

zhoukejun的专栏

≡ 目录视图

≡ 摘要视图

RSS 订阅

个人资料



zhoukejun

访问：73122次

积分：701

等级：BLOG > 3

排名：千里之外

原创：11篇

转载：0篇

译文：0篇

评论：14条

文章搜索

文章存档

2009年05月 (3)

2009年03月 (1)

2008年11月 (3)

2008年10月 (1)

2008年04月 (1)

展开

阅读排行

汉字一、二级字库的汉字 (23426)

中文拼音表，完全包括Gl (22804)

Linux内核proc文件系统 (10123)

make android 时的 libw (7008)

使用wildpackets omnipe (3525)

深圳远峰YFDVK-255-I开 (1063)

深圳远峰YFDVK-255-I开 (1060)

深圳远峰YFDVK-255-I开 (755)

u-boot-1.3.4-yf255 在 wv (709)

一起来学cdecl (492)

【公告】博客系统优化升级

【收藏】Scala 资源一应俱全

博乐招募开始啦

程序员七夕表白礼品指南

Linux内核proc文件系统的冰山一角 - - 源自对/proc/net/dev文件中各网卡参数的疑问

标签：linux内核 struct file linux action token

2009-05-06 00:3210123人阅读评论(2)收藏举报

版权声明：本文为博主原创文章，未经博主允许不得转载。

最近在调试网卡驱动时一直对/proc/net/dev 文件中各网卡参数是如何得来的有很大的疑问。
insmod 以太网卡驱动后，cat /proc/net/dev 可以看到多了eth0的信息。开始，一直以为是驱动的代码调用某个写/proc文件的函数，从而把eth0的信息写到/proc/net/dev文件里。可是反复看代码也没有发现相关的函数。今天突然想先拿/proc/interrupt 来分析一下，看了看linux-2.6.28的代码，就有了这篇《Linux内核proc文件系统的冰山一角——源自/proc/net/dev文件中各网卡参数的疑问》。

(一) 先分析/proc/interrupts

linux-2.6.28/arch/arm/kernel/irq.c --> show_interrupt()

在linux-2.6.28/fs/proc/interrupt.c
module_init(proc_interrupts_init);
|
-->proc_interrupts_init()
|
--> proc_create("interrupts", 0, NULL, &proc_interrupts_operations); /*参数NULL说明是在/proc 目录下创建名字为interrupts的文件。*/

一位伟大的程序员说过 -- "代码就是最好的文档". (黄金必杀句)
static const struct file_operations proc_interrupts_operations = {
.open = interrupts_open,
...
};

static int interrupts_open(struct inode *inode, struct file *filp)
{
return seq_open(filp, &int_seq_ops);
}

static const struct seq_operations int_seq_ops = {

评论排行

- 深圳远峰YFDVK-255-I开 (5)
- 中文拼音表，完全包括G (4)
- Linux内核proc文件系统 (2)
- 汉字一、二级字库的汉字 (2)
- 使用wildpackets omnipe (1)
- 深圳远峰YFDVK-255-I开 (0)
- 深圳远峰YFDVK-255-I开 (0)
- 深圳远峰YFDVK-255-I开 (0)
- 一起来学cdecl (0)
- u-boot-1.3.4-yf255 在 w (0)

推荐文章

- * 致JavaScript也将征服的物联网世界
- * 从苏宁电器到卡巴斯基：难忘的三年硕士时光
- * 作为一名基层管理者如何利用情商管理自己和团队（一）
- * Android CircleImageView圆形ImageView
- * 高质量代码的命名法则

最新评论

- 深圳远峰YFDVK-255-I开发板之b
riaa4: zhongsong86@163.com
- 深圳远峰YFDVK-255-I开发板之b
riaa4: 大神能不能传个YFSJF下
载软件给小弟
- 深圳远峰YFDVK-255-I开发板之b
riaa4: 大神能不能传个YFSJF下
载软件给小弟
- 深圳远峰YFDVK-255-I开发板之b
riaa4: 大神能不能传个YFSJF下
载软件给小弟
- 深圳远峰YFDVK-255-I开发板之b
riaa4: 大神能不能传个YFSJF下
载软件给小弟
- 使用wildpackets omnipeek、wi
zhanghailun413: fdgfdgfsd
- Linux内核proc文件系统的冰山一
xduanliang6: good
- 汉字一、二级字库的汉字与unicc
shl198838: 谢了，找了很长时间
了
- Linux内核proc文件系统的冰山一
匿名用户: 一个煞白，还充满魔力
的show
- 中文拼音表，完全包括GB2312等
匿名用户:

```
...
.show = show_interrupts
};

linux-2.6.28/arch/arm/kern
show_interrupts
{
...
for_each_present_cpu(cpu) {
sprintf(cpuname, "CPU%d", cpu);
seq_printf(p, " %10s", cpuname);
}
...
seq_printf(p, "%3d: ", i);
for_each_present_cpu(cpu)
seq_printf(p, "%10u ", kstat_cpu(cpu).irqs[i]);
seq_printf(p, " %10s", irq_desc[i].chip->name ? : "-");
seq_printf(p, " %s", action->name);
for (action = action->next; action; action = action->next)
seq_printf(p, ", %s", action->name);
...
}
```

在这里如果能够想到irq_desc[]这个全局的结构数组, 就都明白了。

下面是一个实际运行 cat /proc/interrupts 的结果

```
CPU0
8: 2 GPIO-I eth0
11: 0 SC pxa25x_udc
14: 404 SC AC97
22: 43 SC FFUART
25: 0 SC DMA
26: 19388 SC ost0
67: 0 GPIO UCB1400
Err: 0
```

（二）再来看看/proc/net/dev

直接看代码吧，linux-2.6.28/net/core/dev.c

```
static int __net_init dev_proc_net_init(struct net *net)
{
...
if (!proc_net_fops_create(net, "dev", S_IRUGO, &dev_seq_fops))
goto out;
...
}
```

```
static int dev_seq_open(struct inode *inode, struct file *file)
{
    return seq_open_net(inode, file, &dev_seq_ops,
        sizeof(struct seq_net_private));
}

static const struct file_operations dev_seq_fops = {
    ...
    .open = dev_seq_open,
    ...
};

static int dev_seq_open(struct inode *inode, struct file *file)
{
    return seq_open_net(inode, file, &dev_seq_ops,
        sizeof(struct seq_net_private));
}

static const struct seq_operations dev_seq_ops = {
    ...
    .next = dev_seq_next,
    ...
    .show = dev_seq_show,
};

/*
 * Called from the PROCfs module. This now uses the new arbitrary sized
 * /proc/net interface to create /proc/net/dev
 */
static int dev_seq_show(struct seq_file *seq, void *v)
{
    if (v == SEQ_START_TOKEN)
        seq_puts(seq, "Inter-| Receive "
            " | Transmit/n"
            " face |bytes packets errs drop fifo frame "
            "compressed multicast|bytes packets errs "
            "drop fifo colls carrier compressed/n");
    else
        dev_seq_printf_stats(seq, v);
    return 0;
}

static void dev_seq_printf_stats(struct seq_file *seq, struct net_device *dev)
{
    struct net_device_stats *stats = dev->get_stats(dev);
```

```

seq_printf(seq, "%6s:%8lu %7lu %4lu %4lu %4lu %5lu %10lu %9lu "
"%8lu %7lu %4lu %4lu %4lu %5lu %7lu %10lu/n",
dev->name, stats->rx_bytes, stats->rx_packets,
stats->rx_errors,
stats->rx_dropped + stats->rx_missed_errors,
stats->rx_fifo_errors,
stats->rx_length_errors + stats->rx_over_errors +
stats->rx_crc_errors + stats->rx_frame_errors,
stats->rx_compressed, stats->multicast,
stats->tx_bytes, stats->tx_packets,
stats->tx_errors, stats->tx_dropped,
stats->tx_fifo_errors, stats->collisions,
stats->tx_carrier_errors +
stats->tx_aborted_errors +
stats->tx_window_errors +
stats->tx_heartbeat_errors,
stats->tx_compressed);
}

```

```

void *dev_seq_next(struct seq_file *seq, void *v, loff_t *pos)
{
    struct net *net = seq_file_net(seq);
    ++*pos;
    return v == SEQ_START_TOKEN ?
first_net_device(net) : next_net_device((struct net_device *)v);
}

```

下面是一个实际运行 `cat /proc/net/dev` 的结果

Inter-|Receive Transmit

**face |bytes packets errs drop fifo frame compressed multicast|bytes packets errs drop fifo colls
carrier compressed**

lo:16200 185 0 0 0 0 0 0 16200 185 0 0 0 0 0

eth0:13817124 13810 0 0 0 0 0 0 3416037 13030 0 0 0 0 0

"%6s: %8lu %7lu %4lu %4lu %4lu %5lu %10lu %9lu " "%8lu %7lu %4lu %4lu %4lu %5lu %7lu %10lu/n",

17个参数完全匹配上了。这里需要指出的是Linux的网卡设备是由linux的链表管理的，`dev_seq_next`的作用就不言而喻了。

真是无语了，贴了一堆代码。不过最起码还有点代码，要不然岂不成了无code（哈哈，多么天真无邪的笑声）。

认真想一下，突然醒悟了，“驱动的代码调用某个写/proc文件的函数把eth0的信息写到/proc/net/dev文件”的想法是不正确的，起码没有深刻的理解Linux下文件的含义。Linux的vfs里各文件的读写等操作可以有不同的实现，而上述的几个/proc文件都没有write的实现，这就是从驱动的角度来跟踪内核代码没有收获的原因了，因为根本就没有去写/proc/net/dev这个文件，自然找不到向/proc/net/dev添加eth0内容的函数了。

但是为什么cat /proc/net/dev的内容出现了变化呢？先说个看似不相关的话，魔术师大卫·科波菲尔的“消失自由女神像”是真的把自由女神像移走了吗？当然没有，但是他给观众show出的景象里自由女神像是不存在的，观众就认为

自由女神像消失了。Linux的vfs就魔术师一样，/proc的那些文件的操作函数很特殊，不需要直接对/proc文件的内容进行写操作来添加内容，就能用充满魔力的show函数来给用户提供的信息。

顶

0

踩

0

上一篇

u-boot-1.3.4-yf255 在 [www.github.com](#) 建立 repo

下一篇

make android 时的 libwebcore.so error

猜你在找

- CentOS7 Linux系统管理实战视频课程
- 《Linux内核编程》第四章proc文件系统
- 嵌入式Linux项目实战：三个大项目(数码相框、摄像头)
- Linux内核设计基础二之Proc文件系统
- linux嵌入式开发+驱动开发
- 通过proc文件系统输出必要的Linux内核信息中
- 从零写Bootloader及移植uboot、linux内核、文件系统
- Linux内核模块编程—proc文件系统
- 话说linux内核-uboot和系统移植第14部分
- Linux内核通信之一—proc文件系统详解

ubuntu手机

无线上网卡

wifi无线网卡

上海周浦房价

内置无线网卡

网卡参数

冰山一


查看评论

2楼 [xduanliang6](#) 2012-04-07 10:53发表



good

1楼 [匿名用户](#) 2010-04-24 17:07发表



一个煞白，还充满魔力的show

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服

杂志客服

微博客服

webmaster@csdn.net

400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

