

嵌入式笔记 - 第八章

本章主要介绍嵌入式GUI

嵌入式GUI系统

Microwindows

Microwindows是嵌入式系统中广为使用的一种图形用户接口。这个项目的早期目标是在嵌入式Linux平台上提供和普通个人电脑上类似的图形用户界面。

作为PC上X-Windows的替代品，Microwindows提供了和X-Windows类似的功能，但是占用的内存要少得多，根据用户得配置，Microwindows占用得内存资源在100KB-60KB。

Microwindows的核心基于显示设备接口，因此可移植性很好，microwindows有自己的Framebuffer，因此它并不局限于Linux开发平台，在eCos、FreeBSD、RTEMS等操作系统上都能很好地运行。

此外，Microwindows能在宿主机上仿真目标机。这意味着基于Linux的Microwindows应用程序的开发和调试可以在普通的个人电脑上进行，而不需要使用普通嵌入式软件的“宿主机 - 目标机”调试模式，从而大大加快了开发速度。

Mincrowindows是完全免费的一个用户图形系统。

MiniGUI

MiniGUI是由北京飞漫软件技术有限公司主持的一个自由软件项目(遵循GPL条款)，其目标是为基于实时嵌入式系统提供一个轻量级的图形用户界面支持系统。

MiniGUI为应用程序定义了一组轻量级的窗口和图形设备接口。利用这些接口，每个应用程序可以建立多个窗口，而且可以在这些窗口中绘制图形。用户也可以利用

MiniGUI建立菜单、按钮、列表框等常见的GUI元素。

TinyX

TinyX是一个针对嵌入式系统设计的X Windows,相对PC版的X Windows而言，它占用的资源少很多。

所有能够在标准X下运行的都能运行在TinyX上

从字符界面到启动TinyX只需要2秒左右的时间

不需要libc，兼容任何linux版本。

Qt/Embedded

Qt是KDE等项目使用的GUI支持库

Qt/Embedded是面向嵌入式系统的Qt版本

许多基于X Windows程序可以非常方便地移植到Qt/Embedded版本上

Qtopia and Qt/Embedded – Qtopia是一个用于PDA，掌上电脑和网络应用产品的移动设备窗口环境和应用套件，它基于Qt/Embedded，包括PIM应用程序，Internet客户端，娱乐和游戏程序，Qtopia/PDA可以使用商业版权或开源协议，但Qtopia Phone Edition 只能用于商业。

Qt

概述

Qt是挪威的Trolltech（奇趣）公司的旗舰产品，作为跨平台的应用程序框架，是开源的桌面系统KDE的基石。Qt是一个基于C++的跨平台图形用户界面应用程序开发框架，它既可以开发GUI程序，也可用于开发非GUI程序，比如控制台工具和服务器等。

和Java的“一次编译，到处运行”不同的是，Qt是源代码级的跨平台，一次编写，随处编译。一次开发的Qt应用程序可以移植到不同的平台上，只需要重新编译即可运行。

Qt编程核心技术

QT对象模型：

- 元对象系统
- 信号和插槽

QObject类：

- 对象树
- 事件

元对象系统

是一个C扩展，使得QT更适合真正的组件GUI编程；

使用元编译器moc产生能被标准C编译器访问的附加C代码；

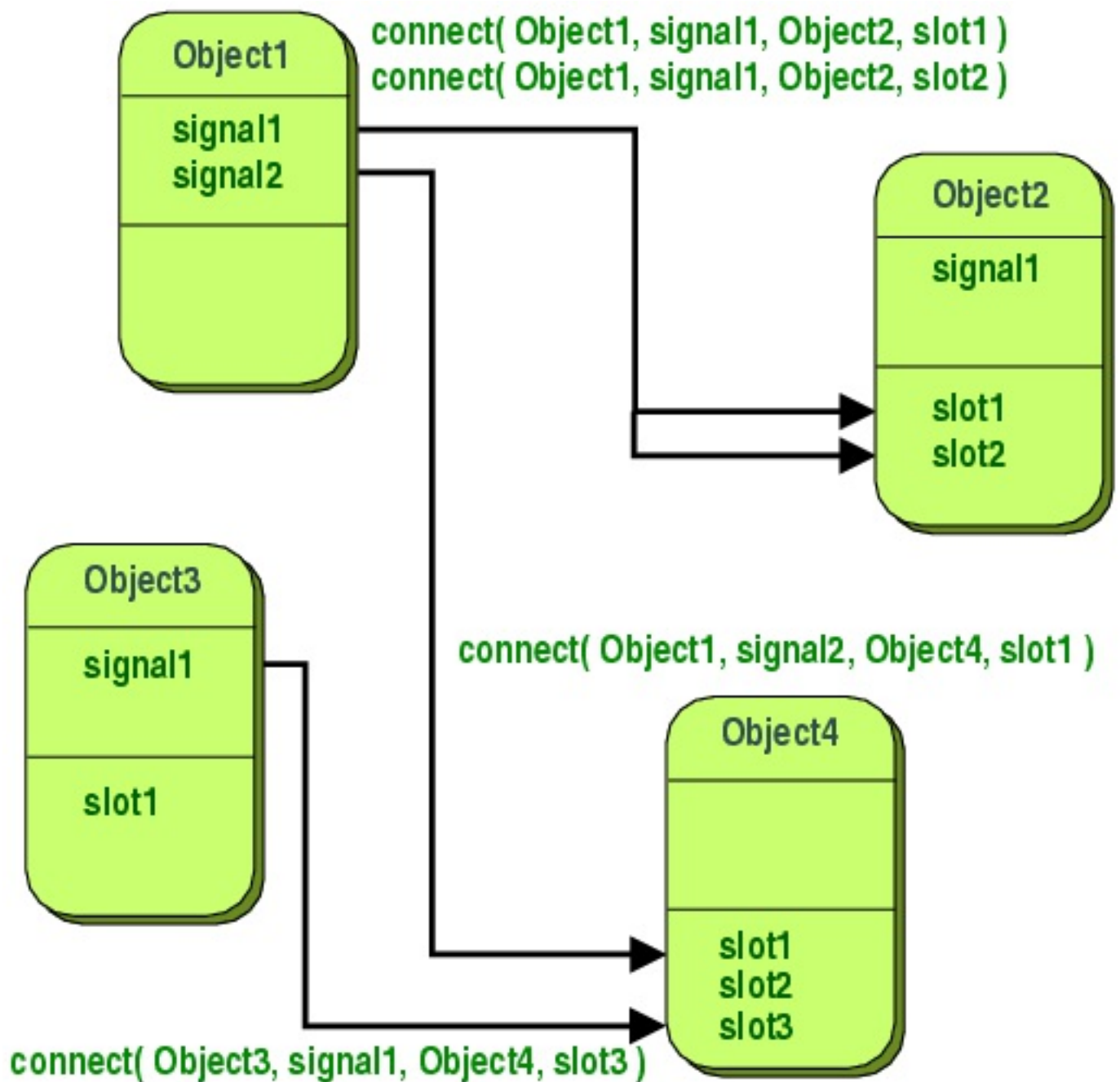
带有moc预编译器的C基本上提供了面向对象的灵活性，并保持了C++的执行效率和扩展性。

信号和槽

首先，我们必须理解信号(signal)和槽(slot)。这一技术有点不同于传统的回调(callback)函数。信号和槽技术是由Troll Tech公司独立开发的一而不是C++的功能。

信号和槽用于对象间的通信，该机制是Qt的一个中心特征，并且最能体现Qt与其他

工具包它们之间有什么不同。在图形用户界面编程中，我们经常希望一个窗口部件的一个变化被告知给另外一个窗口部件。简单的说，我们希望任何一类的对象可以和其他对象进行通信。



Qt的窗口部件有很多预定义的槽，当一个特定事件发生的时候，一个信号被发射，对信号感兴趣的槽就会调用对应的响应函数。

信号/槽机制是在QObject类中实现的。在QObject类或者其一个子类(比如：QWidget类)继承的所有类中，都存在于信号和槽。当对象改变其状态的时候，信号被发送，对象不关心有没有其他对象接收到这个信号。槽是类的正常成员函数，可

以将信号和槽通过connect()函数连接。当一个信号被发射，它所连接的槽会被立即执行，如同一个普通函数调用一样。

QObject类

QObject是Qt类体系的唯一基类，是Qt各种功能的源头活水。

就象MFC中的CObject和Dephi中的TObject。

对象树

Qt中使用对象树（object tree）来组织和管理所有的QObject类及其子类的对象。当创建一个QObject时，如果使用了其他的对象作为其父对象（parent），那么这个QObject就会被添加到父对象的children()列表中，这样当父对象被销毁时，这个QObject也会被销毁。实践表明，这个机制非常适合于管理GUI对象。例如，一个QShortcut（键盘快捷键）对象是相应窗口的一个子对象，所以当用户关闭了这个窗口时，这个快捷键也可以被销毁。

QWidget作为能够在屏幕上显示的所有部件的基类，扩展了对象间的父子关系。一个子对象一般也就是一个子部件，因为它们要显示在父部件的区域之中。例如，当关闭一个消息对话框（message box）后要销毁它时，消息对话框中的按钮和标签也会被销毁，因为按钮和标签是消息对话框的子部件。

事件

事件是由窗口系统或qt本身对各种事务的反应而产生的。当用户按下、释放一个键或鼠标按钮，一个键盘或鼠标事件被产生；当窗口第一次显示，一个绘图事件产生，从而告知最新的可见窗口需要重绘自身。大多数事件是由于响应用户的动作而产生的，但还有一些，比如定时器等，是由系统独立产生的。

Gt/Embedded

Qt/Embedded(简称QtE)是一个专门为嵌入式系统设计图形用户界面的工具包。Qt是挪威Trolltech软件公司的产品，它为各种系统提供图形用户界面的工具包，QtE就是Qt的嵌入式版本。

Qtopia

Trolltech公司在QtE的基础上开发了一个应用的环境—Qtopia，这个应用环境为移动和手持设备开发。

其特点就是拥有完全的、美观的GUI，同时它也可提供上百个应用程序用于管理用户信息、办公、娱乐、Internet交流等。

已经有很多公司采用了Qtopia来开发他们主流的PDA。