

# GUI-Programmierung mit GTK+

## Erster Teil

Florian Pelz

E-Mail: [pelzflorian@pelzflorian.de](mailto:pelzflorian@pelzflorian.de)

XMPP: [pelzflorian@chat.pelzflorian.de](xmpp:pelzflorian@chat.pelzflorian.de)

# Heute...

Überblick

Installation

C-Grundlagen

Hallo GTK+!

Graphische GUI-Editoren

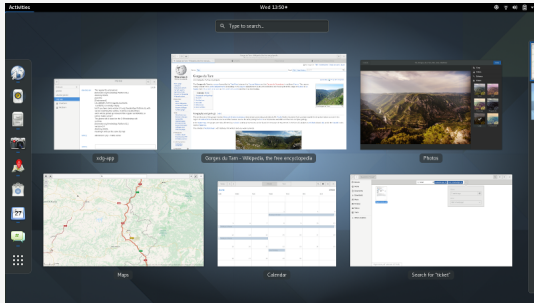
Ein eigener Texteditor

# GTK+



- ▶ GUI-Toolkit.
- ▶ Teil des GNOME-Projekts.

# GNOME



Beinhaltet:

- ▶ Benutzeroberfläche,
- ▶ Anwendungen,
- ▶ zu Grunde liegende Software-Bibliotheken wie GTK+.

# GNU



- ▶ Freies Betriebssystem.
- ▶ Studieren, Modifizieren und Weiterverbreiten ist erwünscht.



- ▶ GTK+ darf nur als freie Software weiterverbreitet werden.
- ▶ Software, die GTK+ benutzt, muss nicht frei sein.

THE  
**C**  
PROGRAMMING  
LANGUAGE

- ▶ Hardwarenahe Programmiersprache.
- ▶ GTK+ ist in C geschrieben.
- ▶ Auch mit anderen Sprachen nutzbar.
- ▶ Wir verwenden C.

## Installation von Entwicklerwerkzeugen

- ▶ Siehe Handouts.



# Einfaches C-Programm

```
1 int
2 main ()
3 {
4     return 5;
5 }
```

- ▶ Speichern als `fuenf.c`.
- ▶ Kompilieren und Testen:

```
$ gcc -o fuenf fuenf.c
$ ./fuenf
$ echo $?
5
```

# Daten speichern

```
1 int
2 main ()
3 {
4     int a; /* Variable a deklarieren. */
5     a = 5; /* 5 in a speichern.      */
6     return a;
7 }
```

## while-Schleife

```
1  int
2  main ()
3  {
4      int a;
5      int i;
6      a = 1;
7      i = 1;
8      while (i < 5)
9          {
10             a = a * i;
11             i = i + 1;
12         }
13     return a;
14 }
```

```
1  int
2  main ()
3  {
4      int a, i;
5      a = i = 1;
6      while (i < 5)
7          {
8             a *= i;
9             i++;
10         }
11     return a;
12 }
```

## for-Schleife

```
1 int
2 main ()
3 {
4     int a, i;
5     a = 1;
6     for (i = 1; i < 5; i++)
7     {
8         a *= i;
9     }
10    return a;
11 }
```

```
1 int
2 main ()
3 {
4     int a, i;
5     a = 1;
6     for (i = 1; i < 5; i++)
7         a *= i;
8     return a;
9 }
```

# Funktionsaufrufe

```
1  int
2  fakultaet (int n)
3  {
4      if (n == 0)
5          return 1;
6      else
7          return n * fakultaet (n-1);
8  }
9
10 int
11 main ()
12 {
13     return fakultaet (4);
14 }
```

# Modularisierung

► fakultaet.c:

```
1 int
2 fakultaet (int n)
3 {
4     if (n == 0)
5         return 1;
6     else
7         return n * fakultaet (n-1);
8 }
```

► main.c:

```
1 int fakultaet (int n);
2
3 int
4 main ()
5 {
6     return fakultaet (4);
7 }
```

► \$ gcc -o main fakultaet.c main.c

# Header

► fakultaet.h:

```
1 int fakultaet (int n);
```

► main.c:

```
1 #include "fakultaet.h"  
2  
3 int  
4 main ()  
5 {  
6     return fakultaet (4);  
7 }
```

# Standardbibliothek

```
1 #include <stdio.h>
2 #include "fakultaet.h"
3
4 int
5 main ()
6 {
7     printf ("Hallo_\u00a0Welt!\n");
8     printf ("4!_\u00a0=\u00a0%d.\n", fakultaet (4));
9     return 0;
10 }
```



## Eigene Datentypen

```
1 struct _Vector {
2     double x;
3     double y;
4     double z;
5 };
6
7 typedef struct _Vector Vector;
```

# Hallo GTK+!

- ▶ Siehe Handouts.
- ▶ 

```
$ gcc `pkg-config --cflags gtk+-3.0` -o hello  
hello.c `pkg-config --libs gtk+-3.0` -Wall
```

# Hallo GTK+!

- ▶ Fügen wir Text hinzu!

```
1 int
2 main ()
3 {
4     GtkWidget *window;
5     GtkWidget *hello_label;
6     gtk_init (0, NULL);
7     window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
8     hello_label = gtk_label_new ("Hello World!");
9     gtk_container_add (GTK_CONTAINER (window),
10                        hello_label);
11     ...
```

# Hallo GTK+!

- ▶ Und einen Knopf → Handouts.

# Was kann GTK+?

- ▶ Was kann man noch machen?
  - ▶ Text und Knopf vertauschen.
  - ▶ Mit dem Knopf den Text ändern.
  - ▶ ...
- ▶ <https://developer.gnome.org/>
- ▶ `gtk3-widget-factory`
- ▶ `gtk3-demo`
- ▶ `gtk3-icon-browser`, `gnome-characters`, ...
- ▶ <https://git.gnome.org>, Mailing-Listen, IRC, ...

Pause.

# Graphische GUI-Editoren

- ▶ Eine GUI kann auch
  1. mit Glade graphisch zusammengestellt,
  2. als XML-Datei gespeichert und
  3. dann mit GTK+ geladen werden.
- ▶ Siehe Handouts.

- ▶ Das waren die GTK+-Grundlagen.
- ▶ Wie kann man nun üben?
  - ▶ Ein eigener Texteditor
  - ▶ oder ein Bildbetrachter
  - ▶ oder ein Taschenrechner
  - ▶ oder Galgenmännchen
  - ▶ oder ...
  - ▶ und dann ein Mini-Inventarverwaltungssystem.



## Bildquellen etc.

- ▶ Folien-Design basiert auf <https://git.gnome.org/browse/presentation-templates/>.
- ▶ GNOME-Screenshot von <https://people.gnome.org/~engagement/screenshots/> CC-BY-SA GNOME Project.
- ▶ GNU-Kopf von [https://www.gnu.org/graphics/heckert\\_gnu.html](https://www.gnu.org/graphics/heckert_gnu.html) u.A. CC-BY-SA Etienne Suvasa, Peter Gerwinski. Markenrechtlich geschützt.
- ▶ Weitere Bilder aus Wikimedia Commons:  
[https://en.wikipedia.org/wiki/File:GTK%2B\\_logo.svg](https://en.wikipedia.org/wiki/File:GTK%2B_logo.svg),  
<https://en.wikipedia.org/wiki/File:Copyleft.svg>,  
[https://en.wikipedia.org/wiki/File:The\\_C\\_Programming\\_Language\\_logo.svg](https://en.wikipedia.org/wiki/File:The_C_Programming_Language_logo.svg)
- ▶ Ideen zum Üben teils aus „Foundations of GTK+ Development“ von Andrew Krause.
- ▶ Übriges von mir CC-BY-SA, Code auch CC0. Siehe <https://pelzflorian.de/git/gui-prog-gtk-2016/>.

