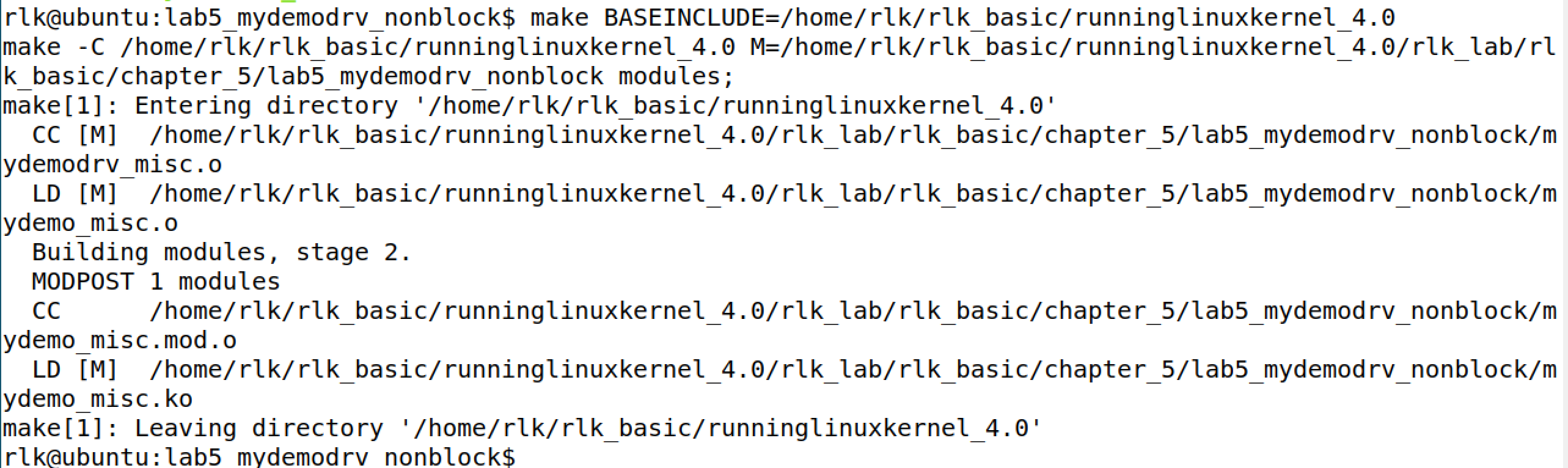
# Lab6\_把虚拟设备驱动改成阻塞模式

## 基本实验步骤

1. 进入对应的lab目录。

|  |
| --- |
| # export ARCH=arm  # export CROSS\_COMPILE=arm-linux-gnueabi-  # make BASEINCLUDE= 你的runninglinuxkernel\_4.0目录的绝对路径 |

这里BASEINCLUDE指定你当前runninglinuxkernel\_4.0的目录路径，每个人机器上路径可能不一样，请确保路径正确。



编译test测试程序。

|  |
| --- |
| # arm-linux-gnueabi-gcc test.c -o test --static |

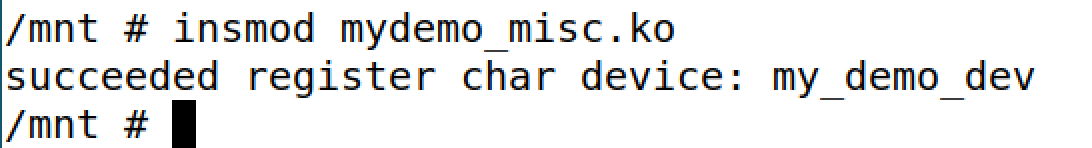
然后把ko内核模块和test程序拷贝到runninglinuxkernel\_4.0/kmodules目录下面。

运行如下脚本启动Qemu。

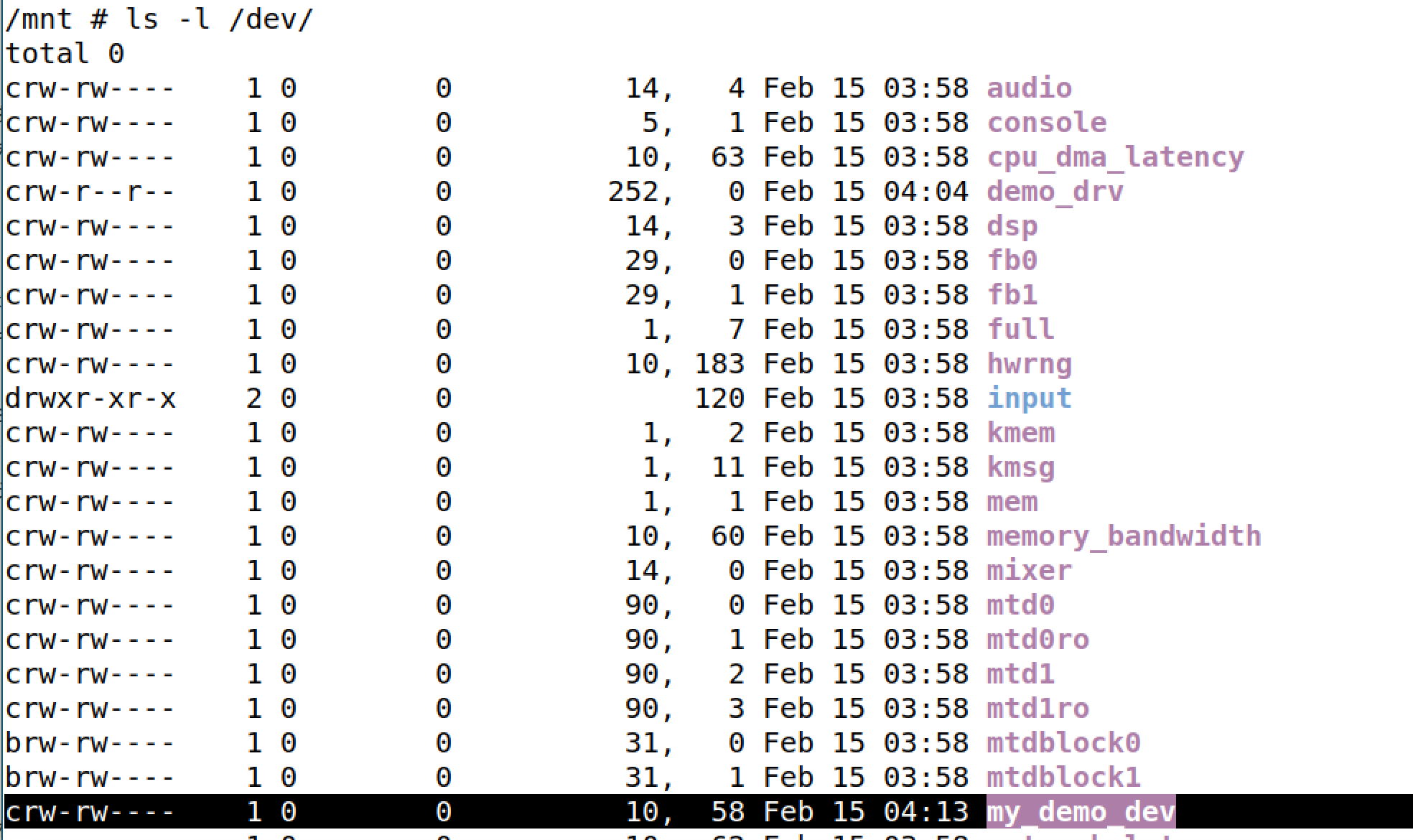
|  |
| --- |
| #cd runninglinuxkernel\_4.0  # sh run.sh arm32 #启动虚拟机，启动虚拟机之前，请先编译runninglinuxkernel\_4.0整个内核 |

在Qemu虚拟机:

|  |
| --- |
| #cd /mnt  # insmod xxxx.ko //加载刚才编译的内核模块 |

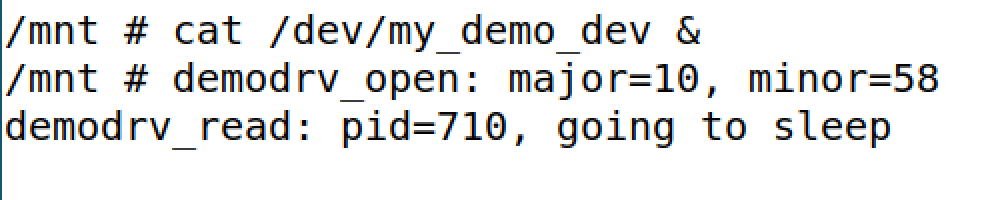


然后查看dev目录情况。

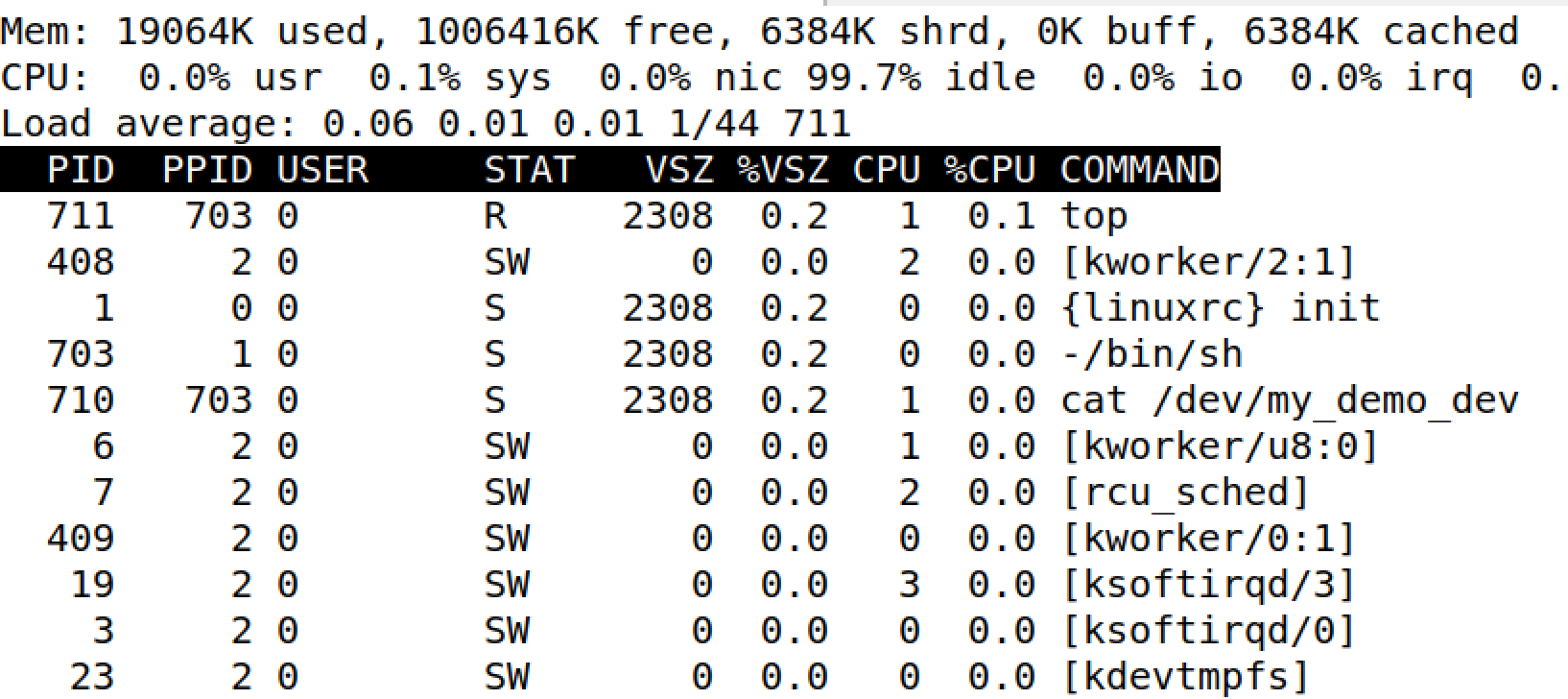


可以看到my\_demo\_dev设备自动创建了，主设备号是10，次设备号是58.

我们先用cat命令来打开读操作。

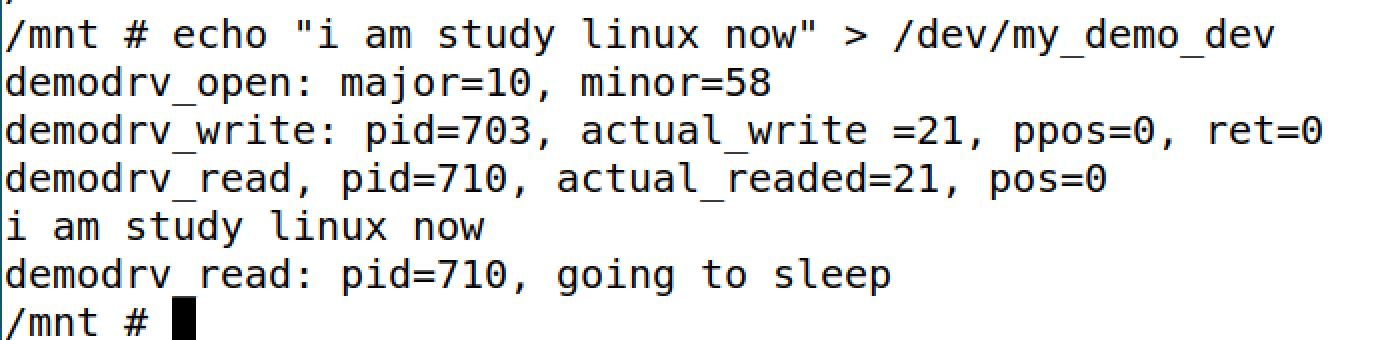


看到cat进程的pid是710，由于设备里没数据，所以，它进入睡眠状态了。



我们也可以通过top命令来看，cat进程的确是睡眠状态。

接下来使用echo命令来写操作。



从log里可以看到，写操作也就是echo命令是另外一个进程，pid为703.

写完之后，我们看到cat命令把echo命令写入的字符串打印出来了，然后继续睡眠去了。

测试成功了！

## 进阶思考

这个字符设备最经典的操作了，怎么进入睡眠，怎么被唤醒，希望大家好好体会一下。

1. 当wake\_up（）函数唤醒一个等待队列，它是唤醒所有进程还是一个进程？
2. 阅读现有内核代码的设备驱动，找一个现成的驱动，研究他们是怎么进行睡眠等待和唤醒的？

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>

