

Übungsblatt 1 – Libraries

Aufgabe 1: Schauen Sie sich noch einmal alle Folien aus der Vorlesung an und stellen Sie sicher, dass Sie sie verstanden haben. Falls in der Vorlesung nicht alle Folien besprochen wurden, dann arbeiten Sie die nicht besprochenen Folien selbstständig durch. Benutzen Sie dabei auch die Literatur. Falls Fragen offen bleiben sollten, besprechen Sie diese während der Übungsstunde mit dem Betreuer.

Aufgabe 2: Besorgen Sie sich ein oder zwei Lehrbücher über Betriebssystemprogrammierung in UNIX, beispielsweise aus der Bibliothek. **Wikipedia ist kein Ersatz für ein gutes Lehrbuch! Google auch nicht!**

- Arbeiten Sie das Einführungskapitel dieses Buchs durch und vergleichen die Inhalte dort mit dem Foliensatz.
- Wiederholen Sie (falls erforderlich) die Abschnitte über den Aufbau von Rechnern, der Funktionsweise von CPUs, bzw. der von-Neumann-Architektur.
- Erstellen Sie Ihre persönliche handschriftliche Zusammenfassung der wichtigsten Inhalte der ersten Lehrveranstaltung.
- Überlegen Sie sich drei Verständnisfragen zur ersten Lehrveranstaltung und halten diese schriftlich fest.

Aufgabe 3: Auf jedem UNIX-System ist das Handbuch online verfügbar. Sie können es mit dem Kommando `man` lesen.

- Aus welchen Kapiteln besteht das Handbuch?
- Welche Bestandteile (Abschnitte) enthält jede Handbuchseite? Schauen Sie sich die Abschnitte einer Handbuchseite an.
- Wie blättert man durch eine längere Handbuchseite (vorwärts, rückwärts, usw.)? Wie kann man bestimmte Texte in einer Handbuchseite suchen?
- Finden Sie heraus, welche Handbuchseiten sich mit dem Thema „grep“ befassen.

Aufgabe 4: In dieser Aufgabe sollen Sie eine eigene **statische Bibliothek** (engl. *static library*) schreiben und erstellen. Anschließend schreiben Sie ein Programm, das die Funktionen der Bibliothek aufruft. Bearbeiten Sie die folgenden Teilaufgaben:

- Schreiben Sie zwei Dateien `file1.c` und `file2.c` in denen jeweils eine Funktion implementiert wird. In `file1.c` soll die Funktion `int addSeven(int)` implementiert werden. In `file2.c` soll die Funktion `int addTwo(int)` implementiert werden. Achten Sie darauf, dass Sie diese Funktionen später aus einem anderen Programm benutzen wollen.
- Automatisieren Sie das Übersetzen (engl. *to compile*) mit Hilfe eines **Makefile** und übersetzen die beiden Dateien aus a). Schalten Sie alle Compilerwarnungen ein und benutzen den Debug-Modus. Falls der Compiler noch Warnungen ausgibt, sind Sie mit der Implementierung nicht fertig!

- c) Überlegen Sie, wie sie aus den beiden Objektdaten eine statische Library bauen können. Hinweise dazu finden Sie unter:
<http://www.faqs.org/docs/Linux-HOWTO/Program-Library-HOWTO.html>
Automatisieren Sie auch diesen Schritt im Makefile.
- d) Schreiben Sie ein Testprogramm `testprog.c` in dem Sie eine Zahl einlesen und dann die beiden Funktionen aus a) mit dieser Zahl aufrufen. Schreiben Sie das Ergebnis der Funktionsaufrufe auf den Bildschirm. Übersetzen und binden Sie das Testprogramm mit Ihrer statischen Library aus c). Automatisieren Sie auch das im Makefile.
- e) Starten Sie das Testprogramm und überprüfen die Ausgabe.

Hinweise:

- Achten Sie auf eine vernünftige Formatierung und Kommentierung aller Dateien, die sie selbst schreiben.
- Schreiben Sie alle Programme so, dass Sie das Programme auch in zwei Jahren noch verstehen und ggf. erweitern können. Verzichten Sie auf vermeintlich elegante und kompakte Formulierungen.
- Programme werden i. d. R. **einmal** geschrieben und **häufig** gelesen. Bevorzugen Sie immer die besser lesbare Variante vor der Variante mit weniger Tipparbeit!
- Achten Sie darauf, dass Ihre Programme Fehler immer melden und nicht einfach weitermachen. Programmieren Sie so, dass Ihr Programm Sie bei einer Fehlersuche unterstützt.

Aufgabe 5:

Erstellen Sie in Ihrem *home directory* ein Unterverzeichnis `lib`

Wiederholen Sie die vorherige Aufgabe mit einer **dynamischen Bibliothek** (engl. *dynamic library*), d.h. erstellen Sie eine dynamische Bibliothek, installieren diese in Ihrem privaten `lib`-Verzeichnis und benutzen Funktionen aus dieser in einem Testprogramm. Hinweise dazu finden Sie unter:

<http://www.faqs.org/docs/Linux-HOWTO/Program-Library-HOWTO.html>

Aufgabe 6:

- a) Prüfen Sie mit dem Kommando `nm` welche Namen bzw. Symbole in Ihren Libraries definiert sind.
- b) Vergleichen Sie die Größe der beiden ausführbaren Testprogramme. Welches ist größer und warum?
- c) Schauen Sie mit dem Programm `ldd` nach, von welchen dynamischen Libraries Ihre beiden Testprogramme abhängen und vergleichen die Ausgaben.