

**BEZEICHNUNG**

`sed` – (stream editor) ist ein Editor zur nicht-interaktiven Textbearbeitung

**SYNTAX**

`sed` [**-nV**] [**--quiet**] [**--silent**] [**--version**] [**-e** *Editorkommando*] [**-f** *Scriptdatei*] [**--expression=** *Editorkommando*] [**--file=** *Scriptdatei*] [*Datei...*]

**BESCHREIBUNG**

`sed` ist ein Editor zur automatischen Textbearbeitung.

Die Bearbeitung erfolgt mit Editorkommandos, die dem `sed` in einer separaten *Scriptdatei* oder direkt in der Kommandozeile übergeben werden. Um in der Kommandozeile mehrere *Editorkommandos* zu übergeben, kann die `-e` Option mehrfach verwendet werden. Die Editorkommandos können auch durch ein Semikolon getrennt werden. Wird nur ein einziges Editorkommando in der Kommandozeile übergeben, kann die Kennzeichnung mit der `-e` Option auch weggelassen werden. Damit die Shell keine Veränderungen an der Zeichenkette mit dem Editorkommando vornimmt, muß sie in Hochkommata eingeschlossen werden.

Eine *Scriptdatei* kann beliebig viele Editorkommandos enthalten, die durch Zeilenende oder Semikolon von einander getrennt werden müssen.

Jedes Kommando besteht aus einem Adreßteil und einem Funktionsteil. Der Adreßteil gibt an, welche Zeilen einer Textdatei mit diesem Kommando bearbeitet werden sollen, und der Funktionsteil beschreibt die Veränderung, die an den im Adreßteil bestimmten Zeilen vorgenommen werden soll. Wenn kein Adreßteil angegeben ist, wird die Funktion mit jeder Zeile ausgeführt.

Die Bearbeitung eines Textes erfolgt, indem die Eingabe zeilenweise in einen Arbeitsspeicher gelesen wird, und dann die Adreßteile aller Editorkommandos der Reihe nach mit dem Text im Arbeitsspeicher verglichen werden. Die Funktionen der passenden Kommandos werden in der Reihenfolge ihres Auftretens ausgeführt. Normalerweise wird nach der Bearbeitung aller Kommandos der Inhalt des Arbeitsspeichers auf die Standardausgabe ausgegeben und danach durch die nächste Eingabezeile ersetzt. Die automatische Ausgabe des Arbeitsspeichers nach jeder Zeile kann mit der Option `-n` unterdrückt werden.

Zusätzlich zu dem Arbeitsspeicher gibt es noch einen Zwischenspeicher (Puffer), der von verschiedenen Funktionen benutzt werden kann.

Der Arbeitsspeicher kann auch mehrere Zeilen auf einmal enthalten.

Im Adreßteil können die Zeilen entweder durch ihre Zeilennummern, oder durch reguläre Ausdrücke ausgewählt werden. Alle Funktionen außer den `a`, `i`, `q` und `=` akzeptieren einen Adressbereich, bei dem eine Start- und eine Endadresse durch ein Komma getrennt angegeben werden. Ein Dollarzeichen steht für die letzte Zeile. Wenn eine Endadresse mit einem regulären Ausdruck bezeichnet ist, wird die erste passende Zeile als Bereichsende eingesetzt.

Reguläre Ausdrücke müssen in einfachen Schrägstrichen (Slashes `/`) eingeschlossen werden. `sed` benutzt die gleichen Routinen zur Auswertung regulärer Ausdrücke wie `emacs` oder `grep`. Darüberhinaus kann auch die an die `sed` Syntax angelehnte Konstruktion `\#Muster\#` (mit jedem beliebigen Zeichen für `#`) benutzt werden, die wie `/Muster/` interpretiert wird.

Im *Muster* kann auch ein `\n` vorkommen, das auf das Zeilenende paßt.

Der `sed` kann folgende Funktionen ausführen:

**a***Text* schreibt den *Text* in die Standardausgabe, bevor die nächste Eingabezeile gelesen wird

**b** *Marke*

springt zur der mit der `:` *Marke* markierten Zeile im Script (nicht im Text) und fährt dort mit dem Programm fort

**c***Text* die im Arbeitsspeicher von `sed` befindliche Zeilen werden gelöscht, und der *Text* in die Standardausgabe geschrieben; wenn ein Adressbereich angegeben ist, wird der *Text* erst am Bereichsende einmal ausgegeben

**d** alle aktuell im Arbeitsspeicher von `sed` befindlichen Zeichen werden gelöscht und die nächste Eingabezeile gelesen; die auf diesen Befehl folgenden Befehle werden nicht mehr bearbeitet, auch

- wenn die Zeilen im Arbeitsspeicher im passenden Bereich liegen
- D** die erste Zeile im Arbeitsspeicher von **sed** wird gelöscht und die nächste Zeile wird gelesen; die auf diesen Befehl folgenden Befehle werden nicht mehr bearbeitet, auch wenn die Zeilen im Arbeitsspeicher im passenden Bereich liegen
- g** der Arbeitsspeicher von **sed** wird durch den Inhalt des Puffers ersetzt; der Inhalt des Arbeitsspeichers geht dabei verloren
- G** der Pufferinhalt wird an den Inhalt des Arbeitsspeichers angehängt
- h** der Inhalt des Arbeitsspeichers wird in den Puffer geschrieben; der Inhalt des Puffers geht dabei verloren
- H** der Inhalt des Arbeitsspeichers von **sed** wird an den Puffer angehängt
- i|Text** (insert) der Text wird sofort in die Standardausgabe geschrieben
- l** der Inhalt des Arbeitsspeichers von **sed** wird ausgegeben; nichtdruckbare Zeichen werden als Oktalzahl dargestellt
- n** der Inhalt des Arbeitsspeichers wird unverändert in die Ausgabe geschrieben und der Arbeitsspeicher durch die nächste Eingabezeile ersetzt
- N** die nächste Eingabezeile wird an den Arbeitsspeicher angehängt; das Zeilenende wird mit in den Arbeitsspeicher geschrieben; die Zeilennummer des aktuellen Bereiches erhöht sich um eins
- p** der Inhalt des Arbeitsspeichers wird in die Standardausgabe geschrieben
- P** die erste Zeile im Arbeitsspeicher wird in die Standardausgabe geschrieben
- q** beendet **sed**; es werden keine weiteren Befehle ausgeführt und keine Zeilen mehr gelesen
- r Datei** der Inhalt der Datei wird ausgegeben, bevor die nächste Zeile gelesen wird
- s/Ausdruck/Ersetzung/[Modus]**  
 (substitute) ersetzt den (ersten) auf den regulären Ausdruck passenden Text durch den Ersetzungstext; es kann auch ein beliebiges anderes Zeichen anstelle von / benutzt werden; als Modus können ein oder mehrere der folgenden angegeben werden
- n** eine Zahl von 1 bis 512 ersetzt nur das n-te Auftreten des Musters
- g** (global) alle auf den Ausdruck passenden Textteile werden ersetzt
- p** wenn eine Ersetzung stattgefunden hat, wird der Inhalt des Arbeitsspeichers von **sed** in die Standardausgabe geschrieben
- w Datei**  
 wenn eine Ersetzung stattgefunden hat, wird der Inhalt des Arbeitsspeichers in die *Datei* geschrieben
- t Marke**  
 verzweigt zur mit der *Marke* versehenen Zeile in der Programmdatei, wenn eine Ersetzung am Inhalt des Arbeitsspeichers vorgenommen wurde, seit die letzte Eingabezeile gelesen wurde, oder seit der letzte **t** Befehl bearbeitet wurde; wenn keine *Marke* angegeben ist, wird an das Ende der Programmdatei verzweigt
- w Datei**  
 schreibt den Inhalt des Arbeitsspeichers in die benannte *Datei*
- x** vertauscht den Inhalt des Puffers mit dem Arbeitsspeicher
- y/Zeichenkette1/Zeichenkette2/**  
 vertauscht jedes auftretende Zeichen aus der Zeichenkette1 mit dem entsprechenden Zeichen der Zeichenkette2; die beiden Zeichenketten müssen gleich lang sein

**! Funktion**

führt die Funktion für alle Zeilen aus, die NICHT in den Bereich passen

**:Marke** setzt eine *Marke* für den *b* und den *t* Befehl

**{...}** die von den Klammern eingeschlossenen und durch Zeilenende oder Semikolon getrennten Funktionen werden als Einheit behandelt

**=** gibt die aktuelle Eingabezeilennummer aus

**\#** leitet einen Kommentar ein; alle folgenden Zeichen bis zum Zeilenende werden ignoriert

**OPTIONEN**

**-n** gibt nur die Zeilen aus, die explizit (durch die Anweisung *p*) ausgedruckt werden sollen

**-V** gibt die Versionsnummer und eine Kurzhilfe aus

**-eZeichenkette**

wendet die Editorbefehle aus *Zeichenkette* auf den Text an

**-f Datei**

liest die Editorbefehle aus der *Datei*

**SIEHE AUCH**

das LunetIX Linuxhandbuch

**AUTOR**

Unbekannt