

# Linux

## 与嵌入式系统 (第2版)

李善平 刘文峰 王焕龙 等编著



清华大学出版社

# Linux 与嵌入式系统

## (第2版)

李善平 刘文峰 王焕龙 等编著

清华大学出版社

北 京

## 内 容 简 介

Linux 是个优秀的操作系统,嵌入式系统则是近年来的热门话题。本书既是作者在运用 Linux 研究和开发嵌入式系统的过程中资料与经验的总结,也是把二者结合起来的集中体现。书中有对主流嵌入式操作系统、Linux 操作系统、基于 Linux 的嵌入式操作系统的综述,更主要的是系统介绍了利用 Linux 开发嵌入式系统的关键技术和方法。同时,本书精选了作者研发的几个嵌入式系统的案例(机顶盒、宽带路由器、基于 Linux 的家用网关的设计与实现等)奉献给大家,希望能对读者有所帮助。

本书内容丰富、层次清晰、力求较强的实践性和应用性,可作为高等学校有关嵌入式系统教学的教材,也可作为嵌入式系统开发与应用的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

Linux 与嵌入式系统/李善平,刘文峰,王焕龙等编著. —2 版. —北京:清华大学出版社,2006.3

ISBN 7-302-12486-8

I. L… II. ①李… ②刘… ③王… III. Linux 操作系统-程序设计 IV. TP316.89

中国版本图书馆 CIP 数据核字(2006)第 006260 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客户服务:010-62776969

组稿编辑:钟志芳

文稿编辑:李虎斌

封面设计:范华明

版式设计:杨 洋

印 刷 者:北京国马印刷厂

装 订 者:三河市春园印刷有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:27.5 字数:610 千字

版 次:2006 年 3 月第 2 版 2006 年 3 月第 1 次印刷

书 号:ISBN 7-302-12486-8/TP·8005

印 数:1~5000

定 价:38.00 元

## 第 2 版说明

《Linux 与嵌入式系统》一书于 2003 出版后，已多次印刷。这期间我们收到许多全国各地读者的电话和电子邮件，他们对本书给予了充分的肯定，并提出了非常中肯的意见和建议。根据这些反馈意见，结合我们的项目研发实践，总结我们多年积累的教学经验，在第 1 版的基础上，我们对本书进行了内容的更新和补充。

与第 1 版相比，第 2 版更新和补充了许多新的内容，而且结构的安排更加合理。主要体现在：

(1) 精简了第 1 版中部分繁杂的理论介绍，更新了部分过时的内容，合并了较分散的层次结构，使本书更具系统性。

(2) 增加了作者研发的几个嵌入式系统案例，如机顶盒、宽带路由器等的设计开发案例，增强了本书的实用性和可操作性，对广大嵌入式开发人员及高等学校相关专业的学生、教师等提供了很有价值的参考。这些内容是我们多年来在 Linux 嵌入式系统的研究总结，其中部分研究成果在实际项目中得到了应用。

嵌入式系统技术是目前电子产品设计领域最为热门的技术之一，目前广泛地应用于网络通信、信息家电、工业控制等各个领域。Linux 是一个内核源代码开放、具备一整套工具链、有强大的网络支持及成本低廉的优秀操作系统，因此嵌入式 Linux 自诞生起就秉承了这众多独特优势，使它正在并越来越多地受到人们的关注。本书既是我们在运用 Linux 研究和开发嵌入式系统的过程中资料与经验的总结，也是把二者结合起来的集中体现。我们由衷地希望本书能对高等学校相关专业的教师和学生、从事嵌入式设计与研发的技术人员有所帮助。

本书第 2 版的修订工作由全体工作者共同完成，增加的内容主要体现在第 9 章到第 13 章。其中，第 9 章主要讨论嵌入式 Linux 系统的网络互联及其优化技术。第 10 章主要介绍嵌入式系统集成开发环境，重点讲解 JTAG 在线调试技术的实现。第 11 章结合一个实际产品——IPTV 机顶盒的设计开发过程，简单综述了整个嵌入式系统的开发。第 12 章介绍如何设计与实现宽带路由器的嵌入式系统，以及在此基础上的网络地址转换和路由等功能。第 13 章详细介绍家用网关的搭建过程。

总之，经过此次修订，本书的实用性更强，系统性更强，有助于读者的学习。但由于作者水平有限，书中可能还存在疏漏与不足，还望各位读者不吝赐教。

作 者

于浙江大学计算机学院

2006 年 1 月

# 前 言

嵌入式系统具有巨大的市场需求前景，可广泛应用于移动计算设备、网络设备、信息电器、工控设备、车载设备、娱乐设施、仪器仪表等场合；开发和应用嵌入式操作系统的技术难度远大于普通的计算机应用系统。就这两条，足以构成极大的诱惑，至少对我们是这样。

Linux 本身就是一个优秀的操作系统，再加上它的源代码是开放的（人人可以研读），而开放的源代码做得相当漂亮，所以就对它产生了浓厚的兴趣。我们读了几年，又把它介绍给浙江大学计算机学院的历届本科生和研究生。我们利用 Linux 做了很多操作系统的研究工作，开发了很多应用系统。

当需要寻找一个操作系统作为嵌入式系统的核心原型时，我们自然想到了 Linux。于是有了逐年递增的经验积累，有了相关关键技术的总结和提高，于是也就有了本书的出版。

翻开书，您将看到我们对主流嵌入式操作系统、Linux 操作系统，以及基于 Linux 的嵌入式操作系统的综述。更主要的是，全书用大部分篇幅，系统介绍了利用 Linux 开发嵌入式系统的关键技术和方法。同时，选取了几个作者研发的嵌入式系统的案例。目的只有一个：希望对您有帮助，或者希望您少走弯路。

全书分 13 章。第 1 章和第 2 章综述嵌入式系统和 Linux 操作系统，介绍典型的嵌入式系统，典型的基于 Linux 的嵌入式系统，由李善平、高庆、马天驰、刘文峰、王焕龙、王伟波、陈鲁川、李程远、解超等执笔，虞云翔改编。第 3 章描述嵌入式系统的开发过程，由刘文峰执笔。第 4 章讨论嵌入式系统的几个特殊的开发技术，由李程远、王焕龙、谢科先执笔。第 5 章至第 9 章系统论述了嵌入式操作系统的进程管理、存储管理、文件系统管理、GUI、网络互联，分别由李程远、王焕龙、王伟波、解超、刘文峰、陈鲁川执笔。第 9 章的嵌入式 Linux 系统的网络互联及其优化技术，来自于王少平、虞云翔的贡献。第 10 章的嵌入式系统集成开发环境，来自于王伟波贡献，由王伟波、赖宗濂执笔。第 11 章的嵌入式系统案例——机顶盒，由王焕龙、李天佑执笔。第 12 章的嵌入式系统案例——宽带路由器，来自于解超、王明伟的贡献，由解超、王明伟、虞云翔执笔。第 13 章的嵌入式系统案例——基于 Linux 的家用网关，由周强执笔。我们全体作者共同完成书稿的校对和修改。

本书的出版，得到了清大金地公司的钟志芳老师的大力帮助。钟老师的敬业，给我们留下了极深的印象，在此表示感谢。

李善平

于浙江大学计算机学院

2005 年 10 月

# 目 录

|                                        |    |
|----------------------------------------|----|
| 第 1 章 嵌入式系统 .....                      | 1  |
| 1.1 关于嵌入式系统 .....                      | 1  |
| 1.1.1 嵌入式系统概念 .....                    | 1  |
| 1.1.2 嵌入式处理器 .....                     | 2  |
| 1.2 关于嵌入式操作系统 .....                    | 4  |
| 1.2.1 从嵌入式系统到嵌入式操作系统 .....             | 4  |
| 1.2.2 典型的嵌入式操作系统 .....                 | 5  |
| 1.3 嵌入式操作系统的应用 .....                   | 7  |
| 1.4 典型嵌入式系统 .....                      | 9  |
| 1.4.1 VxWorks 和 pSOS .....             | 9  |
| 1.4.2 OSKit .....                      | 16 |
| 1.4.3 Palm OS .....                    | 21 |
| 1.4.4 $\mu$ C/OS 和 $\mu$ C/OS-II ..... | 24 |
| 1.4.5 QNX 实时操作系统 .....                 | 28 |
| 第 2 章 Linux .....                      | 33 |
| 2.1 Linux 简介 .....                     | 33 |
| 2.1.1 Linux 的历史 .....                  | 33 |
| 2.1.2 Linux 的发行版本 .....                | 34 |
| 2.1.3 Linux 的特点 .....                  | 36 |
| 2.2 Linux 内核 .....                     | 37 |
| 2.2.1 内核概况 .....                       | 37 |
| 2.2.2 进程管理 .....                       | 39 |
| 2.2.3 存储管理 .....                       | 47 |
| 2.2.4 文件系统管理 .....                     | 58 |
| 2.2.5 设备管理 .....                       | 68 |
| 2.2.6 系统启动 .....                       | 73 |
| 2.3 Linux 应用 .....                     | 75 |
| 2.3.1 Linux 在大型服务器上的应用 .....           | 75 |
| 2.3.2 Linux 在桌面端的应用 .....              | 76 |
| 2.3.3 Linux 在嵌入式环境中的应用 .....           | 77 |
| 2.4 主流嵌入式 Linux 系统 .....               | 78 |

|              |                                |            |
|--------------|--------------------------------|------------|
| 2.4.1        | $\mu$ Clinux .....             | 78         |
| 2.4.2        | RTLinux 和 RTAI .....           | 80         |
| 2.4.3        | MontaVista Linux .....         | 84         |
| 2.4.4        | LynxOS .....                   | 87         |
| <b>第 3 章</b> | <b>嵌入式系统开发过程 .....</b>         | <b>90</b>  |
| 3.1          | 需求分析和概要设计 .....                | 90         |
| 3.1.1        | 嵌入式系统的概念和组成 .....              | 90         |
| 3.1.2        | 系统需求分析 .....                   | 90         |
| 3.1.3        | 系统结构模型建立方式 .....               | 91         |
| 3.1.4        | 系统结构模型实现流程 .....               | 92         |
| 3.2          | 硬件 .....                       | 93         |
| 3.2.1        | 嵌入式处理器 .....                   | 93         |
| 3.2.2        | 存储设备 .....                     | 94         |
| 3.2.3        | 外围设备 .....                     | 95         |
| 3.2.4        | 典型的嵌入式系统开发平台 .....             | 97         |
| 3.2.5        | Lineo 的 $\mu$ CSimm .....      | 98         |
| 3.2.6        | ARM 的 ARM Evaluator-7T .....   | 102        |
| 3.3          | 软件 .....                       | 106        |
| 3.3.1        | 嵌入式系统的软件组成 .....               | 106        |
| 3.3.2        | 嵌入式软件的基本开发流程 .....             | 109        |
| 3.3.3        | 嵌入式操作系统 .....                  | 110        |
| 3.4          | 嵌入式系统软件开发过程 .....              | 111        |
| 3.4.1        | 开发 .....                       | 111        |
| 3.4.2        | 编译和连接 .....                    | 112        |
| 3.4.3        | 重定位和下载 .....                   | 113        |
| 3.4.4        | 调试 .....                       | 113        |
| <b>第 4 章</b> | <b>嵌入式 Linux 系统的开发技术 .....</b> | <b>116</b> |
| 4.1          | 快速启动 .....                     | 116        |
| 4.2          | 开机画面 .....                     | 121        |
| 4.2.1        | 隐藏字符信息 .....                   | 121        |
| 4.2.2        | 显示用户定义的画面 .....                | 122        |
| 4.3          | ramdisk 技术 .....               | 123        |
| 4.4          | 系统小型化 .....                    | 125        |
| 4.4.1        | 内核配置 .....                     | 125        |
| 4.4.2        | 嵌入式 C 库 .....                  | 127        |
| 4.4.3        | 小型 shell .....                 | 130        |

|                                     |            |
|-------------------------------------|------------|
| 4.5 系统移植.....                       | 132        |
| 4.5.1 建立交叉编译环境.....                 | 132        |
| 4.5.2 内核移植.....                     | 136        |
| 4.5.3 bootloader.....               | 139        |
| <b>第 5 章 嵌入式 Linux 系统的进程.....</b>   | <b>141</b> |
| 5.1 Linux 在实时方面的缺陷.....             | 141        |
| 5.2 Linux 实时化的关键问题.....             | 143        |
| 5.3 Linux 实时化改造的典型方法.....           | 144        |
| 5.3.1 外部实时性扩展.....                  | 144        |
| 5.3.2 内部实时性改造.....                  | 160        |
| 5.4 实时应用程序的编写.....                  | 164        |
| 5.5 实时程序的测试.....                    | 167        |
| <b>第 6 章 嵌入式 Linux 系统的存储管理.....</b> | <b>169</b> |
| 6.1 缺少 MMU 支持的内存管理.....             | 169        |
| 6.1.1 3 种内存管理模型.....                | 170        |
| 6.1.2 标准 Linux 的内存管理.....           | 170        |
| 6.1.3 $\mu$ Clinux 的内存管理.....       | 171        |
| 6.1.4 $\mu$ Clinux 内存管理的局限性.....    | 172        |
| 6.2 $\mu$ Clinux 内存管理的实现.....       | 173        |
| 6.2.1 内存管理数据结构.....                 | 173        |
| 6.2.2 物理空间管理.....                   | 174        |
| 6.2.3 内核内存的申请和释放.....               | 175        |
| 6.2.4 内存映射.....                     | 176        |
| 6.3 内存管理模块的启动初始化.....               | 178        |
| 6.3.1 setup_arch().....             | 178        |
| 6.3.2 paing_init().....             | 179        |
| 6.3.3 free_area_init().....         | 180        |
| 6.3.4 mem_init().....               | 180        |
| 6.4 可执行程序的加载.....                   | 180        |
| 6.4.1 用户程序的内存分布.....                | 180        |
| 6.4.2 reloc 段机制.....                | 181        |
| 6.4.3 flat 可执行文件格式.....             | 182        |
| 6.4.4 执行文件加载流程.....                 | 183        |
| <b>第 7 章 嵌入式 Linux 系统的文件系统.....</b> | <b>187</b> |
| 7.1 嵌入式系统的存储.....                   | 187        |

|       |                                            |     |
|-------|--------------------------------------------|-----|
| 7.2   | 文件系统类型.....                                | 188 |
| 7.2.1 | cramfs.....                                | 188 |
| 7.2.2 | romfs.....                                 | 190 |
| 7.3   | 日志文件系统.....                                | 193 |
| 7.3.1 | 文件系统完整性.....                               | 193 |
| 7.3.2 | 日志文件系统.....                                | 194 |
| 7.3.3 | jffs 与 jffs2 文件系统.....                     | 195 |
| 7.3.4 | ReiserFS 文件系统.....                         | 197 |
| 7.3.5 | ext3 文件系统.....                             | 198 |
| 7.3.6 | xfs 文件系统.....                              | 199 |
| 7.4   | 文件系统的目录结构.....                             | 200 |
| 7.4.1 | /etc 目录.....                               | 201 |
| 7.4.2 | /bin 目录.....                               | 204 |
| 7.4.3 | /sbin 目录.....                              | 205 |
| 7.4.4 | /dev 目录.....                               | 206 |
| 7.5   | 小结.....                                    | 207 |
| 第 8 章 | 嵌入式 Linux 的 GUI.....                       | 208 |
| 8.1   | 嵌入式 GUI.....                               | 208 |
| 8.1.1 | 嵌入式 GUI 的概念.....                           | 208 |
| 8.1.2 | 一些成熟的嵌入式 GUI.....                          | 209 |
| 8.2   | 典型的嵌入式 GUI—Microwindows.....               | 210 |
| 8.2.1 | Microwindows 的设备驱动.....                    | 211 |
| 8.2.2 | 与设备无关的图形引擎.....                            | 214 |
| 8.2.3 | Nano-X 的窗口管理和 API.....                     | 218 |
| 8.2.4 | Microwindows 在 Linux Frame Buffer 上运行..... | 229 |
| 8.3   | 其他知名嵌入式 GUI.....                           | 230 |
| 8.3.1 | 面向实时的 MiniGUI.....                         | 230 |
| 8.3.2 | 应用广泛的 Qt/Embedded.....                     | 232 |
| 8.4   | 嵌入式 GUI 的应用.....                           | 233 |
| 8.4.1 | 嵌入式 GUI 上的其他软件.....                        | 233 |
| 8.4.2 | 嵌入式 GUI 的应用.....                           | 236 |
| 8.5   | 小结.....                                    | 237 |
| 第 9 章 | 嵌入式 Linux 系统的网络互联及其优化技术.....               | 238 |
| 9.1   | 嵌入式 Internet 及其应用.....                     | 238 |
| 9.1.1 | 嵌入式 Internet 技术的兴起.....                    | 238 |
| 9.1.2 | 嵌入式 Internet 技术.....                       | 240 |

|                                         |            |
|-----------------------------------------|------------|
| 9.2 网络协议栈的定制.....                       | 243        |
| 9.2.1 Linux 的网络代码分布及特点.....             | 243        |
| 9.2.2 网络部分内核编译配置.....                   | 245        |
| 9.2.3 TCP/IP 协议的定制.....                 | 247        |
| 9.3 Linux 中 TCP/IP 协议应用优化技术.....        | 253        |
| 9.3.1 实现快速上下文切换的设想.....                 | 253        |
| 9.3.2 网络服务器.....                        | 255        |
| 9.4 Linux IP 报文转发及防火墙实现分析.....          | 260        |
| 9.4.1 IP 数据报文流.....                     | 260        |
| 9.4.2 netfilter 框架.....                 | 263        |
| 9.4.3 netfilter 的网络地址转换 (NAT) 的实现.....  | 267        |
| 9.4.4 IPTables: netfilter 体系的规则载体.....  | 271        |
| 9.5 基于 Linux 的家庭网关及其优化技术.....           | 273        |
| 9.5.1 改变网络设备的工作模式.....                  | 274        |
| 9.5.2 转发报文不再往上经由 IP 层.....              | 277        |
| 9.5.3 NAT 的重写.....                      | 280        |
| <b>第 10 章 嵌入式系统集成开发环境.....</b>          | <b>286</b> |
| 10.1 综述.....                            | 287        |
| 10.1.1 集成开发环境.....                      | 287        |
| 10.1.2 嵌入式软件开发.....                     | 288        |
| 10.1.3 Python 开发环境.....                 | 291        |
| 10.2 代码编辑模块.....                        | 292        |
| 10.2.1 代码编辑功能.....                      | 292        |
| 10.2.2 PygtkScintilla.....              | 293        |
| 10.2.3 EideP 代码编辑模块的实现.....             | 294        |
| 10.3 软件调试模块.....                        | 296        |
| 10.3.1 软件调试功能简介.....                    | 296        |
| 10.3.2 gdb 前端 (front-end) 软件分析.....     | 297        |
| 10.3.3 gdb 图形前端的实现.....                 | 298        |
| 10.3.4 输出信息提取的实现.....                   | 300        |
| 10.3.5 功能实现.....                        | 301        |
| 10.4 JTAG 在线调试的实现.....                  | 302        |
| 10.4.1 标准的 JTAG 协议.....                 | 303        |
| 10.4.2 PXA255 中 JTAG 调试接口协议.....        | 305        |
| 10.4.3 PXA255 中 debug 相关的寄存器及其访问方式..... | 307        |
| 10.4.4 PXA255 中 debug 相关的 JTAG 命令.....  | 312        |

|               |                             |            |
|---------------|-----------------------------|------------|
| 10.4.5        | JTAG 在线调试实现思想 .....         | 314        |
| 10.4.6        | 底层 JTAG 操作模块 .....          | 315        |
| 10.4.7        | host debugger .....         | 318        |
| 10.4.8        | Pygdbserver .....           | 326        |
| 10.4.9        | debug handler .....         | 329        |
| 10.4.10       | 用 JTAG 调试的例子 .....          | 333        |
| 10.5          | 小结 .....                    | 335        |
| <b>第 11 章</b> | <b>嵌入式系统案例——机顶盒 .....</b>   | <b>336</b> |
| 11.1          | 机顶盒需求分析 .....               | 336        |
| 11.1.1        | 机顶盒基本概念 .....               | 336        |
| 11.1.2        | 所需功能 .....                  | 337        |
| 11.2          | 机顶盒概要设计 .....               | 338        |
| 11.2.1        | 建立需求库 .....                 | 338        |
| 11.2.2        | 进行需求映射 .....                | 339        |
| 11.3          | 机顶盒硬件设计 .....               | 340        |
| 11.4          | 机顶盒软件设计 .....               | 341        |
| 11.4.1        | GUI 设计 .....                | 343        |
| 11.4.2        | 系统裁减和压缩 .....               | 348        |
| 11.4.3        | 网络设计 .....                  | 355        |
| 11.4.4        | 系统启动优化 .....                | 358        |
| 11.5          | 小结 .....                    | 359        |
| <b>第 12 章</b> | <b>嵌入式系统案例——宽带路由器 .....</b> | <b>360</b> |
| 12.1          | 宽带路由器及 Nanos 操作系统 .....     | 360        |
| 12.1.1        | 典型宽带路由器的功能 .....            | 360        |
| 12.1.2        | Nanos 操作系统 .....            | 361        |
| 12.2          | 操作系统的设计 .....               | 361        |
| 12.2.1        | Nanos 操作系统的设计思想 .....       | 361        |
| 12.2.2        | 关键部分的设计思想 .....             | 362        |
| 12.3          | 任务管理的实现 .....               | 363        |
| 12.3.1        | 任务状态 .....                  | 363        |
| 12.3.2        | 任务控制块和就绪表 .....             | 364        |
| 12.3.3        | 任务的创建与删除 .....              | 366        |
| 12.3.4        | 任务等待和挂起 .....               | 367        |
| 12.3.5        | 任务调度 .....                  | 368        |
| 12.3.6        | 信号量 .....                   | 369        |
| 12.4          | 内存管理的实现 .....               | 371        |

|                                             |            |
|---------------------------------------------|------------|
| 12.4.1 内存寻址 .....                           | 371        |
| 12.4.2 伙伴系统算法 .....                         | 373        |
| 12.4.3 Cache 与 Slab 分配器 .....               | 375        |
| 12.5 中断与异常 .....                            | 378        |
| 12.6 Nanos 的 PCI 总线驱动和 8139 网卡驱动 .....      | 380        |
| 12.6.1 PCI 总线驱动 .....                       | 380        |
| 12.6.2 设计通用 PCI 设备驱动接口 .....                | 383        |
| 12.6.3 rtl8139 网卡驱动程序的实现 .....              | 384        |
| 12.7 操作系统的初始化 .....                         | 386        |
| 12.7.1 系统的引导 .....                          | 386        |
| 12.7.2 内存管理的初始化 .....                       | 387        |
| 12.7.3 其他部分的初始化 .....                       | 388        |
| 12.8 网卡的半轮询工作模式 .....                       | 389        |
| 12.9 网络地址转换——NAT .....                      | 391        |
| 12.9.1 模块的定义 .....                          | 391        |
| 12.9.2 NAT 模块 .....                         | 392        |
| 12.9.3 模块的实现 .....                          | 393        |
| 12.10 小结 .....                              | 406        |
| <b>第 13 章 嵌入式系统案例——基于 Linux 的家用网关 .....</b> | <b>407</b> |
| 13.1 系统概述 .....                             | 407        |
| 13.1.1 硬件设计 .....                           | 407        |
| 13.1.2 软件设计 .....                           | 408        |
| 13.1.3 准备工作 .....                           | 408        |
| 13.2 搭建基本系统 .....                           | 409        |
| 13.2.1 编译工具链 .....                          | 410        |
| 13.2.2 内核的编译 .....                          | 411        |
| 13.2.3 BusyBox 的编译和安装 .....                 | 412        |
| 13.2.4 系统的配置 .....                          | 413        |
| 13.2.5 内核的安装 .....                          | 414        |
| 13.3 创建网络连接 .....                           | 415        |
| 13.3.1 静态 IP 的设置 .....                      | 416        |
| 13.3.2 动态 IP 客户端 .....                      | 417        |
| 13.3.3 宽带拨号客户端 PPP 和 PPPoE .....            | 417        |
| 13.4 安装和配置网络服务 .....                        | 418        |
| 13.4.1 telnetd 服务 .....                     | 418        |
| 13.4.2 网页服务器 .....                          | 419        |

|        |                       |     |
|--------|-----------------------|-----|
| 13.4.3 | iptables 的安装 .....    | 420 |
| 13.4.4 | NAT、端口映射和防火墙的实现 ..... | 420 |
| 13.4.5 | 其他的网络服务 .....         | 422 |
| 13.5   | 文件系统的压缩 .....         | 422 |

# 第 1 章 嵌入式系统

## 1.1 关于嵌入式系统

随着数字信息技术和网络技术的高速发展,现在已经进入数码时代。建立在其基础上的嵌入式系统已经广泛地渗透到科学研究、工程设计、军事技术、商业文化艺术、娱乐业以及人们的日常生活等方方面面。随着国内外嵌入式产品(如车载计算机、机顶盒等)的进一步开发和推广,嵌入式技术越来越和人们的生活紧密结合。有人可能从来没有接触过计算机,但他不可能从来没有接触过嵌入式系统,因为嵌入式系统无处不在。从家里的洗衣机、电冰箱、作为交通工具的自行车、小汽车,到办公室里的远程会议系统等,这些都属于可以使用嵌入式技术进行开发和改造的产品。

### 1.1.1 嵌入式系统概念

#### 1. 嵌入式系统的定义

借用电气工程师协会(IEE)的一个定义(<http://www.iee.org/Policy/Areas/Y2K/w-43.cfm>):嵌入式系统是用来控制、监视或辅助设备、机器或工厂运行的装置。“嵌入式”一词表明嵌入式系统是以上大规模系统中不可或缺的一部分。

嵌入式系统具备下列特性:

- 通常只执行特定功能。这一点与一般桌上型办公设备或数据库系统有很大区别。
- 以微型计算机与周边器件构成核心,其规模可在很大范围内变化。如从 8051 芯片到 x86 芯片。
- 严格的时序和稳定性要求。这是因为在机器控制的大型系统中,程序运行稍有差错就可能使整个系统失去控制,甚至酿成灾害。
- 全自动操作循环。

嵌入式系统是计算机软件与硬件的综合体,它是以应用为中心,以计算机技术为基础,软硬件可裁剪,从而能够适应实际应用中对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。嵌入式计算机在应用数量上远远超过了各种通用计算机。一台通用计算机的外部设备中就包含 5~10 个嵌入式微处理器,键盘、鼠标、软驱、硬盘、显示卡、显示器、Modem、网卡、声卡、打印机、扫描仪、数码相机和 USB 集线器等均是由嵌入式处理器进行控制的。在制造工业、过程控制、通信、仪器、仪表、汽车、船舶、航空、航天、军事装备和消费类产品等方面,嵌入式计算机都有用武之地。

美国汽车大王福特公司的高级经理曾宣称,“福特出售的‘计算能力’已超过了 IBM”,由此可以想象嵌入式计算机工业的规模和广度。美国著名未来学家尼葛洛庞帝 1999 年 1 月访华时预言,4~5 年以后嵌入式智能(计算机)工具将是 PC 和互联网之后最伟大的发明。

## 2. 嵌入式系统的特点

嵌入式系统具有以下特点:

- 嵌入式系统是将先进的计算机技术、半导体技术以及电子技术与各个行业的具体应用相结合的产物,这一点就决定了它必然是一个技术密集、资金密集、不断创新的知识集成系统。通用计算机行业中,占整个计算机行业 90% 的个人计算机产业绝大部分采用的是 Intel 的 x86 体系结构,而芯片厂商则集中在 Intel、AMD、IBM 和 Motorola 等几家公司,操作系统方面更是被微软占居垄断地位。但这样的情况却不会在嵌入式系统领域出现。这是一个分散的、充满竞争、机遇与创新的工业,没有哪个公司的操作系统和处理器能够垄断市场。
- 嵌入式系统通常是面向用户、面向产品、面向特定应用的。嵌入式系统中的 CPU 与通用型 CPU 的最大不同就是前者大多工作在为特定用户群设计的系统中。通常,嵌入式系统 CPU 都具有低功耗、体积小、集成度高等特点,能够把通用 CPU 中许多由板卡完成的任务集成在芯片内部,从而有利于整个系统设计趋于小型化。在对嵌入式系统的硬件和软件进行设计时必须重视效率,去除冗余,针对用户的具体需求,对系统进行合理配置,才能达到理想性能。与此同时,系统设计还受市场供求关系的影响。嵌入式处理器的发展也体现出稳定性,一个体系一般要存在 8~10 年的时间。一个体系结构及相关的片上外设、开发工具、库函数、嵌入式应用产品合在一起,可以构成一套复杂的知识系统。
- 嵌入式系统和具体应用有机地结合在一起,其升级换代也是和具体产品同步进行的。因此嵌入式系统产品一旦进入市场,便具有较长的生命周期。
- 为了提高执行速度和系统可靠性,嵌入式系统中的软件一般都固化在存储器芯片或单片机中,而不是存储于磁盘等载体中。由于嵌入式系统的运算速度和存储容量仍然存在一定程度的限制,另外,由于大部分嵌入式系统必须具有较高的实时性,因此对程序的质量,特别是可靠性,有着较高的要求。
- 嵌入式系统本身并不具备在其上进行进一步开发的能力。在设计完成以后,用户如果需要修改其中的程序功能,也必须借助于一套开发工具和环境。
- 通用计算机的开发人员通常是计算机科学或者计算机工程方面的专业人士,而嵌入式系统开发人员却往往是各个应用领域中的专家,这就要求嵌入式系统所支持的开发工具易学、易用、可靠、高效。

### 1.1.2 嵌入式处理器

嵌入式系统的核心是各种类型的嵌入式处理器。据不完全统计,全世界嵌入式处理器的品种总量已经达到 1000 多种,流行体系结构有 30 多个系列,其中 8051 体系的占多数。

生产 8051 单片机的半导体厂家有 20 多个,开发了 300 多种衍生产品,仅 Philips 就有近 100 种。目前几乎每个半导体制造商都生产嵌入式处理器,越来越多的公司拥有自己的处理器设计部门。嵌入式微处理器的体系结构经历了从 CISC 到 RISC 和 Compact RISC 的转变;位数由 4 位、8 位、16 位、32 位到 64 位;寻址空间一般为 64KB~16MB,处理速度为 0.1MIPS~2000MIPS;常用的封装为 8~144 个引脚。根据其现状,嵌入式处理器可以分为嵌入式微处理器 (Embedded Microprocessor Unit, EMPU)、嵌入式微控制器 (Embedded Microcontroller Unit, EMCU)、嵌入式 DSP 处理器 (Embedded Digital Signal Processor, EDSP) 和嵌入式片上系统 (Embedded System on Chip, ESOC) 4 类。

嵌入式微处理器的基础是通用计算机中的 CPU,它一般装配在专门设计的电路板上,只保留与嵌入式应用有关的母板功能,但是电路板上必须包括 ROM、RAM、总线接口、各种外设等器件。嵌入式处理器目前主要有 Am186/88、PowerPC、MIPS 和 ARM 系列等。

嵌入式微控制器又称单片机,也就是在一块芯片中集成了整个计算机系统。嵌入式微控制器一般以某种微处理器内核作为核心,芯片内部集成 ROM/EPROM、EEPROM、Flash、RAM、总线、总线逻辑、定时/计数器、WatchDog、I/O 口、脉宽调制输出、A/D 和 D/A 等各种必要功能和外设。微控制器由于比微处理器体积小,功耗和成本低,可靠性高,因而是目前嵌入式工业的主流,品种和数量都很多。其中,比较具有代表性的通用系列有 8051、P51XA、MCS-251、MCS-96/196/296、MC68HC05/11/12/16 和 C166/167 等。另外还有许多半通用系列,如支持 USB 接口的 MCU 8XC930/931、C540 和 C541;支持 CAN-Bus、LCD 的众多专用 MCU 和兼容系列。

DSP 处理器对系统结构和指令进行了特殊设计,使其适合于执行 DSP 算法,编译效率较高,指令执行速度也快。DSP 应用正从在通用单片机中以普通指令实现 DSP 功能,发展到采用嵌入式 DSP 处理器。嵌入式 DSP 处理器的长处在于能够进行向量运算、指针线性寻址等运算量较大的数据处理。比较有代表性的产品是 Motorola 的 DSP56000 系列、Texas Instruments 的 TMS320 系列,以及 Philips 公司基于可重置嵌入式 DSP 结构制造的低成本、低功耗的 R. E. A. L DSP 处理器。

而所谓的片上系统 SOC 则是在一个硅片上实现一个更为复杂的系统。各种处理器内核将作为 SOC 设计公司的标准库,成为 VLSI 设计中一种标准的器件,用标准的 VHDL 语言描述,存储在器件库中。SOC 可以分为通用和专用两类。通用系列包括 Infineon (Siemens) 的 TriCore、Motorola 的 M-Core、某些 ARM 系列器件等。而专用的 SOC 专用于某个或者某类系统中,不为一般用户所知。比如 Philips 的 Smart XA,它将 XA 单片机内核和支持超过 2048 位复杂 RSA 算法的 CCU 单元制作在一块硅片上,形成一个可以加载 Java 或 C 语言的专用的片上系统。

嵌入式系统是个复杂的高技术系统。由于市场竞争规则要求产品能够很快上市,而在短期内很难掌握并开发出所有功能,因此需要易于掌握和使用的嵌入式系统开发工具平台予以辅助。一个好的嵌入式系统开发工具除了能够开发出处理器的全部功能以外,还应该对用户友好的界面。目前的开发工具平台主要分为以下几类:实时在线仿真系统 ICE (In-Circuit Emulator)、高级语言编译器 (Compiler Tools)、源程序模拟器 (Simulator) 和实时多任务操作系统 (Real Time multi-tasking Operation System, RTOS)。其中 RTOS

是嵌入式系统的软件开发平台，它的引入解决了随着嵌入式系统中软件比重不断上升、应用程序越来越大而带来的嵌入式软件开发标准化的难题。

## 1.2 关于嵌入式操作系统

### 1.2.1 从嵌入式系统到嵌入式操作系统

早期的嵌入式系统很多都不用操作系统，它们只是为了实现某些特定功能，使用一个简单的循环控制对外界的控制请求进行处理，不具备现代操作系统的基本特征（如进程管理、存储管理、设备管理、网络通信等）。不可否认，这对一些简单的系统而言是足够的。但是当系统越来越复杂，利用的范围越来越广泛时，缺少操作系统就成为了一个最大的缺点，因为每添加一项新功能都可能需要从头开始设计，否则只能增加开发成本和系统复杂度。

C 语言的出现使操作系统开发变得越来越简单，可以利用 C 语言很快地写出一个小型的、稳定的操作系统。众所周之，《C 程序设计语言（The C Programming Language）》的作者 Dennis M. Ritchie 和 Brian W. Kernighan 利用 C 语言写出了著名的 UNIX 操作系统，直接影响了这 30 年计算机业的发展。同时，C 语言的出现对开发嵌入式系统来说，在效率和速度上都提高了很多。

从 20 世纪 80 年代开始，出现了各种各样的商业用嵌入式操作系统。这些操作系统大部分都是为专有系统而开发，从而形成了目前多种形式的商用嵌入式操作系统百家争鸣的局面，如 VxWorks、pSOS、PalmOS 和 Windows CE 等。

现在，网络在人们生活中的应用越来越广泛，在嵌入式系统中使用网络系统也自然成为一项基本的要求。在嵌入式系统中实现网络协议栈，对日常生活中的需要有着广泛的意义。利用嵌入式系统中的网络功能，可以实现下面将要介绍的信息电器这一即将取代 PC、从而在后 PC 时代占据市场主体的商品。我们知道，如果在上面所说的那种采用循环控制的嵌入式系统中加入网络协议栈，其程序复杂度会呈指数级增长。相反，在嵌入式操作系统中增加网络协议模块要方便得多，并且还能方便各种网络应用程序在不同平台之间移植。

在未来的社会中，使用嵌入式系统的情形会越来越多，人们可以不接触计算机，但是不接触嵌入式系统的情况是不存在的。嵌入式系统可能存在于生活的各个角落：家可能就是通过一个嵌入式系统控制的中心，管理家里的所有家电，控制家庭和外界网络的连接，让人们的生活更为方便；出门旅行时，在饭店的门把手中安装有电子识别控制系统，利用嵌入式系统根据旅客手上的特征，判断什么时候应该给他开门；在坐车时，汽车计算机可以通过 GPS（全球卫星定位系统）来判断自己的具体位置，利用嵌入式智能系统判断应该走哪条路比较方便；如果人们一不小心生病住院，医院的设备仪器中都有嵌入式系统，用来选择和控制仪器，提高诊断、手术的准确性和方便程度。

嵌入式系统的应用领域日益扩大，提供的应用功能也越来越复杂，当初的控制程序被