

# Linux-Befehle

Nikola Mihaylov

September 12, 2024

## Contents

1	clear	3
2	pwd	3
3	--help	3
4	man	3
5	cd	3
6	ls	4
7	mkdir	4
8	touch	4
9	mv	4
10	su -	4
11	whoami	6
12	history	6
13	!n	6
14	reboot	6
15	shutdown	6
16	sudo	7
17	apt	7
18	ssh	7
19	systemctl	8
20	nano	8
21	ip	8
22	docker	9

## 1 clear

Löscht den Inhalt der aktuellen Terminalbildschirm. Kann auch mit der Tastenkombination Strg+L ausgeführt werden.

```
clear
```

## 2 pwd

Gibt den absoluten Pfad zum aktuellen Arbeitsverzeichnis aus.

```
pwd
```

## 3 --help

Allgemeines Flag, das an das Ende von Befehlen angehängt werden kann, um eine kurze Hilfeseite mit Optionen und der richtigen Syntax eines Befehls anzuzeigen.

```
nano --help
```

## 4 man

Handbuchseite für einen Befehl. Entspricht -help, ist aber viel detaillierter und scrollbar.

```
man nano
```

## 5 cd

Wechselt das Verzeichnis, akzeptiert entweder einen relativen oder absoluten Pfad.

`cd ..` — navigiert zum "oberen" Verzeichnis im Verhältnis zum aktuellen

`cd /etc/ssh` — navigiert zum absoluten Verzeichnis (beachten Sie den /, der das Stammverzeichnis als Startpunkt angibt)

`cd Downloads` — navigiert zum relativen Verzeichnis, dieses Beispiel geht davon aus, dass das aktuelle Arbeitsverzeichnis `home/~` ist).

## 6 ls

Listet Dateien und Verzeichnisse im aktuellen Arbeitsverzeichnis alphabetisch auf. Standardmäßig werden versteckte Dateien und Verzeichnisse nicht angezeigt.

`ls`

`ls -a` — listet alle normalen Dateien und Verzeichnisse sowie alle versteckten Dateien und Verzeichnisse auf, die durch einen Punkt (.) am Anfang ihres Namens gekennzeichnet sind.

## 7 mkdir

Erstellt ein Unterverzeichnis relativ zum aktuellen Arbeitsverzeichnis, sofern kein absoluter Pfad angegeben ist.

`mkdir test` — erstellt relativ zum aktuellen Arbeitsverzeichnis ein Unterverzeichnis "test".

`mkdir /home/padawan/test` — Erstellt dasselbe "test"-Verzeichnis, jedoch mit dem absoluten Pfad, um den spezifischen Speicherort des Verzeichnisses anzugeben.

## 8 touch

Erstellt eine leere Datei an einem absoluten oder relativen Speicherort.

`touch config.txt` — erstellt eine Datei namens `config.txt` relativ zum aktuellen Arbeitsverzeichnis. Ein absoluter Speicherort kann ebenfalls angegeben werden, ähnlich der `mkdir`-Syntax von vorhin.

## 9 mv

Verschiebt Dateien und Verzeichnisse oder benennt sie um.

`mv /Downloads/docker-compose.yml /var/docker/Container/docker-compose.yml` — verschiebt eine Quelldatei links zu einem Ziel rechts. Dies kann auch mit Verzeichnissen anstelle von Dateien erfolgen.

`mv /Downloads/docker-compose.yml /Downloads/test-docker-compose.yml` — Wenn sich die Datei am selben Speicherort befindet, aber einen anderen Namen hat, wird sie stattdessen umbenannt.

## 10 su -

Ermöglicht das Wechseln zwischen Benutzern in der aktuellen Terminal-Sitzung. Durch Angabe eines Bindestrichs wechselt `su` zum Root-Benutzerkonto.

`su -`

## 11 whoami

Gibt den Benutzernamen der aktuellen Terminal-Sitzung aus.

```
whoami
```

## 12 history

Gibt eine chronologisch nummerierte Liste der im Shell ausgeführten Befehle aus. Die Abkürzung ist Strg+r, was eine neue Suchaufforderung für die Verlaufsdatei öffnet.

```
history
```

## 13 !n

Kann mit `history` kombiniert werden, um einen Befehl basierend auf seiner chronologischen Nummer in der History-Ausgabe zu wiederholen.

`!20` — dieser entnimmt der Ausgabe von `history` den mit der Nummer 20 bezeichneten Eintrag und führt den gleichen Befehl aus.

## 14 reboot

Plant einen Systemneustart. Erfordert Superuser-Berechtigungen.

```
reboot
```

## 15 shutdown

Plant ein Herunterfahren des Systems. Erfordert Superuser-Berechtigungen.

`shutdown` — Das Ausführen des Befehls ohne Optionen führt zu einer Verzögerung des Herunterfahrens von etwa einer Minute.

`shutdown now` — Dieser Befehl versucht, das System sofort und mit minimaler Verzögerung herunterzufahren.

`shutdown -h 13:45` — ein 24-Stunden-Format als Herunterfahranweisung akzeptiert, wird das System ausgeschaltet, wenn die Uhr die angegebene Zeit anzeigt.

## 16 sudo

Gewährt dem Benutzer vorübergehende Administratorrechte (Superuser) für die Ausführung eines Befehls, der solche Rechte nutzt. Dieser Befehl muss explizit unter Debian installiert werden und erfordert, dass der Root-/Administratorbenutzer einen normalen Benutzer zur Gruppe "sudo" hinzufügt, damit er funktioniert.

`sudo nano /etc/ssh/sshd_config` — führt `nano` mit Superuser-Befugnissen aus, während eine geschützte Datei im Verzeichnis `/etc/` geöffnet wird, da andernfalls keine Änderungen gespeichert werden können.

## 17 apt

Paketverwaltungsprogramm für Debian-basierte Distributionen. Funktioniert als Abstraktion für `dpkg` und kann durch `apt-get` ersetzt werden. Erfordert Superuser-Berechtigungen.

`apt update` — aktualisiert die Metadaten aller verfügbaren Pakete aus dem Repository. Empfohlen vor dem Starten anderer `apt`-Befehle

`apt search vim` — durchsucht die Repository-Metadaten nach einem Paket, das als Argument für den Befehl angegeben wird (in diesem Beispiel wird das Paket `vim` verwendet)

`apt upgrade` — aktualisiert alle installierten Pakete auf die aktuellsten Versionen in Übereinstimmung mit den Metadaten, die mit `update` bezogen wurden

## 18 ssh

Verwendet OpenSSH für die Remote-Verbindung zu einem anderen Computer, auf dem der OpenSSH-Server läuft. Die OpenSSH-Client- und Server-Pakete müssen explizit unter Debian installiert werden und verfügen über zusätzliche Dienstprogramme wie `ssh-keygen`, `ssh-copy-id`, `scp` usw.

`ssh padawan@10.100.26.158` — versucht eine SSH-Verbindung als Benutzer `padawan` unter (@) der IPv4-Adresse `10.100.26.158` herzustellen. Wenn eine Namensauflösung möglich ist, ist auch eine Verbindung unter Verwendung des Hostnamens des Servers statt seiner IPv4 möglich (`ssh padawan@debian-test1`).

`ssh-keygen -t rsa` — generiert SSH-Schlüssel für die Authentifizierung und den passwortlosen Login. Mit `-t` kann eine Verschlüsselungsart definiert werden, in diesem Fall RSA-Verschlüsselung. Standardmäßig werden die Schlüssel im Home-Verzeichnis des Benutzers gespeichert (zum Beispiel `~/.ssh/`).

`ssh-copy-id -i ~/.ssh/id_rsa.pub padawan@10.100.26.158` — kopiert eine angegebene ID-Datei (in diesem Fall die öffentliche Schlüsseldatei, die mit `ssh-keygen` generiert wurde) auf einen Remote-Host, auf dem der OpenSSH-Server läuft. Dies ermöglicht passwortlose Anmeldungen und dient als sicherere Form der Authentifizierung.

## 19 systemctl

Wird zum Verwalten von Hintergrunddiensten verwendet. Erfordert Superuser-Berechtigungen.

**systemctl status sshd** — zeigt den aktuellen Status eines angegebenen Dienstes an (in diesem Fall des OpenSSH-Servers **sshd**). Kann verwendet werden, um zu prüfen, ob ein Dienst ausgeführt wird oder ausgefallen ist und was die Fehlermeldung des ausgefallenen Dienstes ist.

**systemctl start sshd** — startet einen Dienst, der am Ende des Befehls bereitgestellt wird (in diesem Fall des OpenSSH-Servers **sshd**). Im Fehlerfall wird der Benutzer benachrichtigt und aufgefordert, diesen mit **systemctl status** zu überprüfen. Dabei ist zu beachten, dass ein Dienst, der nicht mit **systemctl enable** aktiviert wird, nur für die aktuelle Sitzung aktiviert wird.

**systemctl stop sshd** — stoppt einen laufenden Dienst. Wird ein Dienst mit **systemctl enable** aktiviert, dann wird der Dienst nur für die aktuelle Sitzung gestoppt und beim nächsten Systemstart ohne Benutzereingriff gestartet.

**systemctl enable sshd** — ermöglicht den automatischen Start eines Dienstes beim Hochfahren des Systems.

**systemctl disable sshd** — fungiert als Gegenteil von **enable**, indem es den Dienst als deaktiviert kennzeichnet, um zu verhindern, dass er automatisch mit dem System gestartet wird.

**systemctl enable --now sshd** — eine Verknüpfung, die **systemctl start** und **systemctl enable** zu einem einzigen Befehl kombiniert.

## 20 nano

Ein terminalbasierter Texteditor. Das Paket muss explizit unter Debian installiert werden. Es verfügt über eine eigene Benutzeroberfläche und Tastenkombinationen.

**nano /etc/network/interfaces** — öffnet eine angegebene Datei mit **nano**.

## 21 ip

Zeigt und manipuliert Routing- und Netzwerkgeräte.

**ip address** — funktioniert wie **ipconfig /all** unter Windows. Listet alle an den Computer angeschlossenen Netzwerkgeräte sowie diverse Eigenschaften wie MAC-Adresse, IPv4- und IPv6-Adressen und mehr auf. Kann auch mit **ip addr** und **ip a** abgekürzt werden.

## 22 docker

Befehl zum Verwalten der Docker Engine. Sie muss auf dem System installiert sein, bevor die Befehle verwendet werden können.

`docker compose up -d` — erstellt und startet einen Container basierend auf der Datei `docker-compose.yml` im aktuellen Arbeitsverzeichnis. `-d` startet den Container im Hintergrund.