# 实验十：文件系统编程

## 实验内容:

编写一个程序，实现将一个目录的所有内容复制到另一个目录的功能。要求：

源文件（目录）和目标文件（目录）的属主、权限等信息保持一致；

每复制一个文件（目录），在屏幕提示相应信息；

当遇到符号链接文件时，显示该文件为链接文件，不复制。

答：

```c
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <dirent.h>

#define OK 0    // 成功
#define INPUT_ERROR 1   // 输入参数有误
#define FILE_EXIST 2    // 文件已存在
#define COPY_ERROR 3    // 复制失败
#define NOT_A_DIRECTORY 2   // 源文件不是目录
#define BUFF_SIZE 1024  // 复制文件的缓冲区大小
#define PATH_LEN 100    // 路径占的字节数最大值
#define OWN_CHANG_ERROR 01  // 属主更改失败
#define MOD_CHANG_ERROR 010 // 权限更改失败
#define FOLDER_CREAT_ERROR 1    // 文件夹创建失败

char* getFileName(char* path);
int copyFile(char* dst, char* src);
int changeAttr(char* filename, struct stat* stat);
int copyFolder(char* dst, char* src);
int copy(char* dst, char* src);

int main(int argc, char *argv[]) {
    int fd; // open 函数的返回值
```

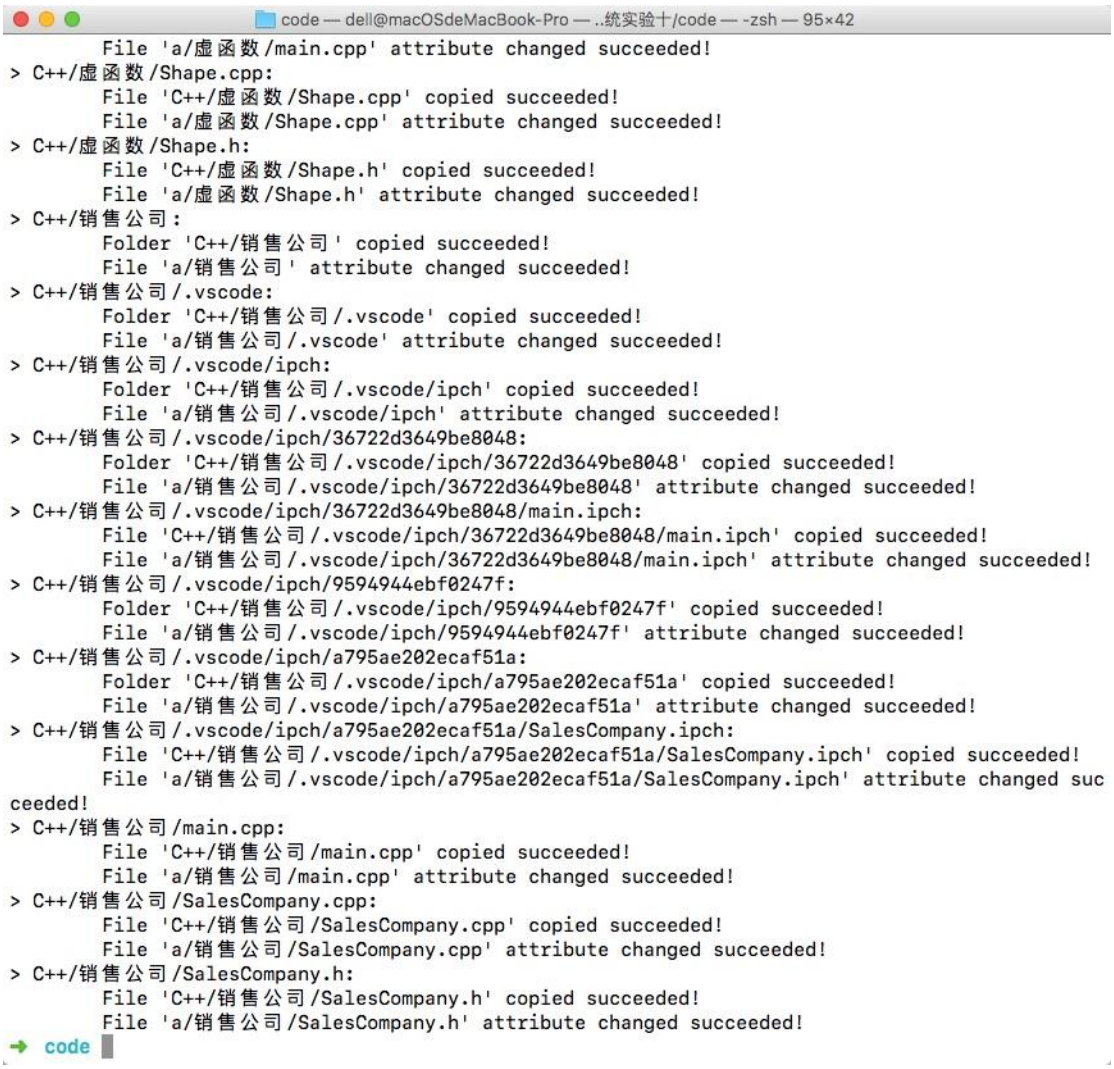
```
28      char* path[2];  // path[0]-srcfile,
    path[1]-dstfile
29      if (argc < 3) {
30          printf(" - Usage: %s srcfile dstfile\n",
    getFileName(argv[0]));
31          return INPUT_ERROR;
32      }
33      path[0] = argv[1];
34      path[1] = argv[2];
35      // 检测源文件是否目录
36      fd = open(path[0], O_DIRECTORY);
37      if (fd < 0) {   // 源文件不是目录
38          printf("ERROR: Source file is not a
    directory!");
39          return NOT_A_DIRECTORY;
40      }
41      // 检测目标文件
42      if (access(path[1], F_OK) != -1) {  // 目标文件已存
    在
43          printf("ERROR: File already exist!");
44          return FILE_EXIST;
45      }
46      copyFolder(path[1], path[0]);    // 创建目录
47      copy(path[1], path[0]);
48      return OK;
49  }
50  // 复制完整目录(包含里面的文件)
51  int copy(char* dst, char* src) {
52      int fd1, fd2;
53      struct dirent* ptr;
54      DIR* dir = opendir(src);
55      char srcpath[PATH_LEN], dstpath[PATH_LEN];
56      while ((ptr = readdir(dir)) != NULL)
57      {
58          // 跳过.和..目录
59          if (strcmp(ptr->d_name, ".") == 0 ||
    strcmp(ptr->d_name, "..") == 0)
60              continue;
```

```
61          // 生成完整路径
62          strcpy(srcpath, src);
63          strcpy(dstpath, dst);
64          if (*(srcpath + strlen(srcpath) - 1) != '/')
65              strcat(srcpath, "/");
66          strcat(srcpath, ptr->d_name);
67          if (*(dstpath + strlen(dstpath) - 1) != '/')
68              strcat(dstpath, "/");
69          strcat(dstpath, ptr->d_name);
70          // 判断文件类型
71          fd1 = open(srcpath, O_DIRECTORY);
72          if (fd1 < 0) {   // 源文件不是目录
73              fd2 = open(srcpath, O_NOFOLLOW);
74              if (fd2 < 0)    // 源文件是符号链接
75                  printf("Message: File '%s' is a soft
    link!");
76              else    // 源文件是普通文件
77                  if (copyFile(dstpath, srcpath) != OK) {
78                      printf("ERROR: File '%s' copy
    error!", srcpath);
79                      return COPY_ERROR;
80                  }
81              close(fd2);
82          } else {    // 源文件是目录
83              if (copyFolder(dstpath, srcpath) != OK) {
84                  printf("ERROR: Directory '%s' copy
    error!", srcpath);
85                  return COPY_ERROR;
86              }
87              if (copy(dstpath, srcpath) != OK) { // 递
    归复制
88                  return COPY_ERROR;
89              }
90          }
91          close(fd1);
92      }
93      return OK;
94  }
```

```
95  // 复制文件夹(不包含里面的文件)
96  int copyFolder(char* dst, char* src) {
97      int fd;
98      struct stat srcstat;
99      fd = open(src, O_NOFOLLOW | O_DIRECTORY);
100     fstat(fd, &srcstat);
101     close(fd);
102     // 创建文件夹
103     if (mkdir(dst, srcstat.st_mode) == -1) {
104         printf("> %s:\n", getFileName(src));
105         printf("\tERROR: Folder '%s' copy failed!\n",
    src);
106         return FOLDER_CREAT_ERROR;
107     }
108     // 设置文件夹权限
109     printf("> %s:\n", src);
110     printf("\tFolder '%s' copied succeeded!\n", src);
111     return changeAttr(dst, &srcstat);
112 }
113 // 复制文件
114 int copyFile(char *dst, char *src)
115 {
116     struct stat srcstat;
117     char buff[BUFF_SIZE];
118     int len, fd1, fd2;
119     fd1 = open(src, O_RDONLY);
120     fd2 = open(dst, O_WRONLY | O_CREAT);
121     fstat(fd1, &srcstat);   // 记录源文件的属性
122     while ((len = read(fd1, buff, BUFF_SIZE)))
123         write(fd2, buff, len);
124     close(fd1);
125     close(fd2);
126     printf("> %s:\n", src);
127     printf("\tFile '%s' copied succeeded!\n", src);
128     changeAttr(dst, &srcstat);   // 修改复制后的文件属性
129     return OK;
130 }
131 // 更改属主、权限等属性，返回值为 OK、MOD_CHANG_ERROR、
```

```
    OWN_CHANG_ERROR 的组合
132 int changeAttr(char *filename, struct stat* stat) {
133     int own, mod, ret = OK;
134     own = chown(filename, stat->st_uid, stat->st_gid);
    // 更改文件属主
135     mod = chmod(filename, stat->st_mode);   // 更改文
    件权限
136     if (own == -1) {    // 属主更改失败
137         printf("\tERROR: File '%s' owner failed to
    change!\n", filename);
138         ret |= OWN_CHANG_ERROR;
139     }
140     if (mod == -1) {    // 权限更改失败
141         printf("\tERROR: File '%s' mode failed to
    change!\n", filename);
142         ret |= MOD_CHANG_ERROR;
143     }
144     if (mod == OK) {    // 更改成功
145         printf("\tFile '%s' attribute changed
    succeeded!\n", filename);
146     }
147     return ret;
148 }
149 // 获取路径中的文件名
150 char* getFileName(char* path) {
151     char* p = path + strlen(path) - 1;
152     if (p > path && *p == '/')
153         *p = '\0';  // 除根目录外，路径不以'/'结尾
154     while (p > path && *(p - 1) != '/')
155         p--;
156     return p;
157 }
```


code — dell@macOSdeMacBook-Pro — ..统实验十/code — -zsh — 80×24
➜ code gcc copy.c -o copy
➜ code ./copy ./dir1 a
> ./dir1:
Folder './dir1' copied succeeded!
File 'a' attribute changed succeeded!
> ./dir1/1.txt:
File './dir1/1.txt' copied succeeded!
File 'a/1.txt' attribute changed succeeded!
➜ code


code — dell@macOSdeMacBook-Pro — ..统实验十/code — -zsh — 95×42
File 'a/虚函数/main.cpp' attribute changed succeeded!
> C++/虚函数/Shape.cpp:
File 'C++/虚函数/Shape.cpp' copied succeeded!
File 'a/虚函数/Shape.cpp' attribute changed succeeded!
> C++/虚函数/Shape.h:
File 'C++/虚函数/Shape.h' copied succeeded!
File 'a/虚函数/Shape.h' attribute changed succeeded!
> C++/销售公司:
Folder 'C++/销售公司' copied succeeded!
File 'a/销售公司' attribute changed succeeded!
> C++/销售公司/.vscode:
Folder 'C++/销售公司/.vscode' copied succeeded!
File 'a/销售公司/.vscode' attribute changed succeeded!
> C++/销售公司/.vscode/ipch:
Folder 'C++/销售公司/.vscode/ipch' copied succeeded!
File 'a/销售公司/.vscode/ipch' attribute changed succeeded!
> C++/销售公司/.vscode/ipch/36722d3649be8048:
Folder 'C++/销售公司/.vscode/ipch/36722d3649be8048' copied succeeded!
File 'a/销售公司/.vscode/ipch/36722d3649be8048' attribute changed succeeded!
> C++/销售公司/.vscode/ipch/36722d3649be8048/main.ipch:
File 'C++/销售公司/.vscode/ipch/36722d3649be8048/main.ipch' copied succeeded!
File 'a/销售公司/.vscode/ipch/36722d3649be8048/main.ipch' attribute changed succeeded!
> C++/销售公司/.vscode/ipch/9594944ebf0247f:
Folder 'C++/销售公司/.vscode/ipch/9594944ebf0247f' copied succeeded!
File 'a/销售公司/.vscode/ipch/9594944ebf0247f' attribute changed succeeded!
> C++/销售公司/.vscode/ipch/a795ae202ecaf51a:
Folder 'C++/销售公司/.vscode/ipch/a795ae202ecaf51a' copied succeeded!
File 'a/销售公司/.vscode/ipch/a795ae202ecaf51a' attribute changed succeeded!
> C++/销售公司/.vscode/ipch/a795ae202ecaf51a/SalesCompany.ipch:
File 'C++/销售公司/.vscode/ipch/a795ae202ecaf51a/SalesCompany.ipch' copied succeeded!
File 'a/销售公司/.vscode/ipch/a795ae202ecaf51a/SalesCompany.ipch' attribute changed succeeded!
> C++/销售公司/main.cpp:
File 'C++/销售公司/main.cpp' copied succeeded!
File 'a/销售公司/main.cpp' attribute changed succeeded!
> C++/销售公司/SalesCompany.cpp:
File 'C++/销售公司/SalesCompany.cpp' copied succeeded!
File 'a/销售公司/SalesCompany.cpp' attribute changed succeeded!
> C++/销售公司/SalesCompany.h:
File 'C++/销售公司/SalesCompany.h' copied succeeded!
File 'a/销售公司/SalesCompany.h' attribute changed succeeded!
➜ code