

实验报告：音乐厅门票订售系统

专业：智能科学与技术 年级：2014 级 姓名：冯立刚 学号：22920142203809

一、实验要求

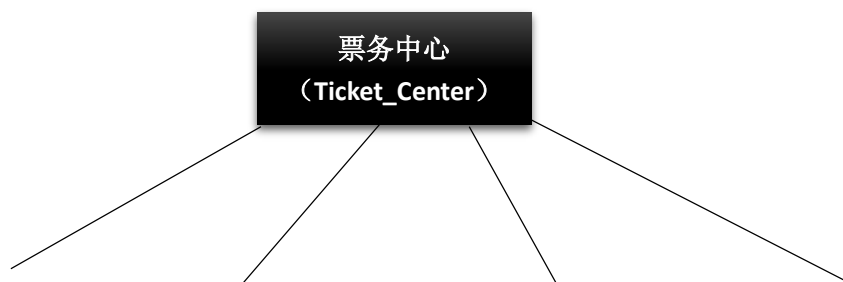
- 1、利用 IPC 机制仿真一个音乐厅门票订售系统；该系统包含若干售票代理，这些代理连接到一个票务中心计算机。
- 2、每个代理可以看成是一个可并发执行的进程，如果某个代理企图订/售一个已被占用的座位，系统应该禁止该操作并打印“座位已占用”的提示信息。如果某交易涉及对不存在的座位进行操作，则系统也应该打印出错信息。
- 3、需要注意的是：购票之前无需先预订，而取消操作只有到涉及的座位被预订但尚未售出时候才有效。
- 4、为使结果显示更明晰，系统在每个交易完成后都应该给出交易的信息摘要，而在系统结束后则应该显示出座位的订售情况（包含客户信息）。

二、实验环境

- 1、系统平台：Ubuntu 64 位, Windows10 64 位；
- 2、集成开发环境：Visual Studio Community 2017；

三、实验内容

- 1、程序框架：



售票代理 1
(Agent)

售票代理 2
(Agent)

售票代理 3
(Agent)

售票代理 4
(Agent)

a. 音乐厅座位状态用一个二维整型数组 `seat` 存储 (1: 可用, -1: 预定 0: 售出), 共享节中的数据变量 `seat` (座位状态), `flag`, `No` (代理商编号), `Lock` (互斥访问标记) 等会被多个进程共享, 其值若在一个进程中被改变, 则该操作对其它进程可见;

```
//-----共享节-----  
#pragma data_seg("Shared")  
  
volatile int seat[N][M] = {0}; //存放音乐厅座位状态  
volatile int flag = 1; //初始化标记  
volatile int No = 1; //售票终端编号  
volatile int Lock = 0; //0: 文件关闭, 1:文件打开  
  
#pragma data_seg  
  
#pragma comment(linker, "/Section:Shared,RWS")
```

b. 票务中心包含以下操作: `Set_ticket()`: 购买票; `Set_reserve()`: 预定票; `Set_cancel()`: 取消票; `Show_ticket()`: 显示用户购买的票。 `Show_reserve()`: 显示用户预定的票; `Get_state()`: 获得音乐厅座位的状态。

票务中心涉及到对座位状态 `seat`、顾客预定单 `reserve`、顾客购买单 `ticket` 的操作, 是执行售票任务的最终操作者。

基本操作为通过 `seat[][]=value` 改变座位状态, 然后从文件中载入顾客 (预定/购买) 订单, 并用新的用户操作改变订单状态, 然后将订单写入文件, 并返回操作结果。至此通过共享节、文件、锁机制实现了进程之间的数据共享。

```
seat[alphabet_To_number(in)][im] = 0;  
getticket();  
if (Iterator_ticket != ticket_record.end()) //在订单中找到了顾客姓名  
    ticket_record[customer].push_back(temp);  
else //没有在订单中找到顾客姓名
```

```

        ticket_vector.push_back(temp);
    ticket_record.insert(map<string, vector<string>>::value_type(customer,
    ticket_vector));
}
giveticket();

```

c. 售票代理 Agent 有一个成员变量 Ticket_center, 借此可以访问票务中心提供的接口函数, 进行完整的音乐厅票务代理操作。

```

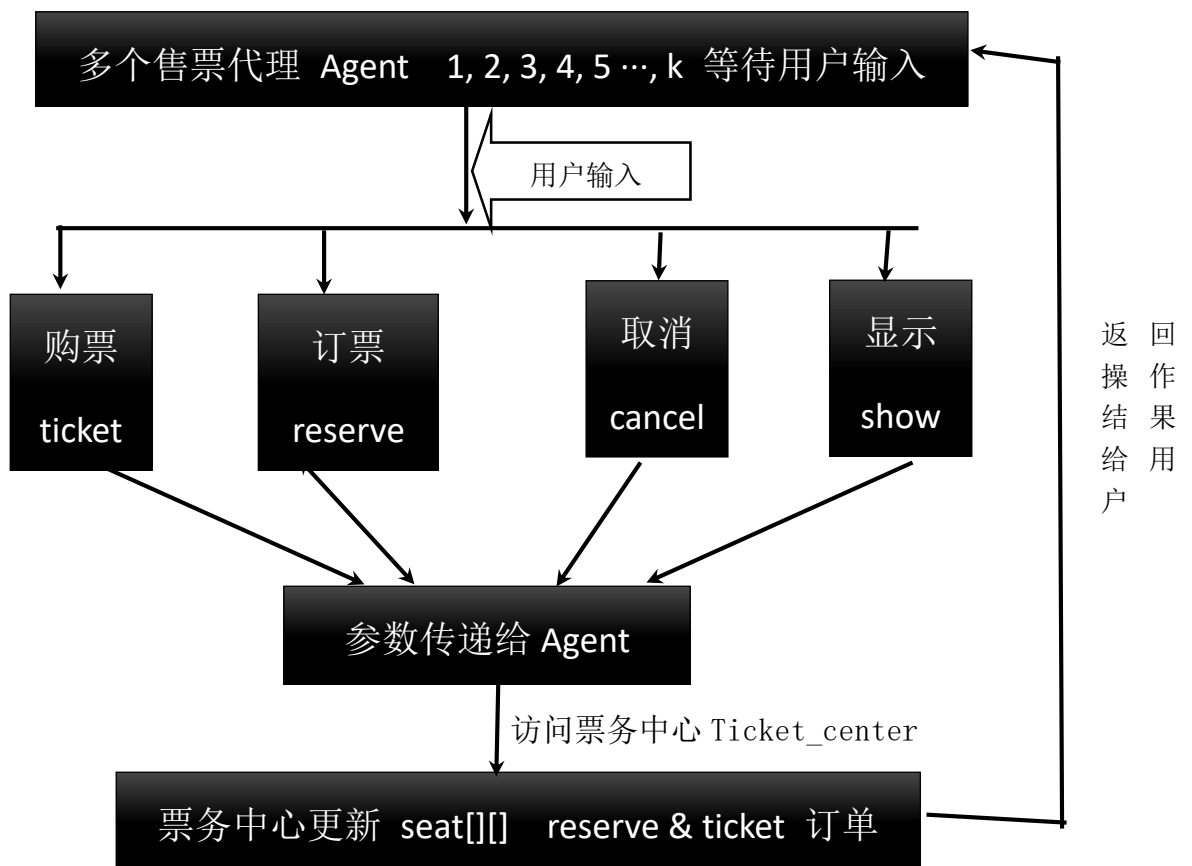
class Agent
{
private:
    Ticket_center T;
public:
    Agent(Ticket_center TT)
    {
        T = TT;
    }
    int reserve(int rows, int columns, string customer)
    {
        return T.set_reserve(rows, columns, customer);
    }
    int ticket(char rows, int columns, string customer)
    int cancel(int rows, int columns, string customer)
    void show_ticket(string customer)
    void show_reserve(string customer)
    int get_state(char rows, int columns)
}

```

2、程序运行流程

程序启动





四、项目测试








编写测试脚本启动 K 个售票代理终端，在脚本中我们企图启动十几个终端进程，但是由于程序限定最多只能有 K 个售票代理，所以该脚本最多启动 K 个进程。

```
@echo off
start musichall.exe
ping 127.1 -n 2 >nul
...
```

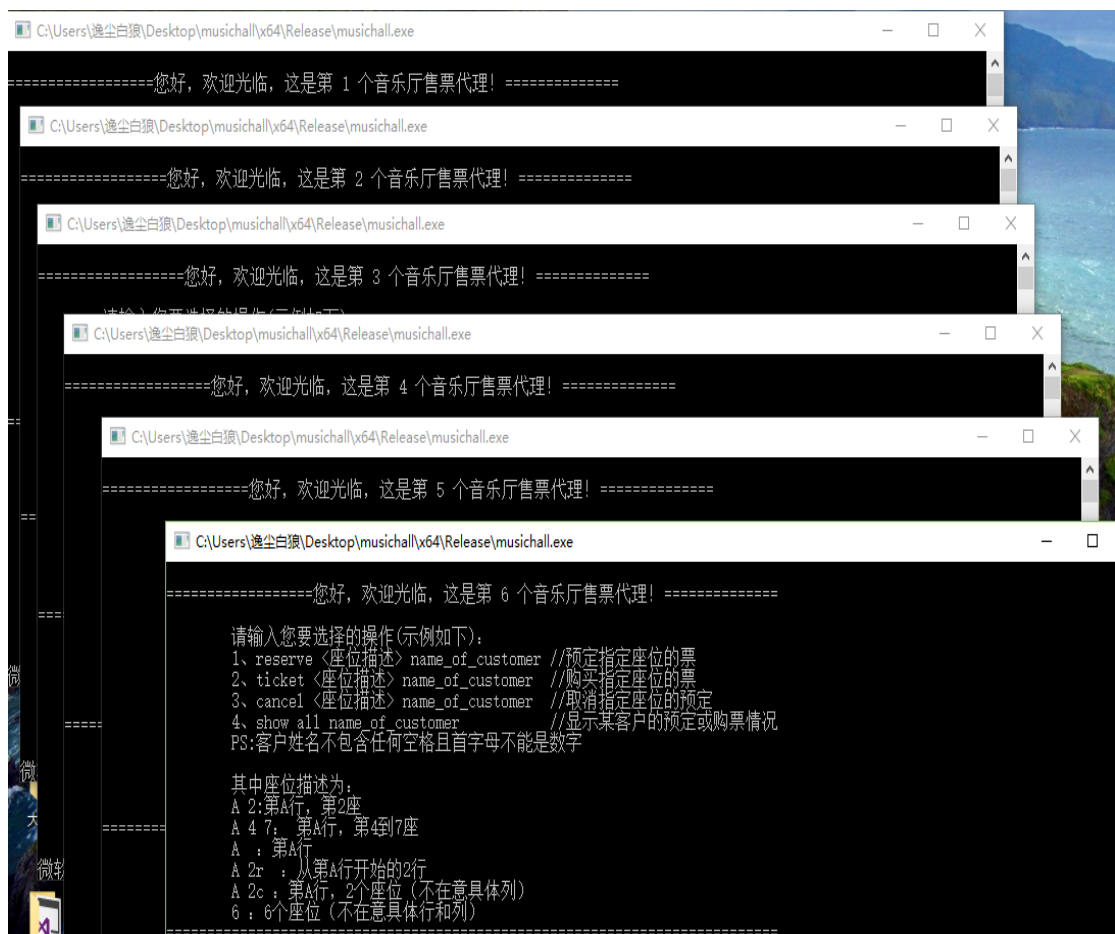
测试项目包括购票、订票、取消订票、显示顾客订单的测试，以及多个售票代理终端数据共享同步的测试。

1、程序正常启动，开启多个售票代理终端：

a. 程序包目录如下：

	musichall.exe	2017/5/2 20:26	应用程序	99 KB
	musichall.iobj	2017/5/2 20:06	IOBJ 文件	2,528 KB
	musichall.ipdb	2017/5/2 20:06	IPDB 文件	908 KB
	musichall.pdb	2017/5/2 20:26	Program Debug...	1,916 KB
	musichall_system.bat	2017/5/2 21:05	Windows 批处理...	1 KB
	reserve.txt	2017/5/2 21:05	文本文档	0 KB
	ticket.txt	2017/5/2 21:05	文本文档	0 KB

b. 执行测试脚本启动多个终端：



可以清晰的看到每个终端的提示信息中显示了售票代理号,共打开 6 个终端,这也证明了进程间实现了数据共享。

2、购票测试 (N=26, M=10, K=6)

测试用例：

ticket 4 gatsby;

```
ticket G gatsby;

ticket B 4 gatsby;

ticket C 4c gatsby;

ticket D 4r gatsby;

ticket F 4 8 gatsby;
```

ticket 4 gatsby:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Release\musichall.exe
4、show all name of customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位 (不在意具体列)
6 : 6个座位 (不在意具体行和列)
=====
ticket 4 gatsby
gatsby 已成功购票:
A0 P7 Q0 H4
请输入您要选择的操作(示例如下):
```

ticket G gatsby:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Release\musichall.exe
4、show all name of customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位 (不在意具体列)
6 : 6个座位 (不在意具体行和列)
=====
ticket G gatsby
gatsby 已成功购票:
A0 P7 Q0 H4 G0 G1 G2 G3 G4 G5 G6 G7 G8 G9
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
```

tikket B 4 gatsby:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Release\musichall.exe

4、show all name of customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位 (不在意具体列)
6 : 6个座位 (不在意具体行和列)
=====
ticket B 4 gatsby
gatsby 已成功购票:
A0 P7 Q0 H4 G0 G1 G2 G3 G4 G5 G6 G7 G8 G9 B4
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
```

ticket C 4c gatsby:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Release\musichall.exe

4、show all name of customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位 (不在意具体列)
6 : 6个座位 (不在意具体行和列)
=====
ticket C 4c gatsby
gatsby 已成功购票:
A0 P7 Q0 H4 G0 G1 G2 G3 G4 G5 G6 G7 G8 G9 B4 C0 C8 C2 C4
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
```

ticket D 4r gatsby:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Release\musichall.exe

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位 (不在意具体列)
6 : 6个座位 (不在意具体行和列)
=====
ticket D 4r gatsby
未成功购买座位 (被占有或被售出):
G0 G1 G2 G3 G4 G5 G6 G7 G8 G9
gatsby 已成功购票:
A0 P7 Q0 H4 G0 G1 G2 G3 G4 G5 G6 G7 G8 G9 B4 C0 C8 C2 C4 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 E0
E1 E2 E3 E4 E5 E6 E7 E8 E9 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
```

ticket F 4 8 gatsby:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Release\musichall.exe
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位 (不在意具体列)
6 : 6个座位 (不在意具体行和列)
=====
ticket F 4 8 gatsby
抱歉, F 排第 4 座到 8 座有座位被占有, 请重新选择!
gatsby 已成功购票:
A0 P7 Q0 H4 G0 G1 G2 G3 G4 G5 G6 G7 G8 G9 B4 C0 C8 C2 C4 D0 D1
E1 E2 E3 E4 E5 E6 E7 E8 E9 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
4、show_all //显示其客户的预定或购票情况
```

3、预定票和取消票测试

测试用例:

reserve 4 bob;

reserve B 4c bob;

cancel ** bob;

reserve 4 bob:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Release\musichall.exe
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位 (不在意具体列)
6 : 6个座位 (不在意具体行和列)
=====
reserve 4 bob
4
bob 已成功订票:
A0 P7 Q0 H4
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
```

reserve B 4c bob:

C:\Users\逸尘白狼\Desktop\musichall\x64\Release\musichall.exe

```
4、show all name of customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位 (不在意具体列)
6 : 6个座位 (不在意具体行和列)
=====
reserve B 4c bob
bob 已成功订票:
A0 P7 Q0 H4 B0 B1 B7 B4
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
```

cancel ** bob:

C:\Users\逸尘白狼\Desktop\musichall\x64\Debug\musichall.exe

```
4、show all name of customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位 (不在意具体列)
6 : 6个座位 (不在意具体行和列)
=====
cancel A 0 bob
bob成功取消座位:
A0
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
4、show all name of customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
```

4、显示顾客订单和代理数据共享同步测试:

测试用例:

reserve 5 joy;

ticket 5 joy;

show all joy;

1 号代理 reserve 5 joy:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Debug\musichall.exe

=====您好，欢迎光临，这是第 1 个音乐厅售票代理！=====

请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
4、show all name_of_customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位(不在意具体列)
6 : 6个座位(不在意具体行和列)

=====
reserve 5 joy
joy 已成功订票:
A0 M8 A4 L5 L7
=====
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
4、show all name_of_customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字
```

2 号代理 ticket 5 joy:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Debug\musichall.exe

=====您好，欢迎光临，这是第 2 个音乐厅售票代理！=====

请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
4、show all name_of_customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位(不在意具体列)
6 : 6个座位(不在意具体行和列)

=====
ticket 5 joy
joy 已成功购票:
D1 F2 R6 V4 C3
=====
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
4、show all name_of_customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字
```

3 号代理 show all joy:

```
C:\Users\逸尘白狼\Desktop\musichall\x64\Debug\musichall.exe

=====您好，欢迎光临，这是第 3 个音乐厅售票代理！=====

请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
2、ticket <座位描述> name_of_customer //购买指定座位的票
3、cancel <座位描述> name_of_customer //取消指定座位的预定
4、show all name_of_customer //显示某客户的预定或购票情况
PS:客户姓名不包含任何空格且首字母不能是数字

其中座位描述为:
A 2:第A行, 第2座
A 4 7: 第A行, 第4到7座
A : 第A行
A 2r : 从第A行开始的2行
A 2c : 第A行, 2个座位(不在意具体列)
6 : 6个座位(不在意具体行和列)

=====
show all joy
joy 已成功购票:
D1 F2 R6 V4 C3
joy 已成功订票:
A0 M8 A4 L5 L7
请输入您要选择的操作(示例如下):
1、reserve <座位描述> name_of_customer //预定指定座位的票
```

五、实验心得

之前我写的许多程序基本上都是单进程的，几个函数要共享变量就使用全局变量或其他方法，相对来说比较简单和容易理解。第一次写多进程程序，并且还要实现多个进程之间的数据共享，想想还是比较麻烦的。然而通过百度和其他工具我找到了实现共享的方法，有许多种，最终选择了最好实现的一种方法完成了多进程程序的编写。

对程序的编写也有了更近一步的理解，编写程序之前对项目需求的透彻理解可以辅助建立完整的框架，并对选择什么工具和方法提供更好的参考，这一点很重要。