



---

Übungen zur Vorlesung  
**Werkzeuge der Informatik**  
– 2. Übungsblatt –

---

Datum: November 7, 2008

Zu diesen Aufgaben finden Sie am Ende einige Hinweise und Hilfestellungen.

**Aufgabe 1 (7 Punkte, Suchpfade)**

Punkte:

- 1) Wechseln Sie in Ihr Home (d.h., dass danach Ihr Home das *current working directory* ist).
- 2) Stellen Sie sicher, dass Sie sich in einer `tcsh` befinden.
- 3) Starten Sie eine Bash, und beenden Sie diese danach wieder mit dem Befehl `exit`.
- 4) Laden Sie den File `bash` von der Homepage zur Vorlesung, und speichern Sie diesen in Ihr Home.
- 5) Machen Sie diesen File executable (ausführbar) für sich (Tip: `chmod`).
- 6) Stellen Sie sicher, dass der Punkt (also das CWD) der erste Pfad in Ihrer Environment-Variable `PATH` ist.
- 7) Geben Sie nun wieder das Kommando `bash` ein — was passiert, und wieso?
- 8) Setzen Sie nun die Environment-Variable `PATH` so, dass der Punkt als Pfad nicht mehr enthalten ist, alle anderen Pfade aber schon noch (hier hilft nur Copy-and-Paste).
- 9) Geben Sie nun wieder das Kommando `bash` ein — was passiert, und wieso?
- 10) Löschen Sie nun den File `bash` wieder aus Ihrem Home!

\_\_\_\_\_ von 34

Gruppe / Tutor:

Name(n) & Matr.-Nr.:

**Aufgabe 2 (4 Punkte, Einfacher Chat)**

Starten Sie in einem Terminal-Fenster den Befehl `cat -u > ~/chat.txt`. Starten Sie in einem zweiten Fenster den Befehl `tail -f ~/chat.txt`.

Tippen Sie nun im ersten Fenster einige Zeilen beliebigen Text ein. Erklären Sie die Ausgabe im zweiten Fenster.

(Wer möchte, kann dies einmal mit einem Kommilitonen ausprobieren, indem sich einer der beiden auf dem Rechner das anderen einloggt per `ssh`; dann muss der "Chat"-Filename natürlich etwas modifiziert werden.)

**Aufgabe 3 (2 Punkte, Prozesse)**

Führen Sie in einer Shell `ps -aeH` aus.

- a) Interpretieren Sie die Ausgabe.
- b) Identifizieren Sie genau den Prozess, der die Shell darstellt, aus der heraus Sie das Kommando abgeschickt haben.

**Aufgabe 4 (4 Punkte, Grep)**

- a) Lassen Sie sich mit `more` oder `cat` den Inhalt der Datei `/etc/services` ausgeben.

**Abgabe:**

12./13. 11. 2008  
zu **Beginn** der Übung



- b) Filtern Sie mit `grep` den Inhalt der Datei, so dass nur die Zeilen, die den String `HTTP` enthalten, ausgegeben werden.
- c) Filtern Sie die Ausgabe von Teil b, so dass nur noch die Zeilen, welche zusätzlich auch `8080` enthalten, ausgegeben werden.

## Aufgabe 5 (6 Punkte, Spaß mit Pipelines)

Die Internet-Movie-Database [www.imdb.com](http://www.imdb.com) enthält eine Liste der besten 250 Filme aller Zeiten (worüber man streiten kann). Wir haben diese Liste von dort abgespeichert und in eine ASCII-Datei mit folgendem Format konvertiert:

```
Ranking:Movie-Title:Year:Rating:Votes
```

Laden Sie die Datei `top250.txt` von der Vorlesungs-Homepage in Ihr Home-Verzeichnis.

- a) Geben Sie den 100-ten Film aus der Liste aus. (Tip: lesen Sie die Man-Page von `head` und `tail`).
- b) Geben Sie die Titel der besten zehn Filme in alphabetischer Reihenfolge aus.
- c) Geben Sie die Jahre aus, in denen es nur ein Film in die Top 250 geschafft hat. (Tip: `uniq`)
- d) Nennen Sie die Anzahl der Jahre, in denen es ein Film in die Top 250 geschafft hat. (Tip: `wc -l`)

## Aufgabe 6 (11 Punkte, Noch mehr Spaß mit Pipelines)

- a) Wieviele Prozesse laufen gerade auf dem System? (Hinweis: `ps -e, wc`)
- b) Zeigen Sie alle User-Namen an, die gerade einen Prozeß auf dieser Maschine laufen lassen. (Hinweis: `ps -ef, cut`).<sup>1</sup>
- c) Wie kann man die erste Zeile ("UID ...") der Ausgabe der vorherigen Kommando-Pipeline verschwinden lassen? (Tip: `grep -v`)
- d) Entfernen Sie Duplikate aus der Ausgabe der vorherigen Kommando-Pipeline. (Hinweis: `sort, uniq`)
- e) Generieren Sie aus der vorherigen Ausgabe eine "Top-Ten-Liste" (d.h.: welche Benutzer lassen die meisten Prozesse laufen?) Hinweis: Sie benötigen eine Option von `uniq` sowie zweimal das Kommando `sort`.
- f) Geben Sie mit dem Kommando `du` die sog. *disk usage* (in kB) aller Files und Verzeichnisse in `/usr` aus. Dabei soll für ein Unterverzeichnis von `/usr` dessen gesamte Usage inklusive aller seiner Unterverzeichnisse aufsummiert ausgegeben werden. Sortieren Sie die Verzeichnisse nach Größe.

---

<sup>1</sup> Diese und die folgenden Teilaufgaben lassen sich wesentlich besser testen, wenn Sie sich per `ssh` auf `login.in.tu-clausthal.de` einloggen, weil dort oft viele User gleichzeitig eingeloggt sind.



## Tips und Hinweise

**grep** : Der Befehl **grep** dient dem Durchsuchen einer Datei nach einem Ausdruck. Die einfachste Form, **grep STRING FILE**, gibt alle Zeilen der Datei **FILE** aus, welche die Zeichenkette **STRING** enthalten. Wird keine Datei angegeben, liest **grep** von **STDIN**.

Lesen Sie sich die Man-Page von **grep** durch, insbesondere die Optionen **-e**, **-i**, **-v**. (Wenn dort von einem *pattern* die Rede ist, dann können Sie für unsere Zwecke einfach einen String annehmen.)

Falls Sie zu Hause kein Linux haben, können Sie die Man-Page auch hier online lesen. (Wie übrigens alle Man-Pages!)

**cut** : **cut --fields=N --delimiter=T** schneidet aus allen Zeilen einer übergebenen Datei bzw. **STDIN**s bestimmte Felder aus. Als Trennzeichen zwischen den Feldern einer Zeile fungiert das Zeichen **T**. Die Option **--fields=N** gibt das/die gewünschte/n Feld/er an (hier Nummer **N**).

**sort** : Sortiert einen File (bzw. **stdin**) zeilenweise alphabetisch. **sort FILE** gibt also die Datei **FILE** zeilenweise sortiert aus. Mit der Option **-n** sortiert **sort** numerisch (dabei nimmt **sort** an, dass die Zahl am Anfang der Zeile beginnt).

**uniq** : Zwei aufeinanderfolgende Zeilen werden mit einander verglichen; sind beide identisch, wird der Doppelgänger entfernt. (Tip: vorher sortieren!)

**du** : Dient zur Ausgabe der sog. Disk-Usage einzelner Files und Verzeichnisse. Tip: starten Sie **du /usr/\***, analysieren Sie die Ausgabe, und suchen Sie sich dann die sinnvolle Option aus der Man-Page zum Lösen der Aufgabe.