

Einführung in die Kommandozeile

...oder auch: Hier hat alles angefangen.

Tom Priebe

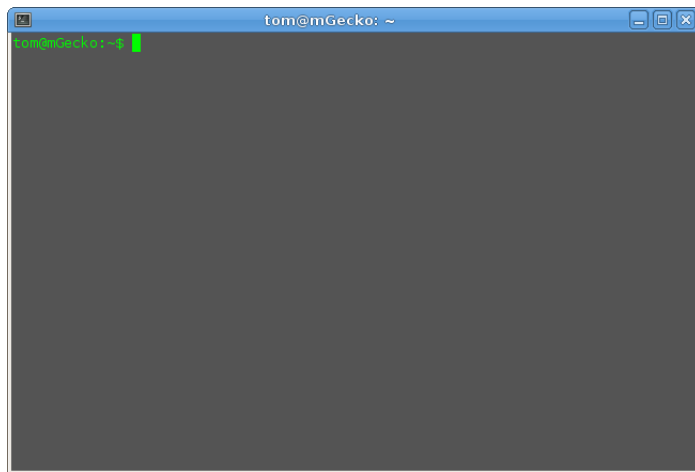
Freitagsrunde – Technische Universität Berlin

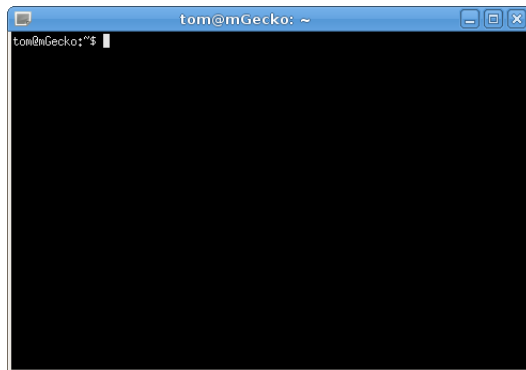
28. Oktober 2011

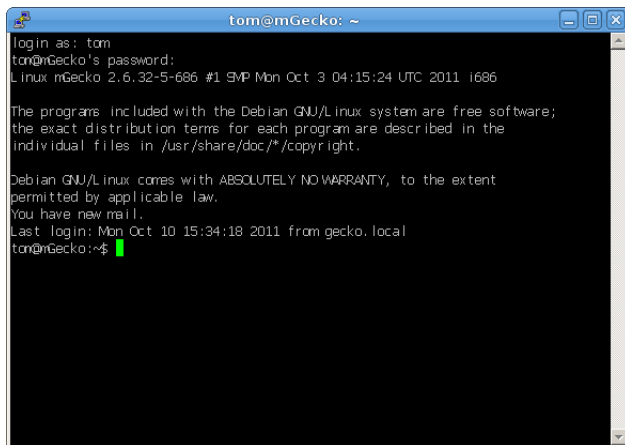


Inhalt

- 1 Einführung
 - Über die Kommandozeile
 - Grundbefehle
 - Parameter
- 2 Kombinierte Befehle
 - Sequenzen
 - Umleitung der Ausgabe
- 3 Troubleshooting
 - Hardware
 - Software
- 4 Bash-Scripting
 - Variablen
 - Schleifen







```
tom@mGecko: ~  
login as: tom  
tom@mGecko's password:  
Linux mGecko 2.6.32-5-686 #1 SMP Mon Oct 3 04:15:24 UTC 2011 i686  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
You have new mail.  
Last login: Mon Oct 10 15:34:18 2011 from gecko.local  
tom@mGecko:~$
```

- Anwendungen ↔ Zubehör

- `ssh username@furor.cs.tu-berlin.de`

- Strg + Alt + F1

- Alt + F2 ↔ `Terminal`

Mauslose Bedienung!

Durch die letzten Befehle scrollen ↑, ↓

Bildschirm scrollen Shift + Bild↑, Bild↓

Befehl/Dateiname vervollständigen Tabulator

Anzeige des Inhalts eines Ordners

Eingabe

```
ls
```

Ausgabe

```
bin Makefile src
```

Wechsel in einen Ordner

Eingabe

```
cd src
```

Ausgabe

Kopieren einer Datei

Eingabe

```
cp terminal.log /home/tom/Documents
```

Ausgabe

Dateien löschen

Eingabe

```
rm /home/tom/Documents/terminal.log
```

Ausgabe

Ordner anlegen

Eingabe

```
mkdir test
```

Ausgabe

Beliebige Programme können so gestartet werden:

- `wget http://ptrace.fefe.de/linuxtag.jpg`
- `nano test/testdatei.txt`

Ordner mit Inhalt löschen

Eingabe

```
rm test
```

Ausgabe

```
cannot remove 'test': Is a directory
```

Lösung

```
rm -r test
```

Kombinierte Parameter

Eingabe `ls -lah --sort=size`

Ausgabe ...eine ganze Menge

Alternative `ls -l -a -h --sort=size`

Mehr Infos

```
man <programmname>
```

Beenden mit q

Inhalt

- 1 Einführung
 - Über die Kommandozeile
 - Grundbefehle
 - Parameter
- 2 Kombinierte Befehle
 - Sequenzen
 - Umleitung der Ausgabe
- 3 Troubleshooting
 - Hardware
 - Software
- 4 Bash-Scripting
 - Variablen
 - Schleifen

```
mkdir test ; cd test
```

```
mkdir test && cd test
```

```
./configure && make && sudo make install
```

```
(mkdir test && cd test) || (mkdir test2 && cd test2)
```

```
ls -R | grep src
```

- Alle Dateien auflisten (rekursiv)
- Statt auf dem Terminal die Ausgabe in `grep` umleiten
- Nur die Verzeichnisse anzeigen, in denen `src` vorkommt

```
du | sort -n
```

- Den Festplattenverbrauch aller Dateien rekursiv anzeigen
- Statt auf dem Terminal die Ausgabe in `sort` umleiten
- Die Dateien nach Größe sortieren

```
ssh tom@stan << cat update.sh
```

- Logge dich via SSH als tom auf stan ein
- Zeige den Inhalt von update.sh an
- Leite den Inhalt von update.sh in die Konsole von stan um

```
echo "Frunde++" > datei.txt
```

- Gib „Frunde++“ aus
- Leite die Ausgabe in die Datei datei.txt um

Inhalt

- 1 Einführung
 - Über die Kommandozeile
 - Grundbefehle
 - Parameter
- 2 Kombinierte Befehle
 - Sequenzen
 - Umleitung der Ausgabe
- 3 **Troubleshooting**
 - **Hardware**
 - **Software**
- 4 Bash-Scripting
 - Variablen
 - Schleifen

Probleme mit Peripherie

- ...wird nicht erkannt
- ...kann nicht gemounted werden
- ...bringt das System durcheinander

`dmesg`

- Anzeige der Kernelnachrichten
- Inclusive Fehler-/Verbindungsmeldungen

Anzeige aller angeschlossenen USB-Geräte

```
lsusb
```

Anzeige aller angeschlossenen PCI-Karten

```
lspci
```

Probleme mit Programmen

- ...die einfrieren
- ...die den PC verlangsamen
- ...die sich einfach nicht beenden wollen

Schritt 1

```
ps -e | grep <Programmname>
```

- alle Prozesse auflisten
- nur die Prozesse (und ihre ID) anzeigen, die <Programmname> heißen

Schritt 2

```
kill <Prozessid>
```

Inhalt

- 1 Einführung
 - Über die Kommandozeile
 - Grundbefehle
 - Parameter
- 2 Kombinierte Befehle
 - Sequenzen
 - Umleitung der Ausgabe
- 3 Troubleshooting
 - Hardware
 - Software
- 4 Bash-Scripting
 - Variablen
 - Schleifen

```
LIP = „Linux Install Party“
```

Zuweisung

```
echo $LIP
```

Ausgabe

```
export LIP
```

Anderen Programmen zur Verfügung stellen

Umgebungsvariablen

Variable	Bedeutung
<code>\$PATH</code>	Programmverzeichnisse
<code>\$LANG</code>	Locale
<code>\$HOME</code>	Home-Verzeichnis

Nützlichkeiten in Shellsripten

Variable	Bedeutung
\$?	Rückgabewert
\$#	Anzahl der Parameter
\$1	1. Parameter
@	Alle Parameter als Array

```
$(id -u)
```

Eigene BenutzerID als Variable

```
$(pidof ssh)
```

ProgrammID von laufenden Secure-Shell

```
[ $(id -u) -ne 0 ]
```

Bedingungen

```
if [ $(id -u) -ne 0 ]; then
```

```
    echo „Be root!“
```

```
fi
```

If-Statement

```
while [ who | grep -c „root“ -ne 1 ]; do  
  
    sleep 2;  
  
done  
  
echo „Vorsicht! Admin anwesend!“
```

```
for i in {0..10}; do
```

```
  echo „Pepsi ist besser als Coca Cola“
```

```
done
```

Danke && Viel Spaß