

磁盘操作

一、问题描述

编写程序模拟磁盘工作。以大文件模拟磁盘，建立 FAT、文件目录等数据结构，进行列目录、文件增加删除等操作，实现文件操作命令：`dir`，`copy`，`del`。可考虑子目录操作：`md`(建子目录)、`cd`(转子目录)。

读写 VHD 虚拟磁盘只能整块(簇)读写，每次能且只能读写一个块(簇)。所以基础的 VHD 读写可编写以下函数：

```
int clusterRead(byte&, int); //int 为簇号
int clusterWrite(byte*, int);
```

- 重点一：熟悉对于二进制文件的随机读写。
- 重点二：熟悉掌握 FAT 文件系统的基本结构。

二、实现原理

本程序主要实现了七种关于文件的基本操作，分别是指令 `dir`，`copy`，`del`，`md`，`cd` 以及对于二进制文件的按块（簇）读写操作（`Read Block` 与 `Write Block`），以下介绍它们的实现原理。

2.1 DIR

该函数实现主要运用了 `io.h` 中的 `_finddata_t` 结构与 `time.h` 中的 `tm` 结构。首先定义一个句柄 `hFile`，将它赋值为 `_findfirst` 函数的返回值，若其值为 -1，则表示该目录下既没有文件也没有文件夹，`dir` 函数直接返回；若否，则持续调用 `_findnext` 函数，只要该函数返回值恒为 0，则表示该目录下的文件或文件夹尚未遍历完毕。每次遍历时，通过 `tm` 结构体得到文件的创建时间，通过 `_finddata_t` 结构体得到文件名与类型（是文件夹还是文件），按一定格式输出即可。

```
do
{
    p = localtime(&fileInfo.time_create);
    cout << p->tm_year + 1900 << "/" << p->tm_mon + 1 << "/" << p->tm_mday << " ";
    if (p->tm_hour < 10)
        cout << "0";
    cout << p->tm_hour << ":";
    if (p->tm_min < 10)
        cout << "0";
    cout << p->tm_min << " ";
    cout << fileInfo.name << (fileInfo.attrib & _A_SUBDIR ? " [folder]" : " [file]") << endl;
} while (_findnext(hFile, &fileInfo) == 0);
```

2.2 COPY

该函数实现主要运用了 C++ 中的文件操作函数 `read` 与 `write`。每次调用时，首先检查待复制的文件是否存在，若存在，则首先调用 `cd` 函数（下面会介绍）将程序的工作目录转到要复制到的目录下，建立一个新的同名文件，然后按块（簇）读取原文件中的内容，并按同样的规则与大小写入新文件。当文件剩余大小不足

一个块的大小时，直接拷贝剩余的全部字节。

```
char buff[CLUSTER_SIZE];
long long totalBytes = 0;
while (in)
{
    in.read(buff, CLUSTER_SIZE);
    out.write(buff, in.gcount());
    totalBytes += in.gcount();
}
```

2.3 DEL

该函数实现主要运用了 C++ 中的文件操作函数 `remove`。检查待删除文件确实存在以后，调用 `remove` 函数删除即可。

```
int flag = remove(fileName.c_str());
if (flag == 0)
{
    cout << "Deleting successful!" << endl;
}
else
{
    cout << "Deleting failed!" << endl;
}
```

2.4 MD

该函数实现主要运用了 `direct.h` 中的 `_mkdir` 函数。直接将子目录名与路径拼接后得到新的路径，调用 `_mkdir` 函数创建即可。

```
int flag = _mkdir(createPath.c_str());
//cout << flag << endl;
if (flag == 0)
{
    cout << "Creating successful!" << endl;
}
else
{
    cout << "Creating failed!" << endl;
}
```

2.5 CD

该函数实现主要运用了 `direct.h` 中的 `chdir` 函数。

```
int flag = chdir(path.c_str());
if (!flag)
{
    return;
}
else
{
    cout << "Error!" << endl;
    return;
}
```

2.6 Read Block

由于读写 VHD 虚拟磁盘只能整块(簇)读写,每次能且只能读写一个块(簇),本程序事先约定每一块(簇)的大小为 512 字节,并据此来按块读写文件。

clusterRead 函数的主要实现原理如下:首先检查要读的文件是否存在,若存在,则通过 **getFileSize** 函数获取该文件的大小,据此得到文件中的块数。然后检查传入的块号是否有效,若有效,则调用 **seekg** 函数将文件指针置于待读的块的开头位置,以二进制形式读取该块的内容(512 个字节),最后将该内容存入一个新的文件 **ReadBufferx** 中,以供查看。

```
fp.seekg(clusterIndex * CLUSTER_SIZE);
char* readBuffer = new char[CLUSTER_SIZE];
fp.read(readBuffer, CLUSTER_SIZE);
char* cntStr = new char[10];
itoa(cnt, cntStr, 10);
string tmp = cntStr;
string writeFile = "ReadBuffer" + tmp;
fstream op(writeFile, ifstream::binary | ifstream::out);
op.write(readBuffer, CLUSTER_SIZE);
```

2.7 Write Block

与 **clusterRead** 函数类似, **clusterWrite** 函数也通过 **seekg** 函数将文件指针置于待写的块的开头位置,以二进制形式将 **writeBuffer** (由用户自行输入,大小为 512 字节,若输入大小不足则以随机形式补全)中的内容写入文件。

```
fp.seekg(clusterIndex * CLUSTER_SIZE);
fp.write(writeBuffer, CLUSTER_SIZE);
```

三、运行环境

1. 操作系统: Windows
2. 编译环境: Microsoft Visual Studio 2017

四、使用方法

1. 运行程序，进入初始界面。第一行显示目前所在的目录，此为初始目录。每次调用 CD 指令后，目录都会相应改变。输入数字选择相应的操作。

```
You are now in C:\Users\翁罗轩\Desktop\Project1\Project1
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit
```

2. 输入 1 选择 **DIR** 操作后，需输入所要查看目录的绝对路径，而非相对路径。

```
1
Enter the whole path.
D:\NodejsFirst-master
2019/6/8 21:17 . [folder]
2019/6/8 21:17 .. [folder]
2019/6/8 21:17 .gitattributes [file]
2019/6/8 21:17 .idea [folder]
2019/6/8 21:17 html [folder]
2019/6/8 21:17 model [folder]
2019/6/8 21:17 node_modules [folder]
2019/6/8 21:17 package-lock.json [file]
2019/6/8 21:17 register.sql [file]
2019/6/8 21:17 web_server.js [file]
```

3. 输入 2 选择 **COPY** 操作后，首先输入待复制的文件名（须在当前目录下），然后输入将要复制的目录（绝对路径）。

```
2
Enter the file name.
Project1.vcxproj.user
Enter the path where you want to copy this file.
C:\Users\翁罗轩\Desktop\新建文件夹
Copying successful!
```

4. 输入 3 选择 **DEL** 操作后，输入待复制的文件名（须在当前目录下）即可。

5. 输入 4 选择 **MD** 操作后，首先输入待建立的子目录名，然后输入它将要建立的根目录（绝对路径）。

```
4
Enter the folder name.
test
Enter the path where you want to create the folder.
C:\Users\翁罗轩\Desktop\新建文件夹
Creating successful!
```

6. 输入 5 选择 CD 操作后，输入新的目录（绝对路径）即可。
7. 输入 6 和 7 选择 Read/Write Block 操作后，首先输入文件名（须在当前目录下），然后输入块（簇）号。
8. 输入 0 退出程序。

五、测试结果

4.1 DIR

输入测试路径，可见程序正确显示了该目录下所有文件与文件夹以及它们的创建时间。（注意程序给出的是创建时间，而非第一张图中的修改时间）

NodejsFirst-master (2) > NodejsFirst-master					
	名称	修改日期	类型	大小	
	.idea	2019/6/11 22:46	文件夹		
	html	2019/6/13 21:13	文件夹		
js	model	2019/6/13 21:15	文件夹		
	node_modules	2018/4/18 16:46	文件夹		
	package-lock	2018/4/18 16:46	文本文档	1 KB	
	register	2018/4/18 16:46	SQL 文件	4 KB	
lanager	web_server	2019/6/13 21:10	JetBrains WebSt...	4 KB	

```

You are now in C:\Users\翁罗轩\Desktop\Project1\Project1
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit
1
Enter the whole path.
C:\Users\翁罗轩\Desktop\NodejsFirst-master (2)\NodejsFirst-master
2019/6/11 13:30 . [folder]
2019/6/11 13:30 .. [folder]
2019/6/11 13:30 .gitattributes [file]
2019/6/11 13:30 .idea [folder]
2019/6/11 13:30 html [folder]
2019/6/11 13:30 model [folder]
2019/6/11 13:30 node_modules [folder]
2019/6/11 13:30 package-lock.json [file]
2019/6/11 13:30 register.sql [file]
2019/6/11 13:30 web_server.js [file]

```

4.2 COPY

复制当前目录下的“源.cpp”文件进入桌面新建的 test 文件夹。

test 文件夹原来为空：



执行程序：

```
You are now in C:\Users\翁罗轩\Desktop\Project1\Project1
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit
2
Enter the file name.
源.cpp
Enter the path where you want to copy this file.
C:\Users\翁罗轩\Desktop\test
Copying successful!
```

复制成功：



4.3 DEL

删除当前目录下新建的 test.txt 文件。

删除前：

<div> <div> </div> <div> <div>此电脑</div> <div>桌面</div> <div>Project1</div> <div>Project1</div> </div> </div>				
	名称	修改日期	类型	大小
<div>ds</div> <div> <div>Project1</div> <div>Project1</div> </div> <div>manager</div>	Debug	2019/3/7 15:47	文件夹	
	x64	2019/4/20 10:48	文件夹	
	Project1.vcxproj	2019/6/15 15:25	VC++ Project	6 KB
	Project1.vcxproj.filters	2019/6/15 20:55	VC++ Project Fil...	0 KB
	Project1.vcxproj.user	2018/11/30 11:21	Per-User Project...	1 KB
	test	2019/6/15 21:08	文本文档	0 KB
	源.cpp	2019/6/15 20:57	C++ Source	7 KB

执行程序：

```

You are now in C:\Users\翁罗轩\Desktop\Project1\Project1
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit
3
Enter the file name.
test.txt
Deleting successful!
  
```

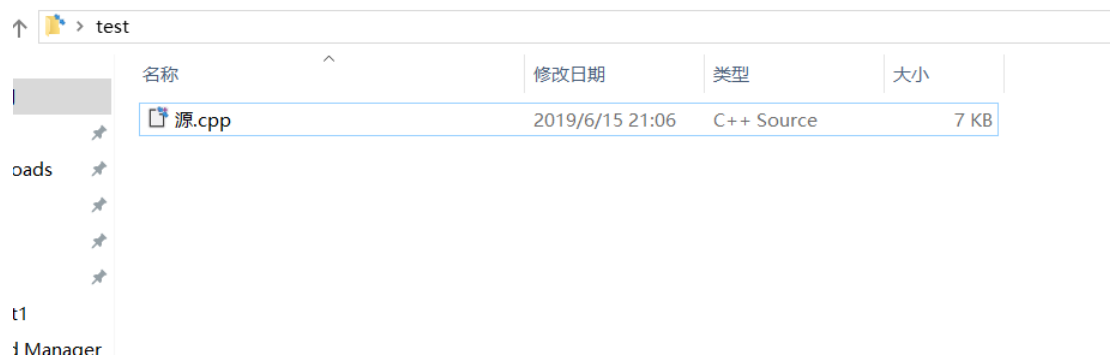
删除成功：

<div> <div> </div> <div> <div>此电脑</div> <div>桌面</div> <div>Project1</div> <div>Project1</div> </div> </div>				
	名称	修改日期	类型	大小
<div>ads</div> <div> <div>Project1</div> <div>Project1</div> </div> <div>Manager</div>	Debug	2019/3/7 15:47	文件夹	
	x64	2019/4/20 10:48	文件夹	
	Project1.vcxproj	2019/6/15 15:25	VC++ Project	6 KB
	Project1.vcxproj.filters	2019/6/15 20:55	VC++ Project Fil...	0 KB
	Project1.vcxproj.user	2018/11/30 11:21	Per-User Project...	1 KB
	源.cpp	2019/6/15 20:57	C++ Source	7 KB

4.4 MD

在桌面的 **test** 文件夹下新建目录 **new**。

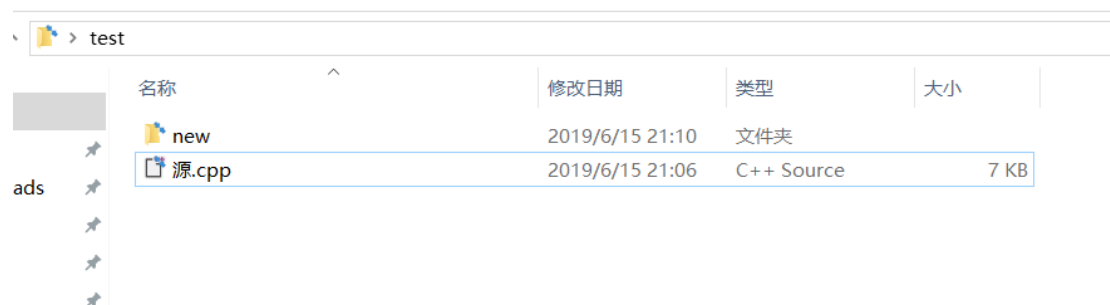
建立之前：



执行程序:

```
You are now in C:\Users\翁罗轩\Desktop\Project1\Project1
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit
4
Enter the folder name.
new
Enter the path where you want to create the folder.
C:\Users\翁罗轩\Desktop\test
Creating successful!
```

建立成功:



4.5 CD

欲访问 C:\Users\翁罗轩\Desktop\test 下的 123.txt 文件, 当前所处目录为 C:\Users\翁罗轩\Desktop\Project1\Project1。

当前目录下不包含 123.txt 文件, 访问失败:

↑ > 此电脑 > 桌面 > Project1 > Project1 >					
	名称	修改日期	类型	大小	
Project1 loads	Debug	2019/3/7 15:47	文件夹		
	x64	2019/4/20 10:48	文件夹		
	Project1.vcxproj	2019/6/15 15:25	VC++ Project	6 KB	
	Project1.vcxproj.filters	2019/6/15 20:55	VC++ Project Fil...	0 KB	
	Project1.vcxproj.user	2018/11/30 11:21	Per-User Project...	1 KB	
源文件	源.cpp	2019/6/15 20:57	C++ Source	7 KB	

```

You are now in C:\Users\翁罗轩\Desktop\Project1\Project1
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit
3
Enter the file name.
123.txt
No such file in this directory!

```

执行程序的 CD 功能，注意程序目前的工作目录已改变：

```

5
Enter the whole path.
C:\Users\翁罗轩\Desktop\test

You are now in C:\Users\翁罗轩\Desktop\test
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit

```

再次试图访问 123.txt 文件，访问成功（说明目录跳转成功）：

```

You are now in C:\Users\翁罗轩\Desktop\test
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit
3
Enter the file name.
123.txt
Deleting successful!

```

4.6 Read Block

尝试对当前目录下的 RecordManager.cpp 文件进行读取操作，运行程序，假设要读取的块号为 6：

```

You are now in C:\Users\翁罗轩\Desktop\Project1\Project1
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit
6
Enter the file name.
RecordManager.cpp
Enter the cluster index.
6
Reading successful, and the content has been stored in ReadBuffer0.

```

查看 ReadBuffer0 文件，发现其内部已写入相应的块的内容：

Project1.vcxproj.user	2018/11/30 11:21	Per-User Project...	1 KB
ReadBuffer0	2019/6/15 22:18	文件	1 KB
RecordManager.cpp	2019/6/15 12:20	C++ S	
源.cpp	2019/6/15 20:57	C++ S	

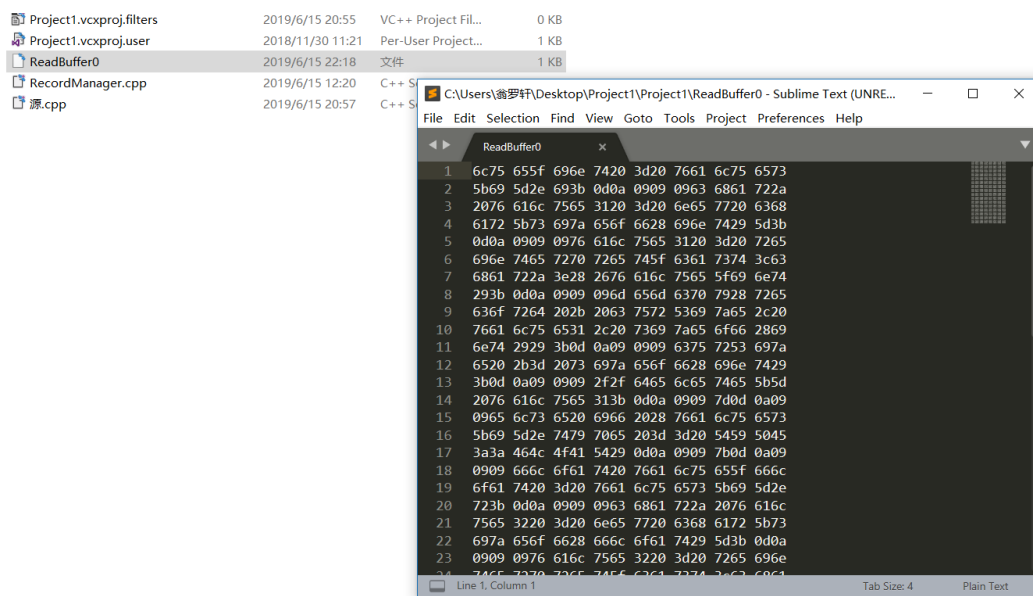
```

C:\Users\翁罗轩\Desktop\Project1\Project1\ReadBuffer0 - Sublime Text (UNRE...
File Edit Selection Find View Goto Tools Project Preferences Help

1 lue_int = values[i].i;
2   char* value1 = new char[sizeof(int)];
3   value1 = reinterpret_cast<char*>(&value_int);
4   memcpy(record + curSize, value1, sizeof(int));
5   curSize += sizeof(int);
6   //delete[] value1;
7 }
8 else if (values[i].type == TYPE::FLOAT)
9 {
10    float value_float = values[i].r;
11    char* value2 = new char[sizeof(float)];
12    value2 = reinterpret_cast<char*>(&value_float);
13    memcpy(record + curSize, value2, sizeof(float));
14    curSize += sizeof(float);
15    //delete[] value2;
16

```

十六进制方式打开文件：

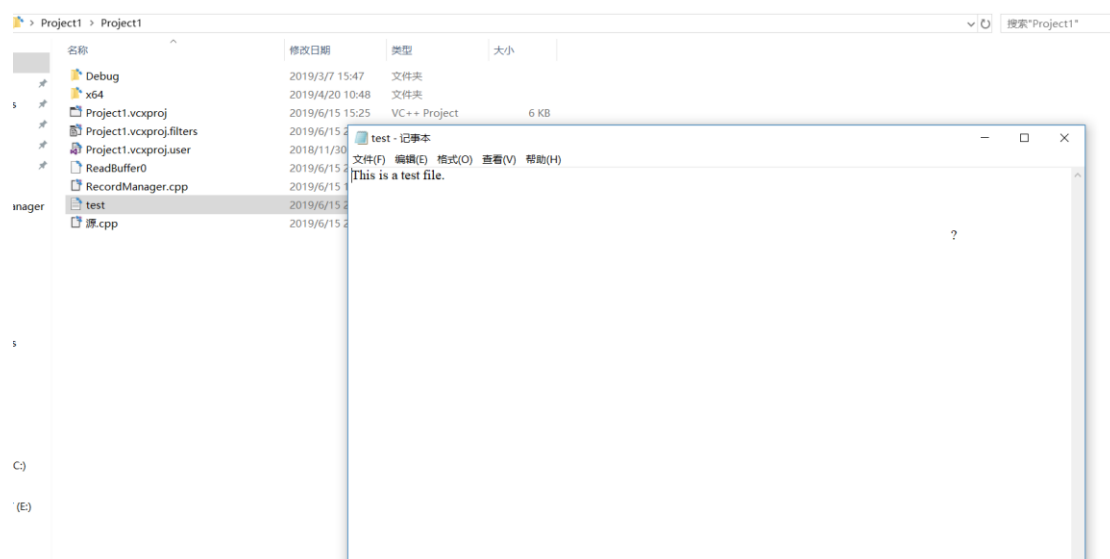


4.7 Write Block

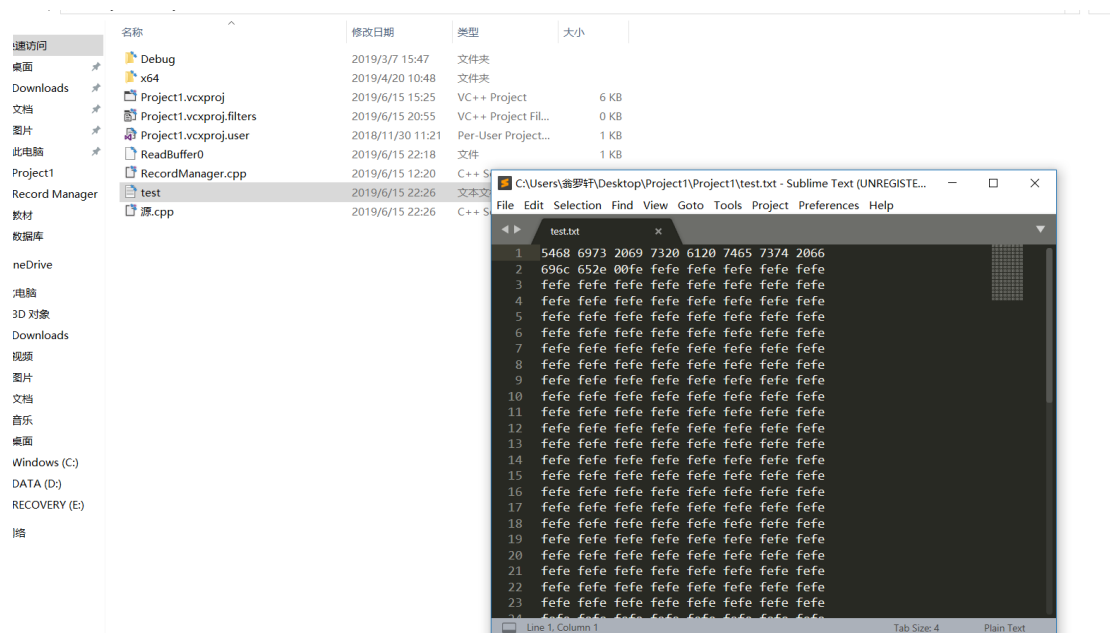
尝试对当前目录下的 `test.txt` 文件进行写入操作，运行程序，假设要写入的块号为 `0`（即文件开头处）：

```
You are now in C:\Users\翁罗轩\Desktop\Project1\Project1
Choose the operation you want to take.
1. DIR
2. COPY
3. DEL
4. MD
5. CD
6. Read Block
7. Write Block
0. Exit
7
Enter the file name.
test.txt
Enter the cluster index.
0
Enter the content you want to write.
This is a test file.
Writing successful!
```

打开 `test.txt` 文件进行检查，发现以上内容已被写入：



十六进制方式打开文件：



由于输入内容的长度可能不是一个块的大小，因此如前所述，程序将会自动写入一些随机数据（乱码）以补足。由于这些乱码皆可以被下一次写入操作覆盖，因此没有影响。