

# Unix – ganz schmerzlos...

## Teil 7 - Shell-Skripte II

Jörn Clausen

jc@Genetik.Uni-Bielefeld.DE

# spezielle Variablenzuweisungen

- eindeutiger Name für temporäre Dateien

```
tmpname=/tmp/tmpdata.$$  
sort > $tmpname
```

- \$\$ enthält pid der Shell

- \$# enthält Anzahl der übergebenen Argumente

- backtick evaluation

```
day=`date +%d`  
month=`date +%m`  
year=`date +%Y`  
echo heute ist der $day. $month. des Jahres $year
```

# Kontrollstrukturen

- unbedingte Schleife
  - wiederhole Vorgang für Liste von Dateien
- Verzweigung
  - wenn – dann – sonst
  - Fallunterscheidung
- bedingte Schleife
  - wiederhole Vorgang solange, bis Bedingung eintritt

# unbedingte Schleifen

- Welcher Drucker ist gerade frei?

```
for printer in w6ps w6color w7ps; do
    lpstat -p $printer
done
```

- \$\* Liste aller Argumente

- konvertiere mehrere Dateien in 2-up PostScript

```
for filename in $*; do
    a2ps $filename > $filename.ps
done
```

# Bedingungen

- Test ob
  - Variable gleich Wert
  - zwei Variablen gleich
  - Datei vorhanden
  - Datei lesbar/schreibbar
  - Datei bestimmtem Benutzer gehört
  - ...
- weiterer Ablauf des Skripts von Bedingung abhängig
- man-page: `test`

# Verzweigungen

- Auswertung von Variablen

```
if [ "$1" = "hallo" ]; then
    echo auch hallo
fi
```

- Verzweigung

```
if [ "$1" = "hallo" ]; then
    echo auch hallo
else
    echo Du kannst ruhig hallo sagen
fi
```

# Verzweigungen, cont.

- Bedingungen mit Dateien

```
if [ -f $HOME/.profile ]; then
    echo Du besitzt eine eigene .profile
else
    echo Du besitzt keine eigene .profile
fi
```

- Bedingungen mit permissions

```
path=$1
if [ -w $path ]; then
    echo Du darfst $path schreiben
else
    echo Du darfst $path nicht schreiben
fi
```

# bedingte Schleifen

- warte, bis Datei vorhanden ist

```
while [ ! -f $HOME/schluss ]; do
    echo .
    sleep 1
done
echo $HOME/schluss gefunden
```

- `juser@hobel> touch schluss`

- sammle Ns

```
n=""
while [ "$n" != "NNNNNNNNNN" ]; do
    echo sammle Ns
    n="N$n"
done
echo 10 Ns: $n
```

- Vorsicht: "N\$n" vs. "\$nN"