

<<精通嵌入式Linux编程>>

图书基本信息

书名：<<精通嵌入式Linux编程>>

13位ISBN编号：9787512400665

10位ISBN编号：7512400667

出版时间：2010-5

出版时间：北京航空航天大学出版社

作者：李玉东，李玉萍 编著

页数：209

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>

## &lt;&lt;精通嵌入式Linux编程&gt;&gt;

## 前言

第一个问题，为什么要写这本书？

现在很多面向嵌入式Linux编程的书籍理论性很强，但并不针对某一领域，对于解决某一领域的特定问题指导性较弱。

利用嵌入式Linux来构建系统，应用于便携式终端产品的居多，就软件方面而言，这类产品主要有两个环节需要把握：一是Linux内核（包括驱动）的移植；二是GUI（Graphical User Interface,即图形用户界面）层与应用层软件的设计。

本书就是基于后一个环节展开的。

有人说，嵌入式Linux的最大问题是其GUI没有统一标准。

但不知道这是它的缺点还是优点，是否应该由如Microsoft或Nokia这种级别的公司在这个操作系统平台上构建一个全世界都一样的用户界面，然后大家都用它的API来开发应用程序呢？

这个问题暂且不讨论。

嵌入式产品对于界面的需要千差万别，MP3、MP4、导航仪、电视机顶盒、手机等五花八门。

如果所有的界面都从“开始”菜单开始，操作起来不一定都很方便，另外，操作方式是用手指、遥控器、鼠标还是别的什么东西是可以选择的，因此，对于嵌入式产品，作者认为个性化用户界面才是合适的，任何一个GUI都不可能具有如此好的适应性和可配置性，把一个PDA风格的GUI系统移植到机顶盒上，或把一个手机风格的GUI移植到工控机里都是没有意义的。

解决问题的最好办法，就是自己构建一个小型的GUI环境，只针对具体应用，与其他系统无关。

那么，可能有人会说，量体裁衣，开发一个适合于自有项目的GUI环境固然很好，但这会不会很复杂，是不是会使项目周期拉长呢？

本书可以告诉你，开发一个小型的嵌入式GUI系统其实很容易！

何况网络上有如此之多的开源代码可供参考。

当然，无偿复制开源软件用于商业目的是不允许的。

但人们的思想是自由的，这一点谁也否认不了。

另外，现在已经开发完并开源的面向嵌入式Linux的GUI系统固然很多，而且还有一些人又在开发这个“柜”、那个“柜”的，但没有人仔细讨论到底一个嵌入式Linux GUI系统的体系结构如何能让使用者从全局把握系统，从而开发出自己的GUI环境，作者认为这是“授之以鱼”还是“授之以渔”的问题。

所以，作者写了这本书，通过嵌入式Linux特定环节的应用实例，即中间件层的GUI软件，来阐释Linux开发，同时使读者对于消息驱动的、轻量级窗口系统的实现有较为彻底的理解。

第二个问题，这本书有什么特点？

本书只针对GUI这个环节讨论技术问题，讨论其如何在嵌入式Linux上实现，并用到了Linux开发的技术细节，所以本书第一个特点是针对性强。

另外，作者不想把这本书搞成一个Linux编程的百科全书，讲清楚一个问题是最重要的，所以本书第二个特点是精炼。

本书出版之前，其早期版本作者一直放在网站上，有很多人下载并在网络上传播。

这样做的目的不为赚钱，只希望对大家都有所帮助。

由于时间有限，书中可能还存在一些错误。

另外，嵌入式Linux以及GUI技术的飞速发展，使得书中提到的一些概念有可能不再新颖，或者其中提到的GUI的实现方法不见得适用于任何项目。

但通过本书，可以了解到作者对于一个小型窗口系统的实现思路。

如果书中提到的概念或用于示例的LGUI实现代码有任何错误，欢迎读者批评指正。

本书能够顺利完成并出版，得到了很多老师与朋友的帮助，首先要感谢我的导师——北京大学人机交互与多媒体实验室的王衡副教授，她给了我大量的指导与帮助。

另外，奚小君、秦艺丹、李夏、李文阳、王文翩、刘彦军、夏华、刘成功、王成、郑浩、张明、张小铃、李志华、赵处一、王成明、李自忠、秦仕军、范佳新、王小良等朋友也给予了大力支持，在此向

他们表示感谢！

## <<精通嵌入式Linux编程>>

### 内容概要

本书针对使用Linux构建嵌入式系统的一个关键环节——图形用户界面（GUI），首先讲述了Linux编程的高级技巧，包括多进程、多线程等技术；然后通过实例重点讲述了窗口系统的基本知识与实现技巧，为读者开发自己的面向嵌入式Linux的GUI环境提供了一个参考实现范例。重点包括：LGUI多窗口的设计与实现、LGUI的消息管理、窗口与无效区的管理、设备上下文与图形设备接口的设计与实现等。

本书适用于使用Linux构建嵌入式系统的软件工程师以及希望深入了解窗口系统实现原理的读者。

## &lt;&lt;精通嵌入式Linux编程&gt;&gt;

## 书籍目录

第1章 概论 1.1 嵌入式系统的基本概念 1.2 嵌入式系统的特征 1.3 选择Linux构建嵌入式系统 1.4 GUI在嵌入式Linux系统中的地位及要求 1.5 用户界面概况 1.5.1 用户界面的历史 1.5.2 图形用户界面的特征 1.5.3 图形用户界面系统的结构模型 1.5.4 用户界面的发展: GUI+新人机交互技术 1.6 Linux图形环境及桌面平台简介 1.7 各种嵌入式Linux上的图形库与GUI系统介绍 1.7.1 Qt/Embedded 1.7.2 MicroWindows/NanoX 1.7.3 MiniGUI 1.7.4 OpenGUI 1.7.5 GTK+ 1.8 Linux系统中的多语言问题 1.9 一个嵌入式LinuxGUI系统开发的实例 1.9.1 开发GUI系统主要考虑的问题 1.9.2 后续讲解的实例第2章 Linux基本编程知识 2.1 编译器的使用 2.2 函数库的使用 2.3 Makefile 2.4 GDB 2.5 建立交叉编译环境 2.5.1 什么是交叉编译环境 2.5.2 交叉编译的基本概念 2.5.3 建立arm\_linux交叉编译环境 2.6 Linux下常见的图形库编程简介 2.6.1 Qt 2.6.2 GTK+第3章 Linux高级程序设计简介 3.1 Linux IPC介绍 3.1.1 信号 3.1.2 管道 3.1.3 消息队列 3.1.4 信号量 3.1.5 共享内存 3.1.6 Domain Socket 3.2 Linux多线程编程介绍 3.2.1 创建线程 3.2.2 线程的退出与取消 3.2.3 线程退出时的同步问题 3.2.4 线程清理函数 3.2.5 线程取消状态 3.2.6 线程同步 3.2.7 第三方函数库 3.3 FrameBuffer编程简介第4章 基本体系结构 4.1 基础知识 4.1.1 嵌入式Linux的GUI到底有什么用 4.1.2 如何定义基本体系结构 4.1.3 为什么用客户机/服务器结构 4.1.4 为什么要多进程 4.1.5 为什么要多线程 4.2 体系结构综述 4.2.1 客户机与服务器之间的通信通道 4.2.2 客户机需要与服务器交换什么信息 4.2.3 服务器对客户机进程的管理 4.3 进程创建与进程的管理第5章 多窗口的设计与实现第6章 GUI中的消息管理第7章 窗口输出及无效区的管理第8章 DC与GDI的设计与实现第9章 控件实现第10章 定制GUI对图像的支持第11章 字库及输入法的实现第12章 GUI的移植第13章 LGUI应用开发模式第14章 GUI系统的效率问题后记——LGUI开发的一些体会参考文献

## 章节摘录

插图：这时候就需要一个“技术部经理”出来说话，他来协调每个人发言的时间，以便每个人表达的信息都能为别人所了解。

那么这个协调与被协调的关系算不算是一个客户机/服务器结构呢？

一般意义上讲应该说不算，因为所谓客户机/服务器结构应该是：客户机发出请求，服务器进行处理，并将处理的结果返回到客户机。

技术部开会的时候并不是每个工程师发请求到技术部经理，由技术部经理完成处理后返回信息到工程师。

在这个系统中，技术部经理只是一个协调者的角色，而不是服务者的角色，所以并不是通常意义上讲的客户机/服务器结构。

但是另一方面，客户端有胖瘦之分，客户端要求服务器端处理的事情可能很复杂，也可能很简单。

在很复杂的情况下，客户端很少自己做事情，大部分事情都由服务器端完成；相反，客户端可能要求服务器做很少的事情，大部分事情由自己完成。

无论何种情况，它们之间有一个请求与被请求的关系、协调与被协调的关系。

所以，在这里不必过多讨论这是不是严格意义上的客户机/服务器结构，姑且认为协调者的角色就是服务器，被协调者的角色就是客户机。

在多个进程同时运行的情况下，任何一个进程在对屏幕进行输出的时候，都需要了解当前屏幕上的哪些区域是町以输出的，哪些区域是不可以输出的。

具体实现的时候，有两种方法：一是所有的输出都由一个服务进程来完成，由这个服务进程来确定当前对于哪些屏幕区域的输出请求是允许的，哪些是不允许的，这样就避免了多个进程对于屏幕区域的竞争；另一种方法就是其他进程只从服务进程那里请求并得到允许输出的区域，而具体的输出操作由自己完成。

前一种方法面临的问题是需要进程之间不停地传递大量数据。

不同进程之间除非通过IPC，否则因为不同的进程空间不允许互相访问数据，大块的数据需要在进程之间传递，这是非常耗费资源的操作，这在嵌入式环境中更是不可取的。

而后一种方法需要输出的进程只请求允许输出的屏幕区域，输出的操作由进程自己完成，相对而言效率会有很大提高。

而LGUI就是采取了这种方式。

4.1.4 为什么要多进程从GUI的角度讲，多进程实际上是多个进程对于屏幕的输出管理。

如果有很多进程在同时运行，但并没有屏幕输出的要求，就谈不上多进程的管理。

LGUI是一个支持多进程、多线程的客户机/服务器系统。

为什么要多进程？

单个进程不是更简单吗？

当然，并不是所有的嵌入式环境都要求多个进程同时运行，或者同时要求进行屏幕输出。

例如，一个机顶盒的GUI系统，就不会这样复杂。

但在一些复杂的嵌入式环境中，多进程是必需的，例如PDA等。

不能要求用户在PDA中添加一项功能，就重新将系统编译一下。

## 后记

当前，手持设备、智能终端、信息家电等嵌入式系统正得到蓬勃的发展。

而现在出现的市场只是冰山一角。

就像十多年前人们的梦想已变成活生生的现实一样，一个网络化、智能化的生活形态已经越来越清晰地展现在人们的面前。

而嵌入式系统的发展，正在为这个大潮起着推波助澜的作用。

与面向桌面计算的PC不同，嵌入式系统的最大特色是“个性化”，没有固定模式，尤其是用户界面。这注定了“定制”将会无处不在。

这也使得嵌入式系统的市场将在很大程度上会“碎块化”，没有哪家公司可以统一市场，形成事实上的标准。

嵌入式系统定制化的特点本身就要求系统是开放的，可量身定做的，这也就是为什么有很多公司从WinCE转向Linux的原因。

另外，Linux的低成本特点，可大大降低最终嵌入式系统的成本。

嵌入式系统的蓬勃发展，使得很多业内的大公司推出了专门针对嵌入式设备的专用芯片。

由于Linux开放源码、易于移植，并且由于Linux目前在嵌入式系统中所占据的举足轻重的地位，几乎所有的芯片制造商在推出新的芯片时，都会投入大量的人力、物力来移植Linux，并提供丰富的开发工具。

有些公司甚至会提供包括CPU、SDRAM、FLASH、LCD、TouchPanel、KeyBoard在内的功能齐全的测试板和所有这些外部设备基于Linux的驱动程序。

这使得嵌入式Linux的GUI系统成为一个完整产品构架中非常重要的环节，如果在GUI基础上进一步构造面向某一行业的应用，为行业用户提供解决方案，将为嵌入式Linux的发展起到一定的推动作用。

开发一个中等规模的系统，首先需要有全局意识，但除此之外，对一个完全陌生的系统，并不完全是“需求—设计—编码—调试”的过程，而是需要不断地反复。

不断地通过编写代码来证明设计是可以实现的，且该种实现方法的非功能指标是满足设计要求的，然后再重新调整最初的框架。

这正是软件工程中讲的“迭代”过程。

唯一满足工程设计标准的文档，就是源代码清单。

对于一个需求明确、功能复杂的系统，可以通过一些图表展示系统设计，但经过验证的框架性的代码也许才是最好的设计文档。

在对系统功能、实现的环境都了解有限的情况下，凭空进行设计，绝大多数情况下是没有任何意义的。

。

## <<精通嵌入式Linux编程>>

### 编辑推荐

《精通嵌入式Linux编程:构建自己的GUI环境》是由北京航空航天大学出版社出版的。



#### 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>