

Überleben auf dem Terminal - texttools++

Michael Prokop und Sven Guckes

akademy@michael-prokop.at / www.michael-prokop.at + akademy@guckes.net / www.guckes.net



Dieses Dokument steht unter der Create Commons 2.0 Lizenz



Inhalt

Einleitung

● Inhalt

Screen

Vim

Mutt

Slrn

Zsh

Utilities

The End...

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



Features 1/2

Einleitung

Screen

● Features 1/2

● Features 2/2

● Kommandos

Vim

Mutt

Slrn

Zsh

Utilities

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 2/2

Einleitung

Screen

● Features 1/2

● Features 2/2

● Kommandos

Vim

Mutt

Slrn

Zsh

Utilities

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 → Abkürzung eines ganzen Kommandonamens auf eine kurze Tastensequenz



Kommandos

Einleitung

Screen

● Features 1/2

● Features 2/2

● Kommandos

Vim

Mutt

Slrn

Zsh

Utilities

The End...

- `screen -help` → Hilfe zu Screen anzeigen
- `screen -ls` → Screen-Sessions anzeigen
- `screen` hört auf `Ctrl-A` aka `C-A` aka `^A` → 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 (gq)

● Falten / Folding

● Automatisierungen

● Fenster

● Buffers

● Programmieren mit Vim

● Nette Kommandos

● Goodies

● Vervollständ., Suche, Markieren

● Histories und Eingebaute Hilfe

● Utilities + Filter

● Probleme?

● Ressourcen

● Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

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: Expansion von Wort auf Phrase → :iab KDE K Desktop Environment
- MaKros: Kommandosequenz → :map <c-j> gqap
- Inkrementelle Suche (incsearch)
- Tab-Completion
- Filetypes, Syntax-Highlighting



Macros/Recording

Einleitung

Screen

Vim

● Fahrplan

● Macros/Recording

● Formatieren von Text (gq)

● Falten / Folding

● Automatisierungen

● Fenster

● Buffers

● Programmieren mit Vim

● Nette Kommandos

● Goodies

● Vervollständ., Suche, Markieren

● Histories und Eingebaute Hilfe

● Utilities + Filter

● Probleme?

● Ressourcen

● Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

The End...

■ Register / "Clipboard":

- ◆ a-z: Ersetzen des Inhalts
- ◆ A-Z: Anfügen von weiterem Inhalt

■ Recording

- ◆ Aufnahme: 'q' + [a-z] + «Kommandos» + 'q'
- ◆ 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 (gq)

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- **Formatieren von Text (gq)**
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

The End...

- gqip (Umbrechen des aktuellen Absatzes [ip = inner paragraph])
- vip + gq (erst Absatz visuell auswählen, und dann umbrechen)
- 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 (gq)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

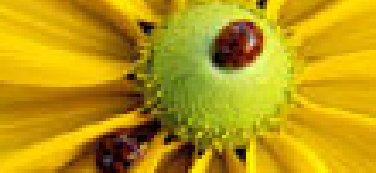
Zsh

Utilities

The End...

- Folding = Zusammenfalten von Blöcken auf einander folgender 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`

Automatisierungen



Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text (gq)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

The End...

- Template bei C-Dateien automatisch einlesen:

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

<pre>:autocmd</pre>	-> automatisches Ausführen von...
<pre>BufNewFile</pre>	-> bei Erzeugung eines neuen Buffers
<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 (gq)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

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

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>`

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text (gq)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

The End...



Programmieren mit Vim

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text (gg)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

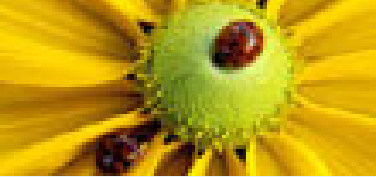
Zsh

Utilities

The End...

- `:set makeprg=...` und `:make`
 - `:cn` (next error), `:cp` (previous error)
 - `ctags[-exuberant]: "ctags *.cpp" 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 (gq)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- **Nette Kommandos**
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

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)
`gqip` -> 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 (gq)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

The End...

- :options → gesetzte Optionen anzeigen
 - :registers → Liste aller Register und deren Inhalte anzeigen
 - :buffers → Liste aller Buffer
 - vimdiff:
 - ◆ :diffupdate
 - ◆ zw. diffs springen: [c und]c
- Synchronisation:
- ◆ :diffget
 - ◆ :diffput



Vervollständ., Suche, Markieren

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text (gq)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

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 (Bsp: xterm -cr green).
- mit 'O' die Ecken horizontal tauschen.



Histories und Eingebaute Hilfe

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text (gq)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

The End...

Histories für Kommandozeile und Sucheingabe:

- mit ':' oder '/' beginnen - dann Pfeiltasten
- Beenden mit Escape: kein Ausführen aber Hinzufügen zu History
- Beenden mit Ctrl-C: kein Ausführen ohne Hinzufügen zu History

Kommandozeilenfenster - History bei Suche und ex-Kommandos (Beenden mit ':q'):

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



Utilities + Filter

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text (gq)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

The End...

- Input: cal, date, pal,... (':r!date')
- Filter: boxes, nl, par, rev, tr
- Fun: cowsay, figlet
- sort, uniq (logstuff!)
- awk, grep, sed (regex!)
- Code-Interface: Perl, Python, Ruby und Tcl (siehe 'vim -version')



Probleme?

Einleitung

Screen

Vim

- Fahrplan
- Macros/Recording
- Formatieren von Text (gg)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

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 (gg)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- Literaturempfehlungen

Mutt

Slrn

Zsh

Utilities

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 - 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 (gg)
- Falten / Folding
- Automatisierungen
- Fenster
- Buffers
- Programmieren mit Vim
- Nette Kommandos
- Goodies
- Vervollständ., Suche, Markieren
- Histories und Eingebaute Hilfe
- Utilities + Filter
- Probleme?
- Ressourcen
- **Literaturempfehlungen**

Mutt

Slrn

Zsh

Utilities

The End...

- Textbearbeitung mit dem vi-Editor (L. Lamb, A. Robbins) - ISBN://3-89721-126-2 und ISBN://1-56592-426-6
- "Vi IMproved – Vim" www.vim.org/book.html von Steve Oualline - ISBN://0735710015
- vim ge-packt (Reinhard Wobst) ISBN://3-8266-1425-9
- Reguläre Ausdrücke (Jeffrey E. F. Friedl) ISBN://3-89721-349-4



Features 1/3

Einleitung

Screen

Vim

Mutt

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

Slrn

Zsh

Utilities

The End...

Minimals:

- "From:" → set from (Adresse!), realname (Name!)
- 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

Utilities

The End...

- tagging + copy/save/delete oder auch reply(!)
- "Haken" (hooks) beeinflussen Kommandos: account-hook, charset-hook, fcc-hook, fcc-save-hook, folder-hook, iconv-hook, message-hook, mbox-hook, pgp-hook, crypt-hook, reply-hook, save-hook, send-hook, unhook
- Farbe im Index und im Pager (Quoting-Level, Signatur, URLs)
- Pager: Anzeige von Headerzeilen → Auswahl, Reihenfolge, Farbe
- Kontext von Index+Pager (pager_context)
- Pipe: Weitergabe des Inhalts an *beliebiges* Kommando
- Attachment-Menü: save + delete (nicht für 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

Utilities

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
- Neue Mails, jünger als 2 Wochen und keine Dublikate:
macro index ,N "<limit>~N ~d <2w ! ~="\"n'
- zusätzliche Headerzeilen mit my_hdr
- IMAP + POP3
- 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

Utilities

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

Utilities

The End...

- MiniKonfig: hostname + username + realname + replyto
- Group Menu: (un)subscribe 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 Menü
- Threading = Anzeige von Diskussionen als "Bäume"
- Ein- und Ausklappen 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

Utilities

The End...

- Ein- und Ausblenden von zitiertem Text mit 'T' (toggle_quotes)
- Keybinding (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) bzw. Reply (privat per E-Mail!)
- Wichtige Server: news.individual.net + news.gmane.org



Ressourcen

Einleitung

Screen

Vim

Mutt

Slrn

● Features - 1/2

● Features - 2/2

● Ressourcen

Zsh

Utilities

The End...

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



Features 1/3

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

● Literaturempfehlungen

Utilities

The End...

- Module: speicherschonend
- (inline) Expansionen: `!!<TAB>`, `rm 'cat foo'<TAB>`
- `"echo {23..42}"`
- `"echo ${37*73}"`
- Kommandoergänzungen (completion control), z.B.:
 - ◆ `compctl -g '*(/)' cd # altes System`
 - ◆ `zstyle ':completion:*:cd:*' tag-order local-directories directory-stack named-directories path-directories # neues System`
- Eingebaute Kommandos ("builtins")
- Builtin `"zmv"` (eingebautes `"mmv"`)
- Builtin `"vared"` ("`vared PATH`")



Features 2/3

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

● Literaturempfehlungen

Utilities

The End...

■ Globale Aliases:

- ◆ alias -g C='| wc -l'
- ◆ alias -g G='| grep'
- ◆ alias -g H='| head'
- ◆ alias -g L='| less'
- ◆ alias -g P='|'
- ◆ alias -g T='| tail'
- ◆ alias -g V='| grep -v'
- ◆ Bsp: «command» G foo L

■ Globbing Patterns:

- ◆ ls (*.bz2|*.gz)
- ◆ hash -d web=/var/www/kde &&
tar jcf foo.tar.bz2 web/**/*(.R)
- ◆ alias lsd='ls -F */' # no drugs!

■ Qualifiers:

”for datei in [A-Z][A-Z]*(.); do mv \$datei \$datei:|; done”



Features 3/3

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

● Literaturempfehlungen

Utilities

The End...

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



Ressourcen

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

● Literaturempfehlungen

Utilities

The End...

- <http://www.zsh.org>
- <news://de.comp.os.unix.shell>
- "From Bash to Z Shell: Conquering the Command Line"



Literaturempfehlungen

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

● Features 1/3

● Features 2/3

● Features 3/3

● Ressourcen

● **Literaturempfehlungen**

Utilities

The End...

- "A User's Guide to ZSH" von Peter Stephenson
- ZSH Lovers
- ZSH Features [von Michael Prokop]
- ZSH Liebhaber Seite [von M. Kopfermann und M. Prokop]



Utilities 1/2

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

Utilities

● Utilities 1/2

● Utilities 2/2

The End...

- wget/puf/proz - Downloadprogramm
- less/more/most - Pager (Viewer)
- (e)links/w3m/lynx - Browser
- bvi - Hex-Editor
- cal+pal - Kalenderprogramm
- gpg - Verschlüsselungsprogramm
- ncftp - FTP-Client
- awk/grep/sed - Textfilter
- sort/uniq - Sortieren



Utilities 2/2

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

Utilities

● Utilities 1/2

● Utilities 2/2

The End...

- boxes/figlet - ASCII-drawing
- nl - Zeilen nummerieren
- tr - Zeichen entfernen
- par - Absätze formatieren
- tpp - text presentation program



Danke!

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

Utilities

The End...

● Danke!

● The End...

Geschafft? ;-)

Danke für die Aufmerksamkeit! :-)

Wir danken...

- der aKademy
- den Communities von \$SOFTWARE und
- Kurt Pfeifle!



The End...

Danke für die Aufmerksamkeit!
Feedback ist willkommen!

Michael Prokop akademy@michael-prokop.at und
Sven Guckes akademy@guckes.net

Viel Spaß noch auf der KDE Community World Summit 2004
"aKademy"!

Einleitung

Screen

Vim

Mutt

Slrn

Zsh

Utilities

The End...

● Danke!

● The End...