

## 大家一起用 gtk 编程 5(窗口布局初探)

# 大家一起用 gtk 编程 5(窗口布局初探)

---

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

## .7、窗口布局初探

上回说到如何使用 gdb 来调试我们的程序,这回简单地介绍下如何有效地进行窗口布局,这也将是 gtk 编程基础入门的最后一部分;在接下来的日子里,笔者将针对 gtk 中的所有内容分专题,对每一部分都进行详细并且深入的描述,好了,进入今天的主题。

运行效果图:

hello_box.c
<pre>/* 本例的主要目的是在主窗口中旋转一个 box 控件,再在 box 控件中  * 放置一个单行编辑框和一个按钮  * 并且单击按钮退出程序  */ #include&lt;gtk/gtk.h&gt;  //按钮“button”的回调函数 void cb_button(GtkWidget *widget, gpointer data) {     //cb = call back     g_print("你单击了按钮\n");     gtk_main_quit(); }  int main(int argc, char *argv[]) {     GtkWidget *main_window; //主窗口对象     GtkWidget *hbox, *button, *editor;      gtk_init(&amp;argc, &amp;argv);</pre>

```

        main_window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
        g_signal_connect(G_OBJECT(main_window), "destroy",
G_CALLBACK(gtk_main_quit), NULL);
        gtk_window_set_title(GTK_WINDOW(main_window),
"Hello, Dubuntu2!");
        gtk_window_set_default_size(GTK_WINDOW(main_window),
200, 50);

        button = gtk_button_new_with_label("退出程序");
        g_signal_connect(G_OBJECT(button), "clicked",
G_CALLBACK(cb_button), NULL);

        editor = gtk_entry_new();

        hbox = gtk_hbox_new(FALSE, 10);

        gtk_box_pack_start(GTK_BOX(hbox), editor, TRUE, TRUE, 10);
        gtk_box_pack_start(GTK_BOX(hbox), button, FALSE, FALSE, 10);
        gtk_container_add(GTK_CONTAINER(main_window), hbox);

        gtk_widget_show_all(main_window);

        gtk_main();
        return 0;

```

编程程序: gcc -o hello\_box hello\_box.c `pkg-config --cflags gtk+-2.0`  
 -g -Wall  
 运行程序: ./hello\_box

组件之间的关系:

程序注释:

1) 几个新增函数:

- `g_print("你单击了按钮\n");` : 这是 `glib` 库提供的面向控制台的输出函数, 其功能同标准 C 的 `printf()` 差不多, 但在 `gtk` 程序中使用 `g_print()` 更具可移植性。
- `gtk_entry_new()`: 构造一个“单行文本输入框”对象并返回此对象的指针。
- `gtk_hbox_new()`: 构造一个“水平”的盒子, 用来装 单行输入框和按钮“
- `gtk_box_pack_start()`: 往盒子中”装“东西(这里是 `enter` 和 `button`)

- `gtk_widget_show_all(main_window)`: 显示 `main_window` 中的所有对象。

小道消息: 在 gtk 中, 所有创建对象的函数都是 "gtk\_对象名\_new()" 的形式

## 2) 什么是 box?

box 是一种不可视 (就是看不见) 的对象, 为什么会有 box 呢?

Gtk 在对待控件大小这点上, 同 VC/VB/Delphi/PB/Qt 等都是不一样的, 在前边这几个语言中, 一般情况下, 界面设计好后, 其所有对象的大小都是固定的, 也就是说不会因为屏幕分辨率或窗口大小发生变化而自动适应 这些变化 (要实现自适应的功能也很麻烦)。而在 gtk 或者 Java 中, 控件在窗口中的位置是个相对值, 程序将根据当前环境自动调整控件的新位置, 具体的调整方法是可以通程序预设的。

box 是一种 "长条形" 的盒子, 有 "垂直" 和 "水平" 两种, 分别使用 `gtk_hbox_new()` 和 `gtk_vbox_new()` 来创建, 可以往 "盒子" 里面放入一个或多个其它的控件, 这是使用 `gtk_box_pack_start()` 函数实现的, 当然, "盒子" 里面也可以放入其它的 "盒子", 这就是 gtk 界面布局的关键, 一般的 gtk 程序都是这样盒子套盒子再套盒子, 在各个盒子中放入按钮, 输入框, 菜单等控件, 这样就构成了一个复杂的界面。

当然, 界面布局也可以使用 `table`, 但用得最多的还是 `box`。

## 3) `gtk_hbox_new()`

```
GtkWidget* gtk_hbox_new (gboolean homogeneous,  
                          gint spacing);
```

功能: 创建一个新的水平 box 控件

参数: `homogeneous`: 如果值为 "TRUE" 则所有放进去的控件都分配相同的大小

`spacing`: 相邻控件之间的间距, 单位为像素 (pixel)

返回值: 新创建的控件的指针。、

与之对应的 `gtk_vbox_new()` 用法完全相同。

## 4) `gtk_box_pack_start()`

```
void gtk_box_pack_start (GtkBox*box,  
                        GtkWidget*child,  
                        gboolean expand,  
                        gboolean fill,  
                        guint padding);
```

功能: 往 "box" 中放入一个控件

参数: `box`: 就是往这个 "box" 中放入新的控件

`child`: 就是放入这个控件

`expand` 和 `fill`: 笔者始终认为这两个参数只可意会, 不可言传

padding: 指示同 “child” 控件相邻控件之间的间距（单位为像素 pixel）
---

#### 5) GTK\_BOX(object) 宏

这个宏用来将 object 参数强制转换为 GtkWidget 对象（当然，也进行类型检查），gtk 中所有的对象都有相似的宏，比如 GtkWidget 对象有 GTK\_WINDOW() 宏，GtkButton 对象有 GTK\_BUTTON() 宏等。

好了，今天的内容就到这里，读者可以自己修改 “gtk\_hbox\_new 和 gtk\_box\_pack\_start” 中的各个参数的值，然后重新编译运行程序，根据运行效果来体会各个参数的关系吧。

推荐：devhelp 帮助，在编程中有不清楚的 gtk 函数或宏，可随时通过它查询。

入门部分结束，笔者的文笔是很烂的，也不知道前边的内容写得怎么样，希望大家能够看明白；在接下来的日子里，我们会分别详细地介绍各个 gtk 控件（可视的与不可视的），并且尽可能写些小示例来演示一些控件或接口函数的功能；在介绍完控件后，还有 glib 专题，”国际化/本地化“专题， glade 专题，解析 hypersrc 这个源代码浏览软件。