Programmazione di GUI con GTK – parte 2

by Özcan Güngör
<ozcangungor(at)netscape.net>

About the author:
Uso Linux dal 1997. Libertà, flessibilità e opensource. Queste sono le caratteristiche che mi piacciono.

Abstract:

In questo articolo discuteremo di box e tabelle. Con questi saremo in grado di posizionare i componenti nel giusto ordine dentro le finestre. Per poter comprendere questo articolo dovreste conoscere le seguenti caratteristiche del linguaggio C:

- Variabili
- Funzioni
- Puntatori

Inoltre è consigliato leggere l'articolo precedente.

Impacchettare Componenti nei Box

Impacchettare componenti significa posizionarli nel giusto ordine dentro le finestre. Uno dei metodi che GTK mette a disposizione è usando i box. L'idea principale dietro i box è di impacchettare i componenti in ordine verticale o orizzontale. Ci sono due tipi di box: box orizzontali e box verticali. Vediamo di spiegarli:

Box Orizzontali

In questo tipo di box i componenti vengono impacchettati orizzontalmente. La seguente funzione viene usata per creare un box orizzontale.

```
gtk_widget *box;
```

```
box=gtk_hbox_new(gboolean homogenous, gint spacing);
```

dove il parametro *homogeneous* viene usato per definire se i componenti vengono distribuiti in modo omogeneo o meno: se è **TRUE** i componenti si allargano in modo da avere una distanza uguale tra di loro, mentre se è **FALSE** vengono impacchettati fianco a fianco. *spacing* viene usato per definire lo spazio minimo tra i componenti.

Box Verticali

In questo tipo di box i componenti vengono impacchettati in verticale. La seguente funzione viene usata per creare un box verticale:

```
gtk_widget *box;
box=gtk_vbox_new(gboolean homogenous, gint spacing);
```

I parametri di questa funzione hanno lo stesso significato della funzione precedente.

Proprietà comuni dei Box

L'aggiunta di componenti può essere effettuata in due modi: il primo è:

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

Usando questa funzione aggiungiamo componenti nel box (a sinistra per un box orizzontale, sopra per uno verticale). *box* è il box a cui vogliamo aggiungere un componente. *child* è il componente da aggiungere. *expand* viene usato per allargare le dimensioni dei componenti in modo che usino tutto lo spazio disponibile. *padding* viene usato per aggiungere spazio sui lati.

La funzione complementare a gtk_box_pack_start è gtk_box_pack_end:

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

Questa funzione ci permette di aggiungere componenti alla fine (destra o sotto) di un box. I parametri hanno lo stesso significato della funzione precedente.

Per inserire un box in una finestra usiamo la seguente funzione:

```
gtk_container_add (GtkContainer *container,GtkWidget *component);
```

container è la finestra a cui verrà aggiunto il box, *component* è il box da aggiungere. Per esempio, per aggiungere il box che abbiamo creato prima alla *window*, usiamo il seguente comando:

```
gtk_container_add(GTK_CONTAINER(window),box);

gtk_box_set_homogeneous (GtkBox *box, gboolean homogenous);
gtk_box_set_spacing(GtkBox *box, gint spacing);
```

La prima delle funzioni qui sopra viene usata per cambiare la proprietà homogeneous di un box, e la seconda per cambiare la spaziatura del box. *box* è il box da modificare.

```
gtk_box_set_child_packing(GkBox *box,GtkWidget *shild,
       gboolean expand, gboolean  fill, guint padding,
       GtkPackType packingtype);
```

Questa funzione ridefinisce le proprietà di un componente già impacchettato. I parametri hanno lo stesso significato della funzione gtk_box_pack_start. *packingtype* può essere GTK_PACK_START o GTK_PACK_END. GTK_PACK_START fa in modo che il componente venga posizionato all'inizio del box se viene impacchettato con gtk_pack_end. GTK_PAC_END fa in modo che venga posizionato alla fine del box se viene impacchettato con gtk_box_pack_start.

Per capire meglio quello che abbiamo spiegato, provate kutular.c.

Tabelle

Le tabelle, come nel codice HTML, ci aiutano a fissare i componenti in celle. Per poterlo fare è sufficiente creare una tabella con un numero sufficiente di righe e colonne. Quindi possiamo posizionare un componente in una cella o in un gruppo di celle (consentito solo se le celle sono affiancate). Per creare una tabella viene usata la seguente funzione:

```
GtkWidget *table;
GtkWidget* gtk_table_new(guint  row, guint column, gboolean homogenous);
```

row è il numero di righe, *column* è il numero di colonne. *homogeneous* viene usato per distribuire in modo omogeneo i componenti.

Per aggiungere componenti a una tabella viene usala la seguente funzione:

```
void gtk_table_attach (GtkTable *table, GtkWidget *child,
         guint left_attach, guint left_attach, guint top_attach,
         guint bottom_attach, GtkAttachOptions xoptions,
         GtkAttachOptions yoptions, guint xpadding, guint ypadding);
```

table è la tabella a cui saranno aggiunti i componenti e *child* è il componente da aggiungere. *left_attach* è il numero di celle a sinistra del componente, *right_attach* è il numero di celle a destra del componente. *top_attach* è il numero di celle sopra il componente e *bottom_attach* è il numero di celle sotto il componente. I componenti possono coprire una o più celle.

xoptions e *yoptions* possono assumere tre diversi valori: GTK_FILL, GTK_EXPAND, GTK_SHRINK. GTK_FILL fa in modo che il componente riempia la cella (o le celle). GTK_EXPAND fa in modo che venga centrato nella cella e GTK_SHRINK fa in modo che il componente venga ristretto alle dimensioni della cella se questa è più piccola. *xoptions* applica questi cambiamenti solo sull'asse x, mentre *yoptions* li applica solo all''asse y.

xpadding inserisce dello spazio a sinistra e a destra del componente lungo l'asse x mentre *ypadding* lo fa lungo l'asse y.

Ecco un codice di esempio:

```
#include <gtk/gtk.h>
```

```
void delete_event( GtkWidget *widget,GdkEvent  *event,gpointer data )
{
    gtk_main_quit ();
}

int main( int   argc,char *argv[] )
{
    GtkWidget *window;
    GtkWidget *button;
    GtkWidget *table;

    gtk_init (&argc, &argv);

    window = gtk_window_new (GTK_WINDOW_TOPLEVEL);

    gtk_signal_connect (GTK_OBJECT (window), "delete_event",
                        GTK_SIGNAL_FUNC (delete_event), NULL);

    table = gtk_table_new (2, 2, TRUE);

    gtk_container_add (GTK_CONTAINER (window), table);

    button = gtk_button_new_with_label ("button 1");
    gtk_table_attach(GTK_TABLE(table), button, 0, 1, 0, 2,GTK_SHRINK,
                     GTK_SHRINK,0,0);
    gtk_widget_show (button);

    button = gtk_button_new_with_label ("button 2");
    gtk_table_attach (GTK_TABLE(table), button, 1, 2, 1, 2,
                      GTK_SHRINK,GTK_SHRINK,0,0);
    gtk_widget_show (button);

    button = gtk_button_new_with_label ("button 3");
    gtk_table_attach (GTK_TABLE(table), button, 1, 2, 0, 1,
                      GTK_SHRINK,GTK_SHRINK,0,0);
    gtk_widget_show (button);

    gtk_widget_show (table);
    gtk_widget_show (window);

    gtk_main ();

    return 0;
}
```

gtk_table_attach ha molti parametri, quindi una nuova funzione, più corta, è stata creata: gtk_table_attach_defaults. Questa esegue lo stesso compito ma con meno parametri.

```
void gtk_table_attach_defaults (GtkTable *table,GtkWidget *child,
                guint left_attach, guint right_attach, guint top_attach,
                guint bottom_attach);
```

Qui i parametri hanno lo stesso significato. *xoptions* e *yoptions* prendono i valori GTK_FILL|GTK_EXPAND. *xpadding* e *ypadding* hanno valore 0.

La seguente funzione viene usata per cambiare il numero di righe e colonne in una tabella esistente:

```
void gtk_table_resize(GtkTable *table, guint rows, guint columns);
```

Usando le seguenti funzioni potete cambiare la spaziatura di una riga o di una colonna:

```
void gtk_table_set_row_spacing (GtkTable *table, guint row,
                                guint spacing);
void gtk_table_set_col_spacing (GtkTable *table, guint column,
                                guint spacing);
```

Le seguenti funzioni cambiano la spaziatura di intere righe o colonne:

```
void gtk_table_set_row_spacings (GtkTable *table, guint spacing);
void gtk_table_set_col_spacings (GtkTable *table, guint spacing);
```

Le seguenti funzioni cambiano il valore di homogeneous di una tabella esistente:

```
void gtk_table_set_homogeneous (GtkTable *table, gboolean homogenous);
```

Conclusioni

In questo articolo abbiamo imparato come impacchettare componenti e quindi abbiamo dato uno sguardo ad alcune funzioni per la modifica delle proprietà di box e tabelle. Sono sempre felice di ricevere domande, commenti e idee dai lettori. Mandatemi un'e−mail...
