

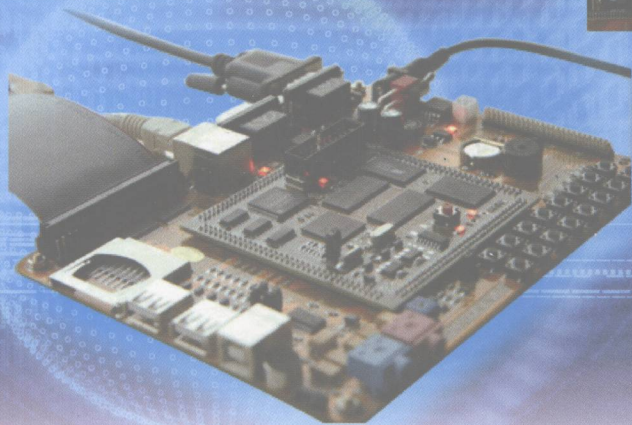
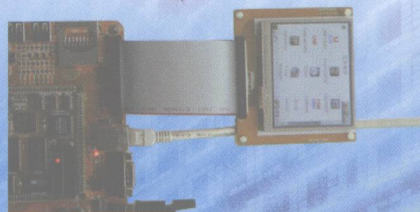
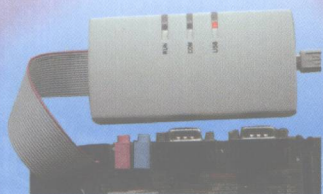
众多专家、厂商联合推荐 · 业界权威培训机构的经验总结

嵌入式Linux应用程序开发 标准教程 (第2版)

华清远见嵌入式培训中心 编著

提供36小时嵌入式专家讲座视频和教学课件

Embeded Linux Application Development



光盘内容

本书源代码
本书配套PPT
嵌入式专家讲座视频

TP316.8/
236=2

嵌入式与移动开发系列

NITE 国家信息技术紧缺人才培养工程
National Information Technology Education Project
国家信息技术紧缺人才培养工程系列丛书

嵌入式Linux应用程序开发 标准教程 (第2版)

华清远见嵌入式培训中心 编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

嵌入式Linux应用程序开发标准教程 / 华清远见嵌入式培训中心编著. —2版. —北京: 人民邮电出版社, 2009. 4
ISBN 978-7-115-19474-9

I. 嵌… II. 华… III. Linux操作系统—程序设计—教材 IV. TP316.89

中国版本图书馆CIP数据核字 (2009) 第001180号

内 容 提 要

本书主要分为 3 个部分, 包括 Linux 基础、搭建嵌入式 Linux 环境和嵌入式 Linux 的应用开发。Linux 基础部分从 Linux 基础、基本操作命令讲起, 为 Linux 初学者能快速入门提供了保证。接着系统地讲解了嵌入式 Linux 的环境搭建, 以及嵌入式 Linux 的 I/O 与文件系统的开发、进程控制开发、进程间通信开发、网络应用开发、基于中断的开发、设备驱动程序的开发以及嵌入式图形界面的开发等, 并且还安排了丰富的实验内容与课后实践, 使读者能够边学边用, 更快更好地掌握所学知识。

本书可作为高等院校电子类、电气类、控制类专业高年级本科生、研究生学习嵌入式 Linux 的教材, 也可供希望转入嵌入式领域的科研和工程技术人员参考使用, 还可作为嵌入式培训班的教材和参考书。

嵌入式 Linux 应用程序开发标准教程 (第 2 版)

- ◆ 编 著 华清远见嵌入式培训中心
责任编辑 屈艳莲
执行编辑 黄 焱
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京昌平百善印刷厂印刷
- ◆ 开本: 787×1092 1/16
印张: 25.75
字数: 654 千字
印数: 18 001—22 000 册

2009 年 4 月第 2 版

2009 年 4 月北京第 1 次印刷

ISBN 978-7-115-19474-9/TP

定价: 49.00 元 (附光盘)

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223

反盗版热线: (010) 67171154

本书编委会

主编

华清远见嵌入式培训中心

指导单位

工业和信息化部软件与集成电路促进中心

编委

孙加兴 李 滨 高 哲 段 冶 刘志利 孙天泽
刘洪涛 宋宝华 张善民 侯茂清 孙纪坤 孙 琼
李 佳 王大亮 王 辉 傅 曦 张 强

顾问

工业和信息化部软件与集成电路促进中心副主任 邱善勤

ARM 中国区总裁 谭军

Symbian 公司中国市场总监 卢竞

Altera 公司中国区总经理 徐平波

广州周立功单片机发展有限公司 周立功

《单片机与嵌入式系统应用》杂志社主编 何立民

北京麦克泰软件技术有限公司董事长 何小庆

中国软件行业协会嵌入式系统分会秘书长 郭淳学

前 言

第2版说明

本书第1版《嵌入式Linux应用程序开发详解》自2006年7月出版以来,受到了广大读者的一致好评,已经多次印刷,累计销量18000册。许多高等院校、职业学校和培训机构也将本书作为嵌入式专业的教材。许多读者提出了宝贵的意见和中肯的建议。

第2版图书在第1版基础上做了以下修订。

- 增加PPT教学课件:本书第2版增加了PPT教学课件,方便老师教学使用。
- 赠送嵌入式专家授课视频:本书第2版免费赠送超值的嵌入式教学视频,所讲内容均为嵌入式开发的热点内容。
- 内容调整:对书中过时的内容进行了升级,对书中的正文、图进行了大量的替换。
- 代码调整:对第1版图书中的代码进行大量的调整,并重新进行了编译、调试,使得第2版内容更加严谨。

本书写作背景

随着嵌入式技术的不断发展,近年来嵌入式技术在国内得到了广泛应用,在消费电子产品中得到了广泛应用;同时,越来越多的学校开始开设嵌入式系统课程,还有不少学校专门开设了嵌入式专业。

国内目前已经有不少嵌入式方面的图书面世,但大多以理论讲解为主,与目前嵌入式技术的实际应用结合不紧密,在这种背景下,我们组织编写了本书。

如何学习嵌入式系统

嵌入式领域是一个综合技术要求较高的技术领域,实际的嵌入式开发需要开发者掌握计算机体系结构、操作系统、多种芯片的原理结构、嵌入式Linux系统开发、嵌入式Linux内核等多种知识体系,这就造成了目前国内嵌入式开发人才极其缺乏的局面。

学习嵌入式系统首先要掌握常用嵌入式处理器、嵌入式操作系统、嵌入式编程语言,即ARM处理器、嵌入式Linux系统、嵌入式C语言,有了这些基础就可以进行嵌入式Linux应用开发、系统开发、驱动开发等的学习。

本书专门为那些已经有较全面的计算机基础,而又希望能快速进入嵌入式Linux相关行业

的开发人员而编写，希望能帮助读者快速跨过嵌入式开发的门槛。

本书的主要内容

本书分为3个部分。

第一部分主要讲解了Linux系统的基础知识。

第1章全面介绍了Linux的基本概念、Linux文件及文件系统。

第2章介绍了Linux的常用命令，包括用户系统常见命令、文件目录相关命令、压缩打包相关命令等，并简要分析了Linux的启动过程和Linux系统服务的配置。

第3章介绍了Linux下的C编程基础，由于C语言是嵌入式开发中最常用的语言，因此熟悉它的开发是非常重要的，本书详细介绍了Linux下的编辑器、编译器、调试器和工程管理等工具使用，并附以具体实例进行讲解。

本书的第二部分主要讲解了如何搭建嵌入式Linux环境。

第4章全面介绍了嵌入式系统的基础知识，包括嵌入式系统的基本概念、几种主流的嵌入式操作系统的介绍，并且简要介绍了ARM处理器及本书的实验平台S3C2410处理器，此外还简要介绍了嵌入式软件的开发流程。

第5章介绍了嵌入式Linux开发环境的搭建和嵌入式系统开发中常用工具的使用，包括如何使用tftp、配置串口、编译Linux内核、制作文件系统以及如何使用u-boot。

本书的第三部分是本书的重点内容——嵌入式Linux的应用开发。

第6章详细讲解了嵌入式Linux的I/O与文件系统的开发，内容包含Linux系统调用及用户编程接口、Linux中文件及文件描述符、嵌入式Linux串口应用开发、标准I/O开发。

第7章介绍了进程控制开发，包括Linux进程控制编程和Linux守护进程。

第8章介绍了进程间通信开发，内容包含管道通信、信号通信、共享内存、消息队列。

第9章介绍了多线程编程，内容包含Linux下线程分类、线程基本操作、线程访问控制。

第10章介绍了嵌入式Linux网络编程，内容包含TCP/IP协议、网络基础编程、网络高级编程等。

第11章介绍了设备驱动程序的开发，内容包含设备驱动概述、字符设备驱动编写、LCD驱动编写实例、块设备驱动编写、中断编程键盘驱动实现等。

第12章介绍了嵌入式图形界面的开发，内容包含嵌入式GUI简介和Qt/Embedded开发入门等。

以上各章在讲解中都给出了翔实的例子和实验，以便于读者尽快了解相关函数的使用。

适合的读者

本书可作为高等院校电子类、电气类、控制类专业高年级本科生、研究生学习嵌入式Linux的教材，也可供希望转入嵌入式领域的单片机工程技术人员参考使用，还可作为嵌入式Linux培训班的教材和参考书。

本书的阅读建议

本书以实践为特色，若读者能够动手操作书中安排的每一个环节，必定能取得很快的提高。嵌入式的开发与具体的硬件环境紧密相关，作者在讲解中尽量考虑了一些通用的方法以

减少具体操作对硬件环境的依赖。因此，本书所述的方法大多是在绝大多数开发板上运行的，对于没有开发板的读者，也可以在PC机上完整地学习嵌入式Linux应用开发的部分，有条件时再转入开发板上实践。

本书之外的内容

本书内容来自北京华清远见科技信息有限公司（www.farsight.com.cn）的培训课程资料，有关本书的相关源代码和嵌入式Linux更多的资料、公开课视频，请参见<http://www.farsight.com.cn/download/>。

本书第2版由赵苍明负责编写，孙天泽为本书审定写作提纲，同时，参与本书编写工作的还有刘燕祎、周晶、周丰、梅乐夫、房明浩、王亮、门店宏、吴洋、石峰、张圣亮、邱文勋、刘鲲、矫津毅、林远长、董前程、朱飞、岂兴明、汤嘉立、刘变红、周建兴、刘会灯、张高煜、邓志宝、刘明辉、李鹏、白学明、步士建等。在此，对以上人员致以诚挚的谢意。

由于时间仓促，加之水平有限，书中的不足之处在所难免，敬请读者批评指正。本书责任编辑的联系方法是huangyan@ptpress.com.cn，欢迎来信交流。

编者

2009年1月

目 录

第 1 章 Linux 快速入门	1	2.2.2 内核引导阶段	44
1.1 嵌入式 Linux 基础	1	2.2.3 init 阶段	45
1.1.1 Linux 发展概述	2	2.3 Linux 系统服务	47
1.1.2 Linux 作为嵌入式操作系统的优势	2	2.3.1 独立运行的服务	48
1.1.3 Linux 发行版本	3	2.3.2 xinetd 设定的服务	48
1.1.4 如何学习 Linux	4	2.3.3 系统服务的其他相关命令	49
1.2 Linux 安装	4	2.4 实验内容	50
1.2.1 基础概念	5	2.4.1 在 Linux 下解压常见软件	50
1.2.2 硬件需求	6	2.4.2 定制 Linux 系统服务	51
1.2.3 安装准备	7	2.5 本章小结	52
1.3 Linux 文件及文件系统	7	2.6 思考与练习	52
1.3.1 文件类型及文件属性	7	第 3 章 Linux 下 C 编程基础	53
1.3.2 文件系统类型介绍	9	3.1 Linux 下 C 语言编程概述	53
1.3.3 Linux 目录结构	10	3.1.1 C 语言简单回顾	53
1.4 实验内容——安装 Linux 操作系统	11	3.1.2 Linux 下 C 语言编程环境概述	54
1.5 本章小结	12	3.2 常用编辑器	55
1.6 思考与练习	12	3.2.1 进入 vi	55
第 2 章 Linux 基础命令	13	3.2.2 初探 emacs	57
2.1 Linux 常用命令	13	3.3 gcc 编译器	60
2.1.1 用户系统相关命令	14	3.3.1 gcc 编译流程解析	60
2.1.2 文件相关命令	21	3.3.2 gcc 编译选项分析	62
2.1.3 压缩打包相关命令	32	3.4 gdb 调试器	67
2.1.4 文件比较合并相关命令	34	3.4.1 gdb 使用流程	67
2.1.5 网络相关命令	38	3.4.2 gdb 基本命令	71
2.2 Linux 启动过程详解	43	3.5 make 工程管理器	75
2.2.1 概述	43	3.5.1 makefile 基本结构	75
		3.5.2 makefile 变量	76

3.5.3	makefile 规则	79	5.1.2	超级终端和 minicom 配置及使用	120
3.5.4	make 管理器的使用	80	5.1.3	下载映像到开发板	123
3.6	使用 autotools	80	5.1.4	编译嵌入式 Linux 内核	126
3.6.1	autotools 使用流程	81	5.1.5	Linux 内核源码目录结构	129
3.6.2	使用 autotools 所生成的 makefile	84	5.1.6	制作文件系统	130
3.7	实验内容	86	5.2	U-Boot 移植	134
3.7.1	vi 使用练习	86	5.2.1	Bootloader 介绍	134
3.7.2	用 gdb 调试程序的 bug	87	5.2.2	U-Boot 概述	136
3.7.3	编写包含多文件的 makefile	89	5.2.3	U-Boot 源码导读	137
3.7.4	使用 autotools 生成包含多文件的 makefile	91	5.2.4	U-Boot 移植主要步骤	142
3.8	本章小结	92	5.3	实验内容——创建 Linux 内核和文件系统	150
3.9	思考与练习	93	5.4	本章小结	151
第 4 章	嵌入式系统基础	94	5.5	思考与练习	151
4.1	嵌入式系统概述	94	第 6 章	文件 I/O 编程	152
4.1.1	嵌入式系统简介	94	6.1	Linux 系统调用及用户编程接口 (API)	152
4.1.2	嵌入式系统发展历史	95	6.1.1	系统调用	152
4.1.3	嵌入式系统的特点	96	6.1.2	用户编程接口 (API)	153
4.1.4	嵌入式系统的体系结构	96	6.1.3	系统命令	153
4.1.5	几种主流嵌入式操作系统分析	97	6.2	Linux 中文件及文件描述符概述	153
4.2	ARM 处理器硬件开发平台	99	6.3	底层文件 I/O 操作	154
4.2.1	ARM 处理器简介	99	6.3.1	基本文件操作	154
4.2.2	ARM 体系结构简介	101	6.3.2	文件锁	158
4.2.3	ARM9 体系结构	101	6.3.3	多路复用	163
4.2.4	S3C2410 处理器详解	104	6.4	嵌入式 Linux 串口应用编程	171
4.3	嵌入式软件开发流程	109	6.4.1	串口概述	171
4.3.1	嵌入式系统开发概述	109	6.4.2	串口设置详解	172
4.3.2	嵌入式软件开发概述	109	6.4.3	串口使用详解	181
4.4	实验内容——使用 JTAG 烧写 Nand Flash	114	6.5	标准 I/O 编程	185
4.5	本章小结	116	6.5.1	基本操作	185
4.6	思考与练习	117	6.5.2	其他操作	189
第 5 章	嵌入式 Linux 开发环境的搭建	118	6.6	实验内容	191
5.1	嵌入式开发环境的搭建	118	6.6.1	文件读写及上锁	191
5.1.1	嵌入式交叉编译环境的搭建	118	6.6.2	多路复用式串口操作	198
			6.7	本章小结	202

6.8 思考与练习	202	8.7.1 管道通信实验	271
第 7 章 进程控制开发	203	8.7.2 共享内存实验	275
7.1 Linux 进程概述	203	8.8 本章小结	280
7.1.1 进程的基本概念	203	8.9 思考与练习	280
7.1.2 Linux 下的进程结构	205	第 9 章 多线程编程	281
7.1.3 Linux 下进程的模式和 类型	205	9.1 Linux 线程概述	281
7.1.4 Linux 下的进程管理	205	9.1.1 线程概述	281
7.2 Linux 进程控制编程	206	9.1.2 线程机制的分类和 特性	282
7.3 Linux 守护进程	217	9.2 Linux 线程编程	282
7.3.1 守护进程概述	217	9.2.1 线程基本编程	282
7.3.2 编写守护进程	218	9.2.2 线程之间的同步与 互斥	286
7.3.3 守护进程的出错处理	221	9.2.3 线程属性	293
7.4 实验内容	225	9.3 实验内容——“生产者消费者” 实验	297
7.4.1 编写多进程程序	225	9.4 本章小结	302
7.4.2 编写守护进程	229	9.5 思考与练习	302
7.5 本章小结	231	第 10 章 嵌入式 Linux 网络编程	303
7.6 思考与练习	232	10.1 TCP/IP 概述	303
第 8 章 进程间通信	233	10.1.1 OSI 参考模型及 TCP/IP 参考模型	303
8.1 Linux 下进程间通信概述	233	10.1.2 TCP/IP 协议族	304
8.2 管道	234	10.1.3 TCP 和 UDP	304
8.2.1 管道概述	234	10.2 网络基础编程	307
8.2.2 管道系统调用	235	10.2.1 socket 概述	307
8.2.3 标准流管道	237	10.2.2 地址及顺序处理	307
8.2.4 FIFO	239	10.2.3 socket 基础编程	312
8.3 信号	243	10.3 网络高级编程	319
8.3.1 信号概述	243	10.4 实验内容——NTP 协议实现	324
8.3.2 信号发送与捕捉	245	10.5 本章小结	330
8.3.3 信号的处理	247	10.6 思考与练习	330
8.4 信号量	254	第 11 章 嵌入式 Linux 设备驱动开发	331
8.4.1 信号量概述	254	11.1 设备驱动概述	331
8.4.2 信号量的应用	255	11.1.1 设备驱动简介及 驱动模块	331
8.5 共享内存	260	11.1.2 设备分类	332
8.5.1 共享内存概述	260	11.1.3 设备号	333
8.5.2 共享内存的应用	260	11.1.4 驱动层次结构	333
8.6 消息队列	266		
8.6.1 消息队列概述	266		
8.6.2 消息队列的应用	266		
8.7 实验内容	271		

11.1.5 设备驱动程序与外界的接口	334	12.1.2 MiniGUI	373
11.1.6 设备驱动程序的特点	334	12.1.3 Microwindows、Tiny X 等	374
11.2 字符设备驱动编程	335	12.2 Qt/Embedded 开发入门	374
11.3 GPIO 驱动程序实例	343	12.2.1 Qt/Embedded 介绍	374
11.3.1 GPIO 工作原理	343	12.2.2 Qt/Embedded 信号和插槽机制	377
11.3.2 GPIO 驱动程序	345	12.2.3 搭建 Qt/Embedded 开发环境	380
11.4 块设备驱动编程	351	12.2.4 Qt/Embedded 窗口部件	382
11.5 中断编程	354	12.2.5 Qt/Embedded 图形界面编程	385
11.6 按键驱动程序实例	355	12.2.6 Qt/Embedded 对话框设计	387
11.6.1 按键工作原理	355	12.3 实验内容——使用 Qt 编写“Hello, World”程序	391
11.6.2 按键驱动程序	356	12.4 本章小结	396
11.6.3 按键驱动测试程序	363		
11.7 实验内容——test 驱动	365		
11.8 本章小结	371		
11.9 思考与练习	371		
第 12 章 Qt 图形编程基础	372		
12.1 嵌入式 GUI 简介	372		
12.1.1 Qt/Embedded	373		

第 1 章

Linux 快速入门

本章目标

嵌入式 Linux 是以 Linux 为基础的操作系统，只有熟练使用 Linux 系统之后，才能在嵌入式 Linux 开发领域得心应手。通过本章的学习，读者能够掌握如下内容。

- 能够独立安装 Linux 操作系统
- 能够熟练使用 Linux 系统的基本命令
- 认识 Linux 系统启动过程
- 能够独立在 Linux 系统中安装软件
- 能够独立设置 Linux 环境变量
- 能够独立定制 Linux 服务

1.1 嵌入式 Linux 基础

自由开源软件在嵌入式应用上受到青睐，Linux 日益成为主流的嵌入式操作系统之一。随着 MOTOROLA 手机 A760、IBM 智能型手表 WatchPad、SharpPDA Zaurus 等一款款高性能“智能数码产品”的出现，以及 Motorola、Samsung、MontaVista、Philips、Nokia、IBM、SUN 等众多国际顶级巨头的加入，嵌入式 Linux 的队伍越来越庞大了。目前，国外不少大学、研究机构和知名公司都加入了嵌入式 Linux 的开发工作，成熟的嵌入式 Linux 产品不断涌现。

2004 年全球嵌入式 Linux 市场规模已达 9150 万美元，2005 年有 1.336 亿美元，2006 年有 1.653 亿美元，2007 年达到 2.011 亿美元，每年平均增长 30%。

究竟是什么原因让嵌入式 Linux 系统发展如此迅速。业界归纳为三大原因：第一，Linux 在嵌入式系统所需的实时性、电源管理等核心技术方面不断发展；第二，国际标准组织（如 OSDL、CELF 等）持续建立嵌入式 Linux 相关标准，有效解决版本分歧与兼容性问题；第三，业界主导组织、开发厂商等不断推出嵌入式 Linux 相关开发工具、维护系统。

嵌入式 Linux 以年费订阅方式为主，与其他的以产品利润为收入方式的嵌入式系统不同，弹性的捆绑销售策略，助其成功地逐年提高市场占有率，从 2004 年的 46.8% 扩大到 2007 年的 56.4%。

国际有名的嵌入式 Linux 操作系统提供商 MontaVista，收购了 PalmSource 的爱可信和奇

趣科技等，加强了对中国市场的投入，并在整个嵌入式操作系统市场中，占据了重要地位。而嵌入式操作系统的领先厂商，也改变了原来的单一产品路线，开始推出自己的 Linux 软件产品，实现“两条腿走路”。国内的嵌入式软件厂商也以 Linux 为突破口，纷纷开发各种基于 Linux 的操作系统产品。这些嵌入式 Linux 厂商已经形成了一个不容忽视的群体。

以下就从 Linux 开始，一层层揭开嵌入式 Linux 的面纱。

1.1.1 Linux 发展概述

简单地讲，Linux 是指一套免费使用和自由传播的类 UNIX 操作系统。人们通常所说的 Linux 是 Linus Torvalds 所写的 Linux 操作系统内核。

当时的 Linus 还是芬兰赫尔辛基大学的一名学生，他主修的课程中有一门课是操作系统，而且这门课是专门研究程序的设计和执行的。最后这门课程提供了一种称为 Minix 的初期 UNIX 系统。Minix 是一款仅为教学而设计的操作系统，而且功能有限。因此，和 Minix 的众多使用者一样，Linus 也希望能给它添加一些功能。

在之后的几个月里，Linus 根据实际的需要编写了磁盘驱动程序以便下载访问新闻组的文件，又编写了个文件系统以便能够阅读 Minix 文件系统中的文件。这样，“当你有了任务切换，有了文件系统和设备驱动程序后，这就是 UNIX，或者至少是其内核。”于是，0.0.1 版本的 Linux 就诞生了。

Linus 从一开始就决定自由传播 Linux，他把源代码发布在网上，于是，众多的爱好者和程序员也都通过互联网加入到 Linux 的内核开发工作中。这个思想与 FSF (Free Software Foundation) 资助发起的 GNU (GNU's Not UNIX) 的自由软件精神不谋而合。

GNU 是为了推广自由软件的精神以实现一个自由的操作系统，然后从应用程序开始，实现其内核。而当时 Linux 的优良性能备受 GNU 的赏识，于是 GNU 就决定采用 Linus 及其开发者的内核。在他们的共同努力下，Linux 这个完整的操作系统诞生了。其中的程序开发共同遵守 General Public License (GPL) 协议，这是最开放也是最严格的许可协议方式，这个协议规定了源码必须可以无偿地获取并且修改。因此，从严格意义上说，Linux 应该叫做 GNU/Linux，其中许多重要的工具如 gcc、gdb、make、emacs 等都是 GNU 贡献的。

这个“婴儿版”的操作系统以平均两星期更新一次的速度迅速成长，如今的 Linux 已经有超过 250 种发行版本，且可以支持所有体系结构的处理器，如 X86、PowerPC、ARM、Xscale 等，也可以支持带 MMU 或不带 MMU 的处理器。到目前为止，它的内核版本也已经从原先的 0.0.1 发展到现在的 2.6.xx。

1.1.2 Linux 作为嵌入式操作系统的优势

从 Linux 系统的发展过程可以看出，Linux 从最开始就是一个开放的系统，并且它始终遵循着源代码开放的原则，它是一个成熟而稳定的网络操作系统，作为嵌入式操作系统有如下优势。

1. 低成本开发系统

Linux 的源码开放性允许任何人获取并修改 Linux 的源码。这样一方面大大降低了开发的成本，另一方面又可以提高开发产品的效率。并且还可以在 Linux 社区中获得支持，用户只需向邮件列表发一封邮件，即可获得作者的支持。

2. 可应用于多种硬件平台

Linux 可支持 X86、PowerPC、ARM、Xscale、MIPS、SH、68K、Alpha、Sparc 等多种体系结构，并且已经被移植到多种硬件平台。这对于经费、时间受限制的研究与开发项目是很有吸引力的。Linux 采用一个统一的框架对硬件进行管理，同时从一个硬件平台到另一个硬件平台的改动与上层应用无关。

3. 可定制的内核

Linux 具有独特的内核模块机制，它可以根据用户的需要，实时地将某些模块插入到内核中或者从内核中移走，并能根据嵌入式设备的个性需要量体裁衣。经裁减的 Linux 内核最小可达到 150KB 以下，尤其适合嵌入式领域中资源受限的实际情况。当前的 2.6 内核加入了许多嵌入式友好特性。

4. 性能优异

Linux 系统内核精简、高效并且稳定，能够充分发挥硬件的功能，因此它比其他操作系统的运行效率更高。在个人计算机上使用 Linux，可以将它作为工作站。它也非常适合在嵌入式领域中应用，对比其他操作系统，它占用的资源更少，运行更稳定，速度更快。

5. 良好的网络支持

Linux 是首先实现 TCP/IP 协议栈的操作系统，它的内核结构在网络方面是非常完整的，并提供了对包括十兆位、百兆位及千兆位的以太网，还有无线网络、Token ring（令牌环）和光纤甚至卫星的支持，这对现在依赖于网络的嵌入式设备来说无疑是很好的选择。

1.1.3 Linux 发行版本

由于 Linux 属于 GNU 系统，而这个系统采用 GPL 协议，并保证了源代码的公开，于是众多组织或公司在 Linux 内核源代码的基础上进行了一些必要的修改加工，然后再开发一些配套的软件，并把它整合成一个自己的发布版 Linux。除去非商业组织 Debian 开发的 Debian GNU/Linux 外，美国的 Red Hat 公司发行了 Red Hat Linux，法国的 Mandrake 公司发行了 Mandrake Linux，德国的 SUSE 公司发行了 SUSE Linux，我国众多公司也发行了中文版的 Linux，如著名的红旗 Linux。Linux 目前已经有超过 250 个发行版本。

下面仅对 Red Hat、Debian、Mandrake 等具有代表性的 Linux 发行版本进行介绍。

1. Red Hat

全世界的 Linux 用户最熟悉的发行版想必就是 Red Hat 了。Red Hat 最早是由 Bob Young 和 Marc Ewing 在 1995 年创建的。目前 Red Hat 分为两个系列：由 Red Hat 公司提供收费技术支持和更新的 Red Hat Enterprise Linux (RHEL, Red Hat 的企业版)，以及由社区开发的免费的桌面版 Fedora Core。

Red Hat 企业版有 3 个版本——AS、ES 和 WS。AS 是其中功能最为强大和完善的版本。而正统的桌面版 Red Hat 版本更新早已停止，最后一版是 Red Hat 9.0。本书就以稳定性高的

RHEL AS 作为安装实例进行讲解。

官方主页：<http://www.redhat.com/>。

2. Debian

之所以把 Debian 单独列出，是因为 Debian GNU/Linux 是一个非常特殊的版本。在 1993 年，伊恩·默多克（Ian Murdock）发起 Debian 计划，它的开发模式和 Linux 及其他开放性源代码操作系统的精神一样，都是由超过 800 位志愿者通过互联网合作开发而成的。一直以来，Debian GNU/Linux 被认为是最正宗的 Linux 发行版本，而且它是一个完全免费、高质量的且与 UNIX 兼容的操作系统。

Debian 系统分为 3 个版本，分别为稳定版(Stable)、测试版(Testing)和不稳定版(Unstable)。每次发布的版本都是稳定版，而测试版在经过一段时间的测试证明没有问题后会成为新的稳定版。Debian 拥有超过 8710 种不同的软件，每一种软件都是自由的，而且有非常方便的升级安装指令，基本囊括了用户的所有需要。Debian 也是最受欢迎的嵌入式 Linux 之一。

官方主页：<http://www.debian.org/>。

3. 我国的发行版本及其他

目前国内的红旗、新华等都发行了自己的 Linux 版本。

除了前面所提到的这些版本外，业界还存在着诸如 gentoo、LFS 等适合专业人士使用的版本。在此不做介绍，有兴趣的读者可以自行查找相关的资料做进一步的了解。

1.1.4 如何学习 Linux

正如人们常说的“实践出真知”，学习 Linux 的过程也一样。只有通过大量的动手实践才能真正地领会 Linux 的精髓，才能迅速掌握在 Linux 上的应用开发，相信有编程语言经验的读者一定会认同这一点。因此，在本书中笔者安排了大量的实验环节和课后实践环节，希望读者尽可能多参与。

另外要指出的是，互联网也是一个很好的学习工具，一定要充分地加以利用。正如编程一样，实践的过程中总会出现多种多样的问题，笔者在写作的过程当中会尽可能地考虑可能出现的问题，但限于篇幅和读者的实际情况，不可能考虑到所有可能出现的问题，所以希望读者能充分利用互联网这一共享的天空，在其中寻找答案。以下列出了国内的一些 Linux 论坛：

<http://www.linuxfans.org>

<http://www.linuxforum.net/>

<http://www.linuxeden.com/forum/>

<http://www.newsmth.net>

1.2 Linux 安装

有了一个初步的了解后，读者是否想亲自试一下？其实安装 Linux 是一件很容易的事情，不过在开始安装之前，还需要了解一下在 Linux 安装过程中可能遇到的一些基本知识以及它与 Windows 的区别。

1.2.1 基础概念

1. 文件系统、分区和挂载

文件系统是指操作系统中与管理文件有关的软件和数据。Linux 的文件系统和 Windows 中的文件系统有很大的区别，Windows 文件系统是以驱动器的盘符为基础的，而且每一个目录与相应的分区对应，例如“E:\workplace”是指此文件在 E 盘这个分区下。而 Linux 恰好相反，文件系统是一棵文件树，且它的所有文件和外部设备（如硬盘、光驱等）都是以文件的形式挂在这个文件树上，例如“/usr/local”。对于 Windows 而言，就是指所有分区都是在一些目录下。总之，在 Windows 下，目录结构属于分区；Linux 下，分区属于目录结构。其关系如图 1.1 和图 1.2 所示。

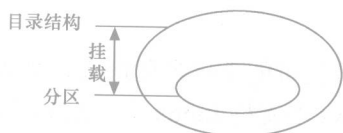


图 1.1 Linux 下目录与分区关系

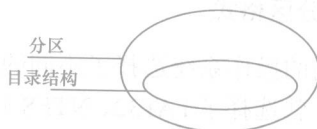


图 1.2 Windows 下目录与分区关系图

因此，在 Linux 中把每一个分区和某一个目录对应，以后再对这个目录的操作就是对这个分区的管理，这样就实现了硬件管理手段和软件目录管理手段的统一。这个把分区和目录对应的过程叫做挂载（Mount），而这个挂载在文件树中的位置就是挂载点。这种对应关系可以由用户随时中断和改变。

●想一想 Linux 文件系统的挂载特性给用户能带来怎样的好处呢？

2. 主分区、扩展分区和逻辑分区

硬盘分区是针对一个硬盘进行操作的，它可以分为：主分区、扩展分区、逻辑分区。其中主分区就是包含操作系统启动所必需的文件和数据的硬盘分区，要在硬盘上安装操作系统，则该硬盘必须要有一个主分区，而且其主分区的数量可以是 1~3 个；扩展分区也就是除主分区外的分区，但它不能直接使用，必须再将它划分为若干个逻辑分区才可使用，其数量可以有 0 或 1 个；而逻辑分区则在数量上没有什么限制。它们的关系如图 1.3 所示。

一般而言，对于先装了 Windows 的用户，Windows 的 C 盘是装在主分区上的，可以把 Linux 安装在另一个主分区或者扩展分区上。为了安装方便安全起见，一般采用把 Linux 装在多余的逻辑分区上，如图 1.4 所示。

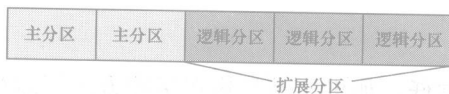


图 1.3 Linux 下主分区、扩展分区、逻辑分区示意图

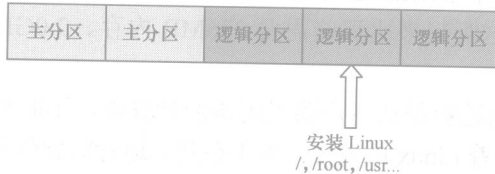


图 1.4 Linux 安装的分区的示意图

✦ 小知识

通常，在 Windows 下的盘符和 Linux 设备文件的对应关系如下：

C 盘 —— /dev/hda1（主分区）
D 盘 —— /dev/hda5（逻辑分区）
E 盘 —— /dev/hda6（逻辑分区）

3. SWAP 交换分区

在硬件条件有限的情况下，为了运行大型的程序，Linux 在硬盘上划出一个区域来当作临时的内存，而 Windows 操作系统把这个区域叫做虚拟内存，Linux 把它叫做交换分区 swap。在安装 Linux 建立交换分区时，一般将其设为内存大小的 2 倍，当然也可以设为更大。

4. 分区格式

不同的操作系统选择了不同的格式，同一种操作系统也可能支持多种格式。微软公司的 Windows 就选择了 FAT32、NTFS 两种格式，但是 Windows 不支持 Linux 上常见的分区格式。Linux 是一个开放的操作系统，它最初使用 EXT2 格式，后来使用 EXT3 格式，但是它同时支持非常多的分区格式，包括很多大型机上 UNIX 使用的 XFS 格式，也包括微软公司的 FAT 以及 NTFS 格式。

5. GRUB

GRUB 是一种引导装入器（类似在嵌入式中非常重要的 bootloader），它负责装入内核并引导 Linux 系统，位于硬盘的起始部分。由于 GRUB 多方面的优越性，如今的 Linux 一般都默认采用 GRUB 来引导 Linux 操作系统。但事实上它还可以引导 Windows 等多种操作系统。

✦ 小知识

在安装 Windows 和 Linux 双系统后，系统是以 Linux 的 GRUB 作为引导装入器来选择启动 Windows 或 Linux 的，因此，若此时直接在 Windows 下把 Linux 的分区删除，会导致系统因没有引导装入器而无法启动 Windows，这点要格外小心。

6. root 权限

Linux 也是一个多用户的系统（在这一点上类似 Windows XP），不同的用户和用户组会有不同的权限，其中把具有超级权限的用户称为 root 用户。root 的默认主目录在“/root”下，而其他普通用户的目录则在“/home”下。root 的权限极高，它甚至可以修改 Linux 的内核，因此建议初学者要慎用 root 权限，不然一个小小的参数设置错误很有可能导致系统的严重问题。

1.2.2 硬件需求

Linux 对硬件的需求非常低。如果要是只想在字符方式下运行，那么一台 386 的计算机已经可以用来安装 Linux 了；如果想运行 X-Windows，那也只需要一台 16MB 内存、600MB 硬盘的 486 计算机即可。这听起来比那些需要 256MB 内存、2.0GHz 的操作系统要好得多，事实上也正是如此。

现在软件和硬件行业的趋势是让用户购买更快的计算机，不断扩充内存和硬盘，而 Linux 却不受这个趋势的影响。随着 Linux 的发展，由于在其上运行的软件越来越多，因此它所需要的配置越来越高，但是用户可以有选择地安装软件，从而节省资源。既可以运行在 Pentium 4 处理