

# Guia de instalação da Lib HwGui para Harbour no Ubuntu 16 LTS

O objetivo desse guia é incentivar os usuários a usarem o Linux como plataforma de desenvolvimento e de produção. Ele ensina como instalar e compilar a HwGui, lib gráfica para Windows e Linux

1 ) Baixe a versão da hwgui no site <https://sourceforge.net/projects/hwgui/> e descompacte o arquivo zip.

2 ) A HwGui para Linux necessita da Lib Gtk para funcionar. Entre no terminal e torne-se root, depois instale a lib :

```
sudo -s (Virei root)
apt-get install libgtk2.0-dev
```

Continue como root

3 ) Diferente do Harbour, a HwGui não vai copiar os compilados para outro local, então você deve copiar tudo posteriormente. Vá para o diretório onde o arquivo zip foi descompactado. Ou se preferir (é até melhor) mova essa diretório para o seu diretório de fontes:

```
cd /<seu diretório de fontes>
mv /home/bolsista/Downloads/hwgui .
```

4 ) Entre no diretório cd hwgui/source/gtk/

5 ) Nesse diretório tem um arquivo chamado build.sh , você precisa atribuir direitos de execução para esse arquivo :

```
chmod 755 build.sh
```

6 ) Se você instalou o Harbour em /usr/local/bin (O local padrão), então pode executar esse script, caso contrário você precisa editar esse script. Vá para o passo 6.1

6.1 ) Aqui vc faz somente se o seu harbour NÃO ESTIVER em /usr/local/bin

a ) Abra o arquivo script.sh

b ) Descomente a linha que tem #export HB\_ROOT=../.. e substitua o valor dessa variável pelo local onde o Harbour está instalado.

7 ) O script é executado sem enviar saídas para a tela. Ele cria dois logs: a1.log e a2.log. O primeiro exhibe as mensagens normais, e o segundo é o stream de erro. No meu caso eu abri a2.log e tinha alguns warnings, mas nenhum erro.

8 ) Para testar eu fiz assim :

a ) Copiei as libs para o diretório onde tem as libs do Harbour. No meu caso :

/usr/local/lib/harbour

b ) Copiei os includes para o local onde tem os includes do Harbour. No meu caso :

/usr/local/include/harbour

As libs e os includes da hwgui foram gerados em hwgui/lib e os includes em hwgui/include. Mais tarde fui ver que existem variáveis de ambiente que vc pode criar para informar o local das libs e dos includes da hwgui.

9 ) Para testar eu fui para a pasta hwgui/samples/gtk\_samples e segui os seguintes passos :

a ) Transformei o build.sh em executável : chmod 755 build.sh

b ) Alterei o build.sh (na verdade coloquei os paths da lib). Ficou assim o meu build.sh (destaquei em negrito os paths, veja que como eu havia copiado para os paths do Harbour eu acabei repetindo):

```
#!/bin/bash
```

```
export SYSTEM_LIBS="-lm"
```

```
export HARBOUR_LIBS="-lhbdebug -lhbvm -lhbtrl -lgtcgi -lhbdebug -lhblang -lhbrrd  
-lhbmacro -lhbpp -lrdntx -lrdcdx -lrdfpt -lhbsix -lhbcommon -lhbcpage"
```

```
export HWGUI_LIBS="-lhwgui -lprocmisc -lhbxml -lhgwdebug"
```

```
export HWGUI_INC=/usr/local/include/harbour
```

```
export HWGUI_LIB=/usr/local/lib/harbour
```

```
export HRB_INC=/usr/local/include/harbour
```

```
export HRB_LIB=/usr/local/lib/harbour
```

```
harbour $1 -n -i$HRB_INC -i$HWGUI_INC -w2 -d__LINUX__ $2
```

```
gcc $1.c -o$1 -I $HRB_INC -L $HRB_LIB -L $HWGUI_LIB -WI,--start-group $HWGUI_LIBS  
$HARBOUR_LIBS -WI,--end-group `pkg-config --cflags gtk+-2.0` `pkg-config gtk+-2.0 --libs`  
$SYSTEM_LIBS
```

```
~
```

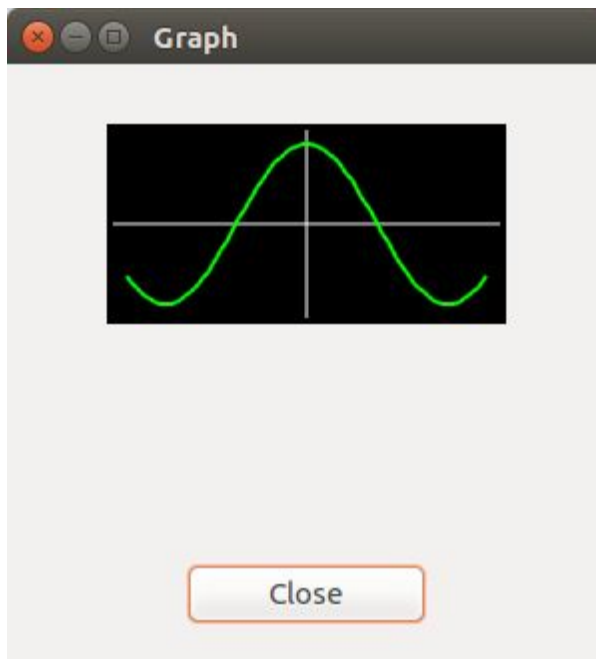
c ) Para compilar eu fiz

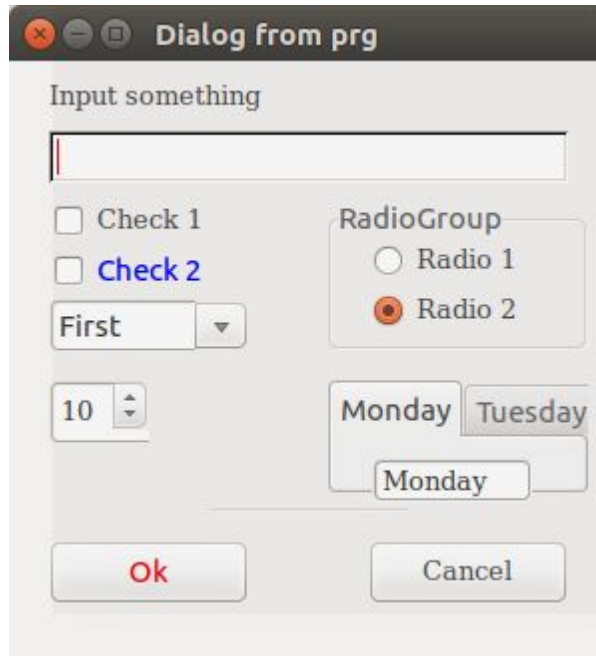
`./build.sh a` (onde `a` é o arquivo que tem lá chamado `a.prg`)

Não informe a extensão quando for usar o `build.sh`, por exemplo `./build.sh a.prg` vai dar erro.

Executei os outros exemplos com sucesso.

Algumas telas :





Achei essas abas Monday e Tuesday muito feias. Será que o efeito era esse ? Será que no windows esses programas funcionam ?

Peguei o exemplo que está no site oficial e salvei como hello.prg :

```
#include "hwgui.ch"

Function Main
Local oMainWnd, oFont
Local aCombo := {"First","Second" }

PREPARE FONT oFont NAME "MS Sans Serif" WIDTH 0 HEIGHT -13

INIT WINDOW oMainWnd TITLE "Example" ;
FONT oFont ;
ON EXIT {{|hwg_MsgYesNo("Really want to quit ?")}}

@ 20,10 EDITBOX "Hello, World!" SIZE 200,30

@ 270,10 COMBOBOX aCombo SIZE 100, 150 TOOLTIP "Combobox"

@ 120,60 BUTTON "Close" SIZE 150,30 ;
ON CLICK {{|oMainWnd:Close()}}

MENU OF oMainWnd
MENUITEM "About" ACTION hwg_MsgInfo("First HwGUI Application")
ENDMENU

ACTIVATE WINDOW oMainWnd

Return
```

Compilei o programa ./build.sh hello

Aqui no meu Ubuntu 16 LTS não ficou com uma aparência boa. Na verdade nem vi direito o programa, nem apareceu uma janela e ficou apenas um menu com uma opção about.

Aparentemente não fiz nada errado.

## Conclusão

### Aspectos positivos

- 1 ) Biblioteca multiplataforma (Windows e Linux) gerando executáveis
- 2 ) Open source
- 3 ) Compilação relativamente simples.

### Aspectos negativos

- 1 ) Os executáveis gerados não ficaram com uma aparência muito boa.
- 2 ) Alguns exemplos apresentaram erros (por ex: example.prg)

Acho que vale a pena estudar a HwGui, mas você não deve confiar que os executáveis irão se comportar com a mesma aparência nas duas plataformas, tem que testar os detalhes. O ideal seria ir testando aos poucos (começando por estudar o a.prg e a documentação que está no site). Tenha duas máquinas (pode até ser virtual) uma com windows e outra com o linux caso vc queira gerar para duas plataformas.

Espero ter ajudado !

Vlademiro

