

Ziel der Übung:

- Reguläre Ausdrücke benutzen können.  
`grep, sed, awk`

## Vorbereitung

Laden Sie sich die das *Tarball* (RegulaereAusdruecke.tar.gz) von der eLearning-Plattform und extrahieren Sie es im Verzeichnis Uebung7.

### grep

#### 1.1: Miniaufgaben

Starten Sie das Shell-Skript `learnRegex.sh`, indem Sie `./learnRegex.sh` ausführen und lösen Sie die Miniaufgaben. Sie können mittels **Strg + c** das Programm beenden oder indem Sie "Skip" eingeben eine Aufgabe überspringen.

#### 1.2: Telefonnummern

Nutzen Sie `grep` mit den Optionen `-Pao` und einen Regulären-Ausdruck, um alle Telefonnummern aus der defekten Datei (`kaputt`) zu extrahieren (Lösung: 13 Telefonnummern). Der Reguläre-Ausdruck soll möglichst genau folgende Formate von Telefonnummern mat-chen und möglichst kurz sein:

```
+XX XX XXXXXXXX
+XX (XX) XXXXX - XX
+XX (X)XX XXXXX-XX
+XX-XX-XXXXXXX
+XX XX XXXXX-XX
XXXX X XXXXXX
X XX / XX XX XX
XXX XXXXX-XX
```

#### 1.3: E-Mail Rejected

Finden Sie alle E-Mailadressen die in der Datei `mail.log` enthalten sind, die den Status `NOQUEUE` haben. Nutzen Sie `grep` mit den Optionen `-Eo`, `sort` und `uniq`, um alle Adresse einmalig auszugeben (Lösungen: 64 Adressen).

## 1.4: **services1**

Die Datei **services** enthält Dienstnamen und deren zugehörige Standard-Portnummer, um eine Service-Namensaufenlösen zu ermöglichen. Filtern Sie die Datei nach folgenden Kriterien und erzeugen Sie daraus die Datei (**services1**).

- Keine Leerzeilen enthalten
- Keine reinen Kommentarzeilen (mit "#" Beginnen) enthalten
- Nicht die Zeichenkette "udp" enthalten
- Nur Zeilen die ein Kommentar zum Dienst enthält

Tipp: Denken Sie an die Option -v (Lösung: 111 Zeilen).

## **sed**

### 2.1: Miniübungen

Nutzen Sie die Option -E für alle sed-Kommandos.

1. Erstellen Sie einen sed-Befehl, welcher jedes Wort klammert.

*Belgien ist eine wunderschöne Stadt → (Belgien) (ist) (eine) (wunderschöne) (Stadt)*

2. ... welcher alle Buchstaben, die nicht am Wortanfang stehen, klein schreibt.

*HaRRry, hOl sChOn mAℓ den WaGEN → Harry, hol schon mal den Wagen*

3. ... welcher alle Vokale groß schreibt.

*Magermilchjoghurt → MAgErmIlchjOghUrt*

4. ... welcher alle leeren Zeilen löscht.

### 2.2: **services2**

Nutzen Sie Ihr Ergebnis aus der Aufgabe 1.4 und entfernen Sie aus der Datei **services1**, mittels **sed**, alle Kommentarzeichen (#) und die Zeichenkette /tcp. Leiten Sie Ihre Ergebnisse in die Datei **services2** um.

## awk

### 3.1: services3

Nutzen Sie Ihre Ergebnis aus Aufgabe 2.2, um die Datei `service3` mit dem folgendem Format zu erzeugen. Nutzen Sie `awk` mit dem Parameter `-F ' +'` (zwei Leerzeichen), um die Spalten korrekt zu separieren.

```
"FTP, data" benutzt Port 20
"FTP. control" benutzt Port 21
"SSH Remote Login Protocol" benutzt Port 22
"Simple Mail Transfer Protocol" benutzt Port 25
"Host Name Server" benutzt Port 42
"Domain Name Server" benutzt Port 53
"World Wide Web" benutzt Port 80
```

### 3.2: Notenschnitt

In der Datei `Notenliste` sind alle Noten einer Prüfung eingetragen. Berechnen Sie den Notenschnitt mittels `awk` (Lösung: Ø2,6 ).

## grep, sed und awk (optional)

### 4.1: Zeilenzählen

Schreiben Sie einen Ausdruck der alle Codezeilen eines Repositories zählt (z.B. alle `*.java` Dateien rekursiv), wobei Kommentarzeilen, Leerzeilen und Zeilen die nur Klammern enthalten nicht mitgezählt werden sollen. Da jede Datei einen Header mit Namen beinhaltet soll am Ende für jeden Namen die Anzahl an geschriebenen Codezeilen ausgegeben werden.

### 4.2: From → To

Suchen Sie aus der Datei `mail.log` alle versendeten E-Mails heraus indem Sie zunächst alle Nachrichten mit der gleichen Message-ID kombinieren und anschließend die Adresse von den Feldern `from=` und `to=` im folgenden Format ausgeben (Lösung: 33 Ergebnisse):

```
abc@sonne.de -> xyz@mond.de
...

```