

## Text Processing Aufgaben

### Einleitung:

Lesen Sie zuerst das Dokument **TextProcessingTheorie** und verschaffen Sie sich einen Überblick über die Utilities zum Verarbeiten von Texten.

In den folgenden Aufgaben werden Sie einige dieser Utilities anwenden.

- ⇒ Führen Sie die Befehle aus.
- ⇒ Beobachten Sie, was ausgegeben wird und schauen Sie nach was in den Dateien steht.
- ⇒ Halten Sie den in der Konsole ausgeführten Befehl mit dem Resultat in Ihrer Doku fest.
- ⇒ Halten Sie den Output in den Dateien in Ihrer Doku fest.
- ⇒ Halten Sie Ihre Erkenntnisse, die Antwort auf die Fragen in Ihrer Doku fest.

### Vorbereitung:

- ⇒ Erstellen Sie in einem Verzeichnis und kopieren Sie die Dateien zu den Übungen in dieses Verzeichnis.

### Aufgabe: Anwenden von join

Mit join kann man den Inhalt zweier Dateien zusammenführen.

Zum Beispiel die Namen von Personen mit generierten Passwörtern.

Person.txt

1	Hans Muster	1
2	Paul Keller	2
3	Jose Pinto	3
4	heinz Herrman	4
5	Ruedi Falk	5

und

Passwort.txt

1	1	user11	123@11
2	2	user12	123@12
3	3	user13	123@13
4	4	user14	123@14
5	5	user15	123@15
6	6	user16	123@16

werden zu

UserAndPassword.txt

1	1	Hans Muster	user11	123@11
2	2	Paul Keller	user12	123@12
3	3	Jose Pinto	user13	123@13
4	4	heinz Herrman	user14	123@14
5	5	Ruedi Falk	user15	123@15

vereint

- ⇒ Das Problem ist, dass Join wissen muss, welche Zeile der Datei Person.txt zu Datei Passwort.txt passt. Was also in der neuen Datei auf einer Zeile stehen soll
- ⇒ Hier helfen uns die ID in den beiden Dateien:
  - Person.txt → Spalte 3
  - Passwort.txt → Spalte 1
- ⇒ Die man verrät uns, mit welcher Option ich join diese Spalten Nummern mitteilen kann:

```
-1 FIELD
    join on this FIELD of file 1

-2 FIELD
    join on this FIELD of file 2
```

- ⇒ Beispiel:  
`join -1 2 -2 3 datei1.txt datei2.txt`

Join sucht die Zeilen mit den gleichen Nummern und zwar in der Spalte 2 der Datei datei1.txt und in der Spalte 3 der Datei datei2.txt.  
 Findet join gleiche Nummern, so vereint es diese Zeilen zu einer Zeile der neuen Datei.

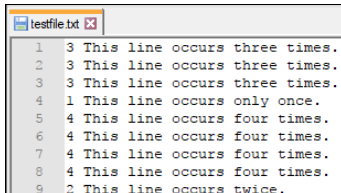
- ⇒ Nun ist das noch nicht der ganze richtige Befehl. Passen Sie den Aufruf an, dass er mit unseren Dateien *Person.txt* und *Passwort.txt* funktioniert. Zudem soll der Output in die Datei *UserAndPassword.txt* umgeleitet werden.

### Aufgabe: Anwenden von *cat*, *uniq* und *sort*

---

Schreiben Sie ein Script, dass

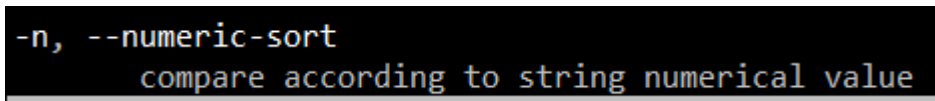
- den Inhalt der Datei *testfile.txt* mittels *cat* ausgibt



```
1 3 This line occurs three times.
2 3 This line occurs three times.
3 3 This line occurs three times.
4 1 This line occurs only once.
5 4 This line occurs four times.
6 4 This line occurs four times.
7 4 This line occurs four times.
8 4 This line occurs four times.
9 2 This line occurs twice.
```

→ *cat testfile.txt*

- Leiten Sie nun den Output über eine Pipe auf den Befehl *uniq* um. Damit werde doppelt vorkommende Zeilen entfernt
- Leiten Sie nun den Output über eine Pipe auf den Befehl *sort* weiter
- Erweitern Sie nun den Befehl *sort* um die Option, damit die Sortierung numerisch nach Zahlen geschieht



```
-n, --numeric-sort
        compare according to string numerical value
```

- Finden Sie selber heraus, wie die Option von *sort* für das "reverse" sortieren ist. Damit also zuerst die grossen, dann die kleine Zahlen erscheinen.

### Aufgabe: Mehrere Dateien mit *cat* ausgeben

---

Cat kann auch gleich mehrere Dateien hintereinander ausgeben:

```
cat datei1.txt datei2.txt
```

Schreiben Sie ein Script, dass

- den Inhalt der drei Dateien *fox1.txt*, *fox2.txt*, *fox3.txt* mittels *cat* ausgibt
- den Output über eine Pipe an *sort* weitergibt (ohne Optionen)
- den neuen Output über eine Pipe an *uniq* ausgibt  
Mehrfaches Vorkommen der gleichen Inhalte nur einmal ausgeben.
- Und das Resultat in die Datei *final.txt* schreibt

## Aufgabe: Eine Datei mit Zeilennummern versehen

---

Eine ganz einfache Aufgabe.

Das Utility *nl* ist klein aber manchmal sehr nützlich. Es nummeriert einfach die Zeilen einer Datei durch.

Probieren Sie es aus:

- Nummerieren Sie die Zeilen verschiedener Dateien mit Hilfe von *nl*

## Aufgabe: Statt Seitenweise Man-Page nur das wichtigste

---

Ausgangslage:

- Die Ausgaben von *man* zu einem Befehl kann viele Seiten umfassen.

```
WC(1)                                User Commands
NAME
  wc - print newline, word, and byte counts for each file
SYNOPSIS
  wc [OPTION]... [FILE]...
  wc [OPTION]... --files0-from=F
DESCRIPTION
  Print newline, word, and byte counts for each FILE, and a total
  specified. A word is a non-zero-length sequence of characters delimi
  With no FILE, or when FILE is -, read standard input.
  The options below may be used to select which counts are printed, alw
  newline, word, character, byte, maximum line length.
  -c, --bytes
        print the byte counts
```

Idee:

- Wir wollen erreichen, dass nur die Kurzfassung ausgegeben wird. Und zwar die Zeile mit der Kurzerklärung:

```
wc - print newline, word, and byte counts for each file
```

Es geht noch weiter...

Plan am Beispiel von *man wc*

1. Wir schneiden aus der Ausgabe von *man* die führenden Zeilen heraus und erhalten so:

```
WC(1)                                     User Commands
NAME
wc - print newline, word, and byte counts for each file
```

In meiner Cygwin Bash sind das die ersten 4 Zeilen.

Genau das kann das Utility *head*.

*man | head -n4*

2. Jetzt hat es aber am Anfang noch einige Zeilen zuviel.  
Diese schneiden wir mit Hilfe von *tail* weg.  
➔ Leiten Sie das Resultat von oben über eine Pipe weiter an *tail -n1*

So erhalten Sie das

```
wc - print newline, word, and byte counts for each file
```

3. Jetzt sollte das ganze noch links bündig geschoben werden.  
Hier hilft uns *cut*, das aus einer Zeile Teile wegschneiden kann

*cut -b 8-*

➔ so schneidet *cut* die ersten Leerzeichen bis zum 8ten Zeichen weg

So erhalten Sie das

```
wc - print newline, word, and byte counts for each file
```

Schreiben Sie ein Script, dass

- Der obige Ablauf soll automatisiert werden.  
Zu einem beliebigen Befehl soll nur die Kurzfassung der *man* ausgegeben werden.
- Der Befehl zBsp *wc* soll als Parameter dem Script übergeben werden.  
Und der Parameter muss in den Befehl eingebaut werden

Wie man das macht, das finden Sie in den "GrundlagenZusammenfassung".  
Schauen Sie nach!!

Uns so soll es aussehen, wenn man das Script aufruft

```
Peter Rutschmann@Odysseus ~
$ ./manShorty.sh wc
wc - print newline, word, and byte counts for each file
```