

程序 14-14 linux/include/asm/io.h

```
1  //// 硬件端口字节输出函数。
2  // 参数: value - 欲输出字节; port - 端口。
3  1 #define outb(value, port) \
4  2 __asm__ ("outb %%a1, %%dx"::"a" (value), "d" (port))
5
6  //// 硬件端口字节输入函数。
7  // 参数: port - 端口。返回读取的字节。
8  5 #define inb(port) ({ \
9  6 unsigned char _v; \
10 7 __asm__ volatile ("inb %%dx, %%a1"::"a" (_v):"d" (port)); \
11 8 _v; \
12 9 })
13
14 //// 带延迟的硬件端口字节输出函数。使用两条跳转语句来延迟一会。
15 // 参数: value - 欲输出字节; port - 端口。
16 11 #define outb_p(value, port) \
17 12 __asm__ ("outb %%a1, %%dx\n" \
18 13         "\tjmp 1f\n" \
19 14         "1: \tjmp 1f\n" \
20 15         "1: :: "a" (value), "d" (port))
21 // 向前跳转到标号 1 处 (即下一条语句处)。
22 // 向前跳转到标号 1 处。
23
24 //// 带延迟的硬件端口字节输入函数。使用两条跳转语句来延迟一会。
25 // 参数: port - 端口。返回读取的字节。
26 17 #define inb_p(port) ({ \
27 18 unsigned char _v; \
28 19 __asm__ volatile ("inb %%dx, %%a1\n" \
29 20                 "\tjmp 1f\n" \
30 21                 "1: \tjmp 1f\n" \
31 22                 "1: :: "a" (_v): "d" (port)); \
32 23 _v; \
33 24 })
34
35
```
