

Arbeiten mit der Shell Teil 1

Linux-Kurs der Unix-AG

Zinching Dang

15. November 2015



TU Kaiserslautern

RH Regionales
Hochschul-
Rechenzentrum **RK**
Kaiserslautern



Übersicht

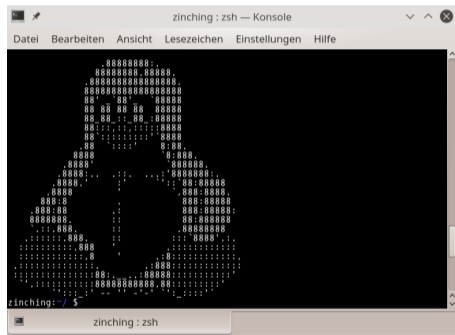
Die Unix-Philosophie

Wiederholung & Vertiefung

Shellbefehle

Befehle & Optionen

Zusammenfassung & Ausblick



Tux im Terminal

Die Unix-Philosophie

Wichtiges Paradigma: „Everything is a File“

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

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

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

Wiederholung: Pfade

Pfadangaben sind

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

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

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 auflisten¹
- ▶ `-r` – „**r**everse“: Sortierreihenfolge umdrehen
- ▶ `-S` – „**S**ort by size“: der Größe nach sortieren
- ▶ `-t` – „**t**ime“: nach Änderungsdatum sortieren

¹Vgl. `cp`, `rm`

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

Verzeichnis wechseln & -befehle – cd & pwd

Besonderheiten von cd

- ▶ hat keine nennenswerten 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` – aktuelles Verzeichnis anzeigen

Verzeichnisse erstellen & löschen – mkdir & rmdir

Wichtige Option

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

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>

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 umbenennen²
- ▶ `-i` – „**i**nteractive“: vor dem Überschreiben nachfragen^{2,3}
- ▶ `-p` – „**p**reserve“: Zusatzinformationen beibehalten
 - ▶ Änderungsdatum, Besitzer, etc.
- ▶ `-r` oder `-R` – „**R**ecursive“: Dateien rekursiv kopieren
 - ▶ ermöglicht das Kopieren von Verzeichnissen

²Vgl. mv

³Vgl. rm

Wichtige Optionen

- ▶ `-b` – „**b**ackup“: vorhandene Dateien vorher umbenennen
- ▶ `-i` – „**i**nteractive“: vor dem Überschreiben nachfragen
- ▶ `-u` – „**u**ppdate“: nur ältere Dateien überschreiben⁴
- ▶ `-v` – „**v**erbose“: Fortschritt anzeigen^{3,4}

⁴Funktioniert auch mit cp

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 überschreiben⁴
- ▶ `-v` – „**v**erbose“: Fortschritt anzeigen^{3,4}

Besonderheiten

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

⁴Funktioniert auch mit `cp`

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

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>

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 ackup“:	-b	✗	✓	✓	✗
„ i nteractive“:	-i	✗	✓	✓	✓
„ R ecursive“:	-R	✓	✓	✗	✓
„ r everse“:	-r	✓	✓ ⁵	✗	✓ ⁵
„ u ppdate“:	-u	✗	✓	✓	✗
„ v erbose“:	-v	✗	✓	✓	✓

⁵kennt nur „recursive“, das selbe wie -R

Zusammenfassung & Ausblick

Zusammenfassung

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

Zusammenfassung & Ausblick

Zusammenfassung

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

Nächstes Mal

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