奔跑吧LINUX内核 * 入门篇

第三章 内核编译和调试

笨叔叔

目 录

- Linux系统内核配置
- 实验

Linux内核配置

内核配置

- make config
  - ✓ 这是基于文本的一种传统的配置方式。它会为内核支持的每一个特性向用户提问，如果用户输入“y”，则把该特性编译进内核；如果输入“m”，则把该特性变成以模块；如果输入为“n”，则表示不编译该特性

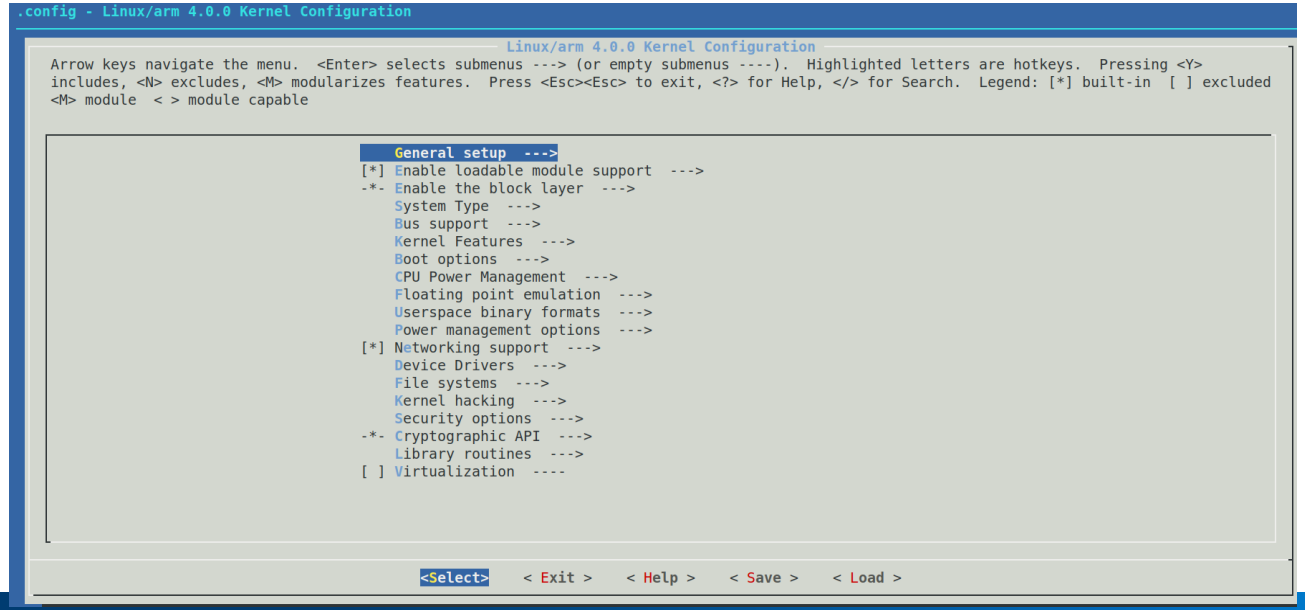
```
ben@ubuntu:~/work/runninglinuxkernel_4.0$ make config
scripts/kconfig/conf --oldaskconfig Kconfig
arch/arm/Kconfig:1399:warning: 'HZ_FIXED': number is invalid
arch/arm/Kconfig:1400:warning: 'HZ_FIXED': number is invalid
*
* Linux/arm 4.0.0 Kernel Configuration
*
*
* General setup
*
Cross-compiler tool prefix (CROSS_COMPILE) []
```

内核配置

- make oldconfig
  - ✓ make oldconfig和make config很类似，也是基于文本的配置工具，只不过它是在现有的内核配置文件的基础上建立一个新的配置文件，在有新的配置选项时会向用户提问。

内核配置

- make menuconfig
  - ✓ make menuconfig是一种基于文本模式的图形用户界面，用户可以通过移动光标来浏览内核支持的特性



内核配置文件

- .config配置文件
  - ✓ 内核内置工具最终会在Linux内核源代码的根目录下生成一个隐藏文件，即.config文件

```
#
# Automatically generated file; DO NOT EDIT.
# Linux/arm 4.0.0 Kernel Configuration
#
CONFIG_ARM=y
CONFIG_ARM_HAS_SG_CHAIN=y
CONFIG_MIGHT_HAVE_PCI=y
CONFIG_SYS_SUPPORTS_APM_EMULATION=y
CONFIG_HAVE_PROC_CPU=y
CONFIG_NO_IOPORT_MAP=y
CONFIG_STACKTRACE_SUPPORT=y
CONFIG_LOCKDEP_SUPPORT=y
CONFIG_TRACE_IRQFLAGS_SUPPORT=y
CONFIG_RWSEM_XCHGADD_ALGORITHM=y
CONFIG_GENERIC_HWEIGHT=y
CONFIG_GENERIC_CALIBRATE_DELAY=y
CONFIG_NEED_DMA_MAP_STATE=y
CONFIG_ARCH_SUPPORTS_UPROBES=y
CONFIG_VECTORS_BASE=0xffff0000
CONFIG_ARM_PATCH_PHYS_VIRT=y
CONFIG_GENERIC_BUG=y
CONFIG_DEFCONFIG_LIST="/lib/modules/$UNAME_RELEASE/.config"
CONFIG_IRQ_WORK=y
CONFIG_BUILDTIME_EXTABLE_SORT=y

```

runninglinuxkernel对应的.config文件

生成.config文件

- 使用板级的配置文件
  - ✓ 一些芯片公司通常会提供基于某款SoC芯片的开发板，读者可以基于此开发板来快速开发产品原型。芯片公司同时会提供板级开发板包，其中包含移植好的Linux内核。

```
ben@ubuntu:~/work/runninglinuxkernel_4.0/arch/arm/configs$ ls
acs5k_defconfig            collie_defconfig          integrator_defconfig  moxart_defconfig      pcm027_defconfig          socfpga_defconfig
acs5k_tiny_defconfig       corgi_defconfig           iop13xx_defconfig     msm_defconfig         pleb_defconfig            spear13xx_defconfig
am200epdkit_defconfig      davinci_all_defconfig     iop32x_defconfig      multi_v5_defconfig    prima2_defconfig          spear3xx_defconfig
ape6evm_defconfig          dove_defconfig            iop33x_defconfig      multi_v7_defconfig    pxa168_defconfig          spear6xx_defconfig
armadillo800eva_defconfig  ebsa110_defconfig         ixp4xx_defconfig      mv78xx0_defconfig     pxa255-idp_defconfig      spitz_defconfig
assabet_defconfig          efm32_defconfig           jornada720_defconfig  mvebu_v5_defconfig    pxa3xx_defconfig          sunxi_defconfig
at91_dt_defconfig          em_x270_defconfig         keystone_defconfig    mvebu_v7_defconfig    pxa910_defconfig          tct_hammer_defconfig
axm55xx_defconfig          ep93xx_defconfig          ks8695_defconfig      mxs_defconfig         qcom_defconfig            tegra_defconfig
badge4_defconfig           eseries_pxa_defconfig     kzm9g_defconfig       neponset_defconfig    raumfeld_defconfig        trizeps4_defconfig
bcm2835_defconfig          exynos_defconfig          lart_defconfig        netwinder_defconfig   realview_defconfig        u300_defconfig
bcm_defconfig              ezx_defconfig             lpc32xx_defconfig     netx_defconfig        realview-smp_defconfig    u8500_defconfig
bockw_defconfig            footbridge_defconfig      lpd270_defconfig      nhk8815_defconfig     rpc_defconfig             versatile_defconfig
cerfcube_defconfig         h3600_defconfig           lubbock_defconfig     nuc910_defconfig      s3c2410_defconfig         vexpress_defconfig
clps711x_defconfig         h5000_defconfig           mackerel_defconfig    nuc950_defconfig      s3c6400_defconfig         viper_defconfig
cm_x2xx_defconfig          hackkit_defconfig         magician_defconfig    nuc960_defconfig      s5pv210_defconfig         vt8500_v6_v7_defconfig
cm_x300_defconfig          hisi_defconfig            mainstone_defconfig   omap1_defconfig       sama5_defconfig           xcep_defconfig
cns3420vb_defconfig        imote2_defconfig          marzen_defconfig      omap2plus_defconfig   shannon_defconfig         zeus_defconfig
colibri_pxa270_defconfig   imx_v4_v5_defconfig      mini2440_defconfig    orion5x_defconfig     shmobile_defconfig
colibri_pxa300_defconfig   imx_v6_v7_defconfig      mmp2_defconfig        palmz72_defconfig     simpad_defconfig
ben@ubuntu:~/work/runninglinuxkernel_4.0/arch/arm/configs$
```

arch/arm/configs目录下包含了众多的ARM板子的配置文件

生成.config文件

- 使用系统配置文件
  - ✓ 当我们需要编译电脑中的Linux系统内核时，可以使用系统自带的config文件。以优麒麟18.04系统为例，boot目录下面有一个config-4.15.0-22-generic文件

```
ben@ubuntu:/boot$ ls -l
total 132996
-rw-r--r-- 1 root root  1536934 Apr 23 21:56 abi-4.15.0-20-generic
-rw-r--r-- 1 root root  1537177 May 14 22:41 abi-4.15.0-22-generic
-rw-r--r-- 1 root root   216807 Apr 23 21:56 config-4.15.0-20-generic
-rw-r--r-- 1 root root   216807 May 14 22:41 config-4.15.0-22-generic
drwxr-xr-x 5 root root     4096 Jun  4 01:44 grub
-rw-r--r-- 1 root root 53760078 Apr 29 02:03 initrd.img-4.15.0-20-generic
-rw-r--r-- 1 root root 53757728 Jun  1 05:59 initrd.img-4.15.0-22-generic
-rw-r--r-- 1 root root   182704 Jan 28  2016 memtest86+.bin
-rw-r--r-- 1 root root   184380 Jan 28  2016 memtest86+.elf
-rw-r--r-- 1 root root   184840 Jan 28  2016 memtest86+_multiboot.bin
-rw-r--r-- 1 root root        0 Apr 23 21:56 retpoline-4.15.0-20-generic
-rw-r--r-- 1 root root        0 May 14 22:41 retpoline-4.15.0-22-generic
-rw------- 1 root root  4038188 Apr 23 21:56 System.map-4.15.0-20-generic
-rw------- 1 root root  4039542 May 14 22:41 System.map-4.15.0-22-generic
-rw-r--r-- 1 root root  8249080 Apr 26 11:40 vmlinuz-4.15.0-20-generic
-rw------- 1 root root  8253176 May 16 23:12 vmlinuz-4.15.0-22-generic
```

Boot目录下面有内核配置文件

实验

实验1：通过QEMU调试ARM Linux内核

- 实验目的
  - ✓ 熟悉如何使用QEMU调试Linux内核。

实验2：通过QEMU调试ARMv8的Linux内核

- 实验目的
  - ✓ 熟悉如何使用QEMU调试ARMv8的 Linux内核。

实验3：通过Eclipse+QEMU单步调试内核

- 实验目的
  - ✓ 熟悉如何使用Eclipse+QEMU以图形方式单步调试Linux内核。

实验4：在QEMU中添加文件系统的支持

- 实验目的
  - ✓ 熟悉如何在QEMU中添加文件系统的支持。

BACKUP


shop115683645.taobao.com
Linux视频课程
奔跑吧
LINUX社区
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
奔跑吧Linux内核
奔跑吧 LINUX社区
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

第1季旗舰篇课程目录

课程名称	时长
序言一： Linux内核学习方法论	0:09:13
序言二： 学习前准备	
序言2.1 Linux发行版和开发板的选择	0:13:56
序言2.2 搭建Qemu+gdb单步调试内核	0:13:51
序言2.3 搭建Eclipse图形化调试内核	0:10:59
实战运维1：查看系统内存信息的工具（一）	0:20:19
实战运维2：查看系统内存信息的工具（二）	0:16:32
实战运维3：读懂内核log中的内存管理信息	0:25:35
实战运维4：读懂 proc meminfo	0:27:59
实战运维5：Linux运维能力进阶线路图	0:09:40
实战运维6：Linux内存管理参数调优(一)	0:19:46
实战运维7：Linux内存管理参数调优(二)	0:31:20
实战运维8：Linux内存管理参数调优(三)	0:22:58
运维高级如何单步调试RHEL— CENTOS7的内核一	0:15:45
运维高级如何单步调试RHEL— CENTOS7的内核二	0:41:28
vim:打造比source insight更强更好用的IDE(一)	0:24:58
vim:打造比source insight更强更好用的IDE(二)	0:20:28
vim:打造比source insight更强更好用的IDE(三)	0:23:25
实战git项目和社区patch管理	
2.0 Linux内存管理背景知识介绍	
奔跑2.0.0 内存管理硬件知识	0:15:25
奔跑2.0.1 内存管理总览一	0:23:27
奔跑2.0.2 内存管理总览二	0:07:35
奔跑2.0.3 内存管理常用术语	0:09:49
奔跑2.0.4 内存管理究竟管些什么东西	0:28:02
奔跑2.0.5 内存管理代码框架导读	0:38:09
2.1 Linux内存初始化	
奔跑2.1.0 DDR简介	0:06:47
奔跑2.1.1 物理内存三大数据结构	0:19:39
奔跑2.1.2 物理内存初始化	0:11:13
奔跑2.1 内存初始化之代码导读一	0:43:54
奔跑2.1 内存初始化之代码导读二	0:23:31
奔跑2.1 代码导读C语言部分（一）	0:27:34
奔跑2.1 代码导读C语言部分（二）	0:21:28
2.2 页表的映射过程	
奔跑2.2.0 ARM32页表的映射	0:08:54
奔跑2.2.1 ARM64页表的映射	0:10:58
奔跑2.2.2 页表映射例子分析	0:11:59
奔跑2.2.3 ARM32页表映射那些奇葩的事	0:09:42
2.3 内存布局图	
奔跑2.3.1 内存布局一	0:10:35
奔跑2.3.2 内存布局二	0:13:30
2.4 分配物理页面	
奔跑2.4.1 伙伴系统原理	0:10:10
奔跑2.4.2 Linux内核中的伙伴系统和碎片化	0:11:14
奔跑2.4.3 Linux的页面分配器	0:21:37
2.5 slab分配器	
奔跑2.5.1 slab原理和核心数据结构	0:18:36
奔跑2.5.2 Linux内核中slab机制的实现	0:16:56
2.6 vmalloc分配	
奔跑2.6 vmalloc分配	0:15:48
2.7 VMA操作	
奔跑2.7 VMA操作	0:16:42
2.8 malloc分配器	
奔跑2.8.1 malloc的三个迷惑	0:17:41
奔跑2.8.2 内存管理的三个重要的函数	0:17:38
2.9 mmap分析	
奔跑2.9 mmap分析	0:23:14
2.10 缺页中断处理	
奔跑2.10.1 缺页中断一	0:31:07
奔跑2.10.2 缺页中断二	0:16:58
2.11 page数据结构	
奔跑2.11 page数据结构	0:29:41
2.12 反向映射机制	
奔跑2.12.1 反向映射机制的背景介绍	0:19:01
奔跑2.12.2 RMAP四部曲	0:07:31
奔跑2.12.3 手撕Linux2.6.11上的反向映射机制	0:07:35
奔跑2.12.4 手撕Linux4.x上的反向映射机制	0:10:08
2.13 回收页面	
奔跑2.13 页面回收一	0:16:07
奔跑2.13 页面回收二	0:11:41
2.14 匿名页面的生命周期	0:26:16
2.15 页面迁移	0:19:07
2.16 内存规整	0:24:03
2.17 KSM	0:28:17
2.20 Meltdown漏洞分析	
奔跑2.20.1 Meltdown背景知识	0:10:13
奔跑2.20.2 CPU体系结构之指令执行	0:11:25
奔跑2.20.3 CPU体系结构之乱序执行	0:11:03
奔跑2.20.4 CPU体系结构之异常处理	0:03:48
奔跑2.20.5 CPU体系结构之cache	0:10:56
奔跑2.20.6 进程地址空间和页表及TLB	0:17:39
奔跑2.20.7 Meltdown漏洞分析	0:06:04
奔跑2.20.8 Meltdown漏洞分析之x86篇	0:12:07
奔跑2.20.9 ARM64上的KPTI解决方案	0:25:39
代码导读	
奔跑2.1 内存初始化之代码导读一	0:43:54
奔跑2.1 内存初始化之代码导读二	0:23:31
奔跑2.1 代码导读C语言部分（一）	0:27:34
奔跑2.1 代码导读C语言部分（二）	0:21:28
代码导读3页表映射	1:12:40
代码导读4分配物理页面	0:55:57
git入门和实战	
git入门与实战：节目总览	0:08:48
git入门与实战1：建立本地的git仓库	0:30:53
git入门与实战2：快速入门	0:12:45
git入门与实战3：分支管理	0:24:27
git入门与实战4：冲突解决	0:20:20
git入门和实战5：提交更改	0:12:15
git入门和实战6：远程版本库	0:13:26
git入门和实战7：内核开发和实战	0:15:52
git入门和实战8：实战rebase到最新Linux内核代码	0:18:07
git入门和实战9：给内核发补丁	0:13:57

第2季旗舰篇课程目录	
课程名称	时长
进程管理	
进程管理1基本概念	0:52:16
进程管理2进程创建	0:53:24
进程管理3进程调度	0:54:51
进程管理4多核调度	0:49:38
中断管理	
中断管理1基本概念	1:04:27
中断管理2中断处理part1	0:46:28
中断管理2中断处理part2	0:10:19
中断管理3下半部机制	0:55:57
中断管理4面试题目	1:13:57
锁机制	
锁机制入门1基本概念	0:56:16
锁机制入门2-Linux常用的锁	0:54:01

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

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！

微店 奔跑吧Linux内核

扫码识别

微店二维码

淘宝店二维码

《奔跑吧Linux内核 * 入门篇》相关的免费视频，或者更多更精彩更in的内容，请关注奔跑吧Linux社区微信公众号


奔跑吧
LINUX社区
旗舰篇一次订阅， 持续更新
微信号: Runing-LinuxKernel