

嵌入式 Linux 案例开发指南



冯 伟 王 博 编著

- Linux系统的进程与线程管理
- 嵌入式Linux的网络体系结构和网卡驱动
- 嵌入式Linux中Socket套接口开发
- 驱动程序的开发
- 嵌入式 Linux Qt程序的开发
- Java与嵌入式开发
- 嵌入式数据库的开发
- 基于ARM9的嵌入式远程视频采集系统的实现方法



中国电力出版社
www.cepp.com.cn

嵌入式
Linux
案例开发指南



中国电力出版社

www.cepp.com.cn

内 容 提 要

本书以案例的形式系统地讲解了嵌入式 Linux 开发所需要的各种知识和实际应用。全书共 15 章, 主要介绍了嵌入式 Linux 系统基础、开发基础、Linux 高级编程、驱动开发, 以及综合实验等部分知识, 主要内容包括 Shell 编程, GDB 调试工具, Linux 系统内核的编译、进程和线程的概念, Socket 编程, Linux 图形化界面开发的技巧, Java 语言在嵌入式 Linux 开发中的应用, 嵌入式 Linux 数据库, Linux 驱动开发基础, 字符设备驱动开发等知识, 最后通过两个综合实验巧妙地将枯燥的知识学习变成了有趣的动手实践。

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, 也可供广大嵌入式领域的科研和工程技术人员参考使用, 还可作为相关培训班的培训教材。

图书在版编目 (CIP) 数据

嵌入式 Linux 案例开发指南 / 冯伟, 王博编著. —北京: 中国电力出版社, 2008
ISBN 978-7-5083-7478-9

I. 嵌… II. ①冯…②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 (2008) 第 143197 号

责任编辑: 夏华香
责任校对: 崔燕菊
责任印制: 郭华清

书 名: 嵌入式 Linux 案例开发指南

作 者: 冯 伟 王 博

出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044

电话: (010) 68362602 传真: (010) 68316497

印 刷: 北京市同江印刷厂印刷

开本尺寸: 185mm × 260mm 印 张: 29 字 数: 687 千字

书 号: ISBN 978-7-5083-7478-9

版 次: 2008 年 11 月北京第 1 版

印 次: 2008 年 11 月第 1 次印刷

印 数: 0001—3000 册

定 价: 49.80 元

敬 告 读 者

本书封面贴有防伪标签, 加热后中心图案消失
本书如有印装质量问题, 我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

一、嵌入式系统的发展前景

近年来,随着计算机技术、通信技术的飞速发展,特别是互联网的迅速普及和 3C(计算机、通信、消费电子)合一的加速,微型化和专业化成为发展的新趋势,嵌入式产品成为信息产业的主流。由于 Linux 系统源码开放、可以定制、易于移植等优点,它在嵌入式系统中获得了越来越广泛的应用。越来越多的企业和研发机构都转向嵌入式 Linux 的开发和研究上,在新兴的嵌入式操作系统领域内也获得了飞速发展。

二、本书的写作意义

伴随着市场对嵌入式开发人才需求的增大,广大读者迫切需要一本快速学习的教材。嵌入式 Linux 的开发技术是一种高起点的技术,需要读者掌握多种知识,包括各种软件和硬件方面的知识,软件方面的知识主要包括操作系统、内核方面的知识,硬件方面包括各种处理器、开发板的性能知识。市场上现有的关于嵌入式 Linux 开发的书籍,大都以知识为导向,按照知识点进行讲解,这样,读者缺乏实践的机会,也难以深刻理解知识点。

本书以案例为线索,通过一个个例子,使读者在实践中理解知识,掌握技巧;把枯燥的知识学习变成有趣的动手实践。也只有通过这样的实践,才能真正具有嵌入式 Linux 开发所需要的实际动手能力。

三、本书主要内容

本书分为 5 个部分共 15 章,主要介绍了嵌入式 Linux 系统基础、Linux 开发基础,以及 Linux 高级编程、驱动开发,以及综合实验等知识。

四、本书阅读技巧

阅读本书的最好方法,就是按照本书各个案例的顺序进行实践,在实践中理解知识,发现问题,寻找解决问题的方法,从而培养读者真正的开发能力。由于嵌入式系统的性能和具体的开发板密切相关,书中的程序,笔者尽量提高了它的通用性,可在大多数开发板上运行。读者可根据自己的开发需要,选择相应的开发板进行开发。

五、致谢

本书在编写的过程中得到了很多人的支持和帮助。首先要感谢我的父母,他们不仅在背后默默支持我,还为本书的部分章节提供了很有价值的建议。同时要感谢我的同事和同学伍坚、江庆松、宋飞、张茂林等,书中的不少内容正是来自于他们的宝贵意见和帮助,也得益于与他们的讨论和交流。另外,本书的编写还得到了孙亮、张玉玲、杨幸子、王泼、黄伟、杨冬等的支持和帮助,在此一并表示感谢。

由于时间仓促,作者水平所限,书中不足之处在所难免,敬请读者批评指正。也欢迎广大朋友来信交流, E-mail: foxandking@hotmail.com。

作 者
2008 年 8 月

目 录

前 言

第 1 章 嵌入式 Linux 系统简介	1
1.1 嵌入式系统	1
1.1.1 概述	1
1.1.2 嵌入式系统的发展历史	2
1.1.3 嵌入式 Linux 的特性	3
1.2 嵌入式 Linux 操作系统简介	3
1.2.1 Linux 系统	3
1.2.2 从 Linux 到嵌入式 Linux	4
1.2.3 常用嵌入式 Linux 系统简介	6
1.3 常见嵌入式处理器简介	7
1.3.1 ARM 处理	7
1.3.2 Motorola 公司处理器	8
1.3.3 IBM Power PC 处理器	8
1.3.4 Intel x86 系列处理器	9
1.3.5 MIPS	9
1.4 选择开发平台	9
1.4.1 硬件平台的选择	10
1.4.2 软件平台的选择	11
1.4.3 集成开发环境考虑的因素	12
1.4.4 硬件调试工具的选择	13
1.4.5 软件组件的选择	13
第 2 章 Linux 编程基础	14
2.1 Shell 编程	14
2.2 Makefile 写法	17
2.2.1 为什么要使用 makefile	17
2.2.2 基本 makefile 结构	18
2.2.3 编写 make 规则	20
2.2.4 一个比较有效的 makefile	23
2.2.5 使用 autotools 建立 makefile	24
2.3 开发工具	28
2.3.1 Vi 工具简介	28
2.3.2 GCC 工具	32

2.3.3	GDB 调试.....	36
2.3.4	GDB 调试嵌入式系统.....	37
第 3 章	Linux 操作系统实践.....	42
3.1	Linux 操作系统分析.....	42
3.1.1	Linux 源码文件系统目录.....	42
3.1.2	Linux 在 x86 PC 机上的启动分析.....	43
3.2	Linux 内核分析.....	45
3.2.1	Linux 进程线程调度.....	45
3.2.2	内存管理.....	49
3.2.3	嵌入式 Linux 实时性解决方案.....	52
3.3	Linux 操作系统的安装.....	56
3.4	内核编译实践.....	68
3.4.1	编译步骤.....	68
3.4.2	内核选项.....	69
第 4 章	嵌入式 Linux 系统交叉编译环境的安装与配置.....	80
4.1	交叉编译环境的安装.....	80
4.1.1	相关软件包的下载.....	80
4.1.2	Linux 头文件的安装.....	82
4.1.3	编译安装 binutils.....	83
4.1.4	编译安装 gcc 的 c 编译器.....	83
4.1.5	编译安装 glibc.....	84
4.2	交叉编译环境的配置.....	84
4.2.1	Bootloader 概述.....	84
4.2.2	常见的 Linux bootloader 的介绍.....	86
4.2.3	三星 SMDK2410 开发板上的 bootloader 分析.....	87
4.2.4	μ -Boot 在 s3c2410 上的移植分析.....	101
第 5 章	Linux 系统的进程与线程管理.....	104
5.1	进程/线程的概念.....	104
5.1.1	进程的概念.....	104
5.1.2	线程的概念.....	108
5.1.3	进程状态及转换.....	110
5.2	进程的基本操作.....	110
5.2.1	进程信息获取函数.....	110
5.2.2	设置进程的属性.....	112
5.2.3	创建进程.....	114
5.2.4	执行进程.....	115
5.2.5	退出进程.....	120
5.2.6	跟踪进程.....	125
5.3	进程通信与同步.....	129

5.4	线程基本操作与相关函数	145
5.4.1	线程的执行层次和流程	145
5.4.2	线程相关函数	146
第 6 章	Linux 驱动开发基础	151
6.1	引言	151
6.2	模块编程	152
6.2.1	模块编程简介	152
6.2.2	helloworld 模块	153
6.2.3	程序编译及注意事项	154
6.2.4	从 2.4 到 2.6 驱动开发的变化	155
第 7 章	嵌入式 Linux 的网络体系结构和网卡驱动	165
7.1	嵌入式 Linux 的网络协议栈结构	165
7.1.1	网络协议栈结构	165
7.1.2	嵌入式系统中实现网络协议	165
7.1.3	Linux TCP/IP 网络协议栈	166
7.1.4	Linux 网络设备驱动程序分析	167
7.1.5	网络设备驱动程序的加载	174
7.2	DM9000 网卡驱动程序的编写	176
7.2.1	DM9000 网卡功能简介	176
7.2.2	DM9000 相关寄存器	176
7.2.3	DM9000 以太网控制器的内存结构	180
7.2.4	DM9000 网卡驱动程序分析	180
7.3	FTP 应用程序在嵌入式 Linux 下的应用	184
7.3.1	FTP 协议的描述	184
7.3.2	嵌入式 FTP 服务程序原理	185
第 8 章	嵌入式 Linux 中 Socket 套接口开发	188
8.1	引言	188
8.2	Socket 编程简介	188
8.2.1	基本概念简介	189
8.2.2	服务方式	190
8.2.3	客户机/服务器模式	191
8.2.4	套接字分类	192
8.3	TCP Socket 案例分析	192
8.3.1	常用函数介绍	192
8.3.2	代码分析	195
8.3.3	运行结果分析	198
8.4	UDP Socket 案例分析	199
8.4.1	UDP Socket 简介	199
8.4.2	代码分析	200

8.4.3	程序运行结果分析	203
8.5	raw socket 案例分析	204
8.5.1	raw socket 简介	204
8.5.2	一个简单的例子	209
8.5.3	程序运行结果分析	211
8.6	总结与提高	212
8.6.1	一个综合实验	212
8.6.2	运行结果分析	230
第 9 章	Linux 字符设备驱动程序	232
9.1	Linux 字符设备驱动程序简介	232
9.1.1	字符设备驱动相关函数和结构体简介	232
9.1.2	一个简单的字符设备驱动程序例子	236
9.2	Linux 系统中断、异常和系统调用的分析	238
9.2.1	Linux 系统中断的分析	238
9.2.2	ARM 嵌入式设备对中断的处理	239
9.2.3	Linux 系统异常的分析	240
9.2.4	ARM 嵌入式设备对异常的处理	241
9.2.5	Linux 系统调用的分析	242
9.2.6	Linux 处理中断的相关函数	243
9.2.7	中断与异常的简单测试	247
9.3	Linux 的 LED 驱动程序编写	248
9.3.1	LED 显示输出的原理和结构	248
9.3.2	简单的三色 LED 驱动程序的编写	249
第 10 章	嵌入式 USB 设备的驱动开发	252
10.1	USB 技术介绍	252
10.1.1	USB 技术的发展历史	252
10.1.2	USB 技术概述	253
10.2	USB 系统构成	254
10.2.1	USB 应用系统设计的组成	255
10.2.2	USB 嵌入式系统的数据传输模式	256
10.2.3	USB 应用系统的设备初始化和数据传输	258
10.3	USB 主机驱动结构和相关函数	259
10.3.1	USB 主机驱动结构	259
10.3.2	主要的通用数据结构和接口函数	261
10.3.3	针对特定芯片组的 USB Host 驱动程序	265
10.4	USB 设备驱动结构和相关函数	268
10.4.1	数据结构 usb_driver	268
10.4.2	USB 数据传输相关函数	268
10.4.3	USB 驱动程序注册函数	271
10.4.4	USB 驱动程序卸载函数	272

10.5	设备类驱动与相关函数	273
10.5.1	Linux 内核中 USB 鼠标实现层次	273
10.5.2	相关结构体和主要函数功能	274
10.6	设备端 USB 驱动	278
10.7	S3C2410 USB 控制器	279
第 11 章	嵌入式 Linux Qt 程序的开发	282
11.1	图形用户界面的发展现状	282
11.2	Qt 简介	283
11.2.1	Qt/Embedded 体系结构	283
11.2.2	Qt 编程的特点	283
11.2.3	QMake 简介	288
11.3	建立 Qt/Embedded 开发环境	288
11.3.1	建立 QTE 开发环境	290
11.3.2	建立 QTE 程序的 PC 机仿真环境	291
11.3.3	Qt 测试: 编写运行 Qt 环境下的测试程序	291
11.4	一个简单的 Qt 程序	293
第 12 章	Java 与嵌入式开发	301
12.1	Java 简介	301
12.1.1	Java 语言简介	301
12.1.2	Java 语言的特点	302
12.1.3	Java 程序的运行过程	302
12.1.4	Java 与嵌入式的结合	303
12.2	Linux 下的 Java 开发简介	304
12.2.1	Java 开发环境设置	304
12.2.2	一个简单的 Java 程序	307
12.3	J2ME	307
12.3.1	J2ME 概述	307
12.3.2	J2ME 的体系结构	308
12.3.3	J2ME 配置简介	309
12.3.4	J2ME 简表	311
12.3.5	其他 J2ME 规范	318
12.4	使用 Java 语言在 Palom OS 上进行开发	320
12.4.1	Sun 的解决方案: KVM、Configuration 和 Profile	321
12.4.2	Kawt 的解决方案	322
12.4.3	IBM 解决方案	323
12.4.4	Waba/SuperWaba 解决方案	324
12.5	Java 虚拟机	325
12.5.1	Java 虚拟机简介	325
12.5.2	Java 虚拟机的体系结构	326
12.5.3	虚拟机移植案例	330

第 13 章 嵌入式数据库的开发	333
13.1 嵌入式数据库介绍	333
13.1.1 嵌入式数据库简介	333
13.1.2 嵌入式实时数据库应用现状	334
13.1.3 常用嵌入式数据库介绍	335
13.2 mSQL 数据库	338
13.2.1 mSQL 构成	338
13.2.2 mSQL 安装	339
13.2.3 mSQL 的配置	342
13.3 mSQL 数据库的使用	344
13.3.1 mSQL 目录结构	344
13.3.2 mSQL 服务器的标准命令	344
13.3.3 mSQL 支持的 SQL 语句	347
13.3.4 mSQL 数据库应用实例	351
13.4 利用 mSQL 提供的 API 进行二次开发	353
13.4.1 mSQL 提供的 API 函数	354
13.4.2 案例分析	361
第 14 章 基于 ARM9 的嵌入式远程视频采集系统的实现	365
14.1 视频采集系统体系结构设计	365
14.1.1 视频采集系统的硬件结构	365
14.1.2 视频采集系统的软件结构	366
14.2 视频采集系统的系统程序配置	366
14.2.1 JFFS2 文件系统的建立	366
14.2.2 摄像头驱动的加载	369
14.3 视频采集系统的应用程序配置	369
14.3.1 视频采集应用程序的设计	369
14.3.2 视频编码程序的设计	373
14.3.3 视频数据网络传输程序的设计	378
14.4 测试结果	383
第 15 章 综合实验	385
15.1 嵌入式开发流程简介	385
15.1.1 嵌入式开发的硬件结构简介	385
15.1.2 嵌入式开发的软件结构简介	386
15.1.3 嵌入式开发模式	386
15.1.4 嵌入式开发流程	387
15.2 项目背景简介	389
15.2.1 MIPv6 基本操作简介	389
15.2.2 新 IPv6 协议, 报文类型和目的选项	390
15.2.3 移动 IPv6 安全概述	402

15.3	实现代码分析	408
15.3.1	数据更新维护模块	410
15.3.2	解析和构造报文	411
15.3.3	移动检测模块	421
15.3.4	移动报文收发与路由优化	423
15.3.5	绑定模块	425
15.4	嵌入式开发环境搭建	430
15.4.1	拓扑环境的介绍	430
15.4.2	宿主机的安装及配置	431
15.4.3	μ -Boot 的启动	435
15.4.4	MIPv6-1.1 的移植	437
15.5	代码的调试	439
15.5.1	在宿主机生成目标板能够利用的 FS (文件系统)	439
15.5.2	minicom 的使用	440
15.5.3	镜像的下载	440
15.5.4	查看启动信息并修改 bug	440
15.6	系统测试	443
15.6.1	软件测试简介	443
15.6.2	嵌入式系统测试简介	445
15.6.3	协议一致性测试	447
参考文献		451

嵌入式 Linux 系统简介

嵌入式技术最初应用于军事和工业控制领域,随着微电子技术和软件技术的发展,嵌入式系统在通信、办公自动化和消费电子领域也获得了广泛应用,日益引起了工业界和学术界的重视。

本章将介绍嵌入式开发所需要的软/硬件基础知识,首先介绍嵌入式 Linux 操作系统的特点;然后介绍市场上流行的主流嵌入式处理器;最后介绍 Linux 操作系统下开发所需要的工具,如 Shell、常用调试工具等。

1.1 嵌入式系统

1.1.1 概述

嵌入式系统(Embedded Systems, ES)是根据应用的要求,将操作系统和功能软件集成于计算机硬件系统中,从而实现软件与硬件一体化的计算机系统。嵌入式系统出现于 20 世纪 60 年代末期,它最初被用于控制机电、电话交换机,如今已被广泛应用于工业制造、过程控制、通信、仪器、仪表、汽车、船舶、航空、航天、军事装备、消费类产品等众多领域。嵌入式系统在数量上远远超过了各种通用计算机系统:计算机系统核心 CPU,每年在全球范围内的产量大概在 20 亿个左右,其中 80% 以上应用于各类专用性很强的嵌入式系统中。

一般来说,凡是带有微处理器的专用软/硬件系统都可以称为嵌入式系统。和通用的计算平台相比,嵌入式系统具有功能单一、体积小、功耗低、可靠性高、剪裁性好、软/硬件集成度高、计算能力相对较低等特点。多年来,嵌入式设备中没有操作系统,其主要原因有两点:一是诸如洗衣机、微波炉、电冰箱这样的设备仅需要一道简单的控制程序,以管理数量有限的按钮和指示灯,没有使用操作系统的必要;二是它往往只具有有限的硬件资源,不足以支持一个操作系统。

然而,随着硬件的发展,嵌入式系统变得越来越复杂,最初的控制程序中逐步加入了许多功能,而这些功能中有很多可以由操作系统提供。于是,在 20 世纪 70 年代末期出现了嵌入式操作系统(Embedded Operating Systems),它的出现极大地简化了应用程序的设计,并可以有效地保障软件质量和缩短开发周期。简单的 ES 一般并不使用操作系统,只包含一些控制流程,但是随着嵌入式操作系统日益复杂,简单的流程控制并不能满足系统的需求,这时就必须考虑使用操作系统作为系统软件。因此,嵌入式操作系统就应运而生了。

1.1.2 嵌入式系统的发展历史

事实上,在很早以前,嵌入式这个概念就已经存在了。在通信方面,嵌入式系统在 20 世纪 60 年代就用于对电子机械电话交换的控制,当时被称为“存储式程序控制系统”(Stored Program Control)。

嵌入式计算机的真正发展是在微处理器问世之后。1971 年 11 月,Intel 公司成功地把算术运算器和控制器电路集成在一起,推出了第一款微处理器 Intel 4004,其后各厂家陆续推出了许多 8 位、16 位的微处理器,包括 Intel 8080/8085、8086, Motorola 的 6800、68000,以及 Zilog 的 Z80、Z8000 等。以这些微处理器作为核心所构成的系统,广泛地应用于仪器仪表、医疗设备、机器人、家用电器等领域。微处理器的广泛应用形成了一个广阔的嵌入式应用市场,计算机厂家开始大量地以插件方式向用户提供 OEM 产品,再由用户根据自己的需要选择一套适合的 CPU 板、存储器板以及各式 I/O 插件板,从而构成专用的嵌入式计算机系统,并将其嵌入到自己的系统设备中。

为灵活兼容考虑,出现了系列化、模块化的单板机。流行的单板计算机有 Intel 公司的 iSBC 系列、Zilog 公司的 MCB 等。后来人们可以不必从选择芯片开始来设计一台专用的嵌入式计算机,而是只要选择各功能模块,就能够组建一台专用计算机系统。用户和开发者都希望从不同的厂家选购最适合的 OEM 产品,插入外购或自制的机箱中就形成新的系统,这样就希望插件是互相兼容的,也就导致了工业控制微机系统总线的诞生。1976 年 Intel 公司推出 Multibus,1983 年扩展为带宽达 40MB/s 的 Multibus II。1978 年由 Prolog 设计的简单 STD 总线广泛应用于小型嵌入式系统中。

20 世纪 80 年代可以说是各种总线层出不穷、群雄并起的时代。随着微电子工艺水平的提高,集成电路制造商开始把嵌入式应用中所需要的微处理器、I/O 接口、A/D、D/A 转换、串行接口以及 RAM、ROM 等部件全部集成到一个 VLSI 中,从而制造出面向 I/O 设计的微控制器,也就是俗称的单片机,成为嵌入式计算机系统异军突起的一支新秀。其后发展的 DSP 产品则进一步提升了嵌入式计算机系统的技术水平,并迅速地渗入到消费电子、医用电子、智能控制、通信电子、仪器仪表、交通运输等各种领域。

20 世纪 90 年代,在分布控制、柔性制造、数字化通信和信息家电等巨大需求的牵引下,嵌入式系统进一步加速发展。面向实时信号处理算法的 DSP 产品向着高速、高精度、低功耗方向发展。Texas 推出的第三代 DSP 芯片 TMS320C30,引导着微控制器向 32 位高速智能化方向发展。在应用方面,掌上电脑、手持 PC 机、机顶盒技术相对成熟,发展也较迅速。特别是掌上电脑,1997 年在美国市场上掌上电脑不过四五个品牌,而 1998 年底,各式各样的掌上电脑如雨后春笋般纷纷涌现出来。此外, Nokia 推出了智能电话,西门子推出了机顶盒, Wyse 推出了智能终端, NS 推出了 WebPAD。装载在汽车上的小型电脑,不但可以控制汽车内的各种设备(如音响等),还可以与 GPS 连接,从而自动操控汽车。21 世纪无疑是一个网络的时代,使嵌入式计算机系统应用到各类网络中去,也必然是嵌入式系统发展的重要方向。在发展潜力巨大的“信息家电”中,人们非常关注的网络电话设备,即 IP 电话,就是一个代表。该设备可以简单到像普通电话一样,可它却是通过互联网来实现双方通话的,花市话的钱可以打长途电话!

1.1.3 嵌入式 Linux 的特性

由于嵌入式系统应用场合的特殊性，决定了它必须具有以下特性。

- (1) 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。
- (2) 以微电脑与外围器件构成核心，其规模可在大范围内变化，如从 8051 芯片到 x86 芯片。
- (3) 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，甚至酿成灾难。
- (4) 全自动操作循环。

嵌入式系统是电脑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了 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、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、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 月访华时曾说过，4~5 年以后嵌入式智能（电脑）工具将是继 PC 和因特网之后人类最伟大的发明。

1.2 嵌入式 Linux 操作系统简介

1.2.1 Linux 系统

Linux 最初来源于芬兰学生 Linus Torvalds 学习操作系统后的练习，由于它的源码公开，因此得到了开源社区的广泛支持，使得 Linux 操作系统在短时间内发展成为稳定、成熟、实用的操作系统。与其他操作系统相比，Linux 系统具有如下特点。

1. 开放性

开放性是指系统遵循世界标准规范，特别是遵循开放系统互联（OSI）国际标准。凡遵循国际标准所开发的硬件和软件，都能彼此兼容，可方便地实现互联。Linux 操作系统的开发最初就在 FSF（软件基金会组织）的 GPL（GNU Public License）的版本控制之下，Linux 内核的所有源代码都采取了开放源代码的方式。因此，全世界的开发者都可以非常方便地免费获得、修改 Linux 的内核源代码，这也是 Linux 得以快速发展、壮大的原因之一。

2. 设备独立性

设备独立性是指操作系统把所有外部设备统一当作文件来看待，只要安装它们的驱动程序，任何用户都可以像使用文件一样，操纵、使用这些设备，而不必知道它们的具体存在形式。

具有设备独立性的操作系统，通过把每一个外围设备看作一个独立文件来简化增加新设备的工作。当需要增加新设备时，系统管理员就在内核中增加必要的连接。这种连接（也称

作设备驱动程序)保证每次调用设备提供服务时,内核以相同的方式来处理它们。当新的更好的外设被开发并交付给用户时,操作系统允许在这些设备连接到内核后,就能不受限制地立即访问它们。设备独立性的关键在于内核的适应能力。其他操作系统只允许一定数量或一定种类的外部设备连接。而设备独立性的操作系统能够容纳任意种类及任意数量的设备,因为每一个设备都是通过其与内核的专用连接独立进行访问的。

Linux 是具有设备独立性的操作系统,它的内核具有高度适应能力,随着更多程序员加入 Linux 编程,会有更多硬件设备加入到各种 Linux 内核和发行版本中。另外,由于用户可以免费得到 Linux 的内核源代码,因此,用户可以修改内核源代码,以便适应新增加的外围设备。

3. 提供了丰富的网络功能

完善的内置网络是 Linux 的一大特点。Linux 在通信和网络功能方面优于其他操作系统。其他操作系统不包含如此紧密地和内核结合在一起的连接网络的能力,也没有内置这些联网特性的灵活性。而 Linux 为用户提供了完善的、强大的网络功能。

支持 Internet 是其网络功能之一。Linux 免费提供了大量支持 Internet 的软件,Internet 是在 Unix 领域中建立并繁荣起来的,在这方面使用 Linux 是相当方便的,用户能用 Linux 与世界上的其他人通过 Internet 网络进行通信。

文件传输是其网络功能之二。用户能通过一些 Linux 命令完成内部信息或文件的传输。远程访问是其网络功能之三。Linux 不仅允许进行文件和程序的传输,它还为系统管理员和技术人员提供了访问其他系统的窗口。通过远程访问,技术人员能够有效地为多个系统服务,即使那些系统位于相距很远的地方。

4. 良好的可移植性和可裁减性

可移植性是指将操作系统从一个平台转移到另一个平台,使之仍然能按其自身的方式运行的能力。Linux 是一种可移植的操作系统,能够在从微型计算机到大型计算机的任何环境中和任何平台上运行。可移植性为运行 Linux 的不同计算机平台与其他任何机器进行准确而有效的通信提供了手段,不需要另外增加特殊的和昂贵的通信接口。最新的 Linux 内核支持多种体系结构的处理器,包括目前流行的 Intel x86、Motorola/IBM PowerPC、ARM、Compaq Alpha、Sun SPARC 等微处理器架构。目前将 Linux 移植到其他硬件平台的工作都在进行中。

根据 GPL 协议规定, Linux 系统内核代码可以免费获得,并在遵守 GPL 协议的条件下进行修改。这就使得修改 Linux 内核来满足自己的需要成为可能。目前,众多的嵌入式 Linux 系统版本,正是得益于 Linux 的这一特点。

1.2.2 从 Linux 到嵌入式 Linux

嵌入式系统是以应用为中心,以计算机为基础,软硬件可裁剪,适用于系统对功能、可靠性、成本、功耗严格要求的专用计算机系统。与普通计算机相比,嵌入式系统具有以下几个特点。

(1) 嵌入式系统是先进的计算机技术、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。也就是说,嵌入式系统并不像通用计算机那样通用,而是与各行业的特定应用相关联的,因此每个行业的嵌入式系统要实现的功能也不一样。因此,在嵌入式系统上没有哪个公司的操作系统和处理器能够垄断市场。

(2) 嵌入式系统实现的功能单一,通常是面向用户、面向产品、面向特定应用的,一般只实现某行业的特定功能。

(3) 成本低、体积小、功耗低及可靠性高,在运行速度和存储容量上有一定的限制。

(4) 嵌入式系统本身不具备在其上进行进一步开发的能力,必须要在相关的开发平台上开发。

(5) 全自动操作循环,一直重复地执行一个特定的程序。

根据上节所述, Linux 系统具有开放源代码、内核稳定、功能强大等特点,因此,它非常适合作为嵌入式系统的操作系统;但是,由于 Linux 系统并不具有强实时性、体积过于庞大,并不适合于嵌入式系统,这就需要对 Linux 操作系统进行改造,做以下三个方面的工作。

1) Linux 的实时性扩充。从操作系统能否满足实时性要求来区分,可把操作系统分成分时操作系统和实时操作系统。分时操作系统按照相等的时间片调度进程轮流运行,分时操作系统由调度程序自动计算进程的优先级,而不是由用户控制进程的优先级。这样的系统无法实时响应外部异步事件。实时操作系统能够在限定的时间内执行完所规定的功能,并能在限定的时间内对外部的异步事件作出响应。

实时性是嵌入式操作系统的基本要求。由于 Linux 还不是一个真正的实时操作系统,内核不支持事件优先级和抢占实时特性,所以在开发嵌入式 Linux 的过程中,首要问题是扩展 Linux 的实时性能。对 Linux 实时性的扩展可以从两方面进行:向外扩展和向上扩展。向外扩展即从范围上扩展,让实时系统支持的范围更广、设备更多。目前的开发所面向的设备仅限于较简单的有实时要求的串/并口数据采集、浮点数据计算等,而像实时网络这样实时系统的高级应用还需进一步发展。向上扩展是扩充 Linux 内核,从功能上扩充 Linux 的实时处理和控制系统。如嵌入式系统 RT-Linux,它的基本原理是将 Linux 本身的任务以及 Linux 内核本身作为一个优先级最低的任务,而实时任务作为优先级最高的任务,即在实时任务存在的情况下运行实时任务,否则就运行 Linux 本身的任务。实时任务不同于 Linux 普通进程。它是以 Linux 的可装载的内核模块 (Loadable Kernel Module, LKM) 的形式存在的,需要运行实时任务的时候,将这个实时任务的内核模块插入到内核中去,实时任务和 Linux 一般进程之间的通信通过共享内存或者 FIFO 通道来实现。

2) 改变 Linux 内核的体系结构。Linux 的内核体系采用的是 Monolithic。在这种体系结构中,内核的所有部分都集中在一起,而且所有的部件在一起编译连接。这样虽然能使系统的各部分直接沟通,有效地缩短任务之间的切换时间,提高系统的响应速度和 CPU 的利用率,且实时性好,但在系统比较大时体积也比较大,与嵌入式系统容量小、资源有限的特点不符。而另外一种内核体系结构 MicroKernel,在内核中只包括了一些基本的内核功能,如创建和删除任务、任务调度、内存管理和中断处理等部分,而文件系统、网络协议栈等部分都是在用户内存空间运行。这种结构虽然执行效率不如 Monolithic 内核,但大幅度减小了内核的体积,同时也方便了整个系统的升级、维护和移植,更能满足嵌入式系统的特点。为此,要使嵌入式 Linux 的应用更加广泛,若将 Linux 目前的 Monolithic 内核结构中的部分结构改造成 MicroKernel 体系结构,可使得到的 Linux 既具有很好的实时性,又能满足嵌入式系统体积小的要求。

另外, Linux 是一个需要占用存储器的操作系统。虽然这可以通过减少一些不必要的功能来弥补,但可能会浪费很多时间,而且容易带来很大的麻烦。许多 Linux 的应用程序都要

用到虚拟内存，这在许多嵌入式系统中是没有价值的。所以，并不是一个没有磁盘的 Linux 嵌入式系统就可以运行任何 Linux 应用程序。

3) 建立、完善 Linux 的集成开发环境。由于嵌入式操作系统本身资源所限，并不能提供开发功能，这就需要在宿主机上建立交叉编译环境。一个完整的嵌入式系统的集成开发环境一般需要提供的工具是编译/连接器、内核调试/跟踪器和集成图形界面开发平台。其中的集成图形界面开发平台包括编辑器、调试器、软件仿真器和监视器等。在 Linux 系统中，具有功能强大的 GCC 编译器工具链，使用了基于 GNU 的调试器 GDB 的远程调试功能，一般由一台客户机运行调试程序调试宿主机运行的操作系统内核。

Linux 系统经过实时化、嵌入式化的改造后，基本上可以满足嵌入式系统的需求，经过交叉编译，可以把它作为操作系统烧到目标板上，在此基础上，可以进一步进行相关应用程序的开发。

1.2.3 常用嵌入式 Linux 系统简介

这里，将介绍几种常见的嵌入式 Linux 操作系统，主要有以下几种。

1. Vxworks

VxWorks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统 (RTOS)，是嵌入式开发环境的关键组成部分。良好的持续发展能力、高性能的内核以及友好的用户开发环境，在嵌入式实时操作系统领域占据一席之地。它以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中，如卫星通信、军事演习、弹道制导、飞机导航等。在美国的 F-16、FA-18 战斗机、B-2 隐形轰炸机和爱国者导弹上，甚至连 1997 年 4 月在火星表面登陆的火星探测器上也使用到了 VxWorks。

2. RT-Linux

这是由美国墨西哥理工学院开发的嵌入式 Linux 操作系统。到目前为止，RT-Linux 已经成功地应用于航天飞机的空间数据采集、科学仪器测控和电影特技图像处理等领域。RT-Linux 开发者并没有针对实时操作系统的特性而重写 Linux 的内核，因为这样做的工作量非常大，而且要保证兼容性也非常困难。为此，RT-Linux 提出了精巧的内核，并把标准的 Linux 核心作为实时核心的一个进程，同用户的实时进程一起调度。这样对 Linux 内核的改动非常小，并且充分利用了 Linux 下现有的丰富的软件资源。

3. μ CLinux

μ CLinux 是 Lineo 公司的主要产品，同时也是开放源码的嵌入式 Linux 的典范之作。 μ CLinux 主要是针对目标处理器没有存储管理单元 MMU (Memory Management Unit) 的嵌入式系统而设计的。它已经被成功地移植到了很多平台上。由于没有 MMU，其多任务的实现需要一定的技巧。 μ CLinux 是一种优秀的嵌入式 Linux 版本，是 micro-Contral-Linux 的缩写。它秉承了标准 Linux 的优良特性，经过各方面的小型化改造，形成了一个高度优化的、代码紧凑的嵌入式 Linux。虽然它的体积很小，却仍然保留了 Linux 的大多数的优点：稳定、良好的移植性、优秀的网络功能、对各种文件系统完备的支持和标准丰富的 API。它专为嵌入式系统做了许多小型化的工作，目前已支持多款 CPU。其编译后目标文件可控制在几百千字节数量级，并已经被成功地移植到很多平台上。