

《奔跑吧 Linux 内核*入门篇》重要实验说明 - 必读

注意：

1. 本书的所有实验都是基于优麒麟 Linux 18.04.2，若您安装了其他发行版或者版本，遇到问题，请自行解决。
2. 建议在尽量物理机上安装优麒麟，实在不行，使用 vmware player 虚拟机也可以。
3. 笨叔会提供安装和配置好整套环境的 vmware 虚拟机镜像给有需要的小伙伴下载。请留意奔跑吧 Linux 社区微信公众号和异步社区官网。
4. 《奔跑吧 Linux 内核*入门篇》是基于 ARM32 架构来表述，大部分实验基于 arm32 架构。
5. 实验代码是在：https://github.com/figozhang/runninglinuxkernel_4.0
分支：rlk_basic
实验代码目录：rlk_lab/rlk_basic
6. 图书勘误：<https://github.com/figozhang/Running-LinuxKernel>
7. 本实验采用的源代码版本：
 - * Linux 内核： 4.0
 - * arm-linux-gnueabi-gcc 编译器： 5.5
 - * aarch64-linux-gnu-gcc 编译器： 5.5

在 ubuntu 里请使用 update-alternatives 命令来切换 gcc 编译到 5.x 版本。
请不要使用高版本的 gcc 编译器，否则会编译有问题。

1. 配套资源下载

1. 配备全球最丰富的教学和学习资源
配套约 600 页课件 课件已经上传到异步社区和 github。
下载地址：
<https://www.epubit.com/book/detail/39321>
或者：
<https://github.com/figozhang/Running-LinuxKernel>
2. 配套全套实验代码：https://github.com/figozhang/runninglinuxkernel_4.0
分支：rlk_basic
实验代码目录：rlk_lab/rlk_basic
3. 配套全套实验环境的 vmware 虚拟机镜像 （基于优麒麟 Linux 18.04）
下载地址：
建议大家到百度网盘上下载最新版本。（2019 年 6 月更新到 v3.0 版本）
百度网盘：https://pan.baidu.com/s/1XIJzdUBH_V7lwNRH1R4jrg

异步社区：<https://www.epubit.com/book/detail/39321> （只有 v1.0 版本）

谷歌网盘: https://drive.google.com/open?id=1cTRQ73f0yzhTBoNt6rhHbkbbPTl8j_3P

4. 配套全套实验的 docker 镜像
<https://hub.docker.com/r/benshushu/runninglinuxkernel>
5. 配套补充视频, 10 小时高清, 33 集真情播放 从 3 月 25 号起, 33 集奔跑吧高清视频在 B 站首发, 每天一集, 天天精彩。请到 B 站 (<https://www.bilibili.com/>) 搜索: 奔跑吧 linux 内核
6. 配套视频下载。
《奔跑吧》入门篇免费视频通过 GPL v2 授权, 大家可以自由下载。
链接: https://pan.baidu.com/s/1AvTkeOyyJUSBJ_K0pxip0Q
提取码: le2s
7. 配套音频节目 从 3 月 25 号起, 33 集配套音频节目上线喜马拉雅
<https://www.ximalaya.com/keji/22283701/>
8. 微信交流群, 请加奔跑吧客服微信号: runninglinuxkernel
9. 免费知识星球



2. 编译运行 busybox 小系统

假设 runninglinuxkernel_4.0 目录是在/home/rlk/rlk_basic/runninglinuxkernel_4.0

```
# cd /home/rlk/rlk_basic/runninglinuxkernel_4.0
# export ARCH=arm
# export CROSS_COMPILE=arm-linux-gnueabi-
# make vexpress_defconfig
```

如果是第一次运行 Qemu 虚拟机，那需要手工创建一个 console 节点。否则运行会提示错误。如下图所示。

```
rlk@ubuntu:runninglinuxkernel_4.0$ sh run.sh arm32
please create console device node first, and recompile kernel
```

```
# cd _install_arm32
# mkdir -p dev
# cd dev
# sudo mknod console c 5 1
```

然后编译内核。

```
# cd /home/rlk/rlk_basic/runninglinuxkernel_4.0
# make -j8 //这里假设你的 Linux 机器有 8 个 CPU
```

3. 内核模块编译和运行举例说明

下面以第 7 章，lab7 为例：

1. 进入 runninglinuxkernel_4.0/rlk_lab/rlk_basic/chapter_7_mm/lab7_pin_page 目录。

```
# cd /home/rlk/rlk_basic/runninglinuxkernel_4.0
# cd rlk_lab/rlk_basic/chapter_7_mm/lab7_pin_page
# export ARCH=arm
# export CROSS_COMPILE=arm-linux-gnueabi-
# make BASEINCLUDE=你机器上 runninglinuxkernel_4.0 的绝对路径
```

如图所示：

```
rlk@ubuntu:lab7_pin_page$ make BASEINCLUDE=/home/rlk/rlk_basic/runninglinuxkernel_4.0
make -C /home/rlk/rlk_basic/runninglinuxkernel_4.0 M=/home/rlk/rlk_basic/runninglinuxkernel_4.0/rlk_lab/rlk_basic/chapter_7_mm/lab7_pin_page modules;
make[1]: Entering directory '/home/rlk/rlk_basic/runninglinuxkernel_4.0'
  CC [M] /home/rlk/rlk_basic/runninglinuxkernel_4.0/rlk_lab/rlk_basic/chapter_7_mm/lab7_pin_page/mydevd_pin_page.o
  LD [M] /home/rlk/rlk_basic/runninglinuxkernel_4.0/rlk_lab/rlk_basic/chapter_7_mm/lab7_pin_page/mydevd_emo-pin-page.o
Building modules, stage 2.
MODPOST 1 modules
  CC /home/rlk/rlk_basic/runninglinuxkernel_4.0/rlk_lab/rlk_basic/chapter_7_mm/lab7_pin_page/mydevd_emo-pin-page.mod.o
  LD [M] /home/rlk/rlk_basic/runninglinuxkernel_4.0/rlk_lab/rlk_basic/chapter_7_mm/lab7_pin_page/mydevd_emo-pin-page.ko
make[1]: Leaving directory '/home/rlk/rlk_basic/runninglinuxkernel_4.0'
rlk@ubuntu:lab7_pin_page$
```

注意：这里 BASEINCLUDE 指定你当前 runninglinuxkernel_4.0 的目录路径，请确保路径是对的，这里要使用绝对路径。

另外也可以在把 `BASEINCLUDE` 设置到系统的环境变量里。比如修改用户目录下面的`.bashrc` 文件。

```
$ vim ~/.bashrc
```

在这个文件里添加

```
export BASEINCLUDE=你机器上 runninglinuxkernel_4.0 的绝对路径
```

然后在让环境变量生效。

```
$ source ~/.bashrc
```

编译 test 测试 app。

```
# arm-linux-gnueabi-gcc test_ok.c -o test_ok --static
```

如图所示：

```
rlk@ubuntu:lab7_pin_page$ arm-linux-gnueabi-gcc test_ok.c -o test_ok --static
rlk@ubuntu:lab7_pin_page$
```

然后把内核模块 `mydevdemo-pin-page.ko` 和测试程序 `test_ok` 拷贝到 `runninglinuxkernel_4.0/kmodules` 目录下面。

```
rlk@ubuntu:lab7_pin_page$ cp mydevdemo-pin-page.ko /home/rlk/rlk_basic/runninglinuxkernel_4.0/kmodules/
rlk@ubuntu:lab7_pin_page$ cp test_ok /home/rlk/rlk_basic/runninglinuxkernel_4.0/kmodules/
rlk@ubuntu:lab7_pin_page$
```

可以在另一个终端上运行如下脚本启动 Qemu。

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

在 Qemu 虚拟机，在虚拟机的 `mnt` 目录就可以看到刚才拷贝的内核模块和应用程序了。

```
/mnt # ls
README                               mydevdemo-pin-page.ko  test_ok
/mnt #
```



```
# 在 Qemu 虚拟机里面
#cd /mnt
# insmod mydevdemo-pin-page.ko
#./test_ok
```

关闭 Qemu 虚拟机，可以输入组合键：Ctrl + A，然后快速按下“x”键

4.编译运行 debian rootfs 系统

奔跑吧 linux 内核支持运行 debian 系统，这样可以通过 apt 命令很方便的安装 arm32 或者 arm64 的软件，而不需要麻烦的交叉编译了。

1. 编译 arm32 的内核

```
$ cd runninglinuxkernel_4.0
$ ./run_debian_arm32.sh build_kernel
```

2. 编译 arm32 的 debian 的 rootfs 系统

```
$sudo ./run_debian_arm32.sh build_rootfs
```

3. 运行

```
$ ./run_debian_arm32.sh run
```

如读者需要跑 arm64 的 debian 系统，可以使用如下方法来编译和运行。

编译 arm64 的内核

```
$ cd runninglinuxkernel_4.0
$ ./run_debian_arm64.sh build_kernel
```

编译 arm64 的 debian 的 rootfs 系统

```
$sudo ./run_debian_arm64.sh build_rootfs
```

运行

```
$ ./run_debian_arm64.sh run
```

debian 系统的用户名为 root，密码为：123

系统登录后，先运行 apt update 命令来更新源，然后就可以使用 apt install 命令来安装软件包。

若 apt update 命令遇到问题，可以先 ifconfig 命令查看 eth0 网络设备是否存在？

另外有可能是系统时间比较旧了，可以使用 date 命令来设置日期。

```
root@benshushu:~# date -s 2019-04-25 #假设最新日期是2019年4月25日
Thu Apr 25 00:00:00 UTC 2019
```

在 debian 系统中运行《奔跑吧 linux 内核*入门篇》的实验，和之前的 busybox 小系统是类似的，都需要交叉编译内核模块，然后在拷贝到 QEMU 系统中运行。

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：



微信号: Running-LinuxKernel

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！

shop115683645.taobao.com

Linux视频课程



微信公众号：奔跑吧 linux 社区

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

shop115683645.taobao.com

配套视频 旗舰篇

第1季 内存管理



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



旗舰篇一次订阅，持续更新

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！



实战死机专题课程目录	
课程名称	时长
上集x86_64	
实战死机专题（上集）part1-kdump+crash介绍	0:30:09
实战死机专题（上集）part2-crash命令详解	0:28:15
实战死机专题（上集）part3-实战lab1	0:12:38
实战死机专题（上集）part4-实战lab2	0:11:03
实战死机专题（上集）part4-实战lab3	0:06:48
实战死机专题（上集）part4-实战lab4	0:15:28
实战死机专题（上集）part4-实战lab5	0:12:21
实战死机专题（上集）part4-实战lab6	0:24:07
实战死机专题（上集）part4-实战lab7	0:59:34
下集arm64	
实战死机专题(下集)part1	0:13:19
实战死机专题(下集)part2	0:20:47
实战死机专题(下集)part3	0:11:22
实战死机专题(下集)part4	0:33:01

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

