

# Assignment 1

## Abgabe bis 6. Woche

Je nach Halbgruppe 3./4. April

### Achtung!

Jeder Studierende muss in der 6. Woche zwei gelöste Aufgaben 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 4. Woche verteilt.

## Aufgabe 1

### **(grep)**

Zeigen Sie den Inhalt aller Dateien mit der Erweiterung ".txt" aus einem Ordner und aus allen seinen Unterordner an. Der Ordner wird als Parameter in der Befehlszeile angegeben. Der Inhalt der Dateien wird nacheinander in der Standardausgabe angezeigt (ohne etwas anderes angezeigt zu werden).

### **(sed)**

Löschen Sie alle Vorkommen einigen Wörtern in einer Datei. Die Datei und die Wörter werden als Parameter in der Befehlszeile angegeben. Der erste Parameter ist immer der Dateiname. Die Datei wird geändert und in der Standardausgabe wird nichts angezeigt.

## Aufgabe 2

### **(grep)**

Zeigen Sie in der Standardausgabe die Namen der Dateien an, die ein bestimmtes Wort enthalten. Die Dateien und das Wort werden als Parameter in der Befehlszeile angegeben. Der erste Parameter ist immer der Dateiname. Zeigen Sie auch die Anzahl dieser Dateien an. Die Dateinamen werden einzeln in einer Zeile angezeigt. Die Wörter werden als durch Leerzeichen getrennt betrachtet.

### **(sed)**

Löschen Sie alle Zeilen aus Dateien, die einen bestimmten Text enthalten. Die Dateien und der Text werden als Parameter in der Befehlszeile angegeben. Der Text wird der erste Parameter sein. Die Dateien werden nicht geändert, aber ihr verarbeiteter Inhalt wird in der Standardausgabe angezeigt.

### Aufgabe 3

#### (grep)

Zeigen Sie bei der Standardausgabe die Namen aller Binärdateien aus einigen Ordnern an. Die Ordner werden als Parameter in der Befehlszeile angegeben. Man kann eine Binärdatei als jede Datei betrachten, die nicht nur Text enthält ("Klartext").

#### (sed)

Löschen Sie aus den ersten 30 Zeilen der Dateien diejenigen, die einen bestimmten Text enthalten. Sowohl die Dateien als auch der angegebene Text werden als Parameter in der Befehlszeile angegeben. Der Text wird der erste Parameter sein. Die Dateien werden nicht geändert, aber ihr verarbeiteter Inhalt wird in der Standardausgabe angezeigt.

### Aufgabe 4

#### (grep)

Schreiben Sie ein Shell-Skript, das einen Ordernamen als Parameter erhält. Das Skript zeigt die Namen der ASCII-Textdateien (nicht nur die mit der Erweiterung ".txt") aus diesem Ordner an. Die Dateinamen werden sortiert und in der Standardausgabe angezeigt.

#### (sed)

Fügen Sie vor jedem Kleinbuchstaben ein Wort hinzu. Das Wort wird als erster Parameter angegeben und die zu verarbeitende Datei wird durch den zweiten Parameter bestimmt. Die Parameter werden in der Befehlszeile angegeben. Das Ergebnis wird in der Standardausgabe angezeigt.

## Aufgabe 5

### (grep)

Zeigen Sie die Namen allen ".c" -Dateien aus einem Ordner an, der mindestens 3 Bibliotheken enthält. Der Ordner wird als Parameter in der Befehlszeile angegeben. Die Dateinamen werden in der Standardausgabe angezeigt.

### (sed)

Löschen Sie alle Wörter aus einer Datei, die mindestens eine Ziffer enthalten. Die Datei wird als Parameter in der Befehlszeile angegeben. Das Ergebnis wird in der Standardausgabe angezeigt. Für diese Aufgabe enthalten die Wörter nur Buchstaben und Zahlen und sind durch Leerzeichen abgetrennt.

## Aufgabe 6

### (grep)

Für eine Liste von Ordnern (und deren Unterordner) löschen Sie alle Dateien, deren Namen in einer angegebenen Datei enthalten sind. Die Ordnerliste und die Datei werden als Parameter in der Befehlszeile angegeben. Die Datei ist der erste Parameter. Das Skript wird nichts in der Standardausgabe anzeigen.

### (sed)

Ersetzen Sie alle Zeichen, die nicht Großbuchstaben sind, durch ein bestimmtes Zeichen. Das Zeichen wird durch den ersten Parameter und die zu verarbeitende Datei wird durch den zweiten Parameter in der Befehlszeile angegeben. Das Ergebnis wird in der Standardausgabe angezeigt.

## Aufgabe 7

### (grep)

Zeigen Sie die Namen allen ".c" -Dateien in einem bestimmten Ordner an, der "define" enthält (#define). Der Ordner wird als Parameter in der Befehlszeile angegeben. Die Dateinamen werden in der Standardausgabe sortiert (nach dem Namen) und einzeln in einer Zeile angezeigt.

### (sed)

Ersetzen Sie alle kleinen Selbstlaute (Vokale) in einer angegebenen Datei als entsprechenden Großbuchstaben um und umgekehrt. Die Datei wird als Parameter in der Befehlszeile angegeben. Das Ergebnis wird in der Standardausgabe angezeigt.

## Aufgabe 8

### (grep)

Schreiben Sie ein Shell-Skript, das als Parameter in der Befehlszeile die Namen einiger Dateien empfängt und in der Standardausgabe nur die Namen der Shell-Skripte anzeigt. Die Namen der Shell-Dateien werden sortiert (nach dem Namen) und einzeln in einer Zeile angezeigt. Eine Datei wird als Shell-Skript betrachtet, wenn das Dienstprogramm "file" diese als Shell-Skript betrachtet.

### (sed)

Ersetzen Sie das dritte Wort in jeder Zeile einer Datei durch das erste Wort. Die Wörter werden durch ':' abgetrennt und enthalten nur Buchstaben und Zahlen. Die zu verarbeitende Datei wird durch den ersten Parameter in der Befehlszeile angegeben. Das Ergebnis wird in der Standardausgabe angezeigt. Es wird davon ausgegangen, dass die Datei in allen Zeilen mindestens 3 Wörter enthält.

## Aufgabe 9

### (grep)

Berechnen Sie die monatlichen Kosten der Kunden eines Finanzberatungsunternehmens. Jeder Kunde hat eine Datei mit seinem Namen mit der Erweiterung ".csv", in der die monatlichen Einnahmen und Ausgaben dieses Kunden gespeichert sind. Eine solche Datei besteht aus mehreren Zeilen, und jede Zeile enthält die folgenden Informationen in dieser Reihenfolge und durch Komma abgetrennt:

- "IN" / "OUT"; "IN", wenn der Betrag ein Einkommen ist, oder "OUT", wenn es sich um eine Ausgabe handelt;
- den Namen dieser Einnahmen oder Ausgaben;
- der Geldbetrag dieser Einnahmen oder Ausgaben.

Schreiben Sie ein Shell-Skript, das den Namen eines Kunden als Parameter in die Befehlszeile erhält und den Betrag der monatlichen Ausgaben dieses Kunden in der Standardausgabe angezeigt.

Beispiel für ein solcher Datei:

IN,Gehalt,3500

OUT,Gasrechnung,200

OUT,RCA,450

### (sed)

Löschen Sie die Wörter aus den Stellen 2 und 4 aus einer Datei. Die Datei wird als Parameter in der Befehlszeile angegeben. Die Wörter enthalten nur Buchstaben und Zahlen und sind durch Leerzeichen abgetrennt. Das Ergebnis wird in der Standardausgabe angezeigt.

## Aufgabe 10

### (grep)

In einer "stoc.csv" Datei sind die Informationen über die Prozessoren aus dem Stock eines Ladens vorhanden. Auf jeder Linie sind die folgenden Informationen über je ein Prozessor in dieser Reihenfolge zu finden: Hersteller des Prozessors, Modell des Prozessors, die Quantität auf Stock. Schreiben Sie ein Shell-Skript, das den Verkäufer zur Überprüfung des Stocks helfen soll, wenn ein Kunde für ein oder mehrere Prozessoren verlangt. Das Skript erhält folgende Informationen als Parametern in der Befehlszeile: der Hersteller oder das gesuchte Modell und wird in der Standardausgabe die Linien aus der "stoc.csv" Datei anzeigen, die zu den Suchergebnissen und zu der Stock Quantität entsprechen.

Beispiel "stoc.csv" Datei:

Intel, i5, 10

Intel, i7, 0

AMD, Ryzen, 7

Beispiel für Befehlszeile: "script.sh intel" wird nur "intel, i5, 10" zeigen.

### (sed)

Wechseln Sie die Wörtern aus den Stellen 1 und 3 aus einer Datei. Die Datei wird als Parameter in der Befehlszeile ausgegeben. Die Wörter enthalten nur Buchstaben und sind von jedem anderen Zeichen getrennt. Das Ergebnis wird in der Standardausgabe angezeigt.