

大家一起用 gtk 编程 2(添加按钮和连接事件)

大家一起用 gtk 编程 2(添加按钮和连接事件)

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

.4、加入按钮和事件处理函数

上回说道,咱们使用 GTK 创建了一个最简单的 GTK 程序,简单到只能显示一个 200x200 点大小的空白窗口,并在标题栏显示“Hello,Dubuntu!”的字样! 虽然简单,但确实是个不错的开端,好了,咱们现在在窗口中添加一个按钮。

运行效果:

代码:

hello_dubuntu2.c	
/* 本例的主要目的是在窗口中显示一个按钮, * 并且单击按钮退出程序 */ #include<gtk/gtk.h> void cb_button(GtkWidget *widget, gpointer data) {// 按钮“button”的回调函数 gtk_main_quit(); } int main(int argc, char *argv[]) { GtkWidget *main_window; //主窗口对象 GtkWidget *button; //将要放置到主窗口中的按钮对象 gtk_init(&argc, &argv);	

```

        main_window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
        gtk_window_set_title(GTK_WINDOW(main_window),
"Hello, Dubuntu2!");
        //设置窗口的默认大小（宽 200，高度 50）
        gtk_window_set_default_size(GTK_WINDOW(main_window),
200, 50);

        button = gtk_button_new_with_label("退出程序");
        gtk_container_add(GTK_CONTAINER(main_window), button);
        //为"button"连接“单击事件”要调用的回调函数
        g_signal_connect(G_OBJECT(button), "clicked",
G_CALLBACK(cb_button), NULL);

        gtk_widget_show(button);
        gtk_widget_show(main_window);
        //上边的两句可以合为 gtk_widget_show_all(window)

        gtk_main();
        return 0;
}

```

编译： gcc -o hello_dubuntu2 hello_dubuntu2.c `pkg-config --cflags --libs gtk+-2.0`

程序注释：

1) 在 GTK 中，默认添加到窗口中的控件会自动占满所有的空间，因此当我们的程序运行时，将窗口放大时按钮也跟着放大了；这点同 Windows 及 Qt 是不相同的。正因为如此，在 GTK 控件库中有几个控件专门用来进行窗口布局，比如说 vbox, hbox, table, GtkPixed 等，通过对它们的灵活操作，可以得到一个非常漂亮的应用程序界面！这将在以后进行详细介绍。

2) 关于事件与回调函数

事件：GTK 同许多其它的图形编程库相似，都采用事件驱动方式来工作，这就关系到出现什么事件做什么工作的问题。（在 GTK 中还有个“信号”的概念，同“事件”的概念不容易分清楚，特别是刚开始学习时，不过没关系，初期我们就把它们当一个概念来理解），在 GTK2.0 中，一般使用 g_signal_connect() 函数来注册每个对象和其对应的处理函数，如本例所示。

回调函数：实际上就是一个普通的函数，不过它并不会被显示地直接调用，而是把它的地址注册到另一个函数，在那个函数中间接地对它进行调用，这也是回调的由来。

3) 在本例中，我们通过下列语句为“button”注册了一个“clicked”事件的回调函数“cb_button”，前缀“cb”即“call back（回调）”的意思。

```

g_signal_connect(G_OBJECT(button), "clicked",
G_CALLBACK(cb_button), NULL);

```

由于注册了上边的回调函数，因此当我们用鼠标单击按钮时，就会激发按钮的“clicked”事件，因此会调用 `cb_button()` 函数，而此函数中的语句 `gtk_main_quit()` 的作用是退出程序，因此我们单击按钮后整个程序就会退出。

4) 当然，这个程序还有个小“bug”，即直接单击右上角的“关闭按钮”关闭程序后，虽然主窗口消失了，但程序并没有直接地退出，还驻留在内存中。这是因为当单击右上角的“关闭按钮”关闭程序后，会产生“destroy”信号，而我们并没有为此信号使用 `g_signal_connect()` 函数注册相关的处理函数，因此它的行为并不是我们所期望的；解决方法就是为主窗口的“destroy”信号注册一个回调函数，并在此回调函数中调用 `gtk_main_quit()` 以便退出程序，将如下代码添加到 `hello_dubuntu2.c` 的 `gtk_main()` 函数调用的上边就行：

```
g_signal_connect(G_OBJECT(main_window), "destroy",
G_CALLBACK(cb_button), NULL);
```

请注意：在这里我们再一次使用了“`cb_button`”函数，这是因为它的行为正是我们想要的，即“调用此回调函数后，退出程序”。

好了，今天的内容差不多了，很多内容都没有深入研究，不过没关系，慢慢来，以后将要以专题的形式对 GTK 的每个部件及 API 进行深入并细致的说明。如果您对本例中的一些概念没有完全理解也没有关系，继续下一步，很容易理解的！千万不要呆在原地不动。

下回预告：将简要说明如何编写简单的 Makefile 文件来组织我们的源程序，另外还将介绍如何利用 gdb 对我们的程序进行调试！精彩不容错过。

附：添加“destroy”信号处理后的 `hello_dubuntu2.c`

```
hello_dubuntu2.c

/* 本例的主要目的是在窗口中显示一个按钮，
 * 并且单击按钮退出程序
 */
#include<gtk/gtk.h>

void
cb_button(GtkWidget *widget, gpointer data)
{
    // 按钮“button”的回调函数
    gtk_main_quit();
}

int
main(int argc, char *argv[])
```

```

{
    GtkWidget *main_window; //主窗口对象
    GtkWidget *button;      //将要放置到主窗口中的按钮对象

    gtk_init(&argc, &argv);

    main_window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title(GTK_WINDOW(main_window),
"Hello, Dubuntu2!");
    //设置窗口的默认大小（宽 200，高度 50）
    gtk_window_set_default_size(GTK_WINDOW(main_window),
200, 50);

    button = gtk_button_new_with_label("退出程序");
    gtk_container_add(GTK_CONTAINER(main_window), button);
    //为"button"连接“单击事件”要调用的回调函数
    g_signal_connect(G_OBJECT(button), "clicked",
G_CALLBACK(cb_button), NULL);

    gtk_widget_show(button);
    gtk_widget_show(main_window);
    //上边的两句可以合为 gtk_widget_show_all(window)

    g_signal_connect(G_OBJECT(main_window), "destroy",
G_CALLBACK(cb_button), NULL);
    gtk_main();
    return 0;
}

```

编译： gcc -o hello_dubuntu2 hello_dubuntu2.c `pkg-config --cflags --libs gtk+-2.0`