

实验序号： 2



《UNIX/LINUX 编程环境》

实验报告

实验名称： Linux 系统文件编程

姓 名： 李昊唐

学 院： 工学院

专 业： 物联网工程

班 级： 1 班

学 号： 1995131017

指导教师： 彭凯

实验地址： 数学学院 416

实验日期： 2021 年 11 月 30 日

实验二 Linux 系统文件编程

一、实验目的

- (1) 了解 Linux 系统的文件、文件系统的概念，并观察 Linux 系统的目录结构。
- (2) 掌握 Linux 操作系统中权限管理以及文件遍历操作

二、实验环境

实验配置

本实验所需的软硬件配置如表 1 所示。

配置	2.6 GHz 六核 Intel Core i7, 16 GB 2400 MHz DDR4, Intel UHD Graphics 630 1536 MB
硬件	MacBook Pro (15-inch, 2019)
系统	macOS 12.0.1
应用 软件	vi, sh

实验环境

本实验的环境为 Macintosh 机，如图 1 所示。



图 1 操作实验环境

三、实验原理

- (1) 创建一个文件，观察其属性并修改权限。
- (2) 设计并实现可遍历子目录中所有文件的函数。
- (3) 编写一个文件拷贝程序。

四、实验步骤和实验结果

(1) 创建一个以你自己名字作为文件名的文本文件 `name.txt` 并在终端中查看文件或目录的属性，可使用下列命令：

- 1) 输入命令：`ls -l`
- 2) 修改了文件 `name.txt` 的权限，所有者和群组获得了写权限，其它用户取消了写权限。(提示：Linux 文件权限操作)



```

Last login: Tue Nov 30 10:27:13 on ttys001
[ houtonglei (e) base ~ vi 李昊唐.txt ]
[ houtonglei (e) base ~ ls -l 李昊唐.txt ]
-rw-r--r-- 1 houtonglei staff 0 11 30 10:27 李昊唐.txt
[ houtonglei (e) base ~ chmod ug+ww,o-w 李昊唐.txt ]
[ houtonglei (e) base ~ ls -l 李昊唐.txt ]
-rw-rw-r-- 1 houtonglei staff 0 11 30 10:27 李昊唐.txt
[ houtonglei (e) base ~ ]
    
```

(2) 设计一个可遍历子目录中所有文件的 C 语言程序

该函数的作用是遍历目录，将其所有的子目录和文件输出到终端上。

```

#include <stdio.h>
#include <string.h>
#include <dirent.h>
#include <sys/stat.h>
#define MAX_PATH_LEN (256)

static void trave_dir(char* path) {
    DIR *d = NULL;
    struct dirent *dp = NULL;
    struct stat st;
    char p[MAX_PATH_LEN] = {0};
    
```

```

if (stat(path, &st) < 0 || !S_ISDIR(st.st_mode)) {
    printf("invalid path: %s\n", path); return;
}
if (!(d = opendir(path))) {
    printf("opendir[%s] error: %m\n", path); return;
}
while ((dp = readdir(d)) != NULL) {
    if (!strcmp(dp->d_name, ".") || !strcmp(dp->d_name,
"..", 2))
        continue;
    snprintf(p, sizeof(p) - 1, "%s/%s", path, dp->d_name);
    stat(p, &st);
    if (!S_ISDIR(st.st_mode)) printf("%s\n", dp->d_name);
    else {
        printf("\n%s/\n", dp->d_name); trave_dir(p);
    }
}
closedir(d);
}

int main(int argc, char **argv) {
    char *path = NULL;
    if (argc != 2) {
        printf("%s\n", argv[0]); path = "./";
    } else {
        printf("%s\n", argv[1]); path = argv[1];
    }
    printf("-----\n");
    trave_dir(path);
    return 0;
}

```

(3) 编写一个文件拷贝程序 (CopyFile.c)

CopyFile.c 可以对输入的文件 name.txt 进行备份，备份后的文件名为“**原文件名+_backup**”，并打印出文件的大小。

格式如下： ./CopyFile test

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

int main(int argc, char const *argv[]) {
    char buff[1024];
    int fo, fb;
    long len;
    char *origin_path = (char *)argv[1];
    fo = open(origin_path, O_RDWR|O_CREAT);
    char *ext = strchr(origin_path, '.');
    char *temp = (char*)malloc((strlen(origin_path) + 8) *
sizeof(char));
    char *backup_path = temp;
    char *suffix = "_backup";
    while (origin_path != ext){
        *temp = *(origin_path++);
        temp++;
    }
    while (*suffix != '\0') {
        *temp = *suffix++;
        temp++;
    }
    while (*ext != '\0'){
        *temp = *ext++;
        temp++;
    }
    fb = open(backup_path, O_RDWR|O_CREAT, S_IRUSR|S_IWUSR);
    while ((len = read(fo, buff, 1024))) write(fb, buff, len);
    struct stat st;
    stat(backup_path, &st);
    printf("%lld 字节\t\t%s", st.st_size, backup_path);
}
```

```
cmake-build-debug -- zsh -- 104x5
Last login: Tue Nov 30 11:57:55 on ttys001
houtonglei (e) base ... > 实验 > 实验2 > cmake-build-debug ./CopyFile /Users/houtonglei/李昊唐.txt
13 字节
/Users/houtonglei/李昊唐_backup.txt
houtonglei (e) base ... > 实验 > 实验2 > cmake-build-debug
```

五、实验总结

相较 C++、Java 等语言，C 语言对字符串或文件的操作较为底层。例如对文件路径的编辑需要通过大量的指针操作。同时，绝大部分对文件的操作需要使用者熟知 C 语言中关于 UNIX 系统操作的一众头文件。