项目二：显示文件长格式信息

1 学时

- 2 学时

2 实验目的

- 理解、掌握、应用文件的基本概念，以及 stat 等函数的使用。

3 实验内容

- 编写程序，实现“ls –l 文件名”和“ls –l 目录名”两个命令的基本功能，将功能独立的部分自定义函数。

4 实验原理

4.1 文件基本概念

Linux 下文件主要分为七类：普通文件、目录文件、符号链接文件、字符设备文件、块设备文件、管道文件、套接口文件。这七种文件可以通过“ls –l”命令显示的长格式信息的第一列观察到。

每个文件都有 i 节点与之对应，每个存在磁盘上的文件都由 i 节点和数据块两部分组成。

Linux 操作系统通过 VFS 支持多种文件系统，所对于 VFS 来说有一种 i 节点数据结构 struct inode，对于具体的文件系统有具体的 i 节点数据结构，如 ext2 文件系统中的 i 节点数据结构为 struct ext2_inode。

一个文件对应一个 i 节点，每个 i 节点都有一个 i 节点号。但是用户是通过文件名字来访问文件的，目录文件中保存了文件名和 i 节点号的对应关系。一个 i 节点可以对应多个文件名字，一个 i 节点对应的文件名字个数即硬链接的个数，也即文件长格式信息中的链接数。

文件系统是按一定规律组织起来的有序的文件组织结构，是管理文件的一组软件的集合。文件系统的种类很多，如：ext2，Minix，MSDOS，VFAT，NTFS， ISO9660（光盘）等。Linux 操作系统通过 VFS（虚拟文件系统）支持多种文件系统。VFS 是建立在具体文件系统之上的虚拟的文件系统，它为用户程序提供一个统一的、抽象的、虚拟的文件系统界面。

4.2 文件描述符

内核（kernel）利用文件描述符（file descriptor）来访问文件。文件描述符是非负整数。打开现存文件或新建文件时，内核会返回一个文件描述符。读写文件也需要使用文件描述符来指定待读写的文件。

习惯上，标准输入（standard input）的文件描述符是 0，标准输出（standard output）是 1，标准错误（standard error）是 2。尽管这种习惯并非 UNIX 内核的特性，但是因为一些 shell 和很多应用程序都使用这种习惯，因此，如果内核不遵循这种习惯的话，很多应用程序将不能使用。

POSIX 定义了 STDIN_FILENO、STDOUT_FILENO 和 STDERR_FILENO 来代替 0、1、2。这三个符号常量的定义位于头文件 unistd.h。

文件描述符的有效范围是 0 到 OPEN_MAX。一般来说，每个进程最多可以打开 64 个文件（0 — 63）。对于 FreeBSD 5.2.1、Mac OS X 10.3 和 Solaris 9 来说，每个进程最多可以打开文件的多少取决于系统内存的大小，int 的大小，以及系统管理员设定的限制。Linux 2.4.22 强制规定最多不能超过 1,048,576 。

文件描述符是由无符号整数表示的句柄，进程使用它来标识打开的文件。文件描述符与包括相关信息（如文件的打开模式、文件的位置类型、文件的初始类型等）的文件对象相关联。

4.3 文件属性

（1）获取文件属性

文件属性保存在 i 节点中（struct inode），可以使用 stat 系列函数获得 i 节点中的文件属性并保存在 struct stat 数据类型中。

```
struct stat
{
    dev_t     st_dev;       //文件所在的设备 ID，unsigned short
    ino_t     st_ino;           //i 节点号,unsigned long
    mode_t    st_mode;      //类型权限,unsigned short
    nlink_t   st_nlink;     //硬链接个数,unsigned short
    uid_t     st_uid;           //uid,unsigned short
    gid_t     st_gid;           //gid,unsigned short
    dev_t     st_rdev; //若是特殊文件，设备 ID，unsigned short
    off_t     st_size;  //文件大小（字节数）,long
    time_t    st_atime;     //上次访问时间,long
    time_t    st_mtime;     //上次修改内容时间,long
    time_t    st_ctime;     //上次文件信息修改时间,long
    blksize_t     st_blksize;   //块大小（字节）,long
    blkcnt_t st_blocks;     //文件块个数,long
};
```

stat 系列函数可以获得文件属性。

```
#include <sys/stat.h>
int stat(const char *path,struct stat *buf);
int lstat(cosnt char *path,struct stat *buf);
int fstat(int fd, struct stat *buf);
```

功能：

stat：通过文件路径得到文件属性

stat：通过文件描述符得到文件属性

lstat：仅查询符号链接本身属性而不是所链接文件属性。

参数：

path：文件路径

fd：文件描述符

buf：返回的文件属性（结构体）

返回值：成功返回 0，否则返回－1（置 errno）

（2）文件类型判断（/usr/include/Linux/stat.h）

根据 st_mode 高四位进行判断

判断　st_mode & S_IFMT 的结果是否与如下宏匹配

S_IFREG　普通文件

S_IFDIR　目录文件

S_IFLNK　符号链接文件

S_IFCHR　字符设备文件

S_IFBLK　块设备文件

S_IFIFO　管道文件

S_IFSOCK　套接口文件

通常可以采用 switch-case 结构进行判断：

假设 m 代表属性的 st_mode 的值。

```
switch(m & S_IFMT)
{
case S_IFREG：putchar(' - ');break;
//……以此类推
}
```

（3）文件权限判断

根据 st_mode 低 9 位进行判断，如

st_mode & S_IRUSR：真代表用户可读，假代表用户不可读

即：st_mode & S_Ipwww：真代表 www 可 p，假代表 www 不可 p。

其中 p 可用 R 或 W 或 X 代替，www 可用 USR 或 GRP 或 OTH 代替。

通常采用 putchar((st_mode & S_IRUSR)?'r':'-');方式显示。

（4）uid 到用户名的转换

使用 getpwuid 函数完成。

```
#include <pwd.h>
```

struct passwd *getpwuid(uid_t uid);

参数：uid－用户 id。

返回值：成功返结构体指针，否则返 NULL(置 errno)。

其中 passwd 结构体定义如下：

```
struct passwd
 {
        char *pw_name;          //用户名
        char *pw_passwd;        //用户密码
        uid_t pw_uid;           //用户 ID
        gid_t pw_gid;           //用户所在组 ID
        char *pw_gecos;         //真实名字
        char *pw_dir;           //用户主目录
        char *pw_shell;         //用户所用 Shell
  };
```

（5）gid 到组名的转换

使用 getgrgid 函数完成。

#include <grp.h>

struct group *getgrgid(gid_t gid);

参数：gid－用户组 id。

返回值：成功返回结构体指针，否则返回 NULL(置 errno)。

其中 group 结构体定义如下：

```
struct group
{
        char *gr_name;          //组名
        char *gr_passwd;        //组密码
        gid_t gr_gid;           //组 ID
        char **gr_mem;          //组成员列表
};
```

（6）关于日期和时间的处理

可直接输出 struct stat 结构体成员 st_size 的值。

还可以按照“ls –l”命令的输出标准输出，参考程序为 print_date 函数，该函数不输出年，除非时间已过 6 个月。

参数：struct stat 结构体指针，代表文件属性信息。

（7）获取符号链接文件指向的文件名

#include <unistd.h>

int readlink(const char *pathname, char *buf,int bufsize);

参数

pathname：符号链接文件名

buf：存放被链接文件名的缓冲区

bufsize：缓冲区大小

返回值：成功返回实际写入缓冲区的字节数，失败返回 0。

（8）对于设备文件，因为文件大小没有意义，所以输出的是主设备号和次设备号，通常最右边 8 位表示次设备号。

假设文件属性结构体为 struct stat file，则打印主设备号和次设备号语句可以如下：

printf("%4u,%4u",(unsigned)(file.st_rdev >> 8), (unsigned)(file.st_rdev & 0xFF));

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 exp42

```
cd /home
mkdir exp42
```

7.2 进入刚创建的目录

```
cd exp42
```

7.3 使用 vi 编辑文件，文件名为 myll.c

```
vi myll.c
```

7.4 编写程序，实现“ls –l 文件名”功能，即执行“myll –l 文件名”能够显示指定文件的长格式信息，所有信息在一行显示。如显示/etc/passwd 文件长格式信息为：

```
-rw-r--r--  1 root  root  1725 12 月  2 23:10 /etc/passwd
```

★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。

（2）类型显示：

普通文件（-），目录文件（d），符号链接文件（l），字符设备文件（c），
块设备文件（b），管道文件（p），套接字文件（s）。

（3）权限显示：

三组用户（拥有者，同组，其他），每组三种权限（r，w，x）。
有则显示相应字符，无则显示“-”。

（4）文件大小

直接显示文件字符数即可。

（5）上次修改时间

显示“年：月：日：时：分”即可，不用带中文字符。

（6）文件名：

如果是非符号链接文件，则直接显示文件名，如果是符号链接文件则要显示该符号链接文件所引用的原文件的名字，如：“s1---->f1”形式。

（7）（选作）

＊如果是设备文件，在文件大小部分显示主设备号和从设备号。

＊将上次修改时间根据 ls –l 命令结果显示（6 个月内的时间显示“月：日：时：分”，6 个月之前的显示“年：月：日”）。

★注意：

（1）根据要求，本程序要通过命令行参数实现

（2）使用 lstat 函数获取文件属性信息

（3）得到的文件属性中，有的可以直接输出，有的需要转换后才能输出。

★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。
}
main(int argc,char *argv[])
{
```

①判断 argc 是否符合要求，应该为 3 个。

②判断 argv[1]是否是“-l”

是： ③调用 printlong 函数，输出 argv[2]文件的长格式信息

否：提示用户信息“usage:program -l filename or dirname”

}

7.5 编译 myll.c 为可执行文件 myll，并能正确执行。

```
gcc  myll.c  -o  myll
```

7.6 执行“myll –l /etc/passwd”和“myll –l ./myll.c”将正确的执行结果屏幕截屏。参考示例如下：

```
cst@ubuntu:~/linuxsys2021/42/teacher$ ./myll -l /etc/passwd
-rw-r--r-- 1  root  root  2907  Jun 28 15:24 /etc/passwd
cst@ubuntu:~/linuxsys2021/42/teacher$ ./myll -l ./myll.c
-rw-rw-r-- 1  cst  cst  3752  Jul 16 16:55 ./myll.c
```

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 42.doc”，交给指定人员。

9 项目思考题

(1)考虑如果用户输入多个文件名，如“myll - l 文件名 1文件名 2”时如何实现。

(2)完善 myll，实现加入参数-a 等功能。