



Wir wollen Meson verwenden, um auch größere Projekte leicht bauen, testen, verbreiten, etc. zu können.

Dazu sollte unser Projekt eine übersichtliche Ordnerstruktur haben. Wir erstellen zum Beispiel

- in einem neuen, leeren Projektordner
 - eine Textdatei `README`, die später grundlegende Erklärungen, wofür die Software da ist, sowie Installationsanweisungen enthalten und die Lizenz nennen soll,
 - eine Datei `AUTHORS`, die später die Namen aller Entwickler auflisten soll,
 - eine Datei `NEWS`, welche später nur die für Nutzer wichtigen Änderungen zwischen den Programmversionen knapp auflisten soll,
 - eine Datei `COPYING` oder `LICENSE`, die die Software-Lizenz enthalten soll (bei mehreren Lizenzen sollte stattdessen ein Ordner `LICENSES` die Lizenzen in je einer Datei enthalten, vgl. <https://reuse.software>),
 - eine noch leere Textdatei namens `meson.build`,
 - ein Unterverzeichnis `src` oder `mycopyshop` oder Ähnliches
 - * mit einer Datei namens `main.c`, die später die `main()`-Funktion enthalten wird,
 - * und einer weiteren `meson.build`-Datei,

als Grundgerüst. In *Markdown* geschriebene Textdateien können auch zusätzlich die Dateiendung `.md` bekommen. Im `src`-Ordner werden später alle C-Quelldateien des Programms stehen.

Wir füllen nun die `meson.build`-Dateien mit den Informationen, woraus sich unser Software-Projekt zusammensetzt.

Im Hauptprojektordner sollte in `meson.build` stehen:

```
project('mycopyshop', 'c')
gtkdep = dependency('gtk+-3.0', version : '>=3.22')
subdir('src')
```

In `project()` sollte in echten Projekten auch die Programmlizenz und -version angegeben werden, siehe <http://mesonbuild.com>. Die nötige GTK+-Version wurde in diesem Beispiel als mindestens 3.22 angegeben.

In `src/meson.build` sollte stehen:

```
executable('copyshop', 'main.c',
  dependencies : gtkdep,
  install : true,
  gui_app : true)
```

Fertig! Wir wollen den nun erreichten Zustand unseres Projekts in einem Git-Repository sichern. Wir öffnen dazu im ursprünglichen Projektordner ein Terminal und führen folgende Befehle aus:

```
$ git init
$ git add README AUTHORS NEWS COPYING meson.build
  src/main.c src/meson.build
```

Mit `git` kann leicht verwaltet werden, welche Dateien zum Projekt dazugehören. Alle fremden Dateien können mit `git clean -df` gelöscht werden. Mit `git status` wird der Zustand der Dateien im Projektordner angezeigt. Hauptsächlich wird Git aber meistens benutzt, um Änderungen am Projekt in je einer eigenen Version („Commit“) zu speichern und zu verwalten, wodurch u.A. mehrere Entwickler effizient zusammenarbeiten können.

Nachdem die Änderungen nun nicht mehr leicht verloren gehen, bauen und installieren wir das Projekt wie folgt für einen einzelnen Nutzer:

```
$ mkdir builddir
$ cd builddir
$ meson ..
$ ninja
$ sudo ninja install
```

Administratoren können das Projekt auch für alle Nutzer installieren. Wenn nun aber noch Glade-Dateien oder Ähnliches geladen werden sollen, müssen sie als Ressource ins Programm eingebunden oder deren Pfad in einer Konfigurationsdatei an den C-Code übermittelt werden. Wir brauchen für letzteres eine `config.h`-Datei; siehe für deren Erstellung das nächste Handout oder das Beispiel:

<https://pelzflorian.de/git/gui-prog-gtk/tree/bin/copyshop/meson.build>