

# Using the power of texttools - texttools++

Michael Prokop und Sven Guckes / GrazerLinuxTage 2004

[michael@linuxtage.at](mailto:michael@linuxtage.at) / [www.michael-prokop.at](http://www.michael-prokop.at) + [sven@guckes.net](mailto:sven@guckes.net) / [www.guckes.net](http://www.guckes.net)



Dieses Dokument steht unter der GNU Free Documentation License



# Inhalt

Einleitung

● Inhalt

Screen

Vim

Mutt

Slrn

Zsh

The End...

- Screen - Terminalmanager
- Vim - Texteditor
- Mutt - Mailprogramm
- Slrn - Newsreader
- Zsh - Shell
- Q&A - "Fragen"?



# Features

Einleitung

Screen

● Features

● Features

● Kommandos

Vim

Mutt

Slrn

Zsh

The End...

- screen ist ein "window manager" für die Konsole und Terminals.
- Es gibt einen sog. "scrollback history buffer" für jedes virtuelle Terminal/Fenster
- damit ist "make \$\* 2>&1 | less" nicht mehr notwendig :-)
- Copy/Paste-Modus - Text austauschen zwischen Fenstern → keine Maus notwendig! Maus kann fehlen, kaputt sein..
- Events: Monitor und Silence - meldet Aktivität bzw. Nicht-Aktivität (seit N Sekunden) in einem Fenster
- Hardcopy → Screenshot



# Features

Einleitung

Screen

● Features

● Features

● Kommandos

Vim

Mutt

Slrn

Zsh

The End...

- Logging → mitloggen in einem Fenster
- Titel → Titel der Fenster sind wählbar
- Session Locking → Sicherheit in Abwesenheit (es pfuscht niemand rein wenn man Kaffee holt ;-)
- Window Backlog → zum Nachlesen und für vergessliche Leute ;-)
- Digraph Mode → Eingabe von Zeichen, die nicht auf der Tastatur sind.
- Key Binding → Zuordnung von Tasten zu Kommandos ®  
nützlich bei ungewohnter Tastatur und bei Problemen mit Tastenbelegung, Aufruf von speziellen Programmen z.B. vom Webbrowser auf Webseiten



# Kommandos

Einleitung

Screen

● Features

● Features

● Kommandos

Vim

Mutt

Slrn

Zsh

The End...

- `screen -help` → Hilfe zu Screen anzeigen
- `screen -ls` → Screen-Sessions anzeigen
- `screen` hört auf `Ctrl-A` aka "Â" → neues Terminal
- `Ctrl-A a` → Wechsel zwischen 2 Terminals
- `Ctrl-A d` → "detach" Verbindung zum Terminal lösen
- `screen -r` → "reattach" - die einzige detachte Session wieder an das aktuelle Terminal binden
- `screen -d -r` → einzige attachte Session vom Terminal lösen und an das aktuelle Terminal binden



# Fahrplan

Einleitung

Screen

Vim

● Fahrplan

- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

- Marken setzen, Text auswählen im visual mode, kopieren, löschen
- Programmieren: Marken setzen, Kompilieren, zu Fehlern springen.
- Suchen+Ersetzen → Substituieren ("substitute" mit ":s")
- Muster = Reguläre Ausdrücke (regular expressions aka "regex")
- Abkürzungen und Macros
- incremental search, tab completion
- filetypes, syntax highlighting



# Macros/Recording

Einleitung

Screen

Vim

● Fahrplan

● Macros/Recording

● Formatieren von Text

● Falten / Folding

● Automatisierungen

● Fenster

● Buffers

● Programmieren mit Vim

● Nette Kommandos

● Goodies

● Vervollständ., Suche, Markieren

● Histories und Eingebaute Hilfe

● Probleme? Fragen richtig stellen!

● Ressourcen

● Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

- Register (a-z) "Clipboard"

- Aufnahme: 'q' + [a-z]

- Abspielen: 'q' + '@' + [a-z]

- Beispiel: Unterstreichen von Text:

qa	-> recording aktivieren
Yp	-> Zeile kopieren
:s,.,-,g	-> durch »-« ersetzen
q	-> recording beenden
@a	-> recording abspielen

- Geht auch sichtbar (z.B. per Skript):

```
:let @a="Yp:s/./-/g \<CR>"
```

- siehe FAQ 22.2.



# Formatieren von Text

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- **Formatieren von Text**
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

- vip + gq (erst Absatz visuell auswählen, und dann umbrechen)
  - gqip (Umbrechen des aktuellen Absatzes)
  - map <c-j> gqip (Macro für Ctrl-J)
- Nebeneffekte durch Optionen:
- set autoindent cindent shiftwidth textwidth



# Falten / Folding

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

- Zusammenfalten von Blöcken aufeinanderfolgender Zeilen
- Anwendung bei Quellcode, komplexen Dateien und Konfigurationsdateien
- erst Markieren im visual mode, dann Falten mit `zc` (close), Öffnen mit `zo` (open)
- Falten mit Hilfe von Markern:
- `set foldmarker=`, `foldmethod=marker`
- nur Information **über den** und nicht **im Text**? → `mkview`



# Automatisierungen

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

## ■ Template bei C-Dateien automatisch einlesen:

```
:autocmd BufNewFile *.*[ch] r template
```

<pre>:autocmd</pre>	-> automatisches Ausführen von...
<pre>BufNewFile</pre>	-> Event bei neuem Buffer
<pre>*.*[ch]</pre>	-> bei Suffix (Dateiendung)
<pre>r</pre>	-> einfügen [read]
<pre>template</pre>	-> einzufügende Datei

## ■ Plugins = vim-Skripte in \$HOME/.vim/plugin - werden beim Start automatisch ausgeführt



# Fenster

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

Alle Fenster-Kommandos beginnen mit 'Ctrl-W'  
gefolgt von einem weiteren Zeichen:

- s - split horizontal
- v - split vertikal
- \_ (Unterstrich / underscore) max. vergrößern horizontal
- | (Senkrechtstrich / pipe ) max. vergrößern vertikal
- = alle Fenster auf dieselbe Größe angleichen
- Cursortasten → Fensterwechsel



# Buffers

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

Ein Buffer ist ein Zwischenspeicher für Text, in dem dieser Text bearbeitet werden kann. Der Inhalt eines Buffers kann in eine Datei geschrieben, bzw. kann der Inhalt einer Datei in einen Buffer geladen werden.

- `:n` - "next" (Vi Befehl!)
- `:rew` - "rewind" (Vi Befehl!)
- `:bn + :bp` ("buffer next/previous")
- `:ls` ("list")
- `:b N` - zum Buffer *N* wechseln
- `:b substring<TAB>`



# Programmieren mit Vim

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

- :make
  - :cn (next error), :cp (previous error)
  - ctags[-exuberant]: "ctags \*.c" oder "ctags -R ."
  - Ctrl-] springt zu Funktionsdefinition,
  - Ctrl-T springt wieder zurück
  - scope: Erweiterung auf Variable usw.
- Achtung: das cscope-Interface ist per default nicht installiert, sondern muss selbst installiert werden!



# Nette Kommandos

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- **Nette Kommandos**
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

Kommandozeile (ex-Modus):

- `:%s/\s\+regex$/` -> Regex + Leerzeichen davor löschen
- `:%s/\s\+$/` -> Leerzeichen am Zeilenende entfernen
- `:%s/\s\+ / /g` -> "Whitespace runs" zu einzelner Leerzeichen
- `:%>` -> den ganzen Buffer einrücken (Methode #1)

Kommandomodus:

- `>ip` -> Block um "shiftwidth" einrücken
- `gg=G` -> den ganzen Buffer einrücken (Methode #2)
- `ggip` -> Absatz an aktuelle Textbreite anpassen
- `==` -> aktuelle Zeile mit 'equalprg' filtern

Insertmodus:

- `Ctrl+D` -> Zeile ausrücken
- `Ctrl+T` -> Zeile einrücken
- `Ctrl+E` -> Zeichen aus nächster Zeile löschen
- `Ctrl+Y` -> Zeichen aus vorheriger Zeile löschen



# Goodies

Einleitung

---

Screen

---

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

---

Slrn

---

Zsh

---

The End...

---

- :options
  - :registers
  - :buffers
  - vimdiff:
    - ◆ :diffupdate
    - ◆ zw. diffs springen: [c und ]c
- Synchronisation:
- ◆ :diffget
  - ◆ :diffput



# Vervollständ., Suche, Markieren

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

Vervollständigung beim Schreiben:

- Ctrl-N: nächstes Wort mit aktuellem Präfix
- Ctrl-P: vorheriges Wort mit aktuellem Präfix

Suche nach aktuellem Wort:

- vorwärts mit \*
- rückwärts mit #

Visuelles Auswählen von Text (visual mode):

- V - zeilenweise, v - zeichenweise, Ctrl-V - Rechteck (block)
- mit 'o' diagonal die Ecken tauschen.
- mit 'O' die Ecken horizontal tauschen.



# Histories und Eingebaute Hilfe

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- **Histories und Eingebaute Hilfe**
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

Histories für Kommandozeile und Sucheingabe:

- mit ':' oder '/' beginnen - dann Pfeiltasten
- Beenden mit Escape

Kommandozeilenfenster - History bei Suche und ex-Kommandos:

- q: (:he cmdwin)
- q/ bzw q?



# Probleme? Fragen richtig stellen!

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

- Versionsinfo mit Anfrage angeben:
- `$ vim --version`
- Sind die Problem nur in eigener Konfiguration?
  - ◆ Starten ohne Setupdateien:
  - ◆ `$ vim -N -u NONE -U NONE`
  - ◆ Ausführlichkeit der Fehlermeldungen erhöhen:
  - ◆ `$ vim -V2`



# Ressourcen

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

The End...

## Eingebaute Hilfe:

- `:help` → Hilfefenster öffnen
- `:helpgrep` → alle Hilfetexte nach Muster durchsuchen
- `:help toc` → "Table of Contents" (Inhaltsverzeichnis) des "User Manual"

## Webseiten:

- [vim.org](http://vim.org) - Skripte, Tipps, Doku
- [vimfaq](http://vimfaq.org)
- [vimtips](http://vimtips.org)

## Newsgruppen:

- `news://comp.editors`
- `news://de.comp.editoren`



# Literaturempfehlungen

Einleitung

---

Screen

---

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Probleme? Fragen richtig stellen!
- Ressourcen
- **Literaturempfehlungen**

Mutt

---

Slrn

---

Zsh

---

The End...

---

- Textbearbeitung mit dem vi-Editor (L. Lamb, A. Robbins) - ISBN://3-89721-126-2
- Reguläre Ausdrücke (Jeffrey E. F. Friedl) ISBN://3-89721-349-4
- "Vi IMproved – Vim" [www.vim.org/book.html](http://www.vim.org/book.html) von Steve Oualline - ISBN://0735710015
- vim ge-packt (Reinhard Wobst) ISBN://3-8266-1425-9



# Features 1/3

Einleitung

Screen

Vim

Mutt

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

Slrn

Zsh

The End...

## Minimals:

- from, realname
- hostname, hidden\_host, use\_from
- alternates + reverse\_name + to\_chars
- Sortierungen (thread, date, unsorted)
- limit (Anzeige einer Teilmenge nach Mustern)
- Kommandozeile für temporäre Änderungen (zum Ausprobieren)
- Indexanzeige -> index\_format
- edit\_headers - (fast) voller Zugriff auf den header per \$EDITOR



## Features 2/3

Einleitung

Screen

Vim

Mutt

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

Slrn

Zsh

The End...

- tagging + copy/save/delete oder auch reply
- "Haken" (hooks) beeinflussen Kommandos
- Farbe im Index und im Pager (Quoting-Level, Signatur, URLs)
- Pager - Anzeige von Headerzeilen - Auswahl, Reihenfolge, Farbe
- Kontext von Index+Pager
- Pipe - Weitergabe des Inhalts an \*beliebiges\* Kommando
- Attachment Menu - save + delete (nicht fuer signed/rypted data)
- display\_filter - Textteile verstecken und korrigieren (zB mit sed)



# Features 3/3

Einleitung

Screen

Vim

Mutt

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

Slrn

Zsh

The End...

- auto\_view von z.B. DOCs (antiword) und PDFs (pdftotext)
- Signaturen und Verschlüsselung mit PGP bzw GPG
- Key Binding + Macros
- bind index zz current-middle
- macro index ,N "<limit>~N ~d <2w ! ~=" \n'
- eigene Headerzeilen mit my\_hdr
- IMAP + POP
- urlview: Extrahieren von URLs und Darstellung in einem Menü; Weitergabe der URL an Browser möglich



# Ressourcen

Einleitung

Screen

Vim

Mutt

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

Slrn

Zsh

The End...

- <http://www.mutt.org>
- <news://comp.mail.mutt>
- <news://de.comm.software.mailreader.misc>



# Features - 1/2

Einleitung

Screen

Vim

Mutt

Slrn

● Features - 1/2

● Features - 2/2

● Ressourcen

Zsh

The End...

- Minimale Konfiguration:  
<http://www.guckles.net/setup/slrrnc.minimal>
- hostname + username + realname + replyto
- Group Menu - (un)sub einfacher als bei Mailinglisten
- Speicherung der Nachrichten auf dem \*Server\*
- "Gedächtnis" → newsrc; nur das Ungelesene zeigen
- Scoring (regex; inexpensive scoring)
- Anzeige von "matching rules" im Menu
- Threading = Anzeige von Diskussionen als "Bäume"
- Aus/Einklappen der Diskussionsbäume
- Wiederherstellen eines gelesenen Threads (Esc 1 Esc P)
- Kennzeichnung von Textarten durch Farbe
- Zitatlevel, \*Emphase\* + Signatur + Tilde + Verbatim



## Features - 2/2

Einleitung

Screen

Vim

Mutt

Slrn

● Features - 1/2

● Features - 2/2

● Ressourcen

Zsh

The End...

- Aus/Einblenden von zitiertem Text mit 'T' (toggle\_quotes)
- Key Binding (wie jedes gute TextTool)
- S-Lang: Funktionen + Macros
- Offline Reading (slrnpull)
- Abspeichern von Artikeln mit 'o' -> Lesen mit "mutt -f file"
- Numerische Tags -> Geordnetes Abspeichern
- Antworten mit Followup (öffentlich) und Reply (privat per E-Mail)
- Wichtige Server: news.individual.net + news.gmane.org
  - ◆ Keine angepasste Hilfe (nur fester Text) :-/
  - ◆ Sortierung: Unübersichtlicher Dialog
  - ◆ Weiterleiten: nur Forward - kein Bounce



# Ressourcen

Einleitung

Screen

Vim

Mutt

Slrn

● Features - 1/2

● Features - 2/2

● Ressourcen

Zsh

The End...

- <http://www.slrn.org/>
- <news://news.software.readers>



# Zsh - Features

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

● Zsh - Features

● Zsh - Features

● Ressourcen

● Literaturempfehlungen

The End...

- Module: speicherschonend
- Expansionen: `!!<TAB>`, `rm 'cat foo'<TAB>`
- `"echo {23..42}"`
- `"echo ${37*73}"`
- Kommandoergänzungen (completion control), z.B.:  
`"compctl -g '*(/)' cd"`
- Eingebaute Kommandos ("builtins")
- builtin `"zmv"` (eingebautes `"mmv"`)
- builtin `"vared"` ("`vared PATH`")
- global aliases:  
`"alias -g L='|less' "` und `"alias lsd='ls -F *(/)' "`
- globbing patterns (zB `"ls (*.bz2|*.gz)"`)
- Qualifiers:  
`"for datei in [A-Z][A-Z]*(.); do mv $datei $datei:l; done"`



# Zsh - Features

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

● Zsh - Features

● Zsh - Features

● Ressourcen

● Literaturempfehlungen

The End...

- Zeileneditor ("line editor") - Editieren über mehrere Zeilen
- "echo \$PATH" vs "echo \$path"
- Felder (Arrays): FOO=(bar baz)
- Prompt an der rechten Seite der Zeile (R\_PROMPT=%T)
- temporäre Datei "mutt -f =(zcat mailfold\*.gz)"
- Umlenkung von stdout an mehrere Ziele ("ls > txt1 >> txt2")
- "setopt autopushd; dirs -v; cd +2"
- "sched +0:1 banner Zsh rules"



# Ressourcen

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

● Zsh - Features

● Zsh - Features

● Ressourcen

● Literaturempfehlungen

The End...

- <http://www.zsh.org>
- <news://de.comp.os.unix.shell>



# Literaturempfehlungen

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

● Zsh - Features

● Zsh - Features

● Ressourcen

● Literaturempfehlungen

The End...

- ZSH Guide
- ZSH Intro [Peter Stevenson]
- ZSH Liebhaber Seite [Matthias Kopfermann]
- ZSH Lover Page [Sven Guckes]



# Q&A

Geschafft? ;-)  
Danke für die Aufmerksamkeit! :-)  
Gibt es Fragen?

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

The End...

● Q&A

● The End...



# The End...

Online: [www.linuxtage.at/prokop/](http://www.linuxtage.at/prokop/)

Danke für die Aufmerksamkeit!  
Feedback ist willkommen!

Michael Prokop [michael@linuxtage.at](mailto:michael@linuxtage.at) und  
Sven Guckes [sven@guckes.net](mailto:sven@guckes.net)

Viel Spaß noch auf den Grazer Linuxtagen!

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

The End...

● Q&A

● The End...