

02. 把文件中的数据存放在容器中

任务目标:

1. 将.ini文件中的数据按行提取，然后放到struct结构体
2. 将提取来的结构体放在vector中

练习:

1. 文件操作，打开文件操作Open，提取文件行操作
2. 字符串处理操作

代码:

1. 文件操作类: CFile
2. 字符串操作类: CCmdStr
3. 日志类: CLogFile

```
/*
 * 程序名: crfsturfdatal.cpp 本程序用于生成全国气象站点观测的分钟数据
 * 作者: Aries Thanks
 * */

#include "_public.h"

/*
 * 省 站号 站名 纬度 经度 海拔高度
 * 安徽, 58015, 砀山, 34.27, 116.2, 44.2
 *
 * */

//全国气象站点参数结构体
struct st_stcode{
    char provname[31]; //省
    char obtid[11]; //站号
    char obtname[31]; //站名
    double lat; //纬度
    double lon; //经度
    double height; //高度
};

vector<struct st_stcode> vstcode;

//把气象站点文件逐行提取放在全局容器中
bool LoadSTCode(const char* inifile);

CLogFile logfile; //日志
```

```

int main(int argc, char* argv[]){
    // 1. inifile 2.outpath 3.logfile
    if(argc!=4){
        printf("Using: ./crtsurfdata1 inifile outpath logfile\n");
        printf("Example: /project/idc1/bin/crtsurfdata1
/project/idc1/ini/stcode.ini /tmp/surfdata /log/idc");

        printf("inifile 全国气象站点的参数文件名字\n");
        printf("outpath 全国气象站点数据文件存放的目录\n");
        printf("logfile 本程序运行的文件日志名字\n");

        return -1;
    }

    if(logfile.Open(argv[3])==false){
        printf("logfile.Open(%s) failed.\n", argv[3]);
        return -1;
    }

    logfile.Write("crtsurfdata1 开始运行。 \n");

    //这里插入业务代码
    if(LoadSTCode(argv[1])==false) return -1;

    return 0;
}

bool LoadSTCode(const char* inifile){
    //打开站点参数文件

    CFile File;

    if(File.Open(inifile,"r")==false){
        logfile.Write("File.Open(%s) failed.\n",inifile);
        return false;
    }

    char strBuffer[301]; //获取行的时候用的buffer

    CCmdStr CmdStr; //拆分字符串

    struct st_stcode stcode;

    /*全国气象站点参数结构体

    struct st_stcode{

```

```

char provname[31]; //省
char obtid[11]; //站号
char obtname[31]; //站名
double lat; //纬度
double lon; //经度
double height; //高度

};*/

while(true)
{
    //从站点参数文件中, 读取一行文件操作对应CFile类中的Fgets(), 类似于C中的库函数fgets()

    // bool Fgets(char * buffer, const int readsize, bool bdelcrt==false)
    // buffer: 用于存放数据的内容, buffer必须大于readsize+1, 否则可能造成内存的溢出
    // readsize: 本次打算读取字符的字节数, 如果已经读到了结束标识符'\n', 函数返回
    // bdelcrt: 是否删除行结束标识符'\r'和'\n', true-删除, false-不删除, 默认false
    // 返回值:true成功, false失败, 一般情况下, false可以认为文件结束
    if(File.Fgets(strBuffer, 300, true)==false) break; //一般来说是要将buffer初始
    化的, 不然会出现问题, 但是Fget中有对buffer的初始化, 所以这里暂时不用管
    logfile.write(("%s\n", strBuffer));

    //把读到的一行进行拆分:
    //拆分字符串的类CCmdStr, 其中的成员变量vector<string> m_vCmdStr用于存放拆分后的字符
    (public)
    /*
        void SplitToCmd(const string &buffer, const char* sepstr, const bool
    bdelSpace=false)
        含义:把字符串拆分到m_CmdStr容器中
        参数1: 被拆分字符串
        参数2:buffer中采用的分隔符, 注意:sepstr参数的数据类型是字符串, 而不是字符
        参数3:拆分后是否删除字段内容前后空格:true-删除, false-不删除, 缺省表示不删除
    */

    /*
        int CmdCount();
        含义:获取拆分后的容器大小, 即m_vCmdStr容器的大小
    */

    /*
        bool GetValue(const int inum, char* value, const int ilen=0);
        含义:从m_vCmdStr容器中获取字段内容
        参数1: 类似于数组的下标
        参数2: 传入变量的地址, 用于存放字段内容
        返回值: true-成功, 如果inum的值超出了m_vCmdStr的大小, 返回失败
        (重载函数很多.....)
    */

    CmdStr.SplitToCmd(strBuffer, ",");

    if(CmdStr.CmdCount() != 6) continue; //第一行数据没用
    //把站点参数的每个数据项保存到站点参数结构体

    CmdStr.GetValue(0, stcode.provname, 30); //省
    CmdStr.GetValue(1, stcode.obtid, 10); //站号

```

```
CmdStr.GetValue(2,stcode.obtname,30); //站名
CmdStr.GetValue(3,&stcode.lat); //纬度
CmdStr.GetValue(4,&stcode.lon); //经度
CmdStr.GetValue(5,&stcode.height); //高度

//把站点参数结构体放入站点参数容器
vstcode.push_back(stcode);

}

for(int i=0;i<vstcode.size();i++){
    logfile.write("provname=%s, obtid=%s,obtname=%s, lat=%.2f,
lon=%.2f,height=%.2f\n",
vstcode[i].provname,vstcode[i].obtid,vstcode[i].obtname,vstcode[i].lat,
vstcode[i].lon,vstcode[i].height);
}
//关闭文件

return true;
}
```