1. 折半打印方法。

可以快速的定义错误

2.在调试的时候可以这样

/\*一般我们可以这样做#\*/

1.#define DBG\_PRINTK printk

2.#define DBG\_PRINTK (x…)//x表示宏的参数时可变的…表示的空的参数

//DBG\_PRINTK("%s %s %d\n",\_\_FILE\_\_,\_\_FUNCTION\_\_,\_\_LINE\_\_);

不需要的话就将宏定义定义为2.#define DBG\_PRINTK (...x)

还可以这样做：

我们知道printk有打印级别的，在我们上节课的学习中了解只有我们的打印的级别小于console\_loglevel才会打印可以参考代码。

KERN\_DEBUG 是7

printk(KERN\_DEBUG "%s %s %d\n", \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_);//所以不会打印

cat /proc/sys/kernel/printk

会打印出来: 7 4 1 7 console\_loglevel默认是7

修改这个文件：

echo “ 8 4 1 7 ” >/proc/sys/kernel/printk 改为8后就可以打印了

所以我们可以通过上边的两种方式可以进行打印。

一般的串口打印比较慢影响启动时间。我们如何将所有的打印去掉。

console\_loglevel 改为 0就行了么有比他还小得了所以都不打印了

在内核的文档中kernel-parameters.txt 有loglevel的介绍

在源码中搜索：

loglevel=会找到：如下在init main 下：

static int \_\_init loglevel(char \*str)

{

get\_option(&str, &console\_loglevel);

return 1;

}

\_\_setup("loglevel=", loglevel);

uboot 将传进来的参数 loglevel 得到一个整数赋给console\_loglevel

所以我们在uboot 传参的时候将loglevel设置为0

set bootargs loglevel=0 concole=tty….保持以前的设置

boot 启动开发板

所以的打印信息都没有了。但是可以使用dmesg 将所有的信息打印出来。

还有如下的参数：

static int \_\_init debug\_kernel(char \*str)

{

if (\*str)

return 0;

console\_loglevel = 10;

return 1;

}

static int \_\_init quiet\_kernel(char \*str)

{

if (\*str)

return 0;

console\_loglevel = 4;

return 1;

}

\_\_setup("debug", debug\_kernel);

\_\_setup("quiet", quiet\_kernel);//默认的是4

set bootargs debug concole=tty….保持以前的设置 //所有的信息都会被打印