

<<Linux嵌入式系统高级程序设计>>

图书基本信息

书名：<<Linux嵌入式系统高级程序设计>>

13位ISBN编号：9787115239372

10位ISBN编号：7115239371

出版时间：20101015

出版时间：人民邮电出版社

作者：Gene Sally

页数：336

译者：郭旭

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>

<<Linux嵌入式系统高级程序设计>>

前言

在大约十年前进入嵌入式Linux领域时，我面对的一个问题是：我是不是应该使用操作系统？选择Linux意味着经常需要将该操作系统移植到目标硬件上运行，并移植所需的工具。随着时间的流逝，事情有了很大的变化，对许多项目来说Linux已经是默认选项，而需要决策的是在具体项目上应该使用该操作系统的哪些特性。

如今的问题是：我应该如何配置我的Linux发布版？

用技术术语来说，这是开发者态度在很短的时间范围内所发生的巨大转变。

在嵌入式领域，Linux是如此普遍，以至于嵌入式处理器和开发板默认情况下都附带了Linux。

买家都期望开发板会启动Linux，并带有随硬件一同提供的、进行嵌入式开发所需的工具。

不同于Linux发展的早期阶段，现在的开发者无需把Linux移植到相应的开发板上，只需要对一个已经能够运行的Linux内核和根文件系统进行配置，使之适合应用的需求即可。

着眼于这样的背景，我以一个用户的视角来撰写本书：他即将开始其项目，而手头已经有一个可以在开发板上运行的Linux发布版。

这不见得是最终随产品交付的Linux发布版，但很可能作为开发的起点。

本书描述了诸如从头开始构建交叉编译器之类的任务，以便读者理解，但读者或许会使用与开发板一同提供的交叉编译器，这样读者就可以集中精力开发实际的应用。

但学习构建和配置用于Linux系统的工具绝非浪费，当读者需要从你的系统中压榨出内存的每一位时，这些就成为基本功了。

更进一步，对于新的片上系统（System on a Chip, SOC）式设计，开发板附带的Linux发布版带有芯片上的设备所需的所有驱动程序。

需要开发驱动程序的情形是比较罕见的。

这意味着，大多数工程师把时间花费在内核的定制上，而不需要构建新的内核组件，而花费在所有内核配置和开发方面的总时间，只是过去几年的几分之一。

随着嵌入式设备中的处理器变得更为强大，开发者逐渐发现可以使用C以外的语言进行开发。

将C++用于开发已经很常见，还有其他高级语言如Python、TCL、Java，甚至Perl或PHP。

对于使用汇编语言开始嵌入式开发的人来说，在嵌入式目标机上使用类似Perl的语言几乎是异端（如果不算是叛教的话）。

但这些高级语言可以大大提高嵌入式开发的效率。

在上市时间至关重要的环境下，高级语言显然会成为主流。

嵌入式项目的开发过程不同于其他的软件项目。

首先会有一个设计过程，这涉及建立一个专用的容器，以及一个用户界面（可能是一个小的LCD显示器和一些按钮）。

但更重要的是软件的部署和更新。

项目的代码不会放到CD或网站上，而必须下载到带有Linux发布版的开发板上。

在软件最初安装后，可能需要将其更新为包含bug修正或其他特性的新版本。

根据系统的配置方式，更新系统本身可能会成为另一个项目。

如果读者是在工作中开始一个嵌入式项目，或购买了某种面向计算机业余爱好者的比较廉价的开发板，我预祝你的努力能够获得好运。

与以前任何时候相比，目前在嵌入式项目中使用Linux都更为容易，且有更多乐趣。

<<Linux嵌入式系统高级程序设计>>

内容概要

时至今日，多数嵌入式产品中都有linux的身影。

嵌入式开发人员可以使用更先进的方法、花费更多的时间来调优linux，并可以利用开源代码构建健壮的、特性丰富的应用程序。

本书内容新颖、逻辑严密，全面介绍了linux嵌入式开发环境，还进一步就目标板和开发环境的选择提供了建议。

全书以项目方式带领读者一步步深入嵌入式linux开发的各种技术，分析了开发中的常见问题。

如果你是专业的嵌入式开发人员，并对linux开发有基本的了解，那么本书恰好适合你阅读。

<<Linux嵌入式系统高级程序设计>>

作者简介

Gene Sally 从十几岁就开始接触计算机，有着近10年的嵌入式项目开发经验。利用Linux开发过会计、保险处理、社会福利系统以及电信管理软件的测试软件。曾任职于一家专门开发嵌入式Linux工具和发布版的公司，担任过工程师、培训师、技术支持等职务。

<<Linux嵌入式系统高级程序设计>>

书籍目录

第1章 嵌入式linux简介 第2章 配置软件环境 第3章 目标机仿真与虚拟机 第4章 开始你的项目 第5章 获取对应于目标板的linux 第6章 从头开始创建linux发布版 第7章 启动目标板 第8章 配置应用开发环境 第9章 应用开发 第10章 调试应用程序 第11章 内核配置和开发 第12章 实时 第13章 使用开源软件项目 第14章 busybox 第15章 系统设计 第16章 系统微调 第17章 部署应用程序 第18章 处理现场更新

<<Linux嵌入式系统高级程序设计>>

章节摘录

插图：1.基于标准Linux操作系统和相关的开源项目遵守工业标准。

大多数情况下，可用的开源实现是某个标准的规范或参考实现。

参考实现包含了对规范的解释，是一致性测试的基础。

简言之，参考实现是用于度量其他实现的标准。

如果读者不熟悉参考实现的概念，可能会有点困惑。

例如，POSIX中用于处理线程和进程间通信的部分，通常称作标准组是IEEE（Institute of Electrical and Electronics Engineers，电气和电子工程师协会）的一部分，它是一个委员会，设计了用于与线程交互的API，但该标准的实现则由另一个标准组完成。

实际上，在标准的工作开始时，委员会的一个或多个参与者会自愿创建相关的代码，即参考实现。

参考实现会包括一个测试套件，其他实现需要通过测试套件，才能证明相关代码是按照规范编写的。

使用基于标准的软件，不仅关系到质量，也关系到平台无关性。

项目基于遵守标准的软件，可以减少因为特定于厂商的特性而锁定到具体平台的可能性。

厂商可能是善意的，但这些额外特性的好处通常会因为缺乏互操作性和自由选择权而被抵消，互操作性和自由选择权已经悄然成为商业事务的一部分，但很少被认真考虑。

在一个有许多嵌入式设备建立连接（很多时候是连接到任意的系统，而非彼此连接）的世界里，标准变得越发重要。

以太网是一种这样的连接方法，但还有其他选择，如Zigbee、CANbus、SCSI，这只是其中几个例子而已。

2.进程隔离与控制Linux内核在最基本的层面上，以通用API的形式提供了这些服务，供访问系统资源之用。

管理任务，使之与内核彼此隔离。

提供统一接口，供访问系统硬件资源之用。

当存在争用时，充当资源的仲裁者。

这些是非常重要的特性，与硬件和资源访问未能得到严密管理的环境相比，这些特性将产生一个更为稳定的环境。

例如，在没有操作系统时，每个运行的程序对所有可用的物理内存都具有同等访问权限。

这意味着，一个程序中的溢出错误，将会写入到由另一个程序使用的内存中，除非对系统中所有的代码进行检查，否则第二个程序的失败原因会看似神秘而无法解释。

资源争用的概念是比较复杂的，不仅仅只是确保两个进程不会同时向串口写数据这么简单，稀缺的资源是处理器时间，而操作系统可以决定在何时运行哪个进程，以便最大化所能处理的工作量。

<<Linux嵌入式系统高级程序设计>>

媒体关注与评论

“如果你从事Linux嵌入式开发，此书绝对不容错过。”
——亚马逊读者讨论

<<Linux嵌入式系统高级程序设计>>

编辑推荐

《Linux嵌入式系统高级程序设计》：Linux嵌入式系统开发最新动态,以项目方式揭示Linux嵌入式开发的全过程,嵌入式系统开发人员必备。

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>