Unix – ganz schmerzlos...

Teil 5 - Shell-Skripte I

Jörn Clausen

jc@Genetik.Uni-Bielefeld.DE

Die Shell

- bisher:
  - interaktive Bedienung
  - Eingabe einzelner Befehle
  - Ausgabe von Ergebnissen
  - Wildcards
  - Ein-/Ausgabe-Umlenkung, pipes
  - einfache Verknüpfung weniger Befehle
- demnächst:
  - Sequenz von mehreren Befehlen
  - Ablaufsteuerung (Bedingungen, Wiederholungen, Verzweigungen)
- vollständige Programmiersprache

Sequenz von mehreren Befehlen

- Datei `recent.sh`:

```
date
cd
ls -ltr | tail -10
```

- Aufruf:

```
juser@hobel> /bin/sh recent.sh
Mon May 15 14:34:46 MEST 2000
-r--------   1 juser   users   1145 Apr 24 10:59 baz.txt
...
drwx------   2 juser   users   1024 May 12 09:47 Mail
```

- Beachte:
  - `ls -l` statt `ll` verwenden
  - current working directory unverändert

Shell-Skript

- Dateianfang:

```
#!/bin/sh
```

- x-Bit(s) setzen:

```
-rwxr-xr-x   1 juser   support   36 May 15 15:13 recent.sh
```

- Skript direkt startbar:

```
juser@hobel> recent.sh
```

- # Kommentarzeichen:

```
# Zeige zuletzt bearbeitete Dateien an
# (C) jc 2000/05/15
```

Ein-/Ausgabe, pipes

- Shell-Skripte kennen stdin und stdout
- in pipes verwendbar: `psjuser.sh`

```
#!/bin/sh
grep juser > /tmp/t1
sort /tmp/t1
rm -f /tmp/t1
```

- Aufruf:

```
juser@hobel> ps -ef | psjuser.sh | more
```

Variablen

- Ablage für textuelle Daten
- Variablenname aus Zeichen zusammengesetzt
- symbolische/sprechende Namen sinnvoll, z.B. `benutzer`
- Zuweisung:

```
benutzer="Joe User"
```

- Inhalt: $

```
#!/bin/sh
benutzer="Joe User"
echo Der Benutzer heisst $benutzer
```

Variablen, cont.

- mehrere Variablen möglich:

```
#!/bin/sh
vorname="Joe"
name="User"
echo Mein Name ist $name,
echo aber Du darfst mich $vorname nennen.
```

- an beliebiger Stelle verwendbar:

```
#!/bin/sh
opt1="-l"
opt2="-latr"
ls $opt1
ls $opt2
```

spezielle Variablen

- Argumente: `$1`, `$2`, `...`, `$9`
- `mixer.sh`:

```
#!/bin/sh
echo Wenn man $1 und $2 mischt erhaelt man $3.
```

- Verwendung:

```
juser@hobel> mixer.sh gelb blau gruen
Wenn man gelb und blau mischt erhaelt man gruen.
juser@hobel> mixer.sh Wasserstoff Sauerstoff Knallgas
...
```

- lesbarer Code:

```
farbe1=$1
farbe2=$2
mischfarbe=$3
echo Wenn man $farbe1 und $farbe2 mischt ... $mischfarbe.
```