Der vi in Übersichten[1]

1 Start einer vi-Sitzung

Eingabe	Funktion
vi *file*	Editiere *file*
vi -r *file*	Editiere die letzte gerettete Version von *file* nach Systemabsturz
vi *file1 ... filen*	Editiere *file1* bis *filen*
vi +*n file*	Aufruf des vi für *file* mit Cursorpositionierung auf Zeile *n*
vi + *file*	Aufruf des vi für *file* mit Cursorpositionierung auf letzte Zeile
vi +/*str file*	Aufruf des vi für *file* mit Cursorpositionierung auf Zeile, die *str* enthält

2 Text sichern und vi verlassen

Eingabe	Funktion
ZZ oder	aktuelles File sichern und vi verlassen
:wq	
:q!	vi verlassen, ohne das aktuelle File zu sichern
:w	aktuelles File sichern
:w *file*	aktuelles File in *file* sichern
:w! *file*	Inhalt von *file* mit dem aktuellen File überschreiben
:*m,n***w** *file*	Zeile *m* bis Zeile *n* des aktuellen Files in *file* sichern

3 Eingeben von Text

Eingabe	Funktion
a	Text nach dem Cursor einfügen
A	Text am Ende der Zeile einfügen
i	Text vor dem Cursor einfügen
I	Text am Anfang der Zeile einfügen
o	unter dem Cursor Leerzeile für Texteingabe eröffnen
O	über dem Cursor Leerzeile für Texteingabe eröffnen
:r *file*	*file* lesen und nach der aktuellen Zeile einfügen
:*n***r** *file*	*file* lesen und nach der *n*-ten Zeile einfügen

[1]Diese Übersicht ist keine vollständige Syntax. Seltener benutzte Fähigkeiten wie Benutzung benannter Puffer, Abarbeiten von Unix–Kommandos, Textmarkierung u.ä. sind im Manual nachzulesen.

4 Rollen und Ausrichten des Bildschirminhalts

Eingabe	Funktion
<CTRL>-**f**	eine Seite vorwärts rollen
<CTRL>-**b**	eine Seite rückwärts rollen
<CTRL>-**d**	eine halbe Seite vorwärts rollen
<CTRL>-**u**	eine halbe Seite rückwärts rollen
z <RET>	aktuelle Zeile wird oberste Zeile des Bildschirms
z.	aktuelle Zeile wird zur mittleren Zeile
z	aktuelle Zeile wird zur untersten Zeile

5 Bewegen des Cursors in einem File

Eingabe	Funktion
← oder **h**	Cursor nach links bewegen
→ oder **l**	Cursor nach rechts bewegen
↓ oder **j**	Cursor nach unten bewegen
↑ oder **k**	Cursor nach oben bewegen
$	Cursor zum Ende der aktuellen Zeile bewegen
0 (Null)	Cursor zum Anfang der aktuellen Zeile bewegen
ˆ	Cursor zum ersten **sichtbaren** Zeichen der aktuellen Zeile bewegen
w	Cursor zum Anfang des nächsten Wortes bewegen
W	Cursor zum Anfang des nächsten Wortes bewegen (ohne Berücksichtigung von Sonderzeichen)
b	Cursor zum Anfang des vorhergehenden Wortes bewegen
B	Cursor zum Anfang des vorhergehenden Wortes bewegen (ohne Berücksichtigung von Sonderzeichen)
e	Cursor zum Ende des nächsten Wortes bewegen
E	Cursor zum Ende des nächsten Wortes bewegen (ohne Berücksichtigung von Sonderzeichen)
H	Cursor auf oberste Zeile des Bildschirms
M	Cursor auf mittlere Zeile des Bildschirms
L	Cursor auf unterste Zeile des Bildschirms
1G	Cursor auf 1. Zeile im File bewegen
*n***G**	Cursor auf *n*-te Zeile im File bewegen
G	Cursor auf letzte Zeile im File bewegen
(	Cursor zum Anfang des Satzes bewegen
)	Cursor zum Anfang des nächsten Satzes bewegen
{	Cursor zum Anfang des Absatzes bewegen
}	Cursor zum Anfang des nächsten Absatzes bewegen

6 Text löschen

Im vi-Kommandomodus:

Eingabe	Funktion
x	durch Cursor markiertes Zeichen löschen
*n***x**	*n* Zeichen ab Cursor löschen
dd	aktuelle Zeile löschen
*n***dd**	die nächsten *n* Zeilen löschen (einschl. der aktuellen)
dw	aktuelles Wort löschen (vom Cursor bis Wortende)
dW	aktuelles Wort einschl. Sonderzeichen löschen
*n***dw**	*n* Wörter ab markiertem Wort löschen
db	vorhergehendes Wort löschen
dB	vorhergehendes Wort einschl. Sonderzeichen löschen
*n***db**	*n* Wörter vor dem Cursor löschen
d$ oder **D**	aktuelle Zeile vom Cursor bis Zeilenende löschen
d0 (Null)	aktuelle Zeile vom Cursor bis Zeilenanfang löschen
d*cursor_cmd*	Text vom Cursor bis zu der durch *cursor_cmd* bestimmten Stelle löschen
:*m,n***d**	Zeilen *m* bis *n* löschen

Im vi-Eingabemodus:

Eingabe	Funktion
<CTRL>-**h** oder Backspace	im Eingabemodus vorhergehendes Zeichen löschen
<CTRL>-**w**	im Eingabemodus vorhergehendes Wort löschen
<CTRL>-**x**	im Eingabemodus zuletzt eingegebenen Text vollständig löschen

7 Ändern von Text

Eingabe	Funktion
r*char*	aktuelles Zeichen durch *char* ersetzen
R*text*	aktuelle(s) Zeichen durch *text* ersetzen
s*text*	aktuelles Zeichen durch *text* ersetzen
S*text* oder **cc***text*	aktuelle Zeile durch *text* ersetzen
cw*text*	aktuelles Wort durch *text* ersetzen
*n***cw***text*	*n* Wörter ab Cursor durch *text* ersetzen
C*text*	aktuelle Zeile von Cursor bis Zeilenende durch *text* ersetzen
c*cursor_cmd text*	Text von Cursor bis *cursor_cmd* durch *text* ersetzen
~	aktuelles Zeichen umschalten groß/klein
.	letztes vi–Kommando wiederholen

8 Änderungen zurücksetzen

Eingabe	Funktion
u	**zuletzt** ausgeführten Befehl rückgängig machen
U	Zustand der **aktuellen** Zeile vor der Änderung herstellen
:q!	vi-Sitzung ohne Sicherung der Änderungen verlassen

9 Text kopieren oder verschieben

Eingabe	Funktion
Y oder **yy**	aktuelle Zeile zwischenspeichern
*n***Y** oder *n***yy**	*n* Zeilen ab Cursor zwischenspeichern
y*cursor_cmd*	Text von Cursor bis *cursor_cmd* zwischenspeichern
dd	aktuelle Zeile löschen
*n***dd**	*n* Zeilen ab Cursor löschen
d*cursor_cmd*	Text von Cursor bis *cursor_cmd* löschen
p	gelöschten oder zwischengespeicherten Text **hinter** (oder unter) dem Cursor einfügen
P	gelöschten oder zwischengespeicherten Text **vor** (oder über) dem Cursor einfügen
xp	zwei Zeichen vertauschen
J	zwei Zeilen zusammenfügen
*n***J**	*n* Zeilen zusammenfügen

10 Globales Suchen und Ersetzen von Text

Eingabe	Funktion
:*m,n***s**/*str1*/*str2*/	von Zeile *m* bis Zeile *n* wird *str1* durch *str2* ersetzt
:*m,n***s**/*str1*/*str2*/g	von Zeile *m* bis Zeile *n* werden alle *str1* durch *str2* ersetzt
&	das letzte **:s**-Kommando wird wiederholt
:g/*str*/*cmd*	*cmd* wird für alle Zeilen ausgeführt, die *str* enthalten
:v/*str*/*cmd*	*cmd* wird für alle Zeilen ausgeführt, die *str* **nicht** enthalten

11 Suchen im Text

Eingabe	Funktion
f*char*	vorwärts suchen von *char* in der aktuellen Zeile
F*char*	rückwärts suchen von *char* in der aktuellen Zeile
/*str* <RET>	vorwärts suchen von *str* ab der aktuellen Zeile
?*str* <RET>	rückwärts suchen von *str* ab der aktuellen Zeile
n	nächstes Muster (vorwärts) suchen