



Estudo de Caso Aprendizado **CRIAÇÃO DE EDITOR DE TEXTO E TUTORIAL VOLTADO** **PARA O APRENDIZADO DE C COM GTK**

Katysco de FARIAS SANTOS (1); Andréa TAVARES (2)

(1) CEFET-PE, Av. Prof. Luís Freire, 500, telefone: (81) 2125-1720, e-mail:

katysco@gmail.com

(2) e-mail: andreatmelo@gmail.com

RESUMO

Profissionais especializados em desenvolver aplicações gráficas utilizando a biblioteca GTK (Gimp ToolKit) sobre plataformas abertas não é comum. Esse trabalho trata-se de um estudo de caso de aprendizagem para criação de uma aplicação gráfica, um editor de texto simples, cujo objetivo é explicar os mecanismos e funções que são acessadas da biblioteca GTK para implementar sua Interface Gráfica. As funcionalidades do Editor foram codificadas em linguagem C que também foi utilizada para acessar as funções da biblioteca GTK. Os processos responsáveis pelo tratamento de eventos suportados pela biblioteca GTK são explicados detalhadamente, enquanto se desenvolve o Editor. Este trabalho é de caráter experimental e sua metodologia foi baseada numa compilação bibliográfica sobre o uso da biblioteca GTK para o desenvolvimento de aplicações. Abrange-se, a instalação do compilador GCC(GNU Compiler Collection) e de diretivas de compilação através da utilização de arquivos make; o uso da biblioteca GTK para linguagem C; e a criação de programas exemplos que incrementalmente vão compor o Editor de Texto Simples. Foi desenvolvido em plataforma aberta sem auxílio de ferramentas geradoras de código de forma validar o uso de Softwares Livres e do ensino de tecnologias livres em curso tecnológicos, além de criar uma documentação de referência para aqueles que almejem aprender a desenvolver aplicações gráficas baseadas em GTK.

Palavras-chave: aplicações graficas, C, GTK, software livre

1. INTRODUÇÃO

O GTK (Gimp Toolkit) é um toolkit de criação de interfaces gráficas utilizado para a criação de diversas aplicações como GNOME, XFCE, GIMP e Anjuta. É escrito em linguagem C sob o paradigma de orientação a objetos e é multiplataforma. Considerando sistemas operacionais GNU/Linux baseados em Debian, e visando o aprendizado de linguagem C com GTK, como iniciativa de aprendizagem de linguagem e alternativas livres, foram criados tutorial e editor de texto como exemplo, os quais serão apresentados neste trabalho.

2. CONTEÚDO

Este trabalho apresentará os seguintes tópicos:

- Conceito, funcionamento, arquitetura e algumas funções do GTK;
- Ferramentas mínimas necessárias para desenvolvimento de aplicações em linguagem C com GTK: GCC e bibliotecas GTK;
- Criação e uso de arquivo make para a compilação;
- Exemplos da utilização de funções GTK;
- Criação de tutorial para servir como guia de referência.
- Criação de editor de texto utilizando conceitos, ferramentas, funções e bibliotecas vistas no tutorial.

3. O GTK

3.1. Conceito

O GTK+, muito comumente chamado de GTK, são bibliotecas para criação de interface gráfica, sob licença LGPL. É multi-plataforma, e diversas linguagens como C, C++, Perl, Ruby, PHP, Lisp, Python, entre outras, podem utilizá-lo. O GTK+ foi inicialmente criado para o GIMP (GNU Image Manipulation Program) e atualmente é muito utilizado no Desktop GNOME, XFCE e diversas aplicações como o Anjuta.

3.2. Arquitetura

O GTK utiliza bibliotecas como:

- Glib - fornece funcionalidades que não são relacionadas a interface gráfica e sim aos tipos de dados, manipulações com strings e arquivos, entre outros;
- Pango - para texto, fornece por exemplo, o GtkTextView, o GtkEntry e o GtkLabel;
- ATK - para acessibilidade;
- GDK - fornece ao GTK+ suporte aos diferentes servidores gráficos.

3.3. Funcionamento

Gtk é um toolkit orientado a evento, de forma que, se nenhum evento ocorrer, ele ficará em loop infinito através do `gtk_main()` fazendo com que a aplicação fique sendo executada permanentemente. Os sinais são uma maneira de customizar o comportamento dos objetos. Cada sinal tem um nome único, e um id inteiro positivo(0, por exemplo é usado para sinalizar um erro). Existem sinais que são comuns a todos os widgets, como o 'destroy'.

A função `gtk_main()` fica em loop permanente(ver Figura 1), esperando que algum evento ocorra, quando o evento ocorre(visulizado como solicitação do callback), a solicitação é executada e então a função `gtk_main()` volta ao loop, e a aplicação fica neste ciclo ate que haja a solicitação para parar a função `gtk_main()` então a aplicação é finalizada.

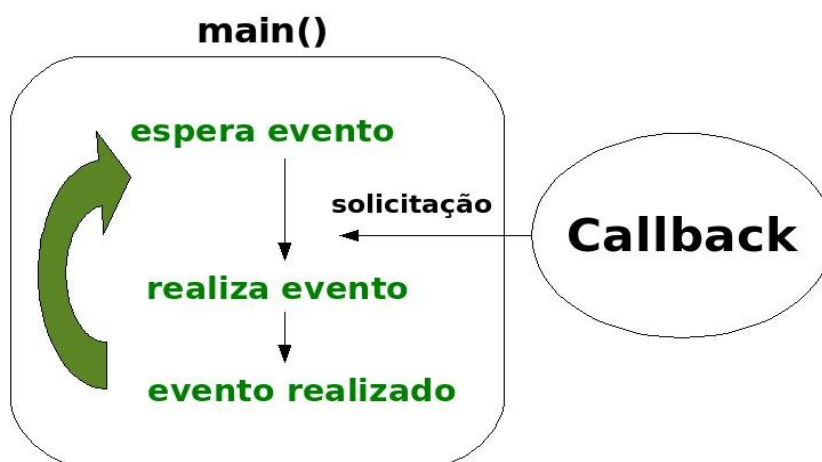


Figura 1 – Esquema de Funcionameto

3.4. Compilação

O compilador utilizado(por linha de comando), foi o GCC, contido, em sistemas GNU/Linux baseados em Debian, no pacote build-essential e já vem instalado por padrão junto com o sistema operacional básico para Desktop. Para que o GTK funcione, além dos pacotes básicos, deve ter a biblioteca "dev" do gtk(libgtk2.0-dev, por exemplo). Supondo que já se tenha o gcc e o Gtk instalados, eis o comando, a ser executado pelo terminal:

```
gcc nomearquivo.c -o nomeoutput -g -Wall `pkg-config --cflags --libs gtk+-2.0`
```

no qual:

- gcc - é o próprio compilador;
- nomearquivo.c é o nome do arquivo a ser compilado;
- nomeaoutput é o nome do executável a ser gerado após a compilação sem erros;
- -g -Wall são para produzir debug e fazer com que alguns 'warnings' não sejam mostrados, respectivamente;
- `pkg-config --cflags --libs gtk+-2.0` - o pkg-config é um programa que dirá ao gcc onde estão as bibliotecas do gtk. Deve-se ter atenção que são utilizadas crases, e não aspas simples.

3.5. Arquivo de Make

O arquivo de *make(makefile)* é um arquivo para configuração de compilação utilizado pelo programa *Make*, com o objetivo de automatizar e otimizar a compilação. O *make* irá procurar pelo arquivo *Makefile*(ou o *makefile*) e então o executará. Utilizamos o seguinte arquivo de *make*:

```
CC=gcc
LIBS=`pkg-config --libs gtk+-2.0 gtksourceview-1.0`
CFLAGS=`pkg-config --cflags gtk+-2.0 gtksourceview-1.0`
OBJS=interface.o funcoes.o compilacao.o impr.o
PROG=editor

all: $(OBJS)
    $(CC) $(LIBS) $(OBJS) -o $(PROG)

interface.o: interface.c globals.h
    $(CC) $(CFLAGS) interface.c -c

funcoes.o: funcoes.c funcoes.h globals.h
    $(CC) $(CFLAGS) funcoes.c -c

compilacao.o: compilacao.c compilacao.h globals.h
    $(CC) $(CFLAGS) compilacao.c -c

impr.o: impr.c impr.h globals.h
    $(CC) $(CFLAGS) impr.c -c

clean:
    rm -rf $(OBJS) $(PROG)
```

Através do mesmo pudemos automatizar a compilação dos arquivos através de um único comando simples e curto: *make*. *CC*, *LIBS*, *CFLAGS*, *OBJS* indicam, respectivamente o compilador, as bibliotecas a serem utilizadas, as *flags*, e quais serão os arquivos de *output*.

4. TUTORIAL E EDITOR DE TEXTO

Escreveu-se tutorial com fundamentos, exemplos e figuras em linguagem C com GTK, para auxílio no aprendizagem. O tutorial está estruturado de forma a se ter uma parte introdutória, no qual se comenta sobre fundamentos do GTK e compilação em sistemas GNU/Linux baseados em Debian, uma outra parte instruindo como programar em C com GTK, com códigos de exemplos e figuras da execução dos mesmos(ver Figura 2 e Figura 3).



Figura 2 - Exemplo do Tutorial

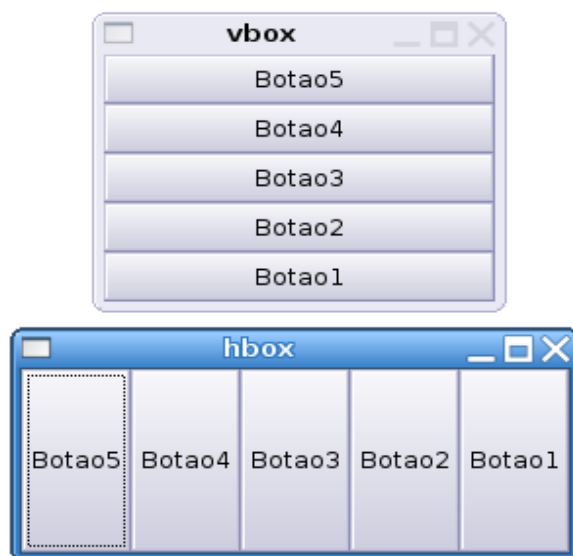


Figura 3 – Exemplo do Tutorial

Em seguida, os exemplos do tutorial foram colocados em prática através de criação de editor de texto(ver Figura 4), que possui campo onde aparece texto explicando o funcionamento do próprio editor, através das funções GTK.

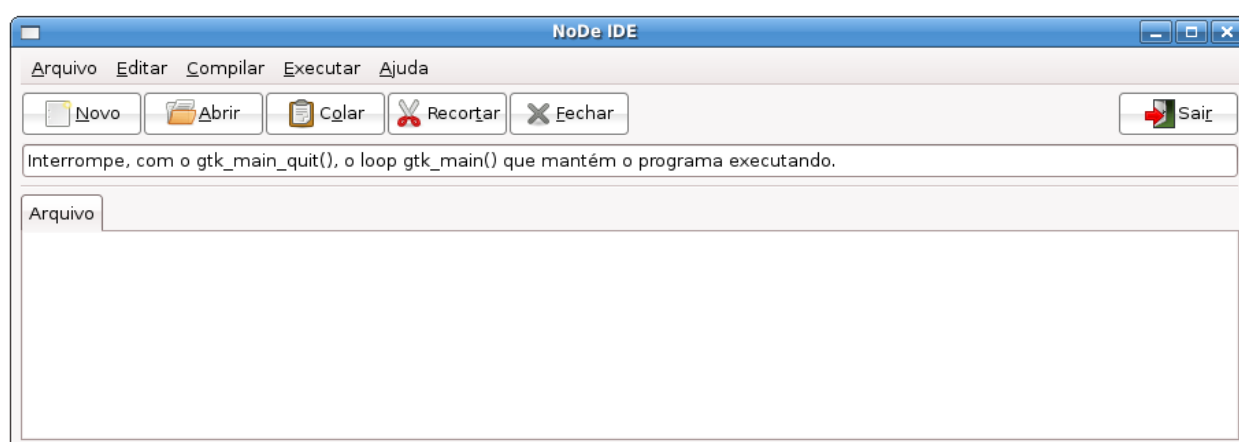


Figura 4 – Editor de texto

REFERÊNCIAS

URTUBIA, H. **How to write a Makefile**. Disponível em:
<http://www.hsrl.rutgers.edu/ug/make_help.html> Acesso em: 11 agosto 2008.

_____. **GTK**: documentação. Disponível em: <<http://www.gtk.org/>> Acesso em: 11 agosto 2008.

_____. **GTK+ Reference Manual**. Disponível em:
<<http://library.gnome.org/devel/gtk/2.12/>> Acesso em: 11 agosto 2008.

_____. **GTK+ Reference Manual**. Disponível em:
<<http://library.gnome.org/devel/gtk-tutorial/stable/c355.html>> Acesso em: 11 agosto 2008.

_____. **GCC, The GNU Compiler Collection**. Disponível em: <<http://gcc.gnu.org/>>
Acesso em: 11 agosto 2008.

_____. **GCC, The GNU Compiler Collection**. Disponível em: <<http://gcc.gnu.org/>>
Acesso em: 11 agosto 2008.