

程序 14-4 linux/include/errno.h

```

1 #ifndef ERRNO_H
2 #define ERRNO_H
3
4 /*
5  * ok, as I hadn't got any other source of information about
6  * possible error numbers, I was forced to use the same numbers
7  * as minix.
8  * Hopefully these are posix or something. I wouldn't know (and posix
9  * isn't telling me - they want $$$ for their f***ing standard).
10 *
11 * We don't use the _SIGN cludge of minix, so kernel returns must
12 * see to the sign by themselves.
13 *
14 * NOTE! Remember to change strerror() if you change this file!
15 */
16 /*
17  * ok, 由于我没有得到任何其他有关出错号的资料, 我只能使用与 minix 系统
18  * 相同的出错号了。
19  * 希望这些是 POSIX 兼容的或者在一定程度上是这样的, 我不知道 (而且 POSIX
20  * 没有告诉我 - 要获得他们的混蛋标准需要出钱)。
21  *
22  * 我们没有使用 minix 那样的_SIGN 簇, 所以内核的返回值必须自己辨别正负号。
23  *
24  * 注意! 如果你改变该文件的话, 记着也要修改 strerror() 函数。
25 */
26 // 系统调用以及很多库函数返回一个特殊的值以表示操作失败或出错。这个值通常选择-1 或者
27 // 其他一些特定的值来表示。但是这个返回值仅说明错误发生了。 如果需要知道出错的类型,
28 // 就需要查看表示系统出错号的变量 errno。该变量即在 errno.h 文件中声明。在程序开始执
29 // 行时该变量值被初始化为 0。
30 extern int errno;
31
32 // 在出错时, 系统调用会把出错号放在变量 errno 中 (负值), 然后返回-1。因此程序若需要知
33 // 道具体错误号, 就需要查看 errno 的值。
34 #define ERROR          99          // 一般错误。
35 #define EPERM         1           // 操作没有许可。
36 #define ENOENT       2           // 文件或目录不存在。
37 #define ESRCH        3           // 指定的进程不存在。
38 #define EINTR       4           // 中断的系统调用。
39 #define EIO          5           // 输入/输出错。
40 #define ENXIO       6           // 指定设备或地址不存在。
41 #define E2BIG       7           // 参数列表太长。
42 #define ENOEXEC    8           // 执行程序格式错误。
43 #define EBADF       9           // 文件句柄(描述符)错误。
44 #define ECHILD     10          // 子进程不存在。
45 #define EAGAIN     11          // 资源暂时不可用。
46 #define ENOMEM    12          // 内存不足。
47 #define EACCES    13          // 没有许可权限。
48 #define EFAULT    14          // 地址错。
49 #define ENOTBLK   15          // 不是块设备文件。
50 #define EBUSY     16          // 资源正忙。
51 #define EEXIST    17          // 文件已存在。

```

```
37 #define EXDEV          18          // 非法连接。
38 #define ENODEV        19          // 设备不存在。
39 #define ENOTDIR       20          // 不是目录文件。
40 #define EISDIR        21          // 是目录文件。
41 #define EINVAL        22          // 参数无效。
42 #define ENFILE        23          // 系统打开文件数太多。
43 #define EMFILE        24          // 打开文件数太多。
44 #define ENOTTY        25          // 不恰当的 I/O 控制操作(没有 tty 终端)。
45 #define ETXTBSY       26          // 不再使用。
46 #define EFBIG         27          // 文件太大。
47 #define ENOSPC        28          // 设备已满(设备已经没有空间)。
48 #define ESPIPE        29          // 无效的文件指针重定位。
49 #define EROFS         30          // 文件系统只读。
50 #define EMLINK        31          // 连接太多。
51 #define EPIPE         32          // 管道错。
52 #define EDOM          33          // 域(domain)出错。
53 #define ERANGE        34          // 结果太大。
54 #define EDEADLK       35          // 避免资源死锁。
55 #define ENAMETOOLONG  36          // 文件名太长。
56 #define ENOLCK        37          // 没有锁定可用。
57 #define ENOSYS        38          // 功能还没有实现。
58 #define ENOTEMPTY     39          // 目录不空。
59
60 /* Should never be seen by user programs */
   /* 用户程序不应该见到下面这两中错误号 */
61 #define ERESTARTSYS   512         // 重新执行系统调用。
62 #define ERESTARTNOINTR 513       // 重新执行系统调用, 无中断。
63
64 #endif
65
```
