

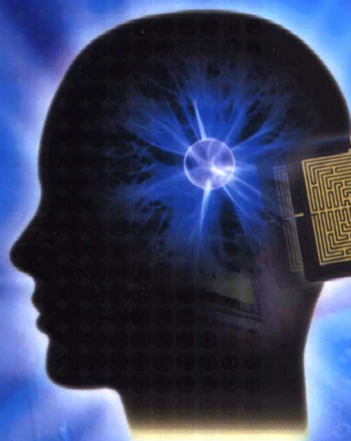
科 技 新 贵 >> 新一代嵌入式系统软件平台技术



Embedded Linux

嵌入式系统原理与实务

陈俊宏 著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Embedded Linux

嵌入式系统原理与实务

陈俊宏 著

中国铁道出版社

2004 • 北京

MJS115/03

北京市版权局著作权合同登记号：01-2004-1385 号

版 权 声 明

本书中文繁体字版由台湾学贯行销股份有限公司出版。本书中文简体字版经台湾学贯行销股份有限公司授权由中国铁道出版社出版。任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

本书贴有学贯行销激光防伪标签，无标签者不得销售。版权所有，侵权必究。

图书在版编目（CIP）数据

Embedded Linux 嵌入式系统原理与实务/陈俊宏著. —北京：中国铁道出版社，2004.3

ISBN 7-113-05784-5

I. E… II. 陈… III. Linux 操作系统 IV. TP316.89

中国版本图书馆 CIP 数据核字(2004)第 016382 号

书 名：Embedded Linux 嵌入式系统原理与实务

作 者：陈俊宏

出版发行：中国铁道出版社（100054，北京市宣武区右安门西街8号）

策划编辑：严晓舟 郭毅鹏

责任编辑：苏茜 孙东川 翟玉峰

封面设计：白雪

印 刷：北京兴顺印刷厂

开 本：787×1092 1/18 印张：21 字数：402 千

版 本：2004 年 4 月第 1 版 2004 年 4 月第 1 次印刷

印 数：1~5000 册

书 号：ISBN 7-113-05784-5/TP·1149

定 价：38.00 元

版权所有 侵权必究

凡购买铁道版的图书，如有缺页、倒页、脱页者，请与本社计算机图书批销部调换。

出版说明

Embedded Linux 虽然是一个相当年轻的名词，但在嵌入式系统的市场上却是相当重要的技术，是一种有关嵌入式系统的软件解决方案。本书在详细介绍了 Embedded Linux 的基本概念后，讲述了如何建立最小化的 Linux 平台及研究 Embedded Linux 所应要学习的知识和 IBM IA32 (x86) 的架构，最后通过完整的嵌入式影像 (video4linux) 范例的介绍，讲述如何设计热门的影像应用程序。本书的出版目的是想帮助已经接触过 Linux 的读者了解并认识 Embedded Linux 这个新一代的嵌入式系统软件技术。

随书所附光盘中 linux 文件夹的内容为书中相关范例，embedded linux 文件夹中提供了 peeweelinux、SDL、microwin 软件包。

本书的专属网站：<http://www.jollen.org>。

本书由学贯行销股份有限公司提供版权，经中国铁道出版社计算机图书中心审选，由陆正武改版，陆正中审校，在改版过程中，刘玉敏、张勇、安冀苗参与讨论并给出了宝贵的意见。

中国铁道出版社

2004 年 2 月

目 录

第 1 章 Embedded Linux 系统架构.....	1
1-1 Embedded Linux 技术介绍	2
1-2 Embedded Linux 技术架构	3
1-3 Embedded Linux 的软件支持	6
Linux kernel.....	6
Linux distributions.....	6
Linux system	7
Linux kernel 的技术面	7
GPL 版权.....	7
System Shared Libraries	8
Applications.....	9
1-4 Embedded Linux 的硬件支持	10
System-on-Chip.....	10
Single Board Computer	10
处理器的选择	11
1-5 Embedded Linux Distributions.....	11
热门的 Embedded Linux Distributions	13
Real-Time Linux 解决方案	14
第 2 章 Embedded Linux 开发环境.....	15
2-1 使用自由软件目录	16
自由软件蓝图	16
2-2 使用自由软件: 软件开发	17
简介.....	17
编译器家族——GCC.....	17
程序调试工具——GDB.....	17
System API 调试工具——strace	18
跨平台辅助工具组	18
编译自动化——GNU Make.....	19
版本控制——CVS	19
解释式程序语言	19
2-3 使用自由软件: 程序库	21

System Calls	21
标准程序库——GLIBC	21
Linux 常用的程序库	23
2-4 使用 GCC 编译程序	24
基本用法	24
编译但不链接: -c 参数	24
定义常数	25
使用非标准头文件 (headers)	26
指定与程序库 (library) 链接	26
与 static library (静态程序库) 链接	26
Ld - Linux 下的 Linker	27
程序最佳化	27
2-5 使用 GDB 进行调试	28
让 gcc 产生符号表	29
gdb 的基本用法	29
断点调试 (breakpoints)	31
强大的 print 命令	33
单步执行	35
底层调试	36
2-6 LSB 标准	41
LSB 标准介绍	41
LSB 标准摘要	42
LSB Common 内容摘要	42
LSB IA32 内容摘要	48
2-7 FHS 标准	50
什么是 FHS	50
FHS 内容架构	51
Root Filesystem	51
/usr 目录	54
/var 目录	56
针对 Linux 的特别内容	58
第3章 Embedded Linux 系统建立	63
3-1 如何打造 Embedded Linux 系统平台	64
Embedded Linux 系统构成要素	64

Embedded Linux 的建立策略	64
应选择何种建立策略	65
3-2 安装 PeeWeeLinux	66
PeeWeeLinux 的特点	66
如何取得 PeeWeeLinux	67
安装 PeeWeeLinux	67
3-3 建立 Embedded Linux 系统平台	68
3-4 利用 PeeWeeLinux 集成 RPM 包	74
3-5 更进一步的 Embedded Linux 系统调整	78
更新与修改 *.tar 包文件	79
PeeWeeLinux 功成身退	79
更进一步的 Embedded Linux 最小化	80
3-6 Embedded Linux+Java 技术	82
信息家电设计-HAVi	84
3-7 Embedded GUI/Windowing	86
Microwindows 与解决方案组合	87
第 4 章 Embedded Linux 程序设计	91
4-1 Process	92
Process ID	92
什么是 Process	93
建立 Process	93
UNIX 下的 Spawn 写法	96
终止 Process	97
等待 Child Process	97
Zombie Process	99
Signal 的处理	100
Linux 的 Signal	101
4-2 Thread	102
Linux Thread 程序设计基本概念	103
Joinable Thread 与 Detached Thread	103
建立 Thread	103
建立 Detached Thread	106
Join Thread	107
Join Thread 返回值	112

4-3	Linux Kernel Module	113
	Linux Module System	115
	Kernel Symbol Table	115
	由应用程序进入 Kernel Module 程序设计	116
	Hello, World!	117
	Printk()函数	117
	编译 Linux Kernel Module	119
	module_init 与 module_exit 宏	121
	其他常用的宏	121
	更多有关 Linux Kernel Module	123
4-4	Process Synchronization	123
	Race Condition	124
	Critical Section	125
	Critical-Section Problem	126
	Semaphores	126
第 5 章	Linux IPC 程序设计	135
5-1	Message-Passing System	136
	Message-Passing System 介绍	136
	persistent communication (持续性通信)	136
	transient communication (传递性通信)	136
	asynchronous communication (异步通信)	137
	synchronous communication (同步通信)	137
	Communication Link (通信链路)	137
	消息 (Message) 的特性	138
	实例介绍: UDP	139
5-2	Shared Memory	139
	Shared Memory	139
	读取 Shared Memory 的范例	142
5-3	Mapped Memory	145
5-4	Pipes 与 FIFOs	148
	Pipes	148
	FIFOs	152
5-5	Socket	153
	Unix-domain socket 与 Internet-domain socket	153

设计 Server 程序.....	154
设计 Client 程序.....	156
程序范例.....	156
第 6 章 80x86 与 Linux 内存管理.....	165
6-1 Memory Hierarchy.....	166
内存管理 (Memory Management)	167
6-2 80x86 的内存管理.....	168
什么是 i386.....	168
Real Mode 与 Protected Mode.....	168
Linux 如何进入 Protected Mode.....	169
Protected Mode 的内存管理.....	171
6-3 操作系统的内存管理.....	171
Main Memory 的基本管理方法.....	172
Main Memory 不足时的管理方法.....	175
利用 Paging 技术来管理 Main Memory.....	176
利用 Virtual Memory 管理 Secondary Storage.....	180
第 7 章 video4linux 嵌入式影像系统.....	183
7-1 video4linux 介绍.....	184
Bt878 芯片与 BTTV 驱动程序.....	184
视频采集卡.....	185
安装视频采集卡.....	186
相关技术名词解释.....	187
7-2 video4linux 应用程序介绍.....	189
BTTV 相关应用软件——xawtv.....	189
7-3 video4linux 程序设计: 入门篇.....	192
基本数据结构与函数实现.....	192
设备文件初始化.....	194
配合应用程序来设计.....	199
7-4 video4linux 程序设计: 提高篇.....	203
mmap 的初始化: v4l_mmap_init()实现.....	203
初始化 grab: v4l_grab_init()实现.....	203
影像采集: v4l_grab_frame()的实现.....	205
模仿 flip-flop 采集: device_next_frame()的实现.....	206
EffecTV.....	207

7-5 利用 JRTPLIB 加入网络的功能.....	211
加入 RTP Protocol.....	211
JRTPLIB 简介.....	211
7-6 输出 JPEG 与 PPM 影像.....	213
7-7 完整范例程序代码.....	216
附录 A 安装 Microwindows for X11.....	225
附录 B bootsect.S/setup.S 源代码.....	229
附录 C head.S/misc.c 源代码.....	263
附录 D Framebuffer 设置.....	277
附录 E run level 设置.....	279
附录 F KDE 最小化文件清单.....	285

1

Embedded Linux 系统架构

1-1 Embedded Linux 技术介绍

Embedded Linux 是一个相当年轻的名词, 虽然如此, 但是 Embedded Linux 在嵌入式系统的市场却是相当重要的一种技术。本书的主旨是介绍 Embedded Linux 的技术、实用的解决方案与应用。

Linux 的开发始于 1991 年, 当时一位芬兰的学生 Linus Torvalds 在芬兰最大的 FTP 站开了一个目录, 叫做 Linux (Linus' UNIX), 他将作品放到网络上与大家分享。当时 Linus 所上传的 Linux 只是一个非常小型的操作系统实现, 也就是 Linux kernel 的雏形, 因此第一个版本的 Linux kernel 就在 1991 年 5 月 14 日正式公开于网络上, 版本编号为 0.01。

在之后的 10 年里, 由于允许自由拷贝、流传与修改的版权声明 (GPL), 因此吸引了无数的程序爱好者为 Linux kernel 贡献程序代码与驱动程序, 到了 Linux kernel 2.0 问世后, Linux kernel 已经具备完整而且功能强大的 UNIX 特点, 至此 Linux 已正式成为名符其实的 UNIX-like system, 同时 Linux kernel 2.0 也是 Linux 开发史上的重大里程碑。目前, Linux kernel 的开发已经进入 2.4 (stable) 与 2.5 (beta) 的时代了。

一个容易应用的 OS 核心必须要有快速的更新与错误修正机制, 而 Linux 闻名的“市场化”开发模式不但没有让 Linux 的开发迟缓, 反倒是让 Linux 的核心开发一日千里; Linux 的文件计划与核心支持运行得相当积极, 这也是让业界认同的主要因素。

“教堂观与市集观”是 1998 年 1 月由 Eric S. Raymond 所提出的论点, 内容是讨论了教堂与市集 2 种不同的软件开发模式:

1. 教堂模式: 由少数人主导, 用户与研发界壁垒分明的开发模式, 例如传统的商业软件公司。
2. 市集模式: 众多兼职的程序员共同开发软件的模式, 例如 Linux 的开发。

在本章接下来的主题里, 首先在 1-2 节将会介绍 Embedded Linux 的技术架构, 本书的内容组织就是由这样的架构所沿伸出去; 接着 1-3 节则是根据 1-2 节的架构讨论软件支持的问题、1-4 节则是硬件支持的问题。接着 1-5 节则会讨论关于 Embedded Linux 包的选择策略、并且会列出几个较普遍的特定用途 Embedded Linux 包, 供读者实现时参考选用。

Linux kernel 是一个强大、可信赖、具有可伸缩性与扩充性的操作系统核心 (OS kernel), Linux kernel 实现了许多现代化操作系统的理论, 并且支持完整的硬件驱动程序、网络通信协议与多处理器的架构, 这对于一个嵌入式操作系统核心而言, 是非常重要的选择依据。

Linux 的应用软件支持相当多, 并且也有图形化的环境; 在程序开发方面, 编译器、调试工具与程序库支持比较完整, 因此 Linux 无疑是一个优秀的开发平台。同时, Linux 的开放源码开发模式, 让这个软件平台更开放, 这使得我们可以很容易掌握所有的技术细节, 有助于系统的研发。

目前已经有许多嵌入式设备使用 Embedded Linux 的技术, 包括: PDA、WebPAD、Cell-phones、IP-phones、AV entertainment devices、gateway、server、wireless access point、network camera 等。

1-2 Embedded Linux 技术架构

Embedded Linux 是 Linux 的一个应用, Embedded Linux 的技术在整个嵌入式系统领域里只占了很小的部分, 对于 Embedded Linux 而言, 我们会将研究工作放在两个主要的方向: 软件支持与硬件支持。对软件而言, 我们要针对应用程序执行环境需求做系统集成; 对硬件而言, 我们要针对不同计算机架构做内核配置与编译。

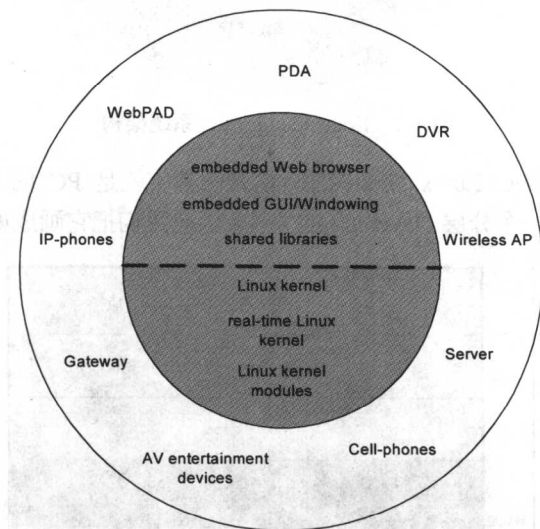


图 1-1 Embedded Linux 系统开发

利用 Embedded Linux 技术开发嵌入式设备的最大好处是, 我们可以将整个开发重心从硬件与汇编程序转移到应用软件上。目前网络上相当多的 Embedded Linux 资源可供我们使用, 包括: Embedded Linux 包集成项目、移植 (porting) 项目、Real-Time Linux kernel 项目等。

Embedded Linux 是一个很好的嵌入式系统开发原型实现平台, 很多时候人们

Embedded Linux 嵌入式系统原理与实务

都已经认为小的 Linux 就是 Embedded Linux 的应用,事实上这真的就是 Embedded Linux 的本质——小型的 Linux (GNU/Linux) 系统。所以读者会发现, Embedded Linux 的架构和 PC 上的 Linux 架构根本就是如出一辙,其实,现在有很多的 Embedded Linux 产品都是纯软件产品,例如:可装在 1.44 MB 软盘上的 firewall 解决方案。

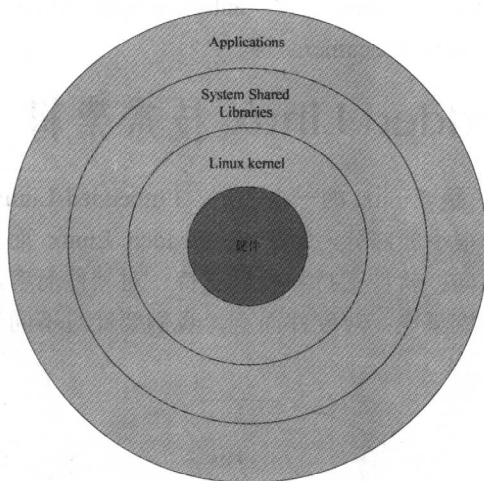


图 1-2 Embedded Linux 系统架构

图 1-2 是 Embedded Linux 的架构图,这张图其实就是 PC 的 Linux 架构图(桌面型 Linux)。这是一个分层(layered)的架构,因此我们把它画成如图 1-3 的方块图。

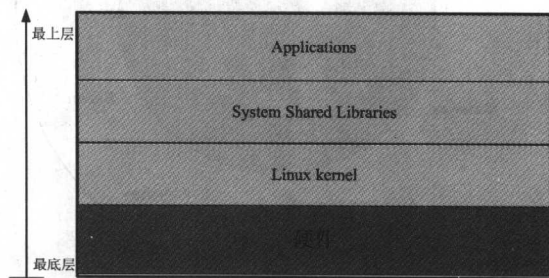


图 1-3 Embedded Linux 系统架构 II

在图 1-3 里,最底层是硬件部分, Linux kernel、system shared libraries、applications 我们都把它们归类到软件部分。后面 1-3 与 1-4 节将分别就软件与硬件两大部分进行说明。

由图 1-3 我们可以清楚地了解, Embedded Linux 是基于特定硬件平台的应用程序(applications)执行解决方案。

根据 William E. Peisel 于 Whitepaper: Embedding Linux 一文对 Embedded Linux 的看法, 我们可以将 Embedded Linux 定位为:

A software platform for embedded systems and devices.

因此研究 Embedded Linux 的起点为选择一套适合硬件平台的 Embedded Linux distributions, 当然了解 Embedded Linux 的定位与架构是必要的。

本书的实验平台

我们只要利用自己的主板与 Intel 80386 以上的 CPU (i386 架构), 即可练习 Embedded Linux 的实现。并且我们会将 Embedded Linux 写到 flash memory (闪存) 里, 因此比较适合解决方案如下:

1. DOM

使用 DOM (Disk-on-Module), 因为 DOM 上有 ATA IDE 的控制芯片, 所以我们可以直接将 DOM 安装在主板上的 IDE 插槽, 并且将 DOM 当成容量很小的硬盘来使用。一般 DOM 容量从 8MB~512MB 不等, 当然也有更大的 DOM, 其中以 32MB/64MB/128MB 的规格较适合 Embedded Linux 的初学者使用。

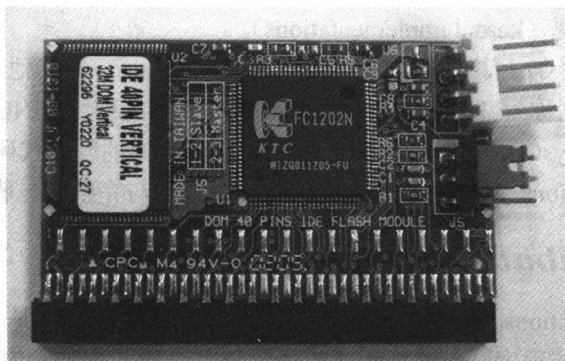


图 1-4 笔者使用的 32MB Disk-on-Module

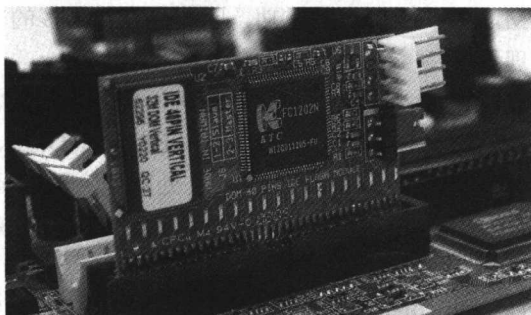


图 1-5 DOM 的安装方式 (直接安插于 IDE 插槽)

2. CF

使用 CF (CompactFlash) 的话, 则是需要搭配卡片阅读机 (Reader) 来使用, 大部分常见的卡片阅读机都是 USB 接口。不过, 如果您的主板不支持 USB 开机启动的话, 只要去买一张 CF 的转接卡 (CF 转 IDE) 来用即可。

1-3 Embedded Linux 的软件支持

在 Embedded Linux 的架构里, 属于软件支持的部分包括: Linux kernel、system shared libraries 与 applications, 并且重点为 Linux kernel 与 applications 方面的研究。对于 system shared libraries 我们只关心程序的开发环境, 而对于 Embedded Linux 的开发应将重点放在 Linux kernel 与 applications 方面。

Linux kernel

Linux 是一个操作系统内核的实现, 我们比较明确的称之为 Linux kernel, 另外, 本书采用与大多数的文件相同的习惯, 当直接称呼 Linux 时, 指的是 Linux system (或 GNU/Linux system), 当书上出现 Linux kernel 时, 才是明确指明操作系统内核的实现 (kernel implementations)。

Linux kernel 加上其他必要的系统工具 (utilities) 与其他项目程序代码的 Linux 操作系统则称为 Linux system, 一般认为标准的称为 GNU/Linux (system), 这是因为 Linux system 使用的系统工具大多是 GNU 的程序代码; 目前广为发布的 Linux distributions 则是基于 GNU/Linux 更成熟与完整的包收集。

Linux distributions

Linux distributions 也称为 Linux collections, 顾名思义, 是 GNU/Linux、包 (packages)、安装程序、包管理程序等的收集与集成, 安装程序与包管理程序会因不同的 Linux distributions 而有所不同, 例如相当受欢迎的包管理程序之一为 RedHat 开发的 RPM, 由于 RPM 的成功, 因此也有相当多的 Linux distributions 也采用 RPM 包管理程序。

不同 Linux distribution 所收录的包也会有所不同, 但一般常见的包在大多数的 Linux distributions 都会被收录, 例如: X、KDE、GNOME、文字处理与编辑软件等。另外一个 Linux distributions 的重要之处则是实现 Linux 的标准, 众多 Linux distributions 的开发都必须跟随一些有关 Linux 的标准进行, 例如: 标准系统工具的收录、标准的层次结构式文件系统等。

标准的实施对于 Linux distributions 而言是相当重要的, 例如: FHS 标准就制定了程序库 (libraries) 的存放目录标准, 这使得应用程序可以在各种符合 FHS 标

准的 Linux distributions 里正常运行。

Linux system

Linux system 的核心为 Linux kernel, 并且包括许多系统程序、工具、图形系统等。Linux system 的组成与架构如下:

- Linux kernel
- System Shared Libraries
- Applications (utilities、shells、compilers、...)

其架构如图 1-3 的 Linux kernel、System Shared Libraries 与 Applications 部分, 是一个分层 (layer) 的关系。

在 Applications 方面, 包括了 utilities、shells、compilers、management programs、tools、GUI 等, 其中也包括了几个有名的项目, 例如: MIT 的 X Window System、Free Software Foundations 的 GNU 项目。

由于 Linux 使用到相当多 GNU 项目的成果, 因此才会有人提倡 GNU/Linux 的标准称呼。

Linux kernel 的技术面

Linux 是 Unix 兼容的 UNIX-like 操作系统, 设计上则依据 POSIX 与 Single UNIX Specification。完整 Unix 操作系统设计应有的特点 Linux 也都有, 包括:

- multitasking
- virtual memory
- shared libraries
- demand loading
- shared copy-on-write executables
- proper memory management
- TCP/IP networking

Linux 已经被移植到其他平台, 除了可在 x86 架构下执行, 在 SPARC、PowerPC、MIPS 等计算机中都可安装。

GPL 版权

Linux kernel 与 FSF 的 GNU 项目成果都是以 GPL (General Public License) 的版权声明所发布。

GPL 是 FSF 自行订定的条文, 当初是由 Richard Stallman 和几位法律专家所拟定, GPL 的主要目的是要捍卫 Free Software Foundation 的精神, 其中有 3 个重点: