

Tekkaman Ninja

tekkamanninja.blog.chinaunix.net

Linux我的梦想，我的未来！ 本博客的原创文章的内容会不定期更新或修正错误！转载文章都会注明出处，若有侵权，请即时同我联系，我一定马上删除！！原创文章版权所有！如需转载，请注明出处： tekkamanninja.blog.chinaunix.net ，谢谢合作！！ 拒绝一切广告性质的评论，一经发现立即举报并删除！

首页 | 博文目录 | 关于我



tekkamanninj

博客访问： 75905
博文数量： 263
博客积分： 15936
博客等级： 上将
技术积分： 13951
用户 组： 普通用户
注册时间： 2007-03-27 11:22

加关注 短消息

论坛 加好友

个人简介

Fedora-ARM

文章分类

- 全部博文（263）
- Red Hat（2）

代码管理（6）

感悟（3）

Linux调试技术（2）

MaxWit（1）

Linux设备驱动程（41）

Android（20）

neo freerunner（2）

计算机硬件技术（9）

网络（WLAN or LA（8）

励志（7）

ARM汇编语言（1）

Linux操作系统的（15）

Linux内核研究（38）

ARM-Linux应用程（19）

建立根文件系统（4）

Linux内核移植（14）

Bootloader（45）

建立ARM-Linux交（7）

未分配的博文（19）

文章存档

2014年（1）

Linux设备驱动程序学习（2）-调试技术

2007-10-25 13:39:12

分类： LINUX

Linux设备驱动程序学习（2）-调试技术

今天进入《Linux设备驱动程序（第3版）》第四章调试技术的学习。

一、内核中的调试支持

在前面已经建议过：学习编写驱动程序要构建安装自己的内核（标准主线内核）。最重要的原因之一是：内核开发者已经建立了多项用于调试的功能。但是由于这些功能会造成额外的输出，并导致能下降，因此发行版厂商通常会禁止发行版内核中的调试功能。

为了实现内核调试，我在内核配置上增加了几项：

```
Kernel hacking --->
[*] Magic SysRq key
[*] Kernel debugging
[*] Debug slab memory allocations
[*] Spinlock and rw-lock debugging: basic checks
[*] Spinlock debugging: sleep-inside-spinlock checking
[*] Compile the kernel with debug info
[*] Magic SysRq key

Device Drivers --->
Generic Driver Options --->
[*] Driver Core verbose debug messages

General setup --->
[*] Configure standard kernel features (for small systems) --->
[*] Load all symbols for debugging/ksymoops
```

书上介绍的还有其他配置，有的我不需要，或是s3c2440不支持，菜单里看不见。

二、通过打印调试

（1）printk

首先，printk有8个loglevel,定义在<linux/kernel.h>中：

```
#define KERN_EMERG    "<0>"    /* system is unusable */
#define KERN_ALERT    "<1>"    /* action must be taken immediately*/
#define KERN_CRIT     "<2>"    /* critical conditions */
#define KERN_ERR      "<3>"    /* error conditions */
#define KERN_WARNING   "<4>"    /* warning conditions */
#define KERN_NOTICE    "<5>"    /* normal but significant condition */
#define KERN_INFO      "<6>"    /* informational */
#define KERN_DEBUG     "<7>"    /* debug-level messages */
```

未指定优先级的默认级别定义在/kernel/printk.c中：

```
#define DEFAULT_MESSAGE_LOGLEVEL 4 /* KERN_WARNING */
```

当优先级的值小于console_loglevel这个整数变量的值，信息才能显示出来。而console_loglevel的初始值DEFAULT_CONSOLE_LOGLEVEL也定义在/kernel/printk.c中：

```
#define DEFAULT_CONSOLE_LOGLEVEL 7 /* anything MORE serious than KERN_DEBUG */
```

2013年 (3)
2012年 (61)
2011年 (66)
2010年 (27)
2009年 (30)
2008年 (23)
2007年 (52)

我的朋友



小蜗牛快



cfm5538



jikaishi



shizhenc



pxy05215



李怀远



yan19900



wkm81018



xiousi

最近访客



apang199



appcount



zaichu



lhxzui



小蜗牛快



小尾巴鱼



erain_30



hushup



wilfred_

订阅

推荐博文

- linux 3.x的 通用时钟架构 ...
- SCN的相关解析
- Flash驱动学习
- 浅谈nagios之state type和 no...
- DB2 (Linux 64位) 安装教程...
- insert语句造成latch:library...
- 2014.06.13 网络公开课《让我...
- MySQL Slave异常关机的处理 (...)
- 巧用shell脚本分析数据库用户...
- 查询linux, HP-UX的cpu信息...

热词专题

- linux系统权限修复——学生误...
- Modbus协议使用
- linux
- busybox原理
- php环境搭建教程

而在运行是改变console_loglevel的程序 (《Linux设备驱动程序 (第3版)》提供, 仅支持glibc2.0之前的版本) 如下:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#define __LIBRARY__ /* _syscall3 and friends are only available through this */
#include <linux/unistd.h>
/* define the system call, to override the library function */
_syscall3(int, syslog, int, type, char *, bufp, int, len);

int main(int argc, char **argv)
{
    int level;
    if (argc==2) {
        level = atoi(argv[1]); /* the chosen console */
    } else {
        fprintf(stderr, "%s: need a single arg\n", argv[0]); exit(1);
    }
    if (syslog(8, NULL, level) < 0) {
        fprintf(stderr, "%s: syslog(setlevel): %s\n",
            argv[0], strerror(errno));
        exit(1);
    }
    exit(0);
}
```

最关键的“**syslog(8, NULL, level)**”是控制内核printk的ring buffer的系统调用syslog的glibc函数接口。通过在ARM9板上的实验表明: 程序是ok的! 我用Hello world模块做了实验, 现象和书上的一致。

```
[Tekkaman2440@SBC2440V4]#cd /tmp/
[Tekkaman2440@SBC2440V4]#./setlevel 1
[Tekkaman2440@SBC2440V4]#cd /lib/modules/
[Tekkaman2440@SBC2440V4]#insmod hello.ko
[Tekkaman2440@SBC2440V4]#rmmod hello
[Tekkaman2440@SBC2440V4]#cd /tmp/
[Tekkaman2440@SBC2440V4]#./setlevel 7
[Tekkaman2440@SBC2440V4]#cd /lib/modules/
[Tekkaman2440@SBC2440V4]#insmod hello.ko
Hello, Tekkaman Ninja !
[Tekkaman2440@SBC2440V4]#rmmod hello
Goodbye, Tekkaman Ninja !
Love Linux !Love ARM ! Love KeKe !
[Tekkaman2440@SBC2440V4]#
```

由于 **glibc 2.0** 中, 由于词汇 **syslog** 使用过于广泛, 这个函数的名称被修改成 **klogctl**, 新代码请参阅我的另一篇博文: 《内核日志及**printk**结构浅析》

还有通过对**/proc/sys/kernel/printk**的访问来改变console_loglevel的值:

```
[Tekkaman2440@SBC2440V4]#echo 1 > /proc/sys/kernel/printk
[Tekkaman2440@SBC2440V4]#cat /proc/sys/kernel/printk
1          4          1          7
[Tekkaman2440@SBC2440V4]#insmod hello.ko
[Tekkaman2440@SBC2440V4]#rmmod hello
[Tekkaman2440@SBC2440V4]#echo 7 > /proc/sys/kernel/printk
[Tekkaman2440@SBC2440V4]#cat /proc/sys/kernel/printk
7          4          1          7
[Tekkaman2440@SBC2440V4]#insmod hello.ko
```

```

Hello, Tekkaman Ninja !
[Tekkaman2440@SBC2440V4]#rmmod hello
Goodbye, Tekkaman Ninja !
Love Linux !Love ARM ! Love KeKe !

```

四个数字的含义：当前的loglevel、默认loglevel、最小允许的loglevel、引导时的默认loglevel。

为了方便的打开和关闭调试信息，《Linux设备驱动程序（第3版）》提供以下源码：

```

/* Macros to help debugging */
#undef PDEBUG /* undef it, just in case */
#ifdef SCULL_DEBUG
# ifdef __KERNEL__
    /* This one if debugging is on, and kernel space */
# define PDEBUG(fmt, args...) printk( KERN_DEBUG "scull: " fmt, ## args)
# else /* This one for user space */
# define PDEBUG(fmt, args...) fprintf(stderr, fmt, ## args)
# endif
#else
# define PDEBUG(fmt, args...) /* not debugging: nothing */
#endif

#undef PDEBUGG
#define PDEBUGG(fmt, args...) /* nothing: it's a placeholder */

```

Makefile中要添加的语句：

```

# Comment/uncomment the following line to disable/enable debugging
DEBUG = y

# Add your debugging flag (or not) to CFLAGS
ifeq ($(DEBUG),y)
    DEBFLAGS = -O -g -DSCULL_DEBUG # "-O" is needed to expand inlines
else
    DEBFLAGS = -O2
endif

CFLAGS += $(DEBFLAGS)

```

为了避免printk重复输出过快而阻塞系统，内核使用以下函数跳过部分输出：

```
int printk_ratelimit(void);
```

典型的应用如下：

```

if (printk_ratelimit( ))
    printk(KERN_NOTICE "The printer is still on fire\n");

```

可以通过修改`/proc/sys/kernel/printk_ratelimit`（重开信息前应等待的秒数）和`/proc/sys/kernel/printk_ratelimit_burst`（在速度限制前可接受的信息数）来定制`printk_ratelimit`的行为。

Linux还提供了打印设备编号的宏（在`<linux/kdev_t.h>`中定义）：

```

int print_dev_t(char *buffer, dev_t dev);
char *format_dev_t(char *buffer, dev_t dev);

```

两个函数的唯一区别是：`print_dev_t`返回打印字符数，`format_dev_t`返回缓冲区指针。注意缓冲区`char *buffer`的大小应至少有20B。

三、通过查询调试

多数情况中，获取相关信息的最好方法是在需要的时候才去查询系统信息，而不是持续不断地产生数据。

使用`/proc`文件系统

`/proc`文件系统是一种特殊的、由软件创建的文件系统，内核使用他向外界导出信息。`/proc`下面的每个文件都绑定于一个内核函

数，用户读取其中的文件时，该函数动态的生成文件的内容。如以前用过的：

```
[Tekkaman2440@SBC2440V4]#cat /proc/devices
```

Character devices:

```
1 mem
2 pty
3 tty
4 /dev/vc/0
4 tty
4 ttyS
5 /dev/tty
5 /dev/console
5 /dev/ptmx
7 vcs
10 misc
13 input
14 sound
81 video4linux
89 i2c
90 mtd
116 alsa
128 ptm
136 pts
180 usb
189 usb_device
204 s3c2410_serial
252 scull
253 usb_endpoint
254 rtc
```

Block devices:

```
1 ramdisk
256 rfd
7 loop
31 mtblock
93 nftl
96 inftl
179 mmc
```

使用/proc的模块必须包含<linux/proc_fs.h>，而使用seq_file接口要包含<linux/seq_file.h>。

具体的应用方法看源程序、做实验更有效果。

至于其他的调试方法，如gdb、LTT、SysRq等方法，在其他的书籍，如：《嵌入式Linux系统开发技术详解—基于ARM》、《构建嵌入式Linux系统》等，上讲解的更为详细，以后专门花时间研究。

四、源码实验

模块程序链接：模块程序

模块测试程序链接：模块测试程序

实验现象：

```
[Tekkaman2440@SBC2440V4]#cd /lib/modules/
```

```
[Tekkaman2440@SBC2440V4]#insmod scull_debug.ko scull_nr_devs=1 scull_quantum=6
scull_qset=2
```

```
[Tekkaman2440@SBC2440V4]#cd /tmp/
```

```
[Tekkaman2440@SBC2440V4]#. /scull_test
```

```
write code=6
```

```
write code=6
```

```
write code=6
```

```
write code=2
```

```

read code=6
read code=6
read code=6
read code=2
[0]=0 [1]=1 [2]=2 [3]=3 [4]=4
[5]=5 [6]=6 [7]=7 [8]=8 [9]=9
[10]=10 [11]=11 [12]=12 [13]=13 [14]=14
[15]=15 [16]=16 [17]=17 [18]=18 [19]=19

[Tekkaman2440@SBC2440V4]#cd /proc/
[Tekkaman2440@SBC2440V4]#ls
1          751      cmdline      kallsyms      stat
2          769      cpu          kmsg          swaps
3          77       cpuinfo      loadavg       sys
4          778      crypto       locks         sysrq-trigger
5          779      devices      meminfo       sysvipc
59         78       diskstats    misc          timer_list
6          781      driver       modules       tty
60         783      execdomains  mounts        uptime
63         785      filesystems  mtd           version
65         79       fs           net           vmstat
707        80       ide          partitions    yaffs
708        819      interrupts   scullmem      zoneinfo
709        asound    iomem        scullseq
710        buddyinfo  ioports      self
742        bus        irq          slabinfo

[Tekkaman2440@SBC2440V4]#cat scullmem

Device 0: qset 2, q 6, sz 20
  item at c071ebd4, qset at c071ef7c
  item at c071ef14, qset at c071eee0
    0: c071eeac
    1: c071ee78
[Tekkaman2440@SBC2440V4]#cat scullseq

Device 0: qset 2, q 6, sz 20
  item at c071ebd4, qset at c071ef7c
  item at c071ef14, qset at c071eee0
    0: c071eeac
    1: c071ee78
[Tekkaman2440@SBC2440V4]#rmmod scull_debug
[Tekkaman2440@SBC2440V4]#ls
1          742      buddyinfo    iomem         self
2          751      bus          ioports       slabinfo
3          769      cmdline      irq           stat
4          77       cpu          kallsyms      swaps
5          778      cpuinfo      kmsg          sys
59         779      crypto       loadavg       sysrq-trigger
6          78       devices      locks         sysvipc
60         781      diskstats    meminfo       timer_list
63         783      driver       misc          tty
65         785      execdomains  modules       uptime
707        79       filesystems  mounts        version
708        80       fs           mtd           vmstat
709        824      ide          net           yaffs
710        asound    interrupts   partitions    zoneinfo

```

阅读 (11787) | 评论 (2) | 转发 (41) |

上一篇: Linux设备驱动程序学习（0）-Hello, world模块
下一篇: Linux设备驱动程序学习（3）-并发和竞态

0

相关热门文章

欢迎hello_simple在ChinaUnix...	linux 常见服务端口	移植 ushare 到开发板
欢迎成之六哥哥在ChinaUnix博...	【ROOTFS搭建】busybox的httpd...	系统提供的库函数存在内存泄漏...
欢迎gzhf168在ChinaUnix博客安...	xmanager 2.0 for linux配置	linux虚拟机 求教
欢迎murak4mi在ChinaUnix博客...	什么是shell	初学UNIX环境高级编程的，关于...
欢迎creazydao在ChinaUnix博客...	linux socket的bug??	chinaunix博客什么时候可以设...

给主人留下些什么吧！~~



tekkamanninja 2012-03-28 08:53:00

jerry20000: syslog(8, NULL, level)//设置日志等级，记录方式，记录目录
setlevel.c这个程序在2.6.30内核编译出错了



.....
这个是4年前用2.6.22.2内核的时候写的，虽然内核的系统调用接口没有变，但是glibc的接口改了。我的博文更新过了，见上面的博文 😊

[回复](#) | [举报](#)



jerry20000 2012-03-27 22:46:01

syslog(8, NULL, level)//设置日志等级，记录方式，记录目录
setlevel.c这个程序在2.6.30内核编译出错了



[回复](#) | [举报](#)

评论热议

登录后评论。
[登录](#) [注册](#)

[关于我们](#) | [关于IT168](#) | [联系方式](#) | [广告合作](#) | [法律声明](#) | [免费注册](#)

Copyright 2001-2010 ChinaUnix.net All Rights Reserved 北京皓辰网域网络信息技术有限公司. 版权所有

感谢所有关心和支持过ChinaUnix的朋友们
京ICP证041476号 京ICP证060528号