Vim für Nicht-Mehr-Beginner und Noch-Nicht-Fortgeschrittene

Agenda

- vimtutor auffrischen
- Einführung in das Hilfesystem
- vim != vi
- Bewegen und Editieren in Dateien
- .vimrc/.viminfo
- Shortcuts
- Makros
- Tabs/visual mode/Skeletons
- noch Fragen?

Aus dem Leben eines SysAdmins

- Konfigurationsdateien editieren
  → vim
- Log-Dateien durchstöbern
  → vim -R
- Dokumentation schreiben
  → Wiki → firefox → it's all text → (g)vim
- Python/Bash-Scripte
  → vim
- Mailing
  → mutt → vim

Vorbereitung

- Wiederholung
  - modaler editor
  - spezielle „Befehle“ zum Navigieren/Editieren
  - <Esc> is your friend (or <C-c>)

- vimtutor anyone?
  - Definition: vimtutor durchgespielt == Nicht-Mehr-Beginner

vimtutor (1)

- hjkl – zum Bewegen
- :q! – zum Beenden
- x – löscht ein Zeichen
- i – zum Einfuegen (insert)
- A – zum Anhaengen ans Zeilenende (append)
- :wq – zum Speichern und Schliessen (write/quit)

vimtutor (2)

- dw – ein Wort loeschen
- d$ – bis zum Ende einer Zeile loeschen
- w – zum Anfang des naechsten Wortes bewegen
- e – zum Ende eines Wortes bewegen
- $ – zum Ende der aktuellen Zeile bewegen
- 2w – zum Anfang des übernächsten Wortes bewegen
- 3e – zum Ende des dritten Wortes bewegen
- 0 – zum Anfang der Zeile bewegen
- d2w – lösche die folgenden zwei Wörter
- dd – lösche aktuelle Zeile
- 2dd – lösche zwei Zeilen
- u – mache Änderung rückgängig
- U – mache Änderungen in einer ganzen Zeile rückgängig
- <C>-r – stelle Änderung wieder her

vimtutor (3)

- p – hänge zuvor gelöschten Text nach dem Cursor an (put)
- r – ersetze (replace) Zeichen unter Cursor
- ce – ändere (change) bis zum Ende des aktuellen Wortes
- c$ – ändere Zeile bis zum Zeilenende

- <C-g> – zeige Dateiposition/-status an
- G – gehe zum Dateiende
- gg – gehe zum Dateianfang
- 3G – gehe an den Anfang der dritten Zeile

vimtutor (4)

- / – suche vorwärts
- ? – suche rückwärts
- n – gehe zum nächsten Treffer der Suche
- N – gehe zum vorherigen Treffer der Suche
- <C-o> – gehe zum vorherigen "Standort" des Cursors
- <C-I> – gehe wieder „vor“
- % – gehe zur zugehörigen Klammer ({[

vimtutor (5)

- :%s/alt/neu/gc – ersetze jedes "alt" durch "neu" in allen Zeilen, frage aber vorher nach
- :!ls – fuehrt den Befehl "ls" aus
- :w name – speichert aktuelle Datei unter "name" ab (ggf. nur per visual mode ausgewaehlten Bereich)
- :r name – liest Inhalt der Datei "name" in aktuelle Datei ein
- :r!ls – schreibt Ausgabe des Befehls "ls" in aktuelle Datei

vimtutor (6)

- o – fuege neue Zeile unter aktueller Position ein
- O – fuege neue Zeile ueber aktueller Position ein
- a – fuege Text nach dem Cursor an (attach)
- A – fuege Text nach dem Ende der Zeile an
- e – springe zum Wortende
- y – kopiere ("yank") Text
- p – fuege kopierten Text ein (put)
- R – ueberschreibe Text bis zum Beenden des Insert-Mode
- :set bla setze Option bla
- :set nobla deaktiviere Option bla (Bsp: ic, is, hls)

Help! (1)

- :help bzw. :h bzw. <F1> (wenn das Terminal das erlaubt)
- :h command/option/key
- jump via tags: <C-]> (zurueck mit <C-o>)
- :h help-summary
- :h i_CTRL-R»(listet Hilfe zu <C-r> im Insert-Mode auf
- :help hls <C-d> (mit Tab durch die Liste wandern)

Help! (2 - windows)

- :help windows
- <C-w><C-w> lässt Hilfe offen und springt in ursprüngliches Fenster
- <C-w>_ maximiert aktuelles Fenster
- <C-w>= bringt beide Fenster auf gleiche Größe
- :q schliesst aktuelles Fenster wieder

Help! (3) - Übung

- Hilfe zum Thema xyz Starten
- zu einem anderen |Thema| navigieren
- Window auf „fullscreen“ vergrößern
- :help! → was bedeutet die Fehlermeldung?
- Hilfe schließen

vim != vi

- (nearly) compat mode
- vim nicht auf allen Plattformen verfügbar
- kein Syntax highlighting, keine Hilfe, kein undo-Stack, kein visual mode, keine Macros
- :h vi_diff

Bewegen (1)

- hjkl UND cursor-Tasten
- gg/G/20G/20gg/:20
- / und ? (suchen)
- n/N (nächster/vorheriger Treffer)
- w/e/b
- W/E/B
- 0/^/$
- H/M/L
- */#

Bewegen (2) - scrolling

- <C-e> / <C-y>
- zt (top)
- zb (bottom)
- zz bzw. z.
- <C-u> / <C-d>
- :set scroll

Bewegen (3) - marker

- mx - setze Marke mit dem "Namen" x
- 'x - gehe in die Zeile, in der die Marke x gesetzt wurde
- `x - gehe direkt an die Stelle, an der die Marke gesetzt wurde
- `` - springe zwischen aktueller Marke und letzter Postion hin und her
- `. - springe zur letzten Aenderung
- :marks

Editieren (1) – copy'n'paste

- y / yy
- ywP
- yyp
- x
- xp
- P
- :reg

Editieren (2) – text objects

- aw / iw
- a“ / i“
- a' / i'
- a(/ i(
- at / it
- ap / ip
- as / is
- :help text-objects

Editieren (3) – text object combos

- d + w
- d + iw
- d + aw
- c + w
- c + iw
- ...

Editieren (4) – misc

- <C-p> / <C-n> – Wort vervollständigen
- <C-x><C-l> – Zeile vervollständigen
- <C-a> / <C-x> – hoch-/runterzaehlen
  - Obacht bei fuehrender Null!
- :read filename
- :read !command
- :!%
- :%s/alt/neu/gc
- :set paste

Editieren (5) – Übung

- Gedicht in die richtige Reihenfolge bringen

  purr, purr. purr,

  ball fur. little of

  kitty, kitty, soft warm

  happy sleepy kitty, kitty,

.vimrc

- :set hlsearch → set hlsearch (in .vimrc kein führender Doppelpunkt!)
- :set number → set number
- :set … → set ...
- vimscript
- Kommentar beginnt mit “

.viminfo

- „history“ (Suchbegriffe, editierte Dateien, „command line“, Register-Inhalte, Markierungen, Makros...)
- ggf. Datei loeschen bei sensiblen Inhalten

Abkürzungen

- :ab fa Feierabend
  - wirkt im Insert-Mode UND in ex-command line
- :iab fa Feierabend
  - wirkt nur im Insert-Mode
- wirkt auch bei copy'n'paste via MMB („middle mouse button“)!
- :iab HAllo Hallo

Mappings

- :map <F4> ddp
- :map <F5> ddkP
- :map V IViele Grüße<CR><CR><Esc>
  - V ist aber durch vim bereits belegt!
- :map <Leader>V IViele Grüße<CR><CR><Esc>

- :let mapleader=“,“
- :nmap/vmap/imap (normal, visual, insert mode)

Abk./Mappings – Übung

- Schreibe folgende „Mail“ und lasse die Abk. durch den „richtigen“ Text ersetzen:

  Hallo,

  hab Dich tel. nicht erreicht, d. h. b. a. W. ist der WLAN-AP für die VoiP-Telko n. v.

  mfg

Makros

- Arbeitsschritte zur Wdhlg. aufzeichnen
- :reg
- qx – beginne mit Aufzeichnung und speichere in Register 'x'
- <Esc> ggf. zum Beenden Insert-Mode
- q – beende Aufzeichnung
- @x – „Abspielen“ des Makros 'x'
- Q → :vi

Makros – Übung

- erzeuge „config file“:
  - fünf Rechnernamen (hostname1, hostname2, hostnameX) in folgendes Schema pressen:

    [workstation;hostnameX]

    address hostnameX

    use_node_name yes

Tabs

- vim -p file1 file2 file3
- gt/gT – geh zum nächsten/vorherigen Tab
- 2gt – geh zum zweiten Tab
- :tabnew – öffne neuen unbenannten Tab
- :tabe blub – öffne neuen Tab mit Datei „blub“
- :tab help gt – öffne Hilfe zu „gt“ in neuem Tab

Visual mode

- v – starte visual mode (s. Status-Zeile)
- V – starte visual mode zeilenweise
- <Esc> – beende visual mode
- gv – markiere letzten Bereich
- :help visual-operators
- :help Visual

Visual block mode

- <C-v> – starte visual block mode (s. Status-Zeile)
- beliebige Bewegung um Block zu markieren
- o – gehe an das andere Ende des Blocks
- O – gehe an das andere Ende des Blocks auf gleicher Zeile
- Zeilenende ($) wird "intelligent" erkannt/angepasst
- I (insert zu Beginn des Blocks) – wird auf markierten Bereich angewandt
- A (append ans Ende des Blocks) – ebenso

Visual mode – Übung

- IMGP001.jpg 10x duplizieren und hochzählen
- anschl. in „html“-Format bringen:

  IMGP001.jpg

  IMGP002.jpg

  ...

Visual (block) mode – Übung

- Output von „cal“ einlesen:
  - :read !cal -h
- „pretty“ format:

```
    ------------------------------------
    |    Juni 2015                     |
    ------------------------------------
    | So | Mo | Di | Mi | Do | Fr | Sa |
    |    |  1 |  2 |  3 |  4 |  5 |  6 |
    |  7 |  8 |  9 | 10 | 11 | 12 | 13 |
    | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
    | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
    | 28 | 29 | 30 |    |    |    |    |
    ------------------------------------
```

Skeletons

- „templates“ für neu erzeugte Dateien
- :help skeleton