

<<嵌入式Linux系统开发教程>>

图书基本信息

书名：<<嵌入式Linux系统开发教程>>

13位ISBN编号：9787302225201

10位ISBN编号：7302225206

出版时间：2010-5

出版时间：清华大学

作者：贺丹丹//张帆//刘峰

页数：360

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<嵌入式Linux系统开发教程>>

前言

嵌入式Linux系统由于具有开源、网络功能强大、内核稳定、高效等特性，在产品开发周期、产品的功能可扩充性、开发时的人力投入等方面都具有显著的优势，因此广泛应用于高、中、低端智能电子设备中。

而它与ARM的结合，更是一种主流的解决方案。

嵌入式Linux+ARM已经广泛应用于机顶盒、掌上电脑、MPC（多媒体个人计算机）、网络设备、工业控制等领域，并且具有良好的市场前景。

嵌入式系统是以应用为中心，以计算机技术为基础，采用可裁剪软硬件，适用于对功能、可靠性、成本、体积、功耗等要求严格的专用计算机系统。

在新兴的嵌入式系统产品中，常见的有MP3、智能手机、GPS、机顶盒、嵌入式服务器、家庭游戏网关、VoIP、PDA、数字视讯录像机及瘦客户机等。

嵌入式系统是未来生活的一个基础平台，将会大大影响人们的生活方式。

本书系统讲解了嵌入式Linux开发流程中的各个步骤，详细解析各个流程中的疑点、难点。

本书分3个部分，共12章：第一部分为基础知识篇，主要讲解嵌入式系统与Linux相关的基础知识，其中第1章为嵌入式系统基础，主要讲解嵌入式相关的概念、历史、应用及前景；第2章为Linux概论，主要是与Linux的基础知识相关，如Linux桌面系统、Linux常用软件的使用及Linux常见命令等；第3章为ARM体系架构，主要是介绍ARM架构的相关知识，如ARM指令集、ARM处理器基本原理等；第4章为嵌入式编程，简明介绍了嵌入式汇编语言及C语言的编程基础。

第二部分为开发入门篇，主要介绍嵌入式开发的基本方法，这部分是本书的重点，也是嵌入式Linux学习的难点，读者要认真学习。

这部分共5章，其中第5章介绍嵌入式开发的软硬件环境，如工具的驱动程序安装、Ubuntu的安装、DNW的使用、NFS的配置和使用，以及Telnet、ftp等的配置使用；第6章主要讲解交叉编译环境的概念以及工具链的编译、获取；第7章介绍boot loader及典型引导程序的制作，如vivi；第8章讲解内核的定制；第9章介绍嵌入式Linux文件系统，这部分内容较多，希望读者重点掌握。

第三部分是提高篇，主要包括第10章驱动程序的开发、第11章嵌入式Linux的图形设计；第12章给出了一个开发实例，使读者能系统地了解嵌入式Linux的开发过程。

<<嵌入式Linux系统开发教程>>

内容概要

本书系统论述了在Linux环境下开发嵌入式系统的设计思想、设计方法及开发流程，通过实例与设计项目，帮助读者尽快掌握嵌入式系统的基本概念，提高嵌入式设计技能。

本书共12章，内容包括嵌入式基础知识、Linux概述、ARM体系架构、嵌入式编程、交叉工具链、bootloader、定制内核、文件系统、驱动程序开发基础、嵌入式图形设计等。

本书的最后给出了一个综合实例，帮助读者理解嵌入式Linux的开发方法和技巧。

本书可作为高校计算机、通信、电子专业相关课程的教材，也可供广大嵌入式开发人员参考。

<<嵌入式Linux系统开发教程>>

书籍目录

第1章 嵌入式系统基础 1.1 嵌入式系统 1.2 嵌入式处理器 1.3 嵌入式操作系统 1.4 嵌入式系统设计 思考与练习第2章 Linux基础 2.1 Linux简介 2.2 图形操作界面 2.3 Linux的基本命令行操作 2.1 Linux内核 思考与练习第3章 ARM体系架构 3.1 ARM微处理器简介 3.2 ARM微处理器系列 3.3 ARM编程模型 3.4 ARM指令系统 3.5 ARM微处理器的应用选型 思考与练习第4章 嵌入式编程 4.1 ARM汇编语言程序设计 4.2 ARM汇编与C语言编程 4.3 基于Linux的C语言编程 思考与练习第5章 软硬件开发环境第6章 交叉编译工具第7章 Bootloader详解及移植第8章 定制内核移植第9章 嵌入式Linux文件系统第10章 嵌入式Linux驱动程序开发基础第11章 嵌入式Linux图形设计第12章 嵌入式视频监控系统设计案例

<<嵌入式Linux系统开发教程>>

章节摘录

插图：Linux给个人计算机带来了能够与UNIX系统相比的速度、效率和灵活性，使个人计算机所具有潜力得到了充分发挥。

Linux与Windows工作方式存在一些根本的区别，这些区别只有在用户对两者都很熟悉之后才能体会到，但它们却是Linux思想的核心。

1.Linux的应用目标是网络Linux设计定位于网络操作系统，它的设计灵感来自于UNIX操作系统，因此它的命令设计比较简单。

虽然现在已经实现Linux操作系统的图形界面，但仍然没有舍弃文本命令行。

由于纯文本可以非常好地跨越网络进行工作，所以Linux配置文件和数据都以文本为基础。

对于熟悉图形环境的用户来说，使用文本命令行的方式看起来比较原始，但是Linux开发更多地关注它的内在功能而不是表面文章。

即使在纯文本环境中，Linux同样拥有非常先进的网络、脚本和安全性能。

Linux执行一些任务所需要的步骤表面看来令人费解，除非能够真正认识到Linux是期望在网络上与其他Linux系统协同执行这些任务。

该操作系统的自动执行能力很强大，只需要设计批处理文件就可以让系统自动完成非常繁琐的工作任务，Linux的这种能力来源于其文本的本质。

2.可选的GUI目前，许多版本的Linux操作系统具有非常精美的图形界面。

Linux支持高端的图形适配器和显示器，完全胜任与图形相关的工作。

但是，图形环境并没有集成到Linux中，而是运行于系统之上的单独一层。

这意味着用户可以只运行GUI，或者在需要时使用图形窗口运行GUI。

Linux有图形化的管理工具以及日常办公的工具，比如电子邮件、网络浏览器和文档处理工具等。

不过在Linux中，图形化的管理工具通常是控制台（命令行）工具的扩展，也就是说，用图形化工具能够完成的所有工作，用控制台命令行同样能够完成。

而使用图形化的工具并不妨碍用户对配置文件进行手工修改，其实际意义可能并不显而易见，但是如果图形化管理工具中所做的任何工作都可以以命令行的方式完成，这就表示这些工作同样可以使用一个脚本来实现。

脚本化的命令可以成为自动执行的任务。

Linux中的配置文件是可读的文本文件，这与过去的Windows中的INI文件类似，但这与Windows操作系统的注册思路有本质的区别。

每一个应用程序都有自己的配置文件，而通常不与其他配置文件放在一起。

不过大部分配置文件都存放于一个目录树（/ecl）下的单独位置，所以在逻辑上看起来是一起的。

文本文件的配置方式可以不通过特殊的系统工具就可以完成配置文件的备份、检查和编辑工作。

3.文件名扩展Linux不使用文件名扩展来识别文件的类型，这与Windows操作系统不同。

Linux操作系统是根据文件的头内容来识别其类型。

为了提高用户的可读性，Linux仍可以使用文件名扩展，这对Linux系统来说没有任何影响。

不过有一些应用程序，比如Web服务器，可能使用命名约定来识别文件类型，但这只是特定应用程序的需要而不是Linux系统本身的要求。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>