Hello, Forth!

A Proof of Concept (POC)
Graphical User Interface (GUI)
Implemented in the
Gnome Tool Kit (GTK)
and the
Forth Language (Gforth)

John E. Harbold
January 27, 2018

Introduction

- A GUI gives ease of use of an application by visual means.
- Forth allows a user to implement an application in a timely manner, if and only if the user is a timely programmer.
- Gforth has a large Forth code source to support many architectures, operating systems and standalone.
- Gforth also allows using preexisting libraries to speed development.
- Because GUI’s require the use of “callbacks”, Gforth satisfies this requirement.
- Size of application is generally smaller than other languages.

GUI

- The visual part of a GUI can be implemented using a GUI builder tool.
- This GUI is implemented under Gnome using GUI builder tool, glade-3.
- Glade generates an XML file representing the GUI.
- Glade itself is an GUI allowing a user to see what the application’s GUI will look like.
- The GUI consists of a window, menu and status bars and button. All these things are known as GUI objects.
- Glade also allow a user to specify the names of the callback functions that get executed when the specified GUI object get selected.

Callback Functions

- Most of these objects have callback functions associated with them.
- Callback functions tie the GUI itself with the action that get executed.
- A callback performs the application specific action when the associated GUI object is selected.

Forth Callbacks

- Gforth has a callback feature that translates a Forth executable token to an address of a corresponding C function.
- This address can be assigned to a constant such that it can be passed as an argument to another C function.
- This allows writing a callback in Forth and have a GUI execute it as a C function.
- The callback declaration represent a Forth word as a C function prototype.
- An example, a GTK callback, void on_gtk_quit_active(GtkMenuItem *menuitem, gpointer user_data), It consist of two pointer parameters and will return a void, essentially nothing.
- In Gforth,, the callback declaration would be c-callback cb_a_a__void a a – void.
- The stack image for a callback is: (xt – cfunc-addr)

Forth C Function & Dynamic Libraries

- Forth has the capability to interface to C libraries such as the GTK GUI library and others.
- This capability allows using preexisting code that otherwise a user would have to write.
- An explanation would be for another presentation.

Hello, Forth! GUI - Building

- Use Glade-3.0 to create a window that contains a menu bar, a toggle button and a status bar. For the Help>About menu item, create a separate about dialog.
- For the window, assign a callback, on_window1_destroy, to the “destroy” signal handler to exit the application.
- For the File>Quit, assign a callback, on_gtk_quit_activate, to the “activate” signal handler to exit the application.

Hello, Forth! GUI - Building

- For the Help>About, assign a callback, on_gtk_about_active, to the “activate” signal handler to display the about dialog.
- For the toggle button, assign a callback, on_toggleButton1_toggled, to the “toggled” signal handler to toggle the text strings on the toggle button.

Callback - on_window1_destroy

- The Forth word, on_window1_destroy, is used to exit the application when the close icon is clicked.
- When called, it has a stack comment of: (gtk-window-addr user-data-addr –) like its signal handler “destroy”.
- Because this callback just destroys the whole application, the stack items will be dropped using, 2DROP.
- Next, the main GTK loop will be terminated using, gtk_main_quit.
- Finally, the Forth application will be terminated using, _exit.

Callback – on_gtk_quit_activate

- The Forth word, on_gtk_quit_activate, does the same as the on_window1_destroy callback and has the same stack comment.
- The code is also the same, except, instead of executing _exit, the Forth word, bye, is executed.

Callback - on_gtk_about_activate

- The Forth word, on_gtk_about_activate, is used to display the about dialog when the Help>About menu item is clicked.
- When called, it has a stack comment of: (gtk-dialog-addr user-data-addr –).
- This callback removes the user-data-addr using, NIP.
- Next, it duplicates the dialog address using, DUP.
- Next, it displays the about dialog using, gtk_dialog_run.
- Finally, after the dialog’s close button is clicked, the dialog is hidden using, gtk_widget_hide.

Callback - on_toggleButton1_toggled

- The Forth word, on_toggleButton1_toggled, is used to display the either, “Press Me!”, or, “Hello, Forth!”, when the toggle button is clicked.
- When called, it has a stack comment of: (gtk-toggleButton-addr user-data-addr –).
- This callback removes the user-data-addr using, NIP.
- Next, it duplicates the dialog address using, DUP.
- Next, it gets the current button label using, gtk_button_get_label.

Callback - on_toggleButton1_toggled

- Next, it is compared to the “Press Me!” string.
- If it matches, then, load the “Hello, Forth!” string.
- If it does not match, then load the “Press Me!” string.
- Next, call the gtk_button_set_label word to set the string in the toggle button’s label.

Callback - on_toggleButton1_toggled

- For the status bar, the status bar widget addres and context ID are pushed on the stack and duplicated using, 2DUP.
- The original status bar context are removed using, gtk_statusbar_pop.
- The current click count is processed into a string using, .clickCount.
- Finally, the new click count string is pushed to the status bar, gtk_statusbar_push.

Start-up Code

- Initially, the GTK system has to be initialized using, GTK_init, with the command line parameters, argc and argv.
- Next, a GTK builder structure has to be created using, gtk_builder_new. The resulting pointer is saved in a variable, builderPtr.
- In order to use the GUI’s XML file, it is used in an GTK call, gtk_builder_add_from_file. It requires as parameters, a pointer to a GTK builder structure, a C-string representing the name of the XML file and a pointer to a pointer for an error return or NULL.

Start-up Code

- Next, the individual widget pointer have to be extracted from the builder structure.
- Next, the callbacks have to be assigned to their respective widgets.
- Next, the status bar’s first message has to be created and assigned to the status bar.

Start-up Code

- Next, the whole GUI is displayed using, gtk_widget_show.
- Finally, the GTK menu processing is started using, gtk_main.

Forth and C Strings

- Forth strings are counted string, the first byte is the number of character in the string.
- C strings are ASCII NUL terminated.
- Forth creates counted strings, but C functions require C strings.
- Gforth can switch between both kinds of strings.
- sstring>cstring (forth-str – c-str)
- cstring>sstring (c-str – forth-str)

Thank you, any questions