

Unterschiede in Dateien mit diff

- `diff datei1 datei2`
vergleicht die Inhalte der beiden Dateien zeilenweise (bei Verzeichnissen die enthaltenen Dateien) und gibt die sich unterscheidenden Zeilen auf stdout aus
- wichtige Optionen:
 - i vergleicht case insensitive
 - r vergleicht Unterverzeichnisse rekursiv
 - y zweispaltige Ausgabe

Word Count `wc`

- `wc [datei ...]` zählt Zeilen, Wörter und Bytes in den als Argumente angegebenen Dateien und gibt diese auf `stdout` aus
- bei mehreren Argumenten: Ausgaben für jede einzelne Datei und der Gesamtsummen
- bei fehlenden Argumenten: zählen in `stdin`
- wichtige Optionen:
 - l nur Zeilen
 - w nur Wörter
 - c nur Bytes (Characters)

Zeilen sortieren mit `sort`

- `sort [datei ...]` sortiert die Zeilen der als Argumente angegebenen Dateien alphabetisch
- bei fehlenden Argumenten wird `stdin` sortiert
- wichtige Optionen:
 - `kn` Sortierung nach Spalte n (*whitespace* als Spaltentrenner)
 - `kn, m` Sortierung nach Spalte n bis Spalte m
 - `t sep` Spaltentrenner ist *sep*
 - `n` numerische Sortierung
 - `u` mehrfach auftretende Zeilen werden nur einmal ausgegeben
- Die Numerierung der Spalten beginnt bei 1.

Zeilen und Spalten ausschneiden

- `head -n [datei ...]` gibt die ersten n Zeilen von stdin bzw. von jeder als Argument genannten Datei auf stdout aus
- `tail -n [datei ...]` gibt die letzten n Zeilen von stdin bzw. von jeder als Argument genannten Datei auf stdout aus
- `cut -c | -f spaltenliste [datei ...]` schreibt die in *spaltenliste* benannten Spalten von stdin bzw. von jeder als Argument genannten Datei auf stdout
 - bei `-c` : jedes Symbol definiert eine Spalte
 - bei `-f` : Spaltentrenner ist *tab*
 - Option `-d sep` : Spaltentrenner ist *sep*
 - *spaltenliste* im Format x_1, x_2, \dots, x_k mit $x_i = n$ oder $x_i = n-m$
 - Die Reihenfolge der Spalten kann nicht verändert werden.
Die Numerierung beginnt bei 1.

Zusammenfügen von Dateien mit `paste`

- `paste datei1 [datei ...]`
fügt die Zeilen der als Argumente genannten Dateien als durch *tab* voneinander getrennte Spalten zusammen
- Ein `-` als Argument fügt an dieser Stelle die Zeilen von `stdin` ein
- Option `-d sep` : Spaltentrenner ist *sep*
- Option `-s` : seriell zusammenfügen der Dateien

Symbolische Übersetzung mit `tr`

- `tr list1 list2`
übersetzt die Zeichen aus *list1* in die entsprechenden (nach Reihenfolge) Zeichen aus *list2*
- `tr` ist ein reiner **Filter**, d.h., `tr` liest stets von `stdin` und schreibt stets auf `stdout`
- wichtige Optionen:
 - d löscht Zeichen aus *list1*
(keine Übersetzung, keine *list2*)
 - s reduziert bei der Übersetzung aufeinanderfolgende gleiche Zeichen auf ein einzelnes

Der Stream-Editor sed

- nicht interaktiv
- `sed [-e] 'sed-Kommando' [datei ...]`
schreibt den Inhalt von `stdin` bzw. von den als Argumente angegebenen Dateien auf `stdout`, nach zeilenweiser Bearbeitung so, wie durch das *sed-Kommando* festgelegt
~> ursprüngliche Dateien bleiben unverändert!
- wichtige *sed-Kommandos*:
 - `m,nd` löscht die Zeilen *m* bis *n*
 - `m,$d` löscht alle Zeilen ab Zeile *m*

<i>/regex/d</i>	löscht alle Zeilen, die eine zu <i>regex</i> passende Zeichenkette enthalten
<i>m, /regex/d</i>	löscht alle Zeilen von Zeile <i>m</i> bis zur ersten Zeile, die eine zu <i>regex</i> passende Zeichenkette enthält
<i>s/regex/string/</i>	ersetzt in allen Zeilen die erste Zeichenkette, die zu <i>regex</i> passt, durch <i>string</i>
<i>s/regex/string/g</i>	ersetzt alle Zeichenketten, die zu <i>regex</i> passen

Anstelle von */* kann jedes beliebige Zeichen verwendet werden, das weder in *regex* noch in *string* vorkommt.

- Option *-e*: Kombination mehrerer sed-Kommandos, z.B.:

```
sed -e '3,$d' -e 's/ */g' text
```