

Editor vi

vi [*option(en)*] [-c *kdo*] [*datei(en)*]

- t *marke* (tag) Editieren der in *tags* mit *marke* festgelegten Datei
- r*datei* (recover) Editieren von *datei* nach Editor- bzw. Systemcrash.
- l Alle gereinigten Dateien nach System- bzw. Editorcrash melden
- wn (window) vi-Fenster sei *n* Zeilen
- R (Read only) *datei(en)* zum Lesen editieren
- c *kdo* *ex-kdo* vor dem Editieren ausführen

3 Arbeitszustände des vi

direkter Kommandomodus

Nach vi-Aufruf immer in diesem Modus. Umschalten in Eingabemodus mögl. mit:
 i, I, a, A insert bzw. append
 c, C, o, O change bzw. open
 s, S substitute
 R Replace

Eingabemodus

Im Eingabemodus ist Texteingabe möglich. Ende der Texteingabe und zurück zum direkten Kommandomodus mit ESC

Zeilen-Kommandomodus

Ein Umschalten vom direkten Kommandomodus in diesen Modus ist möglich mit:
 :*exkdo*-J *exkdo* (ed-ähnlich) ausführen
 /RA-J Vorwärtssuche nach RA
 ?RA-J Rückwärtssuche nach RA
 :kdo-J Unix-kdo ohne vi-Verlassen ausführen

Wichtige vi-Tasten

- ESC - Eingabemodus beenden
- im Zeilen-Modus: *exkdo* abbrechen
- im Zeilen-Modus: *exkdo* ausführen
- im Eingabemodus: neue Zeile
- im Kommandomodus: in nächste Zeile Aktion abbrechen.

Sondertasten im Eingabemodus

- Strg-I Tabulator-Zeichen einfügen
- Strg-T Auf nächste Tabulator-Markie positionieren
- Strg-V Nächstes Zeichen nicht als Kommando interpretieren
- Strg-W Kommando interpretieren
- Strg-H Letztes Wort löschen
- Strg-W Backspace Letztes Zeichen löschen

Interne vi-Puffer

Neben dem Arbeitspuffer gibt es 36 weitere Puffer:
 - 1 allg. Puffer, der letzte Textänderung enthält.
 - 26 benannte Puffer (a, b, c, ..., z): Großbuchstabe bedeutet Anhängen am jeweiligen Puffer
 - 9 Ziffernpuffer (1, 2, ..., 9), in denen die 9 zuletzt gelöschten Texte liegen (in 1 zuletzt gelöschter, in 2 der davor gelöschte Text, usw.).
 Die Puffer (außer allg. Puffer) können mit "x (*x* = Kleimbuchstabe bzw. Ziffer) angesprochen werden.

Sichern und Beenden

- ZZ Sichern und vi beenden
- :wq-J Sichern und vi beenden (wie ZZ)
- :q-J vi beenden, wenn Puffer nicht geändert
- :q!-J vi ohne Puffer-Sicherung beenden
- :w-J Puffer sichern (vi nicht verlassen)
- :w! *datei*-J Puffer in (evtl. existierende) *datei* sichern (vi nicht verlassen)
- :n, *nw* *datei*-J Zeilen *n* bis *m* in *datei* sichern (vi nicht verlassen)

Cursor-Positionierungen

- Zeichen-Positionierung
 l, →, *Blank* ein Zeichen nach rechts
 h, ←, *Strg-H* ein Zeichen nach links
 i, ↓, *Strg-N* nach unten (selbe Spalte / Zeilende)
 k, ↑, *Strg-P* nach oben (selbe Spalte / Zeilende)
 +, -J, *Strg-M* nach unten (zum 1. sichtbaren Zeichen)
 - nach oben (zum 1. sichtbaren Zeichen)
 ^ Zeilenanfang (1. sichtbares Zeichen)
 0 Zeilenende
 \$ zur *n*-ten Spalte
 n| zum nächsten Wort
 w zum *n*-ten nächsten Wort
 nW ein Wort zurück
 e zum Wortende

Zeilen-Positionierung

- H (Home) 1. Bildschirmzeile
- M (Middle) mittlere Bildschirmzeile
- L (Last) letzte Bildschirmzeile
- G letzte Zeile
- nG *n*-te Zeile (1G 1. Zeile)

Weitere Positionierungsmöglichkeiten

- PgDn, *Strg-f* 1 Seite vorblättern
- PgUp, *Strg-b* 1 Seite zurückblättern
- z-J akt. Zeile zur obersten Bildzeile machen
- z- akt. Zeile zur untersten Bildzeile machen
- z, zur mittleren Bildzeile machen
- z, zur vorherigen Position zurück
- % auf korrespondierende Klammer positionieren

Eingeben/Ändern im Eingabemodus

- a (append) Rechts vom Cursor einfügen
- A (Append) Am Zeilenende anhängen
- i (insert) Links vom Cursor einfügen
- I (Insert) Am Zeilenanfang einfügen
- o (open) In neuer Zeile danach einfügen
- O (Open) In neuer Zeile davor einfügen
- s (substitute) Cursorzeile ersetzen
- S (Substitute) Ganze Cursorzeile ersetzen
- R (Replace) Überschreiben einschalten
- cw (change) nächstes Wort ersetzen
- nCW nächste *n* Worte ersetzen
- cc ganze Zeile ersetzen
- C (Change) Rest der Zeile ersetzen

Ändern (nicht im Eingabemodus)

- rz (replace) Zeichen durch z ersetzen
- nRZ nächste *n* Zeichen durch z ersetzen
- ~ Zeichen von Klein- in Großbuchstaben umwandeln bzw. umgekehrt
- n~ nächste *n* Zeichen von Klein- in Großbuchst. umwand. bzw. umgekehrt
- J (Join) Zeilen zusammenfügen:
- nJ *n* Zeilen zusammenfügen
- . Letztes Änderungskommando wiederholen

Löschen, Kopieren und Verschieben

- dw (delete) nächstes Wort löschen
- nDw nächste *n* Worte löschen
- dd ganze aktuelle Zeile löschen
- nDD nächste *n* Zeilen löschen
- dWas Text bis zum nächsten *was* löschen
- dG ab Cursor bis Dateidele alles löschen
- D Zeilenrest löschen (entspricht d\$)
- x Zeichen an Cursorpos. löschen
- nX nächste *n* Zeichen löschen
- X Zeichen vor Cursor löschen
- yy kopiert akt. Zeile in allgemeinen Puffer
- nYY kopiert nächste *n* Zeilen in allgem. Puffer
- y\$ kopiert ab Cursor Zeilenrest in allg. Puffer
- >% Text bis korrespond. Klammer einrücken
- "yyw kopiert nächstes Wort in Puffer x
- "ayy kopiert aktuelle Zeile in Puffer a
- "xnyy kopiert nächste *n* Zeilen in Puffer x
- "add löscht akt. Zeile u. kopiert sie in Puffer a
- "xrdw *n* Worte löschen und in Puffer x kopieren
- p (put) allg. Puffer hinter Cursor kopieren
- xP vertauscht zwei Zeichen
- P (Put) allg. Puffer vor Cursor kopieren
- "xP Puffer x hinter Cursor kopieren
- "xP Puffer x vor Cursor kopieren
- "nP *n*-te letzte Löschung hinter Cursor kopieren (für *n* ist Ziffer 1 bis 9 anzugeben)
- :r *datei* *datei* hinter aktuelle Zeile kopieren

Änderungen rückgängig

- u (undo) macht die letzte Änderung rückgängig
- U (Undo) Änderung in akt. Zeile zurücknehmen
- :e! alle Änderungen seit letz. Sichern wegwerfen
- :q! vi ohne Sichern verlassen

Markieren und Positionieren

- mX Cursorpos. mit x (Kleimbuchst.) markieren
- 'x Cursor auf mit x markierte Stelle positionieren
- 'x Cursor auf 1. Zeichen der Zeile mit Marke x

Editieren mehrerer Dateien

- :e *datei*-J (edit) *datei* in Arbeitspuffer kopieren; alt. Puffer überschreiben, wenn bereits gesichert.
- :e#-J zurück zur vorherigen Datei (wie *Strg-^*)
- :n#J (next) nächste Datei (aus vi-Aufruf) editieren
- :n!J wie *n*-J, jedoch Puffer "blind" überschreiben
- :f *datei*-J Dateinamen in *datei* ändern. Ist *datei* nicht angegeben, wird gerade gemerkter Dateiname und Nummer der akt. Zeile angezeigt (wie *Strg-G*).

Suchen

- Suchen in aktueller Zeile
 fx Auf Zeichen x in aktueller Zeile zurück auf Zeichen x in akt. Zeile
- tx vor Zeichen x in akt. Zeile positionieren
- Tx zurück hinter Zeichen x in akt. Zeile
- ; letz. f, F, t- oder T-Kdo wiederholen
- , letz. f, F, t- oder T-Kdo wiederholen (mit umgekehrter Suchrichtung)

Suchen im Arbeitspuffer

- /RA-J Vorwärtssuche nach RA
- ?RA-J sucht (vorwärts) *n*-ten RA
- ?RA?n Rückwärtssuche nach RA
- /RA?n sucht (rückwärts) *n*-ten RA
- /RA/+n-J Cursor auf *n*-te Zeile nach gefund. RA
- /RA/-n-J Cursor auf *n*-te Zeile vor gefund. RA
- n letz. Suchvorgang (/RA oder ?RA) wiederholen
- N letz. Suchvorgang umgekehrt wiederholen
- RA (regulärer Ausdruck): siehe Ende dieses Kap.

Suchen und gleichzeitiges Ersetzen

- :s/alt/new/g-J in akt. Zeile 1. alt durch neu ersetzen
- :s/alt/new/g-J in Zeile alle alt durch neu ersetzen
- :n,ms/alt/new/g-J in der *n*-ten bis zur *m*-ten Zeile alle alt durch neu ersetzen.
- :1,\$s/alt/new/g-J im gesamten Arbeitspuffer alle alt durch neu ersetzen.
- :1,\$s/alt/new/gc-J im gesamten Arbeitspuffer alle alt durch neu ersetzen, jedoch mit Rückfrage: Soll ersetzt werden, muß y-J eingegeben werden, sonst keine Ersetzung.

Neuaufbau einer Bildschirmseite

- Strg-L, Strg-R Bildschirmseite neu aufbauen
- zn. Bildschirmgröße auf *n* Zeilen festlegen

Makros

- Bei Definition von Makros sind Steuerzeichen durch vorangestelltes Strg-V auszusprechen.
- @x Makro aus Puffer x aufrufen
- :ab *abk* *text*-J Textmakro *abk* für *text* definieren
- :ab-J alle definierten Textmakros anzeigen
- :map x *kdo*s-J Kdo-Makro x für *kdo*s definier.; x muß 1 Zeichen oder #*n* (*n*=0,...,9) sein;
- :map #1 -set *number*<*Strg-V*>-J.J Zeilennummerierung mit F1 einschalten
- :map-J alle definierten Kdo-Makros anzeigen

vi konfigurieren

- Eine Konfiguration ist mit set in *.exrc* (home dir.) oder :set *option*-J Einschalten von *option*
- :set *nooption*-J Ausschalten einer *option*
- :set *option*=wert-J Zuweisen von wert an *option*
- :set Alle geänderten Optionen anzeigen
- :set all-J Belegung aller Optionen anzeigen
- :set *option*?-J Belegung von *option* anzeigen
- :set *showmatch* korrespond. Klammer zeigen
- :set *showmode* Modus immer anzeigen
- :set *number* Zeilennummerierung einschalten