

in include/linux/printk.h

```
1.  /* If you are writing a driver, please use dev_dbg instead */
2.  #if defined(CONFIG_DYNAMIC_DEBUG)
3.  /* dynamic_pr_debug() uses pr_fmt() internally so we don't need it here */
4.  #define pr_debug(fmt, ...) \
5.      dynamic_pr_debug(fmt, ##__VA_ARGS__)
6.  #elif defined(DEBUG)
7.  #define pr_debug(fmt, ...) \
8.      printk(KERN_DEBUG pr_fmt(fmt), ##__VA_ARGS__)
9.  #else
10. #define pr_debug(fmt, ...) \
11.     no_printk(KERN_DEBUG pr_fmt(fmt), ##__VA_ARGS__)
12. #endif
```

在没有enable CONFIG_DYNAMIC_DEBUG的情况下，需要enable DEBUG macro.

比如对drivers/base目录下code而言

in drivers/base/Makefile

```
ccflags-$(CONFIG_DEBUG_DRIVER) := -DDEBUG
```

如果要调试kernel initialization的code

in include/linux/printk.h

```
#define CONSOLE_LOGLEVEL_DEFAULT 7
==>
#define CONSOLE_LOGLEVEL_DEFAULT 8
```

in .config

```
CONFIG_MESSAGE_LOGLEVEL_DEFAULT=8
```

如果是调试dynamic load module

```
echo 8 8 8 8 > /proc/sys/kernel/printk
```

Note

```
#define KERN_DEBUG KERN_SOH "7"
in kernel/printk/printk.c/call_console_drivers()
```

```
1.     if (level >= console_loglevel && !ignore_loglevel)
2.         return;
```

如果level = 7, console_loglevel = 7, printk()还是不会有输出的；只有令 console_loglevel > 7, 比如 8, call_console_drivers()才会放下运行, printk()才会有输出！