

# 一个基于 Qt/Embedded 应用程序集成桌面的设计与实现

吕强 夏俊鸾 钱培德 杨季文

(苏州大学计算机科学与技术学院, 江苏 苏州 215006) (江苏省计算机信息处理技术重点实验室, 江苏 苏州 215006)

**摘 要:** 文章介绍了基于 Qt/Embedded 应用程序的集成桌面在一个嵌入式设备中的设计与实现, 给出了整个系统的构架, 突出了 X Server 层的实现关键, 描述了应用程序的窗口管理, 以及应用程序与驱动程序通信、数据传输等问题的解决方案。

**关键词:** 集成桌面, 窗口管理, 应用间通信

中图法分类号: TP311

文献标识码: A

文章编号: 1000-7180(2005)07-069-04

## A Design and Implementation of Integrated Desktop for Qt/Embedded Applications

LV Qiang, XIA Jun-luan, QIAN Pei-de, YANG Ji-wen

(School of Computer Science and Technology, Soochow University, Suzhou 215006 China)

(Jiangsu Provincial Key Lab of Computer Information Processing, Suzhou 215006 China)

**Abstract:** The paper introduces a design and implementation of desktop based on Qt/Embedded for integrated applications in an embedded device. Giving the framework of the system, the authors emphasis the critical role of X Server layer, and describe the solution of applications window management as well as the communication and data transmission between applications and device driver.

**Key word:** Integrated desktop, Window management, InterApplication communication

### 1 引言

随着面向信息处理的嵌入式设备不断发展, 嵌入式产品中的应用程序逐渐增多, 随之功能也不断强大。在嵌入式操作系统方面出现了 Palm OS 和 Windows CE 等, 但 Linux 以其开放源码、容易定制和扩展、多硬件平台支持和内置网络功能等优良秉性, 逐渐成为嵌入式系统的研究热点和广泛使用的系统平台。嵌入式设备系统中有许多针对设备特定功能的应用程序, 为了在一个设备中让用户协调地操作各个应用程序, 很好地处理这些应用程序与底层驱动以及应用程序之间的通信, 就需要一个机制来整体地管理这些应用程序, 比如 Trolltech 的 Qtopia 集成环境<sup>[1]</sup>, 在高端的信息处理型嵌入式设备中就有很好的应用。

我们的开发设计平台是基于 S3C2410 原形实验板<sup>[2]</sup>的一个固定电话设备, 具有 320\*240, 4bpp LCD, 电话应答设备 TAD(Telephone Answering De-

vice), Modem 和一个特殊的键盘, 键盘上有开发系统中的所有应用程序的快捷键, 暂时没有鼠标设备。系统层采用 Linux 操作系统, 应用程序是基于 Qt/Embedded 的窗口程序, 在设备上实现了如地址簿, 网页浏览器, 拨号程序, 电子邮件等应用程序。由于应用程序很多, 就必须提供一个统一的界面让用户操作, 本文要实现的就是这样一个应用程序的集成桌面, 它必须包括对应用程序的窗口管理, 应用程序与驱动程序以及应用程序之间的通信和数据传输。

### 2 桌面的设计

在 Linux 中的窗口系统是基于客户机/服务器模式的 X Window 系统, 该模式满足了 X 环境设计者的一个目标, 即完全可扩展性<sup>[3]</sup>。服务器也称作 X Server, 管理所有可用于显示器的资源, 所有的窗口绘制, 窗口裁剪都是通过 X Server 来完成的。客户机也称作 X Client, 通过 X Server 传递显示器请求的 X 应用程序。就本系统所采用的 Qt/Embedded 开发工具而言, 我们可以开发两种应用程序, 即 X Server 程序和 X Client 程序。我们决定将集成桌面的功能

收稿日期: 2004-12-20

基金项目: 江苏高校高新技术产业发展项目(JH02-042)

江苏省计算机信息处理技术重点实验室开放课题资助(KJS03062)

与 X Server 融合在一起,形成一个应用程序,将集成桌面的功能变成 X Server 的一部分,图 1 显示了整个系统的层次结构。

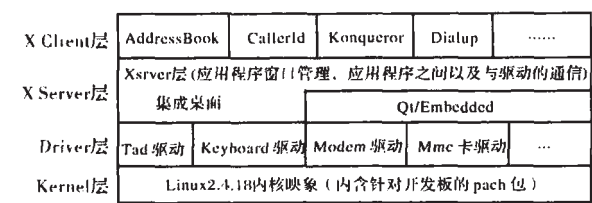


图1 系统层次结构图

本系统中的集成桌面是将一些经常使用的应用程序做成图标置于用户界面上,除了键盘上的应用程序快捷键外,对于在焦点状态的应用程序图标,按下回车键也可以启动或者显示应用程序窗口,但不管是用快捷键还是回车键来启动或者显示应用程序最终都会融合到一个统一的窗口管理框架内。图 2 是本系统桌面的截图,通过左下角的按钮弹出的菜单可将任意程序的快捷方式添加到桌面上,也可以将桌面上的程序快捷方式删除,这样对整个系统的可扩展性带来了很大的方便,用户可以随时定制自己的应用程序。但是如果不采用集成桌面,而是将所有应用程序融合成一个应用程序,那么就没有扩展性可言。

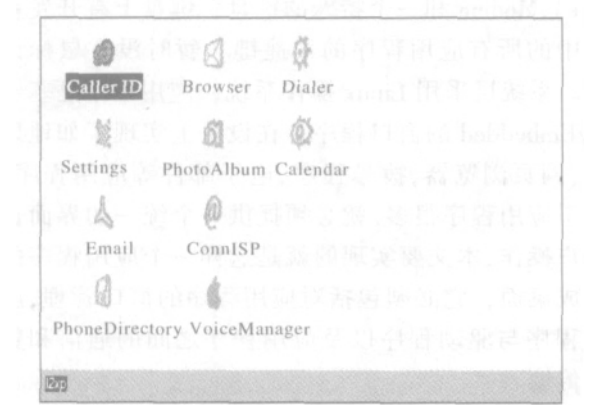


图2 系统桌面截图

3 应用程序窗口管理

在本系统中的应用程序窗口管理只涉及到窗口的显示与关闭,在窗口中隐藏了系统菜单,也不存在任务栏,所以不涉及到窗口的最大化和最小化问题。在 Qt/Embedded 的 X Server 应用程序中已经对系统中存在的窗口信息提供了有力的支持。为了能协调调度系统中每一个应用程序窗口,在系统中要为每个应用程序维护一个当前状态的信息结构,任何应用程序的启动和关闭,以及任何窗口的显示

和关闭都会影响到该结构,该结构的定义如下:

```
emun state = {App_Running, App_Pending, App_End}  
struct Process  
{  
    int iState; // application status  
    QList<QWSWindow> allWins; // all windows belongs to  
    this application  
    int iHotkey; // hotkey to activate this application  
    QWSClient *Application; // the internal id of this applica-  
    tion  
};
```

在上述结构中记录着应用程序的基本信息,分别是应用程序的运行状态,应用程序的所有窗口信息(QWSWindow 是 X Server 中的一个窗口信息类),应用程序快捷键,以及应用程序在 X Server 层的标识符(QWSClient 是在 X Server 中的一个应用程序信息类),随着用户对系统的操作,每个应用程序的数据结构中的值也将不断变化,应用程序窗口管理的流程如图 3 所示,其中①②④的功能都整合在 X Server 层,而③是在 X Client 程序中实现的,这四部分的具体功能如下。

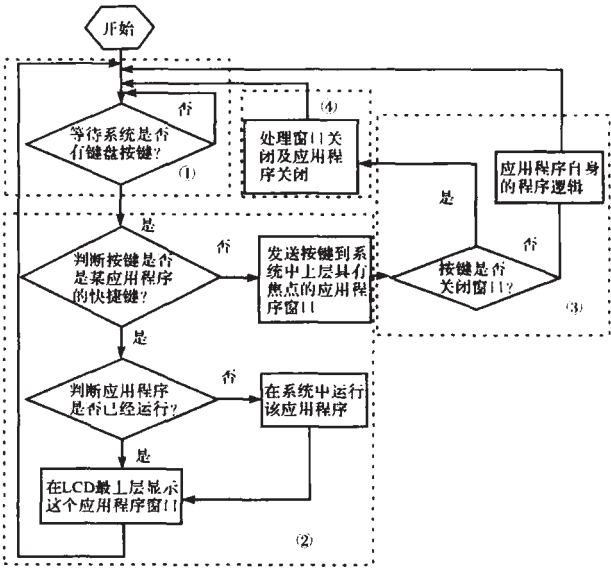


图3 窗口管理流程图

- ① 截获所有的用户键盘按键。
- ② 实现应用程序的窗口显示以及对应用程序信息的修改。
- ③ 接受从 X Server 层传递的按键,如果收到 Esc 键,就关闭窗口,否则执行其自身的程序逻辑。
- ④ 实现对窗口关闭后应用程序信息的修改等。

3.1 系统截键处理

在系统中的硬件设备中有一个特殊的键盘,在

键盘上有所有应用程序的快捷键,所以要显示程序窗口的首要任务就是要截获用户的所有键盘按键,然后分析按键以及后继对所得到的按键进行处理。

X Server 管理着所有可用于显示器的资源,与系统硬件也结合得比较紧密,Qt/Embedded 对其也进行了较好的封装,对于 X Server 层的程序提供了全局变量 `qwsServer`(类 `QWSServer` 的对象),其管理和保存着系统中 X Client 程序窗口的几乎所有信息,另外其中也包含了一个全局键盘钩子类 `KeyboardFilter`,可以在键盘驱动程序将用户按键发送到 X Client 程序之前首先截获到键值,并有权决定是否将按键发送到当前的用户应用程序(具有焦点的应用程序窗口),这个键盘钩子类的定义如下:

```
class KeyboardFilter: public QWSServer::KeyboardFilter
{
public:
    l2xpKeyboardFilter(); //constructor
    bool filter (int unicode, int keycode, int modifiers, bool
isPress, bool autoRepeat); // hook of the keyboard
    virtual ~l2xpKeyboardFilter(); //destructor
};
```

在类 `KeyboardFilter` 中的成员函数 `filter` 就是键盘钩子函数,其输入参数记录着一次键盘按键的所有信息,其返回值为一个布尔量,当返回 `true` 时,即这一次按键不返回给当前 X Client 程序,当返回 `false` 时,将用户按键返回给当前 X Client 程序,这样对用户的按键信号可以先在 X Server 层过滤一下,作出相应的处理后再向上层传递。

### 3.2 窗口显示处理

窗口显示处理集中在键盘钩子函数内,首先判断截获到的按键是否对应于某个应用程序,这可以通过键盘码来判定,如果正好对应于某个应用程序,那么可以分两种情况处理,第一种,如果应用程序还没有启动,那么就启动应用程序,显示应用程序的窗口,这时的程序窗口一定显示在屏幕的最上层,最后设置相应应用程序的信息参数。第二种,如果相应的应用程序已经启动,那么将程序恢复成用户最后一次使用这个应用程序的状态,比如用户正在编辑 Email 时,又调用拨号程序打电话,那么如果用户再次想操作 Email 时,就要恢复到用户上次编辑的状态,在全局变量 `qwsServer` 中记录着所有 X Client 程序窗口的信息,将系统中存在的各个应用程序的窗口用链表的形式组织起来,那么从顶层到底层的窗口串中挑选出从属于当前按键所对应的应用程序的窗口,将它们依次显示出来,就恢复到

用户最后一次使用的状态。

### 3.3 窗口关闭处理

在键盘钩子函数中对于不作处理的按键值就会发送到当前具有焦点的应用程序窗口,按照项目约定,每个窗口接收到 `Esc` 按键就关闭,该处理在图 3 的②与③的连接处,对于 X Client 程序窗口所有变化在 X Server 层都可以捕获到,这些变化状态包括创建(create)、显示(show)、激活(active)、销毁(destroy)等,在 Qt/Embedded 中有特定的信号和槽的绑定技术,在 X Client 程序窗口所有变化都会促发 `qwsServer` 变量中的信号,在上面的窗口显示中也会发出信号,只是我们对窗口显示发出的信号不处理罢了,但是对窗口的关闭就必须有相应的处理函数(也就是上文所说的槽),这个处理函数有两个输入参数,第一个表示哪个窗口的状态发生了变化(`QWSWindow` 窗口类对象),第二个表示窗口变化到什么状态(`QWSServer::WindowEvent` 中的某个状态),一旦有窗口发生状态的变化就会进入处理函数,该处理发生在图 3 的③和④的连接处,在处理函数中我们关注的是当 `QWSServer::WindowEvent` 参数为 `QWSServer::Destroy` 时,即当有某个窗口被销毁关闭时,判别该窗口是否是对应的应用程序的最后一个窗口,如果是则意味着整个应用程序的关闭,那么就要相应地设置该程序的参数,比如将该程序的运行状态改为未运行等,这样就对窗口的关闭处理完毕。

### 3.4 窗口调度的时机

经过上面截键,窗口显示和窗口关闭的三步处理,窗口管理的主体部分就完成了,在整个系统中有三种情况需要涉及到窗口管理调度,第一种情况下是用户显式地按键盘,想调度出某个应用程序,这时会涉及到窗口调度,显示出窗口;第二种情况是当底层的硬件设备的状态改变时,需要显示出与其状态变化相对的某个应用程序的窗口,比如,当用户拿起话筒时,底层的 Tad 设备的状态产生了变化,用户这时极有可能要打电话,所以在显示界面上要变成 `Dialer`(拨号程序)的界面,这时在 Tad 驱动中用 `handlescancode()` 内核函数模拟用户按键,发送该应用程序(`Dialer`)的快捷键,此时在键盘钩子函数中也会捕获到该模拟按键,从而调度出需要的应用程序;第三种情况是在用户操作应用程序时,可能会需要调出另外的应用程序,这时也涉及窗口调度,但是在这种情况下是通过应用程序与 X Server 的管道通信来告诉 X Server 需要调度出什么应



