

程序 14-27 linux/include/linux/sys.h

```

1  /*
2  * Why isn't this a .c file? Enquiring minds...
3  */
4  /*
5  * 为什么这不是一个.c文件? 动动脑筋自己想想....
6  */
7
8  extern int sys_setup();           // 0 - 系统启动初始化设置函数。 (kernel/blk_drv/hd.c)
9  extern int sys_exit();           // 1 - 程序退出。 (kernel/exit.c)
10 extern int sys_fork();           // 2 - 创建进程。 (kernel/system_call.s)
11 extern int sys_read();          // 3 - 读文件。 (fs/read_write.c)
12 extern int sys_write();         // 4 - 写文件。 (fs/read_write.c)
13 extern int sys_open();          // 5 - 打开文件。 (fs/open.c)
14 extern int sys_close();         // 6 - 关闭文件。 (fs/open.c)
15 extern int sys_waitpid();       // 7 - 等待进程终止。 (kernel/exit.c)
16 extern int sys_creat();         // 8 - 创建文件。 (fs/open.c)
17 extern int sys_link();         // 9 - 创建一个文件的硬连接。 (fs/namei.c)
18 extern int sys_unlink();       // 10 - 删除一个文件名(或删除文件)。 (fs/namei.c)
19 extern int sys_execve();       // 11 - 执行程序。 (kernel/system_call.s)
20 extern int sys_chdir();        // 12 - 更改当前目录。 (fs/open.c)
21 extern int sys_time();        // 13 - 取当前时间。 (kernel/sys.c)
22 extern int sys_mknod();       // 14 - 建立块/字符特殊文件。 (fs/namei.c)
23 extern int sys_chmod();       // 15 - 修改文件属性。 (fs/open.c)
24 extern int sys_chown();       // 16 - 修改文件宿主和所属组。 (fs/open.c)
25 extern int sys_break();       // 17 - (kernel/sys.c)*
26 extern int sys_stat();        // 18 - 使用路径名取文件状态信息。 (fs/stat.c)
27 extern int sys_lseek();       // 19 - 重新定位读/写文件偏移。 (fs/read_write.c)
28 extern int sys_getpid();      // 20 - 取进程 id。 (kernel/sched.c)
29 extern int sys_mount();       // 21 - 安装文件系统。 (fs/super.c)
30 extern int sys_umount();      // 22 - 卸载文件系统。 (fs/super.c)
31 extern int sys_setuid();      // 23 - 设置进程用户 id。 (kernel/sys.c)
32 extern int sys_getuid();      // 24 - 取进程用户 id。 (kernel/sched.c)
33 extern int sys_stime();       // 25 - 设置系统时间日期。 (kernel/sys.c)*
34 extern int sys_ptrace();      // 26 - 程序调试。 (kernel/sys.c)*
35 extern int sys_alarm();       // 27 - 设置报警。 (kernel/sched.c)
36 extern int sys_fstat();       // 28 - 使用文件句柄取文件的状态信息。 (fs/stat.c)
37 extern int sys_pause();       // 29 - 暂停进程运行。 (kernel/sched.c)
38 extern int sys_utime();       // 30 - 改变文件的访问和修改时间。 (fs/open.c)
39 extern int sys_stty();        // 31 - 修改终端行设置。 (kernel/sys.c)*
40 extern int sys_gtty();        // 32 - 取终端行设置信息。 (kernel/sys.c)*
41 extern int sys_access();      // 33 - 检查用户对一个文件的访问权限。 (fs/open.c)
42 extern int sys_nice();        // 34 - 设置进程执行优先权。 (kernel/sched.c)
43 extern int sys_ftime();       // 35 - 取日期和时间。 (kernel/sys.c)*
44 extern int sys_sync();        // 36 - 同步高速缓冲与设备中数据。 (fs/buffer.c)
45 extern int sys_kill();        // 37 - 终止一个进程。 (kernel/exit.c)
46 extern int sys_rename();      // 38 - 更改文件名。 (kernel/sys.c)*
47 extern int sys_mkdir();       // 39 - 创建目录。 (fs/namei.c)
48 extern int sys_rmdir();       // 40 - 删除目录。 (fs/namei.c)
49 extern int sys_dup();         // 41 - 复制文件句柄。 (fs/fcntl.c)
50 extern int sys_pipe();        // 42 - 创建管道。 (fs/pipe.c)
51 extern int sys_times();       // 43 - 取运行时间。 (kernel/sys.c)
52 extern int sys_prof();        // 44 - 程序执行时间区域。 (kernel/sys.c)*

```

```

50 extern int sys brk(); // 45 - 修改数据段长度。 (kernel/sys.c)
51 extern int sys setgid(); // 46 - 设置进程组 id。 (kernel/sys.c)
52 extern int sys getgid(); // 47 - 取进程组 id。 (kernel/sched.c)
53 extern int sys signal(); // 48 - 信号处理。 (kernel/signal.c)
54 extern int sys geteuid(); // 49 - 取进程有效用户 id。 (kernel/sched.c)
55 extern int sys getegid(); // 50 - 取进程有效组 id。 (kernel/sched.c)
56 extern int sys acct(); // 51 - 进程记帐。 (kernel/sys.c)*
57 extern int sys phys(); // 52 - (kernel/sys.c)*
58 extern int sys lock(); // 53 - (kernel/sys.c)*
59 extern int sys ioctl(); // 54 - 设备输入输出控制。 (fs/ioctl.c)
60 extern int sys fcntl(); // 55 - 文件句柄控制操作。 (fs/fcntl.c)
61 extern int sys mpx(); // 56 - (kernel/sys.c)*
62 extern int sys setpgid(); // 57 - 设置进程组 id。 (kernel/sys.c)
63 extern int sys ulimit(); // 58 - 统计进程使用资源情况。 (kernel/sys.c)
64 extern int sys uname(); // 59 - 显示系统信息。 (kernel/sys.c)
65 extern int sys umask(); // 60 - 取默认文件创建属性码。 (kernel/sys.c)
66 extern int sys chroot(); // 61 - 改变根目录。 (fs/open.c)
67 extern int sys ustat(); // 62 - 取文件系统信息。 (fs/open.c)
68 extern int sys dup2(); // 63 - 复制文件句柄。 (fs/fcntl.c)
69 extern int sys getppid(); // 64 - 取父进程 id。 (kernel/sched.c)
70 extern int sys getpgrp(); // 65 - 取进程组 id, 等于 getpgid(0)。 (kernel/sys.c)
71 extern int sys setsid(); // 66 - 在新会话中运行程序。 (kernel/sys.c)
72 extern int sys sigaction(); // 67 - 改变信号处理过程。 (kernel/signal.c)
73 extern int sys sgetmask(); // 68 - 取信号屏蔽码。 (kernel/signal.c)
74 extern int sys ssetmask(); // 69 - 设置信号屏蔽码。 (kernel/signal.c)
75 extern int sys setreuid(); // 70 - 设置真实与/或有效用户 id。 (kernel/sys.c)
76 extern int sys setregid(); // 71 - 设置真实与/或有效组 id。 (kernel/sys.c)
77 extern int sys sigpending(); // 73 - 检查暂未处理的信号。 (kernel/signal.c)
78 extern int sys sigsuspend(); // 72 - 使用新屏蔽码挂起进程。 (kernel/signal.c)
79 extern int sys sethostname(); // 74 - 设置主机名。 (kernel/sys.c)
80 extern int sys setrlimit(); // 75 - 设置资源使用限制。 (kernel/sys.c)
81 extern int sys getrlimit(); // 76 - 取得进程使用资源的限制。 (kernel/sys.c)
82 extern int sys getrusage(); // 77 -
83 extern int sys gettimeofday(); // 78 - 获取当日时间。 (kernel/sys.c)
84 extern int sys settimeofday(); // 79 - 设置当日时间。 (kernel/sys.c)
85 extern int sys getgroups(); // 80 - 取得进程所有组标识号。 (kernel/sys.c)
86 extern int sys setgroups(); // 81 - 设置进程组标识号数组。 (kernel/sys.c)
87 extern int sys select(); // 82 - 等待文件描述符状态改变。 (fs/select.c)
88 extern int sys symlink(); // 83 - 建立符号链接。 (fs/namei.c, 767)
89 extern int sys lstat(); // 84 - 取符号链接文件状态。 (fs/stat.c, 47)
90 extern int sys readlink(); // 85 - 读取符号链接文件信息。 (fs/stat.c, 69)
91 extern int sys uselib(); // 86 - 选择共享库。 (fs/exec.c, 42)

```

92

// 系统调用函数指针表。用于系统调用中断处理程序(int 0x80), 作为跳转表。

```

93 fn_ptr sys call table[] = { sys setup, sys exit, sys fork, sys read,
94 sys write, sys open, sys close, sys waitpid, sys creat, sys link,
95 sys unlink, sys execve, sys chdir, sys time, sys mknod, sys chmod,
96 sys chown, sys break, sys stat, sys lseek, sys getpid, sys mount,
97 sys umount, sys setuid, sys getuid, sys stime, sys ptrace, sys alarm,
98 sys fstat, sys pause, sys utime, sys stty, sys gtty, sys access,
99 sys nice, sys ftime, sys sync, sys kill, sys rename, sys mkdir,
100 sys rmdir, sys dup, sys pipe, sys times, sys prof, sys brk, sys setgid,
101 sys getgid, sys signal, sys geteuid, sys getegid, sys acct, sys phys,

```

```
102 sys lock, sys ioctl, sys fcntl, sys mpx, sys setpgid, sys ulimit,
103 sys uname, sys umask, sys chroot, sys ustat, sys dup2, sys getppid,
104 sys getpgrp, sys setsid, sys sigaction, sys sgetmask, sys ssetmask,
105 sys setreuid, sys setregid, sys sigsuspend, sys sigpending, sys sethostname,
106 sys setrlimit, sys getrlimit, sys getrusage, sys gettimeofday,
107 sys settimeofday, sys getgroups, sys setgroups, sys select, sys symlink,
108 sys lstat, sys readlink, sys uselib };
109
110 /* So we don't have to do any more manual updating... */
111 /* 下面这样定义后，我们就无需手工更新系统调用数目了 */
112 int NR_syscalls = sizeof(sys call table)/sizeof(fn ptr);
```
