

LPI 102

Shell, Scripting, Programmieren und Administrative Tätigkeiten

Stefan Hühner - Maurice Massar

`lpi@unix-ag.uni-kl.de`

Universität Kaiserslautern - Unix-AG

Shellskripte

- Was ist das?
- Beispiel: Datei foo

```
latex $1.tex
latex $1.tex
dvips $1.dvi
dvi2pdf $1.dvi
```
- Ausführen mit: `bash foo lpish-fo`
- oder:
- erste Zeile: `#!/bin/bash`
- `chmod +x foo`
- `./foo lpish-fo`

Shellskripte

- Returncode: `exit 0`
- Konvention: `0` $\Leftrightarrow$ kein Fehler
- `$?`: letzter Returncode
- Beispiel: `/bin/false ; echo $?  $\Rightarrow$  1`
- Verkettung mit `&&` und `||`:
`latex lpish.tex || exit 1`
- alternativ zu `|| exit 1`: `set -e` (bei einfachen Befehlen)
- zum Debuggen: `set -x`

Shellskripte

- für Vergleich von Zahlen, Zeichenketten sowie Dateieigenschaften: `test` (Synonym: `[ ... ]`)
- `test -n "$1" && echo $1`
- *if* Bedingung:

```
if [ "$1" = "--help" ]; then  
    echo Hallo  
    exit 0  
fi
```

Shellskripte

● Syntax:

```
if list; then
    list
elif list; then
    list ...
else
    list
fi

case "$variable" in
    Muster1) list ;;
    Muster2) list ;;
    ...
    *) list ;;
esac
```

```
while list; do list; done
for name [ in word ] ; do list ; done
```

Shellskripte

- Beispiel: Datei foo

```
#!/bin/bash
```

```
set -e
```

```
for i in "$@"; do  
    latex $i.tex  
    latex $i.tex  
    dvips $i.dvi  
    dvi2pdf $i.dvi  
done
```

Shellskripte

- Shellvariable setzen: `F00="bar"`
- Shellvariable zu Umgebungsvariablen deklarieren:
`export F00`
- beides zusammen: `export F00="bar"`
- entfernen: `unset F00`
- Substitution: `$F00` $\Rightarrow$ Wert von `FOO` wird an dieser Stelle eingesetzt
- `${F00}` wie `$F00` aber: `${F00}bar` möglich
- `$(command)` oder `\command\`: Ausgabe von `command` wird eingesetzt
- Anzeigen von Shellvariablen: `set`, nur exportierte: `env`

Shellskripte

- Substitution ganz deaktiviert: ' '
- für einzelne ' ist \' zu schreiben
- Für Leerzeichen in einem Parameter aber mit Substitution: " . . . "
- Spezialfälle:
- "\$*" ⇒ "\$1 \$2 \$3 . . ."
- "\$@" ⇒ "\$1" "\$2" "\$3" . . .

Shellskripte

- Funktionen definieren: `name () { list; }`

- Beispiel:

```
add() {  
    echo $(( "$1" + "$2" ))  
    return 0  
}
```

- Aufruf: `add 2 3`
- mit `return` nur numerische Werte von 0 bis 255!

lokale und globale Shellumgebung

- Shellumgebung einrichten:
- Loginsells: `/etc/profile` sowie von `~/.bash_profile`, `~/.bash_login`, `~/.profile` die erste lesbare
- bei Logout noch `~/.bash_logout`
- Interaktive nicht-Loginsells:
`/etc/bash.bashrc` und `~/.bashrc`
- nicht-interaktive, nicht-login Shells: -
- Eingabe via Readline $\Rightarrow$ `~/.inputrc` und `/etc/inputrc`

User und Gruppenmanagement

- `/etc/passwd`: Infos zu Benutzeraccounts
- Beispielzeile:
`magnus:x:26001:100:Nils Magnus:/home/magnus:/`
- Felder: login, password, uid, gid, realname, home, shell
- `passwd` lesbar für alle $\Rightarrow$ Einführung von "shadow"
- Beispielzeilen von `/etc/shadow`
`magnus:gF5ZVbR9ov0Vw:11117:::::::::134542392`
`koppen:$1$c/Kwb/jK$9eNzw8/cSqRq731U4MJ5Y.:124`
- Felder: login, password, last-changed, min-days, max-days, warn-days, disable-after-expire-days, disable-days
- Shadow an: `pwconv`; aus: `pwunconv`. Prüfung: `pwck`

User und Gruppenmanagement

- Analog für Gruppen: `/etc/group` und `/etc/gshadow` sowie `grpconv`, `grpunconv`, `grpck`
- Beispielzeile von `/etc/group`:
`linux:x:208:koppen,kester,kleinhenn,magnus,...`
- Felder: gruppen-name, password, gid, user-liste
- editieren mit `vipw [-s]` bzw. `vigr [-s]`
- Passwort ändern: `passwd`
- nach Liste ändern: `chpasswd`
- Gruppenpasswort: `gpasswd`

User und Gruppenmanagement

- Benutzer anlegen, modifizieren, löschen:
useradd, usermod, userdel
- Gruppen anlegen, modifizieren, löschen:
groupadd, groupmod, groupdel
- Zeiten in shadow anpassen: chage
- Vorlage für \$HOME von neuen Benutzern:
/etc/skel/

cron und at

- Periodisches Ausführen von Kommandos: cron
- Format `/etc/crontab` und `/etc/cron.d/*`:
minute hour day-of-month month day-of-week user
command
- Format für "user"-crontabs
`/var/spool/cron/crontabs/*`:
minute hour day-of-month month day-of-week
command
- In beiden möglich: Umgebungsvariablen setzen
- häufig: `PATH=...` und `MAILTO=...`
- Ändern der Usercrontabs:
`crontab [-u user] {file | -l | -r | -e}`

cron und at

- /etc/crontab

```
SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:.
```

```
# m h dom mon dow user  command
17 * * * * root    run-parts /etc/cron.hourly
```

- /etc/cron.{hourly,daily,weekly,monthly}

- /etc/cron.{allow,deny}

- /etc/anacrontab

```
# period delay job-identifier  command
1      5      cron.daily    run-parts /etc/cron.daily
```

cron und at

- Ausführen von Befehlen zum bestimmtem Zeitpunkt: at
- at: planen von Befehlslisten
- atq, atrm: anzeigen/löschen der geplanten Listen
- atrun: geplante Jobs sofort ausführen
- /etc/at.{allow,deny}: wer darf und wer nicht
- Syntax: at [-V] [-q queue] [-f file]
[-mldbv] TIME
- Beispiel:

```
$ at teatime friday
at> wall 'Feierabend'
at> <EOT>
job 1 at 2004-06-18 16:00
$
```

syslogd

- Systemlogfiles über syslogd
- Konfigurationsfiles: `/etc/syslog.conf` (ohne d!)
- Beispielzeilen:

```
auth,authpriv.*                /var/log/auth.log
*. *;cron,auth,authpriv.none    -/var/log/messages
*. *;cron,auth,authpriv.none    -/dev/tty10
*. *;cron,auth,authpriv.none    @hopser.hotaru.it.F
```

- Logfiles: `/var/log/*` (nach FHS)
- `logrotate`
- `tail -f -F`

Backup

```
Usage: cpio {-o|--create} [-0acvABLV] [-C bytes]
        [-H format] [-M message]
        [-O [[user@]host:]archive]
        [-F [[user@]host:]archive] [...]
        < name-list [> archive]
cpio {-i|--extract} [-bcdfmnrtsuvBSV]
        [-C bytes] [-E file] [-H format]
        [-M message] [-R [user][:][group]]
        [-I [[user@]host:]archive]
        [-F [[user@]host:]archive] [...]
        [pattern...] [< archive]
cpio {-p|--pass-through} [-0adlmuvLV]
        [-R [user][:][group]] [...]
destination-directory < name-list
```

 cpio - -help

Backup

- Einfachste Variante: `dd`
- Aufruf: `dd if=/dev/hda of=/dev/hdc bs=64k`
- `dump -# -u -f Ziel /mount-point`
- Level 0: Vollbackup
- Level i : inkrementell zu $i - 1$ ($\forall i > 0$)
- `restore` zum wiederherstellen einzelner oder aller Files
- `tar {-c | -t | -x} [-vzj] [-C directory] [-f file] files/dir...`

Systemzeit

- Informationen zu allen Zeitzone:
`/usr/share/zoneinfo`
- Lokale Zeitzone: `/etc/timezone`
- Link/Kopie der lokalen Zeitzoneinfos:
`/etc/localtime`
- Systemuhr anzeigen: `date`
- Systemuhr setzen: `date [-utc]
MMDDhhmm [[CC]YY] [.ss]`
- Hardwareuhr anzeigen, setzen:
`hwclock [-utc] {-systohc | -hctosys}`

Systemzeit

- Manuelle Synchronisation über Netz:
`ntpdate 131.246.9.116`
- Automatische ständige Synchronisation: `ntpd`
- `ntpd` rechnet Netzwerkschwankungen raus (genauer!)
- `/etc/ntp.conf`:

```
# /etc/ntp.conf, configuration for ntpd
server 131.246.9.116
driftfile /var/lib/ntp/ntp.drift
```

Ende

- Fragen?
- Nächster Vortrag: Dienstag 22. Juni 2004
- Thema: Netzwerkgrundlagen, Netzwerkdienst, Sicherheit