Event Driven C

Thorsten Schulz

19.6.2014

Betreuer:
Michael Kuhn

Inhaltsverzeichnis

1 — Allgemeines

1.1 Was ist das überhaupt?

Unter Event Driven versteht man hauptsächlich als Ansatz zur nichtlinearen Steuerung des Programmflusses. Der Ablauf des Programmes ist im Vorfeld nicht bekannt und auch nicht unter der Kontrolle des Programmes, sondern lediglich bei den aufrufenden Events.

Dies lässt sich dieses Konzept beispielsweise sowohl in der GUI-Programmierung als auch in der Netzwerkprogrammierung wiederfinden. Dabei wird zusätzlich auch noch zwischen der eigentlichen Event Driven Programmierung und Signal Programmierung unterschieden. Dabei ähneln sich jedoch beide.

Im Allgemeinen lässt sich sagen, dass man zu aller erst event-handler benötigt, welche oft von der jeweiligen Programmierumgebung(hier GTK+ gegeben wird. Anschließend müssen die vorher gesetzten event-handler mit den zugehörigen events gekoppelt werden, sodass die gewünschte Funktionalität auftreten kann.
Im letzten Schritt muss die main loop aufgerufen/verwendet werden, welche auch meist von der jeweiligen Umgebung gestellt wird.

Eine genauere Darstellung des Prinzips findet sich später in Verbindung mit GTK+ in der Ausarbeitung.

2 — Vor-, Nachteile

Vorteile	Nachteile
Erlaubt mehr interaktive und nutzerfreundliche Programme	Programmfluss wird unübesichtlicher
Wartbarkeit: Programm ist in viele einzelne Teile unterteilt	Nicht für alle Aufgaben geeignet
Erweiterbarkeit Einfache Möglichkeit neue Funktionalitäten hinzuzufügen	

3 — Callbacks

In Programmiersprachen sind Callbacks(Rückruffunktionen) Funktionen, welche als Parameter übergeben werden können. Dabei gibt es je nach Programmiersprache unterschiede. In C wird dies mit Funktionspointern realisiert.

Oft müssen Anwendungen unterschiedliche Funktionen aufrufen, wenn eine bestimmte Bedingung erfüllt ist. Dies lässt sich hiermit bewerkstelligen.

Hier einmal zwei Beispiele zu Callbacks in C:

CallbackSample.c

```c
#include<stdio.h>
void callback(int *array)
{
    array[0] = 4;
    array[1] = 2;
}
void funktionmitpointer(int *array, void (*pointer)(int *array))
{
    pointer(array);
}

int main(void)
{
    int array[2];
    funktionmitpointer(array, callback);
    printf("%i%i", array[0], array[1]);
}
```

CallbackSample2.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void schreibeZahl(int (*source)(void))
{
    printf("Die Zahl ist %d\n", source());
}

int erzeugeZahl1(void)
{
    int geheim;
    for(int i=0;i <= 5; i++)
    {
        srand (time(NULL));
        geheim += (rand() % 1000 +1);
    }
    return geheim;
}

int erzeugeZahl2(void)
{
    return 24;
}

int main(void) {
    schreibeZahl(erzeugeZahl1);
    schreibeZahl(erzeugeZahl2);
}
```

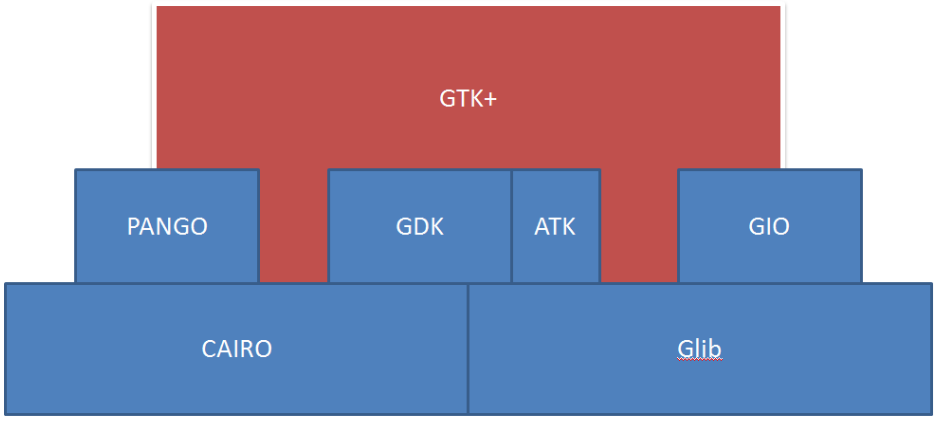
4 — GTK+

4.1 Was ist GTK+?

GTK+ ist eine multi-platform library für GUIs in C/C++ und nutzt das Prinzip des **Signal programming**
Dabei ist es aber auch unter Perl und Python nutzbar.

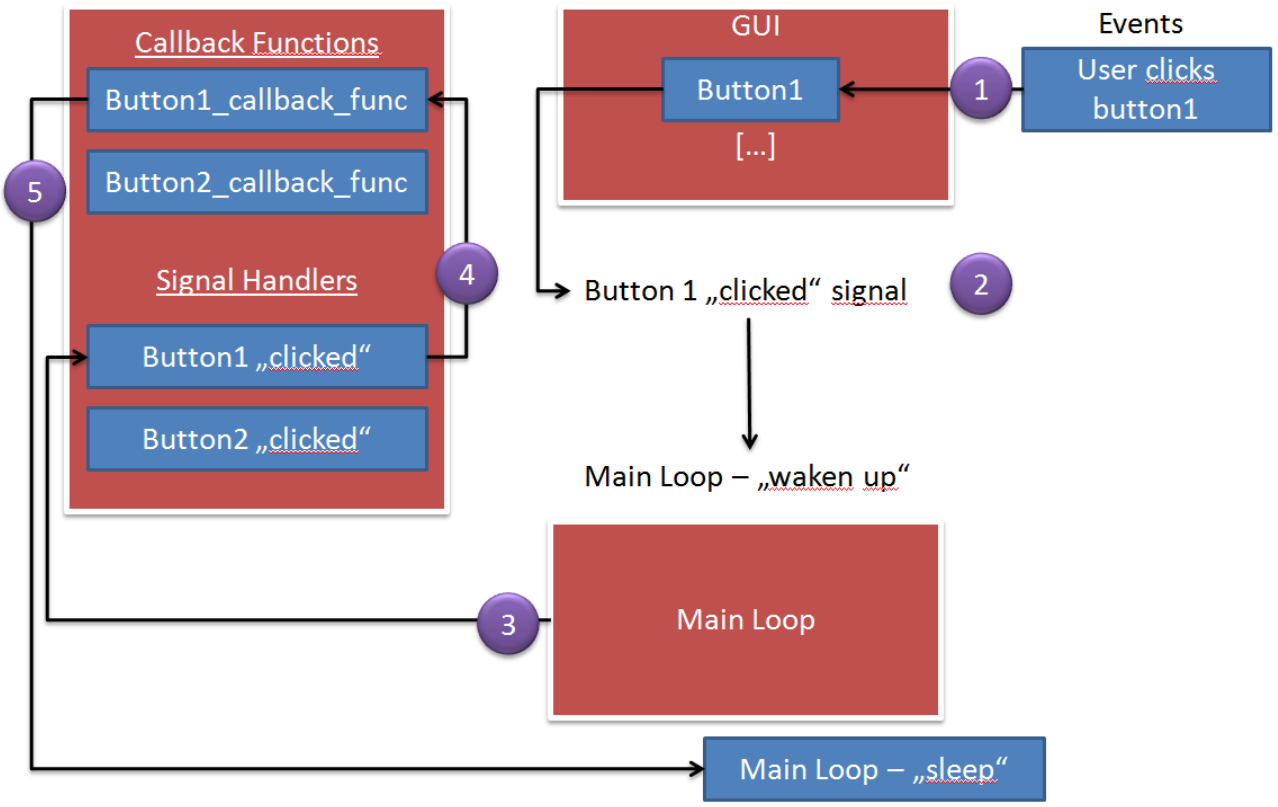
GTK+ oder auch bekannt als 'Gimp Toolkit' wurde ürsprünglich als Werkzeug zum erstellen der GUI für Gimp entwickelt worden, jedoch kann es heutzutage für die verschiedensten Anwendungsgebiete eingesetzt werden. Dabei baut sich GTK+ auf vier Bibliotheken auf:

- GLib...
  - ...wird als Kern der internen Datenstruktur verwendet und bietet Funktionen wie event-loop, threads, etc an (Siehe Präsentation: Gnome Library: GLib).
- Pango...
  - ... wird als Bibliothek mit Schwerpunkt auf Text- und Schriftartendarstellung genutzt.
- Cairo...
  - ...ist eine 2D-Grafik Bibliothek, welche sich rund um Input/Output ansiedelt.
- ATK...
  - ...ist eine Bibliothek, welche eine Basis zur Zugänglichkeit von alternativen Eingabegeräten und ähnlichen bietet.



4.2 GTK+ im Detail

Hier sieht man den ungefähren Ablauf einer simplen GTK+ Anwendung am Schaubild



Button1 erhält ein Event des Users und sendet ein Signal. Dabei wird die Main Loop genutzt, um darüber die zu dem Button1 verbundene Funktion mithilfe eines Callbacks aufzurufen. Wenn dieser Callback abgeschlossen ist, wartet die Main Loop auf die nächsten einkommenden Events.

4.3 GTK+ Funktionen

In GTK+ bietet eine Vielzahl an verschiedenen Funktionalitäten, welche die hier vorgestellten Funktionen weit überschreiten. Die vollständige Dokumentation von GTK+ lässt sich hier finden und erklärt den grundlegenden Aufbau der nutzbaren Funktionen:
https://developer.gnome.org/gtk3/stable/

Im folgenden stelle ich eine Übersicht über die wichtigsten Funktionen für den Anfang an einem Beispiel vor:

4.4 Beispiel

GTKSampleKommentiert.c

```c
   /*
#     GtkBook :
#        A Book for GTK+ users.
#     Copyright (C) July, 2005 Muthiah Annamalai
#
#     This program is free software; Licensed under GPL.
#
#    Source:
#    http://en.wikibooks.org/wiki/Computer_Programming/Event_driven_programming
*/
#include<gtk/gtk.h>

gboolean btn_clicked(GtkWidget *w,gpointer data)
{
  /* Diese Funktion erklärt sich selbst mit den in main() genutzten und beschriebenen Methoden */
  GtkButton *btn=GTK_BUTTON(w);
  gtk_button_set_label(btn,"You Clicked Me");
  return TRUE;
}

int main()
{
  /* GtkWidget ist die basis Klasse aller widgets in GTK+.
     Diese regelt den status und dessen repräsentation
     Beispielsweise GtkWidget --> GtkContainer --> GtBox --> GtkVBox */
  GtkWidget *w,*box,*btn;
  /* Initialisieren */
  gtk_init(NULL,NULL);
  /* Erstellt ein toplevel Fenster, welches andere widgets beinhalten kann
     (GtkWindowType) ist entweder GTK_WINDOW_TOPLEVEL oder GTK_WINDOW_POPUP
     GtkWidget *        gtk_window_new                    (GtkWindowType type); */
  w=gtk_window_new(GTK_WINDOW_TOPLEVEL);

  /* gtk_vbox_new ist ein container, welcher die Anordnung angibt (höhe, abstand und
     weitere GtkVBox widgets
     Achtung: GtkVBox: deprecated für die Zukunft GtkBox nutzen*/
  box=gtk_vbox_new(!FALSE,!FALSE);
  /* setzt den Titel "Button Widget" für das toplevel Fenster gtk_window_new (in diesem Fall w)

     void gtk_window_set_title (GtkWindow *window, const gchar *title); */
  gtk_window_set_title(GTK_WINDOW(w),"Button Widget");
  /* Wie im vorigen Beispiel gtk_window_set_title
     gtk_button_new_with_label (const gchar *label);  */
  btn=gtk_button_new_with_label("Hello World! Click Me:");
  /* Fügt dem Container w das GtkWidget btn hinzu
    void gtk_container_add (GtkContainer *container,  GtkWidget *widget); */
  gtk_container_add(GTK_CONTAINER(w),btn);
  /* Nutzt die GLib
     G_Callback ist die nutzbare callback Funktion, um den callback durchzuführen
     G_OBJECT wird genutzt um Attribute und Methoden für alle Objekttypen in GTK+ zu übergeben.
     void (*GCallback) (void);
      g_signal_connect(instance, detailed_signal, c_handler, data); */
  g_signal_connect(G_OBJECT(btn),"clicked",G_CALLBACK(btn_clicked),NULL);
  /* Setzt die Position des Toplevel Fensters in der Mitte Bildschirms
  void gtk_window_set_position (GtkWindow *window, GtkWindowPosition position); */
  gtk_window_set_position(GTK_WINDOW(w),GTK_WIN_POS_CENTER);
  /* Ruft alle widgets, welche in w enthalten sind rekursiv auf und zeigt diese wiederum an.
     void gtk_widget_show_all (GtkWidget *widget);  */
  gtk_widget_show_all(w);
  /* Am Ende wird die Hauptschleife(main loop) aufgerufen und gestartet. Diese Funktion
     kümmert sich um die eigentlichen Ereignisse  */
  gtk_main();
  return 0;
}

/*
gcc -o button button.c -Wall -ggdb `pkg-config gtk+-2.0 --cflags --libs`
*/
```

Weitere Anwendungsbeispiele lassen sich in der Dokumentation,
als auch gut Online finden.

Zum Thema compilen finden sich hier weitere Informationen:
https://developer.gnome.org/gtk3/stable/gtk-compiling.html
Ein weiteres Beispiel(Hello World) zu GTK+ lässt sich hier finden:
https://developer.gnome.org/gtk3/stable/gtk-getting-started.html
Eine Liste von weuteren alternaitven GUI-Bibliotheken findet sich hier:
http://de.wikipedia.org/wiki/Liste_von_GUI-Bibliotheken#C

5 — Aufgaben

5.0.1

Nenne die notwendigen Schritte, um einen Button in gtk+ zu implementieren.

5.0.2

Kommentiere das oben angegebene CallbackSample2.c.

5.0.3

Denke dir eine eigene sinnvolle Callbackfunktion aus, welche mindestens zwei Funktionen beinhaltet und implementiere diese anschließend.

5.0.4 *optional*

Schreibe deine erste simple Anwendung mit GTK+ mithilfe der
Dokumentation und dem bei **4** kommentierten Beispiel.

Der Link zu der GTK+ Dokumentation findet sich bei **4.3**

6 — Quellen

http://www.gtk.org/
letzter zugriff: 17.6.2014
https://developer.gnome.org/gtk3/stable/
letzter zugriff: 17.6.2014
http://en.wikipedia.org/wiki/Event-driven_programming
letzter zugriff: 15.6.2014
http://de.wikipedia.org/wiki/GTK%2B
letzter zugriff: 15.6.2014
http://en.wikipedia.org/wiki/Callback_%28computer_programming%29

Grafiken:
[1] http://3.bp.blogspot.com/-AOGh1YZD2I8/UffePaNkqlI/AAAAAAAAAq0/GnE72aYt27c/s1600/mainloop.png
[2] http://www.gtk.org/images/architecture.png