# 打印级别

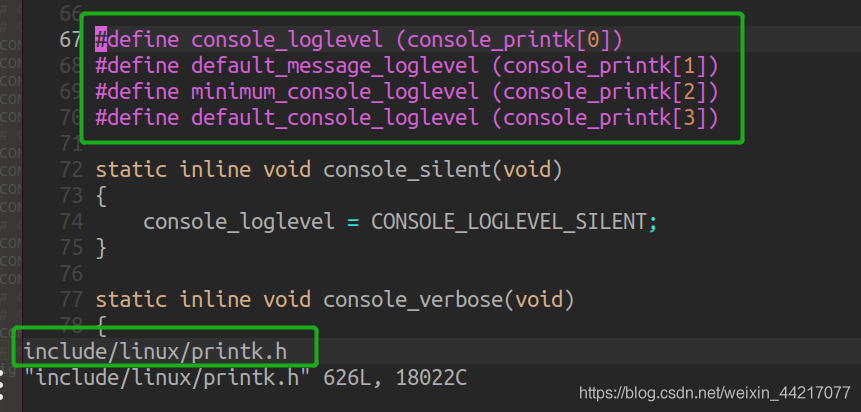
## 内核设置

在在[头文件](https://so.csdn.net/so/search?q=%E5%A4%B4%E6%96%87%E4%BB%B6&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_44217077/article/details/_blank) include/linux/kern\_levels.h 中定义了0~7， 8个打印级别数字越小，打印级别越高。

|  |
| --- |
| #define KERN\_EMERG KERN\_SOH "0" /\* system is unusable \*/  #define KERN\_ALERT KERN\_SOH "1" /\* action must be taken immediately \*/  #define KERN\_CRIT KERN\_SOH "2" /\* critical conditions \*/  #define KERN\_ERR KERN\_SOH "3" /\* error conditions \*/  #define KERN\_WARNING KERN\_SOH "4" /\* warning conditions \*/  #define KERN\_NOTICE KERN\_SOH "5" /\* normal but significant condition \*/  #define KERN\_INFO KERN\_SOH "6" /\* informational \*/  #define KERN\_DEBUG KERN\_SOH "7" /\* debug-level messages \*/ |

## 用户态设置

|  |
| --- |
| $ cat /proc/sys/kernel/printk  4 4 1 7  控制台打印级别 消息的默认打印级别 控制台最大打印级别 控制台最小级别 |



注意：**打印消息的级别大于控制台的打印级别时**，才能出现在终端

并且：数字越大，级别越低

修改：

|  |
| --- |
| $vi rootfs/etc/init.d/rcS  $echo 4 3 1 7 >/proc/sys/kernel/printk #意味着使用printk不指定打印级别的时候，都会出现在终端打印出来 |

# 查看打印信息

|  |
| --- |
| 查看打印信息$ sudo dmesg  清理打印信息  $ sudo dmesg -C #清理所有的内核打印信息  $ sudo dmesg -c #先回显再清理所有的内核打印信息 |

# 第三章 经典的printk 写法

|  |
| --- |
| printk("[lynn--%s@%d]: addr:0x%x \n",\_\_func\_\_,\_\_LINE\_\_,obj->client->addr);  printk("Function:%s,%d,val:%d,%p\n",\_\_func\_\_,\_\_LINE\_\_,val,\_\_builtin\_return\_address(0));  \_\_func\_\_ ：函数名；  \_\_LINE\_\_　：行数；  [ 13.531443] <0>.(0)[1:swapper/0][max--stk3x1x\_i2c\_probe@5031]: addr:0x48；  \_\_builtin\_return\_address(0) ： 变量地址；  //%p 指针变量； |

参考：https://www.cnblogs.com/qiynet/p/5303473.html

# 第四章 内核demo

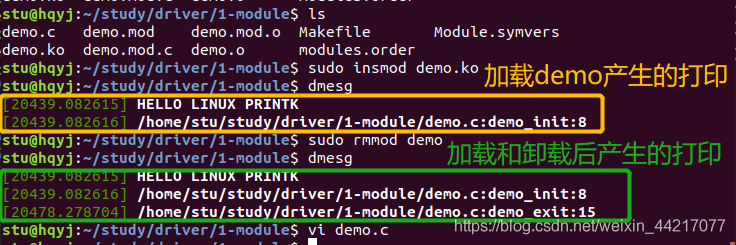
内核打印模块：

|  |
| --- |
| #include<linux/init.h>  #include<linux/module.h>  static int \_\_init demo\_init(void)  {  **printk(KERN\_WARNING "HELLO LINUX PRINTK\n");**  **printk("%s:%s:%d\n", \_\_FILE\_\_, \_\_func\_\_, \_\_LINE\_\_); //没有指定打印级别，使用默认的消息打印级别**  return 0;  }  static void \_\_exit demo\_exit(void)  {  printk("%s:%s:%d\n", \_\_FILE\_\_, \_\_func\_\_, \_\_LINE\_\_);//没有指定打印级别，使用默认的消息打印级别  }  module\_init(demo\_init);  module\_exit(demo\_exit);  MODULE\_LICENSE("GPL"); |

加载ko：

|  |
| --- |
| $ sudo insmod demo.ko  $ dmesg  [20439.082615] HELLO LINUX PRINTK  [20439.082616] /home/stu/study/driver/1-module/demo.c:demo\_init:8  $ sudo rmmod demo  $ dmesg  [20439.082615] HELLO LINUX PRINTK  [20439.082616] /home/stu/study/driver/1-module/demo.c:demo\_init:8  [20478.278704] /home/stu/study/driver/1-module/demo.c:demo\_exit:15 |

测试：



# Android相关设置

## 5.1 userdebug

|  |
| --- |
| 在/bootable/bootloader/lk/app/mt\_boot/mt\_boot.c里修改：  if (!has\_set\_p2u) {  #ifdef USER\_BUILD  sprintf(cmdline,"%s%s",cmdline," printk.disable\_uart=1"); // printk.disable\_uart=1 改为0  #else  sprintf(cmdline,"%s%s",cmdline," printk.disable\_uart=0 ddebug\_query=\"file \*mediatek\* +p ; file \*gpu\* =\_\"");  #endif |

参考：<https://www.cnblogs.com/qiynet/p/5945005.html>

## 5.2 屏蔽LOGD\LOGI等

|  |
| --- |
| #cat /proc/sys/kernel/printk  7 4 1 7  对照上面的信息，就知道现在console中的级别为7，为了屏蔽部分信息，可以通过如下命令  #echo 4 > /proc/sys/kernel/printk |

参考：<https://www.cnblogs.com/qiynet/p/4418101.html>