);

rectangle(r3[0],r3[1],r3[2],r3[3]); //画矩形

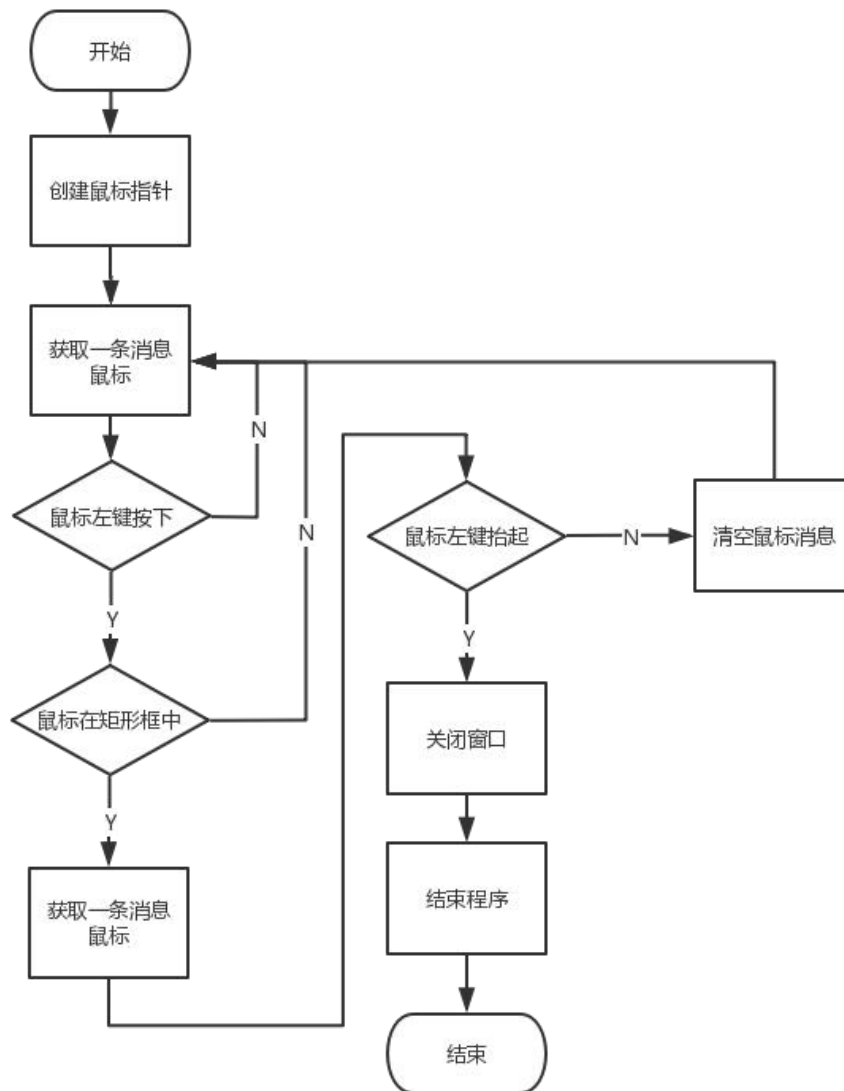
鼠标触发“退出按钮”的代码如下:

MOUSEMSG m;// 鼠标指针

while(1)

```
{
    m = GetMessage();//获取一条鼠标消息
    if(m.uMsg==WM_LBUTTONDOWN)//鼠标左键按下
    {
        if(m.x>r3[0] && m.x<r3[2] && m.y>r3[1] && m.y<r3[3])
        {
            m = GetMessage();//获取一条鼠标消息
            if(m.uMsg==WM_LBUTTONUP)//鼠标左键抬起
            {
                closegraph();
                exit(0);
            }
        }
        else
        {
            FlushMouseMsgBuffer();//单击事件后清空鼠标消息
            //printf("\r\n(%d,%d)",m.x,m.y);//打印鼠标坐标, 方便调试时确定区域
        }
    }
}
```

其过程为: 创建鼠标指针, 获取鼠标信息, 判断信息是否为左键按下, 若为则判断是否在“退出”框中, 若为则再次获取鼠标信息, 判断是否为鼠标左键抬起, 若为则关闭该窗口, 结束程序。其流程图如下:

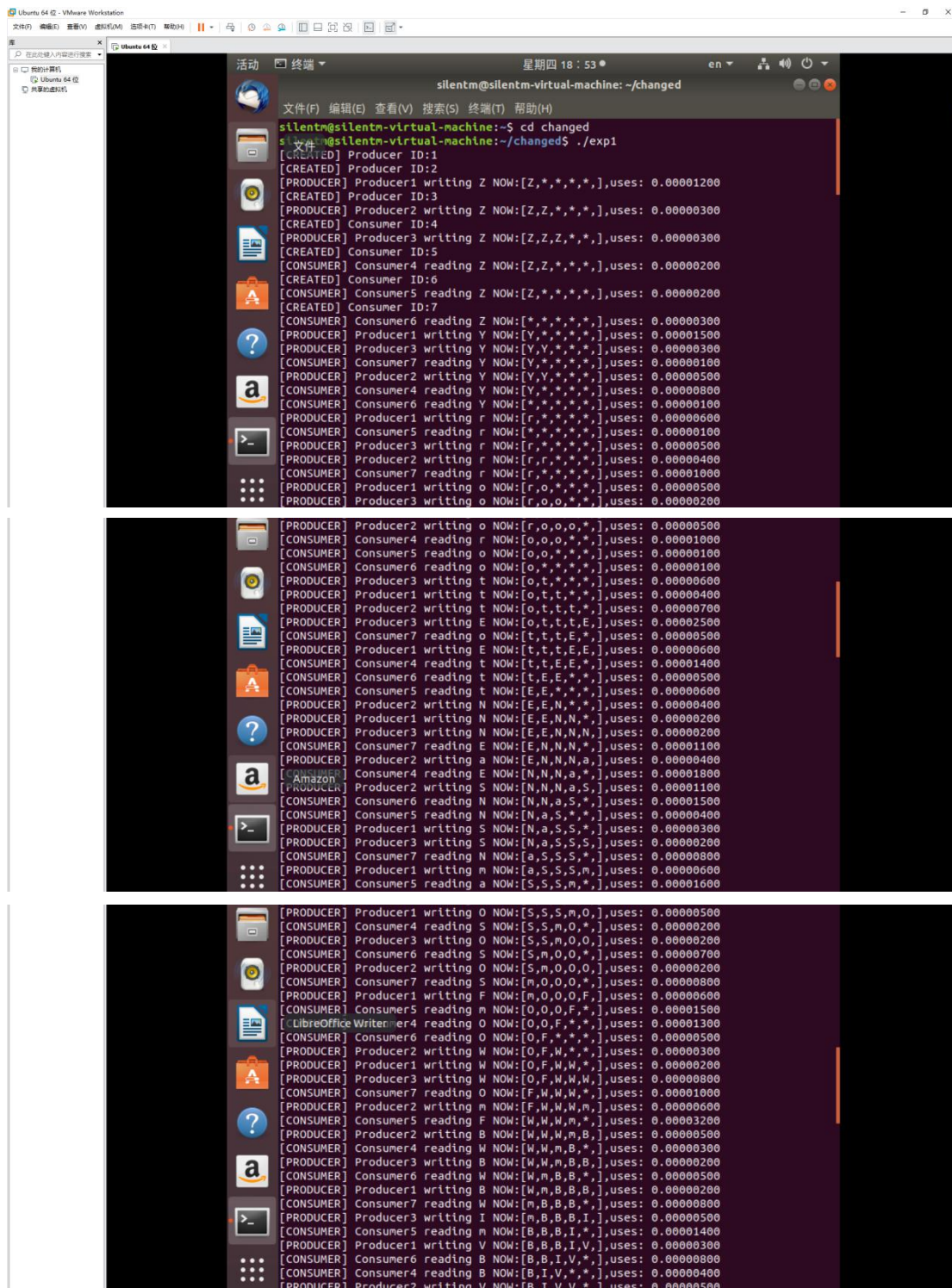


数独解决方案验证器辅助

协助完成数独解决方案验证器代码，完善代码逻辑。

4. 程序运行结果展示

1) Linux 系统下生产者消费者



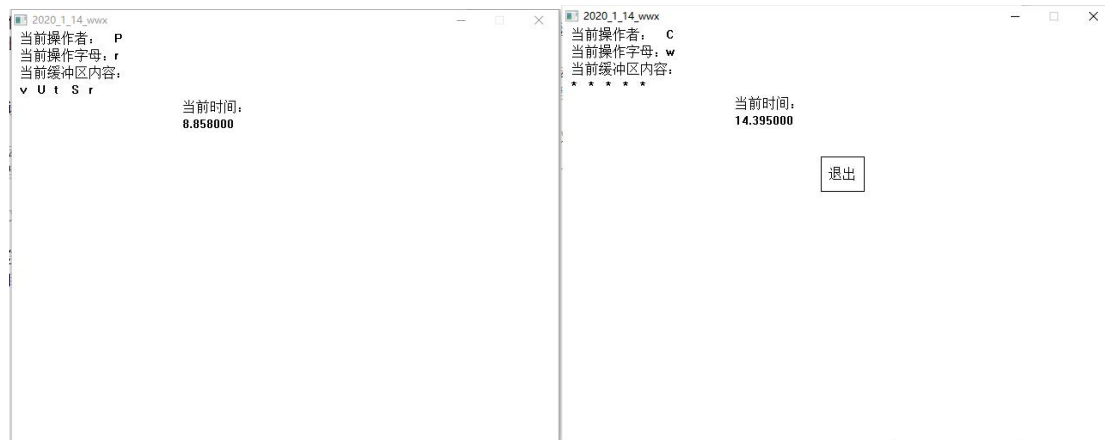
```
silentm@silentm-virtual-machine: ~/changed
$ cd changed
$ ./exp1
[CREATED] Producer ID:1
[CREATED] Producer ID:2
[PRODUCER] Producer1 writing Z NOW:[Z,*,*,*],uses: 0.00001200
[CREATED] Producer ID:3
[PRODUCER] Producer2 writing Z NOW:[Z,Z,*,*],uses: 0.00000300
[CREATED] Consumer ID:4
[PRODUCER] Producer3 writing Z NOW:[Z,Z,Z,*,*],uses: 0.00000300
[CREATED] Consumer ID:5
[CONSUMER] Consumer4 reading Z NOW:[Z,Z,*,*],uses: 0.00000200
[CREATED] Consumer ID:6
[CONSUMER] Consumer5 reading Z NOW:[Z,*,*,*],uses: 0.00000200
[CREATED] Consumer ID:7
[CONSUMER] Consumer6 reading Z NOW:[*,*,*,*],uses: 0.00000300
[PRODUCER] Producer1 writing Y NOW:[Y,*,*,*],uses: 0.00001500
[PRODUCER] Producer3 writing Y NOW:[Y,Y,*,*],uses: 0.00000300
[CONSUMER] Consumer3 reading Y NOW:[Y,Y,*,*],uses: 0.00000100
[PRODUCER] Producer7 writing Y NOW:[Y,*,*,*],uses: 0.00000500
[CONSUMER] Consumer2 reading Y NOW:[Y,*,*,*],uses: 0.00000800
[CONSUMER] Consumer6 reading Y NOW:[*,*,*,*],uses: 0.00000100
[PRODUCER] Producer6 reading Y NOW:[*,*,*,*],uses: 0.00000600
[CONSUMER] Consumer1 writing r NOW:[r,*,*,*],uses: 0.00000100
[PRODUCER] Producer5 reading r NOW:[*,*,*,*],uses: 0.00000500
[PRODUCER] Producer3 writing r NOW:[r,*,*,*],uses: 0.00000400
[CONSUMER] Producer2 writing r NOW:[r,r,*,*],uses: 0.00001000
[PRODUCER] Producer7 reading r NOW:[r,*,*,*],uses: 0.00000500
[CONSUMER] Producer1 writing o NOW:[r,o,*,*],uses: 0.00000200
[PRODUCER] Producer3 writing o NOW:[r,o,o,*,*],uses: 0.00000500
[CONSUMER] Consumer4 reading r NOW:[o,o,o,*,*],uses: 0.00001800
[CONSUMER] Consumer5 reading o NOW:[o,o,*,*],uses: 0.00000100
[CONSUMER] Consumer6 reading o NOW:[o,*,*,*],uses: 0.00000100
[PRODUCER] Producer3 writing t NOW:[o,t,*,*],uses: 0.00000600
[PRODUCER] Producer1 writing t NOW:[o,t,t,*,*],uses: 0.00000400
[PRODUCER] Producer2 writing t NOW:[o,t,t,t,*,*],uses: 0.00000700
[PRODUCER] Producer3 writing E NOW:[o,t,t,t,E],uses: 0.00002500
[CONSUMER] Consumer7 reading o NOW:[t,t,t,E,*],uses: 0.00000500
[PRODUCER] Producer1 writing E NOW:[t,t,t,E,E],uses: 0.00000600
[CONSUMER] Consumer4 reading t NOW:[t,t,E,*],uses: 0.00001400
[CONSUMER] Consumer6 reading t NOW:[t,E,*,*],uses: 0.00000500
[CONSUMER] Consumer5 reading t NOW:[E,*,*,*],uses: 0.00000600
[PRODUCER] Producer2 writing N NOW:[E,E,N,*,*],uses: 0.00000400
[PRODUCER] Producer1 writing N NOW:[E,E,N,N,*,*],uses: 0.00000200
[PRODUCER] Producer3 writing N NOW:[E,E,N,N,N],uses: 0.00000200
[CONSUMER] Consumer7 reading E NOW:[E,N,N,N,*,*],uses: 0.00001100
[PRODUCER] Producer2 writing a NOW:[E,N,N,N,a],uses: 0.00000400
[CONSUMER] Consumer4 reading E NOW:[N,N,N,a,*],uses: 0.00001800
[PRODUCER] Producer2 writing S NOW:[N,N,N,a,S],uses: 0.00001100
[CONSUMER] Consumer6 reading N NOW:[N,N,a,*,*],uses: 0.00001500
[CONSUMER] Consumer5 reading N NOW:[N,a,*,*,*],uses: 0.00000400
[PRODUCER] Producer1 writing S NOW:[N,a,S,*,*],uses: 0.00000300
[PRODUCER] Producer3 writing S NOW:[N,a,S,S,*,*],uses: 0.00000200
[CONSUMER] Consumer7 reading N NOW:[a,S,S,*,*],uses: 0.00000800
[PRODUCER] Producer1 writing m NOW:[a,S,S,S,m],uses: 0.00000600
[CONSUMER] Consumer5 reading a NOW:[S,S,S,m,*],uses: 0.00001600
[PRODUCER] Producer1 writing O NOW:[S,S,S,m,O],uses: 0.00000500
[CONSUMER] Consumer4 reading S NOW:[S,S,m,O,*],uses: 0.00000200
[PRODUCER] Producer3 writing O NOW:[S,S,m,O,O],uses: 0.00000200
[CONSUMER] Consumer6 reading S NOW:[S,m,O,*,*],uses: 0.00000700
[PRODUCER] Producer2 writing O NOW:[S,m,O,O,O],uses: 0.00000200
[CONSUMER] Consumer7 reading S NOW:[m,O,O,*,*],uses: 0.00000800
[PRODUCER] Producer1 writing F NOW:[m,O,O,O,F],uses: 0.00000600
[CONSUMER] Consumer5 reading m NOW:[O,O,O,F,*],uses: 0.00001500
[PRODUCER] Producer4 reading O NOW:[O,O,F,*,*],uses: 0.00001300
[CONSUMER] Consumer6 reading O NOW:[O,F,*,*,*],uses: 0.00000500
[PRODUCER] Producer2 writing W NOW:[O,F,W,*,*],uses: 0.00000300
[PRODUCER] Producer1 writing W NOW:[O,F,W,W,*,*],uses: 0.00000200
[PRODUCER] Producer3 writing W NOW:[O,F,W,W,W],uses: 0.00000800
[CONSUMER] Consumer7 reading O NOW:[F,W,W,*,*],uses: 0.00001000
[PRODUCER] Producer2 writing m NOW:[F,W,W,W,m],uses: 0.00000600
[CONSUMER] Consumer5 reading F NOW:[W,W,W,*,*],uses: 0.00003200
[PRODUCER] Producer2 writing B NOW:[W,W,W,m,B],uses: 0.00000500
[CONSUMER] Consumer4 reading W NOW:[W,m,B,*,*],uses: 0.00000300
[PRODUCER] Producer3 writing B NOW:[W,m,B,B,*,*],uses: 0.00000200
[CONSUMER] Consumer6 reading W NOW:[W,m,B,B,*,*],uses: 0.00000500
[PRODUCER] Producer1 writing B NOW:[W,m,B,B,B],uses: 0.00000200
[CONSUMER] Consumer7 reading W NOW:[m,B,B,*,*],uses: 0.00000800
[PRODUCER] Producer3 writing I NOW:[m,B,B,B,I],uses: 0.00000500
[CONSUMER] Consumer5 reading m NOW:[B,B,B,I,*],uses: 0.00001400
[PRODUCER] Producer1 writing V NOW:[B,B,B,I,V],uses: 0.00000300
[CONSUMER] Consumer6 reading B NOW:[B,B,I,V,*,*],uses: 0.00000800
[CONSUMER] Consumer4 reading B NOW:[B,I,V,*,*,*],uses: 0.00000400
[PRODUCER] Producer2 writing V NOW:[B,I,V,V,*,*],uses: 0.00000500
```

```
[PRODUCER] Producer3 writing V NOW:[B,I,V,V,V,V],uses: 0.00000300
[CONSUMER] Consumer7 reading B NOW:[I,V,V,V,V,*],uses: 0.00001200
[PRODUCER] Producer1 writing g NOW:[I,V,V,V,g],uses: 0.00000500
[CONSUMER] Consumer5 reading I NOW:[V,V,V,g,*],uses: 0.00002100
[PRODUCER] Producer3 writing D NOW:[V,V,V,g,D],uses: 0.00000400
[CONSUMER] Consumer4 reading V NOW:[V,V,g,D,*],uses: 0.00000600
[CONSUMER] Consumer6 reading V NOW:[V,g,D,*,*],uses: 0.00000200
[PRODUCER] Producer2 writing D NOW:[V,g,D,D,*],uses: 0.00000300
[PRODUCER] Producer1 writing D NOW:[V,g,D,D,D],uses: 0.00000500
[CONSUMER] Consumer7 reading V NOW:[g,D,D,D,*],uses: 0.00001300
[PRODUCER] Producer3 writing r NOW:[g,D,D,D,r],uses: 0.00000700
[CONSUMER] Consumer5 reading g NOW:[D,D,D,r,*],uses: 0.00001600
[CONSUMER] Consumer4 reading D NOW:[D,D,r,*,*],uses: 0.00001000
[PRODUCER] Producer1 writing n NOW:[D,D,r,n,*],uses: 0.00000300
[PRODUCER] Producer2 writing n NOW:[D,D,r,n,n],uses: 0.00000200
[CONSUMER] Consumer6 reading D NOW:[D,r,n,n,*],uses: 0.00000100
[PRODUCER] Producer3 writing n NOW:[D,r,n,n,n],uses: 0.00000400
[CONSUMER] Consumer7 reading D NOW:[r,n,n,n,*],uses: 0.00001200
[PRODUCER] Producer2 writing l NOW:[r,n,n,n,l],uses: 0.00000600
[CONSUMER] Consumer4 reading r NOW:[n,n,n,l,*],uses: 0.00001500
[PRODUCER] Producer3 writing D NOW:[n,n,n,l,D],uses: 0.00000400
[CONSUMER] Consumer5 reading n NOW:[n,n,l,D,*],uses: 0.00000200
[PRODUCER] Producer1 writing D NOW:[n,n,l,D,D],uses: 0.00000200
[CONSUMER] Consumer6 reading n NOW:[n,l,D,D,*],uses: 0.00000600
[PRODUCER] Producer2 writing D NOW:[n,l,D,D,D],uses: 0.00000200
[CONSUMER] Consumer7 reading n NOW:[l,D,D,D,*],uses: 0.00001100
[PRODUCER] Producer1 writing M NOW:[l,D,D,D,M],uses: 0.00000300
```

```
[CONSUMER] Consumer4 reading l NOW:[D,D,D,M,*],uses: 0.00001300
[PRODUCER] Producer2 writing r NOW:[D,D,D,M,r],uses: 0.00000300
[CONSUMER] Consumer5 reading D NOW:[D,D,M,r,*],uses: 0.00000200
[PRODUCER] Producer3 writing r NOW:[D,D,M,r,r],uses: 0.00000300
[CONSUMER] Consumer6 reading D NOW:[D,M,r,r,r],uses: 0.00000100
[PRODUCER] Producer1 writing r NOW:[D,M,r,r,r,r],uses: 0.00000800
[CONSUMER] Consumer7 reading D NOW:[M,r,r,r,r,*],uses: 0.00014400
[PRODUCER] Producer3 writing s NOW:[M,r,r,r,r,s],uses: 0.00000900
[CONSUMER] Consumer4 reading M NOW:[r,r,r,r,s,*],uses: 0.00001500
[CONSUMER] Consumer5 reading r NOW:[r,r,r,s,*],uses: 0.00001500
[PRODUCER] Producer1 writing U NOW:[r,r,r,s,U,*],uses: 0.00000400
[PRODUCER] Producer2 writing U NOW:[r,r,r,s,U,U],uses: 0.00000300
[CONSUMER] Consumer6 reading r NOW:[r,s,U,U,*],uses: 0.00000100
[PRODUCER] Producer3 writing U NOW:[r,s,U,U,U],uses: 0.00000400
[CONSUMER] Consumer7 reading r NOW:[s,U,U,U,*],uses: 0.00005300
[PRODUCER] Producer2 writing g NOW:[s,U,U,U,g],uses: 0.00000400
[CONSUMER] Consumer4 reading s NOW:[U,U,U,g,*],uses: 0.00001700
[CONSUMER] Consumer5 reading U NOW:[U,U,g,*,*],uses: 0.00000300
[PRODUCER] Producer3 writing r NOW:[U,U,g,r,*],uses: 0.00000300
[CONSUMER] Consumer6 reading U NOW:[U,g,r,*,*],uses: 0.00000500
[PRODUCER] Producer1 writing r NOW:[U,g,r,r,r],uses: 0.00000200
[PRODUCER] Producer2 writing r NOW:[U,g,r,r,r,r],uses: 0.00000500
[CONSUMER] Consumer7 reading U NOW:[g,r,r,r,r,*],uses: 0.00001300
[PRODUCER] Producer3 writing l NOW:[g,r,r,r,l],uses: 0.00000300
[CONSUMER] Consumer4 reading g NOW:[r,r,r,l,*],uses: 0.00001300
[PRODUCER] Producer1 writing T NOW:[r,r,r,l,T],uses: 0.00000300
[CONSUMER] Consumer5 reading r NOW:[r,r,l,T,*],uses: 0.00004300
```

```
[PRODUCER] Producer2 writing T NOW:[r,r,l,T,T],uses: 0.00000300
[CONSUMER] Consumer6 reading r NOW:[r,l,T,T,*],uses: 0.00000100
[CONSUMER] Consumer7 reading r NOW:[l,T,T,*,*],uses: 0.00001000
[CONSUMER] Consumer4 reading l NOW:[T,T,*,*,*],uses: 0.00001100
[CONSUMER] Consumer6 reading T NOW:[T,*,*,*,*],uses: 0.00000400
[CONSUMER] Consumer5 reading T NOW:[*,*,*,*,*],uses: 0.00000400
All time used is: 0.00590500
silentn@silentn-virtual-machine:~/changed$
```


2) Windows 系统下生产者消费者



```
"D:\Program Files (x86)\Microsoft Visual Studio\MyProjects\2020_1_14_vvwx\Debug\2020_1_14_vvwx.exe"  
[PRODUCER] Producer2 writing[o] 内容[seQCo], uses: 0.00200000  
[CONSUMER] Consumer6 reading[s] 内容[seQCo*], uses: 0.00100000  
[PRODUCER] Producer3 writing[a] 内容[seQCoa], uses: 0.00200000  
[CONSUMER] Consumer7 reading[e] 内容[seQCoa*], uses: 0.00100000  
[PRODUCER] Producer1 writing[M] 内容[seQCoaM], uses: 0.00300000  
[CONSUMER] Consumer5 reading[Q] 内容[seQCoaM*], uses: 0.00300000  
[PRODUCER] Producer2 writing[y] 内容[seQCoaMy], uses: 0.00000000  
[CONSUMER] Consumer4 reading[C] 内容[seQCoaMy*], uses: 0.00000000  
[PRODUCER] Producer3 writing[k] 内容[seQCoaMyk], uses: 0.00100000  
[CONSUMER] Consumer6 reading[o] 内容[seQCoaMyk*], uses: 0.00200000  
[PRODUCER] Producer1 writing[W] 内容[seQCoaMykW], uses: 0.00100000  
[CONSUMER] Consumer7 reading[a] 内容[seQCoaMykW*], uses: 0.00000000  
[PRODUCER] Producer2 writing[I] 内容[seQCoaMykWl], uses: 0.00300000  
[CONSUMER] Consumer5 reading[M] 内容[seQCoaMykWl*], uses: 0.00000000  
[PRODUCER] Producer3 writing[x] 内容[seQCoaMykWlx], uses: 0.00700000  
[CONSUMER] Consumer4 reading[y] 内容[seQCoaMykWlx*], uses: 0.00300000  
[PRODUCER] Producer1 writing[m] 内容[seQCoaMykWlxm], uses: 0.00200000  
[CONSUMER] Consumer6 reading[k] 内容[seQCoaMykWlxm*], uses: 0.00200000  
[PRODUCER] Producer2 writing[b] 内容[seQCoaMykWlxmb], uses: 0.00100000  
[CONSUMER] Consumer7 reading[W] 内容[seQCoaMykWlxmb*], uses: 0.00000000  
[PRODUCER] Producer3 writing[Q] 内容[seQCoaMykWlxmbQ], uses: 0.00100000  
[CONSUMER] Consumer5 reading[I] 内容[seQCoaMykWlxmbQ*], uses: 0.00300000  
[PRODUCER] Producer1 writing[F] 内容[seQCoaMykWlxmbQF], uses: 0.00200000  
[CONSUMER] Consumer4 reading[x] 内容[seQCoaMykWlxmbQF*], uses: 0.00200000  
[PRODUCER] Producer2 writing[u] 内容[seQCoaMykWlxmbQFu], uses: 0.00500000  
[CONSUMER] Consumer6 reading[m] 内容[seQCoaMykWlxmbQFu*], uses: 0.00400000  
[PRODUCER] Producer3 writing[j] 内容[seQCoaMykWlxmbQFuj], uses: 0.00100000  
[CONSUMER] Consumer7 reading[b] 内容[seQCoaMykWlxmbQFuj*], uses: 0.00400000  
[PRODUCER] Producer1 writing[Y] 内容[seQCoaMykWlxmbQFujY], uses: 0.00100000  
[CONSUMER] Consumer5 reading[Q] 内容[seQCoaMykWlxmbQFujY*], uses: 0.00100000  
[PRODUCER] Producer2 writing[N] 内容[seQCoaMykWlxmbQFujYN], uses: 0.00100000  
[CONSUMER] Consumer4 reading[F] 内容[seQCoaMykWlxmbQFujYN*], uses: 0.00300000  
[PRODUCER] Producer3 writing[C] 内容[seQCoaMykWlxmbQFujYNC], uses: 0.00100000  
[CONSUMER] Consumer6 reading[u] 内容[seQCoaMykWlxmbQFujYNC*], uses: 0.00000000  
[PRODUCER] Producer1 writing[r] 内容[seQCoaMykWlxmbQFujYNCr], uses: 0.00300000  
[CONSUMER] Consumer7 reading[j] 内容[seQCoaMykWlxmbQFujYNCr*], uses: 0.00100000  
[PRODUCER] Producer2 writing[g] 内容[seQCoaMykWlxmbQFujYNCrg], uses: 0.00200000  
[CONSUMER] Consumer5 reading[Y] 内容[seQCoaMykWlxmbQFujYNCrg*], uses: 0.00000000  
[PRODUCER] Producer3 writing[V] 内容[seQCoaMykWlxmbQFujYNCrgV], uses: 0.00100000  
[CONSUMER] Consumer4 reading[N] 内容[seQCoaMykWlxmbQFujYNCrgV*], uses: 0.00100000  
[PRODUCER] Producer1 writing[K] 内容[seQCoaMykWlxmbQFujYNCrgVK], uses: 0.00200000  
[CONSUMER] Consumer6 reading[C] 内容[seQCoaMykWlxmbQFujYNCrgVK*], uses: 0.00200000  
[PRODUCER] Producer2 writing[z] 内容[seQCoaMykWlxmbQFujYNCrgVKz], uses: 0.00100000  
[CONSUMER] Consumer7 reading[r] 内容[seQCoaMykWlxmbQFujYNCrgVKz*], uses: 0.00200000  
[PRODUCER] Producer3 writing[o] 内容[seQCoaMykWlxmbQFujYNCrgVKzo], uses: 0.00100000  
[CONSUMER] Consumer5 reading[g] 内容[seQCoaMykWlxmbQFujYNCrgVKzo*], uses: 0.00100000  
[PRODUCER] Producer1 writing[d] 内容[seQCoaMykWlxmbQFujYNCrgVKzod], uses: 0.00100000  
[CONSUMER] Consumer4 reading[V] 内容[seQCoaMykWlxmbQFujYNCrgVKzod*], uses: 0.00000000  
[PRODUCER] Producer2 writing[S] 内容[seQCoaMykWlxmbQFujYNCrgVKzodS], uses: 0.00200000  
[CONSUMER] Consumer6 reading[k] 内容[seQCoaMykWlxmbQFujYNCrgVKzodS*], uses: 0.00200000  
[PRODUCER] Producer3 writing[H] 内容[seQCoaMykWlxmbQFujYNCrgVKzodSH], uses: 0.00200000  
[CONSUMER] Consumer7 reading[z] 内容[seQCoaMykWlxmbQFujYNCrgVKzodSH*], uses: 0.00200000  
[PRODUCER] Producer1 writing[w] 内容[seQCoaMykWlxmbQFujYNCrgVKzodSHw], uses: 0.00200000  
[CONSUMER] Consumer5 reading[o] 内容[seQCoaMykWlxmbQFujYNCrgVKzodSHw*], uses: 0.00200000  
[CONSUMER] Consumer4 reading[d] 内容[seQCoaMykWlxmbQFujYNCrgVKzodSHw**], uses: 0.00400000  
[CONSUMER] Consumer6 reading[S] 内容[seQCoaMykWlxmbQFujYNCrgVKzodSHw***], uses: 0.00200000  
[CONSUMER] Consumer7 reading[H] 内容[seQCoaMykWlxmbQFujYNCrgVKzodSHw****], uses: 0.00200000  
[CONSUMER] Consumer5 reading[w] 内容[seQCoaMykWlxmbQFujYNCrgVKzodSHw*****], uses: 0.00100000  
All time used is: 14.60100000  
Press any key to continue_
```


3) Linux 下数独解决方案验证器

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
xiao@xiao-virtual-machine:~/sudoku$ gcc -o su sudoku.c -pthread
xiao@xiao-virtual-machine:~/sudoku$ ./su
请输入文件名:
su.txt
数据导入完毕!
数据内容:
6 2 4 5 3 9 1 8 7
5 1 9 7 2 8 6 3 4
8 3 7 6 1 4 2 9 5
1 4 3 8 6 5 7 2 9
9 5 8 2 4 7 3 6 1
7 6 2 3 9 1 4 5 8
3 7 1 9 5 6 8 4 2
4 9 6 1 8 2 5 7 3
2 8 5 4 7 3 9 1 6
列满足
行满足
九宫格满足
数独成立
xiao@xiao-virtual-machine:~/sudoku$ s
```

修改最后一个数字

```
xiao@xiao-virtual-machine:~/sudoku$ ./su
请输入文件名:
su.txt
数据导入完毕!
数据内容:
6 2 4 5 3 9 1 8 7
5 1 9 7 2 8 6 3 4
8 3 7 6 1 4 2 9 5
1 4 3 8 6 5 7 2 9
9 5 8 2 4 7 3 6 1
7 6 2 3 9 1 4 5 8
3 7 1 9 5 6 8 4 2
4 9 6 1 8 2 5 7 3
2 8 5 4 7 3 9 1 7
9 列不满足
9 行不满足
九宫格9不满足
```

4) Windows 下数独解决方案验证器

C:\Users\小鱼儿\Desktop\操作系统课程实践\sudo_win.exe

待验证数组内容:

1	4	5	3	2	7	6	9	8
8	3	9	6	5	4	1	2	7
6	7	2	9	1	8	5	4	3
4	9	6	1	8	5	3	7	2
2	1	8	4	7	3	9	5	6
7	5	3	2	9	6	4	8	1
3	6	7	5	4	2	8	1	9
9	8	4	7	6	1	2	3	5
5	2	1	8	3	9	7	6	4

有效数独!

Process returned 0 (0x0) execution time : 0.590 s
Press any key to continue.

C:\Users\小鱼儿\Desktop\操作系统课程实践\sudo_win.exe

待验证数组内容:

1	4	5	3	2	7	6	9	8
8	3	9	6	5	4	1	2	7
6	7	2	9	1	8	5	4	3
4	9	6	1	8	5	3	7	2
2	1	8	4	7	3	9	5	6
7	5	3	2	9	6	4	8	1
3	6	7	5	4	2	8	1	9
9	8	4	7	6	1	2	3	5
5	2	1	8	3	9	7	6	5

第8行8列元素5重复

```
C:\Users\小鱼儿\Desktop\操作系统课程实践\sudo_win.exe
待验证数组内容:
1      4      5      3      2      7      6      9      8
8      3      9      6      5      4      1      2      7
6      7      2      9      1      8      5      4      3
4      9      6      1      8      5      3      7      2
2      1      8      4      7      3      9      5      6
7      5      3      2      9      6      4      8      1
3      6      7      5      4      2      8      1      9
9      8      4      7      6      1      2      3      20
5      2      1      8      3      9      7      6      4

第7行8列元素20越界
```

5) 多线程排序（6 线程）

```
C:\Users\汉斯\Desktop\操作系统课设\上交\sorting6.exe
排序方式: 1, 2线程冒泡排序; 3, 4线程选择排序; 5, 6线程插入排序
插入耗时: 0.14000000
插入耗时: 0.14600000
选择耗时: 0.21600000
选择耗时: 0.21700000
冒泡耗时: 0.30600000
冒泡耗时: 0.30800000
是否要显示数组, 0: 否, 1: 是
```


5.重点、难点分析与课程设计体会

重点难点分析

在实验过程中遇到许多问题，在修改的过程中也出现许多 bug 并一一验证解决，在此简要说明。

`pthread_create` 使用时对于参数传递的问题。由于在 `pthread_create` 创建线程时对于处理函数的表示是指针变量的赋值，而非我们一般习惯于的函数调用，所以无法直接向服务函数传参。在查询资料后，学习到可以通过 `pthread_create` 函数的第四个函数进行传参，并且其类型为句柄，所以需要进行强制的类型转换。在服务函数中进行参数传递时，也需要再进行一次类型转换，使得传入的句柄参数可以被操作。

在使用 `srand` 函数与 `rand` 函数进行重新设置种子与取随机数，并将随机数作为睡眠时长控制随机时长睡眠时，出现问题。在多次调整睡眠时长为固定时长或随机时长，睡眠发生在程序执行前或后发现：当睡眠时长为固定时长时，睡眠发生在执行前、睡眠发生在执行前与后都会发生程序无法停止的问题，在使用随机数作为睡眠时间时，会偶尔出现程序无法停止的问题。在向线程服务函数中加入 `pthread_exit` 退出线程解决了问题。

在 `pthread_create` 函数使用前，使用了 `printf` 输出创建线程情况，在服务函数内有对于当前操作的 `printf` 输出，由于线程执行速度的不确定性，导致线程创建情况的输出会截断当前操作的输出。对于这一问题，通过在进入线程服务函数时，首先睡眠一段时间来保证线程创建的输出结束，再进行线程内的操作。

在 Windows 图形化实现方面，首先选择使用 Windows API 进行实现。在 Windows 系统下的代码基础上，尝试使用 Windows API 进行图形化界面绘制，在大量阅读 CSDN 上教程，观看哔哩哔哩网站上的教学视频后进行到创建窗口与输出图形化组件后，在消息传递获取与处理时遇到问题。在搜索解决办法时，搜索到 C 语言绘制图形化界面设计的 `easyx` 插件，在下载插件后，利用 `easyx` 插件在代码基础上成功绘制图形化窗口。若使用 Windows API 进行图形化界面绘制，则单窗口的创建与基础图形绘制就需要百余行代码，对于鼠标点击等消息的处理需要在服务函数中实现且较难实现。在 `easyx` 插件中，以上工作可以通过简单代码创建，使得 C 语言的图形化窗口绘制难度极大就降低。

课程设计体会

在本次课程实践中，我借助此次机会重新学习了对于线程的操作，以及代码实现方式，通过自学的方式更加深入地了解了操作系统的知识与线程有关的代码操作。

在此次课程实践中，自学了 C 语言通过 Windows API 绘制图形化界面的基本思想与 `easyx` 插件辅助下的图形化界面绘制，这将有助于我更深的了解 Java 等其它语言的图形化界面绘制。

通过团队成员间恰当的分工，增强团队合作能力，提高了工作效率，大大减

少了查错时间。

pthread 与 Windows API 区别

1.线程创建:

pthread: `pthread_create(&id1,NULL,(void *)productor,(void *)&Numflag1);`

API: `CreateThread(NULL,0,producer,(void *)&Numflag1,0, 0);`

Windows API 可以创建线程不立即执行

2. 线程编号类型

pthread 为 `pthread_t` 类型，Windows API 为 `HANDLE` 句柄格式

3.线程调用的函数 Windows API 必须为 `DWORD WINAPI` 类型

4. 阻塞线程

pthread: `pthread_join(id1,NULL);`

Windows API: `WaitForMultipleObjects(3,hdproducer,TRUE,INFINITE);`

Windows API 可以设定为只需单个对象发出信号，可设置的等待的时间

5.P 操作

pthread: `sem_wait(&full);`

Windows API: `WaitForSingleObject(full, INFINITE);`

Windows API 可改变最后一个参数来设置最大等待的时间可设置的等待的时间

6.V 操作

pthread: `sem_post(&mutex);`

Windows API: `ReleaseSemaphore(mutex,1,NULL);`

Windows API 下可设置 V 操作每次自加的数值