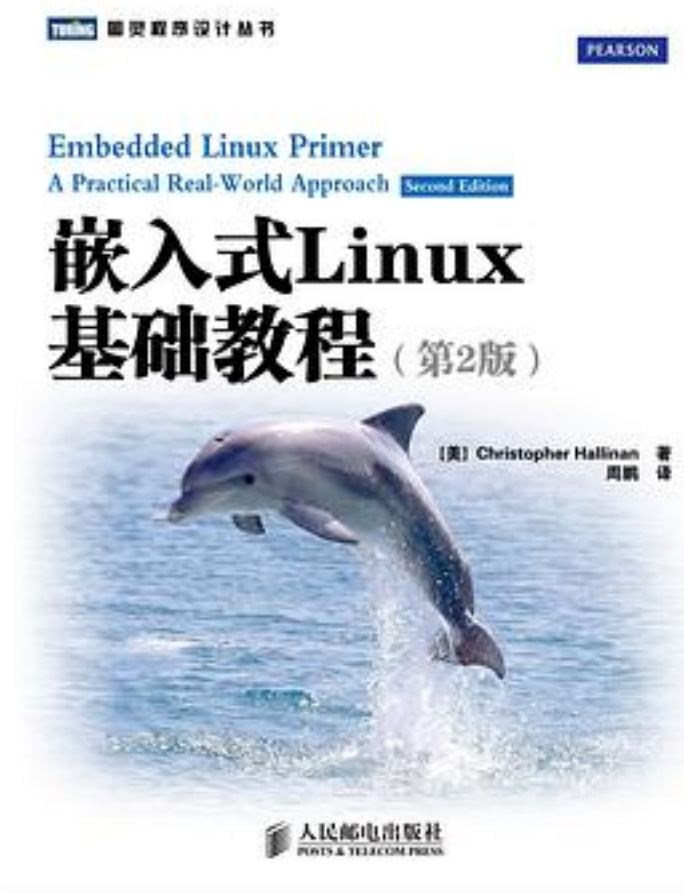


# 嵌入式Linux基础教程（第2版）



[嵌入式Linux基础教程（第2版）\\_下载链接1](#)

著者:[美] Christopher Hallinan

出版者:人民邮电出版社

出版时间:2012-5

装帧:平装

isbn:9787115278272

内容简介：

广泛的硬件支持、高效稳定的内核、开源共享的软件、优秀的开发工具、完善的网络通信和文件管理机制等特点，使嵌入式Linux获得了广泛应用，已成为嵌入式开发的主流

平台。

本书是嵌入式Linux领域的名著。全面深入而又简明地阐述了构建嵌入式Linux系统的精髓。书中不仅剖析了嵌入式Linux系统，而且讲述了处理器、内核、引导加载程序、设备驱动程序、文件系统等关键组件，介绍了嵌入式Linux系统的开发工具、调试技术。作者多年积累总结的嵌入式Linux开发技巧和提示，无论对初学者还是有经验的开发人员，都弥足珍贵。这一版不仅对原有章节进行了全面的修订、更新和改进，还新增了udev、USB和开源构建系统等内容。

作者介绍:

作者简介:

Christopher Hallinan 著名嵌入式Linux技术专家。现任Mentor Graphics公司技术市场工程师，曾任MontaVista软件公司现场应用工程师，3Com公司工程总监，Crosscomm公司工程总监。他有25年以上网络和通信产品的软硬件开发经验。曾担任Linux咨询师，提供定制Linux主板接口、设备驱动程序和引导装入程序等方面的解决方案

目录: 目 录

第1章 入门 1

1.1 为什么选择Linux 1

1.2 嵌入式Linux现状 2

1.3 开源和GPL 2

1.4 标准及相关组织 3

1.4.1 Linux标准基础 3

1.4.2 Linux基金会 4

1.4.3 电信级Linux 4

1.4.4 移动Linux计划: Moblin 4

1.4.5 服务可用性论坛 5

1.5 小结 5

第2章 综述 6

2.1 嵌入与非嵌入 6

2.2 剖析嵌入式系统 7

2.2.1 典型的嵌入式Linux开发环境 8

2.2.2 启动目标板 9

2.2.3 引导内核 10

2.2.4 内核初始化: 概述 12

2.2.5 第一个用户空间进程: init 13

2.3 存储 14

2.3.1 闪存 14

2.3.2 NAND型闪存 15

2.3.3 闪存的用途 16

2.3.4 闪存文件系统 17

2.3.5 内存空间 17

2.3.6 执行上下文 19

2.3.7 进程虚拟内存 20

2.3.8 交叉开发环境 21

2.4 嵌入式Linux发行版 22

2.4.1 商业Linux发行版 24

2.4.2 打造自己的Linux发行版 24

2.5 小结 24

第3章 处理器基础	26
3.1 独立处理器	26
3.1.1 IBM 970FX	27
3.1.2 英特尔奔腾M	27
3.1.3 英特尔凌动TM	28
3.1.4 飞思卡尔MPC7448	28
3.1.5 配套芯片组	28
3.2 集成处理器：片上系统	30
3.2.1 Power架构	30
3.2.2 飞思卡尔Power架构	30
3.2.3 飞思卡尔PowerQUICC I	31
3.2.4 飞思卡尔PowerQUICC II	32
3.2.5 PowerQUICC II Pro	32
3.2.6 飞思卡尔PowerQUICC III	33
3.2.7 飞思卡尔QorlQTM	34
3.2.8 AMCC Power架构	36
3.2.9 MIPS	38
3.2.10 Broadcom MIPS	38
3.2.11 其他MIPS	39
3.2.12 ARM	40
3.2.13 德州仪器ARM	40
3.2.14 飞思卡尔ARM	42
3.2.15 其他ARM处理器	43
3.3 其他架构	43
3.4 硬件平台	43
3.4.1 CompactPCI	43
3.4.2 ATCA	44
3.5 小结	44
第4章 Linux内核：不同的视角	46
4.1 背景知识	46
4.1.1 内核版本	47
4.1.2 内核源码库	48
4.1.3 使用git下载内核代码	49
4.2 Linux内核的构造	49
4.2.1 顶层源码目录	49
4.2.2 编译内核	50
4.2.3 内核主体：vmlinux	52
4.2.4 内核镜像的组成部分	53
4.2.5 子目录的布局	56
4.3 内核构建系统	57
4.3.1 .config文件	57
4.3.2 配置编辑器	58
4.3.3 Makefile目标	61
4.4 内核配置	66
4.4.1 定制配置选项	68
4.4.2 内核Makefile	71
4.5 内核文档	72
4.6 获得定制的Linux内核	72
4.7 小结	73
第5章 内核初始化	75
5.1 合成内核镜像：Piggy及其他	75
5.1.1 Image对象	78
5.1.2 与具体架构相关的对象	78
5.1.3 启动加载程序	79

- 5.1.4 引导消息 80
- 5.2 初始化时的控制流 83
  - 5.2.1 内核入口：head.o 84
  - 5.2.2 内核启动：main.c 85
  - 5.2.3 架构设置 86
- 5.3 内核命令行的处理 87
- 5.4 子系统初始化 92
- 5.5 init线程 94
  - 5.5.1 通过initcalls进行初始化 95
  - 5.5.2 initcall\_debug 96
  - 5.5.3 最后的引导步骤 97
- 5.6 小结 98
- 第6章 用户空间初始化 100
  - 6.1 根文件系统 100
    - 6.1.1 FHS：文件系统层次结构标准 101
    - 6.1.2 文件系统布局 101
    - 6.1.3 最小化的文件系统 102
    - 6.1.4 嵌入式根文件系统带来的挑战 103
    - 6.1.5 试错法 104
    - 6.1.6 自动化文件系统构建工具 104
  - 6.2 内核的最后一些引导步骤 104
    - 6.2.1 第一个用户空间程序 106
    - 6.2.2 解决依赖关系 106
    - 6.2.3 定制的初始进程 107
  - 6.3 init进程 107
    - 6.3.1 inittab 109
    - 6.3.2 Web服务器启动脚本示例 111
  - 6.4 初始RAM磁盘 112
    - 6.4.1 使用initrd进行引导 113
    - 6.4.2 引导加载程序对initrd的支持 113
    - 6.4.3 initrd的奥秘所在：  
linuxrc 115
    - 6.4.4 initrd探究 116
    - 6.4.5 构造initrd镜像 116
  - 6.5 使用initramfs 117
  - 6.6 关机 119
  - 6.7 小结 120
- 第7章 引导加载程序 122
  - 7.1 引导加载程序的作用 122
  - 7.2 引导加载程序带来的挑战 123
    - 7.2.1 DRAM控制器 123
    - 7.2.2 闪存与RAM 123
    - 7.2.3 镜像的复杂性 124
    - 7.2.4 执行环境 126
  - 7.3 通用引导加载程序：Das U-Boot 126
    - 7.3.1 获取U-Boot 127
    - 7.3.2 配置U-Boot 127
    - 7.3.3 U-Boot的监控命令 129
    - 7.3.4 网络操作 130
    - 7.3.5 存储子系统 132
    - 7.3.6 从磁盘引导 133
  - 7.4 移植U-Boot 133
    - 7.4.1 EP405的U-Boot移植 133
    - 7.4.2 U-Boot Makefile中的配置目标 134

- 7.4.3 EP405的第一次构建 135
- 7.4.4 EP405 处理器初始化 136
- 7.4.5 与具体板卡相关的初始化 138
- 7.4.6 移植总结 141
- 7.4.7 U-Boot镜像格式 141
- 7.5 设备树对象（扁平设备树） 143
  - 7.5.1 设备树源码 145
  - 7.5.2 设备树编译器 148
  - 7.5.3 使用DTB的其他内核镜像 148
- 7.6 其他引导加载程序 149
  - 7.6.1 Lilo 149
  - 7.6.2 GRUB 150
  - 7.6.3 其他更多的引导加载程序 151
- 7.7 小结 152
- 第8章 设备驱动程序基础 154
  - 8.1 设备驱动程序的概念 154
    - 8.1.1 可加载模块 155
    - 8.1.2 设备驱动程序架构 155
    - 8.1.3 最小设备驱动程序示例 156
    - 8.1.4 模块构建的基础设施 157
    - 8.1.5 安装设备驱动程序 160
    - 8.1.6 加载模块 160
    - 8.1.7 模块参数 161
  - 8.2 模块工具 162
    - 8.2.1 insmod 162
    - 8.2.2 lsmod 163
    - 8.2.3 modprobe 163
    - 8.2.4 depmod 164
    - 8.2.5 rmmod 165
    - 8.2.6 modinfo 165
  - 8.3 驱动程序方法 166
    - 8.3.1 驱动程序中的文件系统操作 166
    - 8.3.2 设备号的分配 169
    - 8.3.3 设备节点和mknod 169
  - 8.4 综合应用 170
  - 8.5 在内核源码树外构建驱动 171
  - 8.6 设备驱动程序和GPL 172
  - 8.7 小结 173
- 第9章 文件系统 174
  - 9.1 Linux文件系统概念 175
  - 9.2 ext2 176
    - 9.2.1 挂载文件系统 177
    - 9.2.2 检查文件系统的完整性 178
  - 9.3 ext3 180
  - 9.4 ext4 182
  - 9.5 ReiserFS 182
  - 9.6 JFFS2 183
  - 9.7 cramfs 185
  - 9.8 网络文件系统 187
  - 9.9 伪文件系统 190
    - 9.9.1 /proc文件系统 191
    - 9.9.2 sysfs 194
  - 9.10 其他文件系统 196
  - 9.11 创建简单的文件系统 197

9.12 小结	199
第10章 MTD子系统	201
10.1 MTD概述	201
10.1.1 开启MTD服务	202
10.1.2 MTD基础	203
10.1.3 在目标板上配置MTD	205
10.2 MTD分区	205
10.2.1 使用Redboot分区表进行分区	206
10.2.2 使用内核命令行传递分区信息	210
10.2.3 映射驱动	211
10.2.4 闪存芯片驱动	213
10.2.5 与具体板卡相关的初始化	214
10.3 MTD工具	215
10.4 UBI文件系统	219
10.4.1 配置UBIFS	219
10.4.2 构建UBIFS镜像	220
10.4.3 使用UBIFS作为根文件系统	222
10.5 小结	222
第11章 BusyBox	224
11.1 BusyBox简介	224
11.2 BusyBox的配置	225
11.3 BusyBox的操作	227
11.3.1 BusyBox的init	230
11.3.2 rcS初始化脚本示例	232
11.3.3 BusyBox在目标板上的安装	233
11.3.4 BusyBox小应用	235
11.4 小结	235
第12章 嵌入式开发环境	237
12.1 交叉开发环境	237
12.2 对主机系统的要求	241
12.3 为目标板提供服务	242
12.3.1 TFTP服务器	242
12.3.2 BOOTP/DHCP 服务器	243
12.3.3 NFS服务器	245
12.3.4 目标板使用NFS挂载根文件系统	247
12.3.5 U-Boot中使用NFS挂载根文件系统的例子	248
12.4 小结	250
第13章 开发工具	252
13.1 GNU调试器 (GDB)	252
13.1.1 调试核心转储	253
13.1.2 执行GDB	255
13.1.3 GDB中的调试会话	257
13.2 数据显示调试器	258
13.3 cbrowser/cscope	260
13.4 追踪和性能评测工具	261
13.4.1 strace	261
13.4.2 strace命令行选项	264
13.4.3 ltrace	266
13.4.4 ps	267
13.4.5 top	269
13.4.6 mtrace	270
13.4.7 dmalloc	272

- 13.4.8 内核oops 274
- 13.5 二进制工具 276
  - 13.5.1 readelf 277
  - 13.5.2 使用readelf查看调试信息 278
  - 13.5.3 objdump 280
  - 13.5.4 objcopy 281
- 13.6 其他二进制实用程序 281
  - 13.6.1 strip 281
  - 13.6.2 addr2line 282
  - 13.6.3 strings 282
  - 13.6.4 ldd 282
  - 13.6.5 nm 283
  - 13.6.6 prelink 283
- 13.7 小结 284
- 第14章 内核调试技术 286
  - 14.1 内核调试带来的挑战 286
  - 14.2 使用KGDB进行内核调试 287
    - 14.2.1 KGDB的内核配置 288
    - 14.2.2 在开启KGDB时引导目标板 289
    - 14.2.3 一些有用的内核断点 293
    - 14.2.4 KGDB与控制台共享一个串行端口 294
    - 14.2.5 调试非常早期的内核代码 295
    - 14.2.6 主线内核对KGDB的支持 296
  - 14.3 内核调试技术 297
    - 14.3.1 gdb远程串行协议 298
    - 14.3.2 调试优化的内核代码 300
    - 14.3.3 GDB的用户自定义命令 307
    - 14.3.4 有用的内核GDB宏 307
    - 14.3.5 调试可加载模块 315
    - 14.3.6 printk调试 319
    - 14.3.7 Magic SysReq key 320
  - 14.4 硬件辅助调试 321
    - 14.4.1 使用JTAG探测器对闪存进行编程 322
    - 14.4.2 使用JTAG探测器进行调试 323
  - 14.5 不能启动的情况 326
    - 14.5.1 早期的串行端口调试输出 326
    - 14.5.2 转储printk的日志缓冲区 327
    - 14.5.3 使用KGDB调试内核异常 328
  - 14.6 小结 329
- 第15章 调试嵌入式Linux应用程序 331
  - 15.1 目标调试 331
  - 15.2 远程（交叉）调试 331
  - 15.3 调试共享程序库 335
  - 15.4 调试多个任务 340
    - 15.4.1 调试多个进程 340
    - 15.4.2 调试多线程应用程序 342
    - 15.4.3 调试引导加载程序/闪存代码 345
  - 15.5 其他远程调试选项 345
    - 15.5.1 使用串行端口进行调试 345
    - 15.5.2 附着到运行的进程上 346
  - 15.6 小结 346
- 第16章 开源构建系统 348
  - 16.1 为什么使用构建系统 348
  - 16.2 Scratchbox 349

- 16.2.1 安装Scratchbox 349
- 16.2.2 创建一个交叉编译目标 350
- 16.3 Buildroot 351
  - 16.3.1 安装Buildroot 352
  - 16.3.2 配置Buildroot 352
  - 16.3.3 构建Buildroot 353
- 16.4 OpenEmbedded 354
  - 16.4.1 OpenEmbedded的组成 355
  - 16.4.2 BitBake元数据 355
  - 16.4.3 配方基础 356
  - 16.4.4 任务 358
  - 16.4.5 类 359
  - 16.4.6 配置元数据 360
  - 16.4.7 构建镜像 361
- 16.5 小结 362
- 第17章 实时Linux 363
  - 17.1 什么是实时 363
    - 17.1.1 软实时 363
    - 17.1.2 硬实时 364
    - 17.1.3 Linux调度 364
    - 17.1.4 延时 364
  - 17.2 内核抢占 365
    - 17.2.1 抢占的障碍 366
    - 17.2.2 抢占模式 367
    - 17.2.3 SMP内核 368
    - 17.2.4 抢占延时的根源 369
  - 17.3 实时内核补丁 369
    - 17.3.1 实时补丁的特性 370
    - 17.3.2 O(1)调度器 371
    - 17.3.3 创建实时进程 372
  - 17.4 实时内核的性能分析 373
    - 17.4.1 使用Ftrace追踪内核行为 373
    - 17.4.2 检测抢占被关闭的延时 374
    - 17.4.3 检测唤醒延时 375
    - 17.4.4 检测中断被关闭的延时 377
    - 17.4.5 检测Soft Lockup 378
  - 17.5 小结 378
- 第18章 通用串行总线 379
  - 18.1 USB概述 379
    - 18.1.1 USB的物理拓扑结构 379
    - 18.1.2 USB的逻辑拓扑结构 381
    - 18.1.3 USB版本 382
    - 18.1.4 USB连接器 382
    - 18.1.5 USB线缆 383
    - 18.1.6 USB模式 384
  - 18.2 配置USB 384
  - 18.3 sysfs和USB设备命名 388
  - 18.4 实用的USB工具 390
    - 18.4.1 USB文件系统 390
    - 18.4.2 使用usbview 392
    - 18.4.3 USB 实用程序 (lsusb) 394
  - 18.5 通用USB子系统 395
    - 18.5.1 USB大容量存储类 395
    - 18.5.2 USB HID类 397



18.5.3 USB CDC类驱动	398
18.5.4 USB网络支持	400
18.6 USB调试	401
18.6.1 usbmon	402
18.6.2 实用USB杂记	403
18.7 小结	403
第19章 udev	405
19.1 什么是udev	405
19.2 设备发现	406
19.3 udev的默认行为	407
19.4 理解udev规则	409
19.4.1 Modalias	411
19.4.2 典型的udev规则配置	414
19.4.3 udev的初始系统设置	415
19.5 加载平台设备驱动程序	417
19.6 定制udev的行为	419
19.7 持久的设备命名	420
19.8 udev和busybox配合使用	423
19.8.1 busybox mdev	423
19.8.2 配置mdev	425
19.9 小结	425
附录A 可配置的U-Boot命令	427
附录B BusyBox命令	430
附录C SDRAM接口注意事项	440
附录D 开源资源	447
附录E 简单的BDI-2000配置文件	449
• • • • •	<a href="#">(收起)</a>

[嵌入式Linux基础教程（第2版）](#) [下载链接1](#)

## 标签

嵌入式

Linux

嵌入式系统

嵌入式Linux基础教程

linux

完全是嵌入式的介绍

软件开发

计算机

## 评论

内容仍然没有什么特别吸引人的地方，值得一提的是，书里提到几个（4个？）启动box，居然没有busybox了——难道busybox那玩意儿已经过时了？？？

如果你对嵌入式Linux不怎么熟悉，可以拿来一读，如果已经有了较深的理解可以快速阅读，因为里面的知识点是广泛而不深入，和第一行代码有得一拼，我还是比较喜欢这种类型的书的，可以快速阅读而扩展知识，先了解个大概的情况，然后再根据兴趣找到值得深入的点。因为嵌入式Linux基础不扎实，这本书我做了完整的笔记：<http://labrick.cc/books/android/>，以后打算长期进行阅读更新，发现这是一种很好的做笔记的方法，可以随时进行查看更改，很多时候保存在笔记本里的东西永久都不会再看了，显得很悲催。

很全面

好书 新世界

不适合嵌入式初学者阅读，但是似乎非常适合有一定经验的嵌入式工程师阅读。

优点是介绍了很多概念跟工具，读完能知道有哪些内容，行业上普遍是怎么做的，适合入门，不适合深入研读。

内容比较全面，尤其是启动引导部分比较详细，解决了很多开发板启动的疑问，内容有点老，期望有更新的书出现？

-----  
入门必读

-----  
错别字多

-----  
嵌入式linux入门的第一本书，内容浅显易懂，适合有一定linux操作系统基础和硬件基础的嵌入式初学者学习，内容很全面，busybox是有的，是一本内容全面的嵌入式linux入门书籍，知识点大多点到为止，不过深叙述。虽然是基础教程，里边有好些地方我没看懂，有必要之后重读。

-----  
书的内容不错，详细，翻译扣一星。建议找到英文版对照阅读。

-----  
[嵌入式Linux基础教程（第2版）\\_下载链接1](#)

## 书评

A great book for newbies who want to pick up embedded linux development quickly. This book offers an excellent big picture and enough details about embedded linux. The knowledge in this area is so wide and the author nicely chains all necessary topics together...

-----  
第一版出来时，我就找来看了。很快就看完整本了，后来过了一段时间恰好遇到出版社有别人译好的稿子，我就帮着看了一下，但之后就没消息提到出版的事。再过段时间就出来了英文第二版。  
好书，是说配得上它的名字。概念讲得很透彻。适合边动手边看书，再要升级就要自己动手搭个...

-----  
什么亚马逊全五星评价，扯淡。  
我来一看，满共就两个人评价，一看就是托。书超烂，且翻译的绝对垃圾。

嵌入式Linux基础教程（第2版）\_下载链接1\_