

大家一起用 gtk 编程 1(开始与热身)

大家一起用 gtk 编程 1(开始与热身)

转载请注明出处: <http://lvjinhua.cublog.cn>

作者: lvjinhua at gmail dot com

(本文档最后由杨小邪编辑整理)

2009.1.7

.写在最前面

笔者写做本文的目的, 主要是为初学GTK编程的新手们提供一个能够快速上手的学习档案, 能够通过自己将所有的代码输入并编译, 以达到学习的目的。同时, 也希望它成为一个参考手册, 希望GTK中的每一个API都能在本文中找到相应的示例。

笔者将尽力保证所有的代码都是最短小的, 这样才能保证大家都有足够的动力去手工输入它。

由于笔者时间及精力有限, 不能一次将所有内容呈现给所有读者, 因此拟定了一个计划: 每天更新一个章节的内容, 在笔者的Blog上进行连载, 同时会不断地对以前的内容进行修改和完善, 因此为了保证大家能够读到最新的内容, 请随时关注笔者的

Blog: <http://lvjinhua.cublog.cn>

起笔的日子: 2006.09.20

.为什么是 GTK?

为什么不是 GTK? 笔者不想花费太多的口水去辨证为什么选择 GTK 而不是其它的东西; 任何技术都没有好坏之分, 只有使用它的人有能力的强弱之别; 笔者认为比较明智的做法是: 多编些程序, 少做些争论。

.1、GTK 简介

GTK (GIMP Toolkit) 是一套用于创建图形用户界面的工具包。它遵循 LGPL 许可证, 所以可以用它来开发开源软件、自由软件, 甚至是封闭源代码的商业软件, 而不用花费任何费用来购买许可证和使用权。当前, GTK 已经被成功地应用到了大量的自由软件及商业软件中, 已经取得了很大的成功。

GTK 的创建者:

Peter Mattis: petm@xcf.berkeley.edu
Spencer Kimball: spencer@xcf.berkeley.edu
Josh MacDonald: jmacd@xcf.berkeley.edu

当前, GTK 主要由几大组件构成:

*Gtk+: GTK 的主要构件, 包括所有的图形控件及大量实用的 API。

*Glib: 包含一些标准函数的替代函数, 以及一些处理链表等数据结构的函数等。
这些替代函数被用来增强 GTK 的可移植性, 同时提供 libc 的对应函数的增强版本。

*Pango: 该组件用来处理国际化文字输出。

•2、预备

工欲xxx, 必先利其器! 在进行实际的程序设计之前, 应该先明确一些主要的开发工具。

1) GTK是跨平台的, 它能够运行于Linux/Unix/Windows/MacOS及嵌入式系统, 但这不是本文所关注的重点; 实际上, 只要按照GTK所提供的API去编程, 基本上在一个平台上设计的程序, 在其它平台可以不加任何修改而成功地进行编译。

笔者使用的操作系统是: [Dubuntu-6.06](#), 本文中所有的示例代码将在该平台上进行开发并编译运行; 当然, 这并不代表不能使用其它操作系统, 只要安装了GCC编译环境、GTK/Glib/Libc开发库, 基本上就能成功地编译运行本文中的所有示例, 现就Ubuntu Dapper 发行版GTK开发环境安装方法简述如下:

```
sudo apt-get install vim                #是的, 笔者使用 vim 来
编写代码, 当然您可以使用任何自己喜欢的编辑器
sudo apt-get install build-essential    #这将安装
gcc/g++/gdb/make 等基本编程工具
sudo apt-get install gnome-core-devel  #这将安装 libgtk2.0-dev
libglib2.0-dev 等开发相关的库文件
sudo apt-get install pkg-config          #用于在编译 GTK 程序时自动找
出头文件及库文件位置
sudo apt-get install devhelp            #这将安装 devhelp GTK 文
档查看程序
sudo apt-get install libglib2.0-doc libgtk2.0-doc
                                           #这将安装
gtk/glib 的 API 参考手册及其它帮助文档
sudo apt-get instal glade libglade2-dev #这将安装基于 GTK 的界面构造程
序
```

当然, 其它的 Linux 发行版可参考相应的文档进行开发环境的搭建, 各大 Linux论坛是个不错的咨询地: www.linuxsir.org, www.linuxfans.org

如果有热心的同仁能提供其它操作系统 GTK 开发库的安装方法，欢迎提供给我笔者，笔者将在这里分享给大家！先行谢过了。

Windows 平台以MinGW为例的环境设置：

来自[linxh 的Blog](#)

1. 从<http://www.gimp.org/~tml/gimp/win32/downloads.html>下载以下文件：

- [glib-2.12.1.zip](#). Runtime for GLib 2.12.1. Requires libiconv and gettext.
- [glib-dev-2.12.1.zip](#). Corresponding developer package, i.e. headers, import libraries and documentation.
- [gtk+-2.8.20.zip](#). Runtime for GTK+ 2.8.20. Requires GLib 2.12, atk 1.10, Pango 1.12 and cairo 1.2. In addition, if you want to be able to load PNG, JPEG or TIFF images with gdk-pixbuf, you will need libpng and zlib, libjpeg and libtiff respectively.
- [gtk+-dev-2.8.20.zip](#). Corresponding developer package.
- [pango-1.12.3.zip](#). Pango 1.12.3. Pango is used by GTK+ 2 and GIMP 2. Requires GLib 2.12 and cairo 1.2.
- [pango-dev-1.12.3.zip](#). Corresponding developer package.
- [atk-1.10.3.zip](#). Atk is used by GTK+ 2.
- [atk-dev-1.10.3.zip](#). Corresponding developer package.
- [cairo-1.2.2.zip](#). Cairo is used by GTK+ 2 and Pango.
- [cairo-dev-1.2.2.zip](#). Corresponding developer package.
- [pkg-config-0.20.zip](#). [Pkg-config](#) is a neat program that is useful in makefiles etc and extensively used in building software according to the GTK+ and GNOME conventions, using autotools etc. It uses a "database" specifying interdependencies among software packages. It is used to get the compile and link flags needed when building software using libraries that provide pkg-config data. Requires GLib 2.x.
- [GNU libiconv](#) for Win32. Both runtime and developer files. [GNU Libiconv](#) is an implementation of [iconv](#). GLib uses libiconv, so you will need this.
- [gettext-0.14.5.zip](#). Just the GNU libintl (internationalization library) DLL (intl.dll). GLib uses libintl, so you will need this.

- [gettext-dev-0.14.5.zip](#). GNU gettext developer files and tools for Win32, containing the header libintl.h, msgfmt.exe, other programs, documentation, and various other files. libintl.h is needed if you build software that uses GNU gettext for internationalization. Msgfmt is needed if you want to recompile message catalogs. The other commands can be used to manipulate message catalogs in other ways.

解压到一个目录下（以 c:\gtk 为例）。

2. 从<http://gnuwin32.sourceforge.net/packages/libpng.htm>下载 libpng，安装至 c:\gtk 下。

3. 编译：

- (1) 设置环境变量 set PKG_CONFIG_PATH=C:\gtk\lib\pkgconfig
- (2) 运行 pkg-config --cflags --libs gtk+-win32-2.0
- (3) 编译时加上上面的输出作为参数

•3、Hello, Dubuntu!

嗯，早就急不可待了，让我们开始我们的第一个程序吧！

本程序的主要功能：显示一个窗口，并在窗口的标题栏显示字符串“Hello, Dubuntu!”

源代码如下所示，编译成功后，可以通过命令行输入 ./hello_dubuntu 来运行此程序，显示效果如下：

hello_dubuntu.c

```
#include<gtk/gtk.h>
/* 一般可以在 /usr/include/gtk-2.0 下找到上边的头文件
 * 上边的头文件的作用是包含进行 GTK 编程所有可能用到的头
 * 文件，包括 glib.h 等
 */
int
main(int argc, char *argv[])
{
    GtkWidget *window;
    // GtkWidget 是绝大部分可视组件的基类
    gtk_init(&argc, &argv);
    //对程序传入的命令行参数进行标准化处理
```

```

        window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
        //创建窗口(200x200 大小)
        gtk_window_set_title(GTK_WINDOW(window),
"Hello, Ubuntu!");
        //设置窗口标题
        gtk_widget_show(window);
        //显示窗口
        gtk_main();
        //Gtk 程序主循环
        return 0;
}

```

编译程序: `gcc hello_dubuntu.c -o hello_dubuntu `pkg-config --cflags --libs gtk+-2.0`` # 注意: “`” 不是普通的单引号 “'”, 而是同“~”在一起的那个符号!

是的, 上边的程序非常简单, 除了注释就没几条语句了! 但它确实是一个 GTK 程序, 虽然它还不够完善, 缺少信号处理, 因此当您单击“关闭”按钮关闭此窗口后, 实际上程序还在运行。

1) 编译程序的命令行:

如上, 使用 GNU C 语言编译器 `gcc`, 其中 `pkg-config --cflags --libs gtk+-2.0` 用于生成 `gcc` 编译及链接程序时所必须的头文件及库文件列表, 在笔者的机器上运行该命令得到的结果如下:

命令: `pkg-config --cflags --libs gtk+-2.0`

输出:

```

-I/usr/include/gtk-2.0 -I/usr/lib/gtk-2.0/include
-I/usr/include/atk-1.0 -I/usr/include/cairo
-I/usr/include/pango-1.0 -I/usr/include/glib-2.0
-I/usr/lib/glib-2.0/include -lgtk-x11-2.0 -lgdk-x11-2.0 -latk-1.0
-lgdk_pixbuf-2.0 -lm -lpangocairo-1.0 -lfontconfig -lXext -lXrender
-lXinerama -lXi -lXrandr -lXcursor -lXfixes -lpango-1.0 -lcairo -lX11
-lgobject-2.0 -lgmodule-2.0 -ldl -lglib-2.0

```

2) 大家如果对某个 GTK 标准 API 定义, 或者某个类型/常量定义不清楚, 可以在 `devhelp` 帮助程序中进行搜索, 以查看某参数说明。

3) `gtk_init(&argc, &argv)` 函数用于对传入进来的标准命令行参数进行解析并处理, 这些标准命令行参数为 (即: 运行我们的程序时, 通过命令行传入的下列参数会被自动处理):

- `--gtk-module`
- `--g-fatal-warnings`

- gtk-debug
- gtk-no-debug
- gdk-debug
- gdk-no-debug
- display
- sync
- name
- class

4) `gtk_widget_show(window)` 用于告诉程序， 我们的 `window` 构件已经准备完成， 可以显示了。

5) `gtk_main()` ， 程序将会运行到此， 然后等待“事件”的发生；当有事件发生后， 将调用此事件对应的回调函数， 当回调函数执行完毕， 又会重新回到 `gtk_main()` ， 等待新的“事件”发生。

6) 宏 `GTK_WINDOW(window)` 是用于进行类型检查及转换， 它将把一个可以转化为 `GtkWindow` 的构件强制转换为 `GtkWindow` 类型后传递给函数。

好了， `hello_dubuntu.c` 理解了吗？ 什么！ 没有理解！ 不要紧， 我们下边还会有很多这样的小例子， 再多输入几个就会理解了， 相信我！