



西安邮电大学

XI'AN UNIVERSITY OF POSTS & TELECOMMUNICATIONS

Linux 编程技
术



第2章 文件、目录管理

——Linux 文件系统的常用接口-下

文件属性、目录相关的系统调用

ls命令的实现

shell下ls命令获取
文件属性

ls -l命令的结果:

```
-rw-r--r-- 1 root root 71 1月 11 11:43 test.c  
drwxr-xr-x 3 root root 4096 3月 20 08:34 zll
```

程序函数获取文
件属性

- int stat(const char *path, struct stat *buf);
- int fstat(int fd, struct stat *buf);
- int lstat(const char *path, struct stat *buf);

1. 硬链接文件

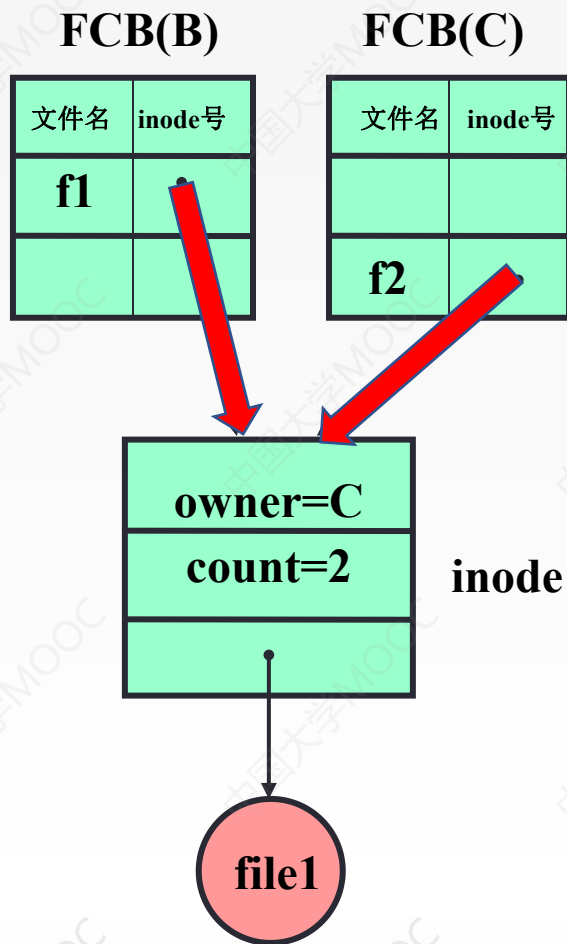


图1 硬链接文件

2. 符号链接文件

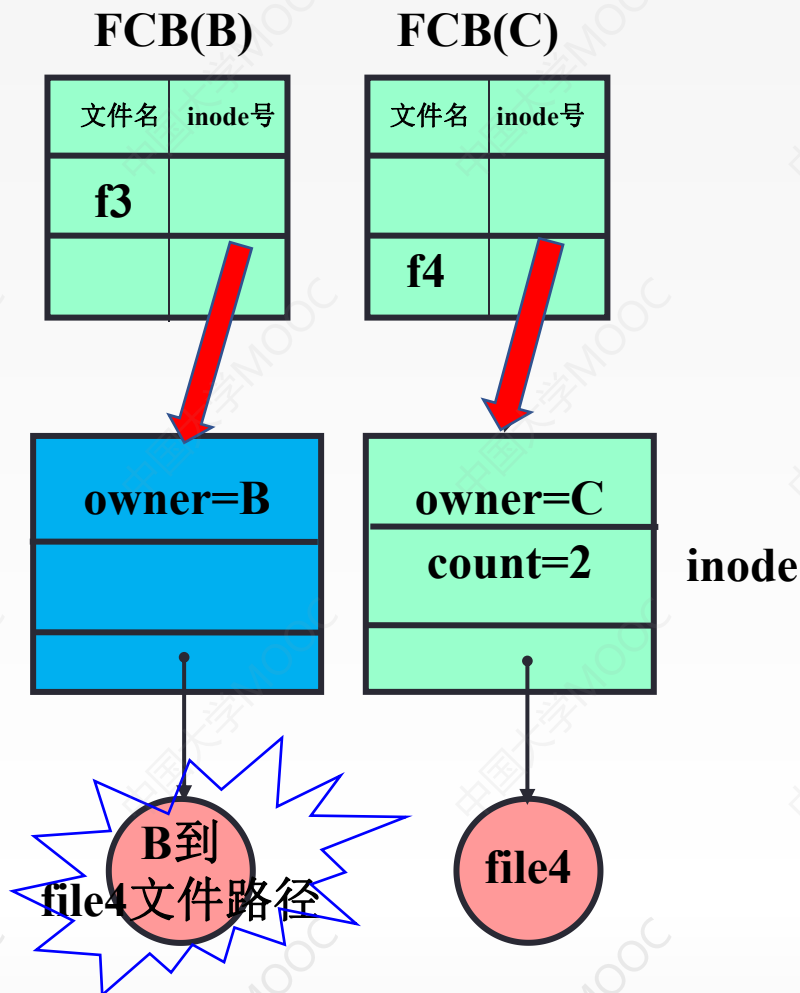


图2 符号链接文件

```
struct stat
{
    dev_t      st_dev;
    ino_t      st_ino;
    mode_t     st_mode;
    nlink_t    st_nlink;
    uid_t      st_uid;
    gid_t      st_gid;
    dev_t      st_rdev;
    off_t      st_size;
    blksize_t  st_blksize;
    blkcnt_t   st_blocks;
    time_t     st_atime;
    time_t     st_mtime;
    time_t     st_ctime;
};
```

st_mode:

文件的类型和存取权限

•st_uid:

文件所有者的用户id

•st_gid:

文件所有者的组id

•st_size:

文件大小

•st_atime:

文件最近一次被访问的时间

•st_mtime:

文件最后一次修改的时间

(1) chmod/fchmod函数

函数原型

```
int chmod(const char *path, mode_t mode);  
int fchmod(int fd, mode_t mode);
```

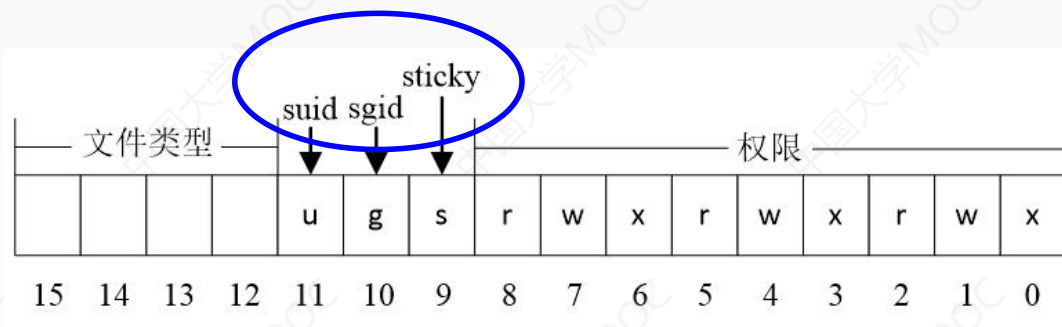


图 3 st_mode结构

```
chmod("hello.c", S_IRUSR|S_IWUSR|S_IWOTH)
```

使用“|”进行权限的组合

(2) suid、sgid、sticky权限

1) suid



usr1

suid

/usr/bin/passwd: 修改账号信息
权限: **rw****s**r-xr-x
文件所有者: root

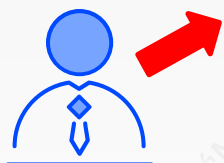
/etc/shadow : 存放账号密码
权限: rw-r-----
文件所有者: root

suid的功能总结为:

- suid权限仅对二进制程序有效; 如passwd命令;
- 执行者对于该程序需要具有x的可执行权限; 如passwd权限为rwsr-xr-x,
- 本权限仅在执行该程序的过程有效; user1执行passwd命令有效;
- 执行者将具有该程序**所有者**的权限。user1具有root权限, root是passwd和shadow文件的所有者。

(2) suid、sgid、sticky权限

2)sgid



usr1

sgid

/usr/bin/locate: 查找文件或目录
权限: `rwX--s--X`
文件所有者: root
文件所属组: slocate



/var/lib/mlocate/mlocate.db:
存放本地所有文件和目录信息
权限: `rw-r-----`
文件所有者: root
文件所属组: slocate

权限: `rw-r-----`

其他用户

sgid的功能总结为:

- sgid权限仅对二进制程序有效; 如locate命令;
- 执行者对于该程序需要具有x的可执行权限; 如locate权限为`rwX--s--X` ;
- 执行者将具有**该程序用户组**的权限。
usr1具有slocate组的读mlocate文件的权限。

(2) suid、sgid、sticky权限

3) sticky

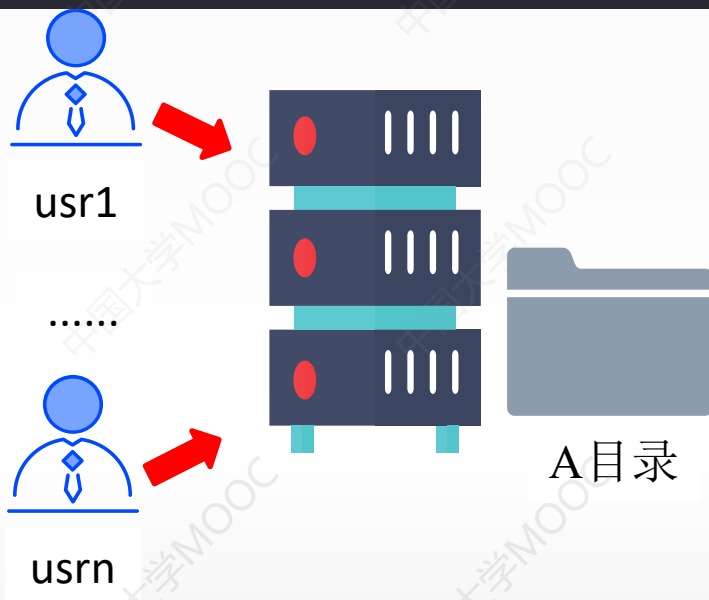
sticky

/tmp目录的权限

权限: `rw-r--r-s`

文件所有者: `root`

文件所属组: `slocate`



sticky的功能总结:

- 仅对目录有效;
- 用户对此目录具有w、x权限;
- 用户在该目录下创建文件或目录时, 仅自己与root账户才有权删除该文件。

1 `DIR *opendir(const char *name)`

打开目录

2 `struct dirent *readdir(DIR *dirp);`

操作目录

3 `int closedir(DIR *dirp)`

关闭目录

DIR 结构体

```
1. struct __dirstream
2. {
3.     void *__fd;
4.     char *__data;
5.     int __entry_data;
6.     char *__ptr;
7.     int __entry_ptr;
8.     size_t __allocation;
9.     size_t __size;
10.     __libc_lock_define(, __lock)
11. };
12.
13. typedef struct __dirstream DIR;
```

struct dirent

dirent 结构体

```
{
    long d_ino; //inode编号
    off_t d_off;
    unsigned short d_reclen;
    unsigned char d_type;
    char d_name[NAME_MAX+1]; //文件名
}
```

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

struct stat

stat 结构体

```
{
    dev_t      st_dev;
    ino_t      st_ino;
    mode_t     st_mode;
    nlink_t    st_nlink;
    uid_t      st_uid;
    gid_t      st_gid;
    .....
};
```

打开目录



遍历读取目录下的子目录或文件，若是子目录则递归遍历读取



关闭目录

递归遍历指定目录

```
void printdir(char *dir, int depth)
{
    DIR *dp;
    struct dirent *entry;
    struct stat statbuf;

    if((dp=openendir(dir))==NULL)
    {
        fprintf(stderr,"cannot open directory :%s\n",dir);
        return;
    }
    chdir(dir);
```

打开目录

```
while((entry=readir(dp))!=NULL)
{
    lstat(entry->d_name,&statbuf);
    if(S_ISDIR(statbuf.st_mode))
    {
        /*发现目录,但是忽略.和..*/
        if(strcmp(".",entry->d_name)==0||strcmp("..",entry->d_name)==0)
            continue;
        printf("%*s%s\n",depth,"",entry->d_name);
        printdir(entry->d_name,depth+4);//递归遍历指定目录
    }
    else
        printf("%*s%s\n",depth,"",entry->d_name);
}

closedir(dp);
}
```

递归
遍历
目录

关闭
目录

□ **ls命令功能：**显示目录或文件的一些属性信息。

□ **ls命令格式：**

ls命令使用格式为：ls [-a][-l] [<路径>]

□ **ls -l <目录或文件>：**

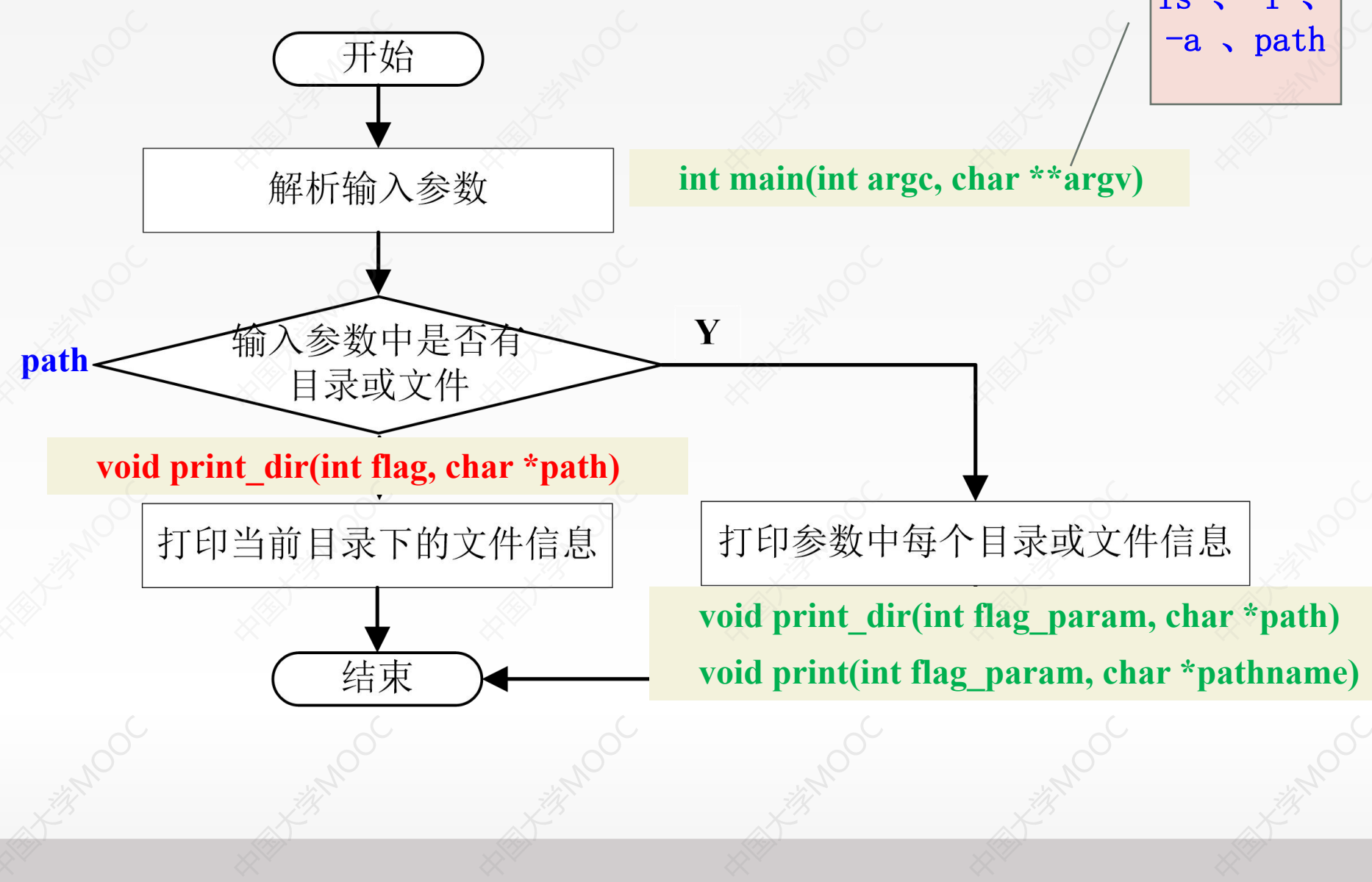
列出目录下的所有子目录和文件的主要属性与权限等数据。

或列出文件主要属性与权限等数据

□ **ls -a <目录>：**

列出目录下的所有子目录和连同隐藏文件在内的全部文件。

接收参数
ls 、 -l 、
-a 、 path




```
void print_dir(int flag_param, char *path)
```

函数流程：

(1) 获取path目录下对象（子目录和文件）的总数、名称、最长对象名。

(2) 对对象名进行字母排序。

(3) 循环遍历path目录下的每个对象，每次循环调用 `print(int flag, char *pathname)` 函数显示每个文件的信息。

文件名

```
void print(int flag_param, char *pathname)
```

flag_param

flag= PARAM_NONE//无参

```
void print_single(char *name)
```

flag= PARAM_A //-a

```
void print_single(char *name)
```

flag= PARAM_L //-l

```
void print_attribute(struct stat buf ,char * name)
```

flag= PARAM_A+PARAM_L //-al或-la

```
void print_attribute(struct stat buf ,char * name)
```

```
root@ubuntu:3# ./myls -al
drwxr-xr-x      2 rootsroot    4096 Wed Jan 29 19:22:23 2020 .
drwxr-xr-x      4 rootsroot    4096 Sun Jan  5 19:31:37 2020 ..
-rw-r--r--      1 rootsroot  12288 Thu Apr  5 12:23:48 2018 .fcntl_lock_demo.c.swp
-rw-----      1 rootsroot  12288 Sun Apr  1 13:32:13 2018 .utime_demo.c.swp
-rw-r--r--      1 rootsroot     390 Tue Mar 27 15:45:10 2018 1
-rw-r--r--      1 rootsroot      2 Wed Mar 28 17:04:27 2018 a.cc
-rwxr-xr-x      1 rootsroot   8824 Tue Mar 27 22:23:17 2018 demo3_2
-rw-r--r--      1 rootsroot     490 Tue Mar 27 22:23:15 2018 demo3_2.c
-rwxr-xr-x      1 rootsroot   8942 Thu Apr  5 14:09:22 2018 fcntl_demo
-rw-r--r--      1 rootsroot     901 Thu Apr  5 11:54:27 2018 fcntl_demo.c
-rw-r--r--      1 rootsroot      0 Thu Apr  5 11:54:40 2018 fcntl_demo_lock.c
-rw-r--r--      1 rootsroot      0 Thu Apr  5 15:28:31 2018 fcntl_demo_lock2.c
```

- chmod/fchmod、stat/lstat/fstat、
opendir/readdir/closedir

- 符号链接文件

- suid、sgid、sticky

- ls实现 (myls)

谢谢大家！

