

lab3 实验说明

写一个简单的内核模块，首先定义一个定时器来模拟中断，再新建一个内核线程。当定时器到来时，唤醒内核线程，然后在内核线程的主程序中输出该内核线程的相关信息，如 PID、当前 jiffies 等信息。

基本实验步骤

1. 进入 `rlk_lab/rlk_basic/chapter_10/lab1` 目录。

```
# export ARCH=arm
# export CROSS_COMPILE=arm-linux-gnueabi-
# make BASEINCLUDE=/home/figo/work/runninglinuxkernel/runninglinuxkernel_4.0
```

这里 `BASEINCLUDE` 指定你当前 `runninglinuxkernel_4.0` 的目录路径。
然后把 `*.ko` 拷贝到 `runninglinuxkernel_4.0/kmodules` 目录下面。

运行如下脚本启动 Qemu。

```
#cd runninglinuxkernel_4.0
# sh run.sh arm32  #启动虚拟机
```

在 Qemu 虚拟机:

```
#cd /mnt
# insmod lab3-test.ko
```

```
/mnt # insmod lab3-test.ko
ben: my lockdep module init
/mnt # show_reg: ktest, pid:706
cpsr:0x20000013, sp:0xee12bf10
show_reg: ktest, pid:706
cpsr:0x20000013, sp:0xee12bf10
show_reg: ktest, pid:706
cpsr:0x20000013, sp:0xee12bf10
show_reg: ktest, pid:706
cpsr:0x20000013, sp:0xee12bf10
show_reg: ktest, pid:706
cpsr:0x20000013, sp:0xee12bf10
show_reg: ktest, pid:706
cpsr:0x20000013, sp:0xee12bf10
show_reg: ktest, pid:706
cpsr:0x20000013, sp:0xee12bf10
show_reg: ktest, pid:706
cpsr:0x20000013, sp:0xee12bf10
show_reg: ktest, pid:706
cpsr:0x20000013, sp:0xee12bf10
```

进阶思考

内核里提供了很多的延迟的方法，比如中断的下半部，软中断，`tasklet`，`workqueue` 等，另外还有 `timer`，内核线程等，那究竟在实际的驱动代码编写过程中，应该有哪些机制来实现呢？大家可以在内核代码中，选择一个你熟悉或者感兴趣的代码进行学习，看看内核里现成的驱动代码是如何实现的？

如果大家对这些问题感兴趣，可以关注笨叔的旗舰篇视频，笨叔会在视频中和大家详细解答。

shop115683645.taobao.com

配套视频 **旗舰篇**

第**1**季
内存管理

旗舰篇一次订阅，持续更新

规划中

| | |
|-----|----------------------|
| 第二季 | 进程管理和调度 / 中断 / 锁（已出） |
| 第三季 | 虚拟化 |
| 第四季 | Linux 内核和应用开发调试必杀技 |
| 第五季 | 红帽系列 |

微信公众号：奔跑吧 linux 社区

shop115683645.taobao.com

Linux 视频课程

1. > 一键订阅，持续更新
2. > 最有深度和广度的 Linux 视频
3. > 手把手解读 Linux 内核代码
4. > 紧跟 Linux 开源社区技术热点
5. > 笨叔叔的 VIP 私密群答疑
6. > 图书 + 视频，全新学习模式

微店：



扫码识别

淘宝店: <https://shop115683645.taobao.com/>

微信公众号:

