



操作系统实验报告

实验五 复制文件



2019-4-2

北京理工大学 计算机学院

谭超 1120161874

操作系统课程设计实验报告

实验名称: _____ 复制文件

姓名/学号: _____ 谭超 1120161874

一、 实验目的

完成 Windows 下和 Linux 下的目录复制命令 mycopy, 要求可以实现该目录下目录和文件的复制。通过实现该命令熟悉 Windows 和 Linux 系统的文件和目录的实现机制, 熟悉相关的 API。

二、 实验内容

Windows:

通过调用 CreateFile, ReadFile(),WriteFile(),CloseHandle()等系统 API 实现目录复制命令 mycopy。

Linux:

通过调用 create,read,write 等 API 实现目录复制命令 mycopy。
要求复制后权限和时间属性都和源文件相同。

三、 实验环境

Windows:

Windows10 1803

处理器 Intel Core i5-6300HQ 2.3GHz ,RAM 8G

VisualStudio2017

Linux:

VMware Workstation 14.1.3

Ubuntu 4.13.0.36

gcc 5.4.0

四、 程序设计与实现

Windows:

mycopy 程序由三个函数组成, 分别是主函数 main, 目录复制函数 mycp(), 文件复制函数 CopyFile()。主函数中首先判断命令行参数个数是否正确, 命令行

应该有三个参数，函数本身、源目录、目标目录。如果参数正确，则查找源文件和目标文件路径的文件，如果源文件目录不存在，退出程序，复制失败，如果目标文件目录不存在，创建一个目标文件目录。然后进行复制。

```
if (argc != 3)
{
    printf("参数错误!\n");
}
else
{
    if (FindFirstFile(argv[1], &lpfindfiledata) == INVALID_HANDLE_VALUE)
    {
        printf("查找源文件路径失败!\n");
        exit(0);
    }
    if (FindFirstFile(argv[2], &lpfindfiledata) == INVALID_HANDLE_VALUE)
    {
        CreateDirectory(argv[2], NULL); //为目标文件创建目录
    }
    mycp(argv[1], argv[2]);
    printf("复制完成!\n");
}
```

在目录复制函数 mycp() 中，利用 FindNextFile() 函数遍历源路径中所有文件或目录

```
while (FindNextFile(hfind, &lpfindfiledata) != 0)
```

如果是一个目录，且不是当前目录或父目录，则通过 CreateDirectory() 在目标路径中创建一个对应的目录，然后递归调用 mycp() 函数

```
CreateDirectory(target, NULL); //为目标文件创建目录
mycp(source, target); //进入子目录复制
```

如果是一个文件，则调用文件复制函数 copyFile() 复制文件。

```
copyFile(source, target); //直接调用文件复制函数
```

在文件复制函数 copyFile() 函数中，通过 CreateFile() 打开源文件，创建目标文件

```
//打开源文件
HANDLE hsource = CreateFile(fsource, GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ, NULL, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
//创建目标文件
HANDLE htarget = CreateFile(ftarget, GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ, NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
```

然后用 ReadFile() 函数从源文件中读入数据，用 WriteFile() 函数写入到目标文件中。

```
//源文件读数据
ReadFile(hsource, BUFFER, size, &wordbit, NULL);
//目标文件写数据
WriteFile(htarget, BUFFER, size, &wordbit, NULL);
```

在目录复制和文件复制函数中，通过 SetFileTime() 函数修改文件和目录的时间属

性。

实验结果:

```
D:\SourceCode\C++SourceCode\VS2017\操作系统实验>mycopy.exe test testresult  
复制完成!
```

```
PS D:\SourceCode\C++SourceCode\VS2017\操作系统实验\test> ls -l ./

    目录: D:\SourceCode\C++SourceCode\VS2017\操作系统实验\test

Mode                LastWriteTime         Length Name
----                -
d-----          2019/4/2         14:54      新建文件夹
-a-----          2019/4/10         16:42           3 a.txt

PS D:\SourceCode\C++SourceCode\VS2017\操作系统实验\test> cd ..
PS D:\SourceCode\C++SourceCode\VS2017\操作系统实验> cd .\testresult\
PS D:\SourceCode\C++SourceCode\VS2017\操作系统实验\testresult> ls -l ./

    目录: D:\SourceCode\C++SourceCode\VS2017\操作系统实验\testresult

Mode                LastWriteTime         Length Name
----                -
d-----          2019/4/2         14:54      新建文件夹
-a-----          2019/4/10         16:42           3 a.txt
```

Linux:

mycopy 程序由三个函数组成,主函数 main, 目录复制函数 mycp(), 文件复制函数 copyFile()。在主函数中, 同样的先判断命令行参数是否正确, 如果正确则打开源目录和目标目录, 如果源目录打开失败则退出程序, 如果目标文件打开失败则调用 mkdir()函数创建一个目标目录。并将源目录的时间属性赋值给目标目录的时间属性。

```
mkdir(argv[2],statbuf.st_mode);//创建目录
timeby.actime = statbuf.st_atime;//文件数据的最后存取时间
timeby.modtime = statbuf.st_mtime;//文件数据的最后修改时间
utime(argv[2],&timeby);//用utime()函数修改目标文件的访问时间和修改时间
```

在目录复制函数 mycp()中, 首先通过 opendir()打开源目录, 通过 opendir()打开目标文件, 如果目标文件不存在, 则创建目标文件。然后通过 readdir()遍历源目录中的目录与文件, 如果是目录且不是当前目录或父目录, 然后递归调用 mycp()函数实现目录的复制。

如果是一个文件，则调用 `copyFile()` 实现文件的复制。

如果是一个符号链接文件即 `dirent` 结构体中 `d_type=10`，则通过 `readlink()` 读出该符号链接的链接地址，然后用 `symlink()` 将其链接到目标文件。并通过 `lutimes()` 将源文件的时间属性赋值给目标文件。

```
else if(entry->d_type==10) //如果是符号链接
{
    stat(source,&statbuff);
    char buff[buf_size];
    int length=readlink(source,buff,MAX_PATH);
    //printf("link path:%s\n",buff);
    if(symlink(buff,target)==-1)
    {
        printf("软连接失败\n");
    }
    //timeby.actime = statbuff.st_atime;
    //timeby.modtime = statbuff.st_mtime;
    //if(utime(target,&timeby)<0)
    //printf("link utime failed\n");
    struct stat statbuff;
    lstat(source,&statbuff);
    struct timeval times[2];
    times[0].tv_sec=statbuff.st_atime;
    times[0].tv_usec=0;
    times[1].tv_sec=statbuff.st_mtime;
    times[1].tv_usec=0;
    lutimes(target,times);
}
```

在 `copyFile()` 函数中，通过 `open()` 打开源文件，通过 `create()` 创建目标文件，然后通过 `read()` 读出源文件中的数据，再通过 `write()` 写到目标文件中。最后通过 `utime()` 将源文件的时间属性赋值给目标文件的时间属性。

实验结果：

```
new@ubuntu:~/test1/OS_experiments/experiment5$ ./mycopy test testresult/
复制完成！
```

```

new@ubuntu:~/test1/OS_experiments/experiment5/testresult$ ls -la
总用量 24
drwxrwxr-x 4 new new 4096 Apr  5 13:30 .
drwxrwxr-x 4 new new 4096 Apr 10 16:41 ..
-rw-rw-r-- 1 new new  10 Apr  2 15:58 a
drwxrwxr-x 2 new new 4096 Apr  5 10:00 b
drwxrwxr-x 2 new new 4096 Apr  5 10:00 d
-rw-rw-r-- 1 new new  4 Apr  5 10:00 f
lrwxrwxrwx 1 new new  42 Apr  5 13:30 到 experiment2 的链接 -> /home/new/test1/
OS_experiments/experiment2
new@ubuntu:~/test1/OS_experiments/experiment5/testresult$ cd ..
new@ubuntu:~/test1/OS_experiments/experiment5$ cd test
new@ubuntu:~/test1/OS_experiments/experiment5/test$ ls -la
总用量 24
drwxrwxr-x 4 new new 4096 Apr  5 13:30 .
drwxrwxr-x 4 new new 4096 Apr 10 16:41 ..
-rw-rw-r-- 1 new new  10 Apr  2 15:58 a
drwxrwxr-x 2 new new 4096 Apr  5 10:00 b
drwxrwxr-x 2 new new 4096 Apr  5 10:00 d
-rw-rw-r-- 1 new new  4 Apr  5 10:00 f
lrwxrwxrwx 1 new new  42 Apr  5 13:30 到 experiment2 的链接 -> /home/new/test1/
OS_experiments/experiment2
new@ubuntu:~/test1/OS_experiments/experiment5/test$ █

```

五、实验收获与体会

在本次实验中，我主要熟悉了 Windows 和 Linux 下文件和目录的存储机制，同时对于文件和目录操作的 API 有了大致的了解。另外对于 Linux 系统中的符号链接 `readlink()`、`symlink()` 等操作更加熟悉。理解了 `dirent` 结构中各个成员特别是 `d_type` 的各个值所代表的含义。