

SHELL WE BEGIN?

EINE EINFÜHRUNG IN DIE SHELL

Max Rosin - mrosin@freitagrunde.org

ANKÜNDIGUNGEN

lists.freitagrunde.org/mailman/listinfo/fopen
freitagrunde.org/TechTalks

WAS MACHEN WIR HEUTE?

- Was ist eine Shell?
- Terminal Emulatoren
- Linux Verzeichnisstruktur
- Dateirechte
- Root vs. User
- Grundlegende Kommandos
- Pipes
- stdin, stderr, stdout
- (Umgebungs-)Variablen
- Fallunterscheidungen
- Schleifen
- Scripting

WAS IST EINE SHELL?

*A shell is a program that provides the traditional, **text-only user interface** for Linux and other Unix-like operating systems. Its primary function is to read commands that are typed into a console or terminal window in a GUI and then execute them.*

<http://www.linfo.org/shell.html>

TERMINAL EMULATOREN

- xterm
- urxvt
- gnome-terminal
- yakuake
- guake
- terminator

VERSCHIEDENE SHELLS

- sh
- bash
- zsh
- ash
- dash
- ksh
- fish

DER PROMPT

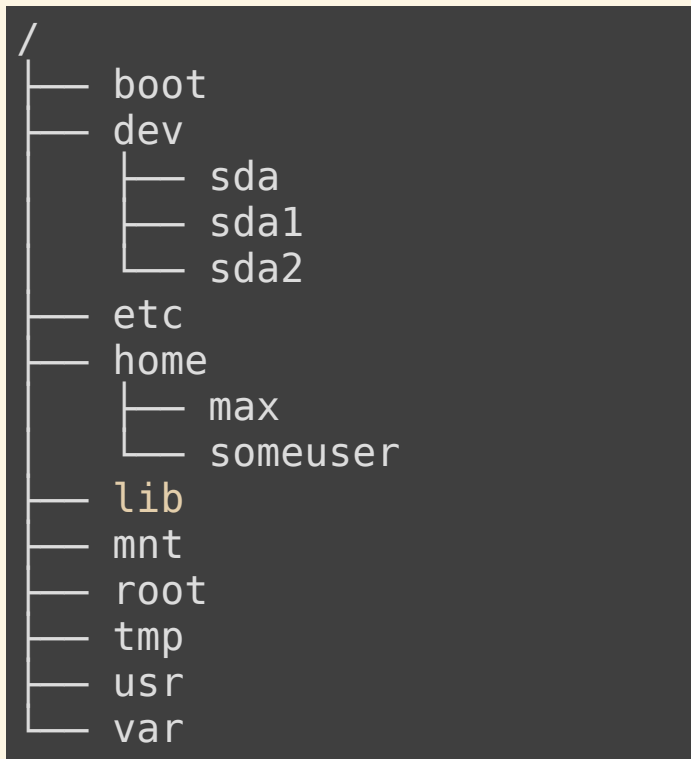
```
[max@gaia ~]$
```

max Username

gaia Hostname

~ aktueller Ordner

LINUX VERZEICHNISSTRUKTUR



Pfad	Bedeutung
~	Referenz auf Home-Verzeichnis
~name	
/home/max/	absoluter Pfad
max/foo.txt	relativer Pfad
.	Verweis auf aktuellen Ordner
..	Verweis auf übergeordneten Ordner

DATEIRECHTE

```
[max@gaia ~]$ ls -lh /home/max
total 104K
drwxr-xr-x 3 max max 4.0K Jan 19 11:30 Privat
drwxr-xr-x 3 max max 4.0K Jan 16 14:04 Studium
-rw-r----- 1 max max 36K  5. Apr 2015  foo.txt
```

- d = directory
- r = read
- w = write
- x = execute

ROOT VS. USER

- Root kann jedes noch so kleine Detail am System verändern (und kaputt machen).
- User können nur auf ihren eigenen Dateien arbeiten.
- Werden eigentlich in unprivileged und privileged user unterschieden.

Immer als User arbeiten und nur wenn nötig als Root!

GRUNDLEGENDE KOMMANDOS: NAVIGATION

Befehl	Beschreibung
pwd	gibt aktuellen Ordner aus
cd	Wechselt das Verzeichnis

GRUNDLEGENDE KOMMANDOS: DATEIEN

Befehl	Beschreibung
ls	gibt den Ordnerinhalt aus
touch	legt Datei an
rm	löscht Ordner/Dateien
cp	kopiert Ordner/Dateien
mv	verschiebt Ordner/Dateien
mkdir	legt Ordner an
rmdir	löscht Ordner

GRUNDLEGENDE KOMMANDOS: NOCH EIN PAAR

Befehl	Beschreibung
echo	gibt Text aus
cat	gibt Datei aus
grep	durchsucht (grept) Datei(en)

PRAKTISCHE TOOLS

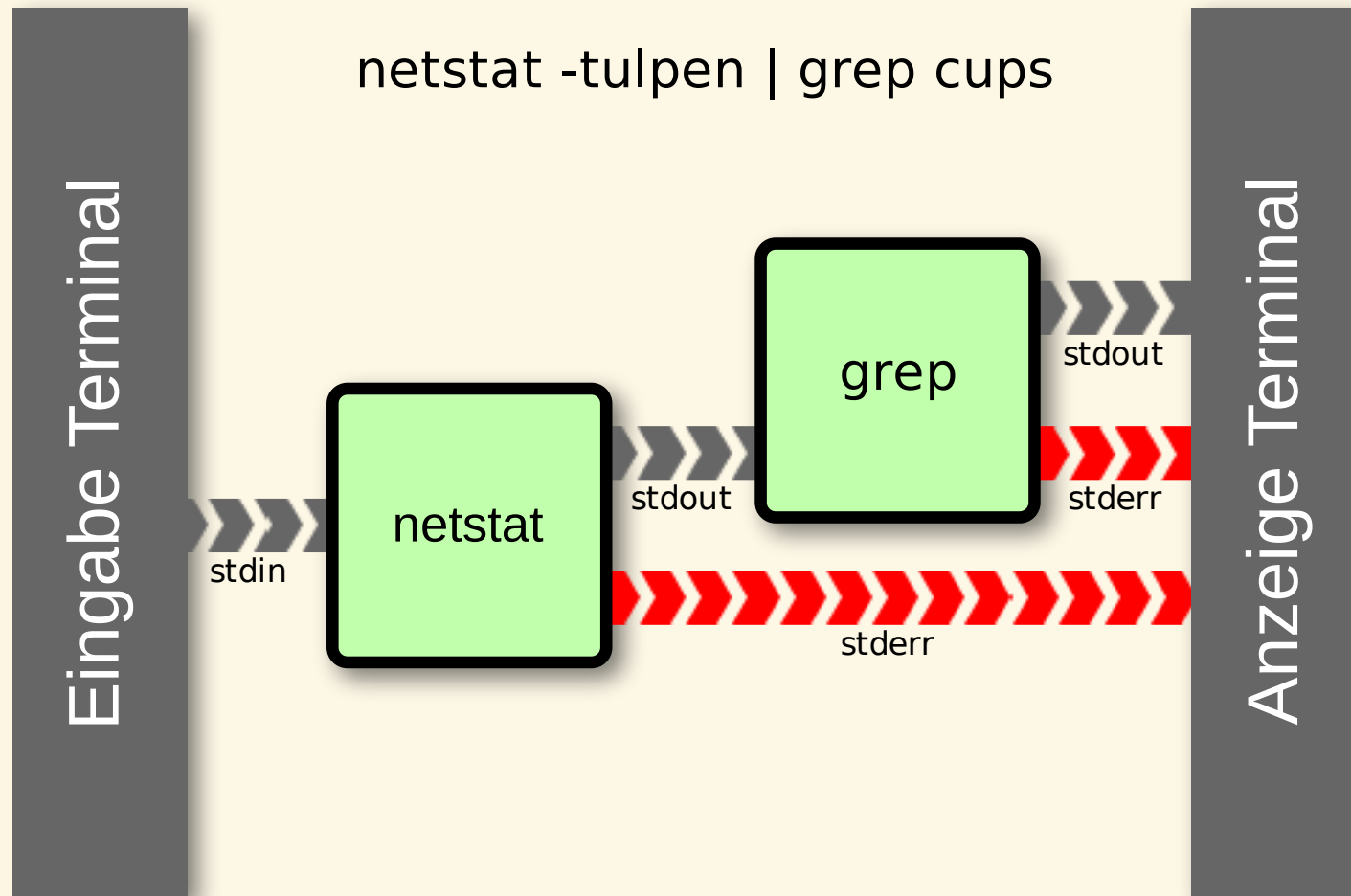
Befehl	Beschreibung
less	Text-Dateien angucken
man	Hilfe zu Kommandos anzeigen
chmod	Dateirechte ändern
tar/zip/unzip	Archive erstellen/entpacken
wget/curl	Dateien runterladen

PIPES

- das ist eine Pipe: |
- verbinden *stdout* und *stdin* zweier Kommandos
- lassen sich verketteten

```
$ ls -lh /home/max/ | grep -i studium  
drwxr-xr-x 3 max max 4.0K Jan 16 14:04 Studium
```

STDIN, STDOUT, STDERR



VARIABLEN

- Variablen speichern Werte
- *var* bei Zuweisung
- *\$var* bei Verwendung

```
text="Hallo Welt"  
zahl=42
```

```
echo "$text"  
echo "$zahl"
```


UMGEBUNGSVARIABLEN

- sind für gesamte Prozessumgebung definiert
- verhalten sich ansonsten wie normale Variablen
- werden mit *export* definiert
- werden oft zum Setzen von Einstellungen genutzt

```
export FORCE_SSL=1
```

FALLUNTERSCHIEDUNGEN

```
foo="bar"

if [[ "$foo" == "bar" ]]; then echo "Yes"; else echo "No"; fi

if [[ "$foo" == "bar" ]]; then
    echo "Yes"
else
    echo "No"
fi
```

```
foo=2

if [[ $foo > 3 ]]; then echo "Yes"; else echo "No"; fi

if [[ $foo > 3 ]]; then
    echo "Yes"
else
    echo "No"
fi
```

SCHLEIFEN

FOR - SCHLEIFE

```
for i in a foobar 2 ab; do  
    echo "$i"  
done
```

WHILE - SCHLEIFE

```
while true; do  
    ls /home/max/  
    sleep 2  
done
```

AUSGABE VON KOMMANDOS VERWENDEN

```
output="$(grep -i error /var/log/syslog)" # bitte benutzen
```

oder

```
output="`grep -i error /var/log/syslog`" # veraltet
```

SCRIPTING

```
#!/usr/bin/env bash

SEARCH_DIR="/home/max/passwordfiles"
TEST_FILE="/tmp/test42"

if [[ -a "$TEST_FILE" ]]; then
    rm -f "$TEST_FILE"
fi

touch "$TEST_FILE"

files="$(ls $SEARCH_DIR)"
for x in $files; do
    grep password "$SEARCH_DIR/$x" >> "$TEST_FILE"
done

echo "Done!"
```

GEFÄHRLICHE KOMMANDOS

```
:(){ :|:& };;:
```

```
rm -rf /
```

```
wget http://microsoft.com/windows_installer.sh | sh
```

```
$(echo 0igpeyA6fDomIH070go= | base64 -d)
```

Den größten Schaden kann man meist verhindern, indem man als User (und nicht als Root) arbeitet.

AUSBlick AUF NÄCHSTEN TALK

- Alltag mit der Shell
- Tipps und Tricks
- 29.01.2016
- 19 Uhr!
- hier im MAR 0.001

FRAGEN?

- Kontakt: mrosin@freitagsrunde.org
- Slides: freitagsrunde.org/TechTalks
- Video: youtube.com/user/Freitagsrunde

ANKÜNDIGUNGEN

- lists.freitagsrunde.org/mailman/listinfo/fopen
- freitagsrunde.org/TechTalks