

程序 14-31 linux/include/sys/stat.h

```

1 #ifndef SYS_STAT_H
2 #define SYS_STAT_H
3
4 #include <sys/types.h>
5
6 struct stat {
7     dev_t    st_dev;        // 含有文件的设备号。
8     ino_t    st_ino;        // 文件 i 节点号。
9     umode_t  st_mode;        // 文件类型和属性（见下面）。
10    nlink_t  st_nlink;       // 指定文件的连接数。
11    uid_t    st_uid;        // 文件的用户（标识）号。
12    gid_t    st_gid;        // 文件的组号。
13    dev_t    st_rdev;       // 设备号（如果文件是特殊的字符文件或块文件）。
14    off_t    st_size;       // 文件大小（字节数）（如果文件是常规文件）。
15    time_t   st_atime;       // 上次（最后）访问时间。
16    time_t   st_mtime;       // 最后修改时间。
17    time_t   st_ctime;       // 最后节点修改时间。
18 };
19
20 //
21 // 下面是为 st_mode 字段所用的值定义的符号名称。这些值均用八进制表示。参见第 12 章文件
22 // 系统中图 12-5（i 节点属性字段内容）。为便于记忆，这些符号名称均为一些英文单词的首
23 // 字母或缩写组合而成。例如名称 S_IFMT 的每个字母分别代表单词 State、Inode、File、
24 // Mask 和 Type；而名称 S_IFREG 则是 State、Inode、File 和 REGular 几个大写字母的组合；
25 // 名称 S_IRWXU 是 State、Inode、Read、Write、eXecute 和 User 的组合。其它名称可以此类推。
26 // 文件类型：
27 #define S_IFMT 00170000    // 文件类型比特位屏蔽码（8 进制表示）。
28 #define S_IFLNK 0120000    // 符号链接。
29 #define S_IFREG 0100000    // 常规文件。
30 #define S_IFBLK 0060000    // 块特殊（设备）文件，如磁盘 dev/fd0。
31 #define S_IFDIR 0040000    // 目录。
32 #define S_IFCHR 0020000    // 字符设备文件。
33 #define S_IFIFO 0010000    // FIFO 特殊文件。
34 // 文件属性位：
35 // S_ISUID 用于测试文件的 set-user-ID 标志是否置位。若该标志置位，则当执行该文件时，进程的
36 // 有效用户 ID 将被设置为该文件宿主的用户 ID。S_ISGID 则是针对组 ID 进行相同处理。
37 #define S_ISUID 0004000    // 执行时设置用户 ID（set-user-ID）。
38 #define S_ISGID 0002000    // 执行时设置组 ID（set-group-ID）。
39 #define S_ISVTX 0001000    // 对于目录，受限删除标志。
40
41 #define S_ISLNK(m)      (((m) & S_IFMT) == S_IFLNK)    // 测试是否符号链接文件。
42 #define S_ISREG(m)      (((m) & S_IFMT) == S_IFREG)    // 测试是否常规文件。
43 #define S_ISDIR(m)      (((m) & S_IFMT) == S_IFDIR)    // 是否目录文件。
44 #define S_ISCHR(m)      (((m) & S_IFMT) == S_IFCHR)    // 是否字符设备文件。
45 #define S_ISBLK(m)      (((m) & S_IFMT) == S_IFBLK)    // 是否块设备文件。
46 #define S_ISFIFO(m)     (((m) & S_IFMT) == S_IFIFO)    // 是否 FIFO 特殊文件。
47
48 // 文件访问权限：
49 #define S_IRWXU 00700    // 宿主可以读、写、执行/搜索（名称最后字母代表 User）。
50 #define S_IRUSR 00400    // 宿主读许可。
51 #define S_IWUSR 00200    // 宿主写许可。
52 #define S_IXUSR 00100    // 宿主执行/搜索许可。

```

```
42
43 #define S_IRWXG 00070 // 组成员可以读、写、执行/搜索（名称最后字母代表 Group）。
44 #define S_IRGRP 00040 // 组成员读许可。
45 #define S_IWGRP 00020 // 组成员写许可。
46 #define S_IXGRP 00010 // 组成员执行/搜索许可。
47
48 #define S_IRWXO 00007 // 其他人读、写、执行/搜索许可（名称最后字母 O 代表 Other）。
49 #define S_IROTH 00004 // 其他人读许可（最后 3 个字母代表 Other）。
50 #define S_IWOTH 00002 // 其他人写许可。
51 #define S_IXOTH 00001 // 其他人执行/搜索许可。
52
53 extern int chmod(const char *_path, mode_t mode); // 修改文件属性。
54 extern int fstat(int fildes, struct stat *stat_buf); // 取指定文件句柄的文件状态信息。
55 extern int mkdir(const char *_path, mode_t mode); // 创建目录。
56 extern int mkfifo(const char *_path, mode_t mode); // 创建管道文件。
57 extern int stat(const char *filename, struct stat *stat_buf); // 取指定文件名的文件状态信息。
58 extern mode_t umask(mode_t mask); // 设置属性屏蔽码。
59
60 #endif
61
```

---