

## <<Qt及Linux操作系统窗口设计>>

### 图书基本信息

书名：<<Qt及Linux操作系统窗口设计>>

13位ISBN编号：9787121024344

10位ISBN编号：7121024349

出版时间：2006-4

出版时间：电子工业出版社

作者：倪继利

页数：449

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>

## <<Qt及Linux操作系统窗口设计>>

### 内容概要

如何在Linux内核上建立窗口系统，这是嵌入式设备软件开发工程师必须了解的。

Qt / Embedded是高端嵌入式设备PDA及手机的主流开发工具。

作为一项成熟的技术，KDE桌面系统对于嵌入式窗口系统的设计有很大参考价值：Qtopia给嵌入式窗口系统提供了很好的框架。

本书不仅阐述了KDE及X Window的机制，分析了Qt及Qt / Embedded开发工具的核心技术，而且还详细介绍了如何在嵌入式设备上建立Qtopia窗口系统。

掌握了本书中所介绍的技能，读者可以在Linux内核上建立窗口系统，开发应用程序并能够在PC上模拟运行嵌入式设备开发的应用程序。

本书主要针对从事开发Linux应用程序的软件开发工程师，也很适合作为大学教材和参考书。

第1章“X Window”阐述了X的体系分层结构、显卡驱动程序、X协议，说明了建立在X上的各种编程方法，还分析了X客户端应用程序Xlib函数如何发出请求以及X服务器的工作原理及源代码。

第2章“Qt编程核心技术”介绍了Qt对象模型、国际化方法、元对象及代码生成、进程间通信、窗口部件的基类、模板库和集合类、Qt线程、鼠标拖放、键盘焦点、会话管理，以及调试等方面的技术。

第3章“KDE窗口系统”介绍了KDE的MIME类型处理、服务、KIO框架、组件框架和国际化等，还分析了KDE的启动过程及kicker应用程序。

第4章“Qt / Embedded客户 / 服务器”讲述了将应用程序下载到嵌入式设备运行的方法，介绍了各种嵌入式GUI窗口系统，阐述了Qt / Embedded的客户 / 服务器模型工作过程。

本章还分析了设备输入事件在客户与服务器之间的传递机制、应用程序窗口的显示原理，并说明了运行Qt / Embedded程序的方法。

第5章“Qtopia核心技术”介绍了Qtopia开放源代码的核心技术及类库，其中着重分析了插件的装载过程、应用程序快速启动器的机制、类MimeType，以及编写输入法的方法等。

第6章“Qtopia服务器”主要说明组成Qtopia服务器的各个类的功能，介绍了Qtopia服务器的启动过程，分析了Qtopia服务器如何启动应用程序及插件，还分析了电源监控、外观设置、应用程序安装与卸载以及AppServices等与系统管理相关的小应用程序。

第7章“Qtopia的sysinfo应用程序分析”详细分析了sysinfo应用程序的源代码。

第8章“Qtopia编译及系统集成”说明了Qtopia的定制与集成，阐述了，pro项目文件的语法。并且详细分析了由pro项目文件生成Makefile文件的过程以及描述了Qtopia的打包、安装和调试方法。

# <<Qt及Linux操作系统窗口设计>>

## 书籍目录

第1章 X Window 1.1 X Window的体系结构 1.1.1 X Server 1.1.2 X Client 1.1.3 X Protocol 1.2 窗口管理器 1.3 X Window启动过程 1.4 XFree86配置文件分析 1.5 X Window程序设计简介 1.5.1 Xlib编程 1.5.2 Motif编程 1.5.3 GTK / GNOME 1.5.4 KDE 1.6 显示驱动程序 1.6.1 显卡驱动方式 1.6.2 帧缓冲 1.6.3 在台式机上使用通用帧缓冲 1.6.4 帧缓冲设备驱动程序 1.7 X Window协议 1.7.1 X协议 1.7.2 x传输接口 1.7.3 XDMCP协议 1.8 x client应用程序源代码分析 1.8.1 xinit分析 1.8.2 XOpenDisplay 1.8.3 函数XCreateWindow分析 1.9 x Server应用程序分析 1.9.1 X Server概述 1.9.2 DIX层 1.9.3 OS层 1.9.4 DDX层 1.9.5 Screen

第2章 Qt编程核心技术 2.1 Qt概述 2.2 Qt对象模型 2.2.1 信号和槽 2.2.2 元对象系统 2.2.3 元对象编译器限制 2.2.4 属性 2.3 QObject类 2.3.1 对象树 2.3.2 事件处理过程 2.3.3 事件运行机制 2.3.4 事件过滤器 2.3.5 定时器 2.3.6 连接函数connect 2.3.7 字符串翻译函数 2.4 Qt国际化 2.4.1 软件中字符串国际化方法 2.4.2 创建译本 2.4.3 编码支持 2.5 QMetaobject元对象类 2.5.1 相关的数据结构 2.5.2 QMetaObject对象 2.5.3 Q.OBJECT宏及moc生成代码分析 2.6 进程间通信——QCopChannel 2.7 窗口部件类 2.7.1 窗口部件的基类QWidget 2.7.2 QFrame类 2.7.3 QScrollView类 2.8 Qt风格机制 2.8.1 风格类 2.8.2 窗口系统风格更新机制 2.9 布局类 2.10 Qt插件 2.11 Qt模板库 2.11.1 迭代器 2.11.2 算法 2.11.3 数据流串行化 2.12 集合类 2.12.1 基于指针的容器的结构 2.12.2 管理集合条目 2.12.3 迭代器 2.13 Qt线程 2.13.1 线程类QThread 2.13.2 线程安全的事件传递 2.14 鼠标拖放 2.15 键盘焦点 2.15.1 焦点移动的方式 2.15.2 焦点策略及操作函数 2.16 会话管理 2.16.1 会话管理 2.16.2 测试和调试会话管理 2.17 调试技术 2.17.1 命令行参数 2.17.2 打印警告和调试消息 2.17.3 调试宏

第3章 KDE窗口系统 3.1 KDE 3.5源代码说明 3.2 KDE类库 3.3 系统资源访问 3.3.1 标准资源目录 3.3.2 KDE系统配置缓冲 3.4 图形支持 3.5 用户界面 3.5.1 Action模式 3.5.2 用XML定义菜单和工具栏 3.5.3 在线帮助 3.5.4 复杂窗口部件 3.6 MIME类型处理 3.6.1 一个应用MIME类型的例子 3.6.2 映射MIME类型到一个应用程序或服务 3.7 KDE服务 3.7.1 定义服务类型 3.7.2 定义共享库服务 3.7.3 定义DCOP服务 3.8 KIO框架 3.8.1 同步使用KIO 3.8.2 同步使用KIO 3.8.3 MetaData 3.8.4 调度 3.8.5 定义一个ioslave 3.9 DCOP 3.9.1 DCOP工具 3.9.2 DCOP收发数据 3.10 KDE应用程序启动其他应用程序的方法 3.11 国际化和本地化 3.12 编译方法 3.13 桌面框架 3.13.1 Panel Applets 3.13.2 控制中心模块 3.14 组件框架 3.14.1 KParts组件 3.14.2 系统托盘 3.14.3 集成Java 3.15 KDE协议说明 3.16 KDE启动 3.16.1 startkde脚本分析 3.16.2 startkde中的应用程序说明 3.16.3 桌面组件 3.17 kicker应用程序 3.17.1 kicker应用程序的类继承关系 3.17.2 桌面配置文件 3.17.3 kicker窗口的建立过程 3.17.4 K菜单

第4章 Qt / Embedded客户 / 服务器 4.1 各种嵌入式GUI 4.1.1 X Window 4.1.2 MICROWindows 4.1.3 MiniGL 4.1.4 QT / Embedded 4.2 Qt/Embedded的应用示例 4.2.1 编译Linux内核 4.2.2 root文件系统 4.2.3 裁剪Qt / Embedded库 4.2.4 编译应用程序 4.2.5 拷贝Qt库及应用程序 4.2.6 打包 4.2.7 下载及运行 4.3 帧缓冲 4.3.1 Direct1.1B介绍 4.3.2 虚拟帧缓冲qvfb 4.4 创建应用程序 4.4.1 QApplication类 4.4.2 QApplication构造函数 4.5 构建服务器 4.5.1 QWSServer类说明 4.5.2 QWSServer类构造函数分析 4.6 Qt / Embedded客户端与服务器通信 4.6.1 服务器相关的类说明 4.6.2 客户与服务器间的接收事件处理 4.6.3 服务器对窗口的协调管理 4.7 输入设备与应用程序通信 4.7.1 鼠标键盘相关类 4.7.2 打开鼠标设备的过程 4.7.3 设备文件与socket连接 4.7.4 服务器捕获设备输入信号 4.7.5 服务器发送事件给客户 4.8 应用程序显示 4.8.1 与显示相关的类 4.8.2 与字体相关的类 4.8.3 函数init display 4.8.4 QWSDisplay构造函数分析 4.8.5 QScreen 4.8.6 与图形显示相关的类 4.8.7 图形显示 4.9 运行Qt / Embedded应用程序 4.10 字符输入 4.11 增加一个加速显示卡驱动程序到Qt / Embedded 4.12 Qt / Embedded作为VNC服务器

第5章 Qtopia的核心技术 5.1 Qtopia简介 5.1.1 Qtopia PDA 5.1.2 Qtopia手机版 5.1.3 手机库 5.1.4 Qtopia Desktop 5.2 信道和消息 5.3 国际化 5.3.1 翻译操作步骤 5.3.2 .qm文件位置 5.3.3 Desktop文件国际化 5.3.4 文档国际化 5.4 Qtopia文件系统标准 5.5 Qtopia插件 5.5.1 建立插件的步骤 5.5.2 装载插件的方法 5.6 输入法插件 5.6.1 创建弹出输入方法 5.6.2 创建复合输入法 5.7 快速启动应用程序 5.7.1 改进主窗口创建时间 5.7.2 Quick Launcher 5.7.3 应用Quick Launcher 5.7.4 Quick Launcher的宏定义 5.7.5 运行Quick Launcher分析 5.8 Qtopia核心类库说明 5.8.1 QPEApplication类 5.8.2 应用程序配置与连接类 5.8.3 PIM管理类 5.8.4 桌面相关类 5.8.5 字体与输入法相关类 5.8.6 StorageInfo和FileSystem类 5.8.7 DeviceButtonManager和DeviceButton类 5.8.8 图像处理相关类说明 5.8.9 多媒体播放相关类说明 5.8.10 网络相关类说明 5.8.11 软件模块接口类 5.8.12 Service和ServiceRequest类 5.8.13 MimeTVpe类 5.8.14 其他类说明

第6章 Qtopia服务器 6.1 Qtopia服务器特定的类 6.1.1 ServerApplication类 6.1.2 Server类 6.1.3 Launcher类 6.1.4 LauncherTabWidget、LaunchernTabBar

## <<Qt及Linux操作系统窗口设计>>

和LauncherTab类6.1.5 LauncherView和LauncherItem类6.1.6 AppLauncher和QuickLauncher类6.1.7 LoadingWidget和DocumentList类6.1.8 TaskBar类6.1.9 StartMenu和StartPopupMenu类6.1.10 InputMethodSelector和InputMethods类6.1.11 RunningAppBar和SysTray类6.1.12服务器的其他类说明6.2 Qtopia服务器启动过程6.2.1 Qtopia服务器的main函数6.2.2 ServerApplication类构造函数6.2.3 Server类构造函数6.2.4创建服务器主窗口界面UI6.2.5创建应用程序图标视图6.3应用程序启动过程6.4文档的打开过程6.5电池监控小应用程序6.5.1电池状态窗口类6.5.2电源状态管理器类6.5.3插件装载6.6外观设置6.6.1 main函数6.6.2 SampleWindow类6.6.3 AppearanceSettings类6.6.4 AppearanceSettings类的构造函数分析6.7安装与卸载应用程序6.8 AppServices应用程序6.8.1 AppServices类6.8.2 ASCheckListItem类6.9插件管理器第7章 Qtopia的sysinfo应用程序分析7.1 SystemInfo类7.2 VersionInfo类7.3 StorageInfoView类7.4负载信息LoadInfo7.5内存信息7.6 DataView类第8章 Qtopia编译及系统集成8.1 qmake介绍8.1.1 qmake编译过程8.1.2.pro文件的语法分析8.2 Qtopia编译方法8.2.1 Qtopia编译步骤8.2.2 Qtopia Desktop编译步骤8.2.3编译一个应用程序8.2.4使用NFS运行应用程序8.2.5应用程序调试8.3为设备定制Qtopia8.3.1定制Qtopia启动器用户界面8.3.2设备硬件因素8.3.3不安装及不支持的组件8.3.4安全模式需要的插件8.3.5电源管理8.3.6可移去的存储卡8.3.7 118N8.3.8访问权限和只读文件系统8.3.9 MMS客户端8.3.10 GSM模块集成8.3.11定制键盘8.3.12配置硬件按钮8.4系统集成8.4.1创建设备特定交叉编译工具配置文件8.4.2定制电源、背景灯控制和键盘扫描码8.4.3在Qt / Embedded中创建键盘处理函数8.4.4配置指针设备8.4.5配置适合设备的Qt / Embedded库8.4.6配置适合设备的Qtopia8.4.7为目标设备创建和安装Qtopia image文件8.4.8在设备上配置适合于Qtopia的环境变量8.4.9安装另外的字体8.4.10集成Java 8.5 Qtopia编译系统8.5.1内部的编译系统项目文件说明8.5.2项目文件生成: Makefile8.5.3 src-components.pro和Src.pro文件8.5.4 configure脚本生成Makefile8.6 Qtopia的编译过程8.6.1编译库8.6.2相互独立的应用程序及插件的编译8.6.3文件的安装8.6.4打包分发8.7生成img文件并烧录主要参考文献

## <<Qt及Linux操作系统窗口设计>>

### 媒体关注与评论

本书特点： 实际：作者在实践中使用QT开发嵌入式设备的就用程序界面，在本书中，他与读者分离自己在外企从事Linux研发的经验。

实用：具有专业人士所需的深度，可以直接拿来运用到开发实战中去。

本书介绍的KDE及X Windows机制、Qt及Qt/Embedded核心技术，有助于读者在嵌入式设备的Linux内核上创建自己的窗口系统。

实用：重点介绍了一些对实际研发很有用的设计方法，包括窗口分层设计、接口标准化、菜单通用化、应用程序查询硬件信息等。

## <<Qt及Linux操作系统窗口设计>>

### 编辑推荐

如何在Linux内核上建立窗口系统，这是嵌入式设备软件开发工程师必须了解的。

Qt / Embedded是高端嵌入式设备PDA及手机的主流开发工具。

作为一项成熟的技术，KDE桌面系统对于嵌入式窗口系统的设计有很大参考价值：Qttopia给嵌入式窗口系统提供了很好的框架。

本书不仅阐述了KDE及X Window的机制，分析了Qt及Qt / Embedded开发工具的核心技术，而且还详细介绍了如何在嵌入式设备上建立Qttopia窗口系统。

掌握了本书中所介绍的技能，读者可以在Linux内核上建立窗口系统，开发应用程序并能够在PC上模拟运行为嵌入式设备开发的应用程序。

本书主要针对从事开发Linux应用程序的软件工程师，也很适合作为大学教材和参考书。

## <<Qt及Linux操作系统窗口设计>>

### 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>