

高级程序语言实验报告

---时钟与闹钟

姓名：童佳燕

学号：1551445

班级：计算机科学与技术1班

指导教师：沈坚

完成日期：2017. 4. 23

装

订

线

1. 实验题目：时钟与闹钟实现过程

1.1 问题描述： cmd 下的为图形界面实现一个时钟及闹钟工具

1.2 题目要求：

1.2.1 整体要求：

- (1) 能以指针式和数字式两种形式显示一个时钟，两种显示方式之间可以快捷键进行切换。
- (2) 当前的年月日，要以某种方式显示出来。
- (3) 时钟的时间取系统时间，如果通过 Windows 控制面板/任务栏调整当前系统时间，则时钟的显示时间要同步变化
- (4) 支持定义闹钟，闹钟要允许多个，闹铃方式（每日/每工作日/单次），到达闹钟设置的时间点后，能够播放音乐。
- (5) 闹钟的定义要能够关闭程序并挨次启动后仍然生效。
- (6) 整体设计自由发挥，自行加入认为可行的部分。

1.2.2 显示要求：

时钟/闹钟中的数字/字符/汉字的大字体显示，可以自行构造或者通过汉字库读入相应的点阵信息并显示。

1.2.3 代码要求

- (1) 尽量使各菜单项中的程序公用函数，用参数解决细微差异。
- (2) 各函数代码长度尽量不要超过 50 行
- (3) 不允许使用全局变量，全局指针以及全局数组，允许使用宏定义。
- (4) 只允许使用目前为止讲过的内容（包括课后补充知识，不允许用类等），可以使用网上查到的系统函数。
- (5) 为代码加入注释，命名统一，方便阅读。

2. 整体设计思路（以题目 3 为主）

主要分成时钟和闹钟两个部分，考虑到时钟的显示需要用小字体，以及利用汉字库输出点阵字体，字体的大小不适合输出较多的字（美观），所以放弃将两者放在同一界面的想法。设计思路如下：

2.1 第一部分：时钟

时钟分为四个主要任务：取系统时间，显示数字式，显示指针式，设置快捷键。

2.1.1 取系统时间

这部分通过查资料，实现较容易：

2.1.2 显示数字式

这部分主要涉及的任务是：点阵文字的显示。

首先，需要获取文字的区位码，再将区位码转换成机内码。中间需要利用 HZK16 以及 HZK16F 文件，从中对应位置读取信息。也就是说需要一个偏移量，才可以获取其机内码。

2.1.3 显示指针式

除了同样需要显示点阵文字意外，这部分需要解决 cmd 界面画圆，画指针，定位画点，填充区域两个主要任务。其中填充区域可以由画若干指针，不擦去达到。画点可以由画圆，并实时擦去达到。因此，主要解决画圆和画指针，同时解决指针长度问题。

2.1.4 设计快捷键

设计快捷键在两者之间切换

这个功能的实现还是较容易的，比如在电子表界面，输入快捷键后，执行打印指针表界面的函数即可。同时为了用户使用的满意度，加入了若干快捷键，比如“返回首页”，“停止音乐”“切换”等。

2.2 第二部分：闹钟

闹钟部分涉及的主要任务是：删除/添加闹钟设置以及相关文件的读写，cmd 播放音乐，是否到时间的判断。

2.2.1 删除/添加闹钟设置以及文件读写：

考虑到程序关闭后再次启动闹钟设置不变的要求，因此需要将设置信息存入文件；配置信息的格式尽量按照方便读写的方式安排。

添加设置，可以直接把信息存放在文件的末尾，而删除设置，需要将要删除的记录后面的

记录先存在临时数组中，然后将文件大小缩小到只有要删的记录前面的记录留下的大小，最后将临时数组中的信息存入配置文件。同时需要进行各种情况的判断。

2.2.2 Cmd 播放音乐

自行查找资料

2.2.3 是否到时间的判断

文件中存储的，为了方便读取，应该是将一条记录存在一行中，包括信息有，日期，时间，音乐 形式等，所以需要将一条记录“析解”，拆开转换成适合类型的数据，方便进行后续处理。遍历文件，寻找是否与系统时间匹配的记录，如果有，则播放音乐；

3. 主要功能的实现

3.1 点阵文字显示

3.1.1 获取文字机内码

在有文字库的前提下，获取机内码实际上只需要获取文字在文字库中的偏移量，然后利用文件指针 seekg 定位，并用二进制读取 32 个字节（ $16\text{bit} \times 16\text{bit} = 256\text{bit} = 32\text{byte}$ ）即可。因此，问题实质是如何获取偏移量。在我理解里，GB2312 是一个 94×94 的二维表，行就是“区”、列就是“位”。而例如 word[3] = “汉”，word[1] 是“汉”字的区码，word[2] 是“汉”字的位码。获取区码和位码之后，偏移量其实显而易见。

$$\text{offset} = (94 * (\text{区码} - 1) + (\text{位码} - 1)) * 32$$

【注意：区码和位码都是从 1 开始计数的】

3.1.2 根据机内码显示文字

由于 16×16 显示，因此采用将 32 字节用 short 型数据存储。这里有一个注意点，也就是计算机内部存储是由低位向高位存储，也就是说 short 行数据 1-16bit，需要先对 9-16 进行处理，输出，在输出 1-8。而决定是否要输出的因素也就是二进制最高位上是否是 1。是 1 则输出，否则不输出。具体实现是（对于后八位而言），循环，一次将 short 型数据 & 0x0080 后，判断是否得到非零值，以判断最高位是否为 1，然后左移一位做继续进行循环。

3.2 画指针/画钟盘

3.2.1 画圆盘

这部分一开始以为是最难得，但事实上，确定圆心之后，写出函数，就能找到圆上的点的坐标，利用 gotoxy 打印圆就可以实现。但是由于屏幕的比例不是 1:1 如果利用圆的函数，打印出的结果会呈现椭圆的形状；因此，选择打印椭圆的函数，显示的恰好是圆形，具体参数需要根据需要一点点调整。【遗留：只针对于已经确定的窗体大小，没有做到根据窗口大小改变】

3.2.2 画指针

指针需要根据时间实时进行改变，肯定是需要传入参数——t 的，因此根据 t 算与参考位置的夹角，利用 $\tan(x)$ 确定斜率，已知斜率和其中一点（圆心）就可以画出直线；而长度则利用两点间距离来确定。同圆盘一样，由于窗口比例，即使使用两点间距离来统一长度，但是不同斜率的直线，在显示上长度依旧是不同的。最终解决是，改变了两点间距离公式，为其中的 y 坐标添加了合适的参数。

同样，由于获取系统时间的函数一秒内能执行很多次，而对于显示而言，是不需要的！因此，设置三重循环，当秒数前后两个值的差值绝对值大于 1 时，才出循环，并打印指针。同时判断分数前后的值是否绝对值大于 1，决定是否重新打印分针，同理时针。

随着时间变化，需要注意的是，当两指针重合时，比如秒针和时针，表针转动的效果是打印当前的指针，擦去之前的指针，因此如果只处理秒针的话，时针就被擦去了。因此需要对是否重合进行判断，并决定是否需要重新输出时针。

4. 调试过程中遇到的问题

4.1 画圆中间部分“截断”

程序一开始，画圆的函数是通过确定 x 的上下限，利用圆的几何函数，确定 y 的值画圆。而问题在每次循环 $x+=1$ ！事实上，在画圆过程中，越接近水平半径，单位 x 的变化给 y 值带来的变化越大！因此，如果是 $x+=1$ ，一段肉眼可见的位置才会出现一个点，也就出现了圆被截断的显示。

解决方法是，加大精确度，也就是减少每次循环 x 的变化量。虽然最终利用 gotoxy 都会被化成 int 值，但是若干个排列之后，会使圆更加完整光滑！精度越大，圆盘也就越光滑，但相应的输出速度就差很多了！综合两方面，最后定在 0.01，并用 printf 输出，尽可能减少速度影响。

4.2 秒针到 59-0 时就停下不再转动了

问题如题，事实上，为了不是每次执行系统获取时间的函数都打印一次指针，作业中是设置了循环，记录前一个数据，当当前秒数-前一个秒数=1 时，跳出循环，进行指针打印。但事实上，当秒针指向 59/0 时，下一秒-当前=-59，因此跳不出循环！

解决方法：之后加入了闹钟之后，设定是，到时间之后播放音乐并且文字提示，同时时间暂停，按键之后才会继续显示正确时间；因此，这中间几乎是无法保证刚好差 1 秒。因此，最终设定是差值小于-1 或大于 1，则跳出循环。因为这个时候的确也需要打印指针了。

4.3 音乐无法播放

这个问题结结实实困扰了我很久很久，逼得我只能怀疑电脑问题了！同样的代码，在别人电脑上能正常跑，但是在自己电脑上就死活放不出声音。函数换了一个又一个，都行不通。最后，在同学的提醒下，发现音乐格式是很重要的原因！

下载的音乐主要都是 mp3 格式，在转换过程中，不是改属性，改文件名就可以的，需要“格式工厂”进行正规的文件类型转换，才可以正常播放。

其次，第二个问题，mciSendString 函数，第一个参数如果是常量字符串，比如说“play 1.wav”就可以正常播放，而如果是数组名，就无法播放，即使强制转换成 LP 格式。这严重影响最后成果的完整性（本来准备提供用户音乐文件选择功能）。

5. 心得体会

5.1 经验教训

(1) 程序设计，先设计再写程序！

(2) 【strcpy 的使用，首先，需要 cstring 头文件，其次，第一个参数必须是字符数组，而不能是“流放”的指针，最后，在 vs2015 中，需要使用 strcpy_s】

修订：问题是_CRT_SECURE_NO_WARNINGS, 而非指针/数组问题

但是仍有问题，将这句话 define 在文件头部，对 strcpy 是起不到“保护”作用的，需要加在“项目”——“属性”——“C/C++”——“预处理器”——“预处理定义”中才可编译通过。

(3) cmd 界面只能呈现一种字体；

<<= !! not <<;

tan (45)! =tan(45 度)，括号里应该是弧度值；

文件读写一定要重视 clear()

5.2 问题分解与函数重用

5.2.1 问题分解

这一次是完全贯彻落实“先分析，后动手”！没有 demo，因此需要自己分析问题，找出几个主要任务，试着去找到解决办法。比如说，这次先分析时钟部分，抽象出的任务包括，获取系统时间，画圆画直线，输出大字体（汉字库）；而输出大字体，包括找到文件中文字对应位置，读取二进制以输出点阵字体两个部分。而闹钟部分，确定闹钟信息的存储模式——文件，结合时间问题设计闹钟模式，音乐播放等。

在正式建工程之前，将主要部分的功能大致实现，并将若干项目整合到一个解决方案里，方便查找！

5.2.2 函数重用

这次作业基本上没有用到以前写过的函数，或者说已经特别熟悉，所以没有复制粘贴。而项目内部而言，主要是画指针的函数，需要画时针，分针/秒针。而时针是 12 小时一圈，分针和秒针是 60 一圈，因此需要区分。通过在参数中加入 isHour 来进行判断。同时，三者的长度也是不同的，因此也加入了参数 len，决定画的直线的长度。

对于类似的重合程度相当大的函数而言，通过函数来进行不同部分的执行，个人认为还是比较合理的，当然，前提是不能太乱太复杂。

6. 源代码(主要函数)

```
Status printBitMapWord(unsigned char *word, int
baseX, int baseY)
{
    HANDLE hout =
    GetStdHandle(STD_OUTPUT_HANDLE);

    ifstream infile("HZK16", ios::in |
ios::binary);
    if (infile.is_open() == 0)
    {
        cout << "open error" << endl;
        return FAIL;
    }
    int offset=(94 * (unsigned int) (word[0] - 0xa0
- 1) + (word[1] - 0xa0 - 1)) * 32;//获得偏移量
    infile.seekg(offset, ios::beg);

    unsigned short bit[16];
    infile.read((char *)bit, 32);
```

```
void printPClock()
{
    HANDLE hout =
    GetStdHandle(STD_OUTPUT_HANDLE);
    printDate();
    char timeArr[64], word[3];
    int numTime[3], preNumTime[3];
    time_t t = time(0);
    strftime(timeArr, 64, "%X",
localtime(&t));
    for (int i = 0; i < 8; i += 3)
    {
        word[0] = timeArr[i];
        word[1] = timeArr[i + 1];
        preNumTime[i / 3] = atoi(word);
        numTime[i / 3] = preNumTime[i / 3];
    }
    drawPointer((double)numTime[2], SLEN,
0, 0);
```

```
infile.close();
for (int i = 0; i < 16; i++)
{
    /*从低位往高位存储*/
    int x = bit[i] & 0x00ff;
    for (int j = 0; j < 8; j++)
    {
        if (x & 0x0080)
        {
            gotoxy(hout, baseX + j, baseY +
i);
            cout << "*";
        }
        else
        {
            gotoxy(hout, baseX + j, baseY +
i);
            cout << " ";
        }
        x <<= 1; //aaaaaaaa <<= !!!!!!! <<

    }

    x = bit[i] & 0xff00;
    for (int j = 0; j < 8; j++)
    {
        if (x & 0x8000)
        {
            gotoxy(hout, baseX + j+8, baseY
+ i);
            cout << "*";
        }
        else
        {
            gotoxy(hout, baseX + j + 8, baseY
+ i);
            cout << " ";
        }
        x <<= 1;
    }
    cout << endl;
}
return OK;
} void drawPointer(double t, double pointerLen, int
isHour, int isClear)
{
    HANDLE hout =
GetStdHandle(STD_OUTPUT_HANDLE);
    double x = X;
    while (1)
    {
        if (t == 0 || t == 60 || isHour == 1 && (int)t %
12 == 0)
        {
            for (double y = Y; y > Y - pointerLen;
y -= 0.1)
```

```
drawPointer((double)numTime[1], MLEN,
0, 0);
drawPointer((double)numTime[0], HLEN,
1, 0);
word[2] = '\0';
while (!GetAsyncKeyState('B'))
{
    for (int i = 0; i < 3; i++)
        preNumTime[i] = numTime[i];
    while (!GetAsyncKeyState('B'))
    {
        time_t t = time(0);
        strftime(timeArr, 64, "%X",
localtime(&t));
        if (itIsTime(timeArr) == YES)
        {
            unsigned char str[9] = "
时间到! ";
            printBitMapStr(str, X +
103, Y - 38);
            /*播放音乐*/;
        }
        for (int i = 0; i < 8; i += 3)
        {
            word[0] = timeArr[i];
            word[1] = timeArr[i + 1];
            numTime[i / 3] =
atoi(word);
        }
        if (numTime[2] - preNumTime[2]
== 1 || numTime[2] - preNumTime[2] == -59) //
强调-59的重要性
            break;
        if (GetAsyncKeyState('K'))
        {
            system("cls");
            printEClockFrame();
            printEClock();
            return;
        }
        if (GetAsyncKeyState('B'))
        {
            HANDLE hout =
GetStdHandle(STD_OUTPUT_HANDLE);
            system("cls");
            setconsoleborder(hout,
100, 50);
            setfontsize(hout, L"新楷
体", 16);
            return;
        }
    }
}
drawPointer((double)preNumTime[2],
```


装

订

线

```

        {
            gotoxy(hout, (int)X, (int)y);
            if (isClear)
                printf("%c", ' ');
            else
                printf("%c", '-');
        }
        break;
    }
    else if (t == 30 || isHour == 1 && (int)t %
6 == 0)
    {
        for (double y = Y; y < Y + pointerLen;
y += 0.1)
        {
            gotoxy(hout, (int)X, (int)y);
            if (isClear)
                printf("%c", ' ');
            else
                printf("%c", '-');
        }
        break;
    }
    double rad;
    if (!isHour)
        rad = 6 * t / 360.0 * 2 * PI;
    else if (t > 12)
        rad = (t - 12) * 30 / 360.0 * 2 * PI;
    else
        rad = t * 30 / 360.0 * 2 * PI;
    double y = (-1) * (tan(0.5 * PI - rad)) * (x -
X) + Y;
    double len = (y - Y) * (y - Y) + (x - X) * (x
- X) / 3.2;

    gotoxy(hout, (int)x, (int)y);
    if (isClear)
        printf("%c", ' ');
    else
        printf("%c", '-');
    if (len > pointerLen * pointerLen)
        break;
    if (t < 30 && isHour != 1 || isHour == 1 &&
((int)t % 12) < 6)
        x += 0.1;
    else
        x -= 0.1;
}
return;
} Status itIsTime(char *timeArr)
{
    fstream file(fname, ios::in |
ios::binary | ios::out);
    if (file.is_open() == 0)

```

```

SLEN, 0, 1);
    drawPointer((double)numTime[2],
SLEN, 0, 0);
    if (numTime[2] == 0
|| numTime[2] - numTime[1] == 1)
    {
        drawPointer((double)preNumTime[1],
MLEN, 0, 1);

        drawPointer((double)numTime[1], MLEN,
0, 0);
    }
    if (numTime[1] == 0 ||
numTime[2] == (numTime[0] % 12) * 5 + 1 ||
numTime[1] == (numTime[0] % 12) * 5 + 1)
    {
        drawPointer((double)preNumTime[0],
HLEN, 1, 1);

        drawPointer((double)numTime[0], HLEN,
1, 0);
    }
}
system("cls");
setconsoleborder(hout, 100, 50);
setfontsize(hout, L"新楷体", 16);
return;
}

void printEClock()
{
    char timeArr[64], word[3];
    int x;
    time_t t = time(0);
    strftime(timeArr, 64, "%X",
localtime(&t));
    int numTime[3], preNumTime[3];
    for (int i = 0; i < 8; i += 3)
    {
        word[0] = timeArr[i];
        word[1] = timeArr[i + 1];
        preNumTime[i / 3] = atoi(word);
        numTime[i / 3] = preNumTime[i / 3];
    }

    /*将前一秒的时间记录在preNumTime中,
相差大于1或小于-1时才做新的打印,
[-1, 1]是考虑闹铃响的时候, 暂停所需的
时间*/

    printDate();
    while (!GetAsyncKeyState('B')) //按下A
停止

```

```

{
    printf("%s", "文件打开失败");
    return NO;
}

char tmp[30];
char s[64];
time_t t = time(0);
strftime(tmp, 30, "%Y %m %d ", localtime(&t));
strcpy(s, tmp);
strcat(s, timeArr);
s[strlen(s)] = '\0';

file.seekg(0, ios::end);
int fileLength = (int)file.tellg();
if (fileLength <= 2)
{
    file.close();
    return NO;
}
file.clear();
file.seekg(0, ios::beg);
char info[80], tjTime[30], tjMusic[80],
tjForm[20];
while (1)
{
    int index1 = (int)file.tellg(); //记录需
    要删除的记录头部的位
    置
    file.clear();
    file.getline(info, 80);
    int index2 = (int)file.tellg(); //记录需
    要删除的记录下一条头部的位
    置
    file.clear();
    getInfo(info, tjTime, tjMusic, tjForm);
    if (strncmp(tjTime, s, 16) == 0)
    {
        if (strcmp(tjForm, "单次") == 0) //
        音乐播放之后就需要删除该条记录
        {
            char *tmpArr = (char
            *)malloc(sizeof(char)*(fileLength - index2 + 10));
            if (tmpArr == NULL)
            {
                printf("%s", "空间申请失败
                ");
                return FAIL;
            }
            file.read(tmpArr, fileLength -
            index2);

            experimental::filesystem::resize_file(fname,
            index1);

            file.seekp(0, ios::end);
            file.write(tmpArr, fileLength -

```

```

{
    x = EBASEX + 7;
    for (int i = 0; i <
    (signed)strlen(timeArr); i++)
    {
        if (timeArr[i] == ':')
            x += 25;
        else
        {
            printFigure(timeArr[i],
            x, EBASEY + 7);
            x += 30;
        }
    }
    while (1)
    {
        time_t t = time(0);
        strftime(timeArr, 64, "%X",
        localtime(&t));
        if (itIsTime(timeArr) == YES)
        {
            unsigned char str[9] = "
            时间到! ";
            printBitMapStr(str,
            EBASEX + 90, EBASEY - 20);
            /*播放音乐*/;
        }
        /*将字符串中的信息分别提取,
        转换成数值型, 便于判断*/
        for (int i = 0; i < 8; i += 3)
        {
            word[0] = timeArr[i];
            word[1] = timeArr[i + 1];
            numTime[i / 3] =
            atoi(word);
        }
        if (numTime[2] -
        preNumTime[2] >= 1 || numTime[2] -
        preNumTime[2] <= -1)
            break;

        /*转换到指针型的表盘*/
        if (GetAsyncKeyState('K'))
        {
            system("cls");
            printPClockFrame();
            printPClock();
            return;
        }
        /*返回主菜单*/
        if (GetAsyncKeyState('B'))
        {
            HANDLE hout =
            GetStdHandle(STD_OUTPUT_HANDLE);

```

```

index2);

        free(tmpArr);
        file.close();
        return YES;
    }
    else if (strcmp(tjForm, "每工作日")
== 0)//需要对是否是工作日进行判断
    {
        char weekDay;
        time_t t = time(0);
        strftime(&weekDay, 1, "%w",
localtime(&t));
        file.close();
        if (weekDay > '0' && weekDay < '6')
            return YES;
        else
            return NO;
    }
    else if (strcmp(tjForm, "每日") == 0)
    {
        file.close();
        return YES;
    }
}
if (file.peek() == EOF)
    break;
}
return NO;
}
Status alarmDelete(fstream &file)
{
    HANDLE hout =
GetStdHandle(STD_OUTPUT_HANDLE);
    int index = 0, index2 = 0;
    char str[80];

    system("cls");
    cin.clear();
    cin.sync();
    printf("\n%s", "请输入您要删除的是第几条
设置: \n\n");
    cin >> index;//用于定位, 不需要从文件中一条一
条比对

    /*将需要删除的记录后面的记录存放在tmpArr动态
数组中
    利用
experimental::filesystem::resize_file(fname,
index);将文件缩小到只剩需要删除记录的前面几条
在文件末尾添加上tmpArr数组中的文件内容*/
    file.clear();
    file.seekg(0, ios::end);
    int fileLength = (int)file.tellg();

```

```

        system("cls");
        setconsoleborder(hout,
120, 50);
        setfontsize(hout, L"新楷
体", 16);
        return;
    }
}
for (int i = 0; i < 3; i++)
    preNumTime[i] = numTime[i];
if (GetAsyncKeyState('B'))
{
    HANDLE hout =
GetStdHandle(STD_OUTPUT_HANDLE);
    system("cls");
    setconsoleborder(hout, 120,
50);
    setfontsize(hout, L"新楷体",
16);
    return;
}
return;
}
Status alarmSet()
{
    HANDLE hout =
GetStdHandle(STD_OUTPUT_HANDLE);
    setconsoleborder(hout, 100,
BORDER_LINE);
    setfontsize(hout, L"新楷体", 16);
    fstream file(fname, ios::in | ios::out
| ios::binary);
    if (!file.is_open())
    {
        printf("%s", "文件打开失败");
        return FAIL;
    }

    char choice;
    while (1)
    {
        printSettedAlarm(file);
        buttonPrint();
        printf("%s", "请输入操作码: ");
        scanf("%c", &choice);
        if (choice == 'I')//添加
            alarmInsert(file);
        else if (choice == 'D')//删除
        {
            if(!alarmDelete(file))//如果
没有找到要求删除的记录
                printf("%s\n", "抱歉, 您没

```

装

订

线

```
file.clear();
file.seekg(0, ios::beg);
for (int i = 0; i < index - 1 || fileLength == 0;
i++) // 如果没有记录也要进来, 输出找不到记录的提示
{
    file.getline(str, 80);
    if (file.peek() == EOF || !fileLength)
    {
        printf("%s\n", "抱歉, 您没有设置该闹钟!");
        gotoxy(hout, 25, 14);
        printf("%s\n", "-----");
        gotoxy(hout, 25, 15);
        printf("%s\n", "| 返回 (B) |");
        gotoxy(hout, 25, 16);
        printf("%s\n", "-----");

        while (cin.get() != 'B') // 读到B返回
            return FAIL;
    }
}
file.clear(); // 很重要
index = (int)file.tellg();

while (1)
{
    file.getline(str, 80);
    if (strlen(str) > 2)
        break;
}
index2 = (int)file.tellg();

file.clear();
char *tmpArr = (char *)malloc((fileLength -
index2 + 10) * sizeof(char));
if (tmpArr == NULL)
{
    printf("%s", "申请空间失败");
    return FAIL;
}

file.read(tmpArr, fileLength - index2);
experimental::filesystem::resize_file(fname,
index);
file.seekp(0, ios::end);
file.write(tmpArr, fileLength - index2);
free(tmpArr);

gotoxy(hout, 25, 10);
printf("%s", " 删除成功!");
gotoxy(hout, 25, 14);
printf("%s\n", "-----");
gotoxy(hout, 25, 15);
```

```
有设置该时间的闹钟");
}
else if (choice == 'B')
{
    system("cls");
    return FAIL;
}
}
file.close();
return OK;
}

void alarmInsert(fstream &file)
{
    HANDLE hout =
GetStdHandle(STD_OUTPUT_HANDLE);
    system("cls");
    char insertInfo[80];
    cin.clear(); // 忽略输入流中前面操作输入的回车键
    cin.sync();
    printf("\n%s", " 请输入您要设置的时间, 音乐 (列表见下) 以及形式 (每日, 每工作日, 单次), \n\n 如 \"2017 04 08 12: 00: 00*a.wav*每天\", \n\n 以*作为分割符, 以#作为结束符: \n\n");
    printf("\n%s", " Maroon 5、Future - Cold.wav\n");
    printf("\n%s", " OneRepublic - Apologize.wav\n");
    printf("\n%s", " Sam Smith - Lay Me Down.wav\n");
    printf("\n%s", " Shawn Mendes - Mercy.wav\n");
    printf("\n%s", " The Weeknd、Daft Punk - Starboy.wav\n");
    printf("\n%s", " 请输入添加信息\n");
    cin.clear(); // 忽略输入流中前面操作输入的回车键
    cin.sync();
    cin.getline(insertInfo, 80, '#'); // 以#作为结束
    int l = strlen(insertInfo);
    insertInfo[l] = '#'; // 配置文件的末尾也需要有#方便读取时进行判断
    insertInfo[l+1] = '\0';

    file.clear();
    file.seekp(0, ios::end);
    file.write(insertInfo,
strlen(insertInfo));
    file.write("\r\n", 2);
```

<pre>printf("%s\n", " 返回 (B) "); gotoxy(hout, 25, 16); printf("%s\n", "-----"); while (cin.get() != 'B');//读到B返回 return OK; }</pre>	<pre>gotoxy(hout, 50, 10); printf("%s", " 添加成功! "); gotoxy(hout, 50, 14); printf("%s\n", "-----"); gotoxy(hout, 50, 15); printf("%s\n", " 返回 (B) "); gotoxy(hout, 50, 16); printf("%s\n", "-----"); while (cin.get() != 'B');//读到B返回 return; }</pre>
--	---

装
订
线