# 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。

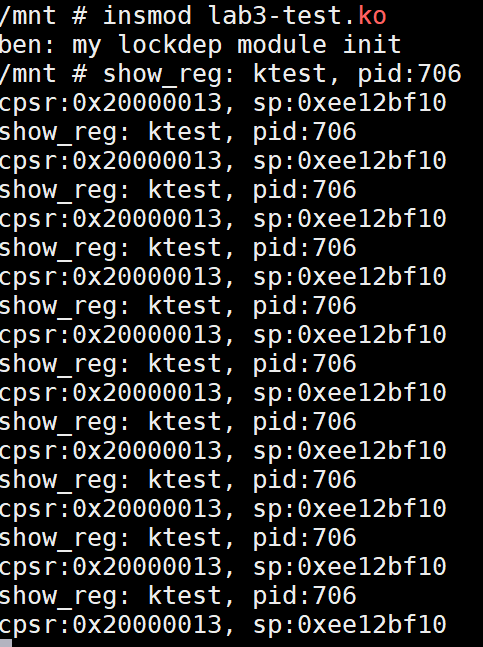
然后把\*.ko拷贝到runninglinuxkernel\_4.0/kmodules目录下面。

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 |



## 进阶思考

内核里有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？

请关注奔跑吧Linux社区微信公众号，里面有更多更有料更好玩的Linux干货：



奔跑吧Linux社区的旗舰篇视频节目，帮您打通Linux和职场任督二脉！





笨叔最新力作，全程约5小时高清，140多页ppt，8大实验，基于x86\_64的Centos 7.6和arm64，提供全套实验素材和环境。全面介绍kdump+crash在死机黑屏方面的实战应用，全部案例源自线上云服务器和嵌入式产品开发实际案例！

不能为客户和老板解决死机黑屏问题的都不是合格的linux运维或者研发攻城狮！





订阅视频请到淘宝店：<https://shop115683645.taobao.com>

