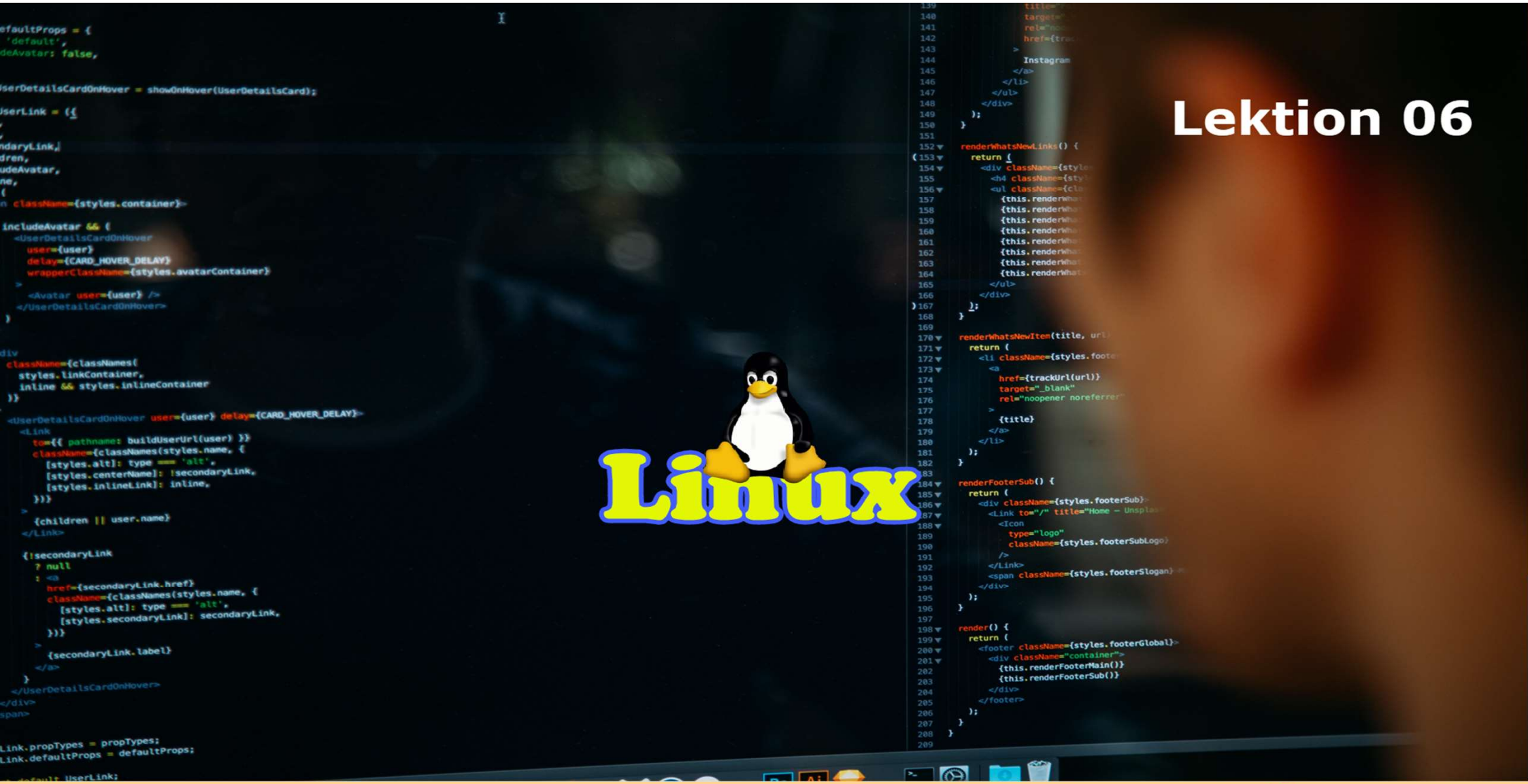


Rechteverwaltung in Linux

- In Linux gibt es drei Kategorien für die man Rechte setzen kann.
  - Den Eigentümer (user)
  - Die Gruppe (group)
  - Alle Anderen (others)
- Die Anfangsbuchstaben der englischen Bezeichnungen sind wichtig!
- Für die drei Kategorien gibt wiederum genau drei Rechte
  - r – lesen (read)
  - w – schreiben (write)
  - x – ausführen (execute)

Kategorien und Rechte

- Mit dem Befehl ls (list) kann man Dateien anzeigen lassen.
- Die Option -l steht für long, also einer ausführlichen Ausgabe.
- Gefolgt von einer Pfad-Angabe und einem Dateinamen kann man sich alle Rechte und Kategorien anzeigen lassen.
- Im Ordner /bin befindet sich u. a. die Datei „nano“ und deren Eigenschaften lassen wir uns anzeigen mit:
- # ls -l /bin/nano

```
helmut@deb-s1:~$ ls -l /bin/nano
-rwxr-xr-x 1 root root 287480 18. Jan 2023  /bin/nano
```

Objekt-Kennzeichen

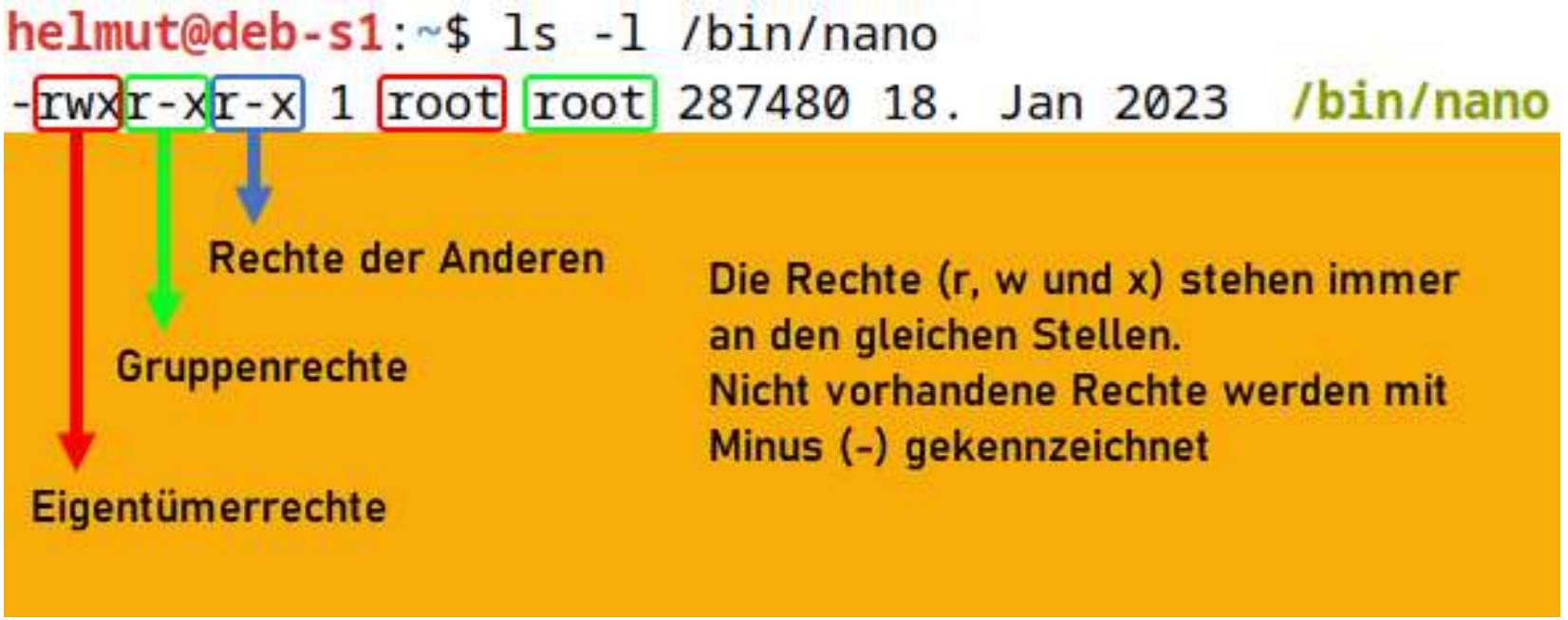
```
helmut@deb-s1:~$ ls -l /bin/nano
-rwxr-xr-x 1 root root 287480 18. Jan 2023  /bin/nano
```

Kennzeichen	Bedeutung
-	Normale Datei (Text, Tabelle, ausführbar, Archiv, …)
d	Ordner, Verzeichnis (directory)
l	Softlink (Verknüpfung)
b	Block-Device (Festplatte, USB-Stick,…)
c	Character-Device (serielle Schnittstelle, serieller Zugriff)

Eigentümer und Gruppe

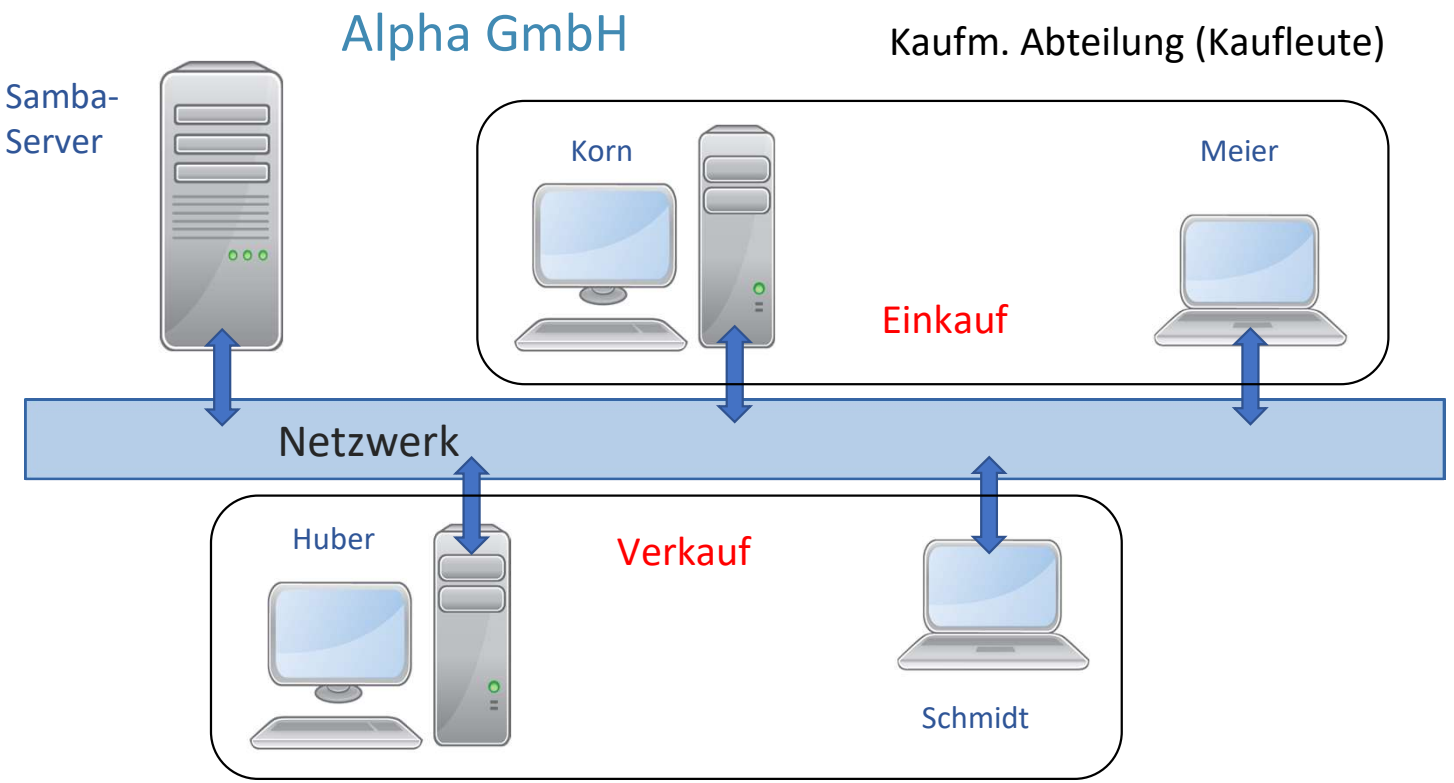
```
helmut@deb-s1:~$ ls -l /bin/nano
-rwxr-xr-x 1 root root 287480 18. Jan 2023  /bin/nano
```

Eigentümer und Gruppe

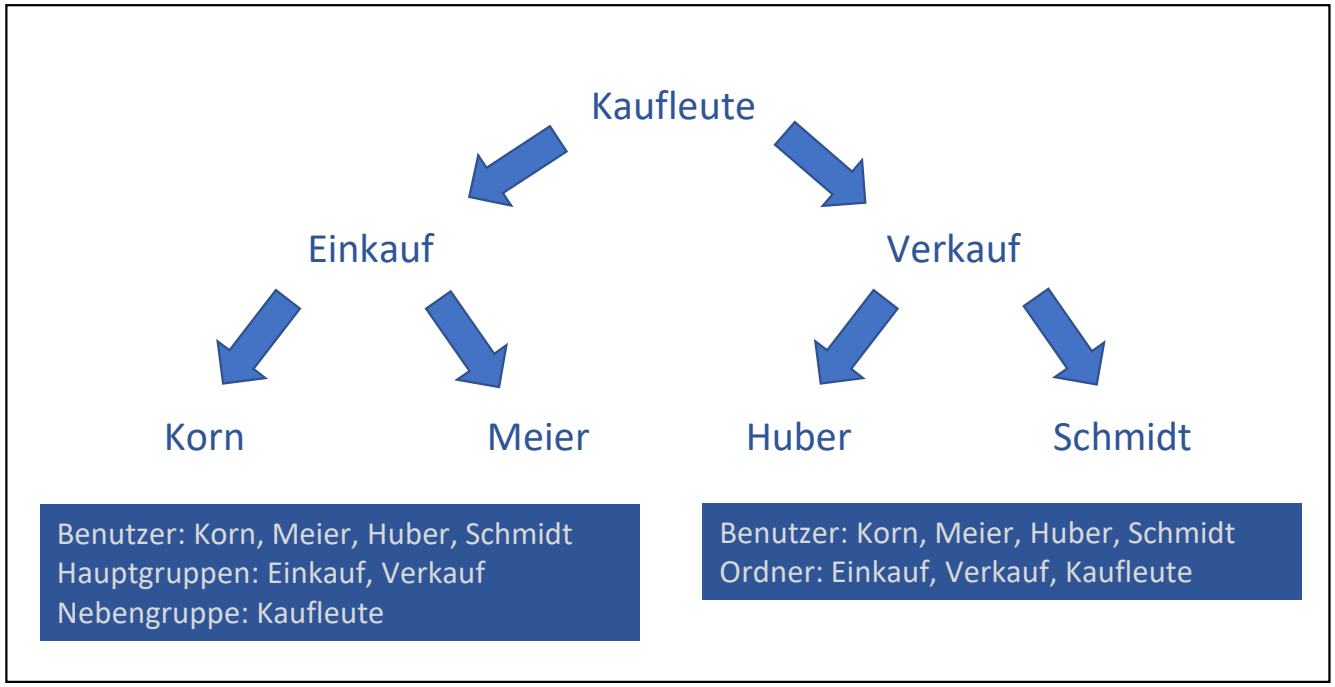


Der Eigentümer **root** besitzt die Rechte lesen (r), schreiben (w) und ausführen (x).
Die Gruppe **root** besitzt die Rechte lesen (r) und ausführen (x), aber keine Schreibrechte (w).
Die **Anderen (der Rest der Welt)** dürfen die Datei ebenfalls lesen (r) und ausführen (x), haben aber keine Schreibrechte (w).

Projekt: kaufmännische Abteilung

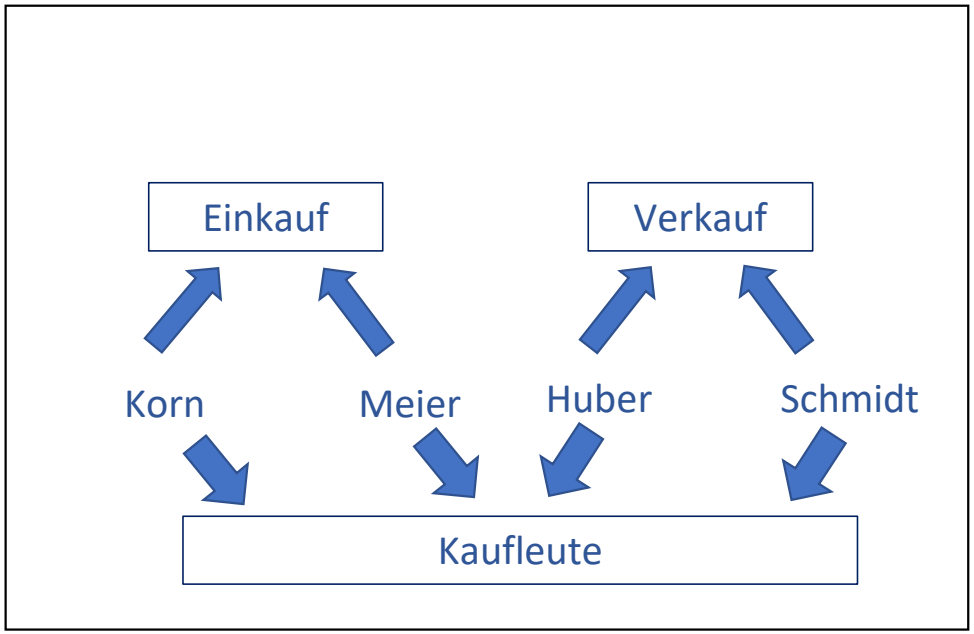


Hierachie der Kaufleute



Zuordnungen

- Der Nebengruppe „Kaufleute“ gehören alle Benutzer an
- Der Hauptgruppe „Einkauf“ gehören nur Korn und Meier an
- Der Hauptgruppe „Verkauf“ gehören nur Huber und Schmidt an
- Jeder Benutzer hat seinen eigenen (privaten) Homeordner

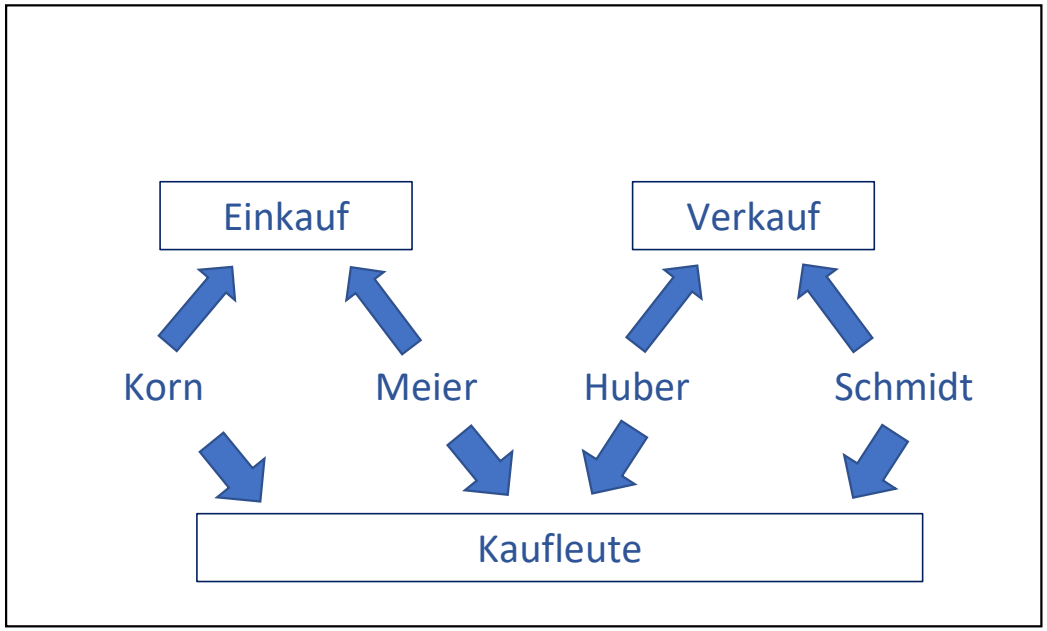


Konventionen

- Alle administrativen Aktionen benötigen root-Rechte
  - Anzeige: #
  - Befehl: su –
- Die Arbeiten werden über ssh auf dem Server deb-s1 ausgeführt
- Alle anzulegenden Elemente werden klein geschrieben
  - Benutzer (Eigentümer)
  - Gruppen
  - Dateinamen
  - Ordnernamen

Gruppen anlegen

- Der Befehl dazu heißt
  # groupadd <Gruppenname>
  - # groupadd kaufleute
  - # groupadd einkauf
  - # groupadd verkauf




```
root@deb-s1:~# groupadd kaufleute
root@deb-s1:~# groupadd einkauf
root@deb-s1:~# groupadd verkauf
```

Ergebnis mit tail anzeigen

```
root@deb-s1:~# groupadd kaufleute
root@deb-s1:~# groupadd einkauf
root@deb-s1:~# groupadd verkauf
root@deb-s1:~# tail -n3 /etc/group
kaufleute:x:1001:
einkauf:x:1002:
verkauf:x:1003:
root@deb-s1:~#
```

User (Eigentümer) anlegen

- Mit # useradd werden die Eigentümer angelegt.
- Mehrere Optionen für diesen Befehl sind möglich:
  - -g Festlegung der Hauptgruppe des Eigentümers
  - -G Festlegung der Nebengruppe(n) für den Eigentümer
  - -m für den Eigentümer wird sein Home-Verzeichnis angelegt
- Aus Übersichtsgründen hier nicht benutzt sind die Optionen:
  - -c Kommentareinträge z.B. Schlüssel-, Raum- oder Telefonnummern
  - -s zu nutzende Shell z.B. /bin/bash

```
root@deb-s1:~# useradd -g einkauf -G kaufleute -m korn
root@deb-s1:~# useradd -g einkauf -G kaufleute -m meier
root@deb-s1:~# useradd -g verkauf -G kaufleute -m huber
root@deb-s1:~# useradd -g verkauf -G kaufleute -m schmidt
```

Kontrolle

- Mit # tail lassen sich die unteren Zeilen eines Dateiinhalts anzeigen
  - Die Datei /etc/group enthält alle Gruppen
  - -n3 zeigt die letzten 3 Zeilen der Datei an

```
root@deb-s1:~# tail -n3 /etc/group
kaufleute:x:1001:korn,meier,huber,schmidt
einkauf:x:1002:
verkauf:x:1003:
```

  - Die Datei /etc/passwd enthält alle Eigentümer, zeigt deren home-Ordner und deren Gruppenzuordnungen an

```
root@deb-s1:~# tail -n4 /etc/passwd
korn:x:1001:1002::/home/korn:/bin/sh
meier:x:1002:1002::/home/meier:/bin/sh
huber:x:1003:1003::/home/huber:/bin/sh
schmidt:x:1004:1003::/home/schmidt:/bin/sh
```

Benutzer verändern

- Mit usermod kann man Veränderungen an den getroffenen Usereinstellungen vornehmen
- Beispiele:
  - `# usermod –d <neues Homeverzeichnis> <Benutzer>`
  - Die Option –m ist nicht erlaubt. Daher muss das neue Homeverzeichnis händisch angelegt und mit den entsprechenden Rechten versehen werden
  - `# usermod –l <neuer Benutzername> <alter Benutzername>`
  - Das home-Verzeichnis wird nicht automatisch mit umbenannt. Es muss händisch umbenannt werden und kann dann mit usermod –d … (siehe oben) neu zugeordnet werden.

Benutzer sperren und freigeben

- Mit usermod kann man einen Benutzer sperren z.B. weil er vorübergehend nicht im Unternehmen ist
  - `# usermod –L <Benutzer>`
  - In `/etc/shadow` wird vor dem verschlüsselten Passwort ein Ausrufezeichen gesetzt und verwehrt damit dem Benutzer die Anmeldung
  - `# usermod –U <Benutzer>`
  - Das Ausrufezeichen vor dem verschlüsselten Passwort in der Datei `/etc/shadow` wird entfernt und der Benutzer kann sich wieder anmelden
- Alle Optionen von useradd (außer -m) lassen sich mit usermod verändern

Benutzer löschen

- Mit `# userdel <Benutzer>` löscht man nicht mehr benötigte Benutzer
- Das home-Verzeichnis des Benutzers wird damit nicht gelöscht
- Soll dieses zusammen mit den Mail-Spool-Verzeichnis gelöscht werden setzt man `# userdel -r <Benutzer>` (-r steht wieder für remove)
- In Produktiv-Systemen sollte man nicht zu schnell komplett löschen
- Das Sperren ist erst einmal die bessere Variante, auch wenn der Benutzer das Unternehmen verlassen hat

Rechte allgemein

- Die Rechte an Dateien und Verzeichnissen werden mit dem Kommando chmod (change mode)zugewiesen oder weggenommen
- Die Nutzung ist nur als Benutzer root möglich
- Das vorangestellte Rautezeichen # vor chmod soll das verdeutlichen
  - `# chmod u+x <Datei>` oder
  - `# chmod 755 <Datei>` (die Gruppe und die Anderen erhalten das Lese- und das Ausführrecht)
- Die beiden Beispiele zeigen, dass es 2 Varianten für chmod gibt
- Welche Variante man bevorzugt nutzt bleibt jedem selbst überlassen

Rechte für Dateien und Verzeichnisse

Recht	Datei	Verzeichnis
Lesen (read r)	Darf zum Lesen geöffnet werden	Datei- und Verzeichnisnamen dürfen gelesen werden, deren Attribute, wie Besitzer, Berechtigungen, usw. sind nicht lesbar
Schreiben (write w)	Darf zum Schreiben geöffnet werden, Ändern und löschen sind darin enthalten	Einträge im Verzeichnis dürfen bearbeitet werden, was das Erstellen, Löschen und Ändern einbezieht
Ausführen (execute x)	Darf ausgeführt werden, was allerdings nur bei Skripten, Programmen und Kommandos sinnvoll ist	Der Wechsel in das Verzeichnis und das Abrufen weiterer Attribute zu den enthaltenen Dateien und Unterordnern sind erlaubt

Variante 1

- Rechtezuweisung mit Buchstaben:
- Die Bedeutung der Buchstaben u (user), g (group) und o (others) muss bekannt sein, genau wie die Buchstaben für die Rechte, also r (read), w (write) und x (execute)
- Mit dem Pluszeichen + werden Rechte zugewiesen mit dem Minuszeichen - werden Rechte weggenommen
- Auch das Gleichheitszeichen (=) kann zum Einsatz kommen

Variante 1 Beispiele (relativ)

- Relativ bedeutet, dass die Rechte zusätzlich zu den vorhandenen Rechten gesetzt oder weggenommen werden, was heißt, dass die nicht genannten Rechte unverändert bleiben
  - `# chmod u+x <Datei>` gibt dem user (u) zusätzlich Ausführrechte (+x) an der <Datei>
  - `# chmod o-rwx <Datei>` nimmt others (o) alle Rechte an der <Datei> weg [lesen (-r), schreiben(-w) und ausführen(-x)]
  - `# chmod g+rw,o-r <Datei>` gibt der group zusätzlich die Rechte lesen(+r) und schreiben(+w) und nimmt others das Leserecht (-r) weg
- Wenn man chmod –v (verbose) verwendet, bekommt man die vorherigen und die neuen Rechte in einer Zeile angezeigt
- `# chmod –v u-w <Datei>`

Variante 1 Beispiele (absolut)

- Absolut bedeutet, dass die Rechte nach dem Gleichheitszeichen gesetzt werden und die nicht genannten Rechte entzogen werden
  - `# chmod -v u=rw <Datei>` setzt die Userrechte auf lesen(r) und schreiben(w). Hatte der User vorher das Recht ausführen(x), so ist es jetzt weggenommen
  - `# chmod -v o=r <Datei>` gibt others (o) das Recht lesen(r), nimmt ihnen aber die vielleicht vorher vorhandenen Schreib- und Ausführrechte
- Mit `ls –l <Datei>` kann und sollte man die Auswirkungen kontrollieren

Variante 2

- Rechteverwaltung mit Ziffern (Oktalmethode):
  - Jedem Recht ist eine logische Bit-Wertigkeit in oktaler Form zugeordnet
  - Da es drei Rechte gibt kommen auch drei Bit zum Einsatz (Bit 0 bis Bit 2)
  - Bit 0 ist dem Ausführrecht (x) zugeordnet und hat die Wertigkeit 1 (okt)
  - Bit 1 ist dem Schreibrecht (w) zugeordnet und hat die Wertigkeit 2 (okt)
  - Bit 2 ist dem Leserecht (r) zugeordnet und hat die Wertigkeit 4 (okt)
- Das klingt erst einmal viel komplizierter als die Buchstabenmethode, ist es aber nicht, denn die 3 Werte hat man sich sehr schnell eingeprägt

Variante 2: Übersicht der Rechte

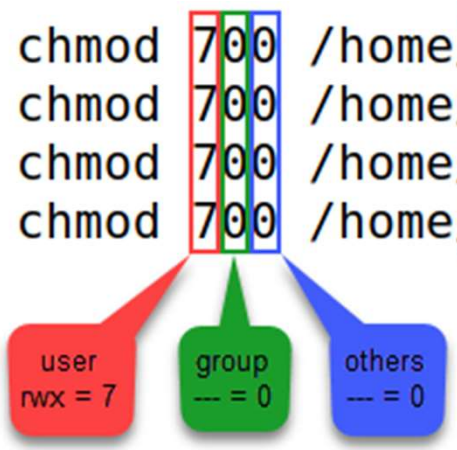
3-Bitmuster für r w x	Wert (oktal)	Kombination der Rechte	
0 0 0	0	---	keine Rechte
0 0 1	1	--x	nur ausführen (nicht Praxistauglich)
0 1 0	2	-w-	nur schreiben (nicht Praxistauglich)
0 1 1	3	-wx	schreiben und ausführen (nicht Praxistauglich)
1 0 0	4	r--	nur lesen
1 0 1	5	r-x	lesen und ausführen
1 1 0	6	rw-	lesen und schreiben
1 1 1	7	rwx	lesen, schreiben und ausführen

- Relevant sind hier nur die Oktalwerte für die Benutzung von chmod und deren zugehörige Rechtekombination

Rechte an den home-Ordnern

- Damit jeder Benutzer alleiniger Eigentümer seines home-Ordners ist, wird dies mit # chmod (Change Modus) festgelegt.
  - 7 – Benutzer (user) hat alle Rechte (r = 4, w = 2, x = 1 → 4 + 2 + 1 = 7)
  - 0 – Gruppe (group) und „alle anderen“ (others) haben keine Rechte (- - - → 0 + 0 + 0 = 0)

```
root@deb-s1:~# chmod 700 /home/korn
root@deb-s1:~# chmod 700 /home/meier
root@deb-s1:~# chmod 700 /home/huber
root@deb-s1:~# chmod 700 /home/schmidt
```



Kontrolle der Rechtevergabe

- Mit `# ls –l /home` kann man sich das Ergebnis der Einstellungen ansehen

```
root@deb-s1:~# ls -l /home
insgesamt 20
drwxr-xr-x 4 helmut  helmut  4096 25. Apr 19:15 helmut
drwx------ 2 huber   verkauf 4096 25. Apr 20:21 huber
drwx------ 2 korn    einkauf 4096 25. Apr 20:20 korn
drwx------ 2 meier   einkauf 4096 25. Apr 20:21 meier
drwx------ 2 schmidt verkauf 4096 25. Apr 20:21 schmidt
```

Passwörter

- Die eingerichteten Benutzer haben zwar Ihre home-Verzeichnisse, können sich aber im Linux-System nicht anmelden
- Es fehlen die Passwörter, die nur root anlegen darf
- Danach kann jeder Benutzer sein zugewiesenes Passwort ändern
- Zum Einrichten der Passwörter verwendet root das Kommando `passwd`
- `# passwd <Benutzer>`
- In unserem Workshop entspricht das Passwort dem Benutzernamen

Passwörter vergeben

- Gibt man nur passwd ohne Benutzer ein, dann kann der angemeldete Benutzer sein Passwort ändern (ohne root-Rechte)
  - Mit root-Rechten können alle Benutzer ein (neues) Passwort erhalten
  - Während der Eingabe sind keine Zeichen sichtbar
  - Legen Sie die Passwörter der Benutzer nach unten stehendem Schema an (korn, huber, meier und schmidt)

```
root@deb-s1:~# passwd korn
Geben Sie ein neues Passwort ein:
Geben Sie das neue Passwort erneut ein:
passwd: Passwort erfolgreich geändert
```

  - Die Passwörter kann man in verschlüsselter Form mit `# tail –n4 /etc/shadow` ansehen

Verschlüsselte Passwörter kontrollieren

```
root@deb-s1:~# tail -n4 /etc/shadow
korn:$y$j9T$Aq8BAUwHOTyDnGDSkupGE0$4ldgGE1iAakdL31iPO.4qPTz9SIB5L96oz3TaAvnDP7:19765:0:99999:7:::
meier:$y$j9T$eeQXT4yhBzhcHdY0ajesY1$CPlAZUTW5pcbHbwGekvmWLdk5N0ik2D7QvfzME5M0h/:19765:0:99999:7:::
huber:$y$j9T$t4/I5AMy6WgE4ZXOUv3320$i65Nat1SxCkHdiAnLL1BMtdo2lbLDeMkx0n8d6SqcwC:19765:0:99999:7:::
schmidt:$y$j9T$Q60qpo.GwRkA5chfteFog0$q5gmj4NasSVZMkEARv8Wc6w/w5vRgBEiroUEhY/NS46:19765:0:99999:7:::
root@deb-s1:~#
```

Gruppenordner anlegen

- Bisher gibt es die Gruppenzuordnungen in hierarchischer Form
- Um Dateien für eine bestimmte Gruppe zu nutzen, benötigen wir noch Verzeichnisse für derartige Dateien.
  - Befehl # mkdir (make a directory)
  - Option –p erstellt gleich eine Verzeichnisstruktur, hier /verteiler/einkauf
  - -p wird nur einmal benötigt, da die Struktur nach dem ersten Einsatz angelegt ist
  - Das Verzeichnis „verteiler“ verhindert, dass die 3 Abteilungen direkt im home-Verzeichnis stehen, wodurch die Übersicht verloren ginge

```
root@deb-s1:~# mkdir -p /home/verteiler/einkauf
root@deb-s1:~# mkdir /home/verteiler/verkauf
root@deb-s1:~# mkdir /home/verteiler/kaufleute
```

Eigentümer und Gruppe

- Eigentümer und Gruppe können gleichwertig Rechte erhalten
- Den Rest der Welt (others) sperrt man aus, um ganz sicher zu gehen, dass kein Unbefugter in das System eindringen kann.
- Daher spielen others bei der Eigentümer- und Gruppenzuweisung keine Rolle
  - Befehl: `# chown (change owner)`
- chown kann den gewünschten Benutzer und die gewünschte Gruppe gleichzeitig ändern (in diesem Projekt wird nur die Gruppe geändert)
  - Allgemeiner Syntax: `# chown Benutzer:Gruppe /Pfad/zum/Ordner`
  - Beispiel: `# chown root:einkauf /home/verteiler/einkauf`

Eigentümer und Gruppe

- Im Projekt werden die erstellten Gruppen als Besitzergruppen festgelegt
- Achtung: Als Benutzer (Besitzer) wird root eingetragen und nur die Gruppe wird entsprechend zugeordnet
- Kein anderer Benutzer darf hier weitergehende Privilegien erhalten

```
root@deb-s1:~# ls -l /home/verteiler/
insgesamt 12
drwxr-xr-x 2 root root 4096 25. Apr 21:08 einkauf
drwxr-xr-x 2 root root 4096 25. Apr 21:08 kaufleute
drwxr-xr-x 2 root root 4096 25. Apr 21:08 verkauf
root@deb-s1:~# chown root:einkauf /home/verteiler/einkauf
root@deb-s1:~# chown root:verkauf /home/verteiler/verkauf
root@deb-s1:~# chown root:kaufleute /home/verteiler/kaufleute
```

Ordnerrechte

- Wiederum wird der Befehl # chmod eingesetzt
  - root hat volle Rechte – 7 an der ersten Stelle
  - Die jeweilige Gruppe hat volle Rechte – 7 an der zweiten Stelle
  - Alle anderen haben keine Rechte – 0 an der dritten Stelle
  - Die 2 vor den Rechten ist das SGID-Bit = Set-Group-ID (roter Rahmen).
  - Ist dieses Bit gesetzt, haben alle Mitglieder der Gruppe Kaufleute gleichzeitig den Zugriff auf diesen Ordner

```
root@deb-s1:~# chmod 770 /home/verteiler/einkauf
root@deb-s1:~# chmod 770 /home/verteiler/verkauf
root@deb-s1:~# chmod 2770 /home/verteiler/kaufleute
root@deb-s1:~# ls -l /home/verteiler/
insgesamt 12
drwxrwx--- 2 root einkauf   4096 25. Apr 21:08 einkauf
drwxrws--- 2 root kaufleute 4096 25. Apr 21:08 kaufleute
drwxrwx--- 2 root verkauf   4096 25. Apr 21:08 verkauf
```

Das SGID-Bit

- Das SGID-Bit = Set-Group-ID-Bit ist eine elegante Lösung konfliktfrei allen Gruppenmitgliedern die nötigen Rechte einzuräumen.
  - Alle neu erstellten Dateien und Unterverzeichnisse werden mit der Gruppe des Verzeichnisses (hier die Gruppe Kaufleute) und nicht mit der Hauptgruppe des Benutzers erstellt, der die Datei oder das Unterverzeichnis erstellt
  - Sonst würde es passieren, dass die jeweils andere Gruppe (einkauf bzw. verkauf) keinen Zugriff hätte

Das SUID-Bit

- Nicht nur für Gruppen stehen Sonderrechte zur Verfügung. Auch Benutzer benötigen hin und wieder besondere Rechte
- Mit dem Kommando passwd kann jeder Benutzer sein Passwort ändern. Die Datei passwd befindet sich im Ordner /usr/bin, also /usr/bin/passwd
- passwd muss das verschlüsselte Passwort in die Datei /etc/shadow schreiben

```
helmut@debian:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1183 Apr 25 13:02 /etc/shadow
```

- Wie man erkennt hat nur root Schreibzugriff auf diese Datei. Wieso kann dann jeder Benutzer durch Eingabe von passwd sein Passwort ändern?

```
helmut@debian:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 63736 Jul 27  2018 /usr/bin/passwd
```

- Das SUID-Bit (Set-User-ID-Bit) steht an der Stelle von x (execute). Ist dieses Bit für ein Programm gesetzt, erhält ein normaler Benutzer für sich die root-Rechte zum Ausführen der Datei

Das Sticky-Bit

- Das Sticky-Bit ist gedacht für Ausführende vom „Rest der Welt“ (others). Es wird durch ein „t“ anstelle des „x“ im dritten Rechteblock dargestellt
- Es trifft nur auf Verzeichnisse zu. Typisch hierfür ist /tmp

```
helmut@debian:~$ ls -ld /tmp
drwxrwxrwt 14 root root 4096 Apr 25 09:56 /tmp
```

- Wäre das Bit nicht gesetzt, würde jeder Benutzer nicht nur Schreibrechte in diesem Verzeichnis haben, sondern auch beliebige – ihm nicht gehörende – Dateien löschen bzw. umbenennen können. So kann er nur seine eigenen Inhalte löschen oder umbenennen
- Damit wird in dem öffentlichen Ordner (/tmp) die Sicherheit der Daten gewährleistet.

Wertigkeit der drei Sonderbits

- Sticky-Bit 1
- SGID-Bit 2
- SUID-Bit 4
- Würde man die schon gesetzten Bits, also die gesetzten Rechte manuell setzen wollen, würde man für `/usr/bin/passwd` eingeben (rein theoretisch, denn SUID ist schon gesetzt):

```
root@debian:~# chmod 4755 /usr/bin/passwd
```

- Das gleiche gilt für das Sticky-Bit des Verzeichnisses /tmp

```
root@debian:~# chmod 1777 /tmp
```

- Man könnte die Rechte auch kombinieren, was aber nicht üblich ist