

# Assignment 1

## Abgabe bis 5. Woche

Je nach Halbgruppe 27./28. März,  
jeweils 10 Minuten vor der Laborstunden

### Achtung!

Jeder Studierende muss in der 5. Woche eine gelöste Aufgabe abgeben.

### Anforderungen:

- Jede Aufgabe wird in einer eigenständigen Datei (.sh) gelöst .
- Jede Halbgruppe lädt die Lösungen in das ihrer Halbgruppe entsprechende Assignment.
- Die Dateien werden nach folgendem Format benannt und hochgeladen:

**Name\_Vorname\_Gruppe\_X.sh**

wobei **X** die Nummer der Aufgabe ist, die Sie lösen müssen.

- Wer andere als die zugewiesenen Aufgaben löst, wird mit Note 0 bewertet.

**Zuordnung der Aufgaben:** Die zur Lösung vorgeschlagenen Aufgaben werden in der Laborstunde der 3. Woche verteilt.

### **Aufgabe 1:**

Schreiben Sie ein Shell-Skript, das alle Dateien aus dem aktuellen Ordner und aus seinem Unterordner bestimmt, die Schreibrechte für "others" haben. Jeder Dateiname wird in einer Zeile geschrieben. Das Ergebnis wird in der Standardausgabe angezeigt.

**Hinweis:** Aus der Sicht der Zugriff auf einer Datei, werden die Benutzer in drei Kategorien unterteilt: Eigentümer (derjenige, der die Datei erstellt), Gruppe und andere (others). Jeder Benutzertyp kann drei Rechte für die jeweilige Datei haben: r (read) - Lesen, w (write) - Schreiben und x (execute) - Ausführen.

### **Aufgabe 2:**

Schreiben Sie ein Shell-Skript, das einen Ordernamen als Parameter erhält und die Gesamtzahl der Zeilen in allen Textdateien aus diesem Ordner und aus allen seinen Unterverzeichnissen bestimmt (die Summe aller Dateien, nicht für jede einzelne Datei). Es wird angenommen, dass jeder Ordner nur Textdateien enthält. Die erforderliche Nummer wird in der Standardausgabe ohne zusätzlichen Text angezeigt.

### **Aufgabe 3:**

Schreiben Sie ein Shell-Skript, das den Inhalt eines Ordners auflistet. Der Ordner wird als Parameter angegeben. Es werden drei Anzeigen erstellt: eine nach Dateiname sortiert, eine nach dem Zeitpunkt des letzten Zugriffs und die dritte nach ihrer Bytegröße. Die Dateien werden einzeln in der Zeile angezeigt. Zwischen den Sortierungen soll eine leere Zeile bleiben. Das Ergebnis wird in der Standardausgabe angezeigt.

#### **Aufgabe 4:**

Schreiben Sie ein Shell-Skript, das alle ".txt" -Dateien aus einem Ordner und aus allen seinen Unterordner umbenennt und ihnen die Erweiterung ".ascii" gibt. Der Ordner wird als Parameter in die Befehlszeile eingegeben. (Die Erweiterung .txt wird durch die Erweiterung .ascii ersetzt)

#### **Aufgabe 5:**

Schreiben Sie ein Shell-Skript, das überprüft, ob eine Zahl eine Primzahl ist oder nicht. Die Zahl wird von der Standardeingabe gelesen und das Ergebnis wird bei der Standardausgabe im folgenden Format angezeigt:

prime - wenn die Zahl eine Primzahl ist

not prime - wenn die Zahl keine Primzahl ist

#### **Aufgabe 6:**

Schreiben Sie ein Shell-Skript, das überprüft, ob drei Punkte in der Ebene kollinear sind. Wenn die Punkte nicht kollinear sind, zeigen Sie den Flächeninhalt dem von ihnen beschriebenen Dreieck an. Die Koordinaten der Punkte werden in der Befehlszeile im folgenden Format gelesen: x1 y1 x2 y2 x3 y3. Die Anzeige erfolgt in der Standardausgabe und der Text "kollinear" wird angezeigt, wenn die Punkte kollinear sind, oder direkt die Fläche des Dreiecks, wenn die Punkte nicht kollinear sind. Die Koordinaten der Punkte werden als positive, ganze Zahlen betrachtet.

**Hinweis:** Bei diesem Problem beziehen wir uns auf den ganzen Teil des Flächeninhalts.

### Aufgabe 7:

Schreiben Sie ein Shell-Skript, das drei Zahlen aus der Befehlszeile liest und deren ggT. zurückgibt, ohne die ggT von zwei von ihnen und dann die dritte zu berechnen.

Das Ergebnis wird in der Standardausgabe angezeigt.

### Aufgabe 8:

Schreiben Sie ein Shell-Skript, das eine Anzahl von Ordner erstellt. Diese Zahl wird in der Befehlszeile als dem ersten Parameter eingegeben. (die Ordner befinden sich auf derselben Ebene). In jeder neu erstellten Ordner werden eine Anzahl von Dateien erstellt. Die Anzahl der Dateien wird als zweiten Parameter in der Befehlszeile eingegeben.

### Aufgabe 9:

Schreiben Sie ein Shell-Skript, das überprüft, ob zwei Vektoren

$$u = ai + bj + ck,$$

$$v = ei + fj + gk,$$

senkrecht sehen oder nicht. Die Koordinaten der Vektoren werden in der Befehlszeile in der folgenden Reihenfolge als Parameter gelesen: a b c e f g

Das Ergebnis wird in der Standardausgabe angezeigt:

ja - falls die Vektoren senkrecht sind

nein - sonst.

### **Aufgabe 10:**

Schreiben Sie ein Shell-Skript, das bestimmt, ob drei Zahlen pythagoreisch sind oder nicht. Die Zahlen werden als Parameter in die Befehlszeile eingegeben.

Das Ergebnis wird in der Standardausgabe angezeigt:

ja - falls die Zahlen pythagoreisch sind

nein - sonst.