

# homework3说明文档

---

161250180 杨郁芩

## homework3说明文档

1. 代码设计
2. 与Linux系统调用源码比较

## 1. 代码设计

---

1. 使用命令行获取参数，可为-l, -a, -i, -d, -r, 获取后调用不同方法

```
// 分析命令行输入
int c;
while((c = getopt( argc , argv , "alidr" )) != -1) {
    switch (c) {
        case 'l':
            caseL(dir);
```

2. `ls -l` 的实现
  - 使用readdir循环读取当前文件夹内容
  - 使用lstat方法得到文件信息
  - 打印信息（因为一些信息的显示格式与规定格式不符，所以调用自己编写的方法进行转换）

```

while ((entry = readdir(dir)) != NULL) {
    char s1[1000];
    strcpy(s1, entry->d_name)
    struct stat buff;

    // 获得文件信息
    lstat(entry->d_name, &buff);
    printFileMode(buff); // 打印文件类型
    printPermission(buff); // 打印文件权限
    printf(" %3d", buff.st_nlink); // 打印硬连接数
    printUser(buff); // 打印用户名
    printGroup(buff); // 打印所属组
    printf(" %5d", (int)buff.st_size); // 打印文件大小
    printf(" %s", (char *)ctime(buff.st_mtime)); // 打印修改时间
    printf(" %s", entry->d_name); // 打印文件名
    /*如果是链接文件，输出链接地址*/
    if(S_ISLNK(buff.st_mode))
    {
        readlink(s1,s1,1000);
        printf(" -> %s ",s1);
    }

    printf("\n"); // 换行
}

```

### 3. *ls -a* 的实现

- 使用readdir循环读取当前文件夹内容
- 打印所有文件的名字，包括.开头的隐藏文件

```

*/
void caseA(DIR* dir) {
    struct dirent *entry;
    while ((entry = readdir(dir)) != NULL) {
        printf(" %s", entry->d_name);
        printf("\n");
    }
}

```

### 4. *ls -r* 的实现

- 使用readdir循环读取当前文件夹内容
- 将除了.开头的文件加入提前声明的字符串数组
- 数组排倒叙，循环打印出来

```

4 void caseR(DIR* dir) {
5     struct dirent *entry;
6     char str[100];
7     string res[100];
8     int count = 0;
9     while ((entry = readdir(dir)) != NULL) {
10         strcpy(str, entry->d_name);
11         if (str[0] != '.') {
12             res[count++] = str;
13         }
14     }
15
16     // 排序并打印
17     sort(res, res+count, cmp);
18     for (int i=0;i<count;i++) {
19         printf("%s", res[i]);
20         printf("\n");
21     }
22 }

```

#### 5. *ls -d* 的实现

- 使用 `getenv("PWD");` 方法得到当前目录，并打印

```

void caseD(DIR* dir) {
    char *dirName = getenv("PWD");
    printf("%s", dirName);
    printf("\n")
}

```

## 2. 与Linux系统调用源码比较

- 源码中在进行ls逻辑处理前，还有对用户权限的设置、输出颜色的设置
- 源码中为了保证安全性，有许多对空数据的条件处理
- 在ls -l中，源码将文件修改时间进行了格式化输出，而我的代码中没有格式化