

程序 12-17 linux/fs/ioctl.c

```

1  /*
2  *  linux/fs/ioctl.c
3  *
4  *  (C) 1991  Linus Torvalds
5  */
6
7  #include <string.h>          // 字符串头文件。主要定义了一些有关字符串操作的嵌入函数。
8  #include <errno.h>          // 错误号头文件。包含系统中各种出错号。
9  #include <sys/stat.h>       // 文件状态头文件。含有文件状态结构 stat {} 和常量。
10
11 #include <linux/sched.h>    // 调度程序头文件，定义了任务结构 task_struct、任务 0 数据等。
12
13 extern int  tty_ioctl(int dev, int cmd, int arg);    // chr_drv/tty_ioctl.c, 第 133 行。
14 extern int  pipe_ioctl(struct m_inode *pino, int cmd, int arg); // fs/pipe.c, 第 118 行。
15
16 // 定义输入输出控制(ioctl)函数指针类型。
17 typedef int (*ioctl_ptr)(int dev, int cmd, int arg);
18
19 // 取系统中设备种数的宏。
20 #define NRDEVS ((sizeof (ioctl_table))/(sizeof (ioctl_ptr)))
21
22 // ioctl 操作函数指针表。
23 static ioctl_ptr ioctl_table[]={
24     NULL,          /* nodev */
25     NULL,          /* /dev/mem */
26     NULL,          /* /dev/fd */
27     NULL,          /* /dev/hd */
28     tty_ioctl,     /* /dev/ttyx */
29     tty_ioctl,     /* /dev/tty */
30     NULL,          /* /dev/lp */
31     NULL};        /* named pipes */
32
33 // 系统调用函数 - 输入输出控制函数。
34 // 该函数首先判断参数给出的文件描述符是否有效。然后根据对应 i 节点中文件属性判断文件
35 // 类型，并根据具体文件类型调用相关的处理函数。
36 // 参数: fd - 文件描述符; cmd - 命令码; arg - 参数。
37 // 返回: 成功则返回 0，否则返回出错码。
38 int sys_ioctl(unsigned int fd, unsigned int cmd, unsigned long arg)
39 {
40     struct file * filp;
41     int dev, mode;
42
43     // 首先判断给出的文件描述符的有效性。如果文件描述符超出可打开的文件数，或者对应描述
44     // 符的文件结构指针为空，则返回出错码退出。
45     if (fd >= NR_OPEN || !(filp = current->filp[fd]))
46         return -EBADF;
47     // 如果文件结构对应的是管道 i 节点，则根据进程是否有权操作该管道确定是否执行管道 IO
48     // 控制操作。若有权执行则调用 pipe_ioctl(), 否则返回无效文件错误码。
49     if (filp->f_inode->i_pipe)
50         return (filp->f_mode&1)?pipe_ioctl(filp->f_inode, cmd, arg):-EBADF;
51     // 对于其他类型文件，取对应文件的属性，并据此判断文件的类型。如果该文件既不是字符设

```

```
// 备文件，也不是块设备文件，则返回出错码退出。若是字符或块设备文件，则从文件的 i 节
// 点中取设备号。如果设备号大于系统现有的设备数，则返回出错号。
40     mode=filp->f_inode->i_mode;
41     if (!S_ISCHR(mode) && !S_ISBLK(mode))
42         return -EINVAL;
43     dev = filp->f_inode->i_zone[0];
44     if (MAJOR(dev) >= NRDEVS)
45         return -ENODEV;
// 然后根据 IO 控制表 ioctl_table 查得对应设备的 ioctl 函数指针，并调用该函数。如果该设
// 备在 ioctl 函数指针表中没有对应函数，则返回出错码。
46     if (!ioctl_table[MAJOR(dev)])
47         return -ENOTTY;
48     return ioctl_table[MAJOR(dev)](dev, cmd, arg);
49 }
50
```
