

9.1 GTK/Einführung

9.1.1 Vorbereitungen

In dieser Reihe sollen Sie lernen, mit der GTK-Bibliothek eine Anwendung mit graphischer Oberfläche zu programmieren. Dazu müssen Sie die Bibliothek zusammen mit ihren Header-Dateien installieren. Unter Debian und Ubuntu geben Sie ein:

```
Terminal
schueler@debian964:~$ apt-get install libgtk2.0-dev
```

9.1.2 Einführendes Beispiel

Hier ist der Quelltext eines einfachen GTK-Programms (beispiel1.c):

```
1 #include <gtk/gtk.h>
2 int main(int argc, char *argv[])
3 {
4     GtkWidget* fenster;
5
6     gtk_init(&argc, &argv);
7     fenster=gtk_window_new(GTK_WINDOW_TOPLEVEL);
8     gtk_widget_show_all(fenster);
9     gtk_main();
10    return 0;
11 }
```

Das Programm wird compiliert und aufgerufen mit:

```
Terminal
schueler@debian964:~$ gcc `pkg-config --cflags --libs gtk+-2.0` -o beispiel1 beispiel.c
schueler@debian964:~$ ./beispiel1
```

Einfacher ist es, das Programm make zu benutzen. Das entsprechende Makefile sieht so aus:

```
1 ZIEL=beispiel
2 GTKVERSION=gtk+-2.0
3 # GTKVERSION=gtk+-3.0
4 CFLAGS=`pkg-config --cflags --libs $(GTKVERSION)`
5
6 $(ZIEL): $(ZIEL).c
7     gcc $(CFLAGS) -o $(ZIEL) $(ZIEL).c
8
9
10 run: $(ZIEL)
11     ./$(ZIEL)
12
13 clean:
14     rm *~ a.out
```

Der Aufruf ist dann:

```
Terminal
schueler@debian964:~$ ZIEL=beispiel1 make -e run
```

9.1.3 Bedeutung der Programmzeilen

Zeile 1 Es reicht, `gtk.h` einzubinden; dadurch werden alle weiteren Header-Dateien automatisch eingebunden.

Zeile 4 `fenster` ist ein Zeiger auf eine Datenstruktur, die die Eigenschaften des Hauptfensters aufnehmen soll. Der Datentyp `GtkWidget` kann grob gesagt alles aufnehmen, was sich auf dem Bild anzeigen lässt (*widget* ist ein Kunstwort aus *window gadget*).

Zeile 6 `gtk_init()` nimmt die Kommandozeilen-Argumente heraus, die für GTK bestimmt sind und lässt den Rest unberührt.

Zeile 7 Hier wird die Datenstruktur für das Hauptfenster erzeugt und ein Zeiger darauf an `fenster` zurückgegeben. Der Prototyp für diese Funktion lautet:

```
GtkWidget *gtk_window_new (GtkWindowType type);
```

Der Parameter `type` hat hier den Wert `GTK_WINDOW_TOPLEVEL`. Das zeigt an, dass es sich um ein normales Fenster mit Rahmen und Titel handelt. Anderfalls müsste man den Wert `GTK_WINDOW_POPUP` benutzen.

Zeile 8 Das Fenster wird zur Anzeige vorbereitet.

```
void gtk_widget_show_all (GtkWidget *widget);
```

Man könnte hier auch genauso den Aufruf `gtk_widget_show()` nehmen. Bei `gtk_widget_show_all()` werden zusätzlich alle Unterelemente des Fensters angezeigt. In diesem Programm gibt es zwar keine, aber für weitere Programme ist diese Eigenschaft nützlich.

Zeile 9 Hier finden wir die Hauptschleife des GTK-Programms. Nun wird das Fenster wirklich angezeigt.

Wenn man GUI-Programmierung *ohne* eine Toolbox (Werkzeugkasten) wie GTK macht, muss man die Hauptschleife selbst programmieren. Und man muss für jedes Element, also jedes Fenster, jeden Knopf, eine Abfrage einbauen, ob dieses Element ein Signal bekommen hat. Wenn ja, wird in der Hauptschleife in eine Verzweigung gelaufen, in der das Signal behandelt wird. *Mit* einer Toolbox wie GTK wird uns die Arbeit abgenommen.

Man kann das Programm wie jedes Programm von der Kommandozeile starten. Zum Beenden muss man im Fenster der Kommandozeile die Tastenkombination `[Strg] - [C]` drücken.

Hilfen zu GTK kann man zwar auch installieren; es empfiehlt sich aber noch mehr die Online-Hilfe unter <https://developer.gnome.org/gtk2/stable/>.

9.1.4 Unser Fenster soll schöner werden

Nun soll das Fenster einen aussagekräftigen Titel bekommen. Dazu fügt man zwischen der Erzeugung der Fenster-Struktur mit `gtk_window_new()` und der Hauptschleife (`gtk_main()`) einen Funktionsaufruf ein:

```
1 gtk_window_set_title(GTK_WINDOW(fenster), "Wat'n_super_Programm!");
```

Damit wird der Fenstertitel gesetzt. Auffällig ist dabei die Stelle mit `GTK_WINDOW()`. Dies ist keine neue Funktion, sondern nur ein Makro. Es prüft, ob es sich bei dem, worauf `fenster` zeigt, wirklich um ein Fenster handelt, denn nur dann kann man den Titel auch setzen. Falls ja, wird noch netterweise eine Typumwandlung (von Zeiger auf `GtkWidget` nach Zeiger auf `GtkWindow`) gemacht. Falls nein, erhält man eine willkommene Fehlermeldung.

9.1.5 Für Experten: Wozu dient das Makro?

Verzichtet man auf das Makro, bekommt man eine Warnung, dass man einen nicht kompatiblen Zeiger verwendet. Das liegt am Prototypen von `gtk_window_set_title()`:

```
void gtk_window_set_title (GtkWindow *window, const gchar *title);
```

Der erste Parameter ist nämlich nicht ein Zeiger auf `GtkWidget`, sondern ein Zeiger auf `GtkWindow`. Und `GtkWindow` ist eine Struktur, die ein paar Daten mehr enthält als ein einfaches `GtkWidget`, z. B. einen Fenstertitel.

Ohne das Makro kann es passieren, dass man (etwa durch einen Programmierfehler) einen Fenstertitel setzt für irgendein Widget, das gar kein Fenster ist, z. B. für eine Schaltfläche. Das Programm schreibt dann Daten an eine Stelle, an die sie nicht gehören, und das kann in C böse Auswirkungen haben. Solange man also noch nicht überall genau weiß, was man tut, sollte man diese Makros benutzen.