

## <<嵌入式系统软件设计>>

### 图书基本信息

书名：<<嵌入式系统软件设计>>

13位ISBN编号：9787563524969

10位ISBN编号：7563524967

出版时间：2011-1

出版时间：北京邮电大学出版社

作者：谢荣生 编

页数：297

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>

## <<嵌入式系统软件设计>>

### 内容概要

本书基于arm处理器和嵌入式linux开发环境，全面讲述了嵌入式系统软件开发流程及主要内容。

《嵌入式系统软件设计》分为五大部分，共10章。

第一部分包括第1、2章，讲述嵌入式linux开发环境的搭建和嵌入式linux软件开发基础；第二部分包括第3、4、5章，分别讲述嵌入式bootloader、内核和根文件系统的基础理论及其移植；第三部分为第6章，讲述嵌入式系统驱动程序设计；第四部分为嵌入式应用程序设计，包括第7、8、9章，主要讲述基于qt的嵌入式gui设计和嵌入式数据库程序设计；第五部分为第10章，通过两个实际的嵌入式软件开发项目，加深对前述内容的理解，提高综合应用能力。

本书可作为高等院校的计算机、电子类相关专业嵌入式系统相关课程的教科书，也可作为基于arm核嵌入式系统软件开发的工程技术人员的参考资料。

## &lt;&lt;嵌入式系统软件设计&gt;&gt;

## 书籍目录

第1章 嵌入式linux开发环境	1.1 嵌入式交叉开发环境	1.2 嵌入式软件开发生的过程
1.2.1 嵌入式软件的生成	1.2.2 嵌入式软件的调试	1.2.3 嵌入式软件的固化
1.3 嵌入式linux软件开发的主要内容	1.4 构建嵌入式linux开发环境	1.4.1 开发平台linux操作系统的安装
1.4.2 嵌入式交叉编译环境的搭建	1.4.3 主机和目标机的串口通信配置	1.4.4 windows与vmware linux的共享
1.4.5 主机与目标机文件的共享和传输	本章小结	习题
第2章 嵌入式linux开发基础	2.1 linux程序的编译和交叉编译	2.1.1 gcc编译器简介
2.1.2 gcc的执行过程	2.1.3 gcc的基本用法和选项	2.1.4 gcc的错误类型分析
2.2 嵌入式linux动态库和静态库的制作与应用	2.2.1 linux静态库和动态库	2.2.2 静态库的制作和应用
2.2.3 动态库的制作和应用	2.3 makefile基础和应用	2.3.1 makefile基本结构
2.3.2 makefile变量	2.3.3 makefile规则	2.3.4 make使用
2.4 嵌入式linux远程调试	2.4.1 嵌入式linux远程调试概述	2.4.2 gdb简介
2.4.3 gdb远程调试	2.5 嵌入式linux多线程应用程序设计	2.5.1 linux线程概述
2.5.2 线程基本编程	2.5.3 线程的同步与互斥	2.5.4 线程属性
2.6 嵌入式linux下c和汇编的混合编程	2.6.1 混合编程概述	2.6.2 c调用汇编
2.6.3 汇编调用c	2.6.4 c内嵌汇编	2.7 嵌入式linuxsocket网络编程基础
2.7.1 socket简介	2.7.2 socket编程基础	2.7.3 socket api及编程流程
本章小结	习题	第3章 嵌入式bootloader原理及移植
3.1 嵌入式bootloader的基本概念	3.2 嵌入式bootloader的两个阶段	3.2.1 bootloader的stage1
3.2.2 bootloader的stage2	3.3 典型嵌入式bootloader(blob)的分析	3.3.1 blob目录分析
3.3.2 blob的两个阶段代码分析	3.3.3 stan&mdash;ld&mdash;script、rest&mdash;ld-script链接脚本分析	3.4 blob在博创pxa270&mdash;s的移植
本章小结	习题	第4章 基于arm的linux内核分析与移植
4.1 内核移植准备	4.1.1 内核源码的获取	4.1.2 内核源码结构
4.1.3 内核配置方法和内容	4.2 linux内核启动过程分析	4.2.1 启动的第一阶段
4.2.2 启动的第二阶段	4.3 内核源码的移植	4.3.1 配置交叉编译环境
4.3.2 建立内核的基本配置文件	4.3.3 编译内核	4.3.4 增加必要的设备驱动
4.3.5 烧写内核到目标机	4.4 嵌入式linux内核调试技术	本章小结
习题	第5章 嵌入式linux根文件系统及其制作	第6章 嵌入式系统驱动程序设计
第7章 基于qt的嵌入式gui程序设计基础	第8章 qt程序设计核心技术	第9章 嵌入式数据库程序设计
第10章 嵌入式软件综合项目实例	参考文献	

## &lt;&lt;嵌入式系统软件设计&gt;&gt;

## 章节摘录

4.数据包的发送 数据包的发送和接收是实现Linux网络驱动程序中两个最关键的过程，对这两个过程处理的好坏将直接影响到驱动程序的整体运行质量。

图6.7中也很明确地说明了网络数据包的传输过程。

首先在网络设备驱动加载时，通过device域中的init函数指针调用网络设备的初始化函数对设备进行初始化，如果操作成功就可以通过device域中的open函数指针调用网络设备的打开函数打开设备，再通过device域中的建立硬件包头函数指针hard\_header来建立硬件包头信息。

最后通过协议接口层函数dev\_queue\_xmit来调用device域中的hard\_start\_xmit函数指针来完成数据包的发送。

该函数将把存放在套接字缓冲区中的数据发送到物理设备，该缓冲区是由数据结构sk\_buff来表示的。

5.数据包的接收 数据包的接收是通过中断机制来完成的，当有数据到达时，就产生中断信号，网络设备驱动功能层就调用中断处理程序，即数据包接收程序来处理数据包的接收，然后网络协议接口层调用netif\_rx函数把接收到的数据包传输到网络协议的上层进行处理。

6.4.5网络设备驱动的实现模式 实现Linux网络设备驱动功能主要有两种形式，一是通过内核来进行加载，当内核启动的时候，就开始加载网络设备驱动程序，内核启动完成之后，网络驱动功能也随即实现了；二是通过模块加载的形式。

比较两者，第二种形式更加灵活，在此着重对模块加载形式进行讨论。

模块设计是Linux中特有的技术，它使Linux内核功能更容易扩展。

采用模块来设计Linux网络设备驱动程序会很轻松，并且能够形成固定的模式，任何人只要依照这个模式去设计，都能设计出优良的网络驱动程序。

先简要概述一下基于模块加载的网络驱动程序的设计步骤，后面结合具体实例来讲解。

首先通过模块加载命令insmod来把网络设备驱动程序插入到内核之中。

然后insmod将调用init\_module（）函数首先对网络设备的init函数指针初始化，再通过调用register\_netdev（）函数在Linux系统中注册该网络设备，如果成功，再调用init函数指针所指的网络设备初始化函数来对设备初始化，将设备的device数据结构插入到dev\_base链表的末尾。

最后可以通过执行模块卸载命令rmmod来调用网络驱动程序中的clean\_up\_module（）函数来对网络驱动程序模块卸载。

具体实现过程如图6.8所示。

<<嵌入式系统软件设计>>

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>