

Programación en C moderno

Álvaro Neira Ayuso <alvaro@soleta.eu>

GTK: Biblioteca para crear la interfaz gráfica de una aplicación.

- - Introducción: Historia y software necesario
- - Ejemplo 1: Creación de widgets, ventanas, botones, etiquetas y contenedores
- - Ejemplo 2: Creación de widget con uso de menús. Uso de menus. Uso de Toolbars

Introducción: Historia y software necesario

- GTK "GIMP Tool Kit" es una biblioteca del equipo GTK+
- Para crear la interfaz gráfica de usuario
- Maneja widgets como ventanas, botones, menús, etiquetas, deslizadores, pestañas...

GTK+

- Biblioteca desarrolladas por la GNOME Foundation el 14 de Abril de 1998
- Programada en C
- Con licencia GNU LGPL

GTK+

- GTK+ o The GIMP Toolkit es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario (GUI)
- Principalmente para los entornos gráficos GNOME, XFCE y ROX.
- Aunque también se puede usar en el escritorio de Windows, Mac OS y otros.

GTK+

- Inicialmente fueron creadas para desarrollar el programa de edición de imagen GIMP
- Junto a Qt es una de las bibliotecas más populares

Bibliotecas de GTK+

- GLib. Proporciona manejo de estructura de datos para C, portabilidad, interfaces para funcionalidades de tiempo de ejecución como ciclos, hilos, carga dinámica o un sistema de objetos.
- GTK. Biblioteca la cual realmente contiene los objetos y funciones para crear la interfaz de usuario.

Bibliotecas de GTK+

- GDK. Biblioteca que actúa como intermediario entre gráficos de bajo nivel y gráficos de alto nivel.
- ATK. Biblioteca para crear interfaces con características de una gran accesibilidad muy importante para personas discapacitadas o minusválidos.
- Pango. Biblioteca para el diseño y renderizado de texto.
- Cairo. Biblioteca de renderizado avanzado de controles de aplicación.

Entornos que utilizan GTK+

- GNOME
- Xfce está basado en GTK+
- LXDE "Lightweight X11 Desktop Environment"

Aplicaciones que utilizan GTK+

- Evolution - Cliente de correo electrónico.
- Firefox - Navegador web.
- GIMP - Editor de gráficos.
- Inkscape - Editor de gráficos vectoriales SVG.
- Nero Linux - Un programa para la edición de discos.
- Pidgin - Cliente de mensajería instantánea.
- VMware Player - Máquina virtual.
- Wireshark - Capturador y analizador de paquetes de redes computacionales.

Biblioteca disponible en

- C
- C++ (gtkmm)
- C# (Gtk Sharp)
- Java (java-gnome)
- Python (PyGTK)
- Javascript

Bibliotecas necesarias en Debian/Ubuntu

- GCC
- Pkg-config, necesario para mostrar información de las bibliotecas de desarrollo
- Paquetes de desarrollo de GLib
- Paquetes de desarrollo de GTK+

Bibliotecas a instalar

- gcc
- libc6-dev
- pkg-config
- libglib2.0-dev
- libgtk2.0-dev

Compilar programas que tienen GTK

- `CFLAGS=`pkg-config --libs --cflags gtk+-2.0``
- `export CFLAGS`
- `gcc $CFLAGS -o programa programa.c`

Métodos principales GTK

```
gtk_init(&argc, &argv);
```

- Con esta función, se inicializa el subsistema de GTK+.

Métodos principales GTK

```
gtk_widget_show(ventana);
```

- Esta función permite que el widget sea visible.

Métodos principales GTK

`gtk_main();`

- Esta función inicializa el loop principal de GTK+, donde se dibujarían todos los controles.

Widgets Básicos

- En informática, un widget o artilugio¹ es una pequeña aplicación o programa
- Usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine
- Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual

Ventanas

- La ventana es el mas necesario de los Widgets en GTK+. La función `gtk_window_new()` devuelve un widget de tipo ventana. El argumento de la función es:
 - `GTK_WINDOW_TOPLEVEL` : generalmente usado, es para crear una ventana nueva que informe al Window Manager
 - `GTK_WINDOW_POPUP` : en caso de querer usar un pop-up, no manejado por el Window Manager

Ventanas

```
GtkWidget *ventana;  
ventana = gtk_window_new(GTK_WINDOW_TOPLEVEL);
```

Etiquetas

- Una etiqueta es un control de texto. Es uno de los controles mas simples. No emiten señales si no están asociados a una ventana.
- La función `gtk_label_new()` crea un widget de tipo etiqueta.
- `GtkWidget * gtk_label_new(const char *string);`

Etiquetas

```
GtkWidget *etiqueta;  
etiqueta = gtk_label_new("Es una etiqueta");
```

Botones

- Este widget es para comunicación de usuario
- En algunos casos puede tener una imagen asociada.
- La función `gtk_button_new(void)` crea un widget de tipo botón
- Genera un evento de tipo "Clicked"

Botones

```
GtkWidget *boton;  
boton = gtk_button_new();
```


Botones con etiquetas

- La función `gtk_button_new_with_label()` crea un widget de tipo botón, con una etiqueta contenida en el botón.
- `gtk_button_new_with_label(char *etiqueta)`

Botones con etiquetas

```
GtkWidget *boton_con_etiqueta;
```

```
boton_con_etiqueta = gtk_button_new_with_label("Estoy  
dentro de un boton");
```

Contenedores

- Un widget puede contener a otro. El mas común es la ventana, que puede encapsular a otros widgets.
- Para contener un widget en otro se utiliza la funcion `gtk_container_add`.

```
gtk_container_add((GtkContainer *)  
widget_contenedor, widget_a_contener);
```

Contenedor

```
GtkWidget *boton;
```

```
GtkWidget *etiqueta;
```

```
GtkWidget *ventana;
```

```
boton = gtk_button_new();
```

```
etiqueta = gtk_label_new("Etiqueta 1");
```

```
ventana = gtk_window_new(GTK_WINDOW_TOPLEVEL);
```

```
gtk_container_add ((GtkContainer *) ventana, boton);
```

```
gtk_container_add ((GtkContainer *) boton, etiqueta);
```

Ejemplo

Métodos de un Widget

- GtkWidget hereda las propiedades y métodos de la clase GObject. Además, posee sus propios métodos y propiedades.
- Cualquier control que sea hijo de la clase GtkWidget hereda sus propiedades y métodos

Métodos de un Widget

- `gtk_widget_show((GtkWidget *) widget);`
- `g_object_set_property((GObject *) ventana, "title", "Titulo de la ventana");`

Widgets de Empaquetamiento

- Algunos de estos widgets "invisibles" son usados para empaquetamiento. Es decir, pueden contener mas widgets dentro de el.
- El widget mas conocido para empaquetamiento es el GtkBox (o caja de empaquetamiento). Lo hay en dos versiones, horizontal (GtkHBox) y vertical (GtkVBox).

Widgets de Empaquetamiento

```
hbox = gtk_hbox_new(homogeneidad, distancia);
```

- La función `gtk_hbox_new()` crea un widget de tipo HBox. Los parámetros que recibe esta función son:
 - `homogeneidad` : (gboolean) si es verdadero, todos los widgets que contiene ocuparan el mismo espacio
 - `distancia` : (gint) numero de píxeles de espacio entre widgets.

Widgets de Empaquetamiento

```
gtk_box_pack_start((GtkBox *) hbox, widget, expandir,  
llenar, margen);
```

- hbox : el widget empaquetador de tipo GtkBox
- widget : el widget a insertar al principio
- expandir : (gboolean) si es verdadero, se le da espacio extra al siguiente hijo. Este espacio se reparte equitativamente entre los widgets ya insertados
- llenar : (gboolean) si es verdadero, el widget hijo se expandira.
- margen : (gint) espacios extras en pixeles entre el widget insertado y sus vecinos.

Widgets de Empaquetamiento

```
gtk_box_pack_end((GtkBox *) hbox, widget, expandir,  
llenar, margen);
```

- hbox : el widget empaquetador de tipo GtkWidget
- widget : el widget a insertar al final
- expandir : (gboolean) si es verdadero, se le da espacio extra al siguiente hijo. Este espacio se reparte equitativamente entre los widgets ya insertados
- llenar : (gboolean) si es verdadero, el widget hijo se expandirá
- margen : (gint) espacios extras en píxeles entre el widget insertado y sus vecinos.

Ejemplo

•Toggle (Botón de dos estados)

- Los botones de tipo toggle o de dos estados, son botones los cuales pueden estar en dos estados: Pulsado o no pulsado.

```
toggle = gtk_toggle_button_new();
```

Ejemplo

Señales

- Es posible definir funciones arbitrarias al momento de recibir algunos eventos.
- La programación gráfica requiere de interacción con el usuario
- GTK+ recibe y emite señales al momento de comunicarse con la capa gráfica (X11).
- En el caso del botón, al momento de hacer click en el se genera la señal "clicked".

Señales: Definición de funciones

```
"clicked" void user_function (GtkButton *  
button, gpointer user_data);
```


Señales: Conectar las funciones con los widgets

```
g_signal_connect((GObject *) widget, const gchar signal,  
(GCallback *) function, gpointer data);
```

- widget : es el widget que origina la señal
- const gchar signal : es el nombre de la señal a recibir
- function : la función que se va a conectar a la señal
- data : puntero arbitrario de envío de datos de la función

Ejemplo

GtkEntry

- Como el nombre lo indica, sirve para ingresar texto.

```
GtkWidget *entrada;  
entrada = gtk_entry_new();
```

Ejemplo

ComboBox

- En un combo box, el usuario selecciona una de varias opciones.

```
GtkWidget *combo;
```

```
combo = gtk_combo_new();
```

```
printf("valor del activo %d\n",  
gtk_combo_box_get_active((GtkComboBox  
)combo));
```

Ejemplo

Spinners

- Estos controles seleccionan un valor, el que se modifica por dos flechas pequeñas (hacia arriba y abajo).
- Para usar un spinner, es necesario hablar de un control llamado GtkAdjustment.

Spinners

- Este control permite representar en un control valores de "borde" o limite. Por ejemplo, para los scrollbars.
- Para crear un GtkAdjustment, se usa la función `gtk_adjustment_new()` con los siguientes parámetros:
 - Valor inicial
 - Valor mínimo
 - Valor máximo
 - Incremento
 - Incremento por página (saltos)
 - Tamaño de página

Spinners

- `gtk_spin_button_new (GtkAdjustment *adjustment, gdouble climb_rate, guint digits);`
 - `adjustment` = Objeto GTK con los ajustes
 - `climb_rate` = Cuanto cambia el spin por cada pulsación de las flechas
 - `digits` = Número de decimales a mostrar

Spinners

```
GtkWidget *spinner;  
spinner = gtk_spin_button_new(adjustment, 1, 1);  
  
printf("value %d\n",  
gtk_spin_button_get_value((GtkSpinButton  
*)spinner))
```

Ejemplo

Sliders

- Los sliders son tanto verticales como horizontales, y sirven para seleccionar también en un rango, no numérico

```
GtkWidget *sliderh;
```

```
sliderh = gtk_hscale_new(ajuste);
```

```
printf("Valores %d\n", gtk_scale_get_digits ((GtkScale  
*)sliderh));
```

Ejemplo

Barra de Progreso

- Las barras de progreso se usan para mostrar el estado de una operación.

```
GtkWidget *progreso;
```

```
progreso = gtk_progress_bar_new();
```

Ejemplo

Menú

- Todos los programas a parte de las acciones tienen menús
- Gtk permite el uso de menú con una serie de funciones que veremos a continuación

Menu

- `menubar = gtk_menu_bar_new();`
- `filemenu = gtk_menu_new();`
- `file = gtk_menu_item_new_with_label("File");`
- `gtk_menu_shell_append(GTK_MENU_SHELL
(menubar), file);`

Ejemplo

Ejercicio

Bibliografía

<https://developer.gnome.org/gtk3/stable/>