

# Linux Befehle

## Allgemeine Befehle

### Prozesse/Programme

Unter Unix/Linux wird jedes Programm mindestens ein Prozess gestartet. Diese Prozesse können mittels `ps` <sup>1)</sup> angezeigt werden. Der Befehl hierzu lautet.

```
ps aux
```

Listet alle Prozesse auf, die im System laufen. Dabei werden alle Informationen angezeigt.

### Suchen von Strings ((Zeichenketten))

Da die Liste sehr lang werden kann, ist eine anschließende Filterung auf den gesuchten Prozess sinnvoll. Der `grep`-Befehl filtert den übergebenen Textbereich nach dem vorgegebenen String. Dies kann wie folgt durchgeführt werden.

```
ps aux | grep -i STRING
```

| <sup>2)</sup>: Ausgabe <sup>3)</sup> von `ps` wird weitergegeben

STRING: gibt die Zeichenkette an, nach der man suchen möchte

-i <sup>4)</sup> lässt `grep` die Groß-/Kleinschreibung ignorieren

### Arbeiten im Terminal bash (Kommandozeile)

Auf der Kommandozeile (terminal) kann man komfortabel mit den Cursor-Pfeilen innerhalb der bereits eingegebenen Befehle navigieren. Dies wird allerdings schnell unpraktisch, wenn man sehr viele Befehle durchwühlen muss. Schneller geht es gezielt nach einem bereits eingegebenen Kommando innerhalb der History zu suchen. Dies kann man mit `STRG+R` anstoßen. Es erscheint:

```
(reverse-i-search)` `:
```

Nun kann man Teile des Befehls eingeben und der gefundene Befehl wird sofort eingeblendet.

TIPP: Mit `TAB` lässt sich auf der Kommandozeile fast alles ergänzen. Ist der bereits eingegebene Teil eindeutig, so wird dieser nach drücken auf `TAB` ergänzt. Ist dieser noch nicht eindeutig, so kann man sich mit erneuten `TAB` die Möglichkeiten anzeigen lassen. Teilweise funktioniert dies auch bei Parametern von Befehlen.

## Kurzeinführung in Skripting

Um immer wiederkehrende Aufgaben zu automatisieren, können sogenannte Skripte verwendet werden. Diese Skripte sind Textdateien, die z.B. Befehlsabfolgen enthalten, die normalerweise sequenziell auf der Kommandozeile eingegeben würden. Beim Ausführen werden diese Texte von einem Interpreter<sup>5)</sup> analysiert und die enthaltenen Befehle werden ausgeführt. Die einfachste Variante sind die bash-Skripte, da hier die Befehle wie auf der Kommandozeile eingegeben werden. Die Kommandozeile ist nämlich nichts anderes als das Programm /bin/bash.

Am Anfang eines Skriptes muss man Linux mitteilen, mit welchem Interpreter der folgende Text analysiert werden soll. Dies geschieht durch einen speziellen Kommentar<sup>6)</sup>.

test.sh

```
#!/bin/bash
echo "Das ist ein Test und gibt nur diesen Text aus."
echo "Jetzt wird der Ordnerinhalt angezeigt:"
ls -l
```

Damit das Skript von Linux überhaupt ausgeführt werden kann, muss es zunächst als ausführbar gekennzeichnet werden (s.o.).

```
chmod +x test.sh
```

Dabei bedeutet chmod so viel wie change modus und +x setzt das eXecute Bit. Danach wird die Datei als ausführbar gelistet, was unter Ubuntu standardmäßig mit einer grünen Schrift angezeigt wird. Ausführen lässt sich das Skript nun, indem man den vollständigen Pfad angibt. Wenn man sich im Ordner befindet, in dem das Skript gespeichert wurde, wäre dies:

```
./test.sh
```

Mit ./ ist der aktuelle Pfad gemeint<sup>7)</sup>. Alternativ kann man ein Skript auch in einen Ordner des Default-Suchpfades<sup>8)</sup> also z.B. /home/USER/.bin/HIERHIN

Die zweite Variante hängt aber vom jeweiligen System ab.

## Netzwerk Bereich

### Tabelle mit allen Befehlen

| Befehl   | Bedeutung  | Beispiel       |
|----------|--|----------------|
| ps       | Listet alle Prozesse auf                               | ps aux         |
| grep     | Suche nach der Zeichenkett STRING                      | grep -i STRING |
| route    | Ausgabe und setzen von Routen                          | route -n       |
| ifconfig | Ausgabe und Änderung einer Schnittstelle <sup>9)</sup> | ifconfig eth0  |

| Befehl    | Bedeutung   | Beispiel  |
|-----------|---|---|
| ip        | Mächtiger Befehl, der route und ifconfig abdeckt.   | <a href="#">hier zu IPv6</a> und <a href="#">hier</a> |
| locate    | Suche nach Programm FILENAME im Dateisystem   | locate FILENAME                                       |
| blkid     | Anzeigen der Block-ID der einzelnen Partitionen; hier sda2                                  | blkid /dev/sda2                                       |
| nc        | netcat ist das „Schweizertaschenmesser“ im Netzwerkbereich                                  | <a href="#">Beispiele</a>                             |
| pidof     | Prozess-ID feststellen  | pidof apache  |
| apt-get   | Installationsprogramm; vgl. mit Programm-Verzeichnis  | apt-get install<br>PACKAGENAME                        |
| apt-cache | Hilfprogramm zu apt-get; Gibt Zugriff Informationen zu installierten/verfügbaren Programmen | apt-cache search<br>SUCHBEGRIFF                       |

## NFS-Freigaben

Konfigurierte Freigaben auf dem Server werden unter Ubuntu mit dem folgenden Befehl exportiert.

```
sudo exportfs -ra
```

Überprüfen kann man die Freigaben durch:

```
showmount -e <nfs-server>
```

## Festplattenbelegung

Mit dem Tool baobab lässt sich graphisch sehr anschaulich der belegte Festplattenbereich anzeigen und so ggf. Platz schaffen.

## Routing-Tabelle einrichten mit route

Um die Routing-Tabelle unter Linux auszugeben können die folgenden Befehle verwendet werden

```
netstat -nr
```

```
route -n
```

Der Schalter -n unterdrückt die standardmäßige Namensauflösung. Der Schalter kann selbstverständlich entfallen.

Hinweis: Um einen Ubuntu-Rechner zum Router zu machen, muss das sogenannte IP-Forwarding <sup>10)</sup> aktiviert sein. Dies kann mit dem folgenden Befehl erfolgen:

```
sudo sysctl net.ipv4.ip_forward=1
```

Um zu überprüfen, ob das IP-Forwarding aktiviert ist, kann folgender Befehl verwendet werden:

```
cat /proc/sys/net/ipv4/ip_forward
```

## Temporäre statische Routen einrichten

Mit dem folgenden Befehl lässt sich eine statische Route hinzufügen. Diese Route existiert wie bei Windows nur bis zum nächsten Reboot. Die Schalter sind selbsterklärend, daher wird hier auf eine detaillierte Beschreibung verzichtet.

```
sudo route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254 wlan0
```

Der Schalter `-host` kann für Host-Routing Einträge verwendet werden. Dabei weiß das System, dass die Netzmaske `255.255.255.255` verwendet werden muss, sodass `netmask` entfallen kann.

```
sudo route add -host 10.0.0.1 gw 192.168.1.254 eth0
```

Zur Anpassung des Default-Gateway lässt sich folgender Befehl verwenden. Hier wird das `wlan0` Interface verwendet.

```
sudo route add default gw 192.168.1.1 wlan0
```

Um eine Route dauerhaft zu speichern, kann diese in die Datei `/etc/rc.local` (unter anderen Linux Derivaten kann diese auch in `/etc/rc.d/rc.local` abgelegt sein) hinterlegt werden. Diese Datei wird nach dem Boot-Vorgang als Letztes aufgerufen (vergleiche `autoexec.bat` unter DOS).

## Löschen von Routen

Mit dem folgenden Befehl kann eine statische Route gelöscht werden.

```
sudo route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254 wlan0
```

## ip als Alternative zu ifconfig und route

Im Folgenden werden eine Reihe von Befehlen aufgelistet, die zum Teil ähnliche Funktionen erfüllen wie `ifconfig` bzw. `route`. Sie verwenden den Basis Befehl `ip` mit entsprechenden Optionen.

### Status (ausführlich) einer Schnittstelle anzeigen

#### Befehl:

```
ip -s link
```

#### Bedeutung:

Zeigt den Status einer Schnittstelle `eth0` sowie eine Statistik zu den übertragenen Daten an.

Hinweis: Ohne den Schalter `-s` wird nur der Schnittstellenstatus angezeigt.

#### Alternative zu:

```
ifconfig /all
```

## IP-Adresse setzen

### Befehl:

```
ip addr set 192.0.0.1/24 broadcast 192.0.0.255 dev eth0
```

### Bedeutung:

Setzt IP-Adresse 192.0.0.1 mit der Netzmaske /24 und der Broadcast-Adresse 192.0.0.255 auf der Schnittstelle eth0

### Alternative zu:

```
ifconfig eth0 192.0.0.1 netmask 255.255.255.0 broadcast 192.0.0.255
```

## Zusätzliche IP-Adresse zu einer Schnittstelle hinzufügen

### Befehl:

```
ip addr add 192.0.0.2/24 broadcast 192.0.0.255 dev eth0
```

### Bedeutung:

Setzt eine weitere IP-Adresse 192.0.0.2 als Alias auf die Schnittstelle eth0.

### Alternative zu:

```
ifconfig eth0:1 192.0.0.2/24
```

Hinweis: Mittels `ip addr [change|replace|del]` lassen sich die bestehenden Adressen manipulieren. Wobei jeweils nur ein Schlüsselwort aus der eckigen Klammer zum Einsatz kommt.

## Routen anzeigen

### Befehl:

```
ip route
```

### Bedeutung:

Es werden alle gesetzten Routen angezeigt.

### Alternative zu:

```
route
```

Hinweis: Mit der Option `-6` werden alle IPv6 Routen angezeigt.

## Default-Gateway einrichten

### Befehl:

```
ip route add default via 192.0.0.1
```

### Bedeutung:

Es werden alle gesetzten Routen angezeigt.

### Alternative zu:

```
route add default gw 192.0.0.1
```

Hinweis: Der Befehl route kann nur genau ein Default-Gateway setzen. Der ip-Befehl ist in der Lage mehrere Default-Gateways einzurichten s. 5.6.

## Weitere Default-Gateways einrichten

### Befehl:

```
cat „1 table2, „ » /etc/iproute2/rt_tables  
ip route add 10.0.0.1/24 dev eth1  
src 10.0.0.20 table table2  
ip route add default via 10.0.0.1 dev eth1 table table2  
ip rule add from 10.0.0.20/32 table table2  
ip rule add to 10.0.0.20/32 table table2
```

### Bedeutung:

Spezialfall: Pro Schnittstelle werden eigene Default-Gateways eingerichtet.

### Alternative zu:

Hierzu gibt es keine Alternative, da dies s.o. mit dem alten route-Befehl nicht möglich ist.

- 1) ps: process show
- 2) pipe
- 3) STDOUT: Standard Output; es gibt noch STDIN (Standard Input) und STDERR (Standard Error)
- 4) ignor case
- 5) z.B. bash oder perl
- 6) # ist das Kommentarzeichen; alles dahinter wird vom Interpreter selbst ignoriert

7)

vgl. `pwd`

8)

vgl. PATH-Umgebungsvariable

9)

if: **Interface**

10)

IP-Weiterleitung zwischen zwei Schnittstellen

From:

<http://www.kopfload.de/> - **kopfload - Lad Dein Hirn auf!**

Permanent link:

[http://www.kopfload.de/doku.php?id=network:linux\\_cmds&rev=1385654280](http://www.kopfload.de/doku.php?id=network:linux_cmds&rev=1385654280)

Last update: **2025/11/19 16:12**

