

sed - Anwendung

- Typische Anwendungen von `sed`:

```
Mehrere Dateien automatisch und fehlerfrei bearbeiten.  
Umformulierung von Texten.  
Konvertierung von Daten.  
Suchen und Ersetzen von Texten.
```

- Der `sed` kann **verschachtelte Strukturen** (z.B. *BEGIN-END-Paare* oder *Klammern*) nicht gemäß ihrer **Hierarchie** verarbeiten.
 - Dazu wird der `awk` benötigt.

Version 3.1
7.9.2004

UNIX Aufbaukurs
© Thomas Birnhäler, OSTC GmbH

176

sed - Aufruf

```
sed [OPTIONEN] [-e] 'KOMMANDO' [DATEI...]  
sed [OPTIONEN] -f SKRIPT [DATEI...]
```

- Zeichenstrom-orientierter Editor (**stream editor**).
- Interpretiert **Kommandos** auf der Kommandozeile (**execute**) oder aus einer `sed`-Skriptdatei (**file**).
- `sed`-Kommandos sollten in **einfachen Hochkommata** `'...'` stehen, da sie häufig `;`-Zeichen enthalten.
- Die **Eingabe** kommt aus der angegebenen `DATEI` oder von der Standard-Eingabe.
- Die **Ausgabe** erfolgt auf die Standard-Ausgabe.

Version 3.1
7.9.2004

UNIX Aufbaukurs
© Thomas Birnhäler, OSTC GmbH

177

sed - Funktionsweise

Funktionsweise des `sed`

- Die Eingabe wird **Zeile für Zeile** gelesen.
- Jede Eingabezeile wird in den **Musterbereich** kopiert.
- Alle** Editierkommandos werden **der Reihe nach** auf diese eine Eingabezeile angewendet.
 - Ändert ein Kommando eine Eingabezeile, so wirken alle nachfolgenden Kommandos auf diese (**geänderte**) **Kopie** der Eingabezeile im Musterbereich (*nicht auf die ursprüngliche Eingabezeile*).

Version 3.1
7.9.2004

UNIX Aufbaukurs
© Thomas Birnhäler, OSTC GmbH

178

sed - Funktionsweise

- Nach der Abarbeitung **aller Kommandos** wird die Ergebniszeile (*standardmäßig*) auf der Standard-Ausgabe **ausgegeben**.
 - Die **ursprüngliche Eingabedatei bleibt unverändert**.
- Die Editierkommandos werden auf **alle Eingabezeilen** angewendet, es sei denn, sie werden durch **Zeilenadressierung** auf bestimmte Zeilen beschränkt.

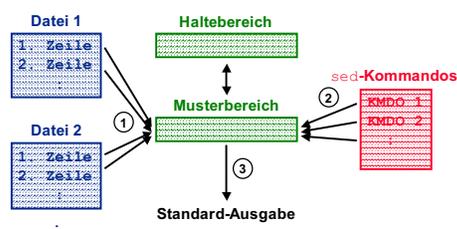
Version 3.1
7.9.2004

UNIX Aufbaukurs
© Thomas Birnhäler, OSTC GmbH

179

sed - Funktionsweise

Funktionsweise des `sed`



Version 3.1
7.9.2004

UNIX Aufbaukurs
© Thomas Birnhäler, OSTC GmbH

180

sed - Syntax

Syntax der `sed`-Kommandos

- `sed`-Kommandos haben die allgemeine Form `[ADRESSE][,ADRESSE][!]
[KOMMO][ARGUMENT]...`
- Ein `sed`-Kommando kann 0, 1 oder 2 **Adressen** enthalten, die seine Wirkung beschränken.
- Eine **Adresse** kann sein:
 - Eine **Zeilen-Nummer** 1, 2, ... (*ohne Schrägstriche*).
 - Das Symbol `$`, das für die **letzte Zeile** steht.
 - Ein **Regulärer Ausdruck** / *REGEXP*/ in Schrägstrichen, der alle Zeilen auswählt, zu denen er passt.

Version 3.1
7.9.2004

UNIX Aufbaukurs
© Thomas Birnhäler, OSTC GmbH

181

sed - Adressen

Bei Angabe von:	Gilt Kommando für:
* Keiner Adresse	Jede Eingabezeile
* 1 Adresse	Jede Zeile, die der Adresse entspricht (z.B. 1-100 /BEGIN/ 5)
* 2 durch Komma getrennten Adressen (Bereich)	Die erste passende Zeile und alle weiteren Zeilen bis einschließlich zu einer Zeile, die mit der zweiten Adresse übereinstimmt (wiederholt) (z.B. 5,12 /BEGIN/ /END/)

Folgt auf die Adrefangabe ein !, so gilt das Kommando für alle Zeilen, die **nicht** zur Adresse passen (z.B. \$! / ^BEGIN/ / ^END/!)

sed - Adressen

- Mit **geschweiften Klammern { }** wirkt eine Adresse auf mehrere Kommandos gemeinsam:

```
ADRESS-BEREICH {  
    KMD01  
    KMD02  
}
```

- Die Kommandos werden nacheinander auf alle zum Adreßbereich passenden Zeilen angewendet.
- Mehrere `sed`-Kommandos auch durch ";" trennbar:
`/^#/d;s/#.*//;/^ *$/d`

sed - Kommandos

- `sed`-Kommandos bestehen aus einem **Zeichen** (manche mit einem *Argument*), die wichtigsten sind:

i Zeilen einfügen (insert)	y Zeichen tauschen (yield)
a Zeilen anhängen (append)	r Datei einfügen (read)
c Zeilen ändern (change)	p Zeilen ausgeben (print)
d Zeilen löschen (delete)	q sed beenden (quit)
s Text ersetzen (substitute)	b Sprung (branch)
	t Bedingter Sprung (test)

sed - Kommandos

- Standardmäßig** wird jede Eingabezeile nach der Anwendung aller `sed`-Kommandos ausgegeben.
 - Wird Kommando `p` (**print**) verwendet, gleichzeitig Option `-n` (**noprint**) angeben, um nicht Ergebniszeilen **doppelt** auszugeben.
- Die Kommandos `a` (**append**), `c` (**change**) und `i` (**insert**) sind mit "\" abzuschließen.
 - Angabe **beliebig vieler Zeilen** mit "\" als letztem Zeichen.
 - Letzte Zeile nicht mit "\" abschließen.**

sed - Beispiele

Beispiel

```
sed "s/made/raupe/g" DATEI Ersetzt in allen Zeilen überall made durch raupe (substitute + global).
```

```
sed "/made/d" DATEI Löscht alle Zeilen, die made enthalten (delete).
```

```
sed -n '/made/, $p' DATEI Gibt alle Zeilen zwischen 1. made und dem Dateiende aus (print).
```

```
sed '/made/!d' DATEI Löscht jede Zeile, die nicht made enthält (delete).
```

```
sed -e '/^#/d' -e 's/#.*//' -e '/^ *$/d' /etc/hosts Entfernt Kommentare und Leerzeilen aus Datei /etc/hosts.
```

sed - Beispiele

Beispiel

```
sed "i\\ \b Fügt Text vor Zeile 1 ein (insert).
```

```
Am Textanfang eingefügt"
```

```
sed '$a\\ \b Hängt Text nach der letzten Zeile an (append).
```

```
Am Textende angehängt"
```

```
sed '10,19c\\ \b Ersetzt die Zeilen 10-19 durch die angegebenen 3 Zeilen (change).
```

```
Ersatz Zeile1\\  
Ersatz Zeile2\\  
Ersatz Zeile3'
```

sed - Beispiele

Beispiel

```
sed '/^Nr [1-5]/y/123/ABC/'
```

 Ersetzt in Text der Form Nr N die Ziffern 123 durch die Großbuchstaben ABC (**y**ield).

```
sed '/#include/r info'
```

 Fügt nach der Zeile mit dem Text #include die Datei info ein (**r**ead).

sed - Skript

- Es empfiehlt sich, komplizierte sed-Kommandos in einer **Skript-Datei** zusammenzufassen:
 - Kann durch eine "**Shee-Bang-Zeile**" (*1. Zeile!*) der Form `#!/bin/sed -f` in ein **direkt ausführbares Skript** umgewandelt werden.
 - In der Kommandodatei sind die Quotierungszeichen um Kommandos **wegzulassen**.
 - "#" als 1. Zeichen einer Zeile leitet **Kommentar** ein (*nach einem Kommando ist kein Kommentar erlaubt*).
 - **Leerzeilen** sind erlaubt (z.B. zur *Gliederung*).

Übung 19a

- **Simulieren** Sie das Kommando head mit dem sed.
 - Geben Sie die Zeilen 1-10 von gedicht aus.
- **Simulieren** Sie das Kommando tail mit dem sed.
 - Geben Sie die Zeilen 20-Ende von gedicht aus.
- **Simulieren** Sie eine Kombination von head und tail mit dem sed.
 - Geben Sie die Zeilen 11-19 von gedicht aus.
 - Geben Sie die Zeilen 1-4 und 10-20 von gedicht aus.

Übung 19b

- **Geben** Sie die Datei gedicht mit Hilfe des sed auf dem Bildschirm aus (*analog cat*).
- **Geben** Sie die Zeilen der Datei gedicht aus:
 - Die das Wort made enthalten.
 - Die das Wort made **nicht** enthalten.
 - Die keine **Leerzeilen** sind (*erst Definition überlegen!*).
- **Ersetzen** Sie überall in der Datei gedicht.
 - Das Wort made durch MARMELADE.
 - **Leerzeichen** und die **Vokale** aeiou durch nichts.

Übung 19c

- **Fügen** Sie am Anfang der Datei gedicht zwei Zeilen mit dem aktuellen Datum und der Uhrzeit ein:
date +%D und date +%T
- **Hängen** Sie an das Ende der Datei gedicht folgende Zeile an:
(C) 2002, UNIX-Aufbaukurs
- **Ersetzen** Sie in Datei gedicht jeweils die Zeilen von made bis made durch den Text
Keine Maden mehr vorhanden

Übung 19d

Zusatzaufgabe

- **Löschen** Sie mit dem sed das Verzeichnis "/usr/bin" aus dem Suchpfad PATH.
 - Berücksichtigen Sie, daß "/usr/bin" **am Anfang, in der Mitte** oder **am Ende** des Suchpfades stehen kann.
- **Führen** Sie die gleiche Aufgabe für das **aktuelle Verzeichnis** "." durch.
 - Berücksichtigen Sie, daß auch ein ":" **am Anfang/Ende** des Suchpfades oder zwei ":" **direkt hintereinander** im Inneren des Suchpfades für das aktuelle Verzeichnis stehen.