

## 项目三：实现简单 ls 命令的功能

### 1 学时

- 2 学时

### 2 实验目的

- 理解、掌握、应用目录文件的基本概念，以及 stat 等函数的使用。

### 3 实验内容

- 编写程序，实现 ls 显示指定目录下文件列表的功能，如“ls”、“ls 目录名”、“ls -l 目录名”、“ls -l”的基本功能，将功能独立的部分自定义函数。

### 4 实验原理

目录也是 Linux 系统的一种文件类型，其内容是该目录中文件名和文件 i 节点号的对应关系。

不能对目录文件进行写操作。当用户在一个目录下执行了创建/删除文件或目录的操作，系统会对目录文件的内容进行修改，用户不能自己直接修改目录文件。因此目录文件只能读取其信息，而不能修改。另外对目录文件访问时，不能直接使用 open（）、read（）等系统调用，而是需要使用专门目录的系统调用，如 opendir（）、readdir（）等。

#### （1）打开目录

访问一个目录文件之前，首先应该使用 opendir（）打开该目录文件。

系统调用 opendir 返回一个指向目录文件的指针。如果打开目录失败，则返回一个空指针 NULL。

像对文件操作一样，对目录访问之前也要首先打开目录。打开目录文件时，需要给出目录名。打开后得到一个指向目录文件的指针，通过该指针可以对目录文件进行读目录等操作，访问完目录文件后，通过该指针关闭目录文件。

#### （2）读目录

打开一个目录文件后，就可以通过使用 readdir（）读出该目录文件的内容。

如果执行成功，系统调用 readdir（）返回一个 dirent 结构体类型的指针，该结构体中存放有所读出目录项的信息。如果读到目录文件末尾，或者读目录执行失败，则返回一个空指针 NULL。

读出的目录项存放在一个 dirent 结构体中，该结构体的定义如下：

```
struct dirent
{
    long d_ino;                /* inode number */
    off_t d_off;               /* offset to this dirent */
    unsigned short d_reclen;    /* length of this d_name */
    char d_name [NAME_MAX+1];  /* file name (null-terminated) */
}
```

目录文件的内容存放的是该目录下每一个文件（或子目录）的文件名与其索

引节点 inode 的对应关系。所以每一个目录项都包含有两个主要信息：d\_name 是文件名，d\_ino 是该文件对应的索引节点编号。

每次执行 readdir() 可以读出目录中的一个目录项。如果要读出目录中所有目录项信息，可以通过一个循环结构来调用 readdir()，直到读到目录文件末尾。

### (3) 关闭目录

目录文件访问完之后，可以使用 closedir() 关闭目录。

### (4) 切换目录

系统调用 chdir/fchdir 可以将用户的当前目录切换到指定的目录。

## 5 预习要求和技术准备工作

- 掌握文件，i 节点，文件名，目录之间的关系
- 掌握属性 struct stat 结构体意义
- 掌握 stat 系列函数的使用
- 掌握获得并打印文件属性的方法

## 6 实验环境

- PC 机
- 在 Windows 环境中的 VMware 虚拟机上运行 RedhatLinux9.0 操作系统或者独立的 Redhat Linux 9.0 操作系统
- 基于 Linux 的 vi 编辑器和 gcc 编译器

## 7 实验设计及操作步骤

7.1 以 root 身份登录系统，在/home 目录中创建目录 exp43

```
cd /home
```

```
mkdir exp43
```

7.2 进入刚创建的目录

```
cd exp43
```

7.3 使用 vi 编辑文件，文件名为 myls.c

```
vi myls.c
```

编写程序，实现“ls”、“ls 目录名”、“ls -l 目录名”和“ls -l”功能，即

执行“ls”：能够显示当前目录下文件名和子目录名的信息；

执行“mysls 目录名”：能够显示指定目录下文件名和子目录名的信息；

执行“mysls -l 目录名”：显示指定目录下文件的长格式信息，每个文件所有信息在一行显示；

执行“mysls -l ”显示当前目录下文件长格式信息，每个文件所有信息在一行显示。

★要求：

(1) 将显示文件长格式信息的功能独立做成函数 printlong，可自定义函数原型，也可使用如下原型：

```
void printlong(char *name);
```

参数代表文件名。

★编程思路

```
void printlong(char *name)
```

```
{
```

①调用 `lstat` 获取文件属性，并对返回值做错误判断

②输出一行文件长格式信息，包括：

类型 权限 链接数 拥有者名 组名 大小 日期 文件名

显示要求见上述说明。

```
}
```

```
void lsdir(char *name)
```

```
{
```

①打开目录

②循环读目录，直到目录文件结束，对每个子文件或子目录判断是否是隐藏文件，如果不是隐藏文件则调用 `printlong` 函数输出长格式信息。

③关闭目录

```
}
```

```
main(int argc, char *argv[])
```

```
{
```

① 判断 `argc` 是否为 1 个。

是：执行 `myls` 的功能。

②判断 `argc` 是否为 2 个。

是：

判断 `argv[1]` 是否是 “-l”

是：执行 “`myls -l`” 功能。

否：判断 `argv[1]` 代表的文件名的类型

如果是普通文件、符号链接文件、管道、套接字：输出 `argv[1]` 指定文件名。

如果是目录文件：调用 `lsdir` 函数显示 `argv[1]` 目录下非隐藏文件的文件名。

②判断 `argc` 是否为 3 个。

是：

判断 `argv[1]` 是否是 “-l”

是：

③判断 `argv[2]` 代表的文件名的类型（可自定义函数）

如果是普通文件、符号链接文件、管道、套接字：调用 `printlong`

输出 `argv[2]`指定文件的一行长格式信息。

如果是目录文件：调用 `lsdir` 函数显示 `argv[2]`目录下非隐藏文件的长格式信息。

}

7.4 编译 `myls.c` 为可执行文件 `myls`，并能正确执行。

```
gcc myls.c -o myls
```

7.5 执行 “`myls`”、“`myls -l`”、“`myls /etc/passwd`”、“`myls .`”、“`myls -l /etc/passwd`”和“`myls -l .`”将正确的执行结果屏幕截屏，参考示例如下：

```
cst@ubuntu:~/linuxsys2021/43/teacher$ ./myls
myls.c myls myls.zip
cst@ubuntu:~/linuxsys2021/43/teacher$ ./myls -l
-rw-rw-r-- 1 cst cst 4531 Jul 16 17:00 myls.c
-rwxrwxr-x 1 cst cst 17840 Jul 22 17:18 myls
-rwxrw-rw- 1 cst cst 1492 Feb 28 16:23 myls.zip
cst@ubuntu:~/linuxsys2021/43/teacher$ ./myls /etc/passwd
/etc/passwd
cst@ubuntu:~/linuxsys2021/43/teacher$ ./myls .
myls.c myls myls.zip
cst@ubuntu:~/linuxsys2021/43/teacher$ ./myls -l /etc/passwd
-rw-r--r-- 1 root root 2907 Jun 28 15:24 /etc/passwd
cst@ubuntu:~/linuxsys2021/43/teacher$ ./myls -l .
-rw-rw-r-- 1 cst cst 4531 Jul 16 17:00 myls.c
-rwxrwxr-x 1 cst cst 17840 Jul 22 17:18 myls
-rwxrw-rw- 1 cst cst 1492 Feb 28 16:23 myls.zip
```

## 8 实验报告提交要求：

将实验操作每个步骤中的命令、源程序以及截图写入实验报告，实验报告命名为“学号姓名-实验 43.doc”，交给指定人员。

## 9 项目思考

本例中的参数是有限并且规定好的格式，思考并调研更多参数且顺序不定的时候，如何去判断参数。