<https://www.cnblogs.com/CSU-PL/archive/2013/06/06/3120757.html>

在使用这个结构体和方法时，需要引入：

<sys/types.h>

<sys/stat.h>

struct stat这个结构体是用来描述一个linux系统文件系统中的文件属性的结构。

可以有两种方法来获取一个文件的属性：

1、通过路径：

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

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

两个函数的第一个参数都是文件的路径，第二个参数是struct stat的指针。返回值为0，表示成功执行。

执行失败是，error被自动设置为下面的值：

EBADF： 文件描述词无效

EFAULT： 地址空间不可访问

ELOOP： 遍历路径时遇到太多的符号连接

ENAMETOOLONG：文件路径名太长

ENOENT：路径名的部分组件不存在，或路径名是空字串

ENOMEM：内存不足

ENOTDIR：路径名的部分组件不是目录

这两个方法区别在于stat没有处理字符链接(软链接）的能力，如果一个文件是符号链接，stat会直接返回它所指向的文件的属性；而lstat返回的就是这个符号链接的内容。这里需要说明一下的是软链接和硬链接的含义。我们知道目录在linux中也是一个文件，文件的内容就是这这个目录下面所有文件与inode的对应关系。那么所谓的硬链接就是在某一个目录下面将一个文件名与一个inode关联起来，其实就是添加一条记录！而软链接也叫符号链接更加简单了，这个文件的内容就是一个字符串，这个字符串就是它所链接的文件的绝对或者相对地址。

2、通过文件描述符

int fstat(int fdp, struct stat \*struct\_stat);　　//通过文件描述符获取文件对应的属性。fdp为文件描述符

下面是这个结构的结构

struct stat {

        mode\_t     st\_mode;       //文件对应的模式，文件，目录等

        ino\_t      st\_ino;       //inode节点号

        dev\_t      st\_dev;        //设备号码

        dev\_t      st\_rdev;       //特殊设备号码

        nlink\_t    st\_nlink;      //文件的连接数

        uid\_t      st\_uid;        //文件所有者

        gid\_t      st\_gid;        //文件所有者对应的组

        off\_t      st\_size;       //普通文件，对应的文件字节数

        time\_t     st\_atime;      //文件最后被访问的时间

        time\_t     st\_mtime;      //文件内容最后被修改的时间

        time\_t     st\_ctime;      //文件状态改变时间

        blksize\_t st\_blksize;    //文件内容对应的块大小

        blkcnt\_t   st\_blocks;     //伟建内容对应的块数量

      };

stat结构体中的st\_mode 则定义了下列数种情况：  
    S\_IFMT   0170000    文件类型的位遮罩  
    S\_IFSOCK 0140000    scoket  
    S\_IFLNK 0120000     符号连接  
    S\_IFREG 0100000     一般文件  
    S\_IFBLK 0060000     区块装置  
    S\_IFDIR 0040000     目录  
    S\_IFCHR 0020000     字符装置  
    S\_IFIFO 0010000     先进先出  
  
    S\_ISUID 04000     文件的(set user-id on execution)位  
    S\_ISGID 02000     文件的(set group-id on execution)位  
    S\_ISVTX 01000     文件的sticky位  
  
    S\_IRUSR(S\_IREAD) 00400     文件所有者具可读取权限  
    S\_IWUSR(S\_IWRITE)00200     文件所有者具可写入权限  
    S\_IXUSR(S\_IEXEC) 00100     文件所有者具可执行权限  
  
    S\_IRGRP 00040             用户组具可读取权限  
    S\_IWGRP 00020             用户组具可写入权限  
    S\_IXGRP 00010             用户组具可执行权限  
  
    S\_IROTH 00004             其他用户具可读取权限  
    S\_IWOTH 00002             其他用户具可写入权限  
    S\_IXOTH 00001             其他用户具可执行权限  
  
    上述的文件类型在POSIX中定义了检查这些类型的宏定义：  
    S\_ISLNK (st\_mode)    判断是否为符号连接  
    S\_ISREG (st\_mode)    是否为一般文件  
    S\_ISDIR (st\_mode)    是否为目录  
    S\_ISCHR (st\_mode)    是否为字符装置文件  
    S\_ISBLK (s3e)        是否为先进先出  
    S\_ISSOCK (st\_mode)   是否为socket  
若一目录具有sticky位(S\_ISVTX)，则表示在此目录下的文件只能被该文件所有者、此目录所有者或root来删除或改名，在linux中，最典型的就是这个/tmp目录啦。