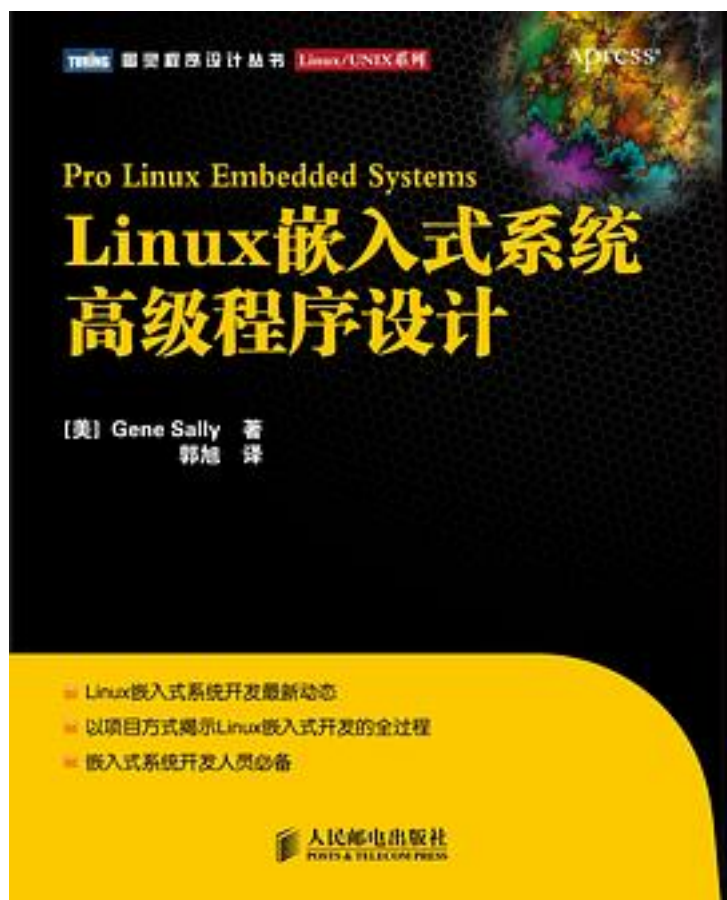


Linux嵌入式系统高级程序设计



[Linux嵌入式系统高级程序设计_下载链接1](#)

著者:Gene Sally

出版者:人民邮电出版社

出版时间:2010-11

装帧:平装

isbn:9787115239372

内 容 提 要

本书共18章，内容包括如何开发嵌入式Linux系统、移植Linux及其最佳实践。深入浅出地剖析了嵌入式Linux项目，讲述了如何创建嵌入式Linux

开发环境、配置和联编嵌入式Linux内核，为嵌入式系统配置和联编开源项目，最小化资源利用和启动时间的方法，以及联编项目的可用开源资源。

本书面向理解基本软件开发理念的嵌入式系统开发人员。

作者介绍:

Gene Sally

从十几岁就开始接触计算机，有着近10年的嵌入式项目开发经验。利用Linux开发过会计、保险处理、社会福利系统以及电信管理软件的测试软件。曾任职于一家专门开发嵌入式Linux工具和发布版的公司，担任过工程师、培训师、技术支持等职务。

目录: 目录

第1章 嵌入式Linux简介 1

1.1 为何使用嵌入式Linux 2

1.1.1 使用嵌入式Linux的技术原因 2

1.1.2 商业上使用嵌入式Linux的原因 7

1.2 1万英尺高空鸟瞰：略述嵌入式Linux开发 9

1.2.1 目标硬件 9

1.2.2 获取Linux 9

1.2.3 启动Linux 9

1.2.4 开发环境 10

1.2.5 系统设计 10

1.3 嵌入式Linux系统的组织结构 10

1.3.1 启动装载程序 11

1.3.2 内核 12

1.3.3 根文件系统 12

1.3.4 应用程序 13

1.3.5 交叉编译器 13

1.3.6 工具 14

1.4 从何处获得帮助 18

1.4.1 Google大学 18

1.4.2 邮件列表和新闻组 18

1.4.3 厂商赞助的资源 19

1.4.4 行业组织和社区兴趣网站 19

1.4.5 IRC 21

1.5 前瞻 21

第2章 配置软件环境 22

2.1 宿主机环境 22

2.1.1 Linux 23

2.1.2 Windows 26

2.2 宿主机服务 36

2.2.1 关闭防火墙 36

2.2.2 TFTP 37

2.2.3 DHCP 38

2.2.4 NFS 39

2.2.5 PXE 42

2.3 连接线缆 43

2.3.1 串行连接（用于控制台） 43

2.3.2 网络 43

2.4 避免IT管理人员恼火的拜访 44

第3章 目标机仿真与虚拟机	46
3.1 为何仿真目标机	46
3.2 通过QEMU进行仿真	47
3.2.1 编译QEMU	47
3.2.2 使用QEMU仿真目标机	48
3.2.3 使用QEMU在仿真模式下进行编译	50
3.3 x86宿主机的虚拟化软件	51
3.4 小结	52
第4章 开始你的项目	53
4.1 大多数目标板包含了Linux发布版	54
4.2 打开目标板包装后需要做什么	55
4.2.1 有Linux吗？启动它	55
4.2.2 访问内核	61
4.2.3 理解根文件系统	64
4.3 根文件系统对项目的适用性	68
4.4 小结	70
第5章 获取对应于目标板的Linux	71
5.1 从目标板厂商获取Linux	72
5.1.1 应该询问板卡厂商的问题	72
5.1.2 现在你是一位顾客了	74
5.2 开源嵌入式Linux发布版	74
5.2.1 嵌入式Linux发布版构建工具存在的原因	75
5.2.2 应该使用某种发布版构建工具吗	75
5.2.3 流行的开源嵌入式Linux发布版	76
5.3 从商业厂商和咨询机构获取Linux	87
5.3.1 你需要考虑商业性的Linux厂商吗	87
5.3.2 预期厂商应提供什么	88
5.3.3 厂商列表	88
5.4 小结	90
第6章 从头开始创建Linux发布版	91
6.1 交叉编译器基础	92
6.1.1 联编软件时需要注意的情况	93
6.1.2 习惯命令行	94
6.2 联编GCC交叉编译器概述	94
6.2.1 C库	95
6.2.2 收集源代码	95
6.2.3 联编GCC	99
6.3 用crosstool-NG联编工具链	109
6.4 创建根文件系统	111
6.4.1 配置环境	111
6.4.2 联编和安装BusyBox	111
6.4.3 库	112
6.4.4 创建设备结点和目录	113
6.4.5 最后修整	113
6.4.6 联编内核	114
6.4.7 解决启动问题	117
6.5 发布发布版	118
6.6 小结	119
第7章 启动目标板	120
7.1 启动Linux系统是一部三幕剧	120
7.1.1 启动装载程序	120
7.1.2 内核层与用户层	123
7.2 启动装载程序	123
7.3 内核启动	129

- 7.3.1 内核入口点 130
- 7.3.2 用户层启动 135
- 7.3.3 BusyBox Init 138
- 7.3.4 用户自定义init 139
- 7.4 前瞻 140
- 第8章 配置应用开发环境 141
- 8.1 选择完成工作的正确工具 141
- 8.2 了解要开发的应用程序 141
- 8.3 使用何种开发工具 143
 - 8.3.1 C语言 144
 - 8.3.2 C++ 144
 - 8.3.3 Java 145
- 8.4 非传统嵌入式语言 146
 - 8.4.1 Python 147
 - 8.4.2 TCL 148
 - 8.4.3 Shell脚本 149
 - 8.4.4 PHP 150
- 8.5 性能和性能剖析工具 151
 - 8.5.1 性能剖析 151
 - 8.5.2 内存泄漏检测 154
 - 8.5.3 静态分析 156
- 8.6 IDE 157
 - 8.6.1 编辑器 + make + shell 157
 - 8.6.2 Eclipse 159
- 8.7 前瞻 163
- 第9章 应用开发 164
- 9.1 开始开发应用程序 164
- 9.2 桌面系统和目标机 164
 - 9.2.1 针对可移植性编写代码 165
 - 9.2.2 系统差别 166
- 9.3 Hello World 167
 - 9.3.1 获取工具 167
 - 9.3.2 使make工作 168
 - 9.3.3 在目标机上运行代码 171
 - 9.3.4 更复杂的项目 172
- 9.4 准备好调试 176
- 9.5 前瞻 178
- 第10章 调试应用程序 179
- 10.1 开始开发应用程序 179
- 10.2 调试的种类 179
- 10.3 远程调试概述 180
- 10.4 调试C和C++ 180
 - 10.4.1 联编GDB 181
 - 10.4.2 GDB前端 182
 - 10.4.3 针对调试进行编译 182
- 10.5 调试Java 190
- 10.6 测量 193
 - 10.6.1 Java测量 195
 - 10.6.2 脚本语言中的测量 196
- 10.7 前瞻 196
- 第11章 内核配置和开发 197
- 11.1 内核项目布局 197
- 11.2 联编内核 200
 - 11.2.1 内核配置程序的工作方式 202

- 11.2.2 默认配置 203
- 11.2.3 手工编辑.config文件 204
- 11.2.4 联编内核 205
- 11.2.5 联编模块 207
- 11.2.6 清理 208
- 11.3 开源社区 209
 - 11.3.1 内核开发过程 209
 - 11.3.2 向Linux内核贡献代码 209
 - 11.3.3 应用补丁 211
- 11.4 前瞻 211
- 第12章 实时 212
 - 12.1 Linux中的实时实现 215
 - 12.2 实时程序设计惯例 218
 - 12.2.1 仅一个实时进程 218
 - 12.2.2 锁定内存 218
 - 12.2.3 避免使用堆 219
 - 12.2.4 需要继承优先级的互斥量 219
 - 12.2.5 I/O是非确定的 220
 - 12.2.6 使用线程池 220
 - 12.2.7 LatencyTOP 221
 - 12.3 常见的硬件缺陷 223
 - 12.3.1 系统管理中断 224
 - 12.3.2 VGA控制台 224
 - 12.3.3 DMA总线独占 224
 - 12.4 小结 224
- 第13章 使用开源软件项目 225
 - 13.1 使用开源软件包 225
 - 13.1.1 开源项目的结构 226
 - 13.1.2 项目团队并非你的员工 226
 - 13.1.3 理解许可证 227
 - 13.1.4 下载 228
 - 13.1.5 使用源代码控制系统取得代码 229
 - 13.1.6 交叉编译 232
 - 13.1.7 使用configure 233
 - 13.1.8 联编和安装 237
 - 13.2 常用项目 238
 - 13.2.1 DirectFB 238
 - 13.2.2 Dropbear 238
 - 13.2.3 QT/Qttopia 238
 - 13.2.4 JamVM 238
 - 13.2.5 Rysz 239
 - 13.2.6 Netcat 239
 - 13.2.7 TinyXML 239
 - 13.2.8 Micro_httpd 239
 - 13.2.9 Stupid-FTPd 240
 - 13.2.10 Quagga 240
 - 13.2.11 tslib 240
 - 13.2.12 fgetty 240
- 第14章 BusyBox 241
 - 14.1 基于BusyBox的系统的组织方式 241
 - 14.2 构建基于BusyBox的系统 242
 - 14.2.1 下载软件 242
 - 14.2.2 配置 243
 - 14.3 BusyBox为什么这样小 248

- 14.4 创建自己的小应用程序 249
- 14.5 获得帮助 253
- 14.6 前瞻 253
- 第15章 系统设计 254
 - 15.1 整体图景 254
 - 15.2 配置启动装载程序和内核 255
 - 15.2.1 U-Boot 255
 - 15.2.2 其他启动装载程序 257
 - 15.2.3 就地执行 257
 - 15.3 选择根文件系统 258
 - 15.3.1 基于块的文件系统 258
 - 15.3.2 MTD文件系统 261
 - 15.3.3 基于内存缓冲区的文件系统 262
 - 15.3.4 文件系统的组合 263
 - 15.4 组装根文件系统 263
 - 15.4.1 创建中间整备区域 264
 - 15.4.2 创建目录框架 264
 - 15.4.3 收集库和其他必需的文件 264
 - 15.4.4 创建初始化脚本 266
 - 15.4.5 设置所有权和权限 269
 - 15.5 安全 270
 - 15.5.1 内建的安全机制 271
 - 15.5.2 SELinux 271
 - 15.5.3 PAM 274
 - 15.6 前瞻 276
- 第16章 系统微调 277
 - 16.1 减小根文件系统的大小 279
 - 16.1.1 从零开始 279
 - 16.1.2 为节省空间而编译 281
 - 16.2 减小内核的大小 284
 - 16.3 最小化启动时间 289
 - 16.3.1 减少内核启动时间 289
 - 16.3.2 测量内核启动时间 291
 - 16.3.3 缩减根文件系统的启动时间 294
 - 16.4 前瞻 298
- 第17章 部署应用程序 299
 - 17.1 嵌入式设备的部署 299
 - 17.1.1 需求 300
 - 17.1.2 工业设计 300
 - 17.1.3 机械设计 301
 - 17.1.4 电气工程 302
 - 17.1.5 制造工程 302
 - 17.1.6 软件设计 303
 - 17.1.7 软件工程 303
 - 17.1.8 制造 304
 - 17.2 部署策略和战术 305
 - 17.3 启动装载程序 306
 - 17.3.1 一般概念 306
 - 17.3.2 UBOOT: 配置初始参数 307
 - 17.3.3 expect 308
 - 17.3.4 启动装载程序只是程序 310
 - 17.4 部署根文件系统 312
 - 17.4.1 应用程序文件和库 312
 - 17.4.2 在工厂进行的第一次现场更新 314

17.5 前瞻	314
第18章 处理现场更新	315
18.1 根文件系统更新	315
18.1.1 基本策略	315
18.1.2 完全更新	316
18.1.3 并行系统	319
18.1.4 自己动手	320
18.1.5 使用包管理器	321
18.1.6 initramfs根文件系统	330
18.2 内核更新	331
18.2.1 基本策略	331
18.2.2 模块	332
18.2.3 完全更新	334
18.3 现场更新故障	334
18.3.1 报告失败，停下	335
18.3.2 故障安全的根文件系统	335
18.3.3 故障安全的内核	335
18.4 综述	336
• • • • •	(收起)

[Linux嵌入式系统高级程序设计 下载链接1](#)

标签

嵌入式

linux

Linux

高级程序设计

计算机

嵌入式系统

计算机-编程

编程

评论

这本书有点挂羊头卖狗肉，我之所以买这本书，就是因为书名“高级程序设计”吸引了我，但是该书通篇所讲的却是如何搭建嵌入式开发环境，其实原作者的“Pro Linux Embedded Systems”挺切合书的内容的，只不过翻译者有点哗众取宠，搞了个故弄玄虚的书名。其实从构建嵌入式环境的角度而言，这本书确实不如O'RELLY的那本书，还是建议买那本书，毕竟O'RELLY的品牌还是在哪儿摆着的！

没啥高级，没有友善之臂的开发板讲的好。

[Linux嵌入式系统高级程序设计_下载链接1](#)

书评

[Linux嵌入式系统高级程序设计_下载链接1](#)