



首页

新闻

博问

专区

闪存

班级

代码改变世界

?

注册

登录

小河沟大河沟

----- 梦想还是要有的，万一实现了呢！纸上得来终觉浅 绝知此事要躬行！



博客园 首页 新随笔 联系 管理 订阅 [HTML](#)

随笔- 829 文章- 51 评论- 70 阅读- 224万

昵称: ranjiewen

园龄: 7年

粉丝: 307

关注: 106

[+加关注](#)

C++用_findfirst 和 _findnext 查找文件

一、这两个函数均在io.h里面。

二、首先了解一下一个文件结构体：

```
struct _finddata_t {
    unsigned      attrib;
    time_t        time_create;
    time_t        time_access;
    time_t        time_write;
    _fsize_t      size;
    char          name[260];
};
```

time_t，其实就是long
而_fsize_t，就是unsigned long

现在来解释一下结构体的数据成员吧。

attrib，就是所查找文件的属性：_A_ARCH（存档）、_A_HIDDEN（隐藏）、_A_NORMAL（正常）、_A_RDONLY（只读）、_A_SUBDIR（文件夹）、_A_SYSTEM（系统）。

time_create、time_access和time_write分别是创建文件的时间、最后一次访问文件的时间和文件最后被修改的时间。

size：文件大小

name：文件名。

三、用_findfirst 和 _findnext 查找文件

1、_findfirst函数：long _findfirst(const char *, struct _finddata_t *);

第一个参数为文件名，可以用“*.*”来查找所有文件，也可以用“*.cpp”来查找.cpp文件。第二个参数是_finddata_t结构体指针。若查找成功，返回文件句柄，若失败，返回-1。

2、_findnext函数：int _findnext(long, struct _finddata_t *);

第一个参数为文件句柄，第二个参数同样为_finddata_t结构体指针。若查找成功，返回0，失败返回-1。

3、_findclose()函数：int _findclose(long);

只有一个参数，文件句柄。若关闭成功返回0，失败返回-1。

```
#include <io.h>
#include <iostream>
#include <fstream>
using namespace std;

bool transfer(string fileName, int exeNum );
void dfsFolder(string folderPath, ofstream &fout);

int main()
{
    _finddata_t file;
    int k;
    long HANDLE;
    k = HANDLE = _findfirst("*.*", &file);
    while (k != -1)
    {
        cout << file.name << endl;
        k = _findnext(HANDLE, &file);
    }
}
```

2022年12月						
日	-	二	三	四	五	六
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

搜索

我的标签

中国大学MOOC-陈越、何钦铭-数据结构-2017春(42)
 tensorflow(29)
 图像处理算法(28)
 树(27)
 递归+DFS+BFS+回溯(26)
 检测网络(22)
 数据结构(21)
 dp算法(21)
 算法之美(17)
 Paper(17)
[更多](#)

随笔分类

C/C++(84)
 CrackingtheCodingInterview(20)
 CV经典问题实践(10)
 GAN(3)
 Git(5)
 Java(2)
 Kaggle(1)
 leetcode(169)
 Linux(17)
 Matlab(18)
 MFC(24)
 OJ(11)
 OpenCV(12)
 Python(34)
 Qt(5)
[更多](#)

文章分类

C/C++(3)
 paper-分类, 检测, 分割, GAN...(20)
 典型算法题目(2)
 机器学习(13)
 深度学习(15)
 图像处理(9)

机器学习达人

红色石头的专栏
 July's Pensieve
 白马负金羁

```

    }

    _findclose(HANDLE);

    transfer("C:\\Windows\\*.exe", 0);
    ofstream o_fstream;

    dfsFolder("E:\\\\WHU\\\\Study", o_fstream);

    return 0;
}

//_findfirst 函数返回的是匹配到文件的句柄，数据类型为long。
//遍历过程可以指定文件类型，这通过FileName的赋值来实现，例如要遍历C : \\WINDOWS下的所有.exe文件

bool transfer(string fileName , int exeNum)
{
    _finddata_t fileInfo;
    long handle = _findfirst(fileName.c_str(), &fileInfo);

    if (handle == -1L)
    {
        cerr << "failed to transfer files" << endl;
        return false;
    }

    do
    {
        exeNum++;
        cout << fileInfo.name << endl;
    } while (_findnext(handle, &fileInfo) == 0);
    cout << ".exe files' number: " << exeNum << endl;

    return true;
}

//遍历文件夹及其子文件夹下所有文件。操作系统中文件夹目录是树状结构，使用深度搜索策略遍历所有文件

//在判断有无子目录的if分支中，由于系统在进入一个子目录时，匹配到的头两个文件(夹)是".." (当前目录)
//需要忽略掉这两种情况。当需要对遍历到的文件做处理时，在else分支中添加相应的代码就好

void dfsFolder(string folderPath, ofstream &fout)
{
    _finddata_t FileInfo;
    string strfind = folderPath + "\\*";
    long Handle = _findfirst(strfind.c_str(), &FileInfo);

    if (Handle == -1L)
    {
        cerr << "can not match the folder path" << endl;
        exit(-1);
    }
    do{
        //判断是否有子目录
        if (FileInfo.attrib & _A_SUBDIR)
        {
            //这个语句很重要
            if ((strcmp(FileInfo.name, "..") != 0) && (strcmp(FileInfo.name, "..."))
            {
                string newPath = folderPath + "\\\" + FileInfo.name;
                dfsFolder(newPath, fout);
            }
        }
        else
        {
            fout<<folderPath.c_str() << "\\\" << FileInfo.name << " ";
            cout << folderPath.c_str() << "\\\" << FileInfo.name << endl;
        }
    } while (_findnext(Handle, &FileInfo) == 0);

    _findclose(Handle);
    fout.close();
}

//#include <iostream>

```

zouxy09的专栏
 Real_Myth--CV
 DFan的NoteBook(细致paper阅读)
 系统学习深度学习
 Poll的笔记
 拾毅者
 Rachel Zhang的专栏
 寒小阳
 AI之路

刷题达人

Grandyang-LeetCode
 Zhu Li
 Code Ganker-leetcode

我的链接

ranjiewen-知乎
 个人主页
 Github
 CSDN
 牛课网

阅读排行榜

- pytorch之dataloader深入剖析(180087)
- 模拟退火算法(83310)
- 大津法---OTSU算法(72614)
- MultiByteToWideChar和WideCharToMultiByte用法详解(68469)
- 图像质量评价方法PSNR+SSIM&&评估指标SROCC, PLCC(58032)
- matlab 重命名文件和文件夹(42809)
- C++用 _findfirst 和 _findnext 查找文件(29887)
- Python in/not in --- if not/if + for...[if]...构建List+ python的else子句(28502)
- matlab特征值分解和奇异值分解(27984)
- C++实现ping功能(25965)

```

//#include <string>
//#include <iostream>
//using namespace std;
//
//int main()
//{
//    _finddata_t file;
//    long longf;
//    string tempName;
//    // _findfirst返回的是long型; long __cdecl _findfirst(const char *, struct _fi
//    if ((longf = _findfirst("E:\\WHU\\Study\\*.*", &file)) == -1)
//    {
//        cout << "文件没有找到!\n";
//        return 0;
//    }
//    do
//    {
//        cout << "文件列表:\n";
//        tempName = file.name;
//        if (tempName[0] == '.')
//            continue;
//        cout << file.name<<endl;
//
//        if (file.attrib == _A_NORMAL)
//        {
//            cout << " 普通文件 ";
//        }
//        else if (file.attrib == _A_RDONLY)
//        {
//            cout << " 只读文件 ";
//        }
//        else if (file.attrib == _A_HIDDEN)
//        {
//            cout << " 隐藏文件 ";
//        }
//        else if (file.attrib == _A_SYSTEM)
//        {
//            cout << " 系统文件 ";
//        }
//        else if (file.attrib == _A_SUBDIR)
//        {
//            cout << " 子目录 ";
//        }
//        else
//        {
//            cout << " 存档文件 ";
//        }
//        cout << endl;
//    } while (_findnext(longf, &file) == 0); //int __cdecl _findnext(long, struct
//
//    _findclose(longf);
//
//    return 0;
//}

```



C/C++基本语法学习 STL C++ primer

分类: C/C++

标签: 文件操作



3
推荐

0
反对

« 上一篇：各种距离算法汇总
» 下一篇：[ofstream的使用方法](#)

posted @ 2016-10-14 16:26 ranjiewen 阅读(29888) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】阿里云新人特惠，爆款云服务器2核4G低至0.46元/天



编辑推荐：

- 应届毕业生程序员在面试时如何做好自我介绍？
- Redis 网络模型究竟有多强
- 不规则图形背景排版高阶技巧 -- 酷炫的六边形网格背景图
- [深度思考] 如何优雅的校验参数？
- 生产事故-错误密钥引发的接口通信问题

阅读排行：

- 外包公司中的"炼狱"，极度摧残，避免踩坑。
- 就聊聊不少小IT公司的技术总监
- 应届毕业生程序员在面试时如何做好自我介绍？
- 从Java开源项目中总结出的常见坏实践（Bad Practice）
- 学习.NET MAUI Blazor（三）、创建.NET MAUI Blazor应用并使用AntDe