

利用Python和GTK+的Linux GUI应用的快速开发

赵 毅, 马 捷

(中科院计算所国家智能计算机研究开发中心, 北京 100080)

摘 要: 介绍了Python语言、GTK+图形包、glade图形界面生成器以及用Python+GTK+开发Linux GUI应用的基本方法。并以机群网络通信监控系统的设计与实现为例, 对用Python+GTK+开发Linux GUI应用的方法进行了阐述。

关键词: 图形用户界面; Python; GTK+; glade; 线程; Popen3; rc文件

Rapidly Developing Linux GUI Application With Python and GTK+

ZHAO Yi, MA Jie

(National Research Center of Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

[Abstract] This paper introduces the Python language, the GTK+ graphical library, the GUI builder for GTK and the method of rapidly developing Linux GUI application with Python and GTK+, and explains the method using the development of the communication monitor system of cluster network for example.

[Key words] Graphical user interface(GUI); Python; GTK+; Glade; Thread; Popen3; rc file

Linux系统在近几年发展神速,但是由于其GUI应用程序开发方法及标准的不统一导致开发效率低、界面不友好。目前Linux应用软件领域亟需一种学习容易、成本低、效率高的GUI应用程序开发方法。Python是一种简单的、解释型的、交互式的、可移植的、面向对象的高级编程语言。GTK+是一个用于创造图形用户接口的图形库,它使用一些称为“构件”的对象来构造丰富的用户界面。本文介绍一种用Python+GTK+快速开发Linux GUI应用的方法,并通过一个开发实例对该方法进行了论述。

1 开发工具

1.1 Python语言

Python是一种不复杂但是很健壮的编程语言,它既具备传统编译型程序设计语言强大的功能和复杂性,又在某种程度上具备比较简单的脚本和解释型程序设计语言的易用性。Python有一个交互式的开发环境,因为Python是解释运行的,这大大节省了每次编译的时间。Python语法简单,且内置有几种高级数据结构,如字典、列表等,使用简单。Python具有大部分面向对象语言的特征,可完全进行面向对象编程。它可以在MS-DOS、Windows、Windows NT、Linux、Solaris、Amiga、BeOS、OS/2、VMS、QNX等多种OS上运行,是一种可移植的程序设计语言。

Python语言的这些特点使其成为很多应用的合理选择。通常情况下,安装好Linux之后,大部分散布套件都会自动安装Python。

1.2 GTK+、glade和libglade

当您希望复杂的用户交互,生成用户界面时,Python实际上什么也不能提供。就像Java通常为每个带JVM的平台提供Swing和AWT这些基本的GUI图形库一样,Python也必须和图形库相结合来开发GUI应用程序。但是Python没有标准的GUI库,开发人员可以按照自己的偏好选择TK、GTK+、QT和MFC等图形库搭配。其中,GTK+是在Gdk(GIMP Drawing Kit, GIMP绘图包)的基础上创建的一个用于创造图形用户接口的图形库,它提供了窗口、按钮、框架、列表

框、组合框、树、状态条等很多构件,可以构造丰富的用户界面。GTK+是基于LGPL授权的,因此常用GTK+开发开放源代码的自由软件或商业的非自由软件。

在用GTK+库开发Linux下的GUI时,开发人员需要编写两方面的代码,即应用于实现图形界面的代码和用于实现程序运行的代码,然后将两部分代码结合起来。如果有一种软件能将图形界面的设计及时展现于开发人员的面前,而且设计完之后能自动生成代码,就能使开发人员的主要精力集中于程序的应用上,同时提高开发的效率。目前GNOME下已经开发出了glade这一可视化的图形界面生成工具。

glade使用Gnome/GTK+图形构件来生成用户界面。通过glade,用户界面可以以XML文件形式保存也可以直接生成C代码。glade还可以定义用户界面消息处理函数的函数名。

libglade是由James Henstridge编写的一个库。通过libglade库提供的函数在运行Python程序时载入扩展名为glade的文件,产生由glade生成的XML文件代表的用户界面,并且自动绑定glade文件中定义的消息处理函数的句柄与Python程序中实现的消息处理函数的代码。图1说明了三者之间的关系。

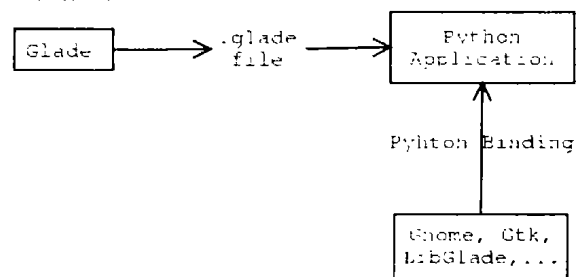


图1 Python、glade和libglade关系图

基金项目: 国家“863”计划基金资助项目(863-306-ZD01-01)

作者简介: 赵毅(1976—),女,硕士生,研究方向:计算机通信与测试;马捷,副研究员

收稿日期: 2003-04-02 **E-mail:** zy@gatekeeper.ncic.ac.cn

2 开发的基本方法

2.1 基本方法

无论在什么平台下,无论用什么开发工具,GUI应用程序的基本开发方法都是相似的。开发人员都要编写两方面的代码,即应用于实现图形界面的代码和用于实现程序运行的代码,然后再将这两部分的代码结合起来。在图形界面相对复杂的情况下,程序员就不得不将大量的精力集中于图形界面代码的编制与调试上。如果通过图形界面生成工具快速地生成图形界面,就能使程序员的主要精力集中于程序的应用上。这就是所谓的可视化编程思想。本文介绍的就是一种用glade生成图形界面,用Python实现程序运行代码的Linux GUI应用的快速开发方法。

2.2 设计界面

在Linux上运行xWindow,调出图形化界面后,在虚拟控制台上输入glade或在命令菜单上找到开发子菜单下的glade项,调出glade。调出glade后一般能看到3个窗口:一个工程管理窗口(主窗口),一个常用构件窗口和一个属性管理窗口。这3个窗口构成了一个GUI的开发环境。

生成用户界面的具体方法与Windows环境下,VB、Delphi等可视化开发环境用法相似。生成用户界面后,通过glade,用户界面可以以XML文件形式保存也可以直接生成C代码。glade还可以定义用户界面消息处理函数的函数名。

2.3 编写运行代码

glade的输出是一个XML格式的glade文件,它描述了生成的用户界面的构件和消息处理函数的函数名。然后通过libglade库提供的函数在运行python程序的时候载入扩展名为glade的文件,产生由glade生成的XML文件代表的用户界面,并且自动绑定glade文件中定义的消息处理函数的句柄与Python程序中实现的消息处理函数的代码。具体方法如下:

- (1)通过import libglade载入libglade库。
- (2)通过函数libglade.GladeXML(.glade)在运行时载入glade文件。参数是由glade生成的描述用户界面的glade文件。
- (3)通过函数get_widget('widget_name')获得界面构件的指针。参数为构件的名称。
- (4)用signal_autoconnect()绑定消息句柄与消息处理函数。glade文件中定义的消息处理函数的句柄与Python程序实现的消息处理函数的函数名一一对应。
- (5)用Python编写消息处理函数的代码,完成程序功能。

3 开发实例

下面介绍的机群网络通信监控系统是一个用Python+GTK+开发Linux GUI应用的实例。

作为国家“863”计划的重大研究成果——曙光3000超级服务器的后继,曙光Linux超级服务器由64个SMP节点组成,操作系统为Linux,机群内部各节点均通过高速的Myrinet网相连。Myrinet网卡通过自行开发研制的通信协议BCL(Basic Communicate Library)使用。本文所述的机群网络通信监控系统正是对该机群系统的通信情况进行监控,包括对节点网络设备状态信息的查询、网络阻塞状态和负载情况的实时监控等。

3.1 系统结构

机群网络通信监控系统从功能上可以分为4个模块:通信状态监控模块,数据采集模块,可视化显示模块和用户界面模块(如图2所示)。

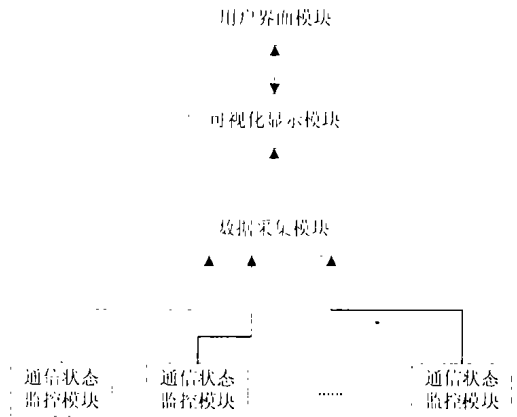


图2 监控系统模块结构图

3.2 系统实现

(1)设计用户界面

监控系统通过glade生成用户界面,并保存为XML格式,扩展名为glade的文件。用户界面如图3、图4所示。

All	Col0	Col1	Col2	Col3	Col4	Col5	Col6	Col7
Row0	Node0	Node1	Node2	Node3	Node4	Node5	Node6	Node7
Row1	Node8	Node9	Node10	Node11	Node12	Node13	Node14	Node15
Row2	Node16	Node17	Node18	Node19	Node20	Node21	Node22	Node23
Row3	Node24	Node25	Node26	Node27	Node28	Node29	Node30	Node31
Row4	Node32	Node33	Node34	Node35	Node36	Node37	Node38	Node39
Row5	Node40	Node41	Node42	Node43	Node44	Node45	Node46	Node47
Row6	Node48	Node49	Node50	Node51	Node52	Node53	Node54	Node55
Row7	Node56	Node57	Node58	Node59	Node60	Node61	Node62	Node63

图3 监控系统主界面

node	lanal-speed	State	ISR	Send-U	Send-S	Recv-U	Recv-S
Node13	133	0	016ea0c0	2	0	2	0
Node14	66	0	016ea0c0	3	0	1	0
Node15	133	0	016ea0c0	0	0	0	0
Node6	133	0	016ea0c0	0	0	0	0
Node8	133	0	016ea0c0	0	0	0	0
Node9	133	0	016ea0c0	0	0	0	0

图4 监控系统信息显示界面

(2)应用程序与用户界面的接口

用glade生成用户界面后,在运行Python程序的时候载入glade文件,获得界面上构件的指针,并通过signal_autoconnect()绑定glade文件中消息处理函数的句柄与Python程序中实现的处理函数的代码。以下是关键的程序代码:

```
/*载入libglade库*/
import libglade
/*运行时载入描述用户界面的glade文件*/
xml=libglade.GladeXML("./ddview.glade",
domain="ddview")
/*获得主窗口mainWindow及其它构件的指针*/
self.toplevel=xml.get_widget('mainWindow')
self.result_clist=xml.get_widget('resultCList')
```

(3) “多线程”通信机制

在Python语言中有一个Popen3的对象，它可以执行指定的shell命令，并把标准输出和标准错误输出返回到指定文件中。本系统对类Popen3进行重载，使得shell命令的执行结果存入指定的out和err字符串。

```
/* 调用shell命令*/
command=" ./rsh s 5" +node+" ./gr 0 1" +" </dev/null"
child=Popen3(command,Nonc,"stderr")
child.err和child.out就是调用command命令的错误输出和标准输出信息。

/* 调用timeout函数*/
self.timer[node]=timeout_add(5000,self.do_
check,node)
self.do_check函数中通过Popen3调用command命令,node是调用do_check函数时的输入参数。
```

(4)构件的动态配置

style "widget name"

其中, fg表示构件的前景色, bg表示设置构件的背景色, 还可以通过bg_pixmap和font设置构件的背景图片和字体。STATE表示构件的不同的状态。NORMAL表示构件的普通状态。PRELIGHT表示当鼠标放在构件上的状态。ACTIVE表示当构件被选中的状态。

```

/*表示通信停止的节点类型*/
style "stop_button"
{
    bg[PRELIGHT] = { 1.0, 1.0, 0 }
    bg[ACTIVE] = { 1.0, 1.0, 0 }
}
/*rc文件中定义stop_button类型*/
widget "**stop_button" style "stop_button"
在Python程序中调入rc文件，并设置构
rc_parse("./ddview.rc")
self.button[node].set name("stop button"

```

然后在Python程序中调入rc文件，并设置构件的颜色。如下所示：

4 结论

机群网络通信监控系统实现了对曙光Linux超级服务器机群网络的实时监控，在实际的工程应用中取得了不错的效果。本文以机群网络通信监控系统的实现为例，介绍了一种用Python+GTK+快速开发Linux GUI程序的方法，对Linux下开发GUI程序具有一定的借鉴意义。

用Python+GTK+开发Linux平台的图形用户界面应用程序,容易理解和学习、开发成本低、开发效率高,而且具有良好的移植性,这对于快速开发Linux GUI应用软件、丰富Linux平台应用软件的种类增强Linux应用软件功能,从而更好地普及Linux系统的应用都有巨大的推动作用。

参考文献

- 1 (美) Chun W J. 杨 涛, 王建桥, 杨晓云等译. Python核心编程[M]. 北京: 机械工业出版社, 2001
- 2 Python Library Reference [EB/OL]. <http://www.python.org/doc/current/lib/lib.html>
- 3 Python Reference Manual [EB/OL]. <http://www.python.org/doc/current/ref/ref.html>
- 4 PyGTK Tutorial [EB/OL]. <http://www.mocraiki.com/pygtkutorial/pygtkutorial/index.html>

☆☆

(上接第56页)

参考文献

- 1 Sun T D, Mudde R. Image Reconstruction of an Electrical Capacitance Tomography System Using an Artificial Neural Network[C]. 1st World Congress on Industrial Process Tomography, Buxton, Greater Manchester, 1999-04:14-17
- 2 肖化, 胡广莉. 基于分组BP神经网络的两相流电容层析技术[J]. 计量学报, 1998, 11(3)
- 3 蔡芳, 马宁. 电容层析成像的BP网络重建[J]. 武汉大学学报

- (自然科学版), 1997, 43(1)
- 4 阎平凡, 张长水. 人工神经网络与模拟进化计算[M]. 北京: 清华大学出版社, 2000
- 5 魏 颖, 王 师, 赵进创等. 电阻层析成像(ERT)图像重建算法的研究[J]. 东北大学学报, 2000, 21(1): 185-188
- 6 周 健. 一种改进的最近邻聚类算法[J]. 山东建材学院学报, 1999, 13(2)
- 7 潘正君. 演化计算[M]. 南宁: 广西科技出版社, 北京: 清华大学出版社, 1999