

程序 15-8 linux/lib/open.c

```
1 /*
2  * linux/lib/open.c
3  *
4  * (C) 1991 Linus Torvalds
5  */
6
7 #define LIBRARY
8 #include <unistd.h> // Linux 标准头文件。定义了各种符号常数和类型，并声明了各种函数。
// 如定义了__LIBRARY__，则还含系统调用号和内嵌汇编_syscall10()等。
9 #include <stdarg.h> // 标准参数头文件。以宏的形式定义变量参数列表。主要说明了一个
// 类型(va_list)和三个宏(va_start, va_arg 和 va_end)，用于
// vsprintf、vprintf、vfprintf 函数。
10
11 // 打开文件函数。
12 // 打开并有可能创建一个文件。
13 // 参数: filename - 文件名; flag - 文件打开标志; ...
14 // 返回: 文件描述符, 若出错则置出错码, 并返回-1。
15 // 第 13 行定义了一个寄存器变量 res, 该变量将被保存在一个寄存器中, 以便于高效访问和操作。
16 // 若想指定存放的寄存器 (例如 eax), 那么可以把该句写成 "register int res asm("ax");"。
17 int open(const char * filename, int flag, ...)
18 {
19     register int res;
20     va_list arg;
21
22     // 利用 va_start() 宏函数, 取得 flag 后面参数的指针, 然后调用系统中断 int 0x80, 功能 open 进行
23     // 文件打开操作。
24     // %0 - eax (返回的描述符或出错码); %1 - eax (系统中断调用功能号 __NR_open);
25     // %2 - ebx (文件名 filename); %3 - ecx (打开文件标志 flag); %4 - edx (后随参数文件属性 mode)。
26     va_start(arg, flag);
27     __asm__ ("int $0x80"
28             : "=a" (res)
29             : "" (__NR_open), "b" (filename), "c" (flag),
30               "d" (va_arg(arg, int)));
31     // 系统中断调用返回值大于或等于 0, 表示是一个文件描述符, 则直接返回之。
32     if (res >= 0)
33         return res;
34     // 否则说明返回值小于 0, 则代表一个出错码。设置该出错码并返回-1。
35     errno = -res;
36     return -1;
37 }
38
```