Event Driven C

Thorsten Schulz

19.6.2014

Gliederung

Event Driven C

Was ist das überhaupt?

Ein Ansatz zur nichtlinearen Steuerung des Programmflusses.
Dabei liegt der Ablauf des Programmes nicht unter dessen
Kontrolle, sondern lediglich bei den aufrufenden Events.

- Wo lässt sich dies wiederfinden?
  - GUI Programmierung
  - Netzwerkprogrammierung

Event Driven C

Was ist das überhaupt?

Ein Ansatz zur nichtlinearen Steuerung des Programmflusses.
Dabei liegt der Ablauf des Programmes nicht unter dessen Kontrolle, sondern lediglich bei den aufrufenden Events.

- Wo lässt sich dies wiederfinden?
  - GUI Programmierung
  - Netzwerkprogrammierung

Event Driven C

- Main Loop
- Events
  - Event Handling
  - Exception Handling

Vor-, Nachteile

Vorteile

- Erlaubt mehr interaktive Programme
- Wartbarkeit
- Erweiterbarkeit

Nachteile

- Programmfluss wird unübersichtlicher
- Plattformunabhängigkeit nicht immer gegeben
- Nicht für alle Aufgaben geeignet

Vor-, Nachteile

Vorteile

- Erlaubt mehr interaktive Programme
- Wartbarkeit
- Erweiterbarkeit

Nachteile

- Programmfluss wird unübersichtlicher
- Plattformunabhängigkeit nicht immer gegeben
- Nicht für alle Aufgaben geeignet

Callbacks

- Ein Callback bezeichnet eine Funktion, die eine andere Funktion als Parameter übergeben wird
- Dadurch mehr oder weniger eine Variable
- Unterscheiden sich in oft in verschiedenen Programmiersprachen. In C mit Funktionspointern

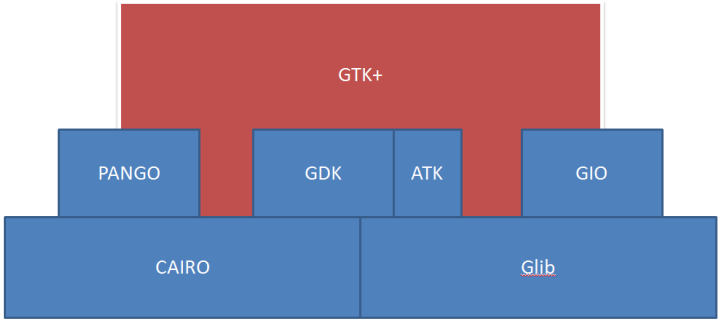
Callbackbeispiel

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]);
            }
```

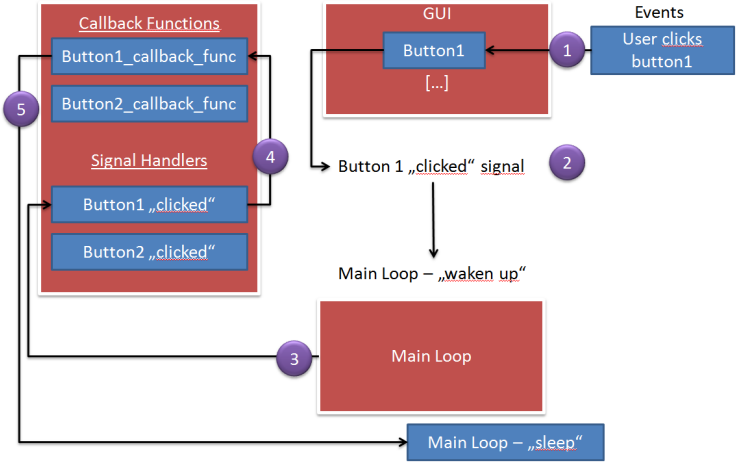
GTK+ Überblick

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.



[1]

GTK+ Main Event Loop



[2]

Grundlagen

- Was muss im Vorfeld getan werden?
  - Umgebung initialisieren
  - Widgets 'erzeugen'
  - Callback Funktionen hinzufügen, um erwünschte Funktionalitäten aufrufbar zu machen
  - Widget anordnung definieren
  - Widgets anzeigen
  - Eventloop aufrufen.

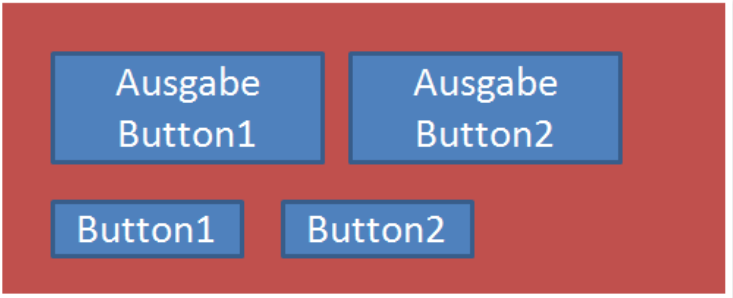
GTK+

GTKSamplePraesentation.c

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

gboolean btn_clicked(GtkWidget *w,gpointer data)
{
  GtkButton *btn=GTK_BUTTON(w);
  gtk_button_set_label(btn,"You Clicked Me");
  return TRUE;
}
int main()
{
  GtkWidget *w,*box,*btn;
  gtk_init(NULL,NULL);

  w=gtk_window_new(GTK_WINDOW_TOPLEVEL);

  box=gtk_vbox_new(TRUE,!FALSE);
  gtk_window_set_title(GTK_WINDOW(w),"Button Widget");

  btn=gtk_button_new_with_label("Hello World! Click Me:");
  gtk_container_add(GTK_CONTAINER(w),btn);

  g_signal_connect(G_OBJECT(btn),"clicked",G_CALLBACK(btn_clicked),NULL);
  gtk_window_set_position(GTK_WINDOW(w),GTK_WIN_POS_CENTER);
  gtk_widget_show_all(w);

  gtk_main();
  return 0;
}
```

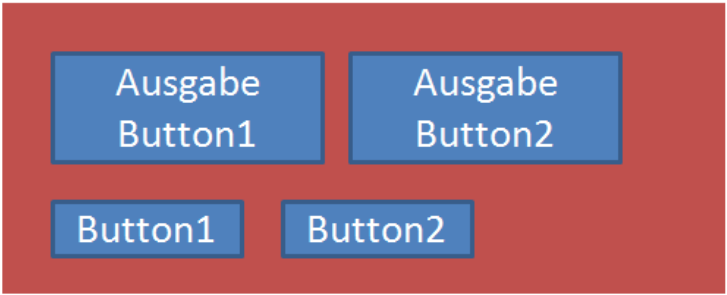
Aussicht

- Problem: responsiveness
  - Lösung: Threads



Aussicht

- Problem: responsiveness
  - Lösung: Threads



Literatur

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