**一.文件属性与目录**

在文件操作中，文件属性与目录类的操作比较特殊。

采用的函数比较特殊，这些函数分散在man手册第2页与第3页（系统调用+库函数）。

**1.1 相关函数**

stat\fstat\lstat //文件属性

opendir\readdir\closedir //目录类

**二、函数备注**

int stat(const char \*path, struct stat \*buf);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： 获取文件信息

\* Input: path：路径名，

\* Output: buf:获取的信息保存的地址

\* Return: 成功返回0， 出错返回-1

\* Others:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct stat

{

dev\_t st\_dev; //

ino\_t st\_ino; //

mode\_t st\_mode; //文件的类型和存取的权限

nlink\_t st\_nlink; //

uid\_t st\_uid; //所属用户

gid\_t st\_gid; //所属用户组

dev\_t st\_rdev; //

off\_t st\_size; //文件大小, 以字节计算

unsigned long st\_blksize; //

unsigned long st\_blocks; //

time\_t st\_atime; //文件最近一次被存取或被执行的时间

time\_t st\_mtime; //文件最后一次被修改的时间

time\_t st\_ctime; //最近一次被更改的时间

};

int lstat(const char \*path, struct stat \*buf);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： lstat()与stat()作用完全相同, 都是取得参数file\_name所指的文件状态,

其差别在于, 当文件为符号连接时, lstat()会返回该link本身的状态。

\* Input: path：路径名，buf:获取的信息保存的地址

\* Return: 成功返回0， 出错返回-1

\* Others:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

DIR \*opendir(const char \*name);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： 打开目录

\* Input: name：路径名

\* Return: 成功则返回DIR\* 型态的目录流(类似FILE \*), 打开失败则返回NULL

\* Others:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct dirent \* readdir(DIR \* dir);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： 读目录

\* Input: dir：目录流

\* Return: 成功则返回目录下的信息（struct dirent\*），失败或到达末尾则返回NULL

\* Others:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int closedir(DIR \*dirp);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Description： 关闭目录

\* Input: dirp：目录流

\* Return:

\* Others: 成功返回0 失败返回-1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct dirent

{

ino\_t d\_ino;

off\_t d\_off;

unsigned short d\_reclen;

unsigned char d\_type; /\* 文件类型 \*/

char d\_name[256]; /\* 文件名 \*/

};

------------------------------------------------------------------------------

//用于解析uid\_t 和gid\_t

struct passwd \*getpwuid(uid\_t uid); //得到用户信息

struct group \*getgrgid(gid\_t gid); //得到用户组信息

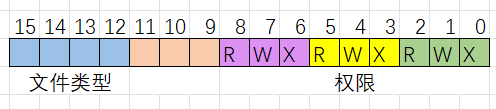
**三、补充**

**3.1 st\_mode**

在 Linux 系统中，st\_mode 是 struct stat 结构体的核心成员之一（定义在头文件 <sys/stat.h> 中），用于存储文件的​​类型​​和​​权限信息​​。

**3.1.1 st\_mode位结构**

st\_mode 的 16 位可分为三部分。



15bit ~ 12bit 保存文件类型

11bit ~ 9bit 保存执行文件时设置的信息

8bit ~ 0bit 保存文件访问权限

文件类型

文件类型位于【15:12】，通过以下宏 可以获取其值并判断类型

#define S\_IFMT 0170000 文件类型的位遮罩

#define S\_IFSOCK 0140000 socket

#define S\_IFLNK 0120000 符号链接(symbolic link)

#define S\_IFREG 0100000 一般文件

#define S\_IFBLK 0060000 区块装置(block device)

#define S\_IFDIR 0040000 目录

#define S\_IFCHR 0020000 字符装置(character device)

#define S\_IFIFO 0010000 先进先出(fifo)

权限

权限位为于【8:0】，共9位，分3组，每组 3 位，分别对应​​所有者​​、​​所属组​​、​​其他用户​​的权限。

通过位运算可以获取对应的权限

|  |  |  |  |
| --- | --- | --- | --- |
| 权限组 | 读（r）掩码 | 写（w）掩码 | 执行（x）掩码 |
| 所有者 | S\_IRUSR | S\_IWUSR | S\_IXUSR |
| 所属组 | S\_IRGRP | S\_IWGRP | S\_IXGRP |
| 其他用户 | S\_IROTH | S\_IWOTH | S\_IXOTH |

//bit8 ~ bit0，权限属性区域

//文件所有者（owner）

#define S\_IRWXU 00700/\* mask for file owner permissions \*/

#define S\_IRUSR 00400/\* owner has read permission \*/

#define S\_IWUSR 00200/\* owner has write permission \*/

#define S\_IXUSR 00100/\* owner has execute permission \*/

//组用户（group）

#define S\_IRWXG 00070/\* mask for group permissions \*/

#define S\_IRGRP 00040/\* group has read permission \*/

#define S\_IWGRP 00020/\* group has write permission \*/

#define S\_IXGRP 00010/\* group has execute permission \*/

//其他用户（other）

#define S\_IRWXO 00007/\* mask for permissions for others (not in group) \*/

#define S\_IROTH 00004/\* others have read permission \*/

#define S\_IWOTH 00002/\* others have write permission \*/

#define S\_IXOTH 00001/\* others have execute permission \*/

**作业**

1. 通过文件io相关函数，实现7月3号的作业。