

```
1. printk("i2c_test: write (io) successfully\n");
```

这对应到什么log level呢？

该message将使用default_message_loglevel指定的log level

printk()调用vprintk_emit()的代码如下

```
1. vprintk_emit(0, -1, NULL, 0, fmt, args);
```

即level = -1

in kernel/printk/printk.c

```

1.  asmlinkage int vprintk_emit(int facility, int level,
2.                               const char *dict, size_t dictlen,
3.                               const char *fmt, va_list args)
4.  {
5.
6.      .....
7.
8.
9.      /* strip kernel syslog prefix and extract log level or control flags */
10.     if (facility == 0) { ①
11.         int kern_level = printk_get_level(text); ②
12.
13.         if (kern_level) { ③
14.             const char *end_of_header = printk_skip_level(text);
15.             switch (kern_level) {
16.                 case '0' ... '7':
17.                     if (level == -1)
18.                         level = kern_level - '0';
19.
20.                 case 'd': /* KERN_DEFAULT */
21.                     lflags |= LOG_PREFIX;
22.             }
23.
24.             /*
25.              * No need to check length here because vsnprintf
26.              * put '\0' at the end of the string. Only valid and
27.              * newly printed level is detected.
28.              */
29.
30.             text_len -= end_of_header - text;
31.             text = (char *)end_of_header;
32.         }
33.     }
34.
35.
36.
37.     if (level == -1) ④
38.         level = default_message_loglevel;
39.
40.     .....
41.
42. }

```

①

facility = 0, 成立

②

buffer ==> "i2c_test: write (io) successfully\n"

```
1. static inline int printk_get_level(const char *buffer)
2. {
3.     if (buffer[0] == KERN_SOH_ASCII && buffer[1]) {
4.         switch (buffer[1]) {
5.             case '0' ... '7':
6.             case 'd': /* KERN_DEFAULT */
7.                 return buffer[1];
8.             }
9.         }
10.
11.     return 0;
12. }
```

返回值为0,数值0

③

kern_level = 0 , 所以if不成立

④

level没有修改 , 所以还是为-1,这样

```
1. level = default_message_loglevel;
```