

# Linux Grundlagen

---

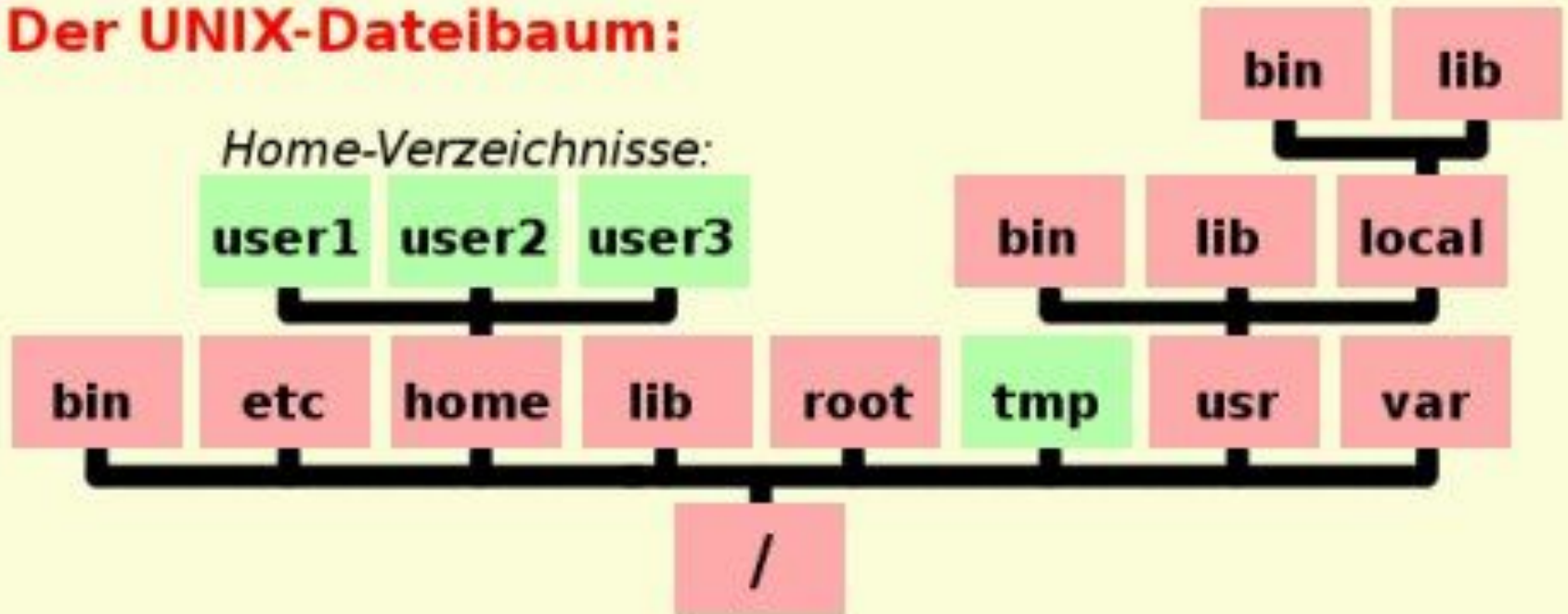
absolut basics

# Linux System

- Linux ist ein Multiuser-System
- Die User (engl., Benutzer) werden zwar mit ihren Namen verwaltet, intern arbeitet ein Unix-System aber mit Usernummern
- Jeder Benutzer ist auch Mitglied mindestens einer Gruppe. Es kann beliebig viele Gruppen in einem System geben und auch sie haben intern Nummern (GroupID oder GID)

# Verzeichnisstruktur

## Der UNIX-Dateibaum:



# BASH

- Eine Shell (engl. Hülle, Schale, Außenhaut) bezeichnet die traditionelle Benutzerschnittstelle unter Unix-Betriebssystemen.
- Es gibt viele nützliche Programme, für die es keine graphische Oberfläche gibt und auch Programme mit graphischer Oberfläche kennen oft Optionen, die nur von der Kommandozeile aus zugänglich sind.

# BASH

- Die Bash speichert alle eingegebenen Befehle. Mit den **Pfeiltasten** ↑ und ↓ können sie zurückgerufen und (eventuell modifiziert) wieder abgesetzt werden. Sollen alle gespeicherten Befehle ausgegeben werden, genügt die Eingabe `history`, gefolgt von ENTER.
- Tippt man den Anfang eines Befehls und drückt dann die **Tabulatortaste**, versucht die Bash den Befehl zu vervollständigen.

# Betrachten, Navigieren, Suchen

## **pwd**

Wir sehen uns das "aktuelle" Verzeichnis an, das heißt, das Verzeichnis, in dem wir uns gerade befinden:

## **ls / ls -l**

Wir bekommen alle Dateien und Verzeichnisse im Home-Verzeichnis aufgelistet.

## **cd verzeichnisname**

Wir wechseln ins Verzeichnis "verzeichnisname":

Wir können auch nach Dateien suchen:

**find verzeichnisname -name dateiname → find /etc -name fonts.conf**

# Textdateien lesen/schreiben

`less /etc/fonts/fonts.conf`

`cat /etc/fonts/fonts.conf`

`head /etc/fonts/fonts.conf`

`tail /etc/fonts/fonts.conf`

## **Editoren:**

`vi/vim, nano`

# Absolute und relative Pfade

**Absolute Pfade** werden von der Dateiwurzel aus angegeben. Sie beginnen mit einem /. Ein Beispiel:

```
ls /usr/bin  
cd /
```

**Relative Pfade** werden vom aktuellen Verzeichnis aus angegeben. Sie beginnen mit einem Datei- oder Verzeichnis-Namen.



# Dateien anlegen/kopieren/verschieben/umbenennen

Verzeichnis anlegen

**mkdir verzeichnisname**

leere Dateien erzeugen

**touch datei.txt**

Dateien kopieren

**cp quelle ziel(verzeichnis)**

Zum Verschieben und Umbenennen benutzen wir mv (move):

**mv quelle ziel(verzeichnis)**

# Dateien löschen

**rm (-optionen) dateiname**

**rm hat keinen Papierkorb!** Mit Administratorrechten ausgeführt, kann im schlimmsten Fall das gesamte Betriebssystem gelöscht werden.

Zum Löschen von leeren Verzeichnissen gibt es auch rmdir (remove directory):

**rmdir verzeichnisname**

# Dateirechte

- Unter Linux hat jede Datei (auch Verzeichnisse) einen Besitzer und bestimmte Zugriffsrechte.
- Wir unterscheiden zwischen Lese-, Schreib- und Ausführungsrechten.
- Diese können bei jeder Datei für den Besitzer selbst, für Angehörige einer definierten Gruppe und für alle anderen festgelegt werden.

r	für Lesen (read)
w	für Schreiben (write)
x	für Ausführen (execute) - bzw. bei Verzeichnissen, den Inhalt auflisten
-	wenn das entsprechende Recht nicht vorhanden ist

# Dateirechte ändern

chmod (change modus)

**chmod modus dateiname**

Es gibt 2 Methoden "modus" anzugeben:

u	steht für den Besitzer (user)
g	steht für die Gruppe (group)
o	steht für Andere (others)
a	steht für alle (all)

chmod a+x datei.txt

1	Ausführen
2	Schreiben
4	Lesen

chmod 600 datei.txt

# Besitzer ändern

Eine Datei oder ein Verzeichnis wird auf den neuen Besitzer "besitzer" übertragen.

**chown "besitzer" dateiname**

Mit chown kann auch gleich die Gruppe geändert werden

**chown besitzer:gruppe dateiname (nicht ausprobieren!)**

Soll nur die Gruppe geändert werden gibt es chgrp:

**chgrp gruppe dateiname**

# System konfigurieren

Über einen Eintrag in der Datei `.bashrc` im Home-Verzeichnis können Aliase auch dauerhaft eingerichtet werden.

```
nano ~/.bashrc
```

Variablen definieren

**variable=wert**

Der Aufruf der Variablen erfolgt dann immer mit einem vorangestellten \$:

**echo \$variable**

# System konfigurieren

Man kann der bash auch neue Kommandos beibringen, indem man einen Alias erzeugt

**alias aliasname='programm (-optionen ...)'**

Ein so gesetzter Alias existiert nur bis zum Beenden der Shell, in der er gesetzt wurde.

# Umgebungsvariablen

echo \$HOME liefert den Pfad des Home-Verzeichnisses

echo \$HOSTNAME den Rechnernamen

echo \$LOGNAME den Benutzernamen

echo \$UID die Benutzer ID

echo \$RANDOM



# Umleitungen

Ausgabe eines Programms in eine Datei schreiben

**programm (-optionen parameter) > dateiname**

Wir können der Datei auch noch etwas hinzufügen. > überschreibt eine bestehende Datei, >> fügt ihr etwas hinzu:

**programm (-optionen parameter) >> dateiname**

Pipes (engl., Rohrleitungen) leiten den Datenstrom eines Programms an ein anderes weiter.

**programm1 (...) | programm2 (...)**

# Systemadministration

Der Administrator heißt auf UNIX-Systemen eigentlich "root" (engl. Wurzel), wird aber von Programmen oft auch als Superuser bezeichnet.

In der Regel meldet man sich immer als einfacher Benutzer an und wird nur im Bedarfsfall zum User "root".

Das Programm su (switch user) erlaubt es, die Identität zu wechseln:

**su (-) (benutzername)**

**sudo programm (-optionen ...)**

# Programmdokumentationen

Zu fast jedem Programm gibt es eine Dokumentation

Diese kann man mit man (manual) anzeigen lassen

**man programmname**

**man man**

Manpages nach Schlüsselwörtern durchsuchen

**man -k**

Manpages zu einem Programm finden

**Man -f Programmname**

# Aufgabe

- Legt im temporären System Verzeichnis einen Ordner “Aufgabe” an
- Legt einen neuen Benutzer und eine Gruppe an (adduser/addgroup)
- Sorgt dafür, dass nur der neue User Eigentümer des Ordners Aufgabe wird und den Ordner Lesen/Schreiben darf.
- Außerdem soll die neue Gruppe Zugriff auf den Ordner bekommen und nur Leserechte haben
- Wechselt zu dem neuen Benutzer und legt im Ordner Aufgabe eine Textdatei mit dem Inhalt Text1 an