

Übungen Betriebssysteme (BS)

U0 – Erste Schritte in C

Alwin Berger

TU Dortmund - AG Systemsoftware

Organisatorisches: Übungsaufgaben

- **Aufgabenblätter** im Moodle
<https://moodle.tu-dortmund.de/course/view.php?id=46829>
- **Übungsvorbesprechungsfolien** ebenfalls unter dieser URL
- **Lösungsvorschläge** werden in der Übung präsentiert
- Notwendig für **Studienleistung**:
mindestens 50% der Punkte **in beiden Aufgabenblöcken**
(je Blatt 10 Punkte + ggf. Sonderpunkte)
 1. **Block**: A0, A1, A2 → Summe mindestens 15 Punkte
 2. **Block**: A3, A4, A5 → Summe mindestens 15 Punkte

Bestehen der Studienleistung ist Voraussetzung für die Klausurteilnahme!

Organisatorisches: Abgaben

- **Theoriefragen und praktische Programmieraufgaben**
- **Bearbeitung in Dreiergruppen** (Ausnahme: A0)
 - Gruppenmitglieder sollten (müssen aber nicht) in derselben TÜ angemeldet sein.
 - **Hilfestellung:** HelpDesk
- **Abgabe über AsSESS**
- **Abgabetermin abhängig von Woche der Übung:**
 - W1 – Übungstermin in **geraden KW** -> Donnerstag 10:00 in der Woche nach Ausgabe der Aufgabe
Aufgabe A0: Übung U0 in KW16, Abgabe Do. 25.04.2024 10:00
 - W2 – Übungstermin in **ungeraden KW** -> Montag 10:00, direkt bevor das nächste Blatt kommt
Aufgabe A0: Übung U0 in KW17, Abgabe Mo. 29.04.2024 10:00

Organisatorisches: Kommunikation

Wir haben **keine** Telegram-Gruppe!

Die Fachschaft Informatik hat aber ggf. eine in der wir nicht sind.

Für verbindliche Antworten gibt es den Matrix-Helpdesk

<https://matrix.to/#/#bs-helpdesk:fachschaften.org>

UNIX-Benutzerumgebung

- Umgebung, Terminal, Shell
- UNIX-Kommandos
- GNU Compiler Collection (gcc)

Benutzerumgebung

- Diese Punkte machen (u.a.) einen UNIX-Account aus:
 - Benutzername
 - Identifikation (User-ID und Group-IDs)
 - Home-Directory
 - Eingestellte Login-Shell

Terminal

- Terminal
 - “Kommandozeile”
 - Früher: dedizierte Endgeräte zur Kommunikation mit Zentralrechner
 - Heute: *Terminalemulation* (z.B. Konsole, gnome-terminal)



Ein Physisches Terminal

Terminal Demo

Live Demo

Terminal-Sonderzeichen

- Einige Zeichen haben unter UNIX besondere Bedeutung
- Funktionen: u.a.
 - Korrektur von Tippfehlern
 - Steuerung der Bildschirm-Ausgaben
 - Einwirkung auf den Ablauf von Programmen
- Zuordnung Zeichen \leftrightarrow Funktion leider nicht einheitlich
- Kann mit einem Kommando (`stty(1)`) verändert werden

Terminal-Sonderzeichen

- Übersicht:
 - <Backspace> letztes Zeichen löschen
 - <Ctrl>-U alle Zeichen der Zeile löschen
 - <Ctrl>-C Interrupt – Programm abbrechen
 - <Ctrl>-Z Stop – Programm wird angehalten
 - <Ctrl>-D End Of File
 - <Ctrl>-S / <Ctrl>-Q Bildschirmausgabe anhalten/fortsetzen
 - Auf deutschen Tastaturen: <Strg> statt <Ctrl>

UNIX-Kommandointerpreter: Shell

- Meist stehen verschiedene Shells zur Verfügung: sh, bash, zsh, fish, pwsh ...
- Auf GNU-Systemen gebräuchlich: **bash**
- Beim Öffnen eines Terminals startet die **ausgewählte Login-Shell**
- Wechsel der Login-Shell: **chsh(1)**

Aufbau eines UNIX-Kommandos

```
ls -l --all mein_ordner
```

Aufrufe von UNIX-Kommandos bestehen aus ...

1. **Kommandoname** (der Name einer Datei, in der ein ausführbares Programm oder eine Kommandoprozedur für die Shell abgelegt ist)
 - nach dem Kommando wird automatisch in allen Verzeichnissen gesucht, die in der Umgebungs-Variable PATH gelistet sind
 - Die Pfade in PATH werden durch einen Doppelpunkt : getrennt
 - daher kann man normalerweise „ls“ schreiben statt „/bin/lS“

Aufbau eines UNIX-Kommandos

```
ls -l --all mein_ordner
```

2. einer Reihe von **Optionen** und **Argumenten**

- Abtrennung Kommandoname/Optionen/Argumente durch Leerzeichen oder Tabulatoren
- Optionen sind meist einzelne Buchstaben mit einem vorangestellten „-“ (Minuszeichen) (z.B. „ls -l“)
- Oft gibt es auch „lange“ Optionen, bestehend aus Worte mit zwei vorangestellten „-“
- Nach vielen Optionen muss ein Argument folgen
- Argumente sind häufig Dateinamen, die von einem Kommando verarbeitet werden

UNIX-Kommandos

- man-Pages
- Dateisystem
- Benutzer
- Prozesse
- Diverse Werkzeuge
- Texteditoren

man-Pages

- Aufgeteilt in verschiedene Sections (mehr Infos: `man man`)
 - (1) Kommandos
 - (2) Systemaufrufe
 - (3) C-Bibliotheksfunktionen
 - (5) Dateiformate (spezielle Datenstrukturen etc.)
 - (7) Verschiedenes (z.B. IP, GPL, Zeichensätze, ...)
- man-Pages werden normalerweise mit der Section zitiert: `printf(3)`
sonst teilweise mehrdeutig (`printf(1)` vs. `printf(3)`)!

man-Pages

- Aufruf unter Linux:

```
man [section] Begriff
```

z.B.

```
man 3 printf
```

- Suche nach Sections:
man -f Begriff
- Suche nach Stichwort:
man -k Stichwort

Dateisystem - Navigation

cd Verzeichnis wechseln; im Terminal integriert

„.“ aktuelles Verzeichnis

„..“ übergeordnetes Verzeichnis

„-“ Verzeichnis, in dem man vor der letzten Navigation war

Ohne Argumente: navigiert cd in das Heimverzeichnis des Nutzers

ls Verzeichnis auflisten; wichtige Optionen:

„-l“ langes Ausgabeformat

„-a“ listet auch mit „.“ beginnende Dateien und Verzeichnisse

pwd Aktuelles Verzeichnis ausgeben

Üblicherweise wird „~“ zum Heimverzeichnis expandiert: **cd ~/Downloads** navigiert

also z.B. zu **/home/nutzer/Downloads**

(**cd ~nutzer** wechselt in das Verzeichnis des Benutzers “nutzer”)

Dateisystem – Manipulation

chmod	Zugriffsrechte einer Datei ändern
cp	Datei(en) kopieren
mv	Datei(en) verlagern (oder umbenennen)
ln	Datei linken (Verweis auf dieselbe Datei)
ln -s	Symbolischen Link erzeugen
rm	Datei(en) löschen
mkdir	Verzeichnis erzeugen
rmdir	Verzeichnis löschen (muss leer sein!)

Benutzer

`id, groups`

eigene Benutzer-ID, Gruppenzugehörigkeit

`who`

am Rechner angemeldete Benutzer

Zum Nachschlagen:

`getuid(2)`

gibt die Nutzer-ID zurück (C-Programmschnittstelle)

`getgid(2)`

gibt die Hauptgruppen-ID zurück (C-Schnittstelle)

... und weitere! Alle Funktionen der C-Standardbibliothek besitzen einen Handbucheintrag.

Prozesse

ps Prozessliste ausgeben

-u x Prozesse des Benutzers x

-ef alle Prozesse (-e), ausführlich (-f)

top -o %CPU Prozessliste, sortiert nach aktueller Aktivität

kill <pid> Prozess „terminieren“ (Prozess kann sich geordnet beenden, oder ignorieren)

kill -9 <pid> Prozess „töten“ (Beenden ohne Aufräumen wird erzwungen)

Diverse Werkzeuge

<code>cat</code>	Dateien hintereinander ausgeben
<code>less</code>	Dateien bildschirmweise ausgeben
<code>head, tail</code>	Anfang/Ende einer Datei ausgeben (10 Zeilen)
<code>cal</code>	Kalender im Terminal anzeigen
<code>wc</code>	Zeilen, Wörter und Zeichen zählen
<code>grep, fgrep, egrep</code>	Nach bestimmten Mustern o. Wörtern suchen
<code>find</code>	Dateibaum durchlaufen

Diverse Werkzeuge

<code>sed</code>	Stream-Editor, z.B. zum Suchen/Ersetzen
<code>tr</code>	Zeichen aufeinander abbilden oder löschen
<code>date</code>	Datum auf diverse Art und Weise ausgeben
<code>cut</code>	Einzelne Felder aus Zeilen ausschneiden
<code>sort</code>	Sortieren

Texteditoren

- Geschmackssache – aber einen solltet ihr beherrschen!
- Klassiker mit Nerdfaktor: **vim/neovim, emacs**
- Minimalist: **nano**
- Weitere mit GUI:
Kate, Geany, Gedit, VSCode/VSCodium, Eclipse, QtCreator, ...
- Zum Programmieren **nicht** geeignet:
Office-Programme (**MS Word, LibreOffice Writer, ...**)

GNU Compiler Collection

- Ursprünglich: GNU C Compiler
- Mittlerweile: Sammlung von verschiedenen Compilern (u.a. C, C++, *Objective-C*, *Fortran 95*, *Rust*, ...)
- Viele verschiedene Zielplattformen (x86, AMD64, ARM, ...)

GNU Compiler Collection

- C-Compiler: **gcc**
- Compilieren und Linken eines C-Programms:
 - Wall** alle Warnungen ausgeben
 - o <Ziel>** Name für ausführbare Datei
- Weitere nützliche Parameter (siehe man-Page):
 - std=c11** -**Werror** -**ansi** -**Wpedantic** -**D_POSIX_SOURCE**
- **Warnungen** sind grundsätzlich ernstzunehmen und zu beseitigen, daher möglichst immer mit -**Werror** übersetzen.

Übung macht den Meister!

- Mit UNIX-Umgebung experimentieren
 - in der Rechnerübung,
 - in der Linux-VM von der BS-Website,
 - auf den Linux-Servern der IRB,
 - im Windows Subsystem for Linux (WSL) oder
 - in einer eigenen Linux-Installation
- Anleitung für die Entwicklungsumgebung auf der Veranstaltungswebseite

Grundlagen der C-Programmierung

Wie sind eure Erfahrungen mit C(++)?

Braucht ihr in dieser Gruppe eine Einführung?

Falls ja → Foliensatz C-Einführung (bis Folie 23)

Aufgabe 0: Erste Schritte in C

- Programm:

```
#include <stdio.h>
int is_even(int x) {
    if (x % 2 == 0) return 1;
    else return 0;
}
int main(void) {
    printf("%d\n", is_evne(4));
    return 0;
}
```

- Compilieren/Linken:

```
└─$ gcc -Wall -o test test.c
test.c: In function 'main':
test.c:8:24: warning: implicit declaration of function 'is_evne'; did you mean 'is_even'?
[-Wimplicit-function-declaration]
   8 |         printf("%d\n", is_evne(4));
     |                        ^
     |                        is_even
/usr/bin/ld: /tmp/ccuxtqv3.o: in function `main':
test.c:(.text+0x36): undefined reference to `is_evne'
collect2: error: ld returned 1 exit status
```

Aufgabe 0: Erste Schritte in C

- Programm:

```
#include <stdio.h>
int is_even(int x) {
    if (x % 2 == 0) return 1;
    else return 0;
}
int main(void) {
    printf("%d\n", is_even(4));
    return 0;
}
```

- Compilieren/Linken:

```
$ gcc -Wall -o test test.c
```

- Ausführen:

```
$ ./test
1
```

Compiler Demo

Live Demo

Quiz-Zeit!

Beantwortet einige Fragen zu den Inhalten. Manche sind single-choice, manche multiple-choice.

Zukünftig wollen wir Süßigkeiten als Preise für die meisten ersten richtigen Antworten ausloben.

Leider noch nicht heute.

Quiz-Zeit!

Welche Aussage über Terminal und Shell stimmen?

A: Terminal und Shell sind gleichbedeutend.

B: Eine Shell ist traditionell ein Teil der Hardware, Terminal eine Software ist.

C: Ein Terminal ist ein textbasiertes Interface, die Shell interpretiert Eingaben.

Quiz-Zeit!

Welchen der Folgenden Befehle sollte man niemals Aufrufen?
Single-choice mit Begründung!

A: `rm -rf /`

B: `cat /etc/passwd | grep bash`

C: `cal -m 1`

Quiz-Zeit!

Wie werden Kommandos auf dem System gefunden? Single-choice.

A: Es wird immer nur im aktuellen Verzeichnis gesucht.

B: Es werden alle Ordner durchsucht, die `bin` heißen.

C: Es werden alle Ordner in der Umgebungsvariablen `PATH` durchsucht.

D: Es werden alle Ordner in der Umgebungsvariablen `BIN` durchsucht.

Quiz-Zeit!

Mit welchem Sonderzeichen kann man ein laufendes Programm direkt abbrechen? Single-choice.

A: <Ctrl>-<Backspace>

B: <Ctrl>-S

C: <Ctrl>-C

D: <Ctrl>-V

Gruppensuche

Falls ihr noch keine Dreiergruppe habt ist jetzt ein guter Moment, um eine zu suchen.

► **Warum?**

