

# Arbeiten mit der Shell Teil 1

Linux-Kurs der Unix-AG

Zinching Dang

2. Mai 2017



TU Kaiserslautern



# Übersicht

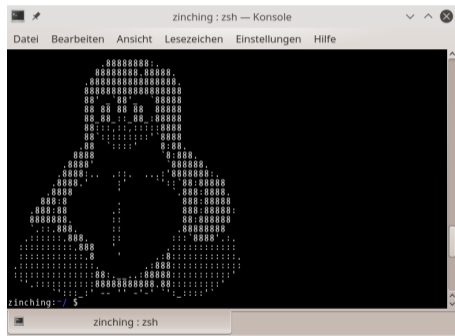
Wiederholung & Vertiefung

Die Unix-Philosophie

Shellbefehle

Befehle & Optionen

Zusammenfassung & Ausblick



Tux im Terminal

# Wiederholung: Pfade

## Pfadangaben sind

- ▶ absolut: vom Wurzelverzeichnis `/` ausgehend
- ▶ relativ: vom aktuellen Verzeichnis absteigend

## Besondere Pfade

- ▶ `.` – aktuelles Verzeichnis
- ▶ `..` – darüberliegendes Verzeichnis, „Elternverzeichnis“
- ▶ `~` oder `$HOME` – eigene Home-Verzeichnis

# Wiederholung: Befehlseingabe

## Befehle

- ▶ werden mit <ENTER> ausgeführt
- ▶ haben (manchmal optionale) Argumente
- ▶ können durch Optionen Verhaltenweise ändern

## Optionen

- ▶ können **oft** eine beliebige Reihenfolge haben
- ▶ können kombiniert werden
  - ▶ `ls -l -h` wird zu `ls -lh`

# Wiederholung & Vertiefung: Shellbefehle

## Bereits kenengelernte Befehle

- ▶ `ls` – „**l**ist“: Verzeichnisinhalte auflisten
- ▶ `cd` – „**c**hange **d**irectory“: Verzeichnis wechseln
- ▶ `mkdir` – „**m**ake **d**irectory“: Verzeichnis erstellen
- ▶ `rmdir` – „**r**emove **d**irectory“: Verzeichnis löschen
- ▶ `nano` – Konsoleneditor

# Verzeichnisinhalte auflisten – ls

## Wichtige Optionen

- ▶ `-a` – „**a**ll“: versteckte Dateien und Verzeichnisse anzeigen
- ▶ `-h` – „**h**uman readable“: menschenlesbares Ausgabeformat
- ▶ `-l` – „**l**ong listing“: weitere Informationen anzeigen
- ▶ `-R` – „**R**ecursive“: Verzeichnisse rekursiv auflisten<sup>1</sup>
- ▶ `-r` – „**r**everse“: Sortierreihenfolge umdrehen
- ▶ `-S` – „**S**ort by size“: der Größe nach sortieren
- ▶ `-t` – „**t**ime“: nach Änderungsdatum sortieren

---

<sup>1</sup>Vgl. `cp`, `rm`

# Verzeichnis wechseln & Verzeichnispfad – cd & pwd

## Besonderheiten von cd

- ▶ hat keine nennenswerte Optionen
- ▶ `cd -` – in das vorherige Verzeichnis wechseln
- ▶ `cd ~` oder `cd` – in das eigene Home-Verzeichnis wechseln
- ▶ `cd ..` – in das darüberliegende Verzeichnis (Elternverzeichnis) wechseln

## Aktuelles Verzeichnis feststellen

- ▶ `pwd` – „**p**rint **w**orking **d**irectory“
  - ▶ aktuellen Verzeichnispfad anzeigen

# Verzeichnisse erstellen & löschen – mkdir & rmdir

## Wichtige Option

- ▶ `mkdir -p` – fehlende Unterverzeichnisse erstellen
- ▶ `rmdir -p` – leere Unterverzeichnisse auch löschen

## Besonderheiten

- ▶ `rmdir` kann nur leere Verzeichnisse löschen



# Dateien bearbeiten mit nano

## Bedienung

- ▶ Dateiname als Argument
  - ▶ existierende Datei wird zum Bearbeiten geöffnet
  - ▶ nicht-existierende Datei wird neu erstellt
- ▶ speichern mit <STRG+O>
- ▶ beenden mit <STRG+X>

# Die Unix-Philosophie

## Wichtiges Paradigma: „Everything is a File“

- ▶ alles ist eine Datei
- ▶ keine Registry, Systemsteuerung, Laufwerksbuchstaben, etc.

## Viele kleine Programme

- ▶ lösen effizient ihre jeweiligen Aufgaben
- ▶ können verknüpft werden, um komplexe Aufgaben zu lösen

# Die Unix-Philosophie

## Das Rad nicht neu erfinden

- ▶ viele Programme haben gleiche Optionen
- ▶ gleichlautende Optionen bewirken **oft** dasselbe

## Auf das Wesentliche konzentrieren

- ▶ keine unnötigen Ausgaben auf der Kommandozeile
- ▶ viele Programme geben keine explizite Erfolgsmedung aus

# Übersicht: Shellbefehle

## Neue Befehle

- ▶ `cp` – „**copy**“: Dateien und Verzeichnisse kopieren
- ▶ `mv` – „**move**“: Dateien und Verzeichnisse verschieben
- ▶ `rm` – „**remove**“: Dateien löschen
- ▶ `man` – „**man**ual“: Handbuch zu Befehlen

# Dateien und Verzeichnisse kopieren – cp

## Wichtige Optionen

- ▶ `-b` – „**b**ackup“: vorhandene Dateien vorher umbenennen<sup>2</sup>
- ▶ `-i` – „**i**nteractive“: vor dem Überschreiben nachfragen<sup>2,3</sup>
- ▶ `-p` – „**p**reserve“: Zusatzinformationen beibehalten
  - ▶ Änderungsdatum, Besitzer, etc.
- ▶ `-r` oder `-R` – „**R**ecursive“: Dateien rekursiv kopieren
  - ▶ ermöglicht das Kopieren von Verzeichnissen

---

<sup>2</sup>Vgl. mv

<sup>3</sup>Vgl. rm

# Dateien und Verzeichnisse verschieben – mv

## Wichtige Optionen

- ▶ `-b` – „**b**ackup“: vorhandene Dateien vorher umbenennen
- ▶ `-i` – „**i**nteractive“: vor dem Überschreiben nachfragen
- ▶ `-u` – „**u**ppdate“: nur ältere Dateien überschreiben<sup>4</sup>
- ▶ `-v` – „**v**erbose“: Fortschritt anzeigen<sup>3,4</sup>

## Besonderheiten

- ▶ umbenennen: `mv <alterName> <neuerName>`

---

<sup>4</sup>Funktioniert auch mit `cp`

# Dateien löschen – rm

## Wichtige Optionen

- ▶ `-f` – „**f**orce“: Löschen erzwingen
  - ▶ Datei noch geöffnet, Schreibschutz, etc.
- ▶ `-i` – „**i**nteractive“: vor dem Löschen nachfragen
- ▶ `-r` oder `-R` – „**R**ecursive“: Dateien rekursiv löschen

## Besonderheiten

- ▶ `rm` löscht standardmäßig ohne Nachfrage unwiederruflich
- ▶ `rm -r` löscht auch Verzeichnisse
- ▶ `rm -rf` besonders gefährlich

## Bedienung

- ▶ `man <Befehl>` – Handbuch zum <Befehl> anzeigen
- ▶ Navigation mit Pfeiltasten
- ▶ beenden mit <q>

## Wichtige Optionen

- ▶ `-k <Stichwort>` – „**k**eyword“: Sucht Befehle zum <Stichwort>



# Wichtige Befehle & Optionen

		ls	cp	mv	rm
„ <b>b</b> ackup“:	-b	✗	✓	✓	✗
„ <b>i</b> nteractive“:	-i	✗	✓	✓	✓
„ <b>R</b> ecursive“:	-R	✓	✓	✗	✓
„ <b>r</b> everse“:	-r	✓	✓ <sup>5</sup>	✗	✓ <sup>5</sup>
„ <b>u</b> ppdate“:	-u	✗	✓	✓	✗
„ <b>v</b> erbose“:	-v	✗	✓	✓	✓

<sup>5</sup>kennt nur „recursive“, das selbe wie -R

# Zusammenfassung & Ausblick

## Zusammenfassung

- ▶ Unix-Philosophie: alles einfach und übersichtlich halten
- ▶ grundlegende Shellbefehle kennengelernt

## Nächstes Mal

- ▶ weitere Shellbefehle
- ▶ Dateiinhalte verarbeiten
- ▶ Befehle miteinander verknüpfen