

1 About

1.1 die Zsh

- Unix Shell
- 1990 von Paul Falstad, Princeton University
- wird aktiv weiterentwickelt
- aktuelles Release 4.3.17 von 2012-02-23
- Hilfreiche Community
 - `\#zsh@irc.freenode.net`
 - <http://www.zsh.org/mla>
- Umfangreiche (!) Feature-Liste
 - 17 man-pages
 - + fast Superset der Features von bash, ksh, csh

2 BashUser

2.1 Von Bash zu Zsh ohne Einarbeitungszeit

- zsh in default-config der bash sehr ähnlich
- bash mit zsh ersetzen Alleine bringt schon Vorteile

Features

- Tippfehlerkorrektur
- gemeinsame History über mehrere laufende Shells hinweg
- vcs_info prompt
- ZLE statt readline
 - Menü-Auswahl
 - multi-line command prompt editing
- wesentlich mächtigere Tab-Vervollständigung
- Verzeichnis-Vervollständigung `/u/in/X -> /usr/include/X11/`
- Variablen-expansion `$TERM<tab>`
- floating point math `echo $((2.2+3.3**2))`
- korrekte Behandlung von Quoting:
 - `[[-n $var]]`
 - `echo "abc $(echo "d e f") def"`

2.2 Weitere Features

- `[[-n $var || $var == *abc* || $var =~ ^var$]]`
- Assoziative Arrays
- Dateimanager mit ZLE Menu
- ZLE Undo
- function hooks
- shorthands `for a (array)`
- global aliases, suffix aliases (z.b.: für `>/dev/null`)
- `cmd &!`
- Umfangreiche Konfigurierbarkeit
 - `man zshoptions`
 - `zstyle`

3 Redirection

3.1 Redirection in zsh

- Alles was in bash geht, geht auch in zsh
- Stdout und Stderr in Prozess pipen `helloouterr |& less`
- Prozess-Output als Fifo oder Temp-File verwenden `less <(sort -n tosort) less =(sort -n tosort)`
- Dateiinhalt in die CL `echo $(< file)`
- CL Zeichenfolge in Temp-Datei `cat =(<<< "text der in ein tmp file kommt")`
- Multi-Target Redirect ! `echo HalloHallo > file1 > file2`
- File durch Prozess pipen und überschreiben `sort -n =(<sortme) >! sortme`
- Shell-Optionen: `noclobber` und `histallowclobber`

4 Globbing

4.1 Globbing

```
=<cmd>    expand executable cmd to full path
<m-n>    number between m and n.
~B       files not matching B
A~B~C    files matching A but not B and not C
(#i)    case insensitive. OFF: (#I)
(#l)    lower case match upper case but not vv
(#b)    backreferences into $match
(#a<n>) approximate match, <n> errors allowed
(#s)    equals ^ in RE
(#e)    equals \$ in RE
```

4.2 Extended Globbing Flags

Syntax: `ls *(qualifiers[^not-qualifiers])`

```

/      directories
F      filled directories
.      plain files
@      symbolic links
=      sockets
p      named pipes (FIFOs)
*      executable plain files (0100)
%      device files (character or block special)
%b     block special files
%c     character special files
r      owner-readable files (0400)
w      owner-writable files (0200)
x      owner-executable files (0100)
A      group-readable files (0040)
...
s      setuid files (04000)

```

4.3 Extended Globbing Flags

Syntax: `ls *(qualifiers[^not-qualifiers])`

```

[m]      only return m'th result (can be negative)
[m,n]    only return results m to n
u<u>     owned by user (<u> is UID or :username:)
g<g>     owned by group (<g> is GID or :groupname:)
D        include .files in *
N        NULL_GLOB: empty string if no match
d<d>     files on device <d>
L<n>     files of size <n> Byte
          (use Lk,Lm,Lg for KiB,MiB,GiB)
m<n>     files modified <n> days ago
          (use mM,mh,mm for Months,Hours,Minutes)
c<n>     metadata changes <n> days ago
          (prepend <n> with +/- for at-least/at-most)
o/O_     sort in asc/desc order by qualifier _
  n      by names
  L      by filesize
  m/a/c  by file m/a/c-time

```

4.4 Globbing Modifiers

Syntax: `print *(<qualifiers>:<modifier1>:<modifier2>)`

```

a      turn filename into absolute path
A      like a but resolve symbolic links too
t      only show basename (tail of filepath)
h      only show head of filepath
e      only show extension of filename
r      remove extension, only show root of filename
q      quote
x      quote, but separate on whitespace
Q      remove one level of quotes
l      convert to lowercase
u      convert to uppercase
s/l/r  substitute l with r once. global: gs/l/r

```

modifier modifiers ... see man-page

5 Hilfe

5.1 Hilfe ? Inline-Hilfe !

- `<Tab>` hilft fast immer (sonst `<Tab><Tab>`)
- `<Strg>Xd` Vorschau für Globbingausdruck unter Cursor
- `<Alt>H` man-page für Cmd unter Cursor
- `<Strg>Xz` Blättert durch Keybinding Hilfe
- `<Strg>Xg` Blättert durch Globbing Hilfe
- `<Strg>Xb` Zeigt Hilfe für `.` Abkürzungen

(für grml config)

6 UseCases

- compile a quick c prog writing to stdout and stderr
- demonstrate multi-line editing and redirection

```
gcc -o helloouterr -x c =( <<<'#include <stdio.h>
int main()
{
    fprintf(stdout, "Hallo StdIn\n");
    fprintf(stderr, "Hallo StdOut\n");
}' )
```

- cd to dir that contains file `*besserverstecker*`

```
cd **/*(\#i)Besserverstecker*([1]:h)
```

- remove dangling symlinks

```
rm **/*(-@)
```

- list all symbolic links that point to ...

```
la *(@e%'[[ ${REPLY:A} == *dira* ]]'%)
la *(@e%'[[ ${REPLY:A} == /tmp* ]]'%)
la *(@e%'[[ ${REPLY:A} == ~ ]]'%)
la *(@e%'[[ ${REPLY:A} == /boot ]]'%)
la *(@e%'[[ ${REPLY:A} == $PWD ]]'%)
```

- rmdir all empty directories in all subdirectories

```
rmdir **/*(^F)
```

- cp new files/photos of last 10 days
- create dir newfotos on the fly

```
cp /media/CAM/**/*(.c-10) newfotos/<Ctrl+Xp>
```

- only rm *.txt files that contain „DELETE ME“

```
echo "IMPORTANT" > {a-f}.txt
echo "DELETE ME" > {g-l}.txt
mkdir z.txt
```

```
contains() { grep -q "$*" $REPLY }
rm *.txt(.e:'contains DELETE ME':)
```

7 Configs

7.1 Grml oder Xro Features

- any
- lsdisk
- purge
- se, simpleextract
- ati, acs
- cdt
- rmdir
- qcp, qmv
- iptables-show
- sum_filesizes
- waitpid

7.2 Empfohlene Zsh Configs

- Grml-Config
 - <http://grml.org/zsh/>
 - apt-get-repository deb <http://deb.grml.org> grml-stable apt-get install grml-etc-core
- Meine Config
 - <https://www.tittelbach.at/zsh/>

```
wget --no-check-certificate \
  https://www.tittelbach.at/zsh/.zshrc{,.pre,.local}
zsh
UpdateMyZshrc
```