

*Talent wins games, but teamwork and  
intelligence wins championships.*  
Michael Jordan



UNIVERSITÄT  
DES  
SAARLANDES

FR Mathematik  
Andreas Buchheit

## 12. (und letzte) Übung zur Vorlesung Programmierung im Sommersemester 2019

Abgabe: Mittwoch, den 10.07.2019 bis spätestens 12 Uhr.

### Aufgabe 12.1. (7 Punkte) Projektmanagement und Bibliotheken

In diesem letzten Übungsblatt der Vorlesung sollen Sie die Grundlagen des Projektmanagements und der Erzeugung und Verwendung von Bibliotheken einüben.

- (a) **(3 Punkte)** Im letzten Übungsblatt haben Sie doppelt verkettete Listen implementiert. Schreiben Sie nun Funktionsprototypen für alle verwendeten Funktionen und fügen Sie diese zusammen mit den Definitionen der verwendeten Datenstrukturen in ein Header-File `list.h` ein und platzieren Sie es in einem Unterorder `include/` ihres Projektverzeichnisses. Versehen Sie das Header-File mit einem Define-Guard, um doppeltes inkludieren zu verhindern. Erklären Sie in einem Kommentar am Anfang des Files, welche Funktionalität bereitgestellt wird. Kommentieren Sie kurz, aber präzise die verwendeten Datenstrukturen. Kommentieren Sie vor jedem Funktionsprototypen, was die zugehörige Funktion tut und wie sie zu verwenden ist. Ihr Ziel ist es, dass das Interface klar ist, so dass die Funktionen verwendet werden können, ohne die genauen Details der Implementierung zu kennen.
- (b) **(2 Punkte)** Schreiben Sie die Implementierung der Funktionen in ein zweites file `list.c` und platzieren Sie dieses in einem Ordner `src/modules/list/`. Inkludieren Sie das zugehörige Header-File. Erzeugen Sie eine Objekt-Datei `file.o` in einem Ordner `build` im Projektordner wobei Sie mit der Flag `-I` den Pfad des `include` Ordners angeben. Vermerken Sie den verwendeten Kommandozeilenbefehl zur Erstellung der Objektdatei in einem Kommentar. Verwenden Sie die Compiler-Flag `-fPIC`.
- (c) **(2 Punkte)** Schreiben Sie in einem Ordner `src/test/` eine Datei `test.c`, welche den Header des `list`-Moduls inkludiert und eine `main`-Funktion enthält, welche die einzelnen Funktionen des Moduls verwendet. Erzeugen Sie eine Objekt-Datei `test.o` im `build` Ordner. Linken Sie dann die Objektdateien zu einer ausführbaren Datei `test` zusammen, die die Funktionalität des Moduls überprüft. Vermerken Sie erneut die verwendeten Kommandozeilenbefehle in einem Kommentar.
- (d) **(3 Bonuspunkte)** Nehmen Sie die Implementierung der `pgm` Funktionen aus Blatt 10 und fügen Sie diese ihrem Projektorder hinzu. Nehmen Sie hierzu die Funktionsprototypen und schreiben Sie dem Vorgehen in (a) folgend in einen Header `pgm.h` in dem Ordner `include`. Die Implementierung der Funktionen wird im Ordner `src/modules/pgm` platziert. Erzeugen Sie erneut eine Objektdatei. Testen Sie die Funktionalität des Moduls mit einem kleinen Programm `test_pgm.c`.
- (e) **(2 Bonuspunkte)** Erzeugen Sie aus den Objektdateien des `list` und `pgm` Moduls eine shared library `libprog.so` und platzieren Sie sie in dem Ordner `lib`. Fassen Sie die zugehörigen Header in einem File `prog.h` zusammen. Verwenden Sie zur Erzeugung der Bibliothek `gcc -shared -fPIC`. Testen Sie ihre Bibliothek! Vergessen Sie nicht, die Umgebungsvariable `LD_LIBRARY_PATH` zu setzen, damit ihre Bibliothek gefunden wird.