

- Die hier angegebenen Beispiele zeigen teilweise einen Dialog zwischen dem Benutzer **Klaus** und **Hannes**
- Ausgegangen wird immer vom Rechner, bzw. vom Benutzeraccount von **Klaus**
- Alle allgemeinen Befehle lassen sich selbstverständlich auch als root ausführen

Allgemeine Befehle

Befehl	Erklärung	Beispielsyntax	Kommentar
man	man ruft die sogenannten Manpages (Hilfeseiten) zu diversen Befehlen auf	man rar man -k suchwort	man rar ruft zum Beispiel die Manpage des Programms "rar" auf. Auf dieser sieht man zum Beispiel alle möglichen Befehlserweiterungen und meist auch eine Beispielsyntax, sowie detaillierte Informationen über das Programm. man -k suchwort durchsucht die Manpages aller installierten Programme nach "suchwort", für Anfänger nützlich um Programme zu finden deren Namen man vergessen hat.
su	Wird benötigt, um sich als root einzuloggen, bzw den Benutzer zu wechseln	su su hannes	Ein einfaches su wechselt zum Benutzer "root" und fordert zur Eingabe des root-Passworts auf su hannes wechselt zum Benutzer hannes und fordert zur Eingabe des Passwortes des Benutzers hannes auf
sudo	Ermöglicht das Ausführen einzelner Befehle unter einem anderen Benutzernamen(zB. root) ohne das entsprechende Passwort zu kennen. In der Datei /etc/sudoers steht, wer welche Befehle mit sudo benutzen darf.	sudo halt sudo -u hannes nano	Auch hier wird der Befehl als Benutzer "root" ausgeführt wenn kein anderer Benutzer angegeben wird, sudo halt fährt den Rechner herunter. Das zweite Beispiel öffnet den Texteditor nano als Benutzer hannes.
df	Zeigt den aktuell verfügbaren Speicherplatz auf den Festplatten	df -h	Das -h erzeugt eine für den Menschen besser lesbare Form, da sonst die Angabe in Blöcken erfolgt (meistens 512 oder 1024 Byte), was gerade im Gigabyte-Bereich sehr unleserlich ist
rar	Erzeugt oder entpackt ein RAR-Archiv	rar a -r archivname.rar /home/klaus/zu_packende_dateien/ rar x archivname.rar	a erstellt das Archiv -r ist die Rekursiv-Option, das heißt, der gesamte inhalt des Verzeichnisses wird verarbeitet x entpackt ein Archiv RAR wurde gewählt, weil diese Archive sich im Gegensatz zu .tar.gz auch unter Windows problemlos verarbeiten lassen

scp	Secure copy überträgt Dateien verschlüsselt von einem Linux-Rechner auf den anderen. Es ist sehr praktisch, da niemand auf dem Übertragungsweg die Dateien mitlesen kann.	scp /home/klaus/beispieldatei.dat hannes@192.168.1.23:/home/hannes/ scp -r /home/klaus/ hannes@192.168.1.23:/home/hannes/	Der erste Befehl überträgt eine spezielle Datei des eigenen Rechners auf den Rechner, welcher sich hinter der IP 192.168.1.23 versteckt. Dort wird die Datei dem Benutzer hannes übergeben, folglich wird beim Ausführen dieses Befehls auch dessen Passwort gefordert. Der zweite Befehl macht im Prinzip das Gleiche, dabei wird hier durch die Rekursiv-Option -r nicht eine spezielle Datei übertragen, sondern der komplette Inhalt des Verzeichnisses /home/klaus/
cd	cd ist der Befehl, um die Verzeichnisebene auf der Konsole zu wechseln	cd .. cd /home/klaus/bilder cd bilder	Der Befehl cd .. wechselt in das übergeordnete Verzeichnis, der Befehl cd /home/klaus/bilder wechselt von überall her direkt in das Verzeichnis /home/klaus/bilder. Der Befehl cd bilder wechselt in das Unterverzeichnis bilder, dies funktioniert allerdings nur von der Ebene /home/klaus/ aus
du	du Gibt die Größe eines Ordners an	du -h	Die Option -h formatiert die Angabe in ein für "h"umans lesbaren Ausdruck.
ls	ls gibt den Inhalt eines Verzeichnisses aus	ls	Ausgegeben werden Dateien, sowie auch Unterverzeichnisse, die in dem Verzeichnis liegen, wo man sich gerade befindet. Versteckte Dateien werden ausgelassen! Versteckte Dateien werden erzeugt, wenn man den Dateinamen abändert. Man hängt einfach einen "." vor den Dateinamen
top	top zeigt eine Übersicht der gerade aktiven Prozesse, nebst Speicherbedarf und Beanspruchung der Rechenkapazität	top	
killall	Mit killall lassen sich Prozesse, also auch Programme direkt beenden, ähnlich wie beim Task-Manager unter Windows	killall firefox	Der obige Befehl killt zum Beispiel alle Prozess mit dem Namen firefox
rm	rm wird verwendet um Dateien und Verzeichnisse zu löschen	rm bild.jpg rm * rm -r /home/klaus/bilder/	Der erste Befehl löscht eine Datei, die sich im aktuellen Verzeichnis befindet, direkt. Der zweite Befehl löscht alle Dateien im aktuellen Verzeichnis, jedoch nicht die Unterverzeichnisse und deren Inhalte Der dritte Befehl löscht rekursiv, also das angegebene Verzeichnis komplett mit allen sich darin befindlichen Unterverzeichnissen und Dateien Hinweis: Dieser Befehl löscht unwiderruflich!
mkdir	mkdir erstellt im aktuellen Verzeichnis ein Unterverzeichnis	mkdir bilder mkdir /home/klaus/bilder/party mkdir -p /home/klaus/bilder/party	Der erste Befehl erstellt im aktuellen Verzeichnis ein Unterverzeichnis mit Namen bilder. Der zweite Befehl erstellt von überall her direkt das Unterverzeichnis party im Verzeichnis /home/hannes/bilder/ Der dritte Befehl erstellt den Pfad /home/klaus/bilder/party und legt alle Verzeichnisse an, die nicht existieren.

cp	cp kopiert Dateien und Verzeichnisse an einen anderen Ort	cp /home/klaus/party/bild.jpg /home/klaus/ cp -r /home/klaus/party/ /home/klaus/	Der erste Befehl kopiert die Datei bild.jpg aus dem Verzeichnis /home/klaus/party/ in das Verzeichnis /home/klaus/ Der zweite Befehl kopiert rekursiv, also den gesamten Inhalt des Verzeichnisses /home/klaus/party/ in das Verzeichnis /home/klaus/
mv	mv verschiebt Dateien und Verzeichnisse an einen anderen Ort	mv /home/klaus/party/bild.jpg /home/klaus/ mv -r /home/klaus/party/ /home/klaus/ mv bild.jpg foto_hannes.jpg	Der erste Befehl verschiebt die Datei bild.jpg aus dem Verzeichnis /home/klaus/party/ in das Verzeichnis /home/klaus/ Der zweite Befehl verschiebt rekursiv, also den gesamten Inhalt des Verzeichnisses /home/klaus/party/ in das Verzeichnis /home/klaus/ Eine Besonderheit ist, dass mv auch zum Umbenennen von Dateien verwendet werden kann, hier wird zum Beispiel aus dem Dateinamen bild.jpg der Dateiname foto_hannes.jpg
screen	screen erzeugt eine oder mehrere virtuelle Konsolen. Die Besonderheiten dabei sind, dass sich nach dem Ausloggen bzw. Trennen der Sitzung die Session erhalten bleibt und dass mehrere Personen parallel an einem Terminal arbeiten können.	screen screen -r	Der einfache Aufruf screen erzeugt eine neue Sitzung. screen -r holt nach einer Trennung (Manuell mit CTRL+a d) die Session zurück. screen kann auch direkt als Login-Shell genutzt werden, was sich gerade bei Modemverbindungen gut macht, da eine Neueinwahl den verlassenen Zustand lädt.
ip	ip ist der Nachfolger von ipconfig und dient zur Einrichtung von Netzwerkverbindungen. Als Benutzer darf man sich die Interfaces nur ansehen, erst als root hat man Konfigurationsmöglichkeiten	ip addr show eth0 ip route show eth0 ip neigh show eth0	addr zeigt die IP-Adressen auf den Interfaces, route die bekannten Routen des TCP/IP-Protokolls und neigh ob das Gateway oder andere Ziele erreichbar sind. eth0 ist optional und bei Angabe werden nur die Informationen über dieses Interface (hier das Netzwerkinterface, welches den Namen eth0 trägt) ausgegeben.
route	Zeigt ähnlich ip route show die eingegebenen Routen an.	route route -n	route zeigt alle Ziele an, incl. Namensauflösung per /etc/hosts oder DNS. route -n mach das gleiche ohne Namensauflösung (geht oft schneller).
find	Ein sehr komplexer Befehl mit vielen Optionen zur Suche von Dateien und Verzeichnissen.	find /home/test -iname '*test*	Dieser Befehl sucht im Verzeichniss /home/test alle Dateien mit dem Begriff test im Namen, egal ob groß oder klein geschrieben (Test, TEST, test,)
last	last Zeigt eine Liste der zuletzt eingeloggten Benutzer an.	last	
w	w Zeigt derzeit eingeloggte Benutzer an und was diese momentan tun.	w	

uptime	uptime zeigt die Betriebsdauer des Systems seit dem letzten Neustart an	uptime	
uname	uname Gibt diverse Systeminformationen wie Kernelversion, etc aus.	uname uname -a	
id	id gibt die reale und die effektive ID des Nutzers aus. Dies kann zum Beispiel genutzt werden um zu sehen, ob man Mitglied einer bestimmten Gruppe ist.	id	
cat	cat gibt den Inhalt einer Datei auf dem Bildschirm aus und ermöglicht auch noch weiteres wie Erstellen von Dateien, Hinzufügen von Code zu Dateien, etc. (bitte Manpage von cat lesen)	cat /etc/network/interfaces cat /proc/cpuinfo	Der erste Befehl gibt den Inhalt der /etc/network/interfaces, also die Netzwerkkonfigurationen auf den Bildschirm aus Der zweite Befehl gibt Auskunft über die im Rechner verbaute CPU (Prozessor) Hinweis: cat gibt Inhalte von Dateien in voller Länge auf den Bildschirm aus, man hat demnach bei großen Dateien keine praktische Möglichkeit zum Scrollen (hierzu bitte den less -Befehl beachten)
echo	echo gibt eine Zeichenfolge so wieder, wie sie eingetippt wurde, ändert aber auch ähnlich wie cat den Inhalt von Dateien	echo Hallo Welt! echo Dies ist ein Test > echotest echo Neue Zeile dazugekommen >> echotest	Gibt den Text "Hallo Welt!" wörtlich wieder auf den Bildschirm aus Erstellt eine Datei mit Namen echotest und fügt "Dies ist ein Test" als erste Zeile in diese Datei ein Fügt zur Datei echotest eine weitere Zeile mit dem Inhalt "Neue Zeile dazugekommen" hinzu
tail	tail gibt die letzten n Zeilen einer Datei aus, standard sind 10 Zeilen. Zudem kann man kontinuierlich den neu hinzugekommenen Inhalt einer Datei ausgeben.	tail testdatei tail -n 100 testdatei tail -f testdatei	Der erste Befehl gibt die letzten 10 Zeilen der Datei "testdatei" aus. Der zweite die letzten 100 Zeilen Der dritte Befehl gibt die letzten 10 Zeilen der Datei aus, wartet dann und gibt jede neu hinzukommende Zeile aus. Abbrechen kann man mit [strg]+[C]
head	head gibt die ersten n Zeilen einer Datei aus. Standard sind 10 Zeilen.	head testdatei head -n 100 testdatei	Der erste Befehl gibt die ersten 10 Zeilen von testdatei aus. Der zweite Befehl gibt die ersten 100 Zeilen von testdatei aus.
sort	sort sortiert eine Liste.	cat liste sort cat liste sort -n	Der erste Befehl gibt die Beispieldatei liste alphabetisch sortiert aus, der zweite Befehl sortiert numerisch (nur sinnvoll, wenn die Zeilen in liste mit Zahlen anfangen).

uniq	uniq entfernt aus einer (sortierten) Liste Duplikate.	cat liste sort uniq	Der Befehl gibt Liste alphabetisch sortiert und ohne Duplikate aus.
wc	wc gibt die Zahl der Zeichen, Wörter und Zeilen einer Datei an.	wc liste wc -l liste	Der erste Befehl gibt die Anzahl der Zeichen, Wörter und Zeilen von liste an, der zweite Befehl nur die Anzahl der Zeilen.
chmod	chmod ändert die Rechte an Dateien („change modus“)		Bitte gesonderte Erläuterung weiter unten beachten und das Augenmerk auf die zwei verschiedenen Syntaxvarianten des Befehls richten
chown	chown ändert den Eigentümer einer Datei („change owner“)	chown hannes bild.jpg chown -R hannes /home/klaus/	Ändert den Eigentümer der Datei bild.jpg auf den Benutzer hannes Ändert den Eigentümer des Verzeichnisses /home/klaus/ inklusive deren Unterverzeichnisse mitsamt allen Dateien auf den Benutzer hannes Hinweis: Kopiert man Dateien als root in das eigene home-Verzeichnis, so ist die Verwendung der kopierten Dateien als Benutzer erst sichergestellt, nachdem man per chown den Eigentümer der kopierten Datei vom Benutzer root auf sich selbst gestellt hat
less	less ermöglicht eine Ausgabe des Inhaltes von Dateien ähnlich wie cat , jedoch nicht aus der Standardausgabe wie cat sondern lädt sie in einen scrollbaren Puffer, in dem man auch suchen kann.	less /etc/network/interfaces	Dieser Befehl zeigt den Inhalt der Datei /etc/network/interfaces an und bietet die Möglichkeit mit den Cursor-Tasten zu scrollen. Zusätzlich kann man mit / in der Datei suchen.
dmesg	dmesg gibt den Puffer des Kernels auf den Monitor aus	dmesg dmesg tail dmesg less	Der nachgestellte Befehl legt die Ausgabeart fest. Bei tail werden nur die letzten zehn Zeilen ausgegeben, bei less wird der Puffer mit Scrollmöglichkeit angezeigt (siehe less) dmesg zeigt insbesondere die geladene Hardware an, zeigt Fehler beim Laden der Hardware (defekte Festplatte, o.ä.) sowie den Namen gerade angesteckter USB-Sticks, o.ä.

Befehle per root-Login

Befehl	Erklärung	Beispielsyntax	Kommentar
aptitude update	Aktualisiert die Paketlisten	aptitude update	Aktuelle Paketlisten werden benötigt, damit der Rechner die benötigten Pakete zur Installation auch korrekt abfragen kann, sonst erfolgen Fehlermeldungen.
aptitude	Aktualisiert das System	aptitude safe-upgrade aptitude full-upgrade	aptitude safe-upgrade aktualisiert das System, soweit dies nicht erfordert, zusätzliche Pakete zu installieren oder andere Pakete zu entfernen. aptitude full-upgrade aktualisiert das System, es werden dann also auch neue Pakete installiert und gegebenenfalls Pakete entfernt. Mit älteren aptitude-Versionen (bis einschließlich Etch) muss aptitude upgrade und aptitude dist-upgrade verwendet werden. Einem aptitude safe-upgrade oder aptitude full-upgrade sollte immer ein aptitude update vorausgehen, wenn das automatische Update der Paketlisten nicht konfiguriert ist (siehe <code>/etc/cron.daily/apt</code>), da sonst keine aktuellen Paketlisten verfügbar sind.
aptitude search	aptitude search sucht in den Paketlisten nach verfügbaren Paketen und nennt deren vollen Namen	aptitude search firefox	Der Beispielfehl listet alle Pakete auf, die den Begriff firefox im Paketnamen haben, oder direkt etwas mit dem Browser Firefox zu tun haben
aptitude install	aptitude install installiert Pakete	aptitude install paket1 paket2 paket3 ...	Bei aptitude install kann man mehrere zu installierende Pakete in Reihe angeben, diese werden dann nacheinander installiert. Zu beachten ist, dass nicht nur die angeforderten Pakete selber installiert werden, sondern auch alle anderen, die ein zu installierendes Programm für den sicheren und zuverlässigen Betrieb benötigt.
ifconfig	ifconfig listet die eingestellten Parameter der Netzwerkkarten auf (IP, MAC, Subnetzmaske, etc.)	ifconfig eth0 ifconfig eth0 192.168.0.1 netmask 255.255.255.0	Das eth0 stellt hierbei den Namen der Netzwerkkarte dar, deren Daten ausgegeben werden sollen. Ersetzt man das eth0 durch -a , und tippt somit ifconfig -a ein, so werden dem System bekannten Karten inklusive ihrer Parameter aufgelistet. Der untere Befehl konfiguriert die Netzwerkkarte neu, es wird ihr eine IP-Adresse zugewiesen und die zu verwendende Subnetzmaske festgelegt Hinweis: Die Standardeinstellungen der Karten finden sich in der <code>/etc/network/interfaces</code>
route	route setzt Netzwerkrouuten ("welches Netzwerk ist über welchen Weg erreichbar"). Insbesondere kann man mittels route das Standardgateway setzen.	route add default gw 192.168.0.254 route add 10.13.37.0 netmask 255.255.255.0 dev eth1 route add 10.23.42.0 netmask 255.255.255.0 gw 10.13.37.1	Der erste Befehl setzt das Standardgateway auf 192.168.0.254 . Der zweite Befehl gibt an, dass das Netz 10.13.37.0/24 über die Netzwerkkarte eth1 (lokal) erreichbar ist. Der dritte Befehl gibt an, dass das Netz 10.23.42.0/24 über das Gateway 10.13.37.1 erreicht werden kann (dazu muss 10.13.37.1 selbst über eine andere Route erreichbar sein).

ip	ip ist das Nachfolgeprogramm zu net-tools (das Paket, das unter anderem ifconfig und route enthält). Mit ip ist es unter anderem möglich, mehrere Adressen an eine Netzwerkkarte zu binden.	ip addr add 192.168.0.1/24 dev eth0 ip route add default via 192.168.0.254 ip route add 10.13.37.0/24 dev eth1 ip route add 10.23.42.0/24 via 10.13.37.1	Alle diese Befehle sind äquivalent zu den Beispielen bei ifconfig und route . Statt /24 kann man auch /255.255.255.0 schreiben.
dhclient	dhclient fordert bei erreichbarem DHCP-Server eine vollständige automatische Konfiguration einer Netzwerkkarte an	dhclient eth0	Der DHCP-Server (beispielsweise ein Router, der automatisch IP-Adressen verteilt) antwortet auf diese Anfrage mit einer Neukonfiguration der betreffenden Netzwerkkarte (hier: eth0)
shutdown	Der Befehl shutdown fährt das System herunter	shutdown -h now shutdown -h 30	Die Option -h sagt dem System, dass es sich herunterfahren und abschalten soll, der Parameter now definiert den Ausführungszeitpunkt, also jetzt . Gibt man stattdessen eine Zahl ein (hier: 30) so fährt sich das System erst nach 30 Minuten herunter. Das ist sehr praktisch, wenn man vor dem Einschlafen noch ein wenig Musik hören will, o.ä.

Der Befehl chmod

Die Rechte

Es gibt drei verschiedene Rechte, die man bei einer Datei besitzen, bzw. vergeben kann:

- Das r-Recht - Die Berechtigung eine Datei zu lesen
- Das w-Recht - Die Berechtigung für Schreibzugriff auf eine Datei
- Das x-Recht - Die Berechtigung, eine Datei auszuführen

Syntax

Die Syntax des Befehls chmod lautet:

- **chmod WerWieWas Datei**
- **chmod Oktalzahl Datei**

Durch WerWieWas bzw. Oktalzahl wird die Rechetabelle definiert. Innerhalb von WerWieWas darf es kein Leerzeichen geben.

- *Wer* bezeichnet den Benutzerkreis, dem man Rechte gibt
- *Wie* bezeichnet die Art, in der die Rechte gegeben werden
- *Was* bezeichnet die Rechte als solche. Im einzelnen können folgende Kürzel benutzt werden:

Für Wer kann eines der folgenden Kürzel oder eine Kombination davon stehen:

- **u** - (user) Rechte für den Dateibesitzer
- **g** - (group) Rechte für die Gruppe (man kann Rechte für komplette Benutzergruppen setzen)
- **o** - (others) Rechte für alle anderen Benutzer
- **a** - (all) Rechte für alle Benutzer (entspricht ugo)

Für Was steht eines der folgenden Kürzel:

- + Die Rechte werden zu den vorhandenen zusätzlich vergeben
- - Die im folgenden genannten Rechte werden entzogen
- = Die im folgenden genannten Rechte ersetzen die bisherigen

Für Wie steht eines der zuvor beschriebenen Rechtekürzel, eine Kombination davon oder

- **u** - Vergibt die Rechte so, wie sie beim Besitzer (user) zur Zeit sind
- **g** - Vergibt die Rechte so, wie sie bei der Gruppe (group) zur Zeit sind
- **o** - Vergibt die Rechte so, wie sie bei anderen Benutzern (others) zur Zeit sind

Beispiele

- Will man sich als Besitzer das x-Recht an der Datei testdatei geben, so geschieht das mit **chmod u+x testdatei**
- Der Entzug des w-Rechts für alle Benutzer erfolgt mit **chmod a-x testdatei**
- Will man sich selbst rwx, der Gruppe rx und anderen nur x gewähren, lautet der Befehl **chmod u=rwx,g=rx,o=x testdatei**

Der letzte Fall lässt sich wesentlich eleganter mit der zweiten Syntaxvariante lösen. "Oktalzahl" (s.o.) ist eine dreiziffrige oktale Zahl, bei der die erste Ziffer die Rechte des Besitzers beschreibt, die zweite die der Gruppe und die dritte wieder die der anderen Benutzer.

Dabei haben die Ziffern folgende Bedeutung:

- 0 - kein Recht
- 1 - x-Recht
- 2 - w-Recht
- 4 - r-Recht

Rechtekombinationen werden durch Addition der Grundwerte beschrieben. Die Rechtekombination rx entspricht zum Beispiel der Ziffer 5, umgekehrt repräsentiert die Ziffer 7 alle Rechte, also rwx.

Ein Vergleich: **chmod 751 testdatei** ersetzt vollständig den wesentlich umständlicheren Befehl **chmod u=rwx,g=rx,o=x testdatei**

Beachte, dass nur die erste Syntax das Entziehen einzelner Rechte oder deren zusätzliche Vergabe gestattet. Durch **chmod 751 testdatei** werden alle bisher vorhandenen Rechte ersetzt. Im Einzelfall liegt es natürlich im eigenen Ermessen, welche Befehlsform man anwenden will.

Der Befehl mount

mount hängt Dateisysteme ein und wieder aus. Steckt man Beispielsweise einen USB-Stick an den Rechner an, so muss man unter Umständen diesen Stick einhängen, bevor man Zugriff hat. Dazu muss man **root** werden! Die angeschlossenen Geräte werden dabei je nach Schnittstellenart unterschiedlich bezeichnet, (die entsprechenden Bezeichnungen findet man im Verzeichnis /dev wieder):

Typ	Festplatte	Partition
SATA-Festplatte, USB-Stick, externe Festplatte	1. Festplatte: sda 2. Festplatte: sdb 3. Festplatte: sdc ...	1. Partition: sda1 2. Partition: sda2 3. Partition: sda3 ...
IDE-Festplatte, CD/DVD-ROM, CD/DVD-Brenner (Bei CD/DVD-Laufwerken ohne Ziffer)	1. Gerät: hda 2. Gerät: hdb 3. Gerät: hdc ...	1. Partition: hda1 2. Partition: hda2 3. Partition: hda3 ...
Floppy (Diskettenlaufwerk)	fd0	im Allgemeinen ist maximal eins vorhanden, wenn überhaupt noch

Die Syntax

Der mount-Befehl benötigt das Gerät und den Zielort, "herkömmliche Laufwerke" wie unter Windows (Laufwerk C, D, etc.) gibt es bei Linux nicht. Es wird alles in Verzeichnisse gegliedert (bei Windows übrigens auch, nur bekommt man als Anwender nichts davon mit)

- **mount /dev/sdc2 /mnt** sagt zum Beispiel, dass man die 2. Partition der 3. Festplatte in das Verzeichnis /mnt einhängt. Der Inhalt des Laufwerks kann somit unter /mnt eingesehen werden
- **umount /mnt** hängt das dort eingehängte Gerät wieder aus

Am Beispiel USB-Stick

1. USB-Stick anstecken

2. Nach etwa 10 sec. **dmesg | tail** ausführen und schauen, welche Gerätebezeichnung der USB-Stick vom System bekommen hat, als Beispiel:

```
sd 7:0:0:0: [sdh] 490496 512-byte hardware sectors (251 MB)
sd 7:0:0:0: [sdh] Write Protect is off
sd 7:0:0:0: [sdh] Mode Sense: 37 00 00 08
sd 7:0:0:0: [sdh] Assuming drive cache: write through
sd 7:0:0:0: [sdh] 490496 512-byte hardware sectors (251 MB)
sd 7:0:0:0: [sdh] Write Protect is off
sd 7:0:0:0: [sdh] Mode Sense: 37 00 00 08
sd 7:0:0:0: [sdh] Assuming drive cache: write through
sdh: sdh1
sd 7:0:0:0: [sdh] Attached SCSI removable disk
```

Der eingesteckte USB-Stick hat hier die Bezeichnung **sdh1** bekommen

3. **mount /dev/sdh1 /mnt** hängt den Stick unter /mnt ein

Nun kann man darauf arbeiten (evtl. noch **chown -R klaus /mnt** ausführen, um den Eigentümer festzulegen und damit Schreibzugriff zu bekommen)

4. Nach Abschluss der Arbeiten **umount /mnt** ausführen, vorher jedoch bitte darauf achten, dass das Verzeichnis des USB-Sticks nirgendwo mehr geöffnet ist, da sich der Stick sonst nicht aushängen lässt