****

实验五 文件复制

班 级： 07111505

姓 名： 徐宇恒

学 号： 1120151839

目录

[一. 实验目的 3](#_Toc512288990)

[二. 实验内容 3](#_Toc512288991)

[三. 实验环境 4](#_Toc512288992)

[四. 实验过程 4](#_Toc512288993)

[4.1. 基本思路 4](#_Toc512288998)

[4.2. Windows调用系统API 4](#_Toc512288999)

[4.2.1. SetFileTime() 4](#_Toc512289000)

[4.2.2. GetFileTime() 4](#_Toc512289001)

[4.2.3. FildFiirstFile() 5](#_Toc512289002)

[4.2.4. FildNextFile() 5](#_Toc512289003)

[4.3. Linux调用系统API 5](#_Toc512289004)

[4.3.1. opendir() 5](#_Toc512289005)

[4.3.2. readdir() 6](#_Toc512289006)

[4.3.3. mkdir() 6](#_Toc512289007)

[4.3.4. stat() 6](#_Toc512289008)

[4.3.5. utime() 6](#_Toc512289009)

[4.3.6. readlink() 6](#_Toc512289010)

[4.3.7. creat() 6](#_Toc512289011)

[4.3.8. write() 6](#_Toc512289012)

[4.3.9. open() 6](#_Toc512289013)

[五. 实验结果 6](#_Toc512289014)

[5.1. Windows实验结果 6](#_Toc512289016)

[5.2. Linux实验结果 8](#_Toc512289017)

[六. 心得体会 9](#_Toc512289018)

# 实验目的

熟悉Windows和Linux环境下的文件操作相关系统API的功能及作用，并尝试完成简单的文件操作，对文件目录进行操作。

# 实验内容

完成一个目录复制命令mycp，包括目录下的文件和子目录，运行结果如下

[beta@bugs.com](mailto:beta@bugs.com) [~/]# ls –l sem total 56

drwxr-xr-x 3 beta beta 4096 Dec 19 02:53 ./

drwxr-xr-x 8 beta beta 4096 Nov 27 08:49 ../

-rw-r--r-- 1 beta beta 128 Nov 27 09:31 Makefile

-rwxr-xr-x 1 beta beta 5705 Nov 27 08:50 consumer\*

-rw-r--r-- 1 beta beta 349 Nov 27 09:30 consumer.c

drwxr-xr-x 2 beta beta 4096 Dec 19 02:53 subdir/ [beta@bugs.com](mailto:beta@bugs.com) [~/]# mycp sem target [beta@bugs.com](mailto:beta@bugs.com) [~/]# ls –l target

total 56

drwxr-xr-x 3 beta beta 4096 Dec 19 02:53 ./

drwxr-xr-x 8 beta beta 4096 Nov 27 08:49 ../

-rw-r--r-- 1 beta beta 128 Nov 27 09:31 Makefile

-rwxr-xr-x 1 beta beta 5705 Nov 27 08:50 consumer\*

说明：

Linux：create，read， write等系统调用

Windows：CreateFile()，ReadFile(), WriteFile(), CloseHandle()等函数。

# 实验环境

|  |  |  |
| --- | --- | --- |
|  | Windows | Linux |
| 操作系统 | Windows10 Pro 64bit | Ubuntu 16.04LTS |
| 编译器 | Visual Studio 2017 IDE | Gcc |

# 实验过程



## 基本思路

对给定的文件夹进行遍历

如果遍历到单一文件，则直接将源文件复制但目标目录下。

如果便利到的是文件夹，那么久子目标目录中创建一个和原文件夹同名的文件夹。并再以遍历刀的文件夹路径作为源目录的参数，将新建的文件夹目录作为目标路径递归调用文件（夹）复制函数。

当源文件（夹）被遍历结束后，程序完成全部文件的复制，结束运行。

## Windows调用系统API

### SetFileTime()

* 功能：设置文件的创建、访问以及上次修改时间

|  |
| --- |
| BOOL  WINAPI  SetFileTime(  \_In\_ HANDLE hFile,//系统文件句柄  \_In\_opt\_ CONST FILETIME \* lpCreationTime,//文件创建时间  \_In\_opt\_ CONST FILETIME \* lpLastAccessTime,//文件上一次访问时间  \_In\_opt\_ CONST FILETIME \* lpLastWriteTime//文件最近一次修改时间  ); |

### GetFileTime()

* 功能：取得指定文件的时间信息

|  |
| --- |
| BOOL WINAPI  GetFileTime(  \_In\_ HANDLE hFile,//系统文件句柄  \_Out\_opt\_ LPFILETIME lpCreationTime,//用于装载文件的创建时间  \_Out\_opt\_ LPFILETIME lpLastAccessTime,//用于装载文件的上一次访问时  \_Out\_opt\_ LPFILETIME lpLastWriteTime//用于装载文件上一次修改时间  ); |

### FildFiirstFile()

|  |
| --- |
| HANDLE WINAPI  FindFirstFileA(  \_In\_ LPCSTR lpFileName, //将搜索的文件名  \_Out\_ LPWIN32\_FIND\_DATAA lpFindFileData//用于装载与找到的文件有关的信息的结构体  ); |

* 功能：根据文件名查找文件

### FildNextFile()

* 功能：根据调用FindFirstFile函数是制定的一个文件名查找下一个文件

|  |
| --- |
| BOOL WINAPI  FindNextFileA(  \_In\_ HANDLE hFindFile,//上一次查找返回的句柄  \_Out\_ LPWIN32\_FIND\_DATAA lpFindFileData//用于装载与找到的文件有关的信息的结构体  ); |

## Linux调用系统API

### opendir()

* 功能：用来打开参数 name 指定的目录, 并返回 DIR\*形态的目录流。

### readdir()

* 功能：readdir 函数返回一个指向 dirent 结构体的指针，该结构体代表了由 dir 指向的目录流中的下一个目录项；如果读到 end-of-file 或者出现了错误，那么返回NULL。

### mkdir()

* 功能：建所输入的新的目录，返回 0 表示成功，返回 -1 表示错误， 并且会设置 errno 值

### stat()

* 功能：通过文件名 filename 获取文件信息，并保存在 buf 所指的结构体中

### utime()

* 功能：用来修改参数 filename 文件所属的 inode 存取时间

### readlink()

* 功能：将参数 path 的符号连接内容存到参数 buf 所指的内存空间, 返回的内容不是以 NULL 作字符串结尾, 但会将字符串的字符数返回。若参数bufsiz 小于符号连接的内容长度, 过长的内容会被截断。

### creat()

* 功能：使用 creat 函数创建一个新文件，如果原来该文件存在，会将这个文件的长度截短为 0

### write()

* 功能：调用 write 函数向打开的文件写入数据

### open()

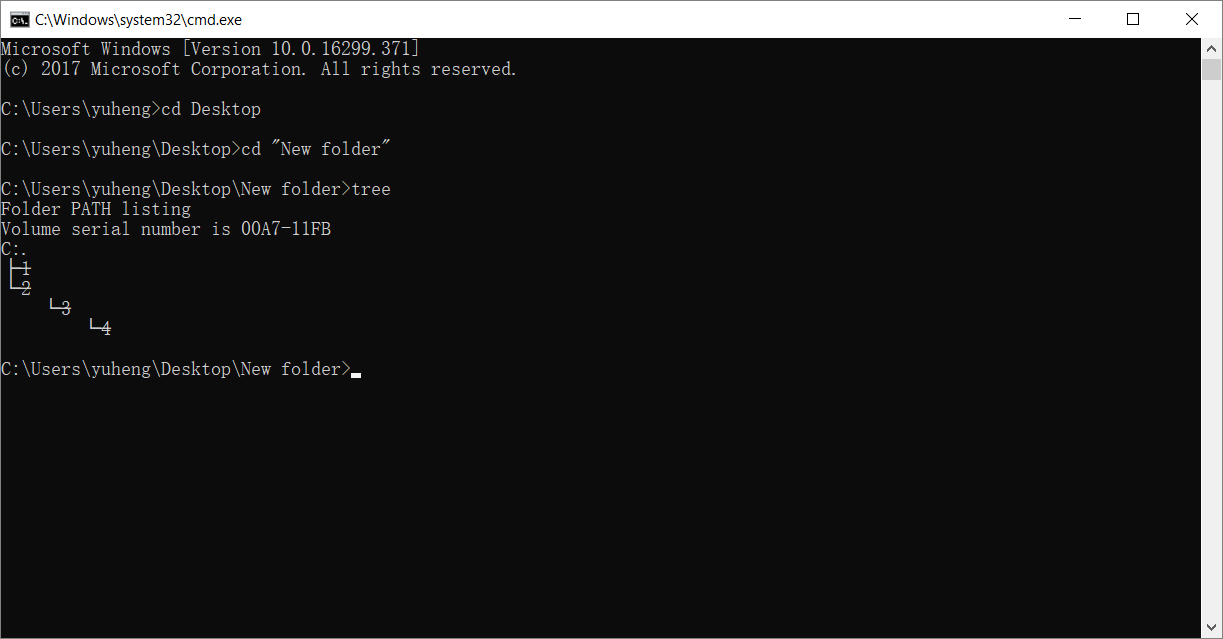
* 功能：调用 open 函数打开或者创建一个文件

# 实验结果

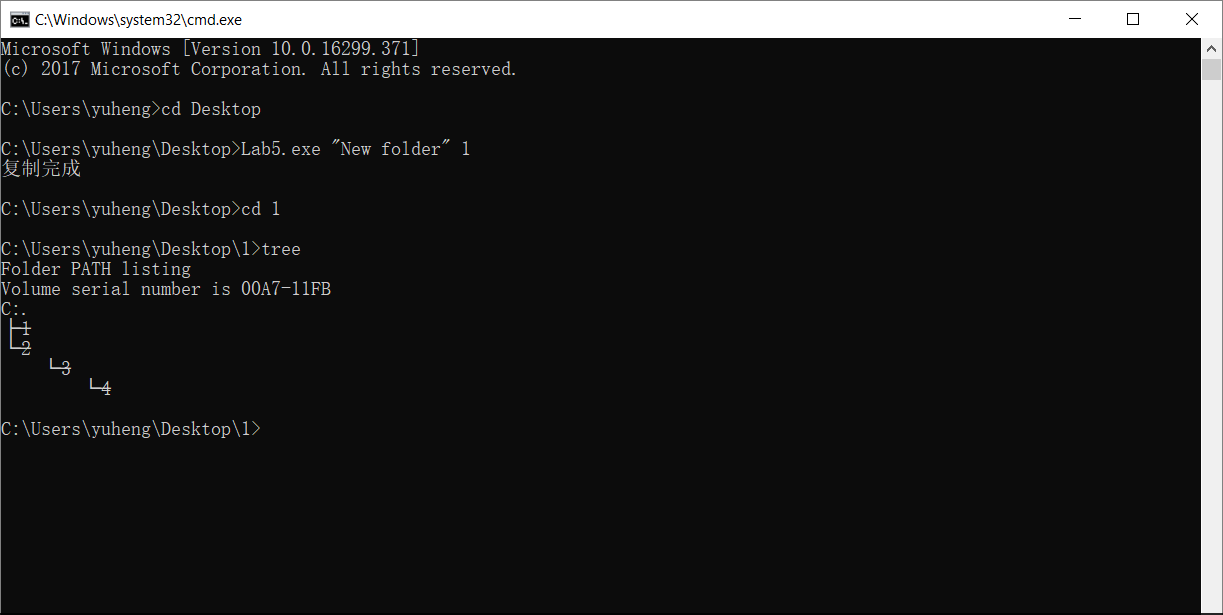


## Windows实验结果

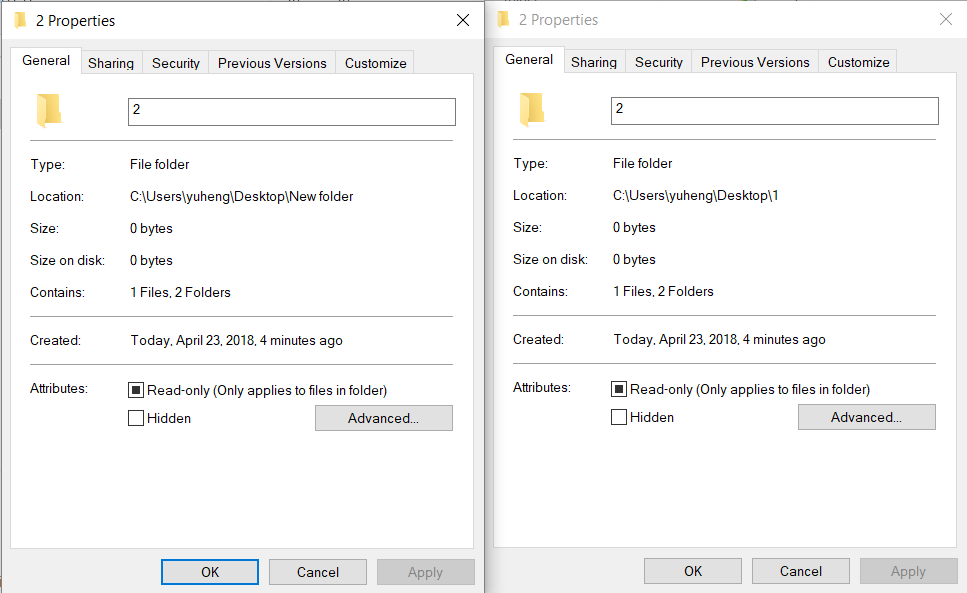
源文件路径结构如下



输入Lab5.exe 源文件 目标文件

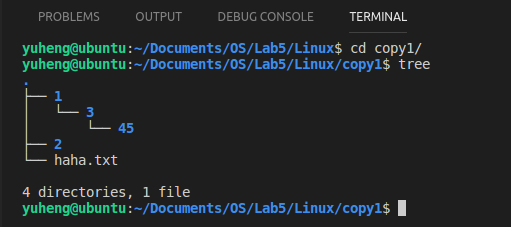


复制时保持文件时间一直



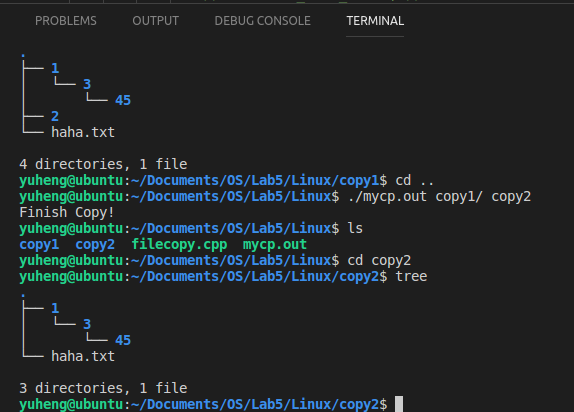
## Linux实验结果

源文件路径结构如下



输入 ./mycp.out 源文件 目标文件。

若没有该目标路径则先创建该路径



# 心得体会

进行了本实验后我进一步了解了 Windows 和 Linux 中系统调用函数的应用，接触到了很多的系统对文件的操作。并且在实验过程中，通过调试各个 API，逐渐了解到系统预定义的很多结构体以及函数调用是真的非常方便。