Freie Universität Berlin

Übung 1

Betriebs- und Kommunikationssysteme — Ausgabe 20.04.2018

Prof. Dr.-Ing. Jochen Schiller, Sommersemester 2018 — Abgabe 04.05.2018, 10:15

Allgemeine Hinweise (gelten für alle Übungszettel):

Bearbeitung der Aufgaben:

- Beantworten Sie alle Aufgaben so verständlich und leserlich wie möglich mit Ihren eigenen Worten.
- Abgaben sind in Englisch und Deutsch möglich.
- Quellcode ist ausreichend zu kommentieren.
- Programmieraufgaben müssen im Linuxpool mit *gcc* kompilierbar sein. Nutzen Sie dazu die Flags *-std=c11 -Wall -Wextra -pedantic*. Es sollten keine Warnungen auftauchen.
- Legen Sie Ihrer Abgabe Testläufe als Fehlerdokumentation bei, wenn Ihr Programm nicht oder nur teilweise funktioniert.
- Bei Quellen abseits der Vorlesungsfolien sind diese anzugeben.
- Die Abgabe eines Plagiats wird mit 0 Punkten bewertet und führt bei Wiederholung zum Ausschluss und Nichtbestehen der Veranstaltung!

Abgabe des Zettels:

- Achten Sie bitte darauf, die Namen aller Gruppenmitglieder auf Ihrer Lösung zu vermerken.
- Bitte geben Sie Ihre Lösung auf Papier im Fach Ihres Tutors oder als PDF im KVV ab.
- Quellcode reichen Sie bitte unkomprimiert im KVV ein.
- Abgaben sind bis zum Beginn der Vorlesung jeweils Freitags (10:15 Uhr) möglich.
- Zu spät eingereichte Lösungen können leider nicht bewertet werden!

Bewertung eines Zettels:

- Es gibt immer zwei (bewertete) Aufgaben.
- Jede der beiden Aufgaben wird mit 3 Punkten bewertet.
- Zum Bestehen des Zettels muss in jeder Aufgabe mindestens 1 Punkt erreicht werden, sowie
- mindestens 3 Punkte auf dem gesamten Zettel.
- Die Punkte sind wie folgt pro Aufgabe definiert:

  3 Punkte:

  - Alles perfekt.
  - Klausur kann kommen.

  2 Punkte:

  - Es funktioniert / ist im Wesentlichen korrekt.
  - Kleinere Mängel sind mit Kommentaren versehen.

  1 Punkt:

  - Es funktioniert nicht, aber richtige Idee mit Fehlerbeschreibung (!) vorhanden.
  - Bzw. grobe Fehler enthalten oder unzureichend beschrieben.
  - Dennoch erkennbarer Aufwand.

  0 Punkte:

  - Aufgabe nicht bearbeitet.
  - Oder kein Ansatz für unabhängige Teilaufgaben vorhanden.
  - Oder sinnvoller Arbeitsaufwand nicht erkennbar.
  - Damit ist der Zettel leider nicht bestanden.

Aufgabe 1: Einführung und Motivation

Begriffe

Beschreiben Sie jeden der folgenden Begriffe **in eigenen Worten** mit maximal einem Satz.

1. Instruction Set
2. Process
3. File
4. System Call
5. Multitasking
6. Protection Rings

Schichtenmodell

Viele Zusammenhänge in Betriebssystemen lassen sich durch Schichtenmodelle darstellen. Die vertikalen Schichten sind hierarchisch angeordnet, d.h. Objekte auf einer höheren Schicht greifen auf die Objekte der darunterliegenden Schicht zu, welche wiederum auf die nächsttiefere Schicht zugreifen usw.

a) In einem Computersystem kann man eine grobe Einteilung in folgende Schichten vornehmen: `Betriebssystem`, `Anwendungsprogramme`, `Hardware` und `Nutzer`. Bringen Sie die Schichten in die richtige Reihenfolge und ordnen Sie folgende Begriffe den einzelnen Schichten zu:

`CPU, Datenbanksystem, Studentin/Student, Scheduler, Transistor, Festplatte, Assembler, Texteditor, Professorin, Speichermanagement, Dateisystem, Graphikprogramm, Compiler, Debugger, Textverarbeitungsprogramm, BIOS, Cache, Gerätemanagement`

b) Eine andere Unterteilung ist die Gliederung in `Hardware` und `Software`. Erklären Sie in einem Satz, welche Aufgabe aus dieser Sicht einem Betriebssystem zukommt.

Aufgabe 2: Hallo (C) Welt

Arbeitsumgebung

In dieser Veranstaltung wird der Umgang mit der Programmiersprache C gelehrt. Da C eine plattformabhängige Sprache ist, sind einige Richtlinien nötig, um die korrekte Funktionalität ihrer Programme auf den Referenzsystemen sicher zu stellen. Es empfiehlt sich ein solches produzierendes System minimal als Virtuelle Maschine einzurichten, oder an den Poolrechnern zu arbeiten. Anforderungen an ihr produzierendes System sind:

1. Linux 64 Bit (z.B. Linux Mint, Ubuntu, ...)
2. GNU C Compiler (GCC) mit Standardbibliotheken
3. Shell (z.B. Zsh, Bash, ...)
4. Editor (z.B. Atom, Emacs, Vim, ...)

Ihr Programm muss mit folgenden Compiler-Flags ohne Warnungen und / oder Fehler kompilieren:

```
$ gcc -std=c11 -o <program.out> -Wall -Wextra -pedantic <program.c>
```

Dateien mit C einlesen

Schreiben Sie ein C-Programm, das eine Datei <test.dat> einliest. Die Datei sei wie folgt aufgebaut:

1. In der ersten Zeile steht eine natürliche Zahl.
2. Ab der zweiten Zeile stehen pro Zeile jeweils zwei natürliche Zahlen, welche durch ein '*' getrennt sind.

Beispiel:
200
1*100
2*50

Ihr Programm soll überprüfen, ob die Summe der Produkte ab der zweiten Zeile gleich der Zahl in der ersten Zeile ist und eine entsprechende Meldung zurückliefern.
Hinweis: Zur Lösung der Aufgabe sollten Sie insbesondere die Standard-Bibliothek <stdio.h> verwenden.