Documentation/printk-formats.txt中有对printk支持的format specificer的描述。

```c
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <asm/page.h>

/*
 * vsprintf() is declared in kernel.h
 */

void test_vsprintf(void);

static char buffer[1000];

void test_vsprintf()
{
    /*
    * %pS output the name of a text symbol with offset
    *
    * "printk+0x0/0x73"
    */
    sprintf(buffer, "%pS\n", printk);
    printk("%s\n", buffer);

    /*
    * %ps output the name of a text symbol without offset
    *
    * "printk"
    */
    sprintf(buffer, "%ps\n", printk);
    printk("%s\n", buffer);

    /*
    * %pF output the name of a function pointer with its offset
    *
    * "printk+0x0/0x73" (对unction而言，同%pS没啥区别)
    */
    sprintf(buffer, "%pF\n", printk);
    printk("%s\n", buffer);

    /*
    * %pK Kernel Pointers
    *
    * "ffffffff8118cbb7"
    */
    sprintf(buffer, "%pK\n", printk);
    printk("%s\n", buffer);

    /*
    * %pa physical Pointers
    *
    * "000000000118cbb7"
    */
    sprintf(buffer, "%pK\n", __pa(printk));
```

```
54.     printk("%s\n", buffer);
55.
56.     /*
57.     * %p Pointers
58.     *
59.     * "ffff88000118cbb7"
60.     */
61.     sprintf(buffer, "%pK\n", __va(__pa(printk)));
62.     printk("%s\n", buffer);
63.
64.     /*
65.     * 奇怪, "ffff88000118cbb7" v.s. "ffffffff8118cbb7"
66.     * ???
67.     * $ sudo cat /boot/System.map-4.4.8-040408-lowlatency | grep " printk$"
68.     * ffffffff8118cbb7 T printk
69.     */
70.
71.     /*
72.     * %pS output the name of a text symbol with offset
73.     * output the symbol of "ffff88000118cbb7"
74.     *
75.     * __va(__pa(symbol) 并不能得到该symbol的真实virtual address
76.     */
77.     sprintf(buffer, "%pS\n", __va(__pa(printk)));
78.     printk("%s\n", buffer);
79.
80.     /*
81.     IPv4 addresses:
82.
83.         %pI4    1.2.3.4
84.         %pi4    001.002.003.004
85.         %p[Ii]4[hnbl]
86.
87.         For printing IPv4 dot-separated decimal addresses. The 'I4' and 'i4'
88.         specifiers result in a printed address with ('i4') or without ('I4')
89.         leading zeros.
90.
91.         The additional 'h', 'n', 'b', and 'l' specifiers are used to specify
92.         host, network, big or little endian order addresses respectively. Wh
    ere
93.         no specifier is provided the default network/big endian order is use
    d.
94.
95.         "120.86.52.18, 120.086.052.018"
96.     */
97.     uint32_t ip4 = 0x12345678;
98.     sprintf(buffer, "%pI4, %pi4\n", &ip4, &ip4);
99.     printk("%s\n", buffer);
100.
101.    /*
102.    DMA addresses types dma_addr_t:
103.
104.    %pad    0x01234567 or 0x0123456789abcdef
105.
```

```
106.      For printing a dma_addr_t type which can vary based on build options,
107.      regardless of the width of the CPU data path. Passed by reference.
108.
109.      "0x0000000000123456"
110.      */
111.
112.      dma_addr_t dma_addr = 0x123456;
113.      sprintf(buffer, "%pad\n", &dma_addr);
114.      printk("%s\n", buffer);
115.
116.  }
117.
118.  static int __init vsprintf_start(void)
119.  {
120.      printk(KERN_INFO "Loading vsprintf module...\n");
121.
122.      test_vsprintf();
123.
124.      return  0;
125.  }
126.
127.  static void __exit vsprintf_end(void)
128.  {
129.      printk(KERN_INFO "Goodbye Mr.\n");
130.  }
131.
132.  module_init(vsprintf_start);
133.  module_exit(vsprintf_end);
```