

GUI-Programmierung mit GTK+

Zweiter Teil

Florian Pelz

E-Mail: `pelzflorian@pelzflorian.de`
XMPP: `pelzflorian@chat.pelzflorian.de`

Heute...

Distribution

Mini-IMS

Internationalisierung

Weiterführendes

Letztes Mal

Wir haben behandelt

- ▶ wie man C-Programme schreibt,
- ▶ wie man in C mit GTK+ Fenster-Anwendungen programmiert,
- ▶ wie man den graphischen Glade-Editor dabei benutzen kann und
- ▶ ein kleines Programm entwickelt.

Probleme

Allerdings

- ▶ wollen Endnutzer ihre Software nicht erst kompilieren und
- ▶ das Kompilieren mit vielen Terminalbefehlen ist auch für Entwickler umständlich.

Maintainer

- ▶ Maintainer übernehmen eine wichtige Rolle für die Software-Infrastruktur.
- ▶ In einem Software-Projekt sorgen Maintainer dafür, dass die Software
 - ▶ leicht konfiguriert, kompiliert, getestet etc. werden kann,
 - ▶ auf allen Systemen läuft, wo sie laufen soll,
 - ▶ leicht an Endnutzer ausgeliefert werden kann und
 - ▶ die übrige Infrastruktur (Webseite, Wiki, Bug-Tracker, ...) funktioniert,
 - ▶ gute und nur gute Patches akzeptiert werden,
 - ▶ der Change Log bzw. die News sauber geführt werden,
 - ▶ alle Autoren und Copyrights genannt werden,
 - ▶ ...
- ▶ Maintainer für ein Betriebssystem nutzen die Infrastruktur, um die Software an Endnutzer auszuliefern.

GNU Autotools

- ▶ `autoconf` automatisiert
 - ▶ das Finden von Betriebssystemwerkzeugen,
 - ▶ das Feststellen von Betriebssystemfähigkeiten,
 - ▶ Compile-Time-Konfigurationen (z.B. Compiler-Auswahl).
- ▶ `automake` automatisiert
 - ▶ das Kompilieren der Software,
 - ▶ das Testen (Beweisen?) der Software,
 - ▶ das Installieren der Software und
 - ▶ den Release einer Version kompilierfähigen Quellcodes.
- ▶ Wir wollen das für unser Programm nutzen!
- ▶ Siehe Handouts.

Versionskontrollsysteme

- ▶ Versionskontrollsysteme wie `git` verwalten mehrere Versionen des Quellcodes.
- ▶ Sie erlauben
 - ▶ paralleles Arbeiten mehrerer Entwickler am gleichen Code,
 - ▶ paralleles Arbeiten eines Entwicklers an unterschiedlichen Teilen des Codes,
 - ▶ eine leichte Rückkehr zu alten Versionen und
 - ▶ leichtes Finden der Version, mit der ein Bug zum ersten Mal auftrat.

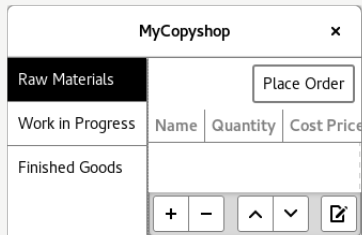
Für die Endnutzer: Pakete

- ▶ Auf GNU/Linux u.Ä. wird Software oft mit Paketmanagern ausgeliefert.
- ▶ Sie bestehen meist hauptsächlich aus
 - ▶ entweder einem getesteten automatischen Skript zum Kompilieren und Installieren
 - ▶ oder durch einen Skript werden die fertigen Dateien der Software an ihren betriebssystemspezifischen Platz.
- ▶ In der Regel heißt das, Pakete werden mit `./configure`, `make` und `make install` erstellt.
- ▶ Details hängen vom Paketmanager ab.

Für die Endnutzer: Bundles

- ▶ Auf den meisten Betriebssystemen sind auch (evtl. erst nach Installation) selbstständig lauffähige Bundles üblich.
- ▶ Ein Bundle ist ein Verzeichnis mit der Software *und allen Abhängigkeiten*.
- ▶ Metadaten und Konfigurationsdateien befinden sich unter Umständen außerhalb des Bundles.
- ▶ Ein solches Bundle kann dann z.B. zum Download im Web, in einem Appstore oder auf physischen Datenträgern vertrieben werden.
- ▶ Sicherheitsrisiko?
- ▶ Quellcode mitliefern oder verfügbar haben!

Mini-IMS



- ▶ Wir bauen uns zum Üben ein kleines Inventarverwaltungssystem für einen fiktiven Copyshop.
- ▶ Da das ein „großes Projekt“ ist, verwenden wir Objekte:
 - ▶ Vererbung.
 - ▶ Modularisierung.
 - ▶ Andere Sprachen.

Lokalisierung

- ▶ Bezeichnet das Anpassen einer Software an die lokale Sprache und Kultur.
 - ▶ Übersetzung,
 - ▶ Darstellung von Datum und Uhrzeit,
 - ▶ Maßeinheiten,
 - ▶ Dezimaltrennzeichen,
 - ▶ ...
- ▶ Kurz: l10n.

Internationalisierung

- ▶ Bezeichnet das Bereitstellen der zur Lokalisierung nötigen Infrastruktur.
- ▶ Kurz: i18n.
- ▶ PO-Dateien (wie „de.po“) beinhalten Übersetzungen zu allen sichtbaren Zeichenketten.
- ▶ Im Code benutzt man statt

```
label = gtk_label_new ("Hello World");
```

nun

```
label = gtk_label_new (_("Hello World"));
```

- ▶ Die Funktion `_()` von GNU Gettext liefert die passende Übersetzung.
- ▶ Das `_()` kann auch als „Markierung“ der Zeichenkette verstanden werden.
- ▶ Ähnliche Markierungen geben noch Kontext für Übersetzer mit, z.B. wo im Programm die Zeichenkette auftaucht.
- ▶ Siehe Handouts.

Accessibility

- ▶ D.h. Software Unterstützungsprogrammen zugänglich machen und „behindertengerecht“ gestalten.
- ▶ Accessibility umfasst:
 - ▶ Kompatibilität mit Screenreadern,
 - ▶ Kompatibilität mit großer Schrift,
 - ▶ Farbwahl und Kontrast,
 - ▶ ...
- ▶ Kurz: a11y.
- ▶ In GTK+ größtenteils automatisch.
- ▶ Aber ATK sollte z.B. zum Angeben von Beziehungen zwischen Widgets benutzt werden.

Unit-Tests

- ▶ Fehler im Programm findet man u.A. durch Testen.
- ▶ GLib enthält ein Framework für Unit-Tests.
- ▶ Autotools kann darin geschriebene Tests ausführen.
- ▶ Siehe z.B. den Quellcode von libsoup für ein Beispiel.
- ▶ reftests vergleichen das Aussehen Pixel für Pixel
→ wird in GTK+ benutzt.

Statische und dynamische Analyse

- ▶ Statisch: Ohne das Programm laufen zu lassen.
 - ▶ Mehrere Compiler.
 - ▶ coala für Coding-Style-Fehler.
 - ▶ ...
- ▶ Dynamisch:
 - ▶ Valgrind.
 - ▶ Profiler.

Profiling

- ▶ Ein Profiler misst, welche Teile des Programms
 - ▶ wie oft laufen und
 - ▶ wie lange brauchen.
- ▶ So weiß man, welcher Code optimiert werden sollte.

gtk-doc

- ▶ Code kann in den Quelldateien dokumentiert
- ▶ gtk-doc erzeugt daraus eine HTML-Dokumentation.
- ▶ Diese kann z.B. mit Devhelp gelesen werden.

Introspection

- ▶ Erlaubt das Abfragen von Informationen über Typen und Klassen:
 - ▶ Methodennamen,
 - ▶ Signale,
 - ▶ Properties,
 - ▶ ...
- ▶ Erlaubt automatisches Generieren von Bindings für andere Programmiersprachen.

D-Bus

- ▶ Für Inter-Process Communication.
- ▶ Insbesondere für Kommunikation mit der Benutzeroberfläche:
 - ▶ Aufrufen von Aktionen im Programm,
 - ▶ Notifications,
 - ▶ ...

CSS

- ▶ CSS ermöglicht es, das Aussehen von Widgets in GTK+ zu verändern.
- ▶ Noch in Arbeit.

Weiteres

- ▶ GSettings speichert Einstellungen, alten Programmzustand etc.
- ▶ GResource ermöglicht das Kompilieren von Binärdaten in die Anwendung hinein.
- ▶ GtkInspector zeigt und ändert GTK+-Informationen zu laufenden Programmen.
- ▶ `g_autoptr` lässt Speicher automatisch freigeben.
- ▶ Human Interface Guidelines beachten.
- ▶ Nicht alles braucht eine GUI.

Ressourcen

- ▶ GLib- und GTK+-Lehrbuch:
<https://people.gnome.org/~swilmet/glib-gtk-dev-platform.pdf>
- ▶ Für C++-Programmierer:
<https://developer.gnome.org/gtkmm-tutorial/stable/>
- ▶ Internationalisierung:
<https://wiki.gnome.org/TranslationProject/DevGuidelines>

