（



“程序设计基础”

课程设计报告

**设计题目**  机房预约系统

**姓 名**  陈嘉乐

**学 号**  2021218152

**专 业**  计算机科学与技术

**班 级**  3班

**完成日期**  2022年1月10日

**目 录**

1. 需求和规格说明 ··············· 2
2. 设计 ···················· 3

1. 菜单设计 ·············· 3

2.主函数设计 ···············6

3.机位信息 ··········· 6

4.功能函数设计 ·············· 8

5.系统参数图 ············· 17

1. 用户手册 ·················· 19
2. 调试及测试 ················· 20
3. 运行实例 ·················· 21
4. 进一步改进 ·················· 23
5. 心得体会 ·················· 24

1. **需求和规格说明**

20台机器，编号1到20，从早八点到晚八点。两小时一个时间段，每次可预定一个时间段。功能要求:

(1）系统以菜单方式工作

(2）查询，根据输入时间，输出机位信息。

(3）机位预定，根据输入的时间查询是否有空机位，若有则预约，若无则提供最近的时间段，另:若用户在非空时间上机，则将用户信息列入等待列表。

(4）退出预定，根据输入的时间，机器号撤销该事件的预定!

(5）查询是否有等待信息，若有则提供最优解决方案（等待时间尽量短若无则显示提示信息。

**（二） 设计**

## 菜单设计

### ·菜单框架设计

### ·菜单绘制

### ·菜单选项

（1）.菜单框架设计

关于菜单框架的设计，我选择用若干个“-”和“|”来组成外围框架，简洁明了。对界面的操作都是最基本的操作。课设中涉及到的有:

system(“cls”); 清除当前控制窗口，方便下一个界面的绘制

system(“pause”); 界面需要按回车才会返回，方便用户查看界面内容。

（2）.菜单绘制

cout << "\t\t\t===========欢迎使用合肥工业大学机房预约系统============" <<endl;

cout << "\t\t \t\t ----------------------------------------------------------------------------" << endl;

cout << "\t\t \t\t | |" << endl;

cout << "\t\t \t\t | |" << endl;

cout << "\t\t \t\t | |" << endl;

cout << "\t\t \t\t | |" << endl;

cout << "\t\t \t\t | |" << endl;

cout << "\t\t \t\t | |" << endl;

cout << "\t\t \t\t | |" << endl;

cout << "\t\t \t\t | |" << endl;

cout << "\t\t \t\t ----------------------------------------------------------------------------" << endl;

这种简单的菜单绘制没什么好说的，每段前面加上\t是为了让菜单能在程序界面的中间部位，更加合理且贴切实际。如果在绘制时不确定最后结果是否是自己想要的，可以复制在另一个程序看一下效果并对其进行适当修改以期达到最好的外观效果。我在编写这段代码时就出现两边“|”过多导致“出头”的结果，并且“\t”我在第一次设计时只加了2个，运行时才发现用4个更合理。

（3）.菜单选项

对于刚刚进入系统的菜单选项我提供了5个选项，对应着5种不同功能，旨在能够更加全面便利，分别是：1.查询机位信息、2.机位预定、3.退出预定、4.查询等待信息、0.退出系统，希望能够向人性化设计贴近。

对于后面的功能分支中，若用户输入错误，我也提供了继续输入和返回菜单两个选项。

## 主函数设计

主函数主要使用了while（）和switch case语句在程序正式启动前先调用下info\_empty()函数（该函数在后面部分会介绍），清空机位信息。由用户输入，在switch case中设置接口并调用相应的函数，若用户输入错误会提示用户需要重新输入。主函数较为简洁，内容不多。

## 机位信息

利用struct结构体将需要的数据如用户姓名，性别，联系方式以及年月日等信息放入其中，由于2小时为一个时间段，早上8点到下午8点，共6个时间段，用0表示空位，1表示无空位。

struct PCinfo //机位信息，用户信息

{

int State[6]; //2小时为一个时间段，早上8点到下午8点，6个时间段，0表示空位，1表示无空位。

char name[20];

char sex[30];

char tel[20];

int year;

int month;

int day;

int start\_time;

int end\_time;

int stayinline; //stayinline用来表示是否有等待信息，0表示无，1表示有。

};

PCinfo info[100][100];

int Month[]={0,31,28,31,30,31,30,31,31,30,31,30,31};

int date[12][30];

## 功能函数设计

### ·查询机位信息（void Query()）

### ·机位预定（void book\_computer()）

### ·退出预定（void book\_exit()）

### ·.查询等待信息（void search\_wait()）

### ·清空机位信息（void info\_empty()）

### ·统计预约次数（int statistical();）

### （1）. 查询机位信息（void Query()）

先让用户输入要查询的时间，故先设置判断用户是否输入正确，再进行相应的操作。

if((start\_time-8)/2>=0 && (end\_time-8)/2<=6){

time = (start\_time-8)/2;

}

else{

cout << "输入时间有误！" <<endl;

system("pause");

return;

}

system("cls");

if(month<1||month>12||day<0||day>Month[month]){

cout << "输入时间错误！" <<endl;

cout << "请重新输入指令" <<endl;

cout << "\t\t\t\t ----------------------------------------------" << endl;

cout << "\t\t\t\t| 1.重新输入 |" << endl;

cout << "\t\t\t\t| |" << endl;

cout << "\t\t\t\t| 2.返回菜单 |" << endl;

cout << "\t\t\t\t ----------------------------------------------" << endl;

cin >> n;

if(n==2){break;}

if(n==1){break;}

}

else break;

}

if(n==2){return;}

if判断语句中的(start\_time-8)/2>=0 && (end\_time-8)/2<=6)是判断其是否在6个时间段内，若不在则提示用户输入错误，并给出提示和选项。若正确则进行下一步输入操作。其中我为了方便将2月份的时间都视为28天，没有考虑闰年有29天的情况。所以这是本程序一处有待改进的地方。

接下来我会让用户输入要查询第几次预约的信息，并且利用for循环将20台计算机的预约情况打印在屏幕上。

cin >> a;

for(i=1;i<=20;i++){

if(date[month][day]==1&&info[i][a].State[time]==1){

cout << "\t\t\t\t第" << i << "号计算机在该时间段已被预约" <<endl;

}

else {cout << "\t\t\t\t第" << i << "号计算机在该时间段未被预约" <<endl;}

}

（2）. 机位预定（void book\_computer()）

该函数是本设计较为核心的地方，用0表示空位，1表示无空位。

还是先让用户输入要预约的机位和时间，利用if进行判断是否有误，有误则给出提示和选项，无误继续下面的程序。

本程序给出两种预约方式，一种是顺序预约，另一种是编号预约。

不管是哪种预约方式，都要先判断一下机位是否有空，利用if来判断。

if(date[month][day]==0||info[i][s].State[time]==0){

date[month][day]=1;

info[i][s].State[time]=1;

cout << "\t\t预约成功！";

system("pause");

}

else{cout << "该机位已被预约！"};

在第二种预约方式中若机位已被预约，则给出选项是否加入等待队列或是换一个机位。若选择前者，则需要用户输入个人信息。若选择后者则利用以下for循环给出最佳方案

for(i=month;i<=12;i++){

for(j=day;j<=Month[i];j++){

for(k=1;k<=6;k++){

for(m=1;m<20;m++){

if(date[i][j]==0||info[m][s].State[time]==0){

printf("你可以预约到最近时间段的计算机是：第%d号,%d年%d月%d日%d时-%d时\n",m, year, i, j, (2\*k+6), (2\*k+8));

}

}

}

}

}

（3）. 退出预定（void book\_exit()）

与上面的函数同样的需要先判断用户的时间输入是否正确，然后再进行相应的操作。

利用下面的代码进行取消预约的操作，并且考虑到用户在该时间段内并无预约的情况。

if(n==2){return;}

cout << "请输入您要取消预约的机号";

cin >> n;

cout << "您要取消第几次预约的机位？";

cin >> s;

if(info[n][s].State[time]==1){

info[n][s].State[time]=0;

cout << "取消预约成功！";

}

else {

cout << "在该时间段内您没有任何预约！";

}

1. . 查询等待信息（void search\_wait()）

cout << "请输入您要查询等待信息的机位号:";

cin >> m;

cout <<endl;

cout << "\t\t\t\t======正在查询中，请稍后~~======" <<endl;

printf("第%d号计算机:\n\n" ,m);

printf("您准备查找第几次预约信息？\n\n");

cin >> s;

if(info[m][s].State[time]==0&&info[m][s].stayinline==0){

cout << "无等待信息！" <<endl;

}

else{

cout << "该机位在该时间段有等待成员！" <<endl;

cout <<endl;

for(i=month;i<=12;i++){

for(j=day;j<=Month[i];j++){

for(k=1;k<=6;k++){

for(m=1;m<20;m++){

if(date[i][j]==0||info[m][s].State[time]==0){

printf("你可以预约到最近时间段的计算机是：第%d号,%d年%d月%d日%d时-%d时\n",m, year, i, j, (2\*k+6), (2\*k+8));

system("pause");

return;

}

}

}

}

}

}

（5）. 清空机位信息（void info\_empty()）

利用for循环对其进行赋0操作

int i, j, k, a;

for(i=0;i<=20;i++){

for(a=0;a<=20;a++){

for(j=0;j<=6;j++){

info[i][a].State[j]=0;

}

}

}

for(j=0;j<=12;j++){

for(k=0;k<=31;k++){

date[j][k]=0;

}

}

（6）. 统计预约次数（int statistical();）

函数较为简单，可说的不多，只是在预约函数中需要调用此函数，让程序更加完善。

## 系统参数图

|  |  |  |
| --- | --- | --- |
| main()参数 | | |
| 类型 | 参数名 | 功能 |
| int | select | 菜单选项输入 |
| Int | State | 机位状态信息 |
| int | year | 年的变量表示 |
| int | month | 月的变量表示 |
| int | day | 日期的变量表示 |
| int | start\_time | 开始时间的变量表示 |
| int | end\_time | 结束时间的变量表示 |
| int | stayinline | 等待信息的变量表示 |
| int | Month[] | 储存月份 |
| int | date[] | 储存日期 |
| int | i | 循环计数变量 |
| int | j | 循环计数变量 |
| int | k | 循环计数变量 |
| int | m | 循环计数变量 |
| int | n | 临时选项输入 |
| int | s | 预定次数 |
| char | name[] | 用户姓名 |
| char | sex[] | 用户性别 |
| char | tel[] | 用户联系方式 |

**（三） 用户手册**

1.程序运行时，首先显示各项功能选项的菜单，等待用户输入。

2.选择相应的需求实现相应的功能。

2.1.查询机位信息：输入要查询的时间，系统会列出在该时间段内各机位的状态信息。

2.2.预定机位：输入预定时间，选择预定方式。若机位在该时间段内已被预定，则系统会给出提示，选择加入等待队列或者系统提供最优方案。选择加入等待队列需要用户输入个人信息，后续会有提醒。

2.3.取消预订：输入时间和机位号进行取消预约的操作，若无预约，系统也会有相应的提示。

2.4.查询等待信息：输入时间和要查询的机位号，若该机位无等待信息会有相应提示，若有等待信息，则会提供等待时间最短的方案。

3.退出系统：顾名思义，直接退出系统。

**（四） 调试及测试**

我在代码编写过程中使用的codeblocks，可能会有部分函数代码在其它编译系统中需要修改后才能使用，但由于本人见识有限，只使用过codeblocks和DevC++，而上述代码在这两套编译系统中都能通过，但不明确是否存在在某个编译环境中会报错的情况。

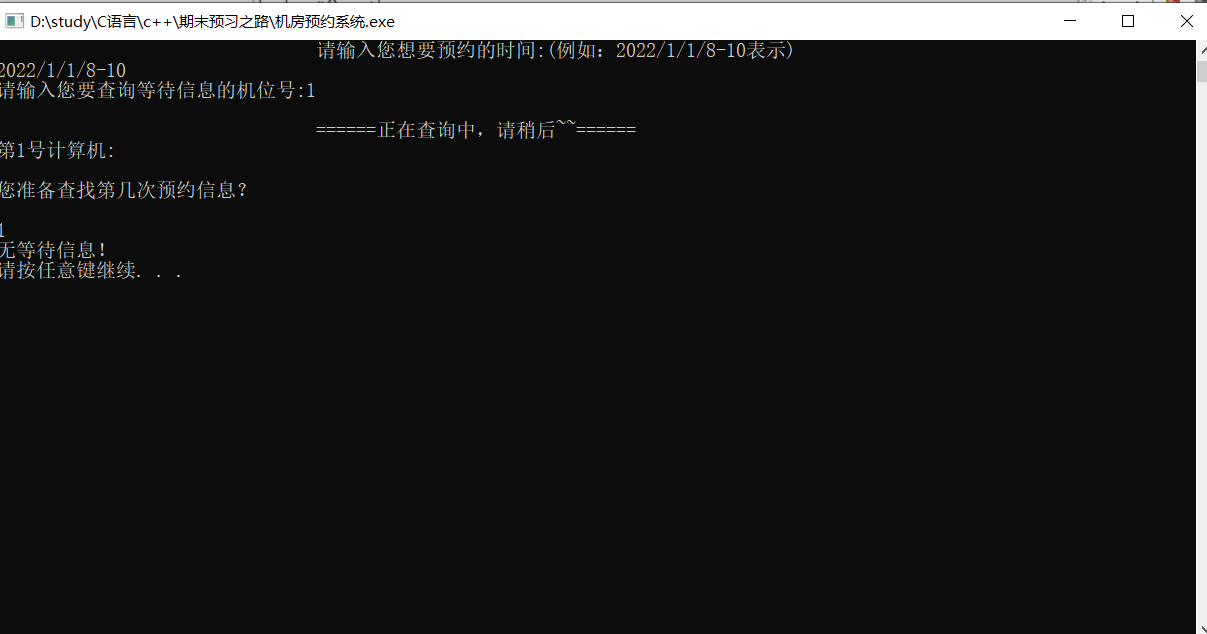
除此之外，我在调试过程中遇到过最多的三个问题是：绘制菜单表格，循环及判断分支过多以及对机位信息状态的判断。对于第一个问题，我在编写这段代码时就出现两边“|”过多导致“出头”的结果，并且“\t”我在第一次设计时只加了2个，运行时才发现用4个更合理，关于第三个问题，2小时为一个时间段，早上8点到下午8点，6个时间段，0表示空位，1表示无空位，我在前面也有详细的叙述。这里我想好好说说第二个问题。

关于第二个问题，循环及判断的分支过多，确实很影响编写和查找错误，尤其是逻辑错误，很容易因为缺少“}”或者因为“}”的位置不对而导致的代码和其相对应的功能不能实现的问题，这里我也犯了不少相关的错误，尤其是机位预定（book\_computer）这个函数里，由于循环和判断分支太多，我搞混了两个“}”的位置，起初导致自定义编号预定机位功能和查找最优方案的功能不能实现。遇到这种逻辑问题，系统是不会发现的，必须要我们静下心来去一个个从头查验，保证自己的清醒状态是很重要的，不能急躁，一步步去试，我也是改了五次才解决这个问题。

**（五） 运行实例：**

# 





**（六）进一步改进**

（1）目前程序中，没有考虑闰年二月份有29天的情况，为了方便起见，二月都以28天作为判断依据，可以加入年份判断函数完善。

（2）没有考虑一个机位可以有储存多个等待信息的情况，可以再设置数组进行表示和存储。

**（七）心得体会**

通过课程设计，自己的收获是很多的，毕竟代码是真的自己一个人独立完成的，当然我在整个课设的制作过程中必然遇到过大大小小的问题，有些至今已经解决，有些苦思冥想也没搞明白，还有些虽然和自己的预想有些差异，但却是向着好的方向发展的，这其中也有身边的朋友给予我一些想法上的帮助，没有他们的热情的帮助，我也很难在半个月内独立完成这个目前来说是我最大的一个代码项目。

通过课程设计，真的让我学到了很多，也让自己的到了锻炼。首先就是对我的代码编写能力的锻炼，这是我第一个代码编写超过四百的项目，而为了这个项目，我实际上敲了远不只是四百行代码。在四百多行的代码中寻找bug真是一件枯燥又有趣的事。除了代码编写能力的提升，我认为提升最大的就是断点调试方法的使用。通过监测各个变量的变化来找出bug 的具体位置，然后修复它。尤其在调试功能的过程，真的很煎熬，一遍遍输入，一遍遍改动，再一遍遍输入，循环往复才能得到最终自己想要的结果。毕竟咱是计算机专业的学生，这些能力都是需要自己实践才能得到，课程设计就是给予我们这样一个机会，在规定的项目下尽可能的完成它，虽然我的代码还有改进空间，但我认为这个是我目前在一个人的情况下所能达到的较好水平。如果代码存在着一些致命性问题，还请老师批评指正，我会努力修改。也希望在未来可以有更多的课设机会锻炼我，以面对未来更长的代码和更对的 bug！

1. **对课程设计的建议**

希望题目的考察知识面更加广泛，能更鼓励学生考虑更多的实际情况，不要增加过多的限制条件。