

交互界面开发工具 - Qt

李全虎 湖南大学软件学院 410000

摘 要: Qt 工具箱是一个广泛应用的跨平台图形用户界面(GUI)开发工具,可用于 Windows, Linux, Mac OS X 和 handheld 平台。Qt 有一个非常好的结构,而且灵活,基于面向对象的组织,以及清晰的文档。

关键词: 交互界面; 跨平台; 国际化; 面向对象

交互界面(Interface)是软件与最终用户的接口,在很大程度上决定着软件的成败。因此,为应用程序选择一个图形用户界面(GUI: Graphical User Interface)开发工具箱是软件开发过程中非常重要的一件事。目前,可供选择的图形用户界面工具箱较多,而且每一个工具箱都有自己的优点和缺点。有些比其它速度快,有些比其它的小,有些易于安装,有些是跨平台的,有的甚至支持用户提出的特殊需求。本文将介绍一种新的交互界面开发工具—Qt。

Qt 是一个为软件开发者设计的图形用户界面工具箱,它简化了开发和维护图形用户界面应用程序的任务。

一、Qt 简介:

Qt 是跨平台的 C++ 图形用户界面工具箱,它是挪威 Troll Tech 公司的标志产品。Qt 工具箱有三种不同的版本,一种是专业版(Professional Edition),另一种是免费版(Free Edition),还有一种是嵌入式免费版(Embedded Free Edition)。由于 Qt 是一个跨平台的 GUI 工具箱,它确保了软件非常方便的应用与绝大多数操作系统。目前 Qt 支持的操作系统包括: 1、MS / Windows - 95, 98, NT, ME, 2000 和 XP; 2、Unix / X11 - Linux, Sun Solaris, HP-UX, Compaq Tru64 Unix, IBM AIX, SGI IRIX 等; 3、Macintosh - Mac OS X。

Qt 具有广阔的应用前景, Troll Tech 公司负责维护 Qt。Qt 已应用于全世界上百个软件开发项目。因此,当需要开发 Unix 或 MS Windows 环境下的 GUI 程序是, Qt 是最佳选择。

二、Qt 的特点

1. 可移植性

Qt 是一个跨平台 GUI 工具箱,不仅适用于 Unix,它同样适用于 MS Windows。如果你的开发平台是 MS Windows,你可能选择的 GUI 工具箱为 Microsoft Foundation Classes (MFC),这样就会失去众多的 Unix 用户。相反,如果你的开发平台是 Unix,你可能选择其它工具箱,如 Motif 等,同样,你会失去众多的 Windows 用户。如果选择的 GUI 工具箱为 Qt,此问题则可迎刃而解。我们所需要的仅仅是你想运行的平台上简单的编译源代码。

Qt 把在处理不同窗口系统时的潜在问题隐藏了起来。为使 Qt 使用方便,Qt 包含了一系列类,这些类使开发人员避免了在文件处理、时间处理等方面存在的依赖操作系统方面的细

节。

2. 国际化(Internationalization)

所谓国际化,通俗的讲,就是指在某国开发的软件可以方便的被其它国家的人使用。Qt 为本地化应用提供完全的支持,所有用户界面文本都可以基于消息翻译表被翻译成各国语言。另外,Qt 完全支持双字节 16bit 国际字符标准。利用 Qt 开发跨平台的国际化软件是一个方便、增量的过程。在开发过程中,只要我们遵循 Qt 的一些国际化的方法,开发的软件就很容易实现国际化。

3. 面向对象

Qt 是一个 C++ 工具箱,它由几百个 C++ 类构成。你在程序中使用这些类,如果 Qt 没有提供真正适合需求的组件,你可以方便的生成自定义的组件,并且非常容易修改组件的特征。Qt 具有模块设计和注重软件构件或元素的可重用性的特点。一个组件不需要知道它的内容,而通过 Qt 特有的 signal 和 slot 机制与外界通信、交流。

4. 可用户化的外观

Qt 支持主题,所以基于 Qt 的应用软件能在 Motif 外观、Windows 外观,以及另一些用户化的外观之间互换,甚至改变运行时间。这些应用程序不管是在 X Window 下还是在 Microsoft Windows 下都可以独立操作、运行。

三、Qt 中的三个主要基类

1. QObject 类

QObject 类是所有能够处理 signal / slot 和事件的 Qt 对象的基类。Qt 为对象的通信以及 signal / slot 连接提供了强大的机制,signal / slot 机制比传统的回调函数更先进。其原型为: `QObject::QObject(QObject*parent=0, const char*name=0)`

对象的父对象可以从对象之间的主从关系看出。比如对话框是其中 Ok 和 Cancel 按钮的父对象。如果 parent 为 0,则构造一个无父的对象;对象名(name)用于标识 QObject。所有 Qt 组件都继承 QObject 和利用 signal / slot 机制。signal / slot 机制允许对象的再利用,因为发出 signal 的对象不需要知道与它相连的是那些 slot。

2. QApplication 类

QApplication 类负责 GUI 应用程序的控制流和主要的设置,包含主事件循环体,负责处理和调度所有来自窗口系统和其它资源的时间,并且处理应用程序的开始、结束以及会话管理,还包括系统和应用程序方面的设置。对于一个应用程序来说,建立此类的对象是必须的。QApplication 是 QObject 的子类。

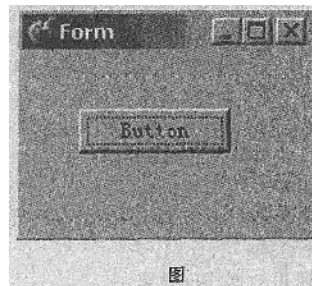
3. QWidget 类

QWidget 类是所有用户接口对象的基类,它继承自 QObject 类。它接收鼠标、键盘和其它从窗口系统来的事件,并把它自己画在屏幕上。QWidget 类有很多成员函数,但是一般不直接使用,而由它的子类继承,使用函数功能。

四、一个简单的例子

```
1 #include <qapplication.h>
2 #include <qpushbutton.h>
3
4 int main(int argc, char *argv[])
5 {
6     QApplication app(argc, argv);
7     QWidget hello;
8     app.setMainWidget(&hello);
9     hello.show();
10    return app.exec();
11 }
12
13 class MyWidget: public QWidget
14 {
15 public:
16     MyWidget(QWidget *parent=0,
17              const char *name=0);
18 private slots:
19     void ok_slot();
20 private:
21     QPushButton *button;
22 };
23 MyWidget::MyWidget(QWidget *parent, const char *name)
24 : QWidget(parent, name)
25 {
26     button = new QPushButton("Button", parent);
27     connect(button, SIGNAL(clicked()),
28            this, SLOT(ok_slot()));
29     resize(200, 200);
30 }
31 void MyWidget::ok_slot()
32 {
33     quit();
34 }
```

程序执行结果如图所示。当用户点击按钮“Button”后,对话框将关闭。当然实现如图所示的效果,有更简单的实现方式,在此仅仅为了展现信号、槽的使用方法。限于文章的篇幅,对程序不作进一步的解释。



图

结论

Qt 跨平台的国际化的交互界面开发工具,并且在很多项目中取得了成功,而且也是学习面向对象思想的一本好教材。

参考文献:

- [1] <http://www.trolltech.com/qt>
- [2] Daniel Dolin 著 袁鹏飞译 24 小时学通 Qt 程序设计. 人民邮电出版社. 2000 年