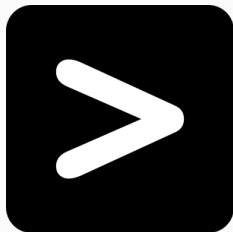


Einführung in vi(m)

Ist gar nicht so schlimm ...



Dirk Deimeke

26. November 2016

My own IT / LinuxDay 2016

Prolog

Dirk Deimeke – d5e.org

Warum vim?

**Anders gefragt:
Warum seid Ihr hier?**

Warum vim?

- Ist auf (nahezu) jedem Linux- und Unix-System – inklusive Mac OS X – vorinstalliert oder kann sehr einfach nachinstalliert werden.
- vim ist ein unglaublich mächtiger Texteditor.
Ich selber kratze auch nur an der Oberfläche, selbst das ist schon beeindruckend.

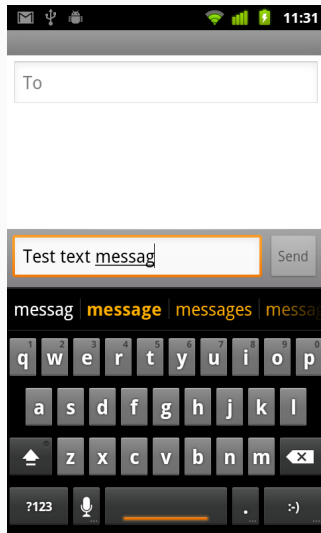
- Je nach Distribution, `vim-enhanced` oder `vim-full` installieren.
- Das Übersetzen aus den Quellen ist auch nicht so schwer.
- Es lohnt sich mit einer Beispielkonfigurationsdatei einzusteigen.
`cp /usr/share/vim/vim74/vimrc_example.vim ~/.vimrc`
(Suchen mit `find /usr/share/vim -name '*vimrc*'`.)

vim



- Ursprungsversion stammt aus 1976
- Terminals damals hatten keine Cursor-Tasten
- vi improved (vim) seit 1991
(25 Jahre alt, so alt wie Linux)
- Im Folgenden geht es um vim,
da kaum noch der Ursprungs-vi installiert wird

Terminals «damals» hatten keine Cursor-Tasten



<https://www.flickr.com/photos/isriya/5365993639/>

Achtung!

vim tippen, wenn Ihr vim meint!

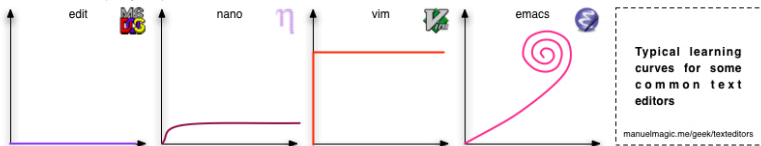
vi ist häufig ein alias für vim, der vim nur mit einem sehr eingeschränkten, veralteten Funktionsumfang startet.

Lernkurven

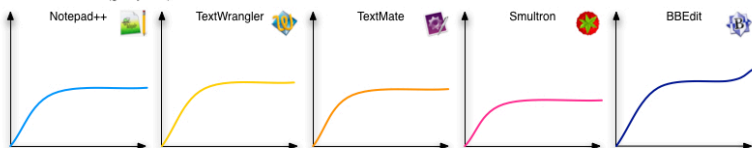
System default editors



Terminal editors (nerdy stuff)



Freeware editors (geeky stuff)



«Modaler» Editor

vim ist ein *modaler Editor*.

Anders als andere Editoren, die sich ständig in einem Textbearbeitungsmodus befinden, unterscheidet vim zwischen zwölf Modi, in denen unterschiedliche Befehle und Verfahren zur Verfügung stehen, mit Text umzugehen.

<https://vimhelp.appspot.com/intro.txt.html#vim-modes>

Bearbeitungsmodi

Im Folgenden nur auf vier Modi ein und dort auch nur auf Kommandos, die ich im täglichen Gebrauch nutze.

Normal Befehle haben einzelne Buchstaben oder Buchstabenkombinationen

Wechsel in den Normalmodus mit Escape

Command Suchen und Ersetzen, grössere Manipulationen, Speichern, Verlassen, externe Kommandos

Wechsel in den Normalmodus mit «:»

Insertmodus Bearbeiten von Text

Wechsel in den Edit-Modus mit i, I, a, A, o, O, r, R

Visual Visuelles Auswählen

Wechsel in den Visual Modus mit v, V

Starten von vim

Starten

vim Datei Startet vim mit der Datei

vim +123 Datei Startet vi mit der Datei und positioniert sich auf Zeile 123

vim +'syn on' Datei Startet mit eingeschaltetem Syntax-Highlighting

vim +'set nu' Datei Startet mit eingeschalteter Zeilennummerierung

vim +"cmd1"+"cmd2"Datei Startet mit cmd1 und cmd2

Starten und Beenden von vi(m)

Beenden aus dem Normalmodus

- :q** Beendet vi
- :q!** Beendet vi, ignoriert Änderungen in aktueller Datei
- :qa!** Beendet vi, ignoriert Änderungen in allen offenen Dateien
- :wq** Analog oben, mit Schreiben der aktuellen Datei
- ZZ** Beenden mit Schreiben der Datei, wenn sie sich geändert hat

i, I «insert»

Einfügen an aktueller Position / Anfang der Zeile

a, A «append»

Einfügen nach aktueller Position / Ende der Zeile

r, R «replace»

Ersetzen eines Buchstabens / Ersetzen und Editieren

o, O Leerzeile nach / vor aktueller Zeile einführen und editieren

Normalmodus: Undo / Redo

u Undo

Ctrl + r Redo

Normalmodus: Bewegen im Text

h, j, k, l Links, Runter, Hoch, Rechts

(,) Anfang / Ende des Satzes

{, } Anfang / Ende des Absatzes

w, b, e Wort vorwärts / rückwärts / vorwärts auf Wortende
(basiert auf Buchstaben)

W, B, E Wort vorwärts / rückwärts / vorwärts auf Wortende
(basiert auf druckbaren Zeichen)

0, \$ Anfang / Ende der Zeile

gg, G, 1G Anfang / Ende des Textes (#G Springe zur Zeile #)

H, M, L Erste Bildschirmzeile, Mitte, letzte Bildschirmzeile

Die Macht der Zahl

Mehrfaches Ausführen

Durch Voranstellen einer Zahl wird nahezu jedes Kommando mehrfach ausgeführt. «3j» springt drei Zeilen nach unten.

Normalmodus: Scrollen

Ctrl + e Cursor bleibt, Text scrollt nach oben

Ctrl + y Cursor bleibt, Text scrollt nach unten

zt Aktuelle Zeile wird erste Zeile der Bildschirmseite

zz Aktuelle Zeile wird mittlere Zeile

zb Aktuelle Zeile wird untere Zeile

Normalmodus: Seitenweise bewegen

Ctrl + b Back – Seite rauf

Ctrl + f Forward – Seite runter

Ctrl + u Up – halbe Seite rauf

Ctrl + d Down – halbe Seite runter

Es wird noch schlimmer ...

version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Esc
normal
mode

~ toggle case	! external filter	@ play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat is	* next ident	(begin sentence	) end sentence	"soft" bol down	+ next line
. goto mark	1	2	3	4	5	6	7	8	9	0 "hard" bol	- prev line	= auto format
Q ex mode	W next word	E end word	R replace mode	T back 'till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.
q record macro	w next word	e end word	r replace char	t 'till	y yank	u undo	i insert mode	o open below	p paste after	* misc	*	misc
A append at eol	S subst line	D delete to eol	F "back" find ch	G eof/ goto ln	H screen top	J join lines	K help	L screen bottom	. ex cmd	" reg. spec	bol/ goto col	
a append	s subst char	d delete	f find char	g extra cmds	h ←	j ↓	k ↑	l →	* repeat 1/T/t/F	' goto mk. bol	\ not used!	
Z quit	X back-space	C change to col	V visual lines	B prev WORD	N prev (find)	M screen mid1	< un-indent	> indent	? find (rev.)			
Z extra cmds	x delete char	c change	V visual mode	b prev word	n next (find)	m set mark	reverse 1/T/t/F	* repeat cmd	/ find			

motion	moves the cursor, or defines the range for an operator
command	direct action command, if red , it enters insert mode
operator	requires a motion afterwards, operates between cursor & destination
extra	special functions, requires extra input
Q	commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line, mk = mark, yank = copy

words: `quux(foo, bar, baz)`
WORDS: `quux(foo, bar, baz)`

Main command line commands ('ex'):

:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:s/s/x/y/g (replace 's' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important commands:

CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:

Move around and type operator to act on selected region (vim only)

Notes:

- (1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay\$ to copy rest of line to reg 'a')
- (2) type in a number before any action to repeat it that number of times (e.g.: 2p, dzw, 5i, daj)
- (3) duplicate operator to act on current line (dd = delete line, >> = indent line)
- (4) ZZ to save & quit, ZQ to quit w/o saving
- (5) zt: scroll cursor to top, zb: bottom, zz: center
- (6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to www.viemu.com - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

<http://www.viemu.com/vi-vim-cheat-sheet.gif>

Normalmodus: Kopieren, Löschen und Ändern (1)

y, yy, yw, y\$ «yank» Kopieren von Inhalten in den Puffer
(ganze Zeile / Wort ab hier / Zeile ab hier)

d, dd, dw, d\$ «delete» Löschen von Inhalten
(verschieben in den Puffer)

c, cc, cw, c\$ «change» Ändern von Inhalten
(verschieben in den Puffer und Einfügemodus)

Normalmodus: Kopieren, Löschen und Ändern (2)

yG, dG, cG ab hier bis zum Ende der Datei

y1G, d1G, c1G von der ersten Zeile bis hier

yj, dj, cj diese und die nächste Zeile

yk, dk, ck diese und die vorherige Zeile

- dap** Lösche aktuellen Absatz
(geht auch mit «c» = change oder «y»)
- >ip** Rücke aktuellen Absatz ein
- di(** Lösche innerhalb der Klammern¹
- da(** Lösche innerhalb und die Klammern

¹Alle Klammern – ([{<>}]) – können verwendet werden.

Normalmodus: Löschen und Einfügen

p, P «paste» Einfügen aus dem Puffer in den Text ab /
vor aktueller Position

x Löscht das Zeichen unter dem Cursor

xp tauscht zwei Zeichen

«3dd» löscht die nächsten drei Zeilen. Was macht «5dw»?

Normalmodus: Einrücken

>, < Einrücken / ausrücken

>>, << Einrücken / ausrücken aktuelle Zeile

4>>, **4<<** Vier Mal Einrücken / ausrücken

- . Letzte Textveränderung wiederholen

Normalmodus: Suchbegriff suchen

/begriff Vorwärts suchen

?begriff Rückwärts suchen

n Zum nächsten Treffer (gleiche Suchrichtung)

N Suchrichtung umdrehen

Ctrl + o Zum letzten Treffer

Ctrl + i zum nächsten Treffer

Gilt nur für die aktuelle Zeile!

- f**, Springe auf nächstes «,»
- F**. Springe auf vorherigen «.»
- t**) Springe vor nächste «)»
- T**(Springe auf vorherige «(»
 - ; Wiederhole in Suchrichtung
 - , Wiederhole in Gegenrichtung

Normalmodus: Manpage finden

- K** Findet manpage zum Kommando, auf dem der Cursor steht.

Kommandomodus: Ersetzen

:%s/alt/neu/g im gesamten Text alt durch neu ersetzen

:%s/alt/neu/gc im gesamten Text alt durch neu nach
Bestätigung ersetzen

:z1,z2s/alt/neu/ alt durch neu in den Zeilen z1 bis z2 ersetzen

Kommandomodus: Teilen des Bildschirms

:split Splittet in einen oberen und unteren Bereich

:vsplit Splittet in einen linken und rechten Bereich

Ctrl + w, danach Ctrl + w Wechselt zwischen den Bereichen

Ctrl + w, danach h, j, k, l Bereich links, unten, oben, rechts

Konfigurationsdatei

Kommandos, die mit `set` beginnen, können so wie beschrieben – allerdings ohne Doppelpunkt – in die Konfigurationsdatei übernommen werden.

Kommandomodus: «Unsichtbare» Zeichen

:set list Zeigt «unsichtbare» Zeichen an.

Kommandomodus: Leerzeichen

:set tabstop=4 Ein Tab steht für vier Zeichen.

:set softtabstop=4 Tab im Einfügemodus vier Zeichen.

:set shiftwidth=4 Standard-Einrückung vier Zeichen.

:set expandtab Ersetzt Tabs durch Leerzeichen.

:set noexpandtab Nimmt das wieder zurück.

:retab! Formatiert den Text nach den oben angegebenen Regeln.

default :set ts=8 sts=0 sw=8 noexpandtab

v markieren

V Zeilenweise markieren

Ctrl + v Spaltenblock markieren

Visueller Modus – Beispielkommandos

>, **<** Einrücken / Einrückung entfernen

y kopieren

d löschen

~ Gross- und Kleinschreibung tauschen

:s/alt/neu/ Suchen und Ersetzen

Kommandomodus: Syntax Highlighting



:syntax on

<https://twitter.com/nixcraft/status/505189951844417538>

Kommandomodus: Zeilennummern

:set nu Zeilennummerierung einschalten

:set nonu Zeilennummerierung ausschalten

Kommandomodus: Automatisches Einrücken

:set ai «autoindent» einschalten

:set noai automatisches Einrücken ausschalten

Epilog

Fragen?

Vielen Dank!

Dirk Deimeke, 2016, CC-BY

dirk@deimeke.net

d5e.org – speakerdeck.com/ddeimeke

PDF bei Speakerdeck herunterladen, dann sind die Links klickbar.

- vim Homepage (Bram Moolenaar)
 - `:help` in vim
 - `vimtutor` ausführen
- Kleine Kommandos (Sven Guckes)
- `vimrc.minimal` (Sven Guckes)
- 68 kurze Screencasts (Drew Neil)