



《操作系统》课第二次实验报告

学院:	软件学院
姓名:	杨万里
学号:	2013774
邮箱:	2013774@mail.nankai.edu.cn
时间:	2022/9/24

0. 开篇感言

和虚拟机不熟？那就多用虚拟机写代码、跑程序吧！

Practice makes perfect.

1. 实验题目

编写一个 C/C++ 程序实现目录拷贝

2. 实验目标

- (1) 编写 C/C++ 程序
- (2) 实现拷贝一个目录以及它的所有子目录
- (3) 使用 GCC 编译
- (4) 使用 IDE 集成开发环境
- (5) 测试目录（从 www.kernel.org 下载最新的 linux 内核 linux-5.19.10.tar.xz）
 - a) 下载压缩包



- b) 解压缩为 linux-5.19.10 文件夹
 - c) 拷贝该文件夹到 linux-5.19.10back 文件夹
- (6) 验证文件夹副本是正确的

3. 原理方法

- (1) 拷贝文件夹的原理方法:

编写拷贝文件夹的函数 readFileList, 输入需要拷贝的文件夹路径 sourcePath 和目标文件夹路径 targetPath。

首先判断源文件夹是否存在, 若存在则循环读取其中的子文件/子文件夹, 在源文件夹和目标文件夹的路径上加上子文件/子文件夹, 通过 ptr->d_type 判断类型, 如果是文件夹, 则在目标文件夹下创建文件夹并 **递归调用 readFileList**; 如果是链接文件, 则调用 link (source,target); 如果是普通文件则调用自己写的 copyfile 函数, 实现文件的拷贝。

(代码如下图所示)

```
int readFileList(char* sourcePath, char* targetPath){
    DIR* dir;
    struct dirent * ptr;
    char source[1000];
    char target[1000];
    // wrong path
    if((dir=opendir(sourcePath)) == NULL){
        perror("Open dir error");
        exit(1);
    }
    while((ptr=readdir(dir)) != NULL){
        //printf(ptr->d_name);
        //printf("\n");
        if(strcmp(ptr->d_name, ".")==0 || strcmp(ptr->d_name, "..")==0)    ///current dir OR parrent dir
            continue;
        //char arr set 0
        memset(target, '\0', sizeof(targetPath));
        memset(source, '\0', sizeof(source));
        strcpy(target, targetPath);
        strcpy(source, sourcePath);
        printf(targetPath);
        printf(" ");
        printf(sourcePath);
        printf("\n");
        strcat(target, "/");
        strcat(source, "/");
        strcat(target, ptr->d_name);
        strcat(source, ptr->d_name);
        if(ptr->d_type == DT_DIR)    ///dir
        {
            int i=mkdir(target, S_IRWXU|S_IRGRP|S_IXGRP|S_IROTH|S_IXOTH);
            readFileList(source, target);
        }
        else if(ptr->d_type == DT_LNK)    ///link
        {
            link (source, target);
        }
    }
}
```



```
}
else if(ptr->d_type == DT_REG){// file
    copyfile(source,target);
}
else{
}
}
closedir(dir);
return 1;
}
```

(2) 拷贝文件的原理方法:

按字节逐个拷贝即可。

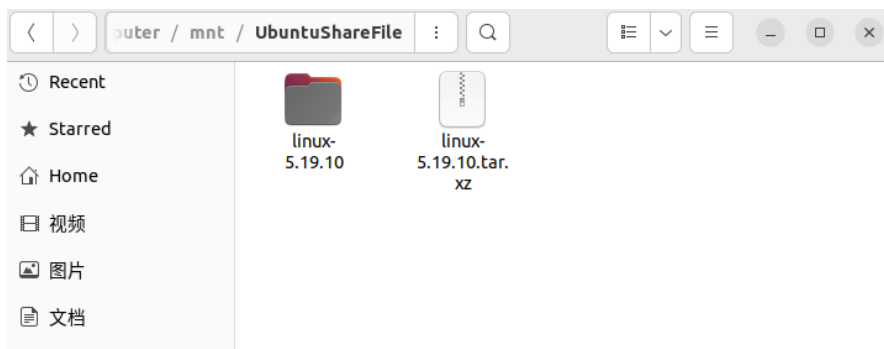
```
int copyfile(const char* dest, const char* src) {
    FILE* fp1=NULL, * fp2=NULL;
    fp1 = fopen(dest, "r");
    if (fp1 == NULL) {
        return -1;
    }
    //fopen_s(&fp2,src,"w");
    fp2 = fopen(src, "w");
    if (fp2 == NULL) {
        return -2;
    }
    char buffer[BUFSIZE];
    int readlen, writelen;
    while ((readlen = fread(buffer, 1, BUFSIZE, fp1))>0)
    {
        writelen = fwrite(buffer, 1, readlen, fp2);
        if (readlen != writelen) {
            return -3;
        }
    }
    fclose(fp1);
    fclose(fp2);
    return 0;
}
```

4. 具体步骤

- (1) 安装 GCC (上次实验已经完成)
- (2) 安装 IDE, 使用 Code::Blocks, 不用 JetBrains CLion

```
wanliyang2013774@wanliyang2013774-virtual-machine:~/桌面$ sudo apt-get install codeblocks
[sudo] password for wanliyang2013774:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
将会同时安装下列软件:
  codeblocks-common libastyle3 libcodeblocks0 libtinyxml2.6.2v5 libutempter0 libwxbase3.0-0v5 libwxgtk3.0-gtk3-0v5 xterm
建议安装:
  codeblocks-contrib libwxgtk3.0-dev xfonts-cyrillic
下列【新】软件包将被安装:
  codeblocks codeblocks-common libastyle3 libcodeblocks0 libtinyxml2.6.2v5 libutempter0 libwxbase3.0-0v5
  libwxgtk3.0-gtk3-0v5 xterm
升级了 0 个软件包, 新安装了 9 个软件包, 要卸载 0 个软件包, 有 14 个软件包未被升级。
需要下载 13.8 MB 的归档。
解压后会消耗 42.4 MB 的额外空间。
您希望继续执行吗? [Y/n] y
```

- (3) 下载并解压 Linux 内核, 文件夹名称为 Linux-5.19.10



(4) 编写程序实现拷贝

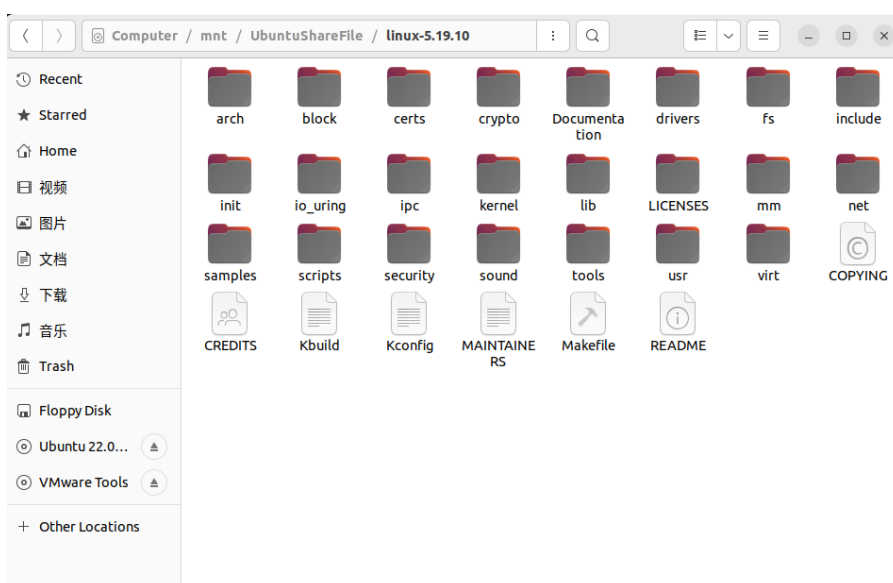
基本原理方法在上一小节已经解释。

拷贝结果截图：（命令行显示程序运行时间）

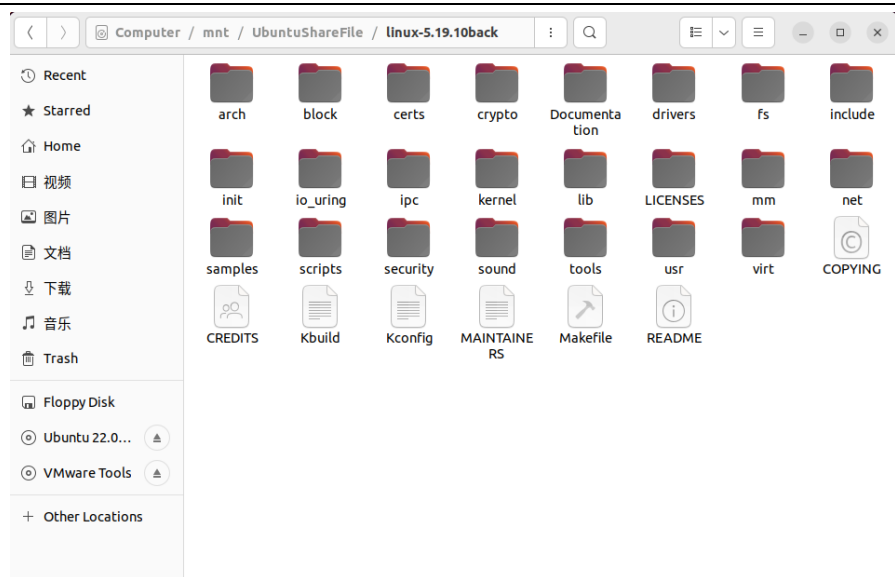
```
/mnt/UbuntuShareFile/linux-5.19.10back/virt/kvm /mnt/UbuntuShareFile/linux-5.19.10/virt/kvm
/mnt/UbuntuShareFile/linux-5.19.10back/virt/kvm /mnt/UbuntuShareFile/linux-5.19.10/virt/kvm
/mnt/UbuntuShareFile/linux-5.19.10back/virt/kvm /mnt/UbuntuShareFile/linux-5.19.10/virt/kvm
/mnt/UbuntuShareFile/linux-5.19.10back/virt /mnt/UbuntuShareFile/linux-5.19.10/virt
/mnt/UbuntuShareFile/linux-5.19.10back/virt/lib /mnt/UbuntuShareFile/linux-5.19.10/virt/lib
/mnt/UbuntuShareFile/linux-5.19.10back/virt/lib /mnt/UbuntuShareFile/linux-5.19.10/virt/lib
/mnt/UbuntuShareFile/linux-5.19.10back/virt/lib /mnt/UbuntuShareFile/linux-5.19.10/virt/lib
/mnt/UbuntuShareFile/linux-5.19.10back/virt /mnt/UbuntuShareFile/linux-5.19.10/virt

real    54m11.980s
user    0m12.377s
sys     3m26.233s
```

Linux-5.19.10（截图展示一级子目录，不包含更低级的子目录，所有子目录都完成了拷贝，不一一截图对比，下文用命令检验是否正确）



Linux-5.19.10back



(5) 验证正确性

```
wanliyang2013774@wanliyang2013774-virtual-machine:/mnt/UbuntuShareFile$ diff -r linux-5.19.10 linux-5.19.10back
wanliyang2013774@wanliyang2013774-virtual-machine:/mnt/UbuntuShareFile$
```

diff 指令没有找到不同的文件，说明两个文件夹完全相同，实验成功！

5. 总结心得

- (1) 拷贝一个文件夹及其所有文件夹主要运用的知识点是文件输入输出流以及递归思想。
- (2) 其中需要注意的是对于一个文件需要判断是文件夹还是链接文件还是常规文件，不同的类型对应不同的处理方式。
- (3) 验证两个文件夹是否相等的底层思路是 MD5 算法，利用哈希的思想，将数据映射为定长的编码。
- (4) 通过本次实验，加深了对虚拟机的了解。

6. 参考资料

实验指导手册《Lab02CopyDirV1_README.pdf》



项目源代码:

```
Open ▾ [icon] *main.c
~/MyWorkspace/OsLab2/copyDir

1#include <stdio.h>
2#include <stdlib.h>
3#include <string.h>
4#include <dirent.h>
5#include <unistd.h>
6#include <sys/types.h>
7#include <sys/stat.h>
8#include <fcntl.h>
9#define BUFSIZE 1024
10
11int copyfile(const char* dest, const char* src) {
12    FILE* fp1=NULL, * fp2=NULL;
13    fp1 = fopen(dest, "r");
14    if (fp1 == NULL) {
15        return -1;
16    }
17    //fopen_s(&fp2,src,"w");
18    fp2 = fopen(src, "w");
19    if (fp2 == NULL) {
20        return -2;
21    }
22    char buffer[BUFSIZE];
23    int readlen, writelen;
24    while ((readlen = fread(buffer, 1, BUFSIZE, fp1))>0)
25    {
26        writelen = fwrite(buffer, 1, readlen, fp2);
27        if (readlen != writelen) {
28            return -3;
29        }
30    }
31    fclose(fp1);
32    fclose(fp2);
33    return 0;
34}
35
36int readFileList(char* sourcePath, char* targetPath){
37    DIR* dir;
38    struct dirent * ptr;
39    char source[1000];
40    char target[1000];
41    // wrong path
42    if((dir=opendir(sourcePath)) == NULL){
43        perror("Open dir error");
44        exit(1);
45    }
46    while((ptr=readdir(dir)) != NULL){
47        //printf(ptr->d_name);
48        //printf("\n");
49        if(strcmp(ptr->d_name,".")==0 || strcmp(ptr->d_name,"..")==0)    //current dir OR parrent dir
50            continue;
51        //char arr set 0
52        memset(target,'\0',sizeof(targetPath));
53        memset(source,'\0',sizeof(source));
54        strcpy(target,targetPath);
55        strcpy(source,sourcePath);
56        printf(targetPath);
57        printf(" ");
58        printf(sourcePath);
59        printf("\n");
60        strcat(target,"/");
61        strcat(source,"/");
62        strcat(target,ptr->d_name);
63        strcat(source,ptr->d_name);
64        if(ptr->d_type == DT_DIR)    //dir
65        {
66            int i=mkdir(target,S_IRWXU|S_IRGRP|S_IXGRP|S_IROTH|S_IXOTH);
67            readFileList(source,target);
68        }
69        else if(ptr->d_type == DT_LNK)    //link
70        {
71            link (source,target);
72        }
73    }
74}
```



```
72     }
73     else if(ptr->d_type == DT_REG){// file
74         copyfile(source,target);
75     }
76     else{
77
78     }
79 }
80 closedir(dir);
81 return 1;
82 }
83
84 int main(void)
85 {
86     char* sourcePath="/mnt/UbuntuShareFile/linux-5.19.10";
87     char* targetPath="/mnt/UbuntuShareFile/linux-5.19.10back";
88     readFileList(sourcePath,targetPath);
89     //verify the result
90
91     return 0;
92 }
93
```