<<嵌入式Linux C语言应用程序设计与实 >

图书基本信息

书名：<<嵌入式Linux C语言应用程序设计与实践>>

13位ISBN编号：9787302225515

10位ISBN编号：7302225516

出版时间：2010-6

出版时间：清华大学出版社

作者：熊茂华 等 著

页数：353

前言

在世界范围内，社会经济的发展产生了一些新的需求，这也促进了嵌入式技术的广泛应用。
而中国正在成为世界制造大国，在消费电子、工业应用、军事国防、网络设备等领域都有嵌入式系统的应用，同时嵌入式技术反过来刺激了许多新的应用需求，如信息家电、医疗电子病历、微小型智能武器等领域，嵌入式技术的应用日益广泛，相应地，企业针对嵌入式技术人才的需求也越来越多。
因此近几年来，各高职高专院校也开始增设了嵌入式技术应用专业或方向。
但是，各院校在嵌入式技术应用专业教学过程中面临教材难觅的困境。
针对教材缺乏的情况，我们调研了几十所已开设了“嵌入式技术应用”专业的高职高专院校的嵌入式技术人才培养方案、课程设置、教材建设与开发、学生的学习情况及嵌入式技术应用人才就业现状，通过认真地整理、分析和研讨，编写了这套适于高等职业院校嵌入式技术人才培养规划的系列教材。
具体包括以下7本：　《ARM体系结构与程序设计》　《嵌入式Linux C语言应用程序设计与实践》　《嵌入式操作系统与编程》　《嵌入式C/OS-II实时操作系统及应用编程》　《嵌入式Linux实时操作系统及应用编程》　《嵌入式windows CE应用开发技术》　《ARM 9嵌入式系统设计与开发应用》　本书主要内容如下。

第1章对ARM微处理器、ARM技术的基本概念做一些简单的介绍，从ARM体系结构的发展及技术特征、ARM微处理器的应用领域及特点、ARM微处理器的体系结构、ARM微处理器的应用选型等方面分别进行阐述。

第2章介绍嵌入式Linux的开发环境、Linux开发工具GNU gcc·的使用、GNU make命令和Makefile文件、gdb调试器、嵌入式Linux编辑器vi的使用、Emacs综合编辑器等。

第3章介绍嵌入式C语言程序设计的一些基本概念。
主要内容包括嵌入式C语言预处理伪指令、嵌入式C语言的基本数据类型、嵌入式C语言程序结构、嵌入式C语言函数、嵌入式C语言数组、指针、结构体和联合等。

第4章介绍嵌入式C语言和汇编语言的混合编程、嵌入式Linux静态链接库与动态链接库及嵌入式Linux下程序调试等知识。

内容概要

《嵌入式Linux C语言应用程序设计与实践》详细介绍嵌入式Linux的开发环境与工具软件、嵌入式C语言程序设计基础、嵌入式Linux C高级编程与调试、Linux下的文件的操作和I/O应用、ARM Linux进程与进程调度、ARM Linux多线程开发、ARM Linux网络编程和嵌入式Linux设备驱动程序开发。

《嵌入式Linux C语言应用程序设计与实践》是嵌入式Linux C语言应用程序设计的一本实用指导书，通过案例详细介绍嵌入式Linux C语言应用程序设计，案例中的程序都取自实际的项目，并且对程序有详细的注解。

《嵌入式Linux C语言应用程序设计与实践》深入浅出，既可作为高等职业院校相关课程的教材，也可作为嵌入式系统编程人员的技术参考书。

<<嵌入式Linux C语言应用程序设计与实 >

书籍目录

章节摘录

2.3 GNU make命令和Makefile文件　　Makefile文件描述了目标文件之间的依赖关系，以及指定编译过程中使用的工具。

Makefile主要包含了5个方面：显式规则、隐晦规则、变量定义、文件指示和注释。

（1）显式规则。

显式规则说明了如何生成一个或多个目标文件。

这由Makefile的书 写者明显指出，要生成的文件、文件的依赖文件、生成的命令。

（2）隐晦规则。

由于make有自动推导的功能，所以隐晦的规则可以比较简略地书写Makefile，这是由make所支持的。

（3）变量定义。

在Makefile中可定义一系列的变量，变量一般是字符串，类似 c语言中的宏，当Makefile被执行时，其中的变量都会被扩展到相应的引用位置上。

（4）文件指示。

其包括3个部分，一个是在一个Makefile中引用另一个Makefile，就 像C语言中的include一样；另一个是指根据某些情况指定Makefile中的有效部分，就像 C语言中的预编译#if一样；还可定义一个多行的命令。

（5）注释。

Makefile中只有行注释，和UNIX的Shell脚本一样，其注释是用“ # ”字 符，如果在Makefile中使用“ # ”字符，可以用反斜杠进行转义，如：“ ＼# ”。

最后，还值得一提的是，在Makefile中的命令，必须要以Tab键开始。

Makefile定义了一系列的规则来指定哪些文件需要先编译，哪些文件需要后编译，哪 些文件需要重新编译，甚至于进行更复杂的功能操作。

Makefile就像一个Shell脚本一 样，其中也可以执行操作系统的命令。

Makefile带来的好处就是“ 自动化编译 ”，一旦写 好，只需要一个make命令，整个工程完全自动编译，极大地提高了软件开发的效率。

GNU的make工作时的执行步骤如下。

（1）读入所有的Makefile。

（2）读入被incllMe的其他Makefile。

（3）初始化文件中的变量。

（4）推导隐晦规则，并分析所有规则。

（5）为所有的目标文件创建依赖关系链。

（6）根据依赖关系，决定哪些目标要重新生成。

（7）执行生成命令。

（1）～（5）为第一个阶段，（6）～（7）为第二个阶段。

在第一个阶段中，如果定义的变量 被使用了，那么，make会把其展开在使用的位置。

但make并不会完全马上展开。

make 程序利用Makefile中的数据和每个文件的最后修改时间来确定哪个文件需要更新，对于 需要更新的文件，make程序执行Makefile数据中定义的命令来更新。

2.3.1 Makefile文件的规则

版权说明