GLib

Hans Ole Hatzel

14.6.2013

Was ist GLib?

- Library des Gnome Projekts.
- Enstand als Teil von GTK+
- Wurde ausgegliedert
- Multiplatform
- Bietet viele Funktionen

(5)

Was ist GLib nicht?

- GLib != glibc
- glibc Implementation der Standard C Library
- GLib bietet erweiterten Funktionsumfang

GLib verwenden

Include Statement:

```
#include <glib.h>
```

Package config:

```
$ gcc example.c `pkg-config --cflags --libs GLib-2.0`
```

Und benutzen:

```
gint x = 5;
```

Primitive Datentypen

g-Prefix

- gbool
- gint
- gint8

Datenstrukturen

- LinkedLists
- HashTables
- Bäume
- Wachsende Arrays
- Strings
- usw.

LinkedList

Beispiel LinkedList

Ein Element besteht aus:

- Seinen Daten
- Vorgänger
- Nachfolger

```
struct GList {
  gpointer data;
  GList *next;
  GList *prev;
};
```

LinkedList Benutzung

```
1  GList *liste;
2
3  liste = g_list_append(liste, "bli");
4  liste = g_list_append(liste, "bla");
5  liste = g_list_prepend(liste, "blup");
6
7  gchar *x = g_list_nth_data(liste, 0);
8  printf("value: %s", x);
```

LinkedList Funktionen

```
g_list_foreach(GList *list, GFunc func, gpointer user_data)
g_list_concat(GList *list1, GList *list2)
g_list_find(GList *list, gconstpointer data)
```

Macros

- Versionen
- Typeconversion
- Endianness

Eventloops

- Schleife wird permanent ausgeführt.
- Wartet auf Event

```
while(running){
    event = getEvent()
    process(event)
}
```

Eventloop in GLib

- Neue Eventloop
- Events hinzufügen
- main loop läuft
- Event tritt ein

Codebeispiel

```
1  gboolean callback(gpointer data){
2       g_print("timeout\n");
3       return TRUE;
4  }
5
6  int main(){
7       GMainLoop *schleife;
8       schleife = g_main_loop_new(NULL, FALSE);
9
10      g_timeout_add(1000, callback, schleife);
11      g_main_loop_run(schleife);
12 }
```

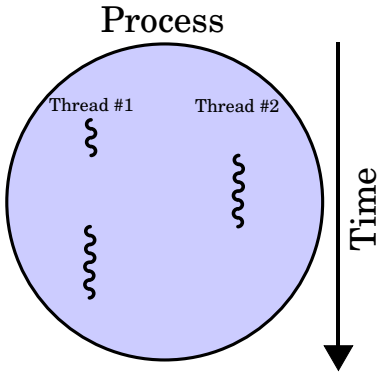
(3)

Anwendungsfall

- Spiele
- GTK+
- Benutzerinteraktion

Was sind Threads?

- Quasi parralele Ausführung
- dem Prozess untergeordnet
- teilen sich Speicher
- Bug anfällig (Threadsafety)



(4)

Threads in GLib

- Multiplatform
- GLib Worker Thread
- Threadsafe

Codebeispiel

```
1  gpointer writeToMem(gpointer data){
2      int *k = malloc(sizeof(int));
3      *k = 20;
4  }
5  int main(){
6      g_thread_new("thread1", writeToMem, NULL);
7      printf("int: %i\n", k);
8  }
```

GObject

- Objektsystem in C
- GUI Programmierung
- Objekte werden über Structs definiert

Utilities

- Zeit und Datum
- Parser
- Testing
- Regular Expressions

Protabilität

- Auf vielen Platformen verfügbar
- Macht Crossplatform programmieren einfacher
- Versuch einheitlich zu sein.

Einfacheit

- Praktische Datentypen
- Eventloops als bestehendes System
- Unterstützt GUI Development

Nachteile

- Embeddedsystems
- Dokumentation
- Große Library

Quellen

[1] ,,GLib Reference Manual" https://developer.gnome.org/glib/stable/

[2] ,,GLib - Wikipedia, the free encyclopedia" https://en.wikipedia.org/wiki/GLib

[3] ,,GLib Event Loop" http://docs.huihoo.com/symbian/nokia-symbian3-developers-library-v0.8/GUID-7FD05006-09C1-4EF4-A2EB-AD98C2FA8866.html

[4] ,,Thread (computing) - Wikipedia, the free encyclopedia" https://en.wikipedia.org/wiki/Thread_(computing)

[5] ,,Multithread programming" http://docencia.etsit.urjc.es/moodle/pluginfile.php/1668/mod_folder/content/1/Slides/GNOME/threads.pdf