

# Arbeiten mit der Shell Teil 1

Linux-Kurs der Unix-AG

Zinching Dang

09. November 2015



UNIX  
AG

TU Kaiserslautern

**RH** Regionales  
Hochschul-  
Rechenzentrum  
Kaiserslautern **RK**

# Die Unix-Philosophie

- ▶ Es gibt viele kleine Programme
  - ▶ Können die jeweiligen Aufgaben gut lösen
  - ▶ Komplexe Aufgaben werden durch Verknüpfen dieser gelöst
- ▶ Wichtiges Paradigma: „Everything is a File“
  - ▶ Alles ist eine Datei, auch Verzeichnisse
  - ▶ Keine Registry, Systemsteuerung, etc.

# Die Komandozeile unter Linux

- ▶ Wird als „Shell“ bezeichnet
- ▶ Unterscheidung zwischen „Konsole“ und „Terminal“
  - ▶ Konsolen sind mit STRG+ALT+F[1-6] erreichbar
  - ▶ Terminals werden unter grafischen Oberflächen geöffnet
- ▶ Der „Prompt“ zeigt an, dass die Shell auf Befehle wartet
  - ▶ `benutzer@rechner:~$`

# Befehle in der Shell

- ▶ Werden eingegeben und mit Enter bestätigt

**Ausgabe** ▶ `benutzer@rechner:~$ls ← Befehl`  
Arbeitsfläche Dokumente Musik Vorlagen  
Bilder Downloads Videos

- ▶ Können Optionen und Argumente haben

`benutzer@rechner:~$ls -lh Dokumente/ ← Argument`  
insgesamt 12K **Optionen**  
▶ `-rw-r-r- 1 benutzer gruppe 4K Nov 6 Linux.txt`  
`-rw-r-r- 1 benutzer gruppe 4K Nov 6 ist.odt`  
`-rw-r-r- 1 benutzer gruppe 4K Nov 6 toll.ods`

# Optionen und Argumente

- ▶ Sind nicht immer notwendig
- ▶ Mehrere Optionen können zusammengefasst werden
  - ▶ `ls -l -h` wird zu `ls -lh`
- ▶ Optionen verändern das Verhalten des Befehls
  - ▶ Vergleiche `ls` und `ls -lh`
  - ▶ Die Option `-lh` zeigt erweiterte Informationen an
- ▶ Argumente sind Ausführungsgegenstand des Befehls
  - ▶ Vergleiche `ls` und `ls Dokumente`
  - ▶ `ls` zeigt den Inhalt des aktuellen Verzeichnisses an
  - ▶ `ls Dokumente` zeigt den Inhalt des Verzeichnisses „Dokumente“ an

# Allgemeines zu Befehlen

- ▶ Groß- und Kleinschreibung ist wichtig
- ▶ Tab-Completion nutzen, um Tippfehler zu vermeiden
- ▶ Viele Befehle dienen der Dateiverwaltung
  - ▶ Argumente sind häufig Dateien oder Verzeichnisse

# Absolute und relative Pfade

- ▶ Pfade zu Dateien und Verzeichnissen können auf verschiedene Arten angegeben werden
- ▶ **Absoluter Pfad:** vom Wurzelverzeichnis `/` ausgehend
  - ▶ `/home/linux-kurs/Bilder/tux.png`
- ▶ **Relativ Pfad:** relativ zum aktuellen Verzeichnis
  - ▶ `Bilder/tux.png`

# Verzeichnisse

- ▶ Absoluten Pfad des aktuellen Verzeichnisses ausgeben: `pwd`  
(„**p**rint **w**orking **d**irectory“)
- ▶ Aktuelles Verzeichnis: `.`
- ▶ Eltern-Verzeichnis des aktuellen Verzeichnisses: `..`
- ▶ Home-Verzeichnis: `~`
- ▶ Kombination mit relativen Pfaden:
  - ▶ `~/Bilder/tux.png`
  - ▶ `../irgend/wo/anders/notizen.txt`

# Übersicht

- ▶ `ls`: Verzeichnisinhalt auflisten
- ▶ `mkdir` und `rmdir`: Verzeichnisse anlegen und löschen
- ▶ `cd`: Verzeichnis wechseln
- ▶ `rm`: Dateien löschen
- ▶ `mv`: Dateien und Verzeichnisse verschieben
- ▶ `cp`: Dateien und Verzeichnisse kopieren

- ▶ `ls` („list“) listet den Verzeichnisinhalt auf
- ▶ wichtige Optionen:
  - ▶ `-a` : Versteckte Dateien anzeigen
  - ▶ `-R` : Rekursives Ausführen
  - ▶ `-l` : Weitere Informationen ausgeben
  - ▶ `-h` : Dateigröße in lesbaren Einheiten anzeigen
- ▶ Argumente:
  - ▶ Optional ein oder mehrere Verzeichnisse, standardmäßig wird das aktuelle Verzeichnis verwendet

- ▶ Versteckte Dateien/Verzeichnisse beginnen mit einem `.`
- ▶ Die Option `-l` zeigt zusätzlich Dateirechte, Besitzer, Größe und Änderungsdatum an
- ▶ Dateigrößen werden normalerweise in Bytes angezeigt
- ▶ Weitere, für die Prüfung wichtige, Optionen: Buch, Kapitel 6.2.2

# mkdir und rmdir

- ▶ `mkdir` („**make directory**“): legt Verzeichnisse an
- ▶ `rmdir` („**remove directory**“): löscht leere Verzeichnisse
- ▶ wichtige Option:
  - ▶ `-p` legt Elternverzeichnisse automatisch an/löscht diese, falls leer
- ▶ Argumente:
  - ▶ Verzeichnisse, die angelegt/gelöscht werden sollen

# mkdir und rmdir

- ▶ `mkdir Dokumente/Studium/WS15/Linuxkurs1`
  - ▶ Legt das Verzeichnis „Linuxkurs1“ in „Dokumente/Studium/WS15/“ an
  - ▶ Das Verzeichnis „Dokumente/Studium/WS15/“ muss existieren
- ▶ `mkdir -p Dokumente/Studium/WS15/Linuxkurs1`
  - ▶ „Dokumente/Studium/WS15/Linuxkurs1“ wird angelegt, ggf. werden fehlende Verzeichnisse auch angelegt
- ▶ `rmdir Dokumente/Studium/WS15/Linuxkurs1`
  - ▶ Verzeichnis „Linuxkurs1“ und „Dokumente/Studium/WS15/“ löschen
- ▶ `rmdir -p Dokumente/Studium/`
  - ▶ Verzeichnis „Dokumente/Studium/“ samt Unterverzeichnisse löschen

# cd

- ▶ `cd` („change directory“): in ein anderes Verzeichnis wechseln
- ▶ keine wichtigen Optionen
- ▶ Argument: Verzeichnis, in das gewechselt werden soll
  - ▶ Angabe als relativer oder absoluter Pfad
  - ▶ `~`, `.` und `..` können im Pfad benutzt werden
  - ▶ `cd -` wechselt in das vorherige Verzeichnis
  - ▶ Wird kein Argument angegeben, wechselt `cd` ins Home-Verzeichnis (entspricht `cd ~`)

- ▶ `rm` („**r**emove“): löscht Dateien und Verzeichnisse
- ▶ wichtige Optionen:
  - ▶ `-r` oder `-R`: löscht Verzeichnisse mit Inhalt rekursiv
  - ▶ `-f`: überhaupt keine Fragen stellen (z. B. bei schreibgeschützten Dateien)
  - ▶ `-i`: vor dem Löschen jeder einzelnen Datei nachfragen
- ▶ Argumente: beliebig viele Dateien und/oder Verzeichnisse

- ▶ Achtung: `rm` löscht Dateien unwiederbringlich ohne Nachfrage!
- ▶ Verzeichnisse können nur mit der Option `-r` gelöscht werden
  - ▶ Löscht zuerst alle Dateien im angegebenen Verzeichnis und dessen Unterverzeichnisse
  - ▶ Anschließend werden leere Verzeichnisse gelöscht
- ▶ `rm -rf` ist besonders gefährlich
  - ▶ Löscht das Verzeichnis samt Inhalt ohne Nachfrage

# Dateien anlegen

- ▶ normalerweise werden Dateien angelegt, wenn man etwas darin speichert
- ▶ zu Testzwecken kann auch `touch` verwendet werden
- ▶ `touch` setzt eigentlich das Änderungsdatum der Datei auf das aktuelle Datum
- ▶ nicht existente Dateien werden leer angelegt

- ▶ `mv` („**move**“) verschiebt Dateien und Verzeichnisse
- ▶ wichtige Optionen:
  - ▶ `-b`: legt Sicherungskopien an, wenn Dateien überschrieben werden
  - ▶ `-i`: fragt vor dem Überschreiben nach
- ▶ Argumente:
  - ▶ zwei Dateinamen: Datei wird umbenannt
  - ▶ beliebig viele Dateien/Verzeichnisse und Verzeichnis als letztes Argument: Dateien werden in das Verzeichnis verschoben

# mv

- ▶ `mv geheim unwichtig`
  - ▶ Datei „geheim“ in „unwichtig“ umbenennen
- ▶ `mv shell1/ shell2/ Teilnehmer.txt VL/`
  - ▶ Verzeichnisse „shell1“ und „shell2“, sowie „Teilnehmer.txt“ in das Verzeichnis „VL“ verschieben

- ▶ `cp` („copy“) kopiert Dateien und Verzeichnisse
- ▶ wichtige Optionen:
  - ▶ `-b` : legt Sicherungskopien an, wenn Dateien überschrieben werden
  - ▶ `-i` : fragt vor dem Überschreiben nach
  - ▶ `-r` oder `-R` : kopiert Verzeichnisse rekursiv
- ▶ Argumente:
  - ▶ zwei Dateinamen: Kopie der Datei wird angelegt
  - ▶ beliebig viele Dateien/Verzeichnisse und Verzeichnis als letztes Argument: Dateien werden in das Verzeichnis kopiert

- ▶ Verzeichnisse mit Inhalt müssen rekursiv kopiert werden
- ▶ weitere, für die Prüfung wichtige, Optionen: Buch, Kapitel 6.4.1

# nano

- ▶ sehr einfach zu bedienender Kommandozeilen-Texteditor
- ▶ optionales Argument: Dateiname
  - ▶ wenn die Datei nicht existiert, wird sie angelegt
  - ▶ vorhandene Datei wird geladen
- ▶ Speichern mit Strg-O, Beenden mit Strg-X (^ entspricht Strg)

# Spickzettel

Befehl	Optionen	Funktion
pwd		Gibt aktuelles Verzeichnis aus
ls	-a, -R, -l, -h	Verzeichnisinhalte auflisten
cd		Verzeichnis wechseln
mkdir	-p	Verzeichnisse anlegen
rmdir	-p	Leere Verzeichnisse löschen
rm	-r / -R, -f	Dateien/Verzeichnisse löschen
touch		Leerer Dateien anlegen
mv	-b, -i	Dateien/Verzeichnisse verschieben
cp	-b, -i, -r / -R	Dateien/Verzeichnisse kopieren
nano		Ein Texteditor