

## 程序 8-11 linux/kernel/printk.c

---

```
1 /*
2  * linux/kernel/printk.c
3  *
4  * (C) 1991 Linus Torvalds
5  */
6
7 /*
8  * When in kernel-mode, we cannot use printf, as fs is liable to
9  * point to 'interesting' things. Make a printf with fs-saving, and
10 * all is well.
11 */
12 /*
13  * 当处于内核模式时，我们不能使用 printf，因为寄存器 fs 指向其他不感兴趣
14  * 的地方。自己编制一个 printf 并在使用前保存 fs，一切就解决了。
15 */
16 // 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
17 // va_start、va_arg 和 va_end，用于 vsprintf、vprintf、vfprintf 函数。
18 #include <stdarg.h>
19 #include <stddef.h> // 标准定义头文件。定义了 NULL，offsetof(TYPE, MEMBER)。
20
21 #include <linux/kernel.h> // 内核头文件。含有一些内核常用函数的原形定义。
22
23 static char buf[1024]; // 显示用临时缓冲区。
24
25 // 函数 vsprintf() 定义在 linux/kernel/vsprintf.c 中 92 行开始处。
26 extern int vsprintf(char * buf, const char * fmt, va_list args);
27
28 // 内核使用的显示函数。
29 int printk(const char *fmt, ...)
30 {
31     va_list args; // va_list 实际上是一个字符指针类型。
32     int i;
33
34     // 运行参数处理开始函数。然后使用格式串 fmt 将参数列表 args 输出到 buf 中。返回值 i
35     // 等于输出字符串的长度。再运行参数处理结束函数。最后调用控制台显示函数并返回显示
36     // 字符数。
37     va_start(args, fmt);
38     i=vsprintf(buf, fmt, args);
39     va_end(args);
40     console_print(buf); // chr_drv/console.c, 第 995 行开始。
41     return i;
42 }
```

---