

# Toolbox Workshop 2013

Igor Babuschkin   Kevin Dungs   Ismo Toijala

29. Juli 2014



**PEP ET AL. E.V.**  
PHYSIKSTUDIERENDE UND  
EHMALIGE PHYSIKSTUDIERENDE  
DER TU DORTMUND

# Unix Shell

# Dateisystem

- bildet *einen* Baum
  - beginnt bei / (root)
  - / trennt Teile eines Pfads
  - auf Groß-/Kleinschreibung achten!
- es gibt ein aktuelles Verzeichnis
- relative vs. absolute Pfade
- spezielle Verzeichnisse:
  - . das aktuelle Verzeichnis
  - .. das Oberverzeichnis
  - ~ das Homeverzeichnis

## man, pwd, cd

<code>man <i>topic</i></code>	„manual“: zeigt die Hilfe für ein Programm
<code>pwd</code>	„print working directory“: zeigt das aktuelle Verzeichnis
<code>cd <i>directory</i></code>	„change directory“: wechselt in das angegebene Verzeichnis

# ls

- ls [*directory*] „list“: zeigt den Inhalt eines Verzeichnisses an
- ls -l „long“: zeigt mehr Informationen über Dateien und Verzeichnisse
- ls -a „all“: zeigt auch versteckte Dateien (fangen mit . an)

## mkdir, touch

<code>mkdir <i>directory</i></code>	„make directory“: erstellt ein neues Verzeichnis
<code>mkdir -p <i>directory</i></code>	„parent“: erstellt auch alle notwendigen Oberverzeichnisse
<code>touch <i>file</i></code>	erstellt eine leere Datei

## cp, mv, rm, rmdir

<code>cp source destination</code>	„copy“: kopiert eine Datei
<code>cp -r source destination</code>	„recursive“: kopiert ein Verzeichnis rekursiv
<code>mv source destination</code>	„move“: verschiebt eine Datei (Umbenennung)
<code>rm file</code>	„remove“: löscht eine Datei (Es gibt keinen Papierkorb!)
<code>rm -r directory</code>	„recursive“: löscht ein Verzeichnis rekursiv
<code>rmdir directory</code>	„remove directory“: löscht ein leeres Verzeichnis

## cat, less, grep, echo

`cat file`

„concatenate“: gibt Inhalt einer (oder mehr) Datei(en) aus

`less file`

(besser als `more`): wie `cat`, aber navigabel

`grep pattern file`

`g/re/p`: sucht in einer Datei nach einem Muster

`grep -i pattern file`

„case insensitive“

`grep -r pattern directory`

„recursive“: suche rekursiv in allen Dateien

`echo message`

gibt einen Text aus



# Ein- und Ausgabe

<code>command &gt; file</code>	überschreibt Datei mit Ausgabe
<code>command &gt;&gt; file</code>	fügt Ausgabe einer Datei hinzu
<code>command &lt; file</code>	Datei als Eingabe
<code>command1   command2</code>	Ausgabe als Eingabe (Pipe)

# Tastaturkürzel

- `Ctrl-C` beendet das laufende Programm
- `Ctrl-D` EOF (end of file) eingeben, kann Programme beenden
- `Ctrl-L` leert den Bildschirm

# Globber

- \* wird ersetzt durch alle passenden Dateien
- {a,b} bildet alle Kombinationen

Beispiele:

\*.log → foo.log bar.log

foo.{tex,pdf} → foo.tex foo.pdf