

**Eine Einführung für Erstsemester**

B.Sc. Malte Grave  
*Fachschaft Informatik*

*Universität Oldenburg*

- 1 Über mich
- 2 Formalia
- 3 Was ist Unix?
- 4 Am Rechner
- 5 Shell
- 6 Dateisystem
- 7 Shell-Programme
- 8 Grafische Oberfläche
- 9 Hilfe, Tipps und Tricks

1 Über mich

2 Formalia

3 Was ist Unix?

4 Am Rechner

5 Shell

6 Dateisystem

7 Shell-Programme

8 Grafische Oberfläche

9 Hilfe, Tipps und Tricks

# Wer bin ich überhaupt?

## ■ B.Sc. Malte Grave

- ▶ 23 Jahre alt
- ▶ Aktuell im 2. Master Semester
- ▶ FSR-Mitglied seit 2019 (WiSe)
- ▶ Wissenschaftlicher Mitarbeiter am OFFIS (Gruppe DCC)
- ▶ Motorrad fahren, Musik hören, spielen und schreiben, ...

## ■ Kontakt

- ▶ Mail: [malte.grave@uni-oldenburg.de](mailto:malte.grave@uni-oldenburg.de)
- ▶ Discord: daylien, Fachschafts Discord
- ▶ Einfach Anquatschen



- 1 Über mich
- 2 Formalia**
- 3 Was ist Unix?
- 4 Am Rechner
- 5 Shell
- 6 Dateisystem
- 7 Shell-Programme
- 8 Grafische Oberfläche
- 9 Hilfe, Tipps und Tricks

# Beantragung eines Logins

- Department für Informatik
  - ▶ Eigenes Rechnernetz, eigene Rechnerräume und eigene **Abteilung Rechner- und Netzbetrieb Informatik** (ARBI)
  - ▶ Informatik-Studierende arbeiten auf **ARBI-Rechnern** mit **Informatik-Logins** und **Informatik-Email-Adresse** (vorname.nachname@informatik.uni-oldenburg.de)

- Jeder Benutzer besitzt:

- ▶ einen Login-Namen
- ▶ ein Passwort

*Diese müssen beim Anmelden am Rechner eingegeben werden.*

- ▶ ein (home-)Verzeichnis (home directory), in dem seine Daten gespeichert werden. *Der Name des home-Verzeichnis ist gleich dem Login-Namen. Die home-Verzeichnisse aller Studierenden eines Semesters befinden sich im selben Unterverzeichnis: /user/stud/s15/*

- Das erste Passwort wird von der ARBI vergeben.
  - ▶ Automatisch generierte, beliebige Folge von Buchstaben und Ziffern
- Passwörter können geändert werden.
- Passwort vergessen:
  - ▶ Jörg Lehnert oder Olaf Wendt ansprechen



- 1 Über mich
- 2 Formalia
- 3 Was ist Unix?**
- 4 Am Rechner
- 5 Shell
- 6 Dateisystem
- 7 Shell-Programme
- 8 Grafische Oberfläche
- 9 Hilfe, Tipps und Tricks

- Betriebssystem
  - ▶ Entwickelt von den Bell Laboratories (Start ca. 1969)
  - ▶ Ken Thompson, Dennis Ritchie
- 1989 Unix System Release 4.0 = heutiger 'Standard'
- BSD-Unix: Erweiterung von Unix, entwickelt von der Universität von Kalifornien in Berkley
- Viele Derivate von Unix: Linux, Solaris/Sun-OS, Ultrix, FreeBSD, Mac OS X, ...
  - ▶ Richard Stallman, Linus Torvalds
- 'Shell' als Kommandoschnittstelle

- Als Mehrbenutzersystem ausgelegt:
  - ▶ Mehrere Personen können zugleich an demselben Rechner arbeiten
- Systemadministrator richtet neuen Benutzern die 'Arbeitserlaubnis', ein Login, auf dem Rechner ein.
- Arbeitsverzeichnisse liegen auf zentralem Server (**weser**).

- 1 Über mich
- 2 Formalia
- 3 Was ist Unix?
- 4 Am Rechner**
- 5 Shell
- 6 Dateisystem
- 7 Shell-Programme
- 8 Grafische Oberfläche
- 9 Hilfe, Tipps und Tricks

# Erste Schritte am Rechner

- 1 Anmelden (Einloggen) am Rechner
  - ▶ Achtung: Der Cursor muss sich auf dem entsprechenden Feld befinden.
  - ▶ Eingabe des Passwortes:  
*Das Passwort muss eingetippt werden. Man sieht nicht, was für Zeichen man tippt!*
- 2 Die 'Erstsemesterbenutzungsoberfläche'
- 3 Ändern des Passwortes
- 4 Sperren des Arbeitsplatzes
- 5 Abmelden

- Ein Unixbefehl hat die Form (Syntax)

Shell

```
<Befehlsname> [-opt] [args]
```

- ▶ Der Befehlsname besteht nur aus Kleinbuchstaben.
- ▶ opt sind Optionen des Befehls.
- ▶ args sind die Argumente des Befehls.
- ▶ bei der Beschreibung eines Befehls bedeutet [...], dass der Teil in den Klammern auch weggelassen werden darf. Beim Aufruf des Befehls werden diese Klammern nicht eingegeben.

- Ein Befehl kann im xterm nach dem Prompt eingegeben werden.

- 1 Über mich
- 2 Formalia
- 3 Was ist Unix?
- 4 Am Rechner
- 5 Shell**
- 6 Dateisystem
- 7 Shell-Programme
- 8 Grafische Oberfläche
- 9 Hilfe, Tipps und Tricks

# Was ist eine Shell?

- Benutzer kommuniziert mit dem Betriebssystem über eine Shell
- Shell = Kommandointerpreter
  - ▶ Nimmt Kommandos des Benutzers entgegen und
  - ▶ setzt sie in Systemaufrufe um, die von Betriebssystemkern ausgeführt werden.



# Was ist eine Shell?

- Unterschiedliche Shells (tcsh, bash, zsh, . . . ) auf einem System einstellbar
- Ist die Shell zum Annehmen eines Befehls bereit, meldet sie sich mit einem so genannten Prompt, z.B. '>'
- Aufbau der Eingabeaufforderung der shell:

Shell

```
<Benutzer>@<Rechnername>:<aktuelles Directory><Prompt>
```

```
ersti@weser:~>
```

## ■ cd

*Change Directory*: Wechseln zwischen Verzeichnissen

Shell

```
> cd <ordner>
```

## ■ Beispiele:

- ▶ Mit `cd Aufgabe1` wechselt man ins Verzeichnis Aufgabe1
- ▶ Mit `cd ..` wechselt man ins übergeordnete Verzeichnis
- ▶ Mit `cd ../ap` wechselt man von Verzeichnis java in das Verzeichnis ap auf der gleichen Ebene

# ls – Listing

Auflisten der Dateien im aktuellen Verzeichnis

- Syntax des ls-Befehls:

Shell

```
> ls [-aAbcCdFfgilMnOpqrRstux1] [ file..]
```

*Die meisten Optionen werden nur selten gebraucht.*

## Beispiele

- Alle Dateien (ohne versteckte Dateien):
- Alle Dateien mit zusätzlichen Informationen:
- Alle Dateien mit zusätzlichen Informationen und in 'human readable':
- Anzeigen auch der versteckten Dateien:

# ls – Listing

Auflisten der Dateien im aktuellen Verzeichnis

- Syntax des ls-Befehls:

Shell

```
> ls [-aAbcCdFgILmnoPqrRstux1] [ file..]
```

*Die meisten Optionen werden nur selten gebraucht.*

## Beispiele

- Alle Dateien (ohne versteckte Dateien):

Shell

```
> ls
```

- Alle Dateien mit zusätzlichen Informationen:
- Alle Dateien mit zusätzlichen Informationen und in 'human readable':
- Anzeigen auch der versteckten Dateien:

# ls – Listing

Auflisten der Dateien im aktuellen Verzeichnis

- Syntax des ls-Befehls:

Shell

```
> ls [-aAbcCdFgILmnoPqrRstux1] [ file..]
```

*Die meisten Optionen werden nur selten gebraucht.*

## Beispiele

- Alle Dateien (ohne versteckte Dateien):
- Alle Dateien mit zusätzlichen Informationen:

Shell

```
> ls -l  
oder  
> ll
```

- Alle Dateien mit zusätzlichen Informationen und in 'human readable':
- Anzeigen auch der versteckten Dateien:

# ls – Listing

Auflisten der Dateien im aktuellen Verzeichnis

- Syntax des ls-Befehls:

Shell

```
> ls [-aAbCdFgILmnoPqRstux1] [ file..]
```

*Die meisten Optionen werden nur selten gebraucht.*

## Beispiele

- Alle Dateien (ohne versteckte Dateien):
- Alle Dateien mit zusätzlichen Informationen:
- Alle Dateien mit zusätzlichen Informationen und in 'human readable':

Shell

```
> ls -lh  
oder  
> lh
```

- Anzeigen auch der versteckten Dateien:

# ls – Listing

Auflisten der Dateien im aktuellen Verzeichnis

- Syntax des ls-Befehls:

Shell

```
> ls [-aAbcCdFgILmnoPqrRstux1] [ file..]
```

*Die meisten Optionen werden nur selten gebraucht.*

## Beispiele

- Alle Dateien (ohne versteckte Dateien):
- Alle Dateien mit zusätzlichen Informationen:
- Alle Dateien mit zusätzlichen Informationen und in 'human readable':
- **Anzeigen auch der versteckten Dateien:**

Shell

```
> ls -a  
oder  
> la
```

# mv; mkdir; rmdir; cp; rm – Verwaltung

- mv

*Move*: Verschieben/Umbenennen von Dateien

- mkdir

*Make Directory*: Ordner erstellen

- rmdir

*Remove Directory*: Leeren Ordner löschen

- cp

*Copy*: Kopieren von Dateien

- rm

*Remove*: Dateien löschen



# mv; mkdir; rmdir; cp; rm – Verwaltung

- mv

*Move: Verschieben/Umbenennen von Dateien*

Shell

```
> mv <quelle> <ziel>
```

- mkdir

*Make Directory: Ordner erstellen*

- rmdir

*Remove Directory: Leeren Ordner löschen*

- cp

*Copy: Kopieren von Dateien*

- rm

*Remove: Dateien löschen*

# mv; mkdir; rmdir; cp; rm – Verwaltung

- mv

*Move*: Verschieben/Umbenennen von Dateien

- mkdir

*Make Directory*: Ordner erstellen

Shell

```
> mkdir <ordnername>
```

- rmdir

*Remove Directory*: Leeren Ordner löschen

- cp

*Copy*: Kopieren von Dateien

- rm

*Remove*: Dateien löschen

# mv; mkdir; rmdir; cp; rm – Verwaltung

- mv

*Move*: Verschieben/Umbenennen von Dateien

- mkdir

*Make Directory*: Ordner erstellen

- rmdir

*Remove Directory*: Leeren Ordner löschen

Shell

```
> rmdir <ordnername>
```

- cp

*Copy*: Kopieren von Dateien

- rm

*Remove*: Dateien löschen

# mv; mkdir; rmdir; cp; rm – Verwaltung

- mv

*Move*: Verschieben/Umbenennen von Dateien

- mkdir

*Make Directory*: Ordner erstellen

- rmdir

*Remove Directory*: Leeren Ordner löschen

- cp

*Copy*: Kopieren von Dateien

Shell

```
> cp <quelle> <ziel>  
> cp -R <quelle> <ordner>
```

- rm

*Remove*: Dateien löschen

# mv; mkdir; rmdir; cp; rm – Verwaltung

- mv

*Move*: Verschieben/Umbenennen von Dateien

- mkdir

*Make Directory*: Ordner erstellen

- rmdir

*Remove Directory*: Leeren Ordner löschen

- cp

*Copy*: Kopieren von Dateien

- rm

*Remove*: Dateien löschen

Shell

```
> rm <datei>  
> rm -R <ordner>
```

# cat; less; tail – Textausgabe

- cat  
Ausgabe von Dateiinhalten
- less  
Anzeige von Dateiinhalten (scroll- und durchsuchbar)
- tail  
Anzeige der letzten Zeilen einer Datei

# cat; less; tail – Textausgabe

- cat

Ausgabe von Dateiinhalten

Shell

```
> cat <datei>
```

- less

Anzeige von Dateiinhalten (scroll- und durchsuchbar)

- tail

Anzeige der letzten Zeilen einer Datei

# cat; less; tail – Textausgabe

- `cat`  
Ausgabe von Dateiinhalten
- `less`  
Anzeige von Dateiinhalten (scroll- und durchsuchbar)

Shell

```
> less <datei>
```

- `tail`  
Anzeige der letzten Zeilen einer Datei



# cat; less; tail – Textausgabe

- cat  
Ausgabe von Dateiinhalten
- less  
Anzeige von Dateiinhalten (scroll- und durchsuchbar)
- tail  
Anzeige der letzten Zeilen einer Datei

Shell

```
> tail <datei>  
> tail -f <datei>
```

# Ein- und Ausgabeströme

- Programmaufrufe produzieren in der Regel Ausgaben
- Oft ist es nützlich diese weiterzuverarbeiten oder zu speichern.
- 4 wichtige Operatoren: `>`, `<`, `|`, `>>`
- `>`: ist die sog. Outputpipe: `cat file > file2`
- `<`: die Inputpipe: `mysql -p < table.sql`
- `|`: Weiterleitung von Ausgaben:  
`cat file | grep pattern`
- `>>`: Genau wie `>`, aber die Ausgabe wird ans Dateiende angehängt

# nano – Einfache Textbearbeitung

- Einfacher Texteditor

- Syntax:

Shell

```
> nano [<dateiname>]
```

- Unterer Teil zeigt mögliche Befehle
- ^X gibt an, dass Ctrl+X eingegeben werden soll

- Zum Teil sehr komplexe Texteditoren
- Haben verschiedene Modi
- Benötigen Einarbeitungszeit

- vim

Shell

```
> vimtutor
```

- Für emacs → Google

- 1 Über mich
- 2 Formalia
- 3 Was ist Unix?
- 4 Am Rechner
- 5 Shell
- 6 Dateisystem**
- 7 Shell-Programme
- 8 Grafische Oberfläche
- 9 Hilfe, Tipps und Tricks

- Arten von Dateien
- Struktur des Dateisystems
- Arbeiten mit Dateiverzeichnissen
- Arbeiten mit Dateien

- Verzeichnisse
- Normale Dateien (files)
  - ▶ Textdateien  
Beliebiger Text, Programmtexte, ...  
Dot-Files (versteckte Dateien) enthalten Unix-Kommandos, Systemeinstellungen, ...  
Die Datei `.forward` enthält die Email-Adresse, an die eingehende emails weitergeleitet werden sollen.

- Normale Dateien (files)

- ▶ Binärdateien

- Ausführbare Programme

- Dateien in speziellen Formaten, die nur mit speziellen Programmen angezeigt werden können

- (pdf-Dateien, Postscript-Dateien, Bilder, ...)

- ⋮



# Einige besondere Dateien

- Verzeichnisse:
  - ▶ bin: ausführbare Programme
  - ▶ Mail: Dokumente zu ein- und ausgehenden Emails
  - ▶ viele weitere
- 'Versteckte' Dateien: Dot-Dateien (Punktdateien)
  - ▶ Dateiname beginnt mit einem Punkt
  - ▶ Beinhalten System- und Programmeinstellungen
  - ▶ Beispiele:
    - .forward
    - .signature

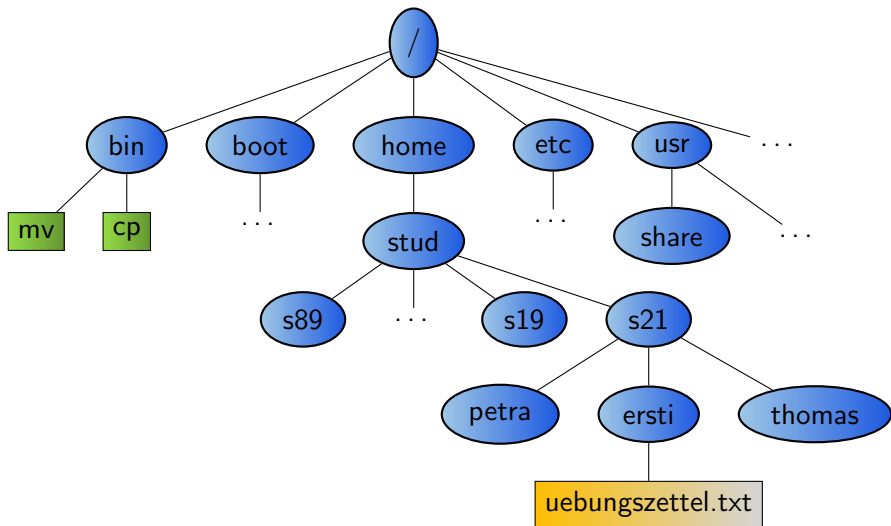
# Zugriffsrechte: `rwxr-xr-x`

- Für jede Datei kann festgelegt werden, wer das Leserecht (r), Schreibrecht (w) und Ausführungsrecht (x) besitzt:
- Die ersten 3 Stellen → Besitzer
- Die zweiten 3 Stellen → Gruppe
- Die letzten 3 Stellen → Andere
- Verändern mit → **chmod**  
z.B. `chmod o-rx bin/`

# Struktur des Dateisystems

- Baumartig (hierarchisch) organisiert: Verzeichnisse (directories) können Dateien und auch wieder Verzeichnisse enthalten
- Das **root**-Verzeichnis ist das Verzeichnis, das alle anderen enthält
- Das **home**-Verzeichnis (home-directory) ist das Verzeichnis, das alle Dateien und Verzeichnisse des entsprechenden Benutzers enthält.
  - ▶ ~ ersti ist das home-Verzeichnis des Benutzers ersti.
- Das Arbeitsverzeichnis (working directory) '.' ist das Verzeichnis, in dem man sich 'gerade befindet'.

# Struktur des Dateisystems – Bild



- Der **Pfad** ist die Wegbeschreibung zu einer Datei
  - ▶ `/home/stud/s19/ersti` ist der Pfad aller Dateien, die im Verzeichnis `testv` gespeichert sind.
  - ▶ Ein **Punkt** (`.`) bezeichnet stets das **aktuelle** Verzeichnis, i.e. Arbeitsverzeichnis.
  - ▶ Eine **Tilde** (`~`) bezeichnet stets das home-Verzeichnis.
  - ▶ `~<login-Name>` ist das home-Verzeichnis des Benutzers mit `<login-Name>`
  - ▶ **Zwei Punkte** (`..`) bezeichnen das übergeordnete Verzeichnis.

- enthalten ausführbare Programme
- Beispiel:
  - ▶ `/usr/local/X11/bin`  
enthält Befehle zum Aufruf von Anwendungsprogrammen, z.B. `xpdf`
- `xpdf` kann ohne Pfadangabe aufgerufen werden, da sein Pfad im `PATH` eingetragen ist.
- Alle Verzeichnisse, die in der `PATH`-Variablen angegeben sind, werden automatisch durchsucht.

- Problem:
  - ▶ In Firefox muss der auszuführende Befehl **mit korrektem Pfad** eingegeben werden.
- Finden des Pfades für einen Befehl:  
Befehl `which <befehl>`  
`which xpdf` liefert `/usr/local/X11/bin/xpdf`

- 1 Über mich
- 2 Formalia
- 3 Was ist Unix?
- 4 Am Rechner
- 5 Shell
- 6 Dateisystem
- 7 Shell-Programme**
- 8 Grafische Oberfläche
- 9 Hilfe, Tipps und Tricks



- Programm um Dateiströme zu filtern.
- Die wichtigsten Switches sind:
  - ▶ -i für caseinsensitiv
  - ▶ -R für rekursiv
  - ▶ -n für die Zeilennummer
- Beispiel:

Shell

```
> grep -in icewm .xsession  
37:icewm
```

# ssh (secure shell)

- SSH ist ein Protokoll zur Fernadministration. Im Gegensatz zu z.B. Telnet wird die Verbindung verschlüsselt.
- Es eignet sich wunderbar um in der Arbi von zuhause aus zu Arbeiten.
- GUI Anwendungen können per X-Forwarding von anderen Rechnern aus gestartet werden. (switch -Y)
- Beispiel:

Shell

```
> ssh -Y ersti@weser.informatik.uni-oldenburg.de
```

# scp (secure copy)

- scp ist ein Tool um per ssh Dateien zu übertragen.

Shell

```
> scp <source> <destination>
```

- Beispiel:

Shell

```
> scp testfile weser:/tmp
```

- Hier ist *testfile* die source und *weser:/tmp* die destination

# lpr; lpq; lprm – Drucken in der ARBI

- 'Manuelles' drucken von Dateien
- Müssen dafür erst ins Postscript-Format umgewandelt werden
- lpr Drucken:

Shell

```
> lpr [-P <Druckername>] <Dateiname>
```

- lpq Warteschlange:

Shell

```
> lpq [-P <Druckername>]
```

- lprm Löschen:

Shell

```
> lprm [-P <Druckername>] <job-id>  
> lprm [-P <Druckername>] all
```

- Oft ist es nicht nötig Dateien 'manuell' in das Postscript-Format umzuwandeln
- Man muss für diese Umwandlung nicht unbedingt eine grafische Oberfläche bemühen
- Druckbefehle für (fast) alle Dateien

Shell

```
> a2ps [-P <Druckername>] <Dateiname>
```

- a2ps wandelte eigentlich alle textbasierten Dateiformate (und PDF) in das Postscript-Format um und druckt es
- Formate wie DOC, DOCX, ODT usw. können so leider nicht gedruckt werden. In diesen Fällen muss erst in PDF umgewandelt werden.

# Wieviel darf ich drucken?

- Für jeden Student wird ein **Druckquota** von maximal 50 Seiten verwaltet, d.h. es dürfen maximal 50 Seiten pro Tag gedruckt werden.
- Bei jedem Druckauftrag wird das Druckquota entsprechend verringert.
- Das Druckquote wird jeden Tag um 5 Seiten erhöht, wächst aber höchstens auf 50 Seiten an.
- Man erhält per EMail eine Benachrichtigung über sein Druckquota-Konto.

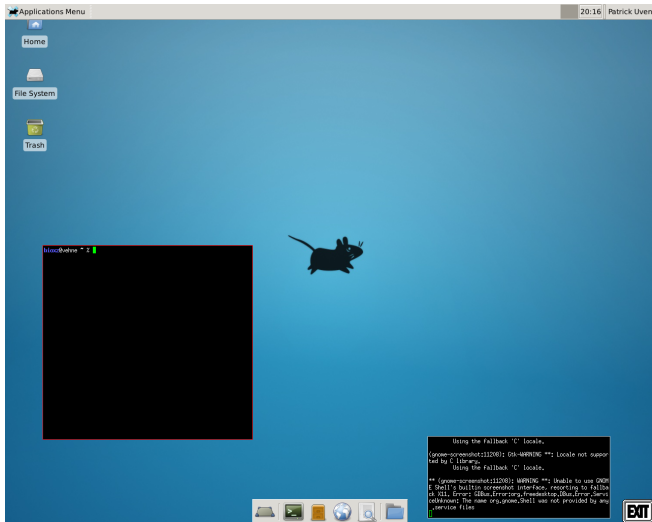
- 1 Über mich
- 2 Formalia
- 3 Was ist Unix?
- 4 Am Rechner
- 5 Shell
- 6 Dateisystem
- 7 Shell-Programme
- 8 Grafische Oberfläche**
- 9 Hilfe, Tipps und Tricks

# Die Ersti-Benutzungsoberfläche

- Die Ersti-Oberfläche besteht aus einem Windowmanager, welcher die Fenster kontrolliert
- sowie aus Panels, welche die Benutzung erleichtern sollen.
- Basis der Ersti-Oberfläche ist das Xfce Desktop Environment
- Weitere grafische Programme für viele Aufgaben verfügbar



# Die Ersti-Benutzungs Oberfläche



- X: *Xfce-Startmenü (Programme, Einstellungen)*
- WindowList: *Übersicht der geöffneten Fenster*
- Workspaces: *Virtuelle Monitore*
- SysTray: *Infoanzeige von Programmen (Instant Messenger)*
- Lock Button: *Sperrt den Bildschirm*
- Logout Button: *Beendet die Session*
- Uhr: *Zeigt Zahlen und einen Doppelpunkt*
- Quickstartpanel: *Bietet Schnellstartverknüpfungen*
- Exit-Button: *Beendet ebenfalls die Session*

## ■ Internet-Browser

- ▶ Mozilla Firefox (firefox), Opera (opera),  
Mozilla Seamonkey (seamonkey)

## ■ PDF-Reader

- ▶ Xpdf (xpdf), Okular (okular (KDE Desktop))

## ■ Office-Tools

- ▶ LibreOffice Writer, Calc & Impress (libreoffice),  
AbiWord (abiword), Texmaker (texmaker)

- 1 Über mich
- 2 Formalia
- 3 Was ist Unix?
- 4 Am Rechner
- 5 Shell
- 6 Dateisystem
- 7 Shell-Programme
- 8 Grafische Oberfläche
- 9 Hilfe, Tipps und Tricks**

# HILFE!!!!1!!11!! Help!

Viele Möglichkeiten, Hilfe zu erhalten:

- man-pages aufrufen → Erläuterungen zu Unix-Befehlen
- Kommilitoninnen und Kommilitonen fragen
- TutorInnen löchern, z.B. die Tutoren des Programmierkurses
- In Internet und Newsgroups informieren
  - ▶ ArchWiki <https://wiki.archlinux.org>
  - ▶ FreeBSD Wiki <https://www.freebsd.org/cgi/man.cgi>
  - ▶ Linux man pages <https://linux.die.net/man/>
- Mitarbeiter der ARBI fragen

- Datei- und Verzeichnisnamen müssen nicht immer vollständig ausgeschrieben werden.
- Die Shell (die im xterm läuft) vervollständigt Dateinamen, wenn die **Tabulatortaste** gedrückt wird.
- Mengen von Namen können mit Hilfe von Sonderzeichen beschrieben werden. Dabei bedeuten:
  - \* : beliebige viele Zeichen
  - ? : genau ein beliebiges Zeichen
  - [ ] : genau ein Zeichen aus der in den Klammern angegebenen Menge
- ??\_\*[a-z] beschreibt somit alle Dateinamen, die an der dritten Stelle einen Unterstrichstrich besitzen und mit einem Kleinbuchstaben enden.

- Installiert euch VirtualBox, QEMU, Hyper-V oder sonstige VM Programme und setzt mal euer eigenes FreeBSD oder andere UNIX Derivate auf
- Anleitungen findet Ihr im Internet überall
- Bei Fragen gerne an mich wenden
- Live demo?

# Unix zum Ausprobieren...

- Wer das hier Gelernte zu Hause ausprobieren möchte, sich aber kein BSD- oder GNU/Linux-System installieren möchte, findet unter

*<http://www.ubuntuusers.de>*

eine Zusammenstellung von GNU/Linux-Software, komplett von CD oder Flashmedien lauffähig.

- Wenn euch die Kommandozeile noch zu kompliziert vorkommt, könnt ihr ein ausführliches Online-Tutorial machen:

*<https://www.codecademy.com/courses/learn-the-command-line>*