

Programmer des interfaces graphiques avec GTK, Partie II



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



Résumé:

L'auteur:

J'utilise Linux depuis 1997. Liberté, flexibilité et opensource. Ce sont les propriétés que je préfère.

Traduit en Français par:

Frédéric Nourry
<frederic.nourry(at)laposte.net>

Dans cet article, nous allons aborder les boîtes et les tables. Grâce à elles, nous serons capables de disposer les composants là où nous le voulons dans les fenêtres. Pour bien comprendre ces articles, vous devez posséder les notions suivantes de programmation en langage C:

- Variables
- Fonctions
- Pointeurs

De plus, il est recommandé de lire l'article précédent.

Organiser les composants grâce à des boîtes

"Packer" des composants signifie placer ces composants dans l'ordre souhaité dans une fenêtre. Une des façons de faire cela avec GTK consiste à utiliser des boîtes. L'objectif principal des boîtes est d'organiser les composants de façon à les placer verticalement ou horizontalement. Il y a deux types de boîtes: les horizontales et les verticales. Voyons en détails ces deux types :

Boîtes horizontales

Dans ce type de boîte, les composants sont disposés horizontalement. La fonction suivante est utilisée pour créer une boîte horizontale.

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

où le paramètre *homogeneous* est utilisé pour indiquer si les composants sont placés de façon homogènes ou non: si oui (**TRUE**), alors les composants remplissent la totalité de la boîte à égale distance les uns des autres, et si non (**FALSE**), ils sont placés côte à côte. *spacing* est utilisé pour définir l'espace minimal entre les composants.

Boîtes verticales

Avec ce type de boîte, les composants sont disposés verticalement. La fonction suivante est utilisée pour créer une boîte verticale.

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

Les paramètres de cette fonction ont la même signification que pour 'gtk_hbox_new'.

Propriétés communes des boîtes

Ajouter des composants peut être fait de deux façons: la première est la suivante :

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

Avec cette fonction, nous pouvons ajouter des composants dans une boîte (sur le coté gauche pour une boîte horizontale ou en haut pour une boîte verticale). *box* est la boîte à laquelle nous voulons ajouter un composant. *child* est le composant à ajouter. *expand* est utilisé pour étendre la taille du composant de manière à ce qu'il prenne tout l'espace disponible. *padding* sert à ajouter un espace additionnel sur la gauche et la droite.

La fonction complémentaire de `gtk_box_pack_start` est `gtk_box_pack_end`

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

Cette fonction nous permet d'ajouter des composants à la fin (à droite ou en bas) d'une boîte. Les paramètres ont la même signification que pour la fonction précédente.

Pour ajouter une boîte dans une fenêtre, la fonction suivante est utilisée:

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

container est la fenêtre dans laquelle sera ajoutée la boîte, *component* est la boîte ajoutée. Par exemple, pour ajouter la boîte créée plus tôt à la fenêtre *window*, on utilise la commande suivante :

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

La première fonction ci-dessus est utilisée pour changer la propriété d'homogénéité d'une boîte et la deuxième pour changer l'espacement des composants dans une boîte. *box* est la boîte à modifier.

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

Cette fonction redéfinit les propriétés d'un composant déjà positionné. Les paramètres ont la même signification que dans la fonction `gtk_box_pack_start`. *packingtype* peut être `GTK_PACK_START` ou `GTK_PACK_END`. `GTK_PACK_START` positionne le composant au début de la boîte s'il a été disposé en utilisant la fonction `gtk_pack_end`. A l'inverse, `GTK_PACK_END` positionne le composant à la fin de la boîte s'il a été disposé par la fonction `gtk_pack_start`.

Pour mieux comprendre ceci, essayez le code suivant : `kutular.c`.

Les tables

Les tables, comme en HTML, nous aident à placer des composants dans des cellules. Pour cela, il suffit de créer une table possédant assez de lignes et de colonnes. Ensuite, nous pouvons placer un composant dans une cellule ou un groupe de cellules (seulement pour celles situées côte à côte). Pour créer une table, on utilise la fonction suivante:

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

row est le nombre de lignes, *column* est le nombre de colonnes, *homogenous* est utilisé pour disposer les composants de façon homogène.

La fonction suivante est utilisée pour ajouter un composant à une table:

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

table est la table dans laquelle seront ajoutés les composants et *child* est le composant à ajouter. *left_attach* est le numéro de la cellule de gauche où le composant sera ajouté. *right_attach* est le numéro de la cellule de droite où le composant sera ajouté. *top_attach* est le numéro de la cellule du dessus où le composant sera ajouté et *bottom_attach* celle du dessous. Des composants peuvent occuper plus d'une cellule.

xoptions et *yoptions* peuvent avoir trois valeurs différentes : `GTK_FILL`, `GTK_EXPAND`, `GTK_SHRINK`. Avec `GTK_FILL` le composant occupe une cellule (ou un groupe de cellules) entièrement. Avec `GTK_EXPAND` le composant est situé au centre de la cellule (ou d'un groupe de cellules) et `GTK_SHRINK` permet de l'adapter (le rétrécir) aux dimensions de la cellule (ou du groupe de cellules). *xoptions* permet de ne jouer que sur l'axe x des cellules et *yoptions* que sur l'axe y.

xpadding ajoute de l'espace à gauche et à droite d'un composant suivant l'axe x, alors que *ypadding* le fait suivant l'axe y.

Voici un exemple de code:

```
#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;
}
```

Comme `gtk_table_attach` possède de nombreux paramètres, une nouvelle fonction plus courte a été créée; `gtk_table_attach_defaults`. Cette fonction fait la même chose avec moins de paramètres.

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

Les paramètres s'utilisent de la même façon. *xoptions* et *yoptions* ont la valeur `GTK_FILL|GTK_EXPAND`. *xpadding* et *ypadding* ont la valeur 0.

La fonction suivante est utilisée pour changer le nombre de lignes et de colonnes d'une table existante :

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

Avec les fonctions suivantes, vous pouvez changer l'espacement dans une ligne ou dans une colonne:

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

Les fonctions suivantes permettent de changer l'espacement pour toutes les lignes ou les colonnes :

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

La fonction suivante change la valeur d'homogénéité d'une table existante :

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

En résumé

Dans cet article, nous avons vu différentes manières d'organiser des composants, puis les fonctions de propriétés des boîtes et des tables. J'ai toujours plaisir à recevoir des questions, des commentaires ou des idées de lecteurs. Envoyez-moi simplement un courrier électronique....

Site Web maintenu par l'équipe d'édition LinuxFocus © Özcan Güngör "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Translation information: tr --> -- : Özcan Güngör <ozcangungor(at)netscape.net> en --> tr: Özcan Güngör <ozcangungor(at)netscape.net> en --> fr: Frédéric Nourry <frederic.nourry(at)laposte.net>
---	--