Prof. Dr. Karsten Urban
Dipl. Math. oec. Klaus Stolle
Dipl. Math. Iris Häcker
Institut für Numerische Mathematik, Universität Ulm

ulm university universität ulm

Wissenschaftliches Arbeiten in CSE

Ulm, 15. Oktober 2013

Unix Grundlagen

Was sind TEXund LATEX?

Das erste LATEX-File

Dokument-Klassen und das Einbinden von Paketen

Unix Grundlagen

- UNIX ist ein portables, einfach aufgebautes Betriebssystem (BS)
  - Multitasking-BS (Multiprocessing-BS)
  - Multiuser-BS (Mehrbenutzer-BS)
  - dialogorientiert
- UNIX ist geeignet für Laptops - Großrechner
- Ken Thompson fing 1969 bei Bell Laboratories mit der Entwicklung von UNIX an, seit 1971 vollständig in C

- LINUX ist ein frei verfügbares Multitasking und Multiuser BS das UNIX-ähnlich ist
- Linus Torvalds begann 1991 LINUX zu entwickeln
- LINUX wird derzeit von Softwareentwicklern auf der ganzen Welt weiterentwickelt, es sind sowohl Unternehmen als auch Non-Profit-Organisationen und Einzelpersonen beteiligt, die dies als Hobby betreiben.
- Kommandozeilen-Befehle von UNIX und LINUX stimmen überein

Shell

Eine Shell dient dazu auf einer direkten Ebene auf Details von Teilen des Betriebssystems zuzugreifen. Dies geschieht durch Kommandos, die durch [ENTER] abgeschlossen werden. Einfache Befehle sind

ls	„list“; listet den Inhalt des aktuellen Verzeichnisses auf
mkdir	„make directory“; legt ein neues Verzeichnis an
cd	„change directory“; wechselt in ein Verzeichnis
cp	„copy“; kopiert Dateien / Verzeichnisse
mv	„move“; benennt Dateien/Verzeichnisse um
rm	„remove“; entfernt Dateien
rmdir	„remove directory“; entfernt leere Verzeichnisse
pwd	„print working directory“; Anzeige des aktuelle Verzeichnisses
man	„manual“; Hilfe zu Befehlen

Shell

Vereinfachte Syntax der wichtigsten Befehle (optionale Argumente werden in eckigen Klammern angegeben):

ls	ls [PFAD]
mkdir	mkdir PFAD
cd	cd [PFAD]
cp	cp QUELLE [QUELLE2 QUELLE3 . . .] ZIEL
mv	mv QUELLE [QUELLE2 QUELLE3 . . .] ZIEL
rm	rm PFAD [PFAD2 PFAD3 . . .]
	PFAD bezeichnet nur Dateien
rm	rm -rf PFAD [PFAD2 PFAD3 . . .]
	rekursives Löschen von Dateien und Verzeichnissen ohne Rückfragen. gefährlich!
rmdir	rmdir PFAD
pwd	pwd
man	man PROGRAMMNAME

Abkürzungen für häufig verwendete Pfade:

.	das aktuelle Verzeichnis
..	das eine Ebene höher liegende Verzeichnis
~	das Heimatverzeichnis

TEX

Abbildung: Donald Knuth

Quelle: `http://www-cs-faculty.stanford.edu/~uno`

- TEX ist Programmiersprache für Textverarbeitung (Textsatzsystem)
  - entwickelt '77-'82 von Prof. Donald Knuth, Stanford University
  - Befehlsumfang etwa 300 Befehle
- TEX ist Freeware, aber eingetragenes Warenzeichen
  - Abkürzung für griechisch $\tau\epsilon\chi\nu\eta$ - Fähigkeit, Kunstfertigkeit, Handwerk
  - entweder TEX oder Tex schreiben!
  - Versionsnummer konvergiert gegen π, aktuelle Version 3.1415926 (März 2008)
  - bei Knuths Tod wird Weiterentwicklung gestoppt und Versionsnummer auf π gesetzt
- TEX gilt als fehlerfreie Software
  - jeder gefundene Fehler wird belohnt
- TEX erlaubt eigenes schreiben von Makros
  - Makros $\approx$ Funktion
  - genauer:
    - Makro = Abkürzung für gewisse Befehlsfolge
    - Interpreter ersetzt beim Übersetzen Abkürzung durch vollständigen Code
    - entspricht der inline-Funktion in C

Makro-Pakete für TEX

- '82 veröffentlichte die American Mathematical Society eine Makro-Sammlung `amstex` für TEX
- '85 veröffentlichte Leslie Lamport die Makro-Sammlung LATEX
  - heute de facto Standard in der Mathematik
  - '89 -'03 Entwicklung von LATEX3 (unvollendet, Projekt als abgeschlossen erklärt)
  - aktuelle Version: LATEX2$_\epsilon$(2003)

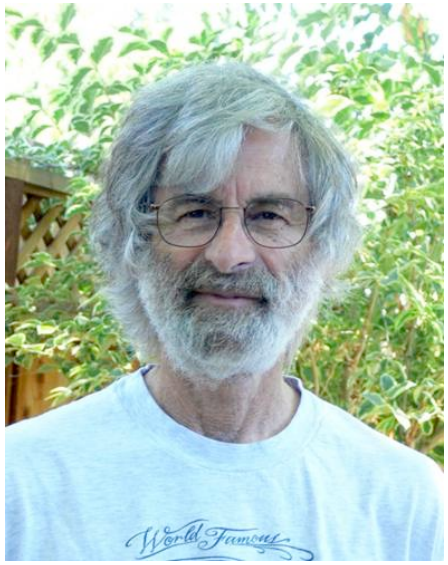

Abbildung: Leslie Lamport
Quelle: `http://www.lamport.org`

Vor- und Nachteile von LATEX

Vorteile von LATEX

- LATEX ist Freeware und für alle gängigen Systeme vorhanden
  - rechner- und betriebssystemunabhängig
  - Output-Dokumente sehen auf jedem System identisch aus
- produziert professionelles Layout
  - Layout-Vorlagen für Artikel/Bücher/Folien
- nur wenige Befehle für die logische Strukturierung eines Schriftstücks notwendig
- mathematische Formeln können gut umgesetzt werden
- Dokumente lassen sich problemlos erweitern, automatische Aktualisierung von
  - Layout
  - Querverweisen
  - Referenzen
  - Inhalts- und Stichwortverzeichnis
- direkte Schnittstelle zu ps/pdf
- WYSWYM = What you see is what you mean

Vor- und Nachteile von LATEX

Nachteile von LATEX

- Einarbeitungszeit
- nicht klickbar
- nicht WYSWYG = What you see is what you get
- eigene Layout-Vorlagen sind vergleichsweise kompliziert zu schreiben

Literatur

Bücher:

- Goossen, M., Mittelbach, F. et a. (2010): *Der LATEX-Begleiter*
  2. Aufl., Addison Wesley, München
- Kopka, H. (2002): *LATEX, Bd. 1: Einführung*
  3. überarb. Aufl., Addison Wesley, München
- Kopka, H. (2002): *LATEX, Bd. 2: Ergänzungen. Mit einer Einführung in METAFONT*
  3. überarb. Aufl., Addison Wesley, München
- Kopka, H. (2002): *LATEX, Bd. 3: Erweiterung: BD3)*
  Korrigierter Nachdruck der 2. Aufl., Addison Wesley, München
- Braune, K., Lammarsch, J. u. M. (2006): *LATEX-Basissystem, Layout, Formelsatz*
  Springer, Berlin Heidelberg

Internetquellen:

- Oetiker, T., Partl, H., Hyna, I. et al. (2003): LATEX2$_\epsilon$-Kurzbeschreibung,
  `http://tobi.oetiker.ch/lshort/lshort.pdf`
- Jürgens, M., Feuerstack, T. (2012): LATEX - eine Einführung und ein bisschen mehr. . . Hrsg. v.d. FernUniversität Hagen,
  `http://www.fernuni-hagen.de/imperia/md/content/zmi_2010/a026_latex_einf.pdf`

Was braucht man, um mit LATEX zu arbeiten?

LATEX-Software

- arbeitet im Hintergrund
- Bestandteile: TEX/LATEX-Programme, Schriften, Skripte ...
- Einfachster Installationsweg: Distribution
  - TeX Live (Unix/Linux/Windows/Mac), MacTeX (Mac OS X), MiKTeX (Windows)

Eingabe-/Steuerungssoftware (Entwicklungsumgebung)

- Texteditor
  - pico(UNIX, frei), Texmaker (Plattformunabhängig, freier LaTeX-Editor), TeXShop (Mac OS X, frei), gedit (GNOME Editor, frei), WinShell (Windows, frei)

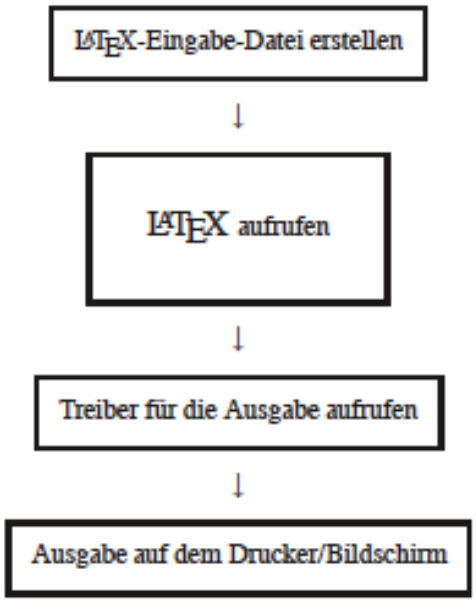

Abbildung: Vorgehensweise beim Arbeiten mit LATEX
Quelle: Jürgens, M., Feuerstack, T.

1. Eingabefile schreiben (Textfile)
2. File mit LATEX bearbeiten
   $\Longrightarrow$ erzeugt Datei, die gesetzten Text in geräteunabhängigem Format (DVI, PDF, PS) enthält
3. Probeausdruck auf Bildschirm anzeigen (Preview)
4. Wenn nötig zurück zu Schritt 2 und Eingabe korrigieren

Wie erstellt man ein LaTeX-File?

1. Möglichkeit:

- Starte Editor pico aus einer Shell mit `pico`
- Schreibe Source-Code
- Abspeichern mit CTRL-O unter dem Dateinamen `name.tex`
  - Endung `.tex` ist Kennung eines TeX/LaTeX-Files
- Compilieren mit `latex name.tex` in einem neuen Tab
  - CMD-T = neuen Tab öffnen
- Falls Code fehlerfrei erhält man
  - `name.dvi`: **D**e**V**ice **I**ndependent file = visualisierbares Output
  - `name.aux`: interne Hilfsdatei (**AUX**iliary file), wichtig für Referenzen! (später!)
  - `name.log`: **LOG**-File = Shell-Output beim Übersetzen des Codes
- **Post-Processing**: Visualisierung mittels DVI-Viewer oder Konvertierung in PS- bzw. PDF-File

Post-Processing

- Visualisierung mittels DVI-Viewer, z.B. `xdvi name.dvi`
- Konvertieren ins Postscript-Format
  - `dvips name.dvi` erzeugt die Datei `name.ps`
  - Öffnen des PS-Files mit `open name.ps`
  - Optionen: `dvips name.dvi -o name2.pdf -Ppdf` erzeugt die Datei `name2.ps`, Option `-Ppdf` um pixel-freies PDF erzeugen zu können
- Konvertieren ins PDF-Format
  - `ps2pdf name.ps` erzeugt die Datei `name.pdf`
  - Öffnen des PDF-Files mit `open name.pdf`
  - `dvi2pdf name.dvi` erzeugt die Datei `name.pdf`, ist aber nicht auf allen Systemen unterstützt

Wie erstellt man ein LaTeX-File?

2. Möglichkeit:

- Starte Editor pico aus einer Shell mit `pico`
- Schreibe Source-Code
- Abspeichern mit CTRL-O unter dem Dateinamen `name.tex`
  - Endung `.tex` ist Kennung eines TeX/LaTeX-Files
- Compilieren mit `pdflatex name.tex` in einem neuen Tab
  - CMD-T = neuen Tab öffnen
- Falls Code fehlerfrei erhält man
  - `name.pdf`: **P**ortable **D**ocument **F**ormat file = visualisiertes Output
  - `name.aux`: interne Hilfsdatei (**AUX**iliary file), wichtig für Referenzen! (später!)
  - `name.log`: **LOG**-File = Shell-Output beim Übersetzen des Codes
- Öffnen des PDF-Files mit `open name.pdf`

Wie erstellt man ein LaTeX-File?

3. Möglichkeit:

- Starte Editor texmaker aus einer Shell mit `open -a texmaker`
- Schreibe den Source-Code in eine neue Datei
- Abspeichern mit CMD-S unter dem Dateinamen `name.tex`
  - Endung `.tex` ist Kennung eines TeX/LaTeX-Files
- Compilieren über Menülleiste z.B. mit PDFLaTex
- Falls Code fehlerfrei erhält man dann
  - `name.pdf`: **P**ortable **D**ocument **F**ormat file = visualisiertes Output
  - `name.aux`: interne Hilfsdatei (**AUX**iliary file), wichtig für Referenzen! (später!)
  - `name.log`: **LOG**-File = Shell-Output beim Übersetzen des Codes
- Ansehen des PDF-Files über Menüleiste

Das erste LaTeX-Programm

Quelldatei (LaTeX)

```
1 %Helloworld.tex
2 \documentclass[a4paper,11pt]{article}
3 \usepackage{fullpage}
4
5 \begin{document}
6 Hello World!
7 \end{document}
```

Ausgabe-Datei (PDF)

Hello World!

- Jedes LaTeX-Programm besitzt die Zeilen 2 ,5, 7
- Übersetzung stets sequentiell von oben nach unten
- Zeilen vor `\begin{document}` bilden den LaTeX-Kopf, -Vorspann oder die Präambel
  - Zeile 2: legt Layout des Dokuments fest
  - Zeile 3: bindet Makro-Pakete ein
  - Definition von eigenen Makros
- `\begin{document} ... \end{document}` beinhaltet eigentliches Dokument
- Zeile 1 ist Kommentarzeile, eingeleitet durch `%`
- LaTeX-Befehle beginnen immer mit \
  - `\documentclass, \usepackage, \begin, \end`
  - Optionale Parameter immer in `[...]`
  - Obligatorische Parameter immer in `{...}`
- Beispiel 1: Erstellen und Übersetzen einer Tex-Datei

Dokument-Klassen

- LaTeX-Befehl:

```
1 \documentclass[options]{documenttyp}
```

- Standard-Dokumenttypen in LaTeX
  - `article` = wiss. Publikation
  - `report` = kurze Bücher, Bachelor-, Masterarbeiten
  - `book` = Bücher
  - `beamer` = Folien, Präsentationen (z.B. vorliegendes Dokument)
  - Beispiel 2: Dokument-Klasse `article`
- **Optionale Parameter für `article`**
  - `10pt, 11pt, 12pt` = Schriftgröße für Standardtext
  - `a4paper` immer wählen! (Papiergröße)
    - Standard ist `letterpaper` = US-Maße
  - `fleqn` = Formeln linksbündig statt zentriert
  - `leqno` = Formeln rechtsbündig statt zentriert
  - `titlepage` = neue Seite nach Titel/Autor etc.
    - Standard ist `notitlepage`
  - `twocolumn` = zweispaltig statt einspaltig
    - Standard ist `onecolumn`
  - `twoside` = zweiseitiges Dokument statt einseitig
    - Standard ist `oneside`
  - `landscape` = Querformat statt Hochformat
  - Beispiel 3: Optionale Parameter

- **Optionale Parameter für `report` und `book`**

  Wie bei `article`, Ausnahmen sind:

  - `notitlepage` = keine neue Seite nach Titelseite
    - Standard ist `titlepage`
  - `twocolumn` = zweispaltig statt einspaltig
    - Standard ist `onecolumn`
  - `oneside` = einseitiges Dokument
    - Standard ist `twoside`
  - `openany` = neue Kapitel beginnen auf neuer Seite
    - Standard ist `openright` = neue Kapitel beginnen stets auf der nächsten rechten Seite

Einbinden von Paketen

- LaTeX-Befehl:

```
\usepackage[options]{packagename}
```

- Bindet das Erweiterungspaket (Makropaket) `packagename` ein
- Übergibt gewisse optionale Parameter `options`
- Pakete:
  - `fullpage` = minimiert Randbereiche
  - `inputenc` = erlaubt direkte Verwendung von Sonderzeichen (Zeichenkodierung)
    - Option `utf8` für deutsche Sonderzeichen (ä, ö, ü, ß)
    - Ohne Option `utf8`: Sonderzeichen werden weggelassen
  - `bable` = Wahl der Sprache des Dokuments
    - Option `ngerman` - Neue deutsche Rechtschreibung
    - beeinflusst automatische Silbentrennung
    - „Kapitel“ statt „Chapter“, etc.

Quelldatei (LaTeX)

```
1 %hellowoerld.tex
2 \documentclass[a4paper,11pt]{article}
3 \usepackage{fullpage}
4 \usepackage[utf8]{inpunenc}
5 \usepackage[ngerman]{babel}
6
7 \begin{document}
8 Hello Wörld!
9 \end{document}
```

Ausgabe-Datei (PDF)

Hello Wörld!

- Beispiel 4: Einbinden von Paketen