

Chinaunix首页 | 论坛 | 问答 | 博客

登录 | 注册

博文 ▼

# Tekkaman Ninja

tekkamanninja. blog. chinaunix. ne

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: 75927 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(3-补)-Linux中的循环缓冲区

2007-12-10 11:25:19

分类: LINUX

# Linux设备驱动程序学习(3-补)

有奖征集: 文集--博客系列博文管理

-Linux中的循环缓冲区

参考资料:《Linux内核中的循环缓冲区》作者:西邮 王聪 严重感谢文章作者! 但是(可能是源码版本问题)有些结论并不正确: "而kfifo\_init只会接受一个已分配好空间的fifo->buffer,不能和kfifo\_free搭配,用kfifo\_init分配的kfifo只能用kfree释放。" 阅读源码可以得出这样的结论:kfifo\_init和kfifo\_alloc分配的kfifo都能用kfree释放。已经用实验证实。

原文链接地址: http://www.kerneltravel.net/jiaoliu/kern-kfifo.html

在学习到第十章 中断处理 时,其中的中断驱动的I/0需要使用缓冲区,我觉得与其自己实现一个缓冲区,不如利用内核已经写好的fifo。内核里有一个通用的循环缓冲区的实现在〈linux/kfifo.h〉。 使用的数据结构如下:

struct kfifo {

unsigned char \*buffer; /\* 使用的缓冲区头指针 \*/

unsigned int size; /\* 缓冲区总大小 \*/

unsigned int in; /\* 己写入缓冲区的数据总量, 当前缓冲区写指针的偏移量: (in %

size) \*/

unsigned int out; /\* 已读出缓冲区的数据总量,当前缓冲区读指针的偏移量: (out %

size) \*/

spinlock\_t \*lock; /\* 为避免竞态的自旋锁 \*/

};/\*当in==out时,缓冲区为空;当(in-out)==size时,缓冲区已满\*/

kfifo提供的循环缓冲的部分函数分为2类:

- (1) 以双下划线开头,没有使用自旋锁函数;
- (2) 没有双下划线开头,需要额外加锁的情况下使用的函数。

其实第二类只是在第一类的基础上进行加锁后,实际的代码如下:

```
unsigned long flags;
```

spin lock irqsave(fifo->lock, flags);

/\*第一类函数\*/

spin\_unlock\_irqrestore(fifo->lock, flags);

以下我按使用的顺序介绍每个函数的使用,部分函数源码在kernel/kfifo.c中定义,这些接口是经过精心构造的,可以小心地避免一些边界情况,原理其实很简单,建议去看源码弄清楚实现的原理,可以学到一些编程技巧。

(0) 声明循环缓冲数据结构指针

struct kfifo \*tekkamanfifo;

(1) 初始化循环缓冲结构体

```
struct kfifo *kfifo_init(unsigned char *buffer, unsigned int size,
```

gfp\_t gfp\_mask, spinlock\_t \*lock);

/\*调用kfifo\_init必须保证size是2的整数次幂,而且buffer只接受一个已分配好空间的指针。也就是说之前要使用kmalloc分配好空间,将返回的指针传递到buffer\*/

struct kfifo \*kfifo\_alloc(unsigned int size, gfp\_t gfp\_mask,

spinlock\_t \*lock);

### 2014年6月17日

- 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

















wkm81018



















hushup

### 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...

- ·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

# Linux设备驱动程序学习(3-补)-Linux中的循环缓冲区-tekkamanninja-ChinaUnix博客

/\*调用kfifo\_alloc不必保证size是2的幂,它内部会把size向上调整到2的整数次幂。空间分配的 内部实现使用kmalloc。函数内部调用kfifo\_init/

buffer: 之前要使用kmalloc分配好的空间指针;

size: 循环缓冲空间大小;

gfp mask: 和kmalloc使用的分配标志(flags)一样。(参阅Linux设备驱动程序学习(8)-分配内存)

lock: 是事先声明并初始化好的自旋锁结构体指针;

返回值 为初始化好的循环缓冲数据结构指针 。

(2) 向缓冲区里写入数据

```
unsigned int kfifo put(struct kfifo *fifo, unsigned char *buffer, unsigned int len);
unsigned int _kfifo_put(struct kfifo *fifo, unsigned char *buffer, unsigned int len);
```

fifo: 要写入数据的缓冲区结构体指针;

buffer: 要写入的数据指针,指向内核空间。如需要用户空间数据,之前要用copy\_from\_user复制数据到 内核空间;

len: 要写入的数据大小:

返回值 为写入缓冲区的数据字节数。

(3) 从缓冲区里读出数据

```
unsigned int kfifo get(struct kfifo *fifo, unsigned char *buffer, unsigned int len);
unsigned int _kfifo_get(struct kfifo *fifo, unsigned char *buffer, unsigned int len);
```

参数定义和kfifo\_put类似。

返回值 为从缓冲区读出的数据字节数。

(4) 得到缓冲区已有的数据字节数

```
unsigned int kfifo len(struct kfifo *fifo);
unsigned int    kfifo len(struct kfifo *fifo);
```

fifo: 要操作的缓冲区结构体指针;

函数返回缓冲区实际已有的数据字节数,内部实现十分简单,就是in - out;

返回值 为缓冲区已有的数据字节数。

(5) 清空缓冲区

```
void kfifo reset(struct kfifo *fifo);
void kfifo reset(struct kfifo *fifo);
```

内部实现十分简单,就是in = out = 0。

(6) 使用结束,释放缓冲区。

```
void kfifo_free(struct kfifo *fifo);
```

所有的kfifo提供的循环缓冲的函数就是这些。在理解内部实现原理的基础上才能更好的使用它,所以再 次建议阅读源码, 因为源码很简单, 但是很精巧。

# ARM9开发板实验

实验模块源码: scull-kfifo

测试程序源码: scull-kfifo-test

实验现象:

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

[Tekkaman2440@SBC2440V4]#insmod scull\_kfifo.ko

[Tekkaman2440@SBC2440V4]#cat /proc/devices

Character devices:

- 1 mem
- 2 pty
- 3 ttyp
- 4 / dev/vc/0
- 4 tty
- 4 t.t.vS

```
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
153 spi
180 usb
189 usb_device
204 s3c2410 serial
252 scull_kfifo
253 usb_endpoint
254 rtc
Block devices:
 1 ramdisk
256 rfd
 7 loop
31 mtdblock
93 nftl
96 inftl
179 mmc
[Tekkaman2440@SBC2440V4]#mknod -m 666 /dev/scull_kfifo c 252 0
[Tekkaman2440@SBC2440V4]#echo 1234567890 > /dev/scull_kfifo
"sh" did write 11 bytes
[Tekkaman2440@SBC2440V4]#/tmp/scull_kfifo_test
scull kfifo: the module can not lseek!
please input the command:1
scull_kfifo: ioctl SCULL_KFIFO_SIZE len=11
please input the command :2
scull_kfifo: SCULL_KFIFO_RESET code=0
please input the command:1
scull_kfifo: ioctl SCULL_KFIFO_SIZE len=0
please input the command :q
"sh" did write 31 bytes
[Tekkaman2440@SBC2440V4]#echo 123456789012345678901234567890 > /dev/scull_kfifo
"sh" did write 31 bytes
[Tekkaman2440@SBC2440V4]#echo 1234567890 > /dev/scull kfifo
"sh" did write 2 bytes
"sh" did write 0 bytes
printk: 204310 messages suppressed.
"sh" did write 0 bytes
1234567890
[Tekkaman2440@SBC2440V4]#/tmp/scull_kfifo_test
```

```
scull_kfifo: the module can not lseek!
please input the command:1
scull kfifo: ioctl SCULL KFIFO SIZE len=64
please input the command :q
[Tekkaman2440@SBC2440V4]#cat /dev/scull kfifo
printk: 1493677 messages suppressed.
"cat" did read 64 bytes
1234"cat" reading: going to sleep
56789012345678901234567890
123456789012345678901234567890\\
12
[Tekkaman2440@SBC2440V4]#/tmp/scull_kfifo_test
scull_kfifo: the module can not lseek!
please input the command:2
scull kfifo: SCULL KFIFO RESET code=0
please input the command :q
[Tekkaman2440@SBC2440V4]#rmmod scull kfifo
[Tekkaman2440@SBC2440V4]#1smod
Module Size Used by Not tainted
[Tekkaman2440@SBC2440V4]#
```

```
阅读(5545) | 评论(0) | 转发(31) |

上一篇: Linux设备驱动程序学习(11) -中断处理

下一篇: Linux设备驱动程序学习(12) -Linux设备模型(底层原理简介)
```

# 相关热门文章

循环队列的实现 linux 常见服务端口 移植 ushare 到开发板
Linux设备驱动程序学习... 【ROOTFS搭建】busybox的httpd... 系统提供的库函数存在内存泄漏...
Linux设备驱动程序学习(1)-... xmanager 2.0 for linux配置 linux虚拟机 求教
Linux设备驱动程序学习(2)-... 什么是shell 初学UNIX环境高级编程的,关于...
Linux设备驱动程序学习(3)-... linux socket的bug?? chinaunix博客什么时候可以设...

## 给主人留下些什么吧! ^^

# 评论热议

请登录后评论。 <u>登录</u> 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

京ICP证041476号 京ICP证060528号