

Kapitola 1

Úvod

GTK je multiplatformní knihovna pro tvorbu grafického uživatelského prostředí napsaná v jazyce C. Jeho použití je ale možné i v jiných jazycích jako například C++, Python, PHP, ... V současné době je aktuální 3. verze knihovny. Jedná se o jednu z nejpopulárnějších knihoven pro tvorbu GUI aplikací v operačních systémech Linux. Programy vytvořené v této knihovně jsou vyhlášeny svou rychlostí (na rozdíl od Qt, ...).

Samotné GTK závisí na knihovnách:

- Glib
- Pango
- ATK
- GDK
- GdkPixbuf
- Cairo

Knihovna **Glib** poskytuje všeobecně užitečné nástroje a datové typy používané v GTK. **Pango** knihovna umožňuje internacionalizaci programovaných aplikací. **ATK** je knihovna, která pracuje s přístupy. Poskytuje nástroje, které umožňují fyzicky měnit způsob jak lidé pracují s počítači. Knihovna **GDK** pracuje s nízkoúrovňovým systémem vykreslování, a tvorby grafických oken zobrazovacím systémem počítače. Knihovna **GdkPixbuf** zajišťuje vkládání a práci s obrázky. Nakonec knihovna **Cairo** poskytuje nástroje s 2D grafikou.

Hlavní nevýhodou GTK je jeho omezená možnost objektového fungování, protože je napsán v jazyce C, které není objektově orientované. Naopak je díky tomu velice rychlé.

Kapitola 2

Základní práce s okny v GTK

Pro práci s GTK je nutné na začátku programu nejprve vložit hlavní soubor *gtk.h*:

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

Před zahájením práce s grafickými objekty je třeba inicializovat gtk knihovnu pomocí funkce `gtk_init`, která přijímá jako parametry odkaz na argumenty příkazové řádky z funkce `main`:

```
int main(int argv, char**argc)
...
gtk_init(&argv, &argc);
...
```

Základní třídou, která slouží pro práci s grafickými widgety je třída *GtkWidget*. Ostatní grafické widgety jsou odvozeny z této třídy. Nové okno se vytvoří pomocí funkce:

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

Funkce vnitřně volá funkci *malloc()*, která zajistí vytvoření nového objektu. Funkce přijímá parametr výčtového typu **GtkWindowType**, který definuje typ nového okna.

Pro zobrazení nově vytvořeného okna slouží funkce `gtk_widget_show`, která přijímá odkaz na objekt okna:

```
gtk_window_show(window);
```

Nakonec je nutné zavolat funkci *gtk_main*, která způsobí, že program vstoupí do pasivní smyčky a čeká na uživatelskou interakci. Funkce nepřijímá žádné parametry. Bez této funkce by se program okamžitě ukončil.

Všechny funkce pracující s oknem (nastavují jeho parametry) se nacházejí v hlavičkovém souboru *gtkwindow.h*.

Je důležité, že zavření okna ve výchozím nastavení neukončí daný proces. To je nutné nastavit pomocí obslužných funkcí: signál-slot.

2.1 Nastavení okna

funkce `GTK_WINDOW(window)` nastaví fullscreen okna.

funkce `gtk_window_set_title(GtkWindow * window, gchar* str)` nastaví titulek okna funkce `gtk_window_set_default_size(GtkWindow *window, gint width, gint height)` nastaví rozměry grafického okna

!!! nikdy nevolat objekty grafického rozhraní GTK z jiného vlákna než ve kterém byly vytvořeny, jinak dojde k vyvolání běhové chyby při přístupu k xlib knihovně!!!

Kapitola 3

Třída GtkWidget

Třída GtkWidget je bázovu třídou pro všechny grafické widget, se kterými knihovnaa GTK pracuje.

```
void gtk_widget_set_size_request(GtkWidget* widget, gint width, gint height)
```

Funkce nastaví minimální rozměry daného widgetu (tlačítka, okna, popisku, ...).

Kapitola 4

Kontejner

GTK uživatelské rozhraní je postaveno tak, že ostatní widgety jsou vloženy do jiných widgetů. Kontejnerové widgety jsou vnitřní uzly výsledného stromu widgetů, které obsahují ostatní widgety. Kontejner, který obsahuje, tedy obaluje, další prvky se nazývá rodič a prvek který je uvnitř nějakého prvku, tedy je obalen, se nazývá potomek.

Existují dva hlavní kontejnerové widgety, které jsou oba odvozené od základní třídy *GtkContainer*.

První typ kontejneru může obsahovat pouze jednoho potomka (widget) a je odvozen z třídy *GtkBin*. Tyto kontejnery jsou dekorace, které přidávají novou funkčnosti do widgetu potomka. Typickým příkladem widgetu, který může obsahovat pouze jednoho potomka je okno (*GtkWindow*). Aby mohlo okno obsahovat více než jeden widget, musí obsahovat widget kontejneru druhého typu.

Druhý typ kontejneru mohou obsahovat více než jeden widget. Jejich účelem je spravovat rozvržení jednotlivých prvků v grafickém okně. Typickým příkladem je *GtkFixed*.

4.1 GtkStack

GtkStack je kontejner, který umožňuje uchovávat více než jeden widget, ale viditelný je v danou chvíli pouze jeden. Mezi zobrazovanými widgety je možné přepínat. Díky tomu lze definovat stránky jednoho okna.

4.2 GtkEventBox

GtkEventBox je typ kontejneru, který umožňuje uchovávat jeden dceřinný widget. Využívá se v situacích, kdy je třeba přidat události grafickému widgetu, který dané události neposkytuje.

```
GtkWidget* gtk_event_box_new(void) - vytvoří nový objekt třídy  
GtkEventBox
```

`void gtk_widget_set_events(GtkWidget* widget, gint events)` - přiřadí dané události k danému widgetu

Kapitola 5

Správci rozvržení

Správci rozvržení jsou druhem kontejnerů, které umožňují uložit více než jeden widget a manipulovat s jejich polohou v okně programu.

5.1 GtkGrid

GtkGrid je kontejnerový správce rozložení, který umožňuje uchovávat více než jeden widget a strukturuje uložené widgety v okně do mřížky. Objekt třídy GtkGrid se vytvoří pomocí funkce:

```
GtkWidget* gtk_grid_new(void)
```

Do mřížky v okně se widgety vloží pomocí funkce:

```
void gtk_grid_attach(GtkGrid* grid, GtkWidget* child, gint left, gint right, gint width, gint height)
```

Kapitola 6

System slot-callback

Gtk je událostmi řízený způsob programování programu. Principem je, že daná akce grafického rozhraní (widgetu) emituje, vyšle signál, že nastala nějaká událost. V programu je k této události daného widgetu připojena obslužná (callback) funkce, která má za úkol vykonat nějakou obsluhu. Funkce pro práci se signály se nacházejí v souboru `gsignal.h`.

Pro připojení callback funkce k určitému signálu daného widgetu slouží funkce:

```
gulong g_signal_connect(gpointer *object, const char* name, GCallback  
func, gpointer data)
```

Prvním argumentem je ukazatel na widget, který vyslal daný signál, druhý parametr je název signálu, který má být zachycen, třetí parametr je název funkce, která se má zavolat v případě zachycení daného signálu, posledním parametrem je ukazatel na data, která se předávají do volané funkce při zachycení signálu.

Formát obslužné funkce má obecný formát:

```
void fce_name(GtkWidget* widget, gpointer data)
```

kde prvním argumentem je ukazatel na widget, který vyslal daný signál a posledním argumentem je ukazatel na data, předávané do funkce při jejím volání. Jedná se ale pouze o obecný formát, který se může u některých widgetů lišit v závislosti na parametrech, které vyžadují předat emitované signály.