登录 | 注册

gdt-A20的专栏

博客文章，如未有特殊说明，均为原创(会有不定期修正)，如需转载请注明出处，谢谢合作！Love kernel！

目录视图 摘要视图 RSS 订阅

个人资料

mostmark

访问： 69106次

博客专家福利 C币兑换礼品剧透 10月推荐文章汇总 加入“技术热心人”，赢丰厚奖品

[置顶] 利用qemu进行内核源码级调试

分类： kernel debug 2012-02-03 15:45 4000人阅读 评论(2) 收藏 举报

debugging makefile ubuntu 虚拟机 x86 android

==========

本文系本站原创,欢迎转载!

转载请注明出处:http://blog.csdn.net/gdt_a20

==========

概要：看kernel代码的时候，变量多了，就记不清楚了，尤其是读内存部分代码的时候，传来传去，常常不知道改成什么样子了，内存布局到底变成什么了，也许是我记忆力太不行了，利用qemu可以在线调试内核，

如同你用gdb调试app一样，断点，打印变量值等等，qemu算是个模拟器吧，说虚拟机也可以，虽然bug很多，

积分： 1137
等级： BLOG 4
排名： 第15599名

原创： 33篇 转载： 10篇
译文： 0篇 评论： 49条

文章搜索

博客专栏

linux kernel 札记
文章：27篇
阅读：40250

文章分类

Bootloader (1)
Kernel (27)
各种线 (1)
转载 (8)
电子电路基础 (5)
硬件设计 (0)
kernel debug (1)
android (2)

文章存档

但是还不失为一个好工具，如果对qemu陌生，那android的模拟器相信大家都很熟悉了吧，那个也是基于开源的qemu修改的,好了让我们赶快开始调试的旅程吧.

一、环境

```
#$cat /etc/issue
  Ubuntu 11.10 \n \l
```

二、gdb准备

```
#$$ gdb --version
GNU gdb (Ubuntu/Linaro 7.3-0ubuntu2) 7.3-2011.08
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
For bug reporting instructions, please see:
<http://bugs.launchpad.net/gdb-linaro/>.
```

我的gdb默认会有问题，当调试开启后gdb vmlinux会出现:

```
Remote 'g' packet reply is too long: 00000000000000000020e30100000000000000000000000060eecf81ffffffff0000000100000000048eecf81fffff
...
```

为了解决这个问题，我们需要重新编译一个gdb，

2012年08月 (1)
2012年03月 (8)
2012年02月 (13)
2012年01月 (12)
2011年10月 (1)
展开

阅读排行
二极管与门电路原理 (6852)
利用qemu进行内核源码级 (3998)
三极管非门电路原理（反 (3690)
关于linux设备模型kobjec (2758)
linux设备模型之mmc，s (2481)
linux设备模型之mmc，s (2202)
linux设备模型之bus，de (2014)
linux设备模型之bus，de (1836)
linux启动流程导读(arm为 (1806)
linux设备模型之spi子系统 (1793)

评论排行
linux设备模型之platform (7)
linux设备模型之bus，de (6)
u-boot 2010.09 支持min (5)
二极管与门电路原理 (5)
关于linux设备模型kobjec (4)
linux设备模型之mmc，s (3)

#$wget ftp://sourceware.org/pub/gdb/releases/gdb-7.4.tar.gz

#$tar xzvf gdb-7.4.tar.gz

#$cd gdb-7.4

我们需要针对上面的问题做如下修改：

在gdb/remote.c中

注释掉：

```cpp
if (buf_len > 2 * rsa->sizeof_g_packet)
           error (_("Remote 'g' packet reply is too long: %s"), rs->buf);
```

并在后面添加：

```cpp
if (buf_len > 2 * rsa->sizeof_g_packet) {
      rsa->sizeof_g_packet = buf_len ;
      for (i = 0; i < gdbarch_num_regs (gdbarch); i++)
      {
         if (rsa->regs[i].pnum == -1)
         continue;

         if (rsa->regs[i].offset >= rsa->sizeof_g_packet)
         rsa->regs[i].in_g_packet = 0;
         else
         rsa->regs[i].in_g_packet = 1;
      }
   }
```

推荐文章

* Qt Quick实现的涂鸦程序
* 优秀的编程风格（Java篇）——高薪必看
* HTML5游戏实战(1)：50行代码实现正面跑酷游戏
* 原来Github上的README.md文件这么有意思——Markdown语言详解
* Android Material Design之Toolbar与Palette实践

最新评论

关于linux设备模型kobject，kset
yy782101688: 谢谢！

linux设备模型之platform
Shimejing: 学习了 楼主

linux设备模型之platform
citycity222: 好文啊，必须支持下，学习咯~

二极管与门电路原理
该昵称已被占用啊: @mr_raptor: 唐老师好巧！！

linux中断导读之--初始化<1>
顾小林: 看中断我表示蛋很疼。

linux设备模型之bus，device，d
jvq: 好文章

configure前我们还要一个库，texinfo，否则gdb编译过程中会出现

makeinfo错误：missing makeinfo --split-size=5000000 ...

#$sudo apt-get install texinfo

好了现在可以configure了,

#$configure --prefix=/opt/gdb_7_4 //临时装在这里吧

#$mkdir /opt/gdb_7_4

#$make && make install

三、kernel准备

我们还需要一个bzImage，也就是个胖内核，qemu只钟爱肥的，由于我们目的只在调试内核，所以别的可以不准备，有兴趣的可以装个整套的，做个虚拟机看看，

#$wget ftp://ftp.kernel.org/pub/linux/kernel/v3.x/linux-3.2.tar.gz

#$tar xzvf linux-3.2.tar.gz

#$cd linux-3.2/

#$make defconfig

暂且用个x86做试验，我们要用gdb连接，需要把调试信息加入内核，

否则gdb时候会出现：no debugging symbols found

这里就别去修改Makefile添加-g了，直接在内核配置选项里打开就可以.

#$make menuconfig

在kernel hacking中选中compile the kernel with debug info,

这样编译出来的kernel才是真正的肥

#$make bzImage

四、准备qemu.

三极管工作原理(表现形式)
conceptcon: devm_request_and_ioremap

二极管与门电路原理
秘制菜鸟: 写的很好，转载了，谢谢卤煮！

关于linux设备模型kobject，kset
genovia: 这个图是用什么软件画的啊？？

linux设备模型之led子系统
kangear: 好文章，不过要先看了bus设备模型再来看这个比较好理解一点。

好了万事具备可以开始qemu了，

我们首先添加一个库，不然qemu运行会卡在:

VNC server running on `127.0.0.1:5900'

#$sudo apt-get install libsdl1.2-dev

#$wget http://wiki.qemu.org/download/qemu-1.0.tar.gz

#$tar xzvf qemu-1.0.tar.gz

#$sudo mkdir /opt/qemu_bin

#$cd qemu-1.0

#$./configure --prefix=/opt/qemu_bin --target-list=x86_64-softmmu

我的系统是64的，这里有i386-softmmu，arm平台等等，可以有选择

#$make && make install

这样就可以了，1.0的版本我这样安装使用后没有遇到单步无法运行的问题，

但这个问题在以前版本确实存在，当初单步会直接跳到中断里面，

如果有遇到这个问题的版本可以参考下面的修改(http://blog.chinaunix.net/space.php?uid=10285909&do=blog&id=2964933):

```
[cpp]

以qemu-0.9.0为例:
  1 target-i386\cpu.h CPUX86State 结构加入:
       int intr_cnt;  //当前的中断深度
       int intr_scnt; //保存的中断深度
  2 vl.c main函数中加入初始化:
       first_cpu->intr_cnt = 0;
  3 target-i386\Helper.c 加入中断深度的计算
       do_interrupt_protected 函数中加入
       env->intr_cnt++;
       helper_iret_protected 函数中加入
       env->intr_cnt--;
```

```
4 Exec.c cpu_single_step执行单步时保存当前的中断深度
     env->intr_scnt = env->intr_cnt;
5 target-i386\Translate.c DisasContext结构加入:
     CPUState *env;
6 target-i386\Translate.c gen_eob函数加入插入debug时的判断
     if (s->singlestep
     if (s->env->intr_
          gen_op_debug
          s->is_jmp =
          return;
     }
   }
   在函数gen_intermedia
   dc->env = env;
```

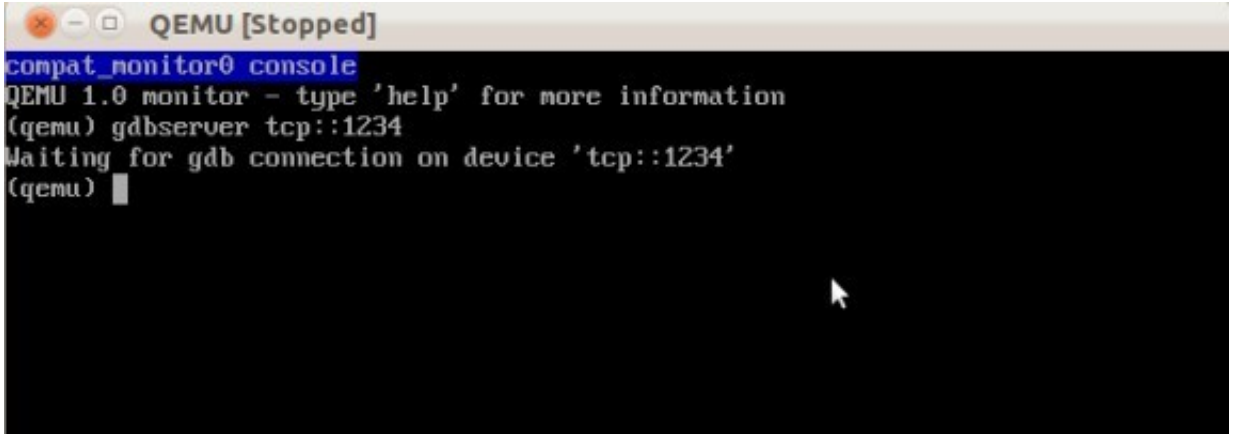
五、运行调试

#$/opt/qemu_bin/bin/qemu-system-i386 -S -kernel arch/x86/boot/bzImage -m 1024

ctrl+alt+1 与 ctrl+alt+2可以切换，前者是屏幕输出，后者是qemu控制台

运行起来后是黑屏，我们要切换到控制台，用鼠标点击窗口，然后ctrl+alt+2，

如下图所示输入，然后回车，ctrl+alt+1切换回来，ctrl+alt切出鼠标.



在另一个终端中，

#$cd linux-3.2/

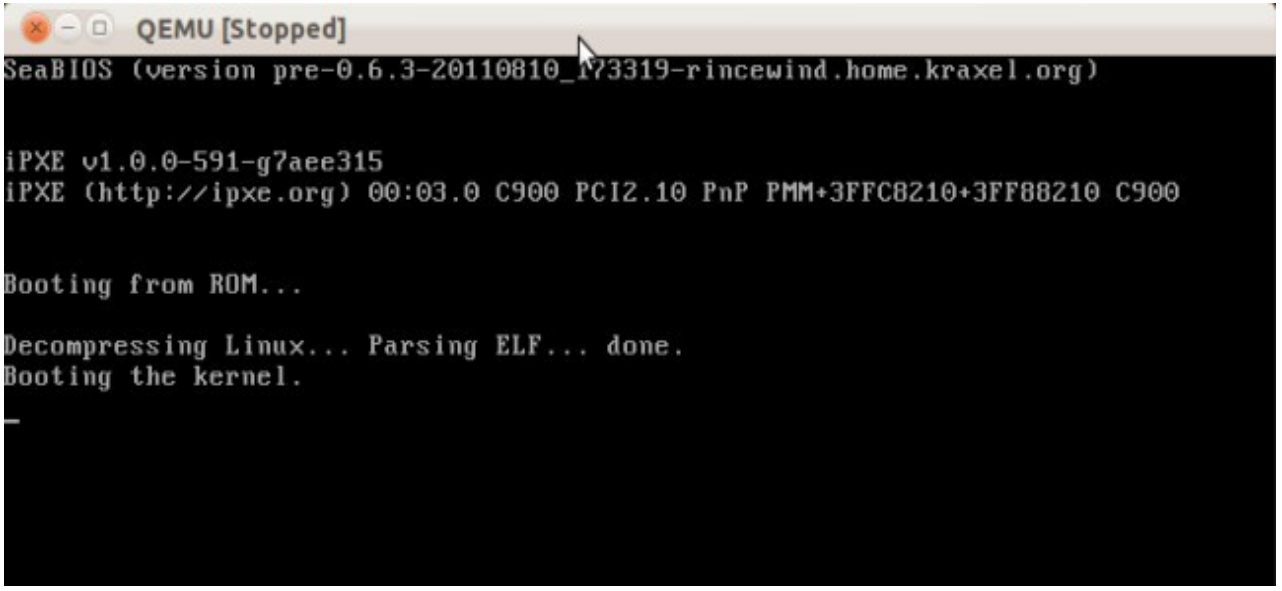
#$/opt/gdb_7.4/bin/gdb vmlinux

(gdb) target remote localhost:1234

好了可以正式开始你的调试之旅了，可以先

#b start_kernel

#continue



#list

#n

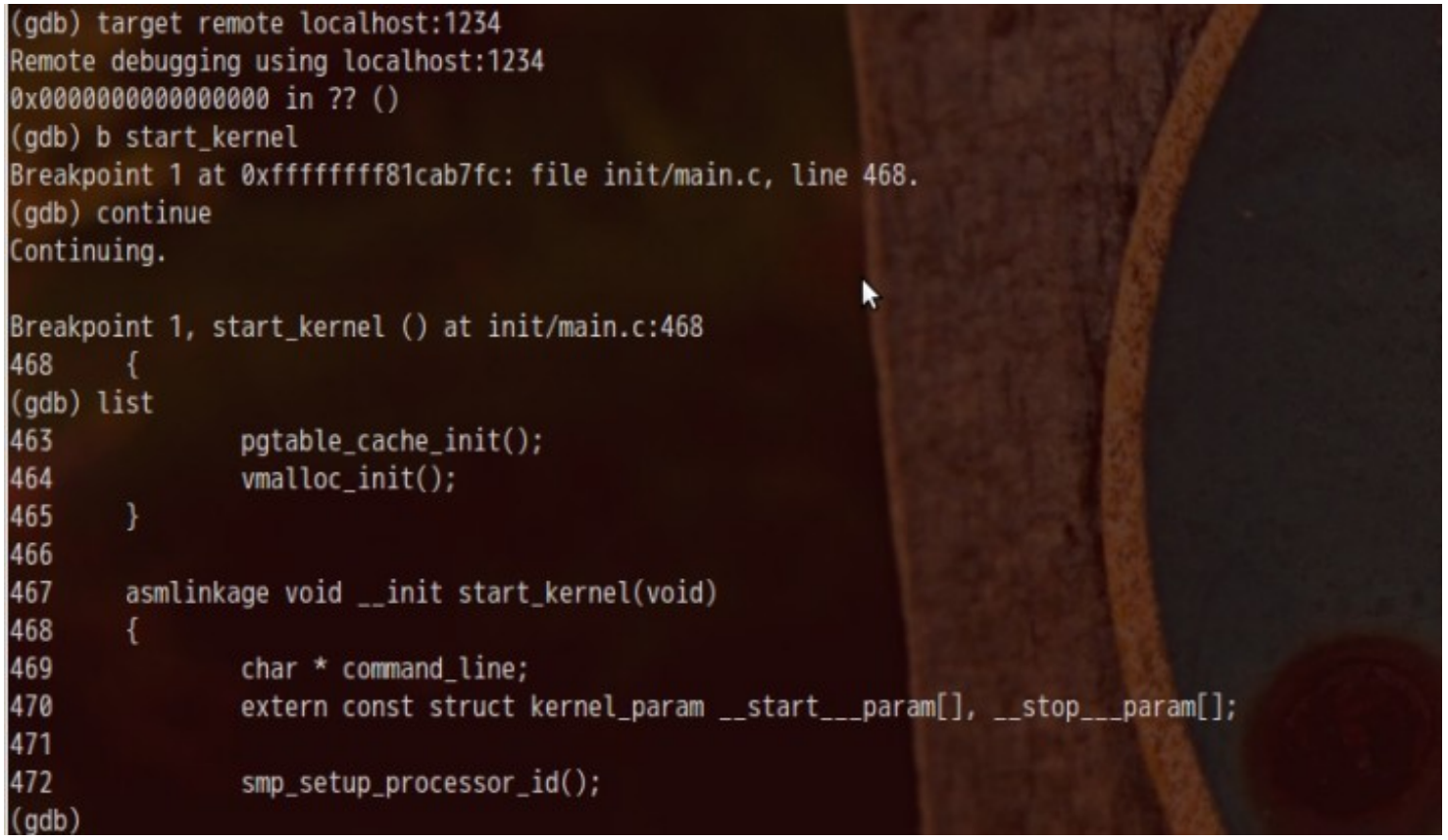
等等.

注意:不要把vmlinux从源码目录考出来运行，因为gdb list是需要当前上下文列出代码执行到哪里的.

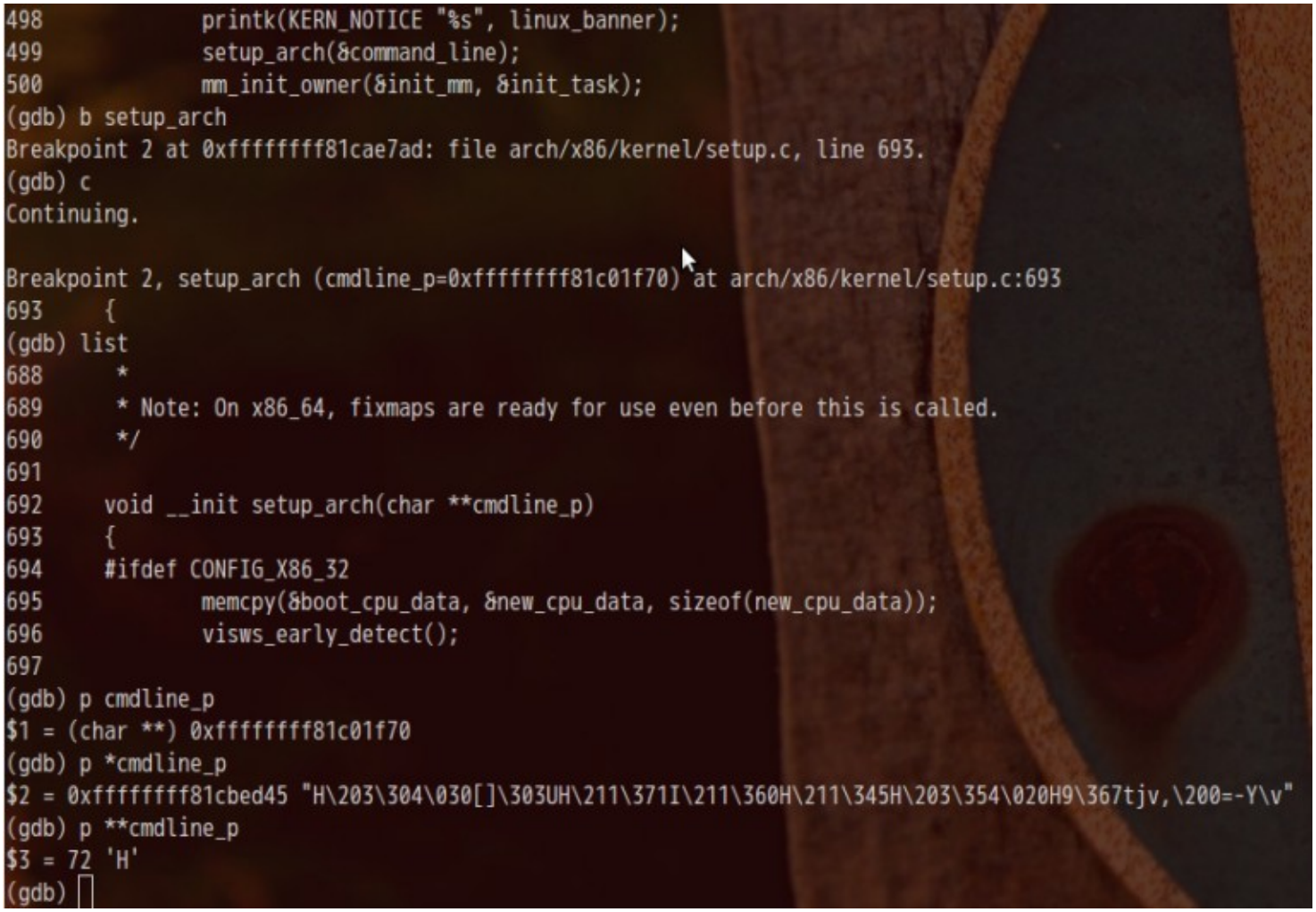
另外-m 参数用于设置内存大小，这个在调试内存时候尤其有用，你可以清晰的打印除normal内存高端内存布局等等^.^!

好了差不多就这么多吧，最后附上几幅截图：

==========

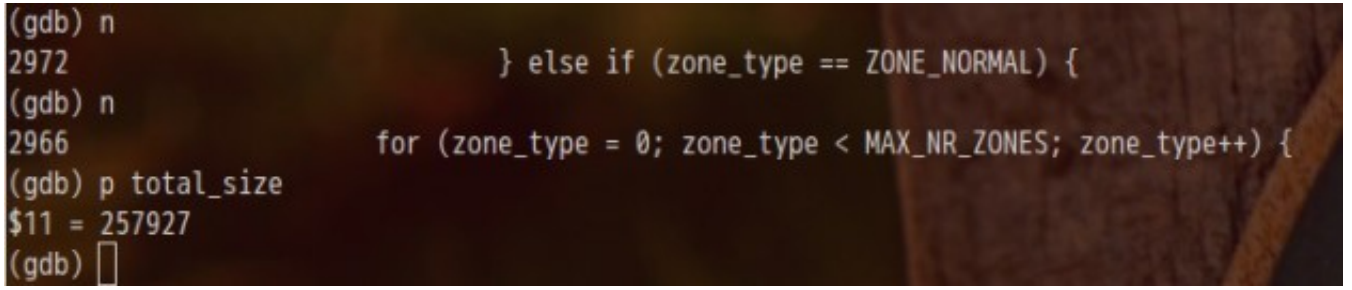


==========


498 printk(KERN_NOTICE "%s", linux_banner);
499 setup_arch(&command_line);
500 mm_init_owner(&init_mm, &init_task);
(gdb) b setup_arch
Breakpoint 2 at 0xffffffff81cae7ad: file arch/x86/kernel/setup.c, line 693.
(gdb) c
Continuing.
Breakpoint 2, setup_arch (cmdline_p=0xffffffff81c01f70) at arch/x86/kernel/setup.c:693
693 {
(gdb) list
688 *
689 * Note: On x86_64, fixmaps are ready for use even before this is called.
690 */
691
692 void __init setup_arch(char **cmdline_p)
693 {
694 #ifdef CONFIG_X86_32
695 memcpy(&boot_cpu_data, &new_cpu_data, sizeof(new_cpu_data));
696 visws_early_detect();
697
(gdb) p cmdline_p
$1 = (char **) 0xffffffff81c01f70
(gdb) p *cmdline_p
$2 = 0xffffffff81cbed45 "H\203\304\030[]\303UH\211\371I\211\360H\211\345H\203\354\020H9\367tjv,\200=-Y\v"
(gdb) p **cmdline_p
$3 = 72 'H'
(gdb)

==========

==========



==========

Thanks

上一篇 二极管与门电路原理

下一篇 二极管或门电路原理

顶 6 踩 2

主题推荐 内核 源码 qemu 控制台 虚拟机

猜你在找

ubuntu server 12.04上的skyeye1.3.5安装和使用(打造最精
如何用gdb进行汇编级的调试
跨平台开源BMP图像格式操作库——“伪”libbmp
[14.05.12]今后讨论班的走向
Linux-USB子系统
Soho Schubert: 调试工具Insight (GDB GUI)
understanding linux usb ehci device driver(1)
Linux内核通知链——notifier_call_chain
Linux查看文件句柄占用多的进程
从入门到深入地学习ＩＢＭ的小型机AS400

查看评论

2楼 slvm_lj 2012-12-05 17:20发表

Nice. 之前看到调试内核是通过kgdb或者kdb,要么需要虚拟机要么需要给内核打补丁,都失败了. 没想到可以用qemu+gdb这么简单就行了.试过确实是可行的. 感谢,顶起.

1楼 bugknightyyp 2012-12-05 16:03发表

顶起

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 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved