

# stat文件

## 概述

```
#include<sys/stat.h>
int stat(const char *restrict pathname,struct stat *restrict buf);
int fstat(int fields,struct stat *buf);
int lstat(const char *restrict pathname,struct stat *restrict buf);
```

## stat

```
int stat(const char *restrict pathname,struct stat *restrict buf);
```

### const char \*restrict pathname

pathname即readdir返回的dirent的dirent->name  
或者是一个全文件名也可以

1. opendir函数打开目录, 返回指向目录的结构体DIR \*
2. readdir(DIR \*)读取目录及目录下所有文件, 返回指向目录下所有文件的struct dirent \*结构体
3. 遍历struct dirent \*结构体, stat(dirent->name, stat\*)来获取每个文件的详细信息

### struct stat \*restrict buf

```
struct stat{
    mode_t st_mode; //文件类型和权限信息
    ino_t st_ino; //inode号
    dev_t st_dev; //文件使用的设备号
    dev_t st_rdev; //设备文件的设备号
    nlink_t st_nlink; //文件的硬连接数
    uid_t st_uid; //用户ID
    gid_t st_gid; //组ID
    off_t st_size; //以字节为单位的文件容量
    time_t st_atime; //最后一次访问的时间
    time_t st_mtime; //文件内容最后一次被更改的时间
    time_t st_ctime; //文件结构最后一次被更改的时间
    blksize_t st_blksize; //包含该文件的磁盘块的大小
    blkcnt_t st_blocks; //该文件所占的磁盘块
};
```

mode\_t st\_mode:

- 普通文件: S\_ISREG()
- 目录文件: S\_ISDIR()
- 块特殊文件: S\_ISBLK()
- 字符特殊文件: S\_ISCHR()
- 套接字: S\_ISSOCK()
- FIFO: S\_ISFIFO()
- 符号链接: S\_ISLNK()

```
//如下使用  
S_ISDIR(buf.st_mode)
```