

Unter der nachstehenden URL können Sie ein Archiv `dateien08.tar.gz` mit Dateien herunterladen, die Sie für die Bearbeitung der Übungsaufgaben benötigen:  
<http://www.math.uni-bielefeld.de/~frettloe/teach/unix/dateien08.tar.gz>

## Aufgabe 1

In dem o.g. Archiv `dateien08.tar.gz` befinden sich unter anderem die Dateien `eins.dat`, `zwei.dat` und `drei.dat`.

Schreiben Sie ein Shellskript `ausgabe.sh`, das den Inhalt dieser Dateien wie folgt ausgibt:

```
./ausgabe.sh
Inhalt von Datei drei.dat:
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 dritte Textdatei
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

Inhalt von Datei eins.dat:
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 erste Textdatei
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Inhalt von Datei zwei.dat:
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 zweite Textdatei
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Zur Lösung dieser Aufgabe benötigen Sie eine `for`-Schleife sowie die Befehle `echo` und `cat`.

## Aufgabe 2

- a) Schreiben Sie mit Hilfe einer `for`-Schleife ein Skript, das in jedem Wort seiner Eingabe die Anzahl der Zeichen zählt und sie wie folgt ausgibt:

```
$ ./skript.sh Kommandozeilenprogrammierung macht Hunger
Kommandozeilenprogrammierung : 28 Zeichen
macht : 5 Zeichen
Hunger : 6 Zeichen
```

Lösungshinweise: Zum Zählen von Zeichen siehe Folie 29 aus Vorlesung 5 zu `wc -m`. Ein Skript kann alle übergebenen Parameter mit Hilfe von `$*` an eine `for`-Schleife weiterreichen.

- b) Erweitern Sie Ihr Skript so, daß es am Ende die Gesamtsumme der Zeichen aller Worte ausgibt:

```
$ ./skript.sh Kommandozeilenprogrammierung macht Hunger
Kommandozeilenprogrammierung : 28 Zeichen
macht : 5 Zeichen
Hunger : 6 Zeichen
39 Zeichen insgesamt
```

- c) Statten Sie Ihr Skript mit einer Abfrage aus, um die folgende Meldung für den Fall auszugeben, dass gar kein Wort angegeben wurde:

```
$ ./skript.sh
Bitte mindestens ein Wort angeben!
```

## Aufgabe 3

Die Tribonacci-Folge  $t_0, t_1, t_2, t_3, \dots$  ist definiert durch folgendes Bildungsgesetz:

$$t_0 = 0, t_1 = t_2 = 1.$$

$$t_n = t_{n-1} + t_{n-2} + t_{n-3} \text{ für } n > 2.$$

Schreiben Sie ein Shellskript, das den Anfang der Tribonacci-Folge bis zu einem anzugebendem Wert wie folgt ausgibt:

```
$ ./trib.sh 5
0,1,1,2,4,7,...
```

```
$ ./trib.sh 15
0,1,1,2,4,7,13,24,44,81,149,274,504,927,1705,3136,...
```

Zur Lösung benötigen Sie eine `for`-Schleife, eine `if`-Abfrage, `seq`, sowie das Rechnen mit Variablen und das Zuweisen von Werten zu Variablen.

## Aufgabe 4

Gegeben sei die CSV-Datei `umsatz.txt` aus dem heruntergeladenen Archiv. Schreiben Sie ein Shellskript, um daraus die Wertpaare aus Datum und Gesamtwert für ein anzugebendes Produkt wie nachfolgend gezeigt auszugeben.

Eingabedatei:

```
Produkt;Min;Max;Gesamt;Datum
Basic;30790;53077;111787;01.01.2021
Premium;21541;41299;81205;01.01.2021
Basic;30635;61166;98351;02.01.2021
Premium;1998;24428;58514;02.01.2021
...
```

Gewünschte Ausgabe:

```
$ ./extract.sh umsatz.txt Premium
01.01.2021 81205
02.01.2021 58514
03.01.2021 28754
...
```

## Aufgabe 5

Die Datei `verbrauch.txt` soll in ein CSV-Format (character separated values) umgewandelt werden, um sie mit einem anderen Werkzeug weiter verarbeiten zu können. Schreiben Sie ein Shellskript, dass

- im Datumsfeld den Monat Januar durch `1.` ersetzt,
- alle Leerzeichen komprimiert und durch einen Doppelpunkt ersetzt.

Erweitern Sie dieses Shellskript, so dass es auch die Spalten umsortiert: Spalte 2 soll vor Spalte 3 rücken. (Eventuell müssen Sie dazu das Ergebnis des ersten in einer Datei zwischenspeichern).

Inhalt der Originaldatei:

```
Stadt      Tief    Hoch    Datum
Bielefeld  20332  25235  1. Jan 2021
Guetersloh  4447   23371  1. Jan 2021
Paderborn  22491  31014  1. Jan 2021
...
```

Beispielausgabe:

```
$ ./mod.sh verbrauch.txt
Stadt:Datum:Tief:Hoch
Bielefeld:1.1.2021:20332:25235
Guetersloh:1.1.2021:4447:23371
Paderborn:1.1.2021:22491:31014
Bielefeld:2.1.2021:2678:8707
...
```

## Aufgabe 6

Schreiben Sie ein Shellskript mit dem Namen `hoch` und zwei ganzzahligen Parametern  $b$  und  $e$ , das die Potenz  $b^e$  ausrechnet. Verwenden Sie zur Berechnung eine `for`-Schleife und `seq`; die Benutzung des Kommandozeilenrechners `bc` ist zur Lösung nicht erlaubt.

Beispiele:

```
$ ./hoch 2 8
256

$ ./hoch 3 3
27

$ ./hoch 5 0
1
```

---

### Downloads (Folien, Übungsblätter)

<https://www.math.uni-bielefeld.de/~frettloe/teach/unix22.html>

### Hinweise zu den Übungen

Die Übungen dienen dem Erlernen von Linux. Es gibt keine Abgabepflicht, es gibt überhaupt keine Abgaben. In der **A&D-Klausur** am Ende des Semesters werden allerdings Linux-Fragen vorkommen. Außerdem brauchen Sie im Verlaufe des Studiums solide Linuxkenntnisse. Daher ist es sinnvoll, dass Sie die Übungen entweder selbständig lösen, oder aber eines der Tutorien besuchen und die Übungen dort bearbeiten. Darüber hinaus können Sie in den Tutorien den Tutoren Fragen stellen zu Übungen und Vorlesung.