Betriebs- und Kommunikationssysteme

WS 14/15

Prof. Dr-Ing. Volker Roth

Freie Universität Berlin

7. Übung

Ausgabe	Abgabe
28.11.14	12.12.14

Bitte bei der Abgabe Name der Mitglieder einer Gruppe, Nummer der Übung/Teilaufgabe und Datum auf den Lösungsblättern nicht vergessen! Darauf achten, dass die Lösungen beim richtigen Tutor abgegeben werden. Achten Sie bei Programmieraufgaben außerdem darauf, dass diese im Linuxpool kompilierbar sind. Nutzen Sie dazu die Flags `-std=c99`, `-Wall` und `-pedantic`. Es sollten keine Warnungen auftauchen.
Zu spät abgegebene Lösungen werden nicht berücksichtigt!

Aufgabe 1: Vorlesung (0 Punkte)

Kommen Sie zur Vorlesung am 5.12. in der **Hörsaal in der Informatik (Takustraße 9)**. Die Vorlesung findet einmalig **nicht** in der Silberlaube statt.

Aufgabe 2: Dateisysteme (2 Punkte)

(a) In einem hierarchischen Dateisystem werde freier Festplattenplatz in einer Liste verwaltet.

(1) Der Zeiger auf die Freispeicherliste sei verloren gegangen. Kann das System die Freispeicherliste wiederherstellen und wenn ja, wie?

(2) Schlagen Sie ein Verfahren vor, bei dem der Zeiger nicht aufgrund eines einzigen Speicherfehlers verloren gehen kann.

(b) Bei dieser Aufgaben geht es um das FAT16 Dateisystem.

(1) Beschreiben Sie kurz das FAT16 Dateisystem.

(2) Erklären Sie genau, warum bei einem FAT16 Dateisystem im Wurzelverzeichnis nur eine begrenzte Anzahl Dateien stehen darf, in einem Unterverzeichnis aber fast beliebig viele. Wie viele Dateien passen maximal in ein Unterverzeichnis? Argumentieren sie anhand der FAT16 Spezifikation und nicht durch Suchen der Lösungen im Internet!

Aufgabe 3: Programmieren in C (3 Punkte)

Lösen Sie folgende Aufgaben mit den Mitteln, die Ihnen die stdio.h zur Verfügung stellt. Geben Sie hierbei die Ausgabe des Programms auf stdout aus und die Fehlermeldungen auf stderr.

(a) Implementieren Sie ein Programm „cat“, das einen Dateinamen als Parameter erhält und den Inhalt dieser Datei ausgibt.

(b) Implementieren Sie ein Programm „wc“, das einen Dateinamen als Parameter erhält und die Anzahl der Zeilen, Worten und Bytes dieser Datei ausgibt.

(c) Implementieren Sie ein Programm „grep“, das einen String und einen Dateinamen als Parameter erhält und alle Zeilen aus der Datei ausgibt, die diesen String enthalten.

Passen Sie nun Ihre Implementierung so an, dass für den Fall, dass kein Dateiname angegeben wurde, die Eingabe von stdin eingelesen wird. Testen Sie Ihre Programme durch geeignete Aufrufe. So soll beispielsweise der Befehl

```
1       \$ ./cat file\_name | ./grep test | ./wc
```

in einer UNIX-Shell die Anzahl von Zeilen, Worten und Bytes ausgeben, die in den Zeilen der Datei file_name enthalten sind, die auch den String „test“ enthalten.