

Linux 下图形用户界面程序的开发与实现

王子强 刘海燕 李媛州

(装甲兵工程学院信息工程系 北京 100072)

摘要 在 Linux 下开发图形用户界面应用程序是 Linux 推广的重要基础。本文首先介绍了 Linux 下图形用户界面程序开发的主流工具,然后重点分析了 Qt 及其提供的一种真正可视化的编程工具 Qt Designer 的特点及开发步骤,最后从我们实际开发经验出发总结了 Qt 的优点。

关键词 Linux 编程工具 Qt Designer

DEVELOPMENT AND REALIZATION OF GRAPHICAL USER INTERFACE PROGRAM ON LINUX

Wang Ziqiang Liu Haiyan Li Yuanzhou

(Department of Information Engineering, Armored Force Engineering Institute, Beijing 100072, China)

Abstract To develop Graphical User Interface program (GUI) on Linux is an important factor to popularize Linux. Firstly, this paper introduces the main GUI programming tools on Linux, then analyses the characteristics and developing steps of Qt and Qt Designer which is a really visual programming tool provided by Qt, finally it summarizes the strongpoints of Qt based on our programming experience.

Keywords Linux Programming tools Qt Designer

0 引言

Linux 操作系统是一个开放源代码的免费操作系统,受到越来越多用户的欢迎。随着 Linux 操作系统在我国的不断普及,有关政府部门将基于 Linux 开发具有自主知识产权的操作系统提高到保卫国家信息安全的高度来看待,因此不难预测今后 Linux 操作系统在我国将会得到更快更大的发展。由此,在 Linux 下开发图形用户界面的应用程序不但是一种时尚,更是一种需求。然而由于缺乏宣传运作,与 Windows 下众多的程序开发人员相比,熟悉 Linux 下开发工具的人实在太少了。

本文首先简单介绍了 Linux 下常用的开发工具,然后详细分析 Qt 的特点,说明 Qt 及其提供的一种真正可视化的编程工具 Qt Designer 的开发步骤,最后根据我们的开发实践总结了 Qt 的优点。

1 Linux 下常见开发工具

在 Linux 下开发图形用户界面程序的主流开发工具包括:

(1) GTK+ GTK (GMP Tool Kit, GMP 工具包)最初是 GMP 的专用开发库,后来发展成为 Linux 下开发图形界面应用程序的主流开发工具之一。GTK+ 是完全自由的工业级的图形界面开发工具,它采用 C 语言设计。它的当前最高版本是 GTK+2.2.4。

(2) Kdevelop 它是 KDE 环境下最优秀的集成开发工具。它界面友好、功能强大。熟悉 Visual C++ 的程序员一定会被这样一个优秀的开发工具所吸引。其当前版本是 Kdevelop3.0。

(3) Kylix 它是 Borland 于 2001 年初推出的 Linux 版本的 Delphi,它和 Windows 下的 Delphi 一样都是使用 Object Pascal 语言。它采用以 Xwindow Qt 开发组件为基础的新的组件架构 CLX,与多数 Linux 系统兼容,是基于 Linux 系统的第一套集界面设计、代码编写、调试运行于一体的 RAD。相信熟悉 Delphi 的程序员对它一定会很快上手。

(4) Qt 它是 Troll Tech 公司的标志性产品,而 Qt Designer 是 Qt 提供的一种真正可视化的编程工具,采用 C++ 作为程序设计语言。Qt 工具包有两种不同的版本,一种是用于商业性质的专业版,另一种是用于开发 X 平台自由软件的自由版。专业版用于创造成功的商业贸易方面的跨平台软件;而自由版已经成为用 C++ GUI 工具包在 Linux 上进行自由软件开发的主流,它是 Linux 上流行的 KDE 桌面环境的基础。另外,它还是一种跨平台的编程工具,能够确保软件非常方便地应用于绝大多数操作系统。

2 Qt 的特点和优势

Qt 作为一种非常优秀的编程工具,具有很多优点和一些其它编程工具所不具备的特征。

(1) 面向对象

Qt 是完全面向对象的 C++ 开发库,使用它进行应用程序的开发,可以充分利用其面向对象和模块化的特征,从繁琐的 X 编程中解脱出来,专注于程序本身的实现。

收稿日期:2004-02-13。王子强,硕士生,主研领域:信息安全技术。

(2) 独有的 Signal-Slot机制

Signal-Slot机制是 Qt最重要的特征。在 MS-Windows中,程序通过消息机制和事件循环来实现图形对象行为的触发和处理;在 Qt中,是在一个类中定义多个 Signal和 Slot, Signal就相当于“事件”,而 Slot就相当于响应事件的“方法”。为了实现“事件驱动”,需要将一个类的 Signal和另一个类的 Slot连接起来(使用 connect)。采用这种机制是一种安全可靠的方法,它允许回调。并支持对象之间在彼此不知道对方信息的情况下进行合作,这使 Qt非常适合于真正的构件编程。

(3) 跨平台的开发工具

Qt是一种跨平台的工具包,它对编程者隐藏了在处理不同窗口系统时的潜在问题,使程序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的细节问题。Qt的跨平台能力非常强,用它开发出来的软件几乎可以应用于所有的操作系统,如 Microsoft Windows 95/98/2000/XP, Microsoft Windows NT, Linux, Solaris, SunOS, HP-UX, Digital Unix (OSF/1), Irix, FreeBSD, BSD/OS, SCO 和 AIX等。

(4) 开发程序的方便性

Qt提供了一个可视化的开发工具 Qt Designer,使用该工具就像在 Windows中使用 Visual C++那样可以直接向项目中添加各种组件,而不需要一步一步地编写代码,这个特点是其它非可视化编程工具望尘莫及的。Qt Designer中包含全面的联机帮助文档,包括超文本方式的 QtManual, man帮助手册页和补充指南。这些帮助不论是对于初学者还是专业人士,无疑都是非常方便的。除此之外,Qt还为应用程序的本地化提供完全支持,所有用户界面文本都可以基于“消息翻译表”被译成各国语言。它还完全支持双字节的 16bit国际字符标准。

(5) 强大的功能

Qt的强大功能主要体现在以下几方面:拥有一套完整的组件,用户可以直接基于这些现成的组件开发出优秀的界面;拥有丰富的 API函数,大概有 250多个 C++类;优越的绘画功能和 2D/3D图形润色功能,Qt的绘画工具类 QPainter,在任意一个绘画设备上都可以润色图形,对于更高级的 2D/3D图形可以结合使用 OpenGL和 Qt,用 OpenGL就像用任意一个 Qt组件一样方便,比纯粹使用 OpenGL作的 3D图形更好。

(6) 高性能的工具

Qt是直接基于 Xlib的,并不依赖 Motif工具包,所以它的执行效率比一般的基于平台的代码的执行效率要高。

据此,使用 Qt Designer作为 Linux下图形用户界面应用程序的开发工具,应该是非常理想的。

3 用 Qt开发程序的方法

3.1 使用 Qt工具包开发程序的基本框架

Qt的可视化开发工具 Qt Designer是在 Qt 2.2版本以后才提供的,所以早期的图形用户界面开发主要使用系统提供的界面元素类。使用这种开发方法设计出来的程序一般需要编写三种类型的文件:头文件(.h)、实现文件(.cpp)和 main(.cpp)主函数文件;在头文件中主要是定义各种类。在实现文件中实现类的构造函数及成员函数。而在主函数文件中进行程序的启动和主窗口的设置等。

在 Qt中,“事件驱动”是通过 Signal-Slot机制实现的。Signal主要是各可视化组件产生的事件,这些事件通常是 Qt控件

预定义好的,如按钮单击事件 clicked等。Slot是对某个事件的处理方法,它也属于类的成员函数,它的编写和编写普通成员函数类似。其区别主要在于:编写 Slot需要在类中定义 Q_OBJECT宏,它的作用是让处理工具 moc知道这个类要使用“Signal-Slot机制”;在定义成员函数访问类型后加 slots关键字;slot的返回值类型必须为 void。

下面这段代码定义了 Slot函数 hello:

```
class classA: public QObject{
    Q_OBJECT
    Public slots:
        Void hello();
};
```

当我们需要处理某组件的 Signal事件并要求把定义好的 Slot作为响应事件的方法时,只要在实现文件中使用 connect函数把 Signal和指定的 Slot连接起来即可,connect的语法是:

```
connect(startobject, SIGNAL(signal()), targetobject, SLOT(slot))
```

其中, startobject是指向发出 Signal的组件的指针, targetobject是指向包含相应 Slot成员的对象指针。

使用 Qt编程有一个基本的步骤和框架。下面以开发一个包含主窗口的程序为例说明其一般步骤:

(1) 编写头文件

在头文件中要定义主窗口类,我们给它取名为 MainWindow,定义如下:

```
mainwindow.h
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <qapplication.h>
#include <qmainwindow.h>
class MainWindow: public QMainWindow
{
    Q_OBJECT
public:
    MainWindow(QWidget * parent = 0, const char * name); 构造函数
public slots:
    ..... 定义 slot函数
};
```

(2) 编写相应的实现文件

```
mainwindow.cpp
#include <mainwindow.h> 把类的定义包含进来
MainWindow::MainWindow(QWidget * parent, const char * name): QMainWindow(parent, name)
{
    setCaption("Example"); 设置窗口标题
    setBackgroundColor(white); 设置窗口背景为白色
    ..... 如需要可定义窗口内的各种组件及对事件定义 slot连接
}
```

(3) 编写 main()主函数

```
main.cpp
#include <mainwindow.h>
#include <qwindowstyle.h>
int main(int argc, int ** argv)
{
    QApplication a(argc, argv);
    a.setStyle(new QWindowsStyle);
```

```

MainWidget w;
w.setGeometry(150, 150, 300, 300);
a = new MainWidget(&w);
w.show();
return a.exec();
}

```

在 main 函数中首先产生一个 QApplication 类的实例,然后定义窗口类实例并把它作为程序的主窗口,再设置了窗口尺寸。最后调用 show() 和 exec() 方法,程序进入消息循环。

一般地,在类的定义和实现过程中,可以将类的定义部分和实现部分放在同一个 .cpp 文件中,但由于使用类时只需了解类的定义,因此编程时最好把类的定义和实现放在不同文件中,而且编写每一个类都应使用单独的头文件和实现文件,这样使用某个类时只需把它的定义包含进来即可。

(4) 编写 Makefile

为了使程序能够编译运行,必须编写一个 Makefile 文件,针对上面程序编写的 Makefile 文件如下:

```

Makefile
CC = g++      指定编译器
QTNC = $(QTDIR) /include  指定 include 所在位置
QTLB = $(QTDIR) /lib      指定 lib 所在位置
MOC = $(QTDIR) /bin/moc   指定 moc 所在位置
example: main.o mainwidget.o mainwidget moc.o 指明可执行文件
        example 依赖于 main.o mainwidget.o 和 mainwidget moc.o
        $(CC) main.o mainwidget.o mainwidget moc.o -o
example -lqt -L $(QTLB) -o 表示编译并输出可执行文件
main.o: main.cpp 指明目标文件 main.o 依赖于 main.cpp
        $(CC) -c main.cpp -I$(QTNC) -c 表示编译只输出目标
        代码,并不输出可执行文件
mainwidget.o: mainwidget.cpp
        $(CC) -c mainwidget.cpp -I$(QTNC)
mainwidget moc.o: mainwidget.cpp
        $(CC) -c mainwidget.cpp -I$(QTNC)
mainwidget moc.cpp: mainwidget.h
        $(MOC) mainwidget.h -o mainwidget moc.cpp 由于像 slot
和 signal 这样的关键字不属于 C++ 关键字,无法直接用 C++ 编译器
编译,因此需要进行转换,moc 就是完成这一功能的。

```

编完 Makefile 后,执行 make 即可编译程序,生成可执行文件 example 了。

3.2 使用 Qt Designer 工具快速开发图形界面程序

(1) 程序的工作原理

在 Qt Designer 的新版本中,为用户应用程序提供了工程管理,这样就可以把所有的 Form 组织在一起,不但方便了管理,而且在不同的 Form 之间可以共享图像和数据。通常,一个工程中包含以下几种类型的文件:

工程文件 (.pro)、窗体文件 (.ui)、窗体实现文件 (.ui.h) 和主函数 (.cpp) 等。其中,工程文件的作用是管理当前项目中所有的文件及其关系,窗体文件的作用是管理窗体中的各种界面元素及其属性。那么它们是如何工作的呢?见图 1,与 Qt Designer 的早期版本相比,在 Qt Designer 3.X 版本中,Qt Designer 除了与 .ui 文件 (如 form.ui) 进行读写外,又增加了 form.ui.h 文件。.ui.h 文件是一个普通的 C++ 文件,主要是 slot 方法的定制。.ui.h 文件由 Qt Designer 负责保持和与

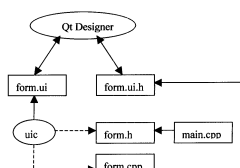


图 1

窗体相关的 slot 一致,无论用户何时给窗体添加、删除 slot 或是改变了 slot 的连接,Qt Designer 都会自动地更新 .ui.h 文件的内容。由另一工具 Uic (User Interface Compiler) 负责读取 .ui 文件 (如 form.ui) 和生成头文件与实现文件。在 main.cpp 中使用 #include form.h 将头文件包含进来,通常 main.cpp 都用来实例化 QApplication 对象和启动事件循环。

(2) 使用 Qt Designer 开发应用程序的一般步骤

这里仍以创建一个包含主窗口的程序为例进行说明:

① 创建新的项目

单击 File|New 打开新建文件对话框,选择 “C++ Project” 创建一个工程,然后单击 OK 进入工程设置窗口进行设置,完成后进入 Save As 对话框,选择一个合适的目录和文件名进行保存即可。

② 创建对话框

单击 File|New 打开新建文件对话框,单击 “Dialog” 后单击 OK 即可生成一个对话框。我们可以在 Property Editor 中修改其 “Name” 和 “Caption” 等属性,最后单击 File|Save 以默认的名字保存。

③ 添加和布置组件

在新建的窗体中根据需要添加各种组件,并修改其属性,然后再合理调整其布局。

④ 连接组件的 Signal-Slot

单击 Edit|Connections 打开连接编辑对话框,然后单击 New 进入一个新的连接,首先选择事件的产生者,然后选择事件的类型,再选择事件的接受者,最后选择或新建响应事件的 Slot。

⑤ 编写响应程序代码

在 Project Overview 窗口单击窗体所对应的 .ui.h 文件打开代码编辑窗口,在适当位置添加代码。

⑥ 添加 main 主函数文件

单击 File|New 打开新建文件对话框,选择 “C++ Main-File (main.cpp)”,然后单击 OK 进入 Main-File 配置对话框,接受默认文件名并选择主窗体后单击 OK 即可。

⑦ 保存并编译执行

编译方法: (如工程名为 metric.pro)

在命令行下转到工程所在目录,键入 qmake -o Makefile metric.pro 以运行 qmake,它能自动生成相应的 Makefile 文件。

在命令行下执行 make 命令编译程序,生成可执行文件 metric。

4 用 Qt Designer 实现 Linux 下主机漏洞扫描系统

我们在编写 Linux 主机漏洞扫描系统时,需要给用户提供一个可视化的管理界面,在综合比较 Linux 下多种开发工具 GTK、Kdevelop、Kylx 之后,我们选择了 Qt Designer。在本系统的实现过程中,不仅使用了多种可视化组件,而且涉及多进程、多线程的创建和使用,涉及与 C 脚本语言等程序设计语言的接口。实践表明,Qt Designer 确实是一个非常强大的、可视化的、面向对象的开发工具,是 Linux 下开发图形用户界面应用程序的理想工具。

参 考 文 献

- [1] 宋国伟编著, GTK+2.0 编程规范, 清华大学出版社。
- [2] Xteam (中国) 软件技术有限公司编著, Qt 程序设计, 清华大学出版社。
- [3] 李宋琛主编, Linux 面向对象窗口高级编程, 科学出版社。