

# Einführung in die GNU Bourne-Again Shell

Johannes Weiß

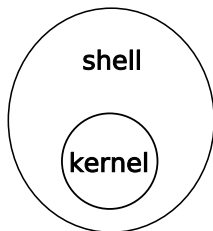
LUG Burghausen

18. November 2006

- 1 Einführung
- 2 Erste Schritte
- 3 Befehlsaufbau
- 4 Bedienungs-Features
- 5 Automatisierung/Ausblick

# Was ist eine Shell?

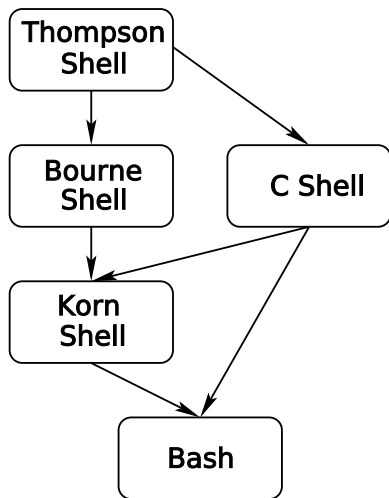
- **Shell** (engl.): Hülle, Schale, Muschel
- Verbergen der inneren Details des Betriebssystems
- Möglichkeit für Benutzer, Befehle abzugeben
- Unter UNIX: Kommandozeileninterpreter (Traditionelle Bedienung)



# Warum eine Shell benutzen?

- Starten von Programmen (oft schneller als Menü)
- Probleme unter Linux manchmal nur über Shell lösbar
- Einzige Möglichkeit zum Bedienen (Netzwerk, alter Rechner)
- Automatisieren von Abläufen, Effizienz

# Geschichte der UNIX-Shells



- 1971 **Thompson Shell**  
Ken Thompson: erste UNIX-Shell
- 1978 **Bourne Shell**  
Stephen Bourne: in UNIX 7
- 1978 **C Shell**  
Bill Joy: in 2BSD, C-ähnliche Syntax, nicht kompatibel mit Bourne Shell
- um 1980 **Korn Shell**  
David Korn: kompatibel zur Bourne Shell, viele Funktionen aus der C-Shell
- 1987 **Bash** („Bourne-Again“)  
Brian Fox: Shell des GNU-Projekts, meist verbreitet
- viele andere Shells, z.B. zsh, rc oder ash

# Die Bash starten

- Auf praktisch allen Linux-Systemen schon installiert
- Aufruf interaktiv oder per Script
- Interaktiv: Direkt nach Login oder in **Terminal Emulator**:  
z.B. XTerm, Gnome-Terminal oder Konsole

# Prompt

Nach dem Start: **Eingabeaufforderung** (*Prompt*), z.B.

```
gast@laptop1:~/verzeichnis1$
```

- `gast`: Benutzername
- `laptop1`: Computer, auf dem man arbeitet
- `~/verzeichnis1`: Aktuelles Verzeichnis

- **Befehle:** Eingeben, mit *Enter* abschließen
- Eingebaute Befehle oder externe Programme

## Kleine Befehlsreferenz

<code>ls</code>	Inhalt von Verzeichnissen auflisten ( <i>list</i> )
<code>pwd</code>	Aktuelles Verzeichnis ausgeben ( <i>print working dir</i> )
<code>cd Verz.</code>	Verzeichnis wechseln ( <i>change dir</i> )
<code>rm Dat.</code>	Datei(en) löschen ( <i>remove</i> )
<code>mv Dat. Verz.</code>	Datei(en) ins Verzeichnis verschieben ( <i>move</i> )
<code>cp Dat. Verz.</code>	Datei(en) ins Verzeichnis kopieren ( <i>copy</i> )
<code>mkdir Verz.</code>	Verzeichnis erstellen ( <i>make dir</i> )
<code>rmdir Verz.</code>	Verzeichnis löschen ( <i>remove dir</i> )

## Kleine Befehlsreferenz

<code>echo <i>Text</i></code>	Text ausgeben
<code>cat <i>Dat.</i></code>	Datei(en) nacheinander ausgeben ( <i>concatenate</i> )
<code>man <i>Befehl</i></code>	Hilfeseite zu einem Befehl anzeigen ( <i>manual</i> )
<code>less <i>Datei</i></code>	Datei anschauen
<code>find <i>Verz.</i></code>	Dateien im Verzeichnis finden
<code>grep <i>Text Datei.</i></code>	Text in Dateien suchen

## Ein kurzes Beispiel

```
gast@laptop1:~$ ls
filme/  mail/  programme/  README  texte/
gast@laptop1:~$ cd texte
gast@laptop1:~/texte$ ls
einkaufszettel  stundenplan
gast@laptop1:~/texte$ cat einkaufszettel
Milch
Brot
Butter
gast@laptop1:~/texte$ rm einkaufszettel
gast@laptop1:~/texte$ ls
stundenplan
gast@laptop1:~/texte$
```

# Einfache Befehle

## Definition

Befehl *Argument1 Argument2 ...*

*Wichtig:* Trennung durch Leerzeichen  
z.B. nicht „cd..“ sondern „cd ..“

## Beispiel

*cp Datei Verzeichnis*

Datei in Verzeichnis kopieren

# Programme als Filter



- Unixprinzip: Programme als **Filter** für Daten (Text)
- Viele Shell-Befehle und UNIX-Programme sind Filter
- **Standardeingabe**: Tastatur  
**Standardausgabe**: Bildschirm

# Pipelines

## Definition

`Befehl1 Argumente | Befehl2 Argumente`

- Senkrechter Strich: Umleitung der Ausgabe von Befehl1 in Eingabe von Befehl2

## Beispiel

```
ls | wc -l
```

```
cat Datei1 Datei2 | sort | uniq
```

# Kette von Befehlen

## Definition

`Befehl1 Argumente ; Befehl2 Argumente`

- Strichpunkt: Befehle/Pipelines hintereinander ausgeführt
- Eingabe und Ausgabe unberührt: Tastatur/Bildschirm

## Beispiel

```
cd Verzeichnis ; ls | wc -l
```

# Quoting

**Quoting:** Aufheben der Spezialbedeutung bestimmter Zeichen

- *Escape-Zeichen (Backslash)*
- *Einfache Anführungszeichen*
- *Doppelte Anführungszeichen*

## Beispiel

```
echo hallo \; ls  
echo '"Das ist ein Zitat."'  
cd "Ein Verzeichnis"
```

# Umleiten von Eingabe und Ausgabe

- Umleiten von Ein/Ausgabe aus/in Dateien möglich:
  - `Befehl > Datei`: Ausgabe von Befehl in Datei schreiben
  - `Befehl < Datei`: Eingabe von Befehl aus Datei lesen
  - `Befehl >> Datei`: Ausgabe an die Datei anhängen
- Noch weitere Umleitungen und Datenkanäle vorhanden
- Dadurch Ergebnisse von Pipelines speicherbar

## Beispiel

```
cat Datei1 Datei2 | sort > Datei3
```

# Dateinamenexpansion

- Auch: Wildcards oder Jokerzeichen
- **Muster:** Wörter mit \*, ?, [ oder ] (unquotiert)
- Vergleich mit allen Dateinamen im aktuellen Verzeichnis, Ersetzung durch Treffer

# Dateinamenexpansion

- „\*“: beliebige Zeichenfolge, auch leere, z.B.:  
haus\* → haus hausgang hausaufgabe
- „?“: **ein** beliebiges Zeichen, z.B.:  
a?a → axe ate ale
- „[...]“: beliebiges Zeichen in [...], z.B.:  
a[xt]a → axe ate

## Typische Anwendung

```
mv *.gif bilder/  
rm *. [ch]
```

# Tab-Vervollständigung

- Drücken der Tab-Taste:  
Versuch, Befehl sinnvoll zu vervollständigen
  - 1. Wort: Aliase und Befehle
  - Folgende Worte: Datei im aktuellen Verzeichnis
- Bei Mehrdeutigkeit: 2. Tab liefert Liste der Treffer

- Speicherung aller eingegebenen Befehle
- **history**: Ausgabe aller gespeicherten Befehle
- **Pfeil-oben** oder **Pfeil-unten**:  
Vorhergehenden oder Nächsten Eintrag anwählen
- **Strg-r**: In der History suchen

- Abkürzungen für andere Befehle
- Zeichenfolge als Alias für anderen Befehl
- **alias** ohne Argumente: Anzeigen bestehender Aliase
- **unalias** *Alias*: Alias löschen

## Beispiel

```
alias schreibe='echo'
```

```
alias bilder_verschieben='mv *.gif'
```

Prozesse (laufende Befehle): Möglichkeit zum Anhalten, in den Hintergrund oder zurück in den Vordergrund holen

- **Strg-Z**: Ein Vordergrund-Job wird angehalten
- **jobs**: Aktive Jobs anzeigen
- **fg Job**: Job in den Vordergrund bringen
- **bg Job**: Job in den Hintergrund stellen
- Befehl **&**: Befehl im Hintergrund starten

# Readline: Editieren der Kommandozeile

- Nützliche Tastenkürzel für **Readline**
- Relativ frei wählbar, Belegungen für *Emacs* und *vi* vorhanden (hier nur eine kleine Auswahl der Emacs-Kommandos)
- Praktisch: Kürzel in jedem Readline-Programm verwendbar

## Readline: Editieren der Kommandozeile

### Cursor-Bewegung:

Strg-a, Pos1

Zum Anfang der Zeile springen

Strg-e, Ende

Zum Ende der Zeile springen

Strg-b, Pfeil-links

ein Buchstabe nach links [backward]

Strg-f, Pfeil-rechts

ein Buchstabe nach rechts [forward]

Alt-b, Strg-Pfeil-links

Wort nach links [backward]

Alt-f, Strg-Pfeil-rechts

Wort nach rechts [forward]

Strg-l

Bildschirm löschen [clear screen]

# Readline: Editieren der Kommandozeile

## Textveränderung:

- Strg-d, Entf Zeichen an Cursorposition löschen [delete]
- Backspace Zeichen vor der Cursorposition löschen
- Alt-t Wörter vertauschen [transpose]

## Löschen:

- Strg-k Vom Cursor bis zum Ende der Zeile löschen [kill]
- Strg-x Backspace Vom Anfang der Zeile bis zum Cursor löschen
- Alt-d Vom Cursor bis zum Ende des Wortes löschen [delete]
- Alt-Backspace Vom Cursor bis zum Anfang des Worten löschen

# Readline: Editieren der Kommandozeile

## History:

Strg-p, Pfeil-oben	History nach oben [prev]
Strg-n, Pfeil-unten	History nach unten [next]
Alt-⟨, Bild-oben	Anfang der History
Alt-⟩, Bild-unten	Ende der History
Strg-r	In der History suchen (nach hinten) [reverse]

## Vervollständigung:

Tab	Versuch der Vervollständigung des Texts beim Cursor
Alt-*	Einfügen aller Möglichkeiten

# Ausblick: Automatisierung

- Bash-Befehle in Dateien speicherbar: **Script**
- Erste Zeile der Datei: `#!/bin/bash`
- Vorteil: Probleme einmal lösen, dann nie wieder

# Ausblick: Automatisierung

Beispiele:

- Große Anzahl von PDF/Word Dateien durchsuchen
- Große Anzahl von Bildern bearbeiten



*Bash Manpage (man bash).*



ELLEN SIEVER, AARON WEBER, ...:

*Linux in a Nutshell.*

O'Reilly Verlag, 2005.



LINUXFIBEL:

*Die Bourne Again Shell.*

<http://www.linuxfibel.de/bash.htm>



LINUX-KOMPENDIUM:

*Einführung in die Bourne Again Shell*

[http://de.wikibooks.org/wiki/Linux-Kompodium:  
\\_Bourne\\_Again\\_Shell](http://de.wikibooks.org/wiki/Linux-Kompodium:_Bourne_Again_Shell)



SELFLINUX:

*Einführung in die Bourne Again Shell (bash).*

[http://www.selinux.org/selflinux/html/bash\\_basic.html](http://www.selinux.org/selflinux/html/bash_basic.html)



/BIN/BASH:

*Treffpunkt für Linux-Shelluser.*

<http://www.bin-bash.de/>