

### 图书基本信息

书名：<<嵌入式Linux驱动程序开发实例教程>>

13位ISBN编号：9787302260592

10位ISBN编号：7302260591

出版时间：2011-9

出版时间：清华大学出版社

作者：张光建，刘政 编著

页数：175

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>

## 内容概要

本书详细介绍了计算机的嵌入式linux系统的三类驱动程序开发方法。全书共分10章。

前面4章简要介绍linux操作系统的操作、嵌入式linux驱动开发环境的建立方法以及linux驱动程序开发所需要的内核基础。

第5章和第6章介绍字符驱动程序的设计方法，第7章和第8章介绍块设备驱动程序的设计方法，第9章和第10章介绍网络驱动程序的设计方法。

每类驱动程序分别从数据结构、驱动架构、驱动模块设计三方面进行了介绍，每类驱动程序都配置了一个模拟驱动小实例，通过这些小实例，可以深入理解各类驱动程序的架构。

每类驱动程序还配置了一个真实外部设备的驱动程序开发实例。

本书实例丰富，通俗易懂，可作为高等学校计算机科学与技术、软件工程等专业学生学习嵌入式技术的教材，也可以作为计算机相关专业学生学习操作系统的提高教材，还可作为工程技术人员设计linux驱动程序的参考书。

## 书籍目录

## 第1章 设备驱动概述

## 1.1设备驱动程序的作用

## 1.1.1嵌入式系统的硬件组成

## 1.1.2嵌入式系统的软件组成

## 1.1.3驱动程序的作用

## 1.2linux设备驱动

## 1.2.1linux设备驱动程序的分类及特点

## 1.2.2linux设备驱动程序与整个软件系统的关系

## 1.3linux设备驱动程序开发技术的学习方法

## 习题1

## 第2章 linux基本操作

## 2.1linux常用命令

## 2.1.1文件及目录操作命令

## 2.1.2显示命令

## 2.1.3文件压缩及解压命令

## 2.1.4网络命令

## 2.1.5改变文件访问权限的命令

## 2.1.6帮助命令

## 2.1.7安装卸载文件系统命令

## 2.2linux基本编程

## 2.2.1emacs编辑器

## 2.2.2使用gcc编译程序

## 2.2.3make命令

## 2.2.4描述文件

## 习题2

## 第3章 嵌入式linux驱动程序开发环境的建立

## 3.1交叉编译环境的建立

## 3.2超级终端的配置

## 3.3文件共享的配置

## 3.3.1配置防火墙

## 3.3.2配置samba使windows与linux共享

## 3.3.3配置nfs使宿主机linux与目标机linux共享

## 习题3

## 第4章 linux设备驱动内核基础

## 4.1linux设备的表示

## 4.2设备文件系统(devfs)

## 4.3模块

## 4.4i/o端口的访问

## 4.5中断管理

## 4.5.1中断的注册

## 4.5.2中断的释放

## 4.5.3中断处理例程

## 4.5.4中断的禁止和使能

## 4.6设备驱动程序中的并发控制

## 4.6.1信号量的初始化

4.6.2信号量的申请

4.6.3信号量的释放

4.7内核空间和用户空间数据复制

4.8使用printk()函数调试设备驱动程序

习题4

## 第5章 字符设备驱动程序

5.1有关字符设备的数据结构

5.1.1file\_operations结构

5.1.2file结构

5.1.3chrdevs数组

5.2字符设备驱动程序的设计

5.2.1字符设备驱动程序的组成

5.2.2file\_operations结构体变量

5.2.3字符设备驱动程序的加载及卸载函数

5.2.4字符设备驱动程序的接口函数

5.3访问字符设备的系统调用

5.3.1open()函数和create()函数

5.3.2close()函数

5.3.3read()函数

5.3.4write()函数

5.3.5应用举例

5.4内核访问字符设备驱动程序的流程

5.4.1open()系统调用的执行流程

5.4.2read()和write()系统调用的执行流程

5.4.3close()系统调用的执行流程

5.5字符设备驱动程序示例：虚拟字符设备驱动程序

5.5.1虚拟字符设备驱动程序代码

5.5.2测试程序代码

5.5.3虚拟字符设备驱动程序的编译

5.5.4虚拟字符设备驱动程序的测试

习题5

## 第6章 字符设备驱动程序实例：s3c2410 adc驱动程序

6.1s3c2410x的adc概述

6.1.1s3c2410x adc的转换频率及转换时间

6.1.2s3c2410x与a/d转换有关的寄存器

6.2s3c2410x的adc驱动程序设计

6.2.1adc驱动程序需要包含的头文件

6.2.2adc驱动程序的file\_operations结构体变量

6.2.3adc驱动程序的加载函数

6.2.4adc驱动程序的卸载函数

6.2.5adc驱动程序的接口函数

6.2.6adc中断处理函数

6.3adc驱动程序的编译及测试

6.3.1测试程序

6.3.2adc驱动程序的编译

6.3.3adc驱动程序的测试

习题6

## 第7章 块设备驱动程序

### 7.1 有关块设备的数据结构

#### 7.1.1 block\_device\_operations结构

#### 7.1.2 gendisk结构

#### 7.1.3 request\_queue结构

#### 7.1.4 buffer\_head结构

#### 7.1.5 request结构

### 7.2 块设备驱动程序的设计

#### 7.2.1 块设备驱动程序的组成

#### 7.2.2 文件包含与宏定义

#### 7.2.3 block\_device\_operations结构体变量

#### 7.2.4 块设备驱动程序的接口函数

#### 7.2.5 块设备驱动程序的request()函数

#### 7.2.6 块设备驱动的加载函数

#### 7.2.7 块设备驱动的卸载函数

### 7.3 块设备驱动程序示例：虚拟块设备驱动程序

#### 7.3.1 虚拟块设备驱动程序代码

#### 7.3.2 虚拟块设备驱动程序的编译

#### 7.3.3 虚拟块设备驱动程序的测试

#### 习题7

## 第8章 块设备驱动程序实例：sd卡驱动程序

### 8.1 sd卡功能概述

#### 8.1.1 总线协议

#### 8.1.2 sd卡的引脚

#### 8.1.3 sd卡的命令

#### 8.1.4 sd卡的响应

#### 8.1.5 sd卡的寄存器

#### 8.1.6 sd卡的状态及操作模式

### 8.2 s3c2410x sdi接口概述

#### 8.2.1 sdi寄存器

#### 8.2.2 sdi的初始化

#### 8.2.3 sd卡命令的发送

#### 8.2.4 数据的读写

### 8.3 基于s3c2410x的sd卡驱动程序设计

#### 8.3.1 sd卡驱动程序的加载函数

#### 8.3.2 sd卡驱动程序的卸载函数

#### 8.3.3 sd卡驱动程序的接口函数

#### 8.3.4 sd卡驱动程序的request()函数

#### 8.4 sd卡驱动程序的编译及测试

#### 8.4.1 sd卡驱动程序的编译

#### 8.4.2 sd卡驱动程序的测试

#### 习题8

## 第9章 网络设备驱动程序

### 9.1 有关网络设备的数据结构

#### 9.1.1 net\_device结构体

#### 9.1.2 sk\_buff结构体

### 9.2 网络设备驱动程序开发常用的内核函数

- 9.2.1 sk\_buff结构操作函数
- 9.2.2 内存申请和释放函数
- 9.2.3 网络驱动程序注册和解除注册函数
- 9.2.4 以太网设备通用初始化函数
- 9.2.5 发送队列的启动、唤醒及停止函数
- 9.2.6 查询网络设备是否在运行的函数
- 9.2.7 向上层传递数据包的函数
- 9.3 网络设备驱动程序的设计
  - 9.3.1 网络设备驱动程序的组成
  - 9.3.2 网络设备驱动程序的加载函数
  - 9.3.3 网络设备驱动程序的卸载函数
  - 9.3.4 网络设备驱动程序的接口函数
  - 9.3.5 网卡中断处理程序
- 9.4 网络设备驱动程序示例：虚拟网络设备驱动程序
  - 9.4.1 虚拟网络设备驱动程序代码
  - 9.4.2 虚拟网络设备驱动程序的编译
  - 9.4.3 虚拟网络设备驱动程序的测试
- 习题9
- 第10章 网络设备驱动程序实例：ax88796驱动程序
  - 10.1 ax88796芯片与cpu的接口
    - 10.1.1 ax88796与cpu的接口信号
    - 10.1.2 s3c2410 cpu与ax88796接口的信号
    - 10.1.3 s3c2410 cpu与网卡芯片接口相关的寄存器
    - 10.1.4 up-netarm 2410-s中ax88796与s3c2410 cpu的连接
  - 10.2 ax88796 mac核心寄存器
    - 10.2.1 mac核心寄存器概述
    - 10.2.2 常用的mac核心寄存器
  - 10.3 ax88796芯片的缓冲区操作
    - 10.3.1 数据包的接收
    - 10.3.2 数据包的发送
    - 10.3.3 填充数据包到发送缓冲区以及从接收缓冲区环移走数据包
  - 10.4 ax88796驱动程序设计
    - 10.4.1 ax88796.h
    - 10.4.2 ax88796驱动程序的加载函数
    - 10.4.3 ax88796驱动程序的卸载函数
    - 10.4.4 ax88796驱动程序的接口函数
    - 10.4.5 ax88796驱动程序的中断处理程序
  - 10.5 ax88796驱动程序的编译及测试
    - 10.5.1 ax88796网络驱动程序的编译
    - 10.5.2 ax88796驱动程序的测试
- 习题10
- 参考文献

## 章节摘录

版权页：插图：任何一个计算机系统的运行都是系统中软硬件协作的结果。

硬件是底层基础，是所有软件得以运行的平台，代码最终会落实为硬件上的组合逻辑与时序逻辑；软件则实现了具体应用，它按照各种不同的业务需求而设计，满足了用户的需求。

硬件较固定，软件则很灵活，可以适应各种复杂多变的应用。

为了尽可能快速地完成设计，应用软件工程师不想也不关心硬件，而硬件工程师也难有足够的闲暇和能力去顾及软件。

例如，应用软件工程师在调用套接字发送和接收数据的时候，他不关心网卡上的中断、寄存器、存储空间、I/O端口、片选以及其他任何硬件词汇；在使用printf（）函数输出信息的时候，他不用知道底层究竟是怎样把相应的信息输出到屏幕或串口的。

因此，应用软件工程师需要看到一个没有硬件的纯粹的软件世界，硬件必须透明地呈现给他们。

谁来实现硬件对应用软件工程师的隐形？

这个艰巨的任务就落在了驱动工程师的头上。

对设备驱动最通俗的解释就是“驱使硬件设备行动”。

设备驱动与底层硬件直接打交道，按照硬件设备的具体工作方式读写设备寄存器，完成设备的轮询、中断处理、DMA通信，进行物理内存向虚拟内存的映射，最终使通信设备能够收发数据，使显示设备能够显示文字和画面，使存储设备能够记录文件数据。

设备驱动充当了硬件和应用软件之间的纽带，它使得应用软件只需要调用系统软件的应用编程接口（API）就可让硬件去完成要求的工作。

在系统中没有操作系统的情况下，工程师可以根据硬件设备的特点自行定义接口，如对串口定义SerialSend（）、SerialRecv（），对LED定义LightOn（）、LightOff（），以及对FLASH定义FlashWrite（）、FlashRead（）等。

而在有操作系统的情况下，设备驱动的架构则由相应的操作系统定义，驱动工程师必须按照相应的架构设计设备驱动，这样，设备驱动才能良好地整合到操作系统的内核中。

驱动程序沟通着硬件和应用软件，相应地驱动工程师沟通着硬件工程师和应用软件工程师。

随着通信、电子行业的发展，全世界每天都有大量的新芯片被生产，大量的新电路板被设计，因此，也会有大量设备驱动需要开发。

这些设备驱动，或运行在简单的单任务环境中，或运行在VxWorks、Linux、Windows等多任务操作系统环境中，都发挥着不可替代的作用。

编辑推荐

《嵌入式Linux驱动程序开发实例教程》是21世纪高等学校嵌入式系统专业规划教材之一。



#### 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>