

GUI Programming with GTK+

First Part

Florian Pelz

E-Mail: pelzflorian@pelzflorian.de

Today...

Overview

Installation

C Basics

Hello GTK+!

Graphical GUI Editors

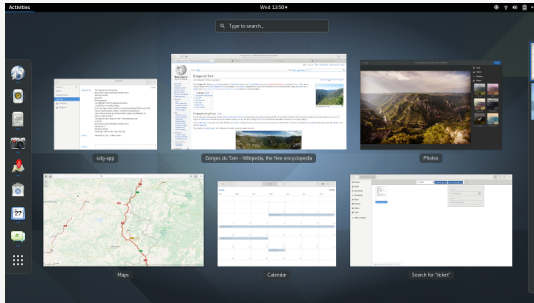
Your own text editor

GTK+



- ▶ GUI toolkit.
- ▶ For windowed applications, not Web apps.
- ▶ Part of the GNOME project.

GNOME



Contains:

- ▶ Desktop environment,
- ▶ applications,
- ▶ underlying software libraries such as GTK+.

GNU



- ▶ Free operating system.
- ▶ Studying, modifying and redistributing is welcome.

GNU LGPL



- ▶ GTK+ may only be redistributed as free software.
- ▶ Software using GTK+ need not be free.

THE
C
PROGRAMMING
LANGUAGE

- ▶ Low-level programming language.
- ▶ GTK+ is written in C.
- ▶ Also usable with other languages.
- ▶ We will use C.

Installation of developer tools

- ▶ See handouts.

Simple C program

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

- ▶ Save as `five.c`.
- ▶ Compile and test:

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

Store data

```
1 int
2 main (void)
3 {
4     int a; /* Declare variable a. */
5     a = 5; /* Store 5 in a.      */
6     return a;
7 }
```

while loop

```
1  int
2  main (void)
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 (void)
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 loop

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

Function calls

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

Modularization

▶ factorial.c:

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

▶ main.c:

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

▶ \$ gcc -o main factorial.c main.c

Header

▶ factorial.h:

```
1 int factorial (int n);
```

▶ main.c:

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

Standard library

```
1 #include <stdio.h>
2 #include "factorial.h"
3
4 int
5 main (void)
6 {
7     printf ("Hello␣world!\n");
8     printf ("4!␣=␣%d.\n", factorial (4));
9     return 0;
10 }
```


Custom data types

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

Hello GTK+!

- ▶ See handouts.
- ▶

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

Hello GTK+!

- ▶ Let's add text!

```
1 int
2 main (void)
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     ...
```

Hello GTK+!

- ▶ And a button → handouts.

What can GTK+ do?

- ▶ What else to do?
 - ▶ Swap text and button.
 - ▶ Have the button change the text.
 - ▶ ...
- ▶ <https://developer.gnome.org/>
- ▶ `gtk3-widget-factory`
- ▶ `gtk3-demo`
- ▶ `gtk3-icon-browser`, `gnome-characters`, ...
- ▶ <https://openclipart.org> etc.
- ▶ <https://git.gnome.org>, mailing lists, IRC, ...

Break.

Graphical GUI Editors

- ▶ A GUI can also be
 1. graphically composed using Glade,
 2. saved to an XML file and then
 3. be loaded from GTK+.
- ▶ See handouts.

- ▶ So much for the GTK+ basics.
- ▶ How to practice?
 - ▶ Make your own text editor
 - ▶ or image viewer
 - ▶ or calculator
 - ▶ or dice
 - ▶ or ...
 - ▶ and then a small Inventory Management System.

Image sources etc.

- ▶ Slide design based on <https://git.gnome.org/browse/presentation-templates/>.
- ▶ GNOME screenshot from <https://people.gnome.org/~engagement/screenshots/> CC-BY-SA GNOME Project.
- ▶ GNU head from https://www.gnu.org/graphics/heckert_gnu.html CC-BY-SA Etienne Suvasa, Peter Gerwinski among others. Trademarked.
- ▶ More images from Wikimedia Commons:
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
- ▶ Ideas for practice partially from “Foundations of GTK+ Development” by Andrew Krause.
- ▶ Everything else by me CC-BY-SA, code additionally CC0. See <https://pelzflorian.de/git/gui-prog-gtk-2017/>.

- ▶ Thank you to Kathrin, Nicole, Michele, Jan, Yasmina and other helpers of the Fachschaft Informatik.

