

## 7 vi

Der Editor vi ist ein leistungsfähiger bildschirmorientierter Editor. Bildschirmorientiert heißt, dass der Editor den Text am Schirm anzeigt. Die Benutzung der Cursortasten ist erlaubt.

vi [-option]... [datei]...

Die folgenden Optionen stehen zur Verfügung:

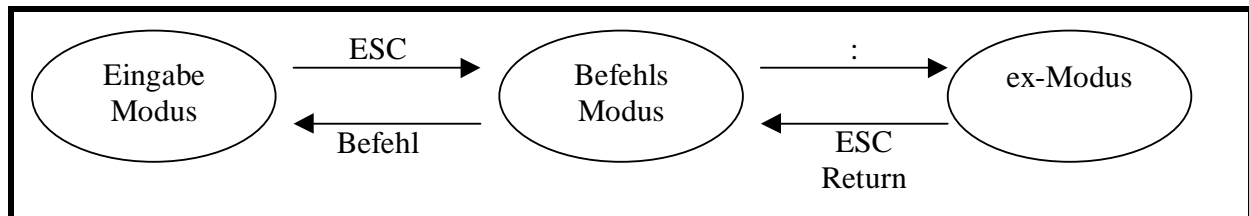
Option	Bedeutung
-R	Die Dateien werden nur lesend geöffnet.
-i	Der Editor wird im Eingabemodus gestartet
+zeilennummer	Es wird sofort zur genannten Zeilennummer gewechselt
+/suchmuster	Es wird sofort zur ersten Zeilennummer mit dem suchmuster gewechselt
-wzeilen	Die Fenstergröße für den vi auf eine entsprechende Zeilenanzahl festlegen

Der eigentliche Text ist im Fenster zu sehen. Die "~" am Zeilenanfang kennzeichnet eine Leerzeile.

Man kann die Cursortasten verwenden um sich im Text bewegen zu können. Eine unmittelbare Eingabe an der Cursorposition (Einfügen von Text) ist nicht möglich.

Der vi kennt die beiden Modi:

- Eingabemodus - der Text kann geändert (Text einfügen, Text überschreiben) werden.
- Befehlsmodus - nimmt Befehle entgegen
- ex-Modus – komplexere Befehle können eingegeben werden (z.B.: reguläre Ausdrücke).



```
Sdflögsdfglösdjf glösdfk jgölsdfkjg lösdfkj glösdfkjg jlösdkfjg
lösdkf glösdkfj gölsdfkjg lösdkf jg

sdf gdfg
sdfg sdfg sd fg sdfgsd
~
~
~
:wq
```

Befehl	Bedeutung
--------	-----------

:q	Verlassen, ohne zu speichern (nur möglich wenn seit der letzten Änderung vorher gespeichert wurde).
:q!	Verlassen, ohne zu speichern (Änderungen seit der letzten Speicherung gehen verloren)
:wq	Speichern und beenden.
:x	Speichern und beenden.
ZZ	Speichern und beenden.

Zum Wechsel in den Eingabemodus gibt es folgende Befehle

Befehl	Bedeutung
i	Einfügen vor der aktuellen Position
I	Einfügen am Zeilenanfang
a	Einfügen nach der aktuellen Position
A	Einfügen am Zeilenende
o	Einfügen einer Zeile nach der aktuellen, Wechsel in die Zeile
O	Einfügen einer Zeile vor der aktuellen, Wechsel in die Zeile
R	Wechsel in den Überschreibmodus
r	Das nächste Zeichen löscht das Zeichen am Cursor, sofortige Rückkehr zum Befehlsmodus
cobjekt	Wechsel in den Eingabemodus bei Löschung von objekt, mögliche Objekte sind Wörter, Zeilen, Zeilenanfang oder -ende und viele mehr.

Der Löschbefehl c verwendet ein Argument, das unmittelbar nach dem c ohne Leerzeichen getrennt einzugeben ist. Dieses Argument beschreibt ein typisches Textobjekt wie Zeichen, Wort oder Zeile. Diese Objekte werden von einer ganzen Reihe von Befehlen benutzt - beispielsweise Lösch- oder Kopierbefehlen.

Kürzel	Objekt
Space	Einzelnes Zeichen
w	Wort
W	Wort mit Sonderzeichen
b	Wort rückwärts
B	Wort rückwärts mit Sonderzeichen
G	Bis zum Textende
^	Text bis zum Zeilenanfang
\$	Text bis zum Zeilenende
(	Text bis zum Satzanfang
)	Text bis zum Satzende
{	Text bis zum Anfang des Absatzes
}	Text bis Ende des Absatzes
[[	Text bis zum Abschnittsbeginn
]]	Text bis zum Abschnittsende

Beispiel:

c}

Löscht den Text von der Cursorposition bis zum Absatzende, um anschließend in den Eingabemodus zu wechseln.

Die Befehlsvielfalt macht den vi zu einem mächtigen, aber für Anfänger auch gefürchteten Werkzeug.

Für die Bearbeitung längerer Texte ist es wichtig sich im Text bewegen zu können.

Befehl	Bedeutung
k	Zeile nach oben (wie ↑)
j	Zeile nach unten (wie ↓)
h	Zeichen nach links (wie ←)
l	Zeichen nach rechts (wie →)
0 (Null)	Zum Zeilenanfang (erstes Zeichen)
^	Zum Zeilenanfang (erstes sichtbares Zeichen)
\$	Zum Zeilenende
w	Zum nächsten Wort in der Zeile
W	Zum nächsten durch Leerzeichen begrenzten Wort in der Zeile
b	Zum vorherigen Wort in der Zeile
B	Zum vorherigen durch Leerzeichen begrenzten Wort in der Zeile
G	Zum Dateiende
nrG	Zur Zeile nr
H	Zum Anfang des Bildschirms
M	Zur Mitte des Bildschirms
L	Zur letzten Bildschirmzeile
(	Zum vorigen Satz
)	Zum nächsten Satz
{	Zum vorhergehenden Absatz
}	Zum nächsten Absatz
[[	Zum vorhergehenden Abschnitt
]]	Zum nächsten Abschnitt
fzeichen	Zur nächsten Position von zeichen (vorwärts)
Fzeichen	Zur nächsten Position von zeichen (rückwärts)
tzeichen	Zur nächsten Position von zeichen (vorwärts), Cursor vor dem Zeichen positionieren
Tzeichen	Zur nächsten Position von zeichen (rückwärts), Cursor vor dem Zeichen positionieren
;	Befehl f, F, t oder T wiederholen
Strg F	Seite vorwärts blättern
Strg B	Seite rückwärts blättern
Strg D	Halbe Seite vorwärts blättern

Strg U	Halbe Seite rückwärts blättern
:nummer	Cursor springt an in die Zeile nummer

Editoren müssen neben der Texterfassung auch Texte verschieben, kopieren und löschen können.

Befehl	Bedeutung
x	Löscht das Zeichen am Cursor (Zeichen im Puffer)
X	Löscht das Zeichen vor dem Cursor (Zeichen im Puffer)
dobjekt	Löscht das Objekt (Objekt im Puffer)
dd	Löscht die ganze Zeile (Zeile im Puffer)
nummerdd	Löscht nummer Zeilen (Zeilen im Puffer)
yobjekt	Kopiert das Objekt (Objekt im Puffer)
yy	Kopiert ganz Zeile (Zeile im Puffer)
nummeryy	Kopiert nummer Zeilen (Zeilen im Puffer)
p	Fügt Pufferinhalt nach aktueller Zeile ein
P	Fügt Pufferinhalt vor aktueller Zeile ein

Eine weitere Funktionalität ist das Suchen und Ersetzen

Befehl	Bedeutung
/muster	Sucht nach muster im Text (vorwärts)
?muster	Sucht nach muster im Text (rückwärts)
n	Wiederholung der letzten Suche
N	Wiederholung der letzten Suche, aber in umgekehrter Richtung
%	Suche nach der zugehörigen Klammer (wichtiges Hilfsmittel zum Editieren von C-Quelltexten)

Geschriebene Texte lassen sich in den laufenden Text einfügen oder Textteile gesondert speichern.

Befehl	Bedeutung
:w [datei]	Speichert den Text in datei
:e datei	Laden der angegebenen Datei zum Editieren. Die bisherige wird aus dem Speicher entfernt.
:r datei	Einfügen der datei in den aktuellen Text
:n	Die nächste Datei wird in den Speicher geladen.

Befehl	Bedeutung
u	Hebt die letzte Änderung des Puffers auf
U	Hebt die letzte Änderung der aktuellen Zeile auf
J	Verbindet die aktuelle Zeile mit der folgenden Zeile

Strg R	Auffrischen des Bildschirms
!befehl	Ausführen eines beliebigen Linux Befehls
:set option wert	Ändern der vi internen Einstellungen
:map kürzel text	Definieren eines Makros (Tastenkürzel)
:unmap kürzel	Aufheben der Makrodefinitionen

Wiederherstellung nach einem Absturz:

Ein System stürzt ab, wenn die Hardware oder das System einen Fehler begeht, und muss dann wieder gestartet werden. Obwohl dies nicht häufig vorkommen sollte, schützt vi den Benutzer vor dem Verlust von Editierzeit, indem eine Kopie des Puffers in einer Datei abgelegt wird. Die sichere Kopie ist auch erhalten falls die Terminalleitung unterbrochen wird. Die Datei kann durch Gebrauch der Option -r beim Aufruf von vi zurückgewonnen werden.

```
vi -r datei.txt
```

Um die Zeilennummer vor jeder Zeile ausgeben zu lassen muss der folgende Befehl eingetippt werden.

```
:set number
```