

Kapitel 1

Einleitung

1.1 $\text{T}_{\text{E}}\text{X}$	6
1.2 Textverarbeitung versus Satzprogramm	6
1.3 Das Prinzip	9
1.4 $\text{T}_{\text{E}}\text{X}$, $\text{pdfT}_{\text{E}}\text{X}$, $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, $\text{LuaT}_{\text{E}}\text{X}$,	10
1.5 Die Distribution	13

»Flotter Oldie, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ – 25 Jahre und kein Ende«, lautete die Überschrift eines Beitrags in der Computerzeitschrift *c't*, was jetzt schon wieder mehr als 15 Jahre her ist. (H. Voß 2005) Die $\text{T}_{\text{E}}\text{X}$ Users Group (<https://tug.org>) setzte ihre Jahrestagung 2010 unter das Motto » $\text{T}_{\text{E}}\text{X}$'s 2⁵ anniversary« (<https://tug.org/tug2010/>). Man hat es also mit einer ziemlich alten Software zu tun, die man auch getrost als Dinosaurier der Programmentwicklung bezeichnen kann. Anfänglich nur für Mathematiker gedacht, kann heutzutage faktisch jeder das Programm $\text{T}_{\text{E}}\text{X}$ benutzen und qualitativ hochwertige Dokumente erstellen. Dabei wird es im Allgemeinen eine Anwendung von $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ sein, welches von Leslie Lamport entwickelt wurde, um das von Donald Knuth geschaffene $\text{T}_{\text{E}}\text{X}$ anwenderfreundlich zu gestalten. (Knuth 1984; Lamport 1995)

Während die Mathematiker eigentlich weltweit nur $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ als eine adäquate Möglichkeit betrachten, ihre mit Formeln gespickten Texte anspruchsvoll zu setzen, sind im letzten Jahrzehnt fast alle anderen Fachbereiche dazugekommen. Sie alle vereint die Überzeugung, dass Dokumente, egal von welchem Umfang und welcher Komplexität, mit $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ garantiert mehr als nur zufriedenstellend gesetzt werden können.

Die Veränderungen im Druckwesen haben den Beruf des Setzers überflüssig werden lassen. Dafür entstand die digitale Druckvorstufe, weshalb nun ein jeder Autor, Lektor und Verlag in einer Person werden kann. Und Kleinauflagen sind mittlerweile finanziell auch nicht mehr das große Problem. Mit $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ lassen sich qualitativ hochwertige PDF-Dokumente erstellen, welche faktisch von der digitalen Druckvorstufe »durchgewunken« werden können und dann als gedrucktes Buch oder Zeitschrift erscheinen.

1.1 T_EX

T_EX kann für sich in Anspruch nehmen, das letzte Softwarepaket zu sein, welches neben SPSS sowohl aus den Anfängen der Unix- als auch der Linux-Ära übrig geblieben ist. Und nicht nur das, es ist mit seinen nun mehr als 42 Jahren aktuell wie nie zuvor. Dies liegt auch an der Komplexität des Textsatzes an sich und der Umsetzung in ein Programm.

Es wurden mehrere Versuche unternommen, einen würdigen Nachfolger für T_EX zu entwickeln, der die heutigen Möglichkeiten an Soft- und Hardware besser nutzen kann. Um manche Dinge im Programmcode von T_EX zu verstehen, muss man sich vergegenwärtigen, dass ein damals zur Verfügung stehender Speicher von circa 100 kByte eben algorithmische Tricks implizierte, die heute nicht mehr notwendig sind. Mit LuaHB_TE_X steht mittlerweile ein funktionsfähiger Nachfolger bereit, mit dem auch dieses Buch gesetzt wurde.

1.2 Textverarbeitung versus Satzprogramm

T_EX ist keine Textverarbeitung im eigentlichen Sinne. Dies wird schon daran deutlich, dass es völlig unerheblich ist, mit welchem System der Quelltext für das *Programm* T_EX erstellt wird. Der Anwender *programmiert*, d. h. er erstellt ein Programm, welches als Ausgabe einen formatierten Text ausgibt. Es erleichtert Anfängern den Umgang mit dem Gesamtsystem, wenn man sich auch als *Programmierer* versteht. Dies bedeutet, dass es völlig unwichtig ist, welche Formatierung der Quelltext aufweist, solange man die Regeln der dem System zugrunde liegenden Befehlssyntax beachtet. Dies betrifft primär den Zeilenumbruch; ein Satzprogramm umbricht grundsätzlich Absätze, wohingegen fast alle Textverarbeitungsprogramme Zeilen umbrechen. Abbildung 1.1 zeigt den Umbruch einer Textverarbeitung; diese beziehen den Umbruch nur auf eine Zeile, indem diese immer dann beendet wird, wenn nicht mehr genug Platz für ein folgendes Wort vorhanden ist. Dies führt dazu, dass die zweite Zeile mit einem sehr großen Wortabstand gesetzt wird, was optisch immer sehr zweifelhaft erscheint.

Neithardt von Gneisenau,

der Kommandant der Festung Kolberg, deren ruhmreiche Verteidigung einen der interessantesten Abschnitte dieses Buches (von <http://www.gutenberg.org/files/23333/23333-8.txt>) bildet, 1760 geboren, hatte schon in einem zu Erfurt garnisonierenden österreichischen und danach in einem der Regimenter des Markgrafen von Ansbach-Bayreuth gedient, die in englischem Solde in und gegen Amerika kämpften, als Friedrich der Große ihn 1786 als Premierleutnant in die preußische Armee aufnahm.

Abbildung 1.1: Zeilenumbruch einer Textverarbeitung (OpenOffice)

*Ungünstige
Zeilen-
umbrüche*

Bei einer Textverarbeitung erfolgt *nach* einem Zeilenumbruch unabhängig von den zukünftigen Zeilenumbrüchen keine Änderung mehr. Diese Tatsache führt immer dann zu ungünstigen Umbrüchen, wenn die Zeile Bestandteile enthält, die nicht getrennt werden sollen oder können. In dem angegebenen Beispiel ist es eine lange

Internetadresse, die ohne externen Eingriff nicht getrennt wird. Vergleicht man dazu Abbildung 1.2, welche die Ausgabe von T_EX darstellt, so fällt sofort auf, dass sich die Größe der Lücken verringert hat.

Neithardt von Gneisenau,
 der Kommandant der Festung Kolberg, deren ruhmreiche Verteidigung einen der interessantesten Abschnitte dieses Buches (von <http://www.gutenberg.org/files/23333/23333-8.txt>) bildet, 1760 geboren, hatte schon in einem zu Erfurt garnisonierenden österreichischen und danach in einem der Regimenter des Markgrafen von Ansbach-Bayreuth gedient, die in englischem Solde in und gegen Amerika kämpften, als Friedrich der Große ihn 1786 als Premierleutnant in die preußische Armee aufnahm.

Abbildung 1.2: Absatzumbruch eines Satzprogramms (T_EX).

Da bei T_EX *immer* der gesamte Absatz für den Umbruch der einzelnen Zeilen herangezogen wird, kann das Aufeinandertreffen von eng und sehr weit gesetzten Zeilen vermieden werden; jeder Umbruch steht in Bezug zum vorhergehenden und nachfolgenden Umbruch. T_EX kennt vier optische Kriterien für den Satz von Zeilen: *eng*, *weniger eng*, *weit* und *sehr weit*. Nach Definition dürfen aufeinander folgende Zeilen nur benachbarte Kriterien erfüllen, beziehungsweise die gleiche Anordnung aufweisen. Dadurch ist es bei T_EX unmöglich, dass eine sehr weit gesetzte Zeile einer eng gesetzten folgt, was sonst zu den unangenehmen Lücken in Abbildung 1.1 auf der vorherigen Seite führen könnte.

Absatz-
umbruch

Neithardt von Gneisenau,
 der Kommandant der Festung Kolberg, deren ruhmreiche Verteidigung einen der interessantesten Abschnitte dieses Buches (von <http://www.gutenberg.org/files/23333/23333-8.txt>) bildet, 1760 geboren, hatte schon in einem zu Erfurt garnisonierenden österreichischen und danach in einem der Regimenter des Markgrafen von Ansbach-Bayreuth gedient, die in englischem Solde in und gegen Amerika kämpften, als Friedrich der Große ihn 1786 als Premierleutnant in die preußische Armee aufnahm.

Abbildung 1.3: Absatzumbruch mit Linkumbruch (L^AT_EX).

Das optische Erscheinungsbild kann weiter verbessert werden: Zum einen durch eine automatische Umbruchmöglichkeit der Internetadresse (Abbildung 1.3) und zum anderen durch einen zusätzlichen Einsatz des Paketes `microtype` (Abbildung 1.4 auf der nächsten Seite), welches Verfahren der Mikrotypografie anwendet. Informationen dazu gibt es im Abschnitt 5.9.3 auf Seite 171 und und allgemein auf <https://www.typollexikon.de/mikrotypografie/>.

Der Umbruch für den Absatz wird nach dem so genannten Best-Fit-Algorithmus¹ vorgenommen. Vereinfacht formuliert: Von allen möglichen Umbrüchen eines Absatzes wird letztlich derjenige ausgewählt, der dem idealen Umbruch am nächsten kommt. Der ideale Umbruch hat zwischen allen Wörtern denselben Abstand und kommt ohne Trennungen aus. T_EX geht hierbei in maximal drei Stufen vor:

¹ <https://de.wikipedia.org/wiki/Behälterproblem>

Neithardt von Gneisenau,

der Kommandant der Festung Kolberg, deren ruhmreiche Verteidigung einen der interessantesten Abschnitte dieses Buches (von <http://www.gutenberg.org/files/23333/23333-8.txt>) bildet, 1760 geboren, hatte schon in einem zu Erfurt garnisonierenden österreichischen und danach in einem der Regimenter des Markgrafen von Ansbach-Bayreuth gedient, die in englischem Solde in und gegen Amerika kämpften, als Friedrich der Große ihn 1786 als Premierleutnant in die preußische Armee aufnahm.

Abbildung 1.4: Absatzumbruch mit Linkumbruch und Anwendung der Mikrotypografie (L^AT_EX).

1. Versuch, einen optimalen Umbruch ohne Trennungen zu erhalten.
 - Genügt das Ergebnis den Vorgaben, wird der Umbruch akzeptiert und als abgeschlossen betrachtet.
 - In der T_EX-Terminologie bedeutet dies, dass die Dehnungspunkte (`\badness`) jeder umbrochenen Zeile den Wert von `\pretolerance=100` nicht überschreiten.
 - Wegen nicht vorhandener Trennstellen benötigt der erste Durchgang sehr wenig Rechenzeit!
2. Versuch, einen optimalen Umbruch mit Trennungen zu erhalten.
 - Es kommen alle sprachenspezifische Trennstellen in Frage, die T_EX durch seinen Trennalgorithmus mit den zugrundeliegenden Trennmustern findet.
 - Der Umbruch wird akzeptiert, falls alternativ
 - die Dehnungspunkte (`\badness`) jeder umbrochenen Zeile den Wert von `\tolerance=200` nicht überschreiten und `\emergencystretch` einen positiven Wert aufweist, oder
 - falls `\tolerance` den Wert 10000 hat; dann kommen zusätzlich die Werte `\hfuzz=200.0pt` und `\hbadness=1073741823` zum Tragen.
3. und letzter Versuch, durch zusätzliche Wortzwischenräume, die vorher nicht erlaubt waren.
 - Der vorgegebene Wert von `\emergencystretch=8.69995pt`, der auch vom Anwender in seinem Quelltext geändert werden kann, wird zur erlaubten Dehnungsmöglichkeit jeder Zeile addiert (siehe auch Abschnitt 5.9.4 auf Seite 173).
 - Durch diese Addition werden die zulässigen Dehnungspunkte einer jeden Zeile verringert.
 - Die abschließende Entscheidung wird mithilfe der Register `\hfuzz=200.0pt` und `\hbadness=1073741823` getroffen.

Hierbei wird, wie oben erwähnt, die Berechnung sofort nach der ersten oder zweiten Stufe beendet, falls der Umbruch den internen Vorgaben entspricht. Auf den sehr aufwändigen Algorithmus der einzelnen Stufen soll hier nicht weiter eingegangen werden; es wird auf die ausführliche Beschreibung in H. Voß (2007) verwiesen. Hier sollte nur gezeigt werden, welchen Aufwand T_EX im Gegensatz zu herkömmlichen Textverarbeitungsprogrammen treibt, um einen optimierten Umbruch zu erreichen.

Die Dissertation »Optimal Pagination Techniques for Automatic Typesetting« von Michael F. Plass (<http://tug.tug.org/docs/plass/plass-thesis.pdf>) beschreibt das

Verfahren im Detail. Auf <https://github.com/bramstein/typeset> findet sich eine JavaScript-Implementation des sogenannten »Knuth-Plass-Algorithmus«, die man für eigene Experimente nutzen kann.

1.3 Das Prinzip

Es wurde bereits darauf hingewiesen, dass man sich eher als *Programmierer* denn als *Texteingabe* bei der Erstellung seines Dokumentes fühlen sollte. Dies erleichtert das Verständnis für das Prinzip bei der Texteingabe. Bei einer optimalen Situation kann man bei Auswahl einer bestimmten Dokumentenklasse erreichen, dass man lediglich die Sprachauswahl und Schriftkodierung vorgeben muss. Sämtliche anderen Vorgaben, die das Layout und die Struktur des Dokumentes betreffen, sind bereits in der gewählten Dokumentenklasse vorgegeben, sodass der so genannte Übersetzungsvorgang ein fertig formatiertes Dokument höchster Qualität liefert.

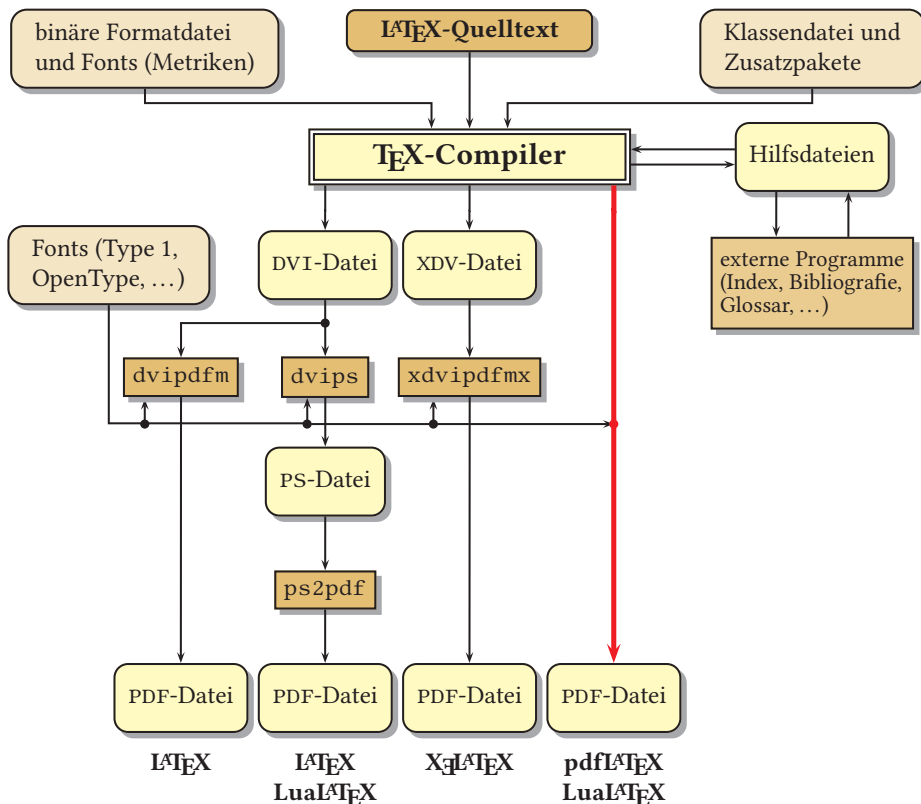


Abbildung 1.5: »Von der Quelle bis zur Mündung ...«.

Abbildung 1.5 zeigt den kompletten Ablauf eines Übersetzungsvorganges zur Erzeugung eines LaTeX-Dokumentes, wie es der augenblickliche Standard ist. Der Quelltext muss wie jeder andere Quelltext eines beliebigen Programms, beispielsweise C++ oder Perl, syntaktisch korrekt sein, wenn eine fehlerfreie Ausgabe erfolgen soll. Je nach

Komplexität des Dokumentes kann es notwendig sein, mehrere Durchläufe sowohl des \TeX -Compilers als auch anderer externer Programme vorzunehmen. Zum Beispiel das Inhaltsverzeichnis: Beim ersten Durchlauf erzeugt der \TeX -Compiler aus allen Überschriften das Inhaltsverzeichnis (die Informationen werden in einer separaten Hilfsdatei gespeichert – Dateiendung `.aux`). Da das Inhaltsverzeichnis normalerweise am Anfang des Dokuments steht, liegt es beim ersten Start noch nicht vor. Erst beim zweiten Durchlauf wird das Inhaltsverzeichnis eingebunden. Dies kann aber dazu führen, dass alle nachfolgenden Seiten nach hinten verschoben werden, wenn das Inhaltsverzeichnis mehrere Seiten umfasst und so eventuelle Verweise auf Seitenzahlen nicht mehr stimmen. Somit muss ein dritter Durchlauf gestartet werden, damit alles komplett ist. Eine normale Textverarbeitung wie LibreOffice macht diesen Vorgang im Wesentlichen auch, nur dass man von diesen mehreren Durchläufen nichts mitbekommt und vor allem keinen Einfluss darauf hat.

Zu den externen Dokumentteilen, die durch eigene Programme erzeugt werden, zählen insbesondere eine Bibliografie, ein Index oder ein Glossar. Die Anforderungen an den Anwender sind hierbei gering, selbst für den Fall, dass er nicht mit einem Makefile oder einer grafischen Entwicklungsumgebung (GUI) für \LaTeX arbeitet. Die gesamte Schrifteinbindung, die in der Abbildung 1.5 auf der vorherigen Seite dargestellt ist, erfolgt vollautomatisch und muss den Anwender nur in seltenen Ausnahmefällen interessieren.

Für einfache Texte gelten nachfolgende Schritte. Verwendet man eine grafische Oberfläche für die Erstellung (siehe Kapitel 2 auf Seite 29), so kann diese einige Schritte automatisch übernehmen und so vereinfachen.

1. Texte im Editor schreiben, beispielsweise mit \TeX Studio: `textstudio meinText.tex`
2. Dokument übersetzen, beispielsweise mit \LuaTeX : `lualatex meinText.tex`
Es werden dabei mindestens folgende Dateien erzeugt:
meinText.log Enthält alle Statusmeldungen des Übersetzungsvorgangs.
meinText.aux Enthält unter anderem die Einträge für Querverweise.
meinText.pdf Das erzeugte PDF-Dokument (nach dem ersten Durchlauf ohne Inhaltsverzeichnis).
3. Erneutes Übersetzen des Dokuments: `lualatex meinText.tex`
4. PDF-Dokument ansehen, beispielsweise für Linux mit dem PDF-Viewer *Evince*.
Ist dieser noch nicht gestartet, dann: `evince meinText.pdf`

1.4 \TeX , pdf \TeX , $\text{\X}\text{\TeX}$, \LuaTeX , ...

Der Anfänger wird mit vielen verschiedenen Begriffen überhäuft, deren genaue Bedeutung auch geübten Anwendern nicht immer klar ist. Das ursprüngliche \TeX existiert im Prinzip nur noch als Kern sämtlicher Nachfolger, von denen pdf(e) \TeX zur Zeit die eigentliche »Maschine« ist. Das »e« steht hier für *extended*, was sich in erweiterten Fähigkeiten des Kernprogramms \TeX äußert. Da Knuth verfügt hat, dass das Original- \TeX beliebig verändert werden darf, jedoch dann einen neuen Namen bekommen muss, wird man bei den aktuellen Nachfolgern oft etwas andere Namen finden, beispielsweise $\text{\LuaHB}\text{\TeX}$.

1.4.1 Ein wenig Hintergrundwissen

Das von Donald Knuth geschriebene Programm T_EX stellt nur die Basisfunktionalität zur Verfügung, ist aber ansonsten genauso nutzlos wie ein C-Compiler ohne Bibliotheken. In beiden Fällen ist die Benutzung extrem umständlich. Knuth hat daher ein Makropaket (`plain.tex`) mitgeliefert, welches den Umgang mit den Basisfunktionen erheblich erleichtert. Genauso wie die Standard-Bibliotheken eines Compilers, die erst ein effizientes Programmieren ermöglichen.

Eine andere Erweiterung ist L^AT_EX von Leslie Lamport. (Lamport 1995) Nun könnte man sich vorstellen, dass man dem Anwender sagt, er solle einfach `\input plain` oder `\input latex.ltx` als erste Zeile in sein Dokument schreiben. Abgesehen davon, dass es nicht sonderlich effizient ist, wird es in der Praxis anders gemacht: Es werden dem Anwender separate Programme zur Verfügung gestellt, nämlich LuaHBT_EX und L^AT_EX, die entweder die `plain`-Makros oder die L^AT_EX-Makros laden. Dabei sind die L^AT_EX-Varianten unter Unix/Linux Links auf das eigentliche T_EX-Programm und unter Windows mehr oder weniger eine Kopie des T_EX-Programms, welches lediglich das L^AT_EX-Modul lädt. Dadurch wird häufig L^AT_EX-Anwendern der Eindruck vermitteln, dass L^AT_EX ein eigenständiges Programm ist. Ein reiner L^AT_EX-Anwender muss aber prinzipiell nicht wissen, wie diese Dinge intern implementiert worden sind.

T_EX wurde später von Hàn Th_o Thành erweitert, um auch direkt eine PDF-Ausgabe erzeugen zu können. Damit kam ein neues Programm, pdfT_EX ins Spiel, später kamen X_YT_EX von Jonathan Kew und LuaT_EX von Taco Hoekwater, beide auf Unicode basierende Erweiterungen von T_EX, hinzu. Alle erzeugen standardmäßig eine PDF-Ausgabe. LuaT_EX hat die Skriptsprache Lua voll integriert, die im normalen Text benutzt werden kann. Es existieren also prinzipiell folgende Programme: T_EX, L^AT_EX, pdfT_EX, pdfL^AT_EX, X_YT_EX, X_YL^AT_EX, LuaT_EX, und LuaHBT_EX und LuaL^AT_EX. Für L^AT_EX-Nutzer sind nur die von Interesse, die ein »La« in ihrem Namen haben. Bevorzugt werden sollte LuaL^AT_EX. (Wright 2024a)

Ähnlich verhält es sich mit dem Ausgabeformat. Wenn der Name des Programms »pdf« enthält, erhält man standardmäßig das PDF-, ansonsten das DVI-, beziehungsweise für X_YT_EX das XDVI-Format. Da die neueren Programme grundsätzlich eine PDF-Ausgabe erstellen, findet man auch kein »pdf« mehr in ihrem Namen. Für X_YT_EX erfolgt die Konvertierung von XDVI nach PDF automatisch im Hintergrund. Wer eine DVI/XDVI-Ausgabe erzeugen will oder muss, kann es aber explizit auf der Kommandozeile angeben oder den Editor entsprechend konfigurieren.

Neben `plain TEX` und L^AT_EX gibt es noch ConT_EXt, welches als das leistungsfähigste System bezeichnet werden kann, aber auch auf dem Kern von T_EX bzw. LMTX aufsetzt. T_EX ist eine relativ leistungsfähige, aber auch sehr gewöhnungsbedürftige Skriptsprache. Viele Probleme lassen sich elegant lösen, bei anderem muss man einen ziemlichen Aufwand treiben. Bezüglich der Gleitpunktarithmetik ist T_EX extrem schwach. LuaHBT_EX kombiniert T_EX mit einer sehr leistungsfähigen aber leicht zu erlernenden Skriptsprache Lua (<https://www.lua.org/>). Daneben gibt es noch mit LuaMETAT_EX eine Neuentwicklung, die bislang nur von ConT_EXt genutzt wird (<http://www.pragma-ade.com/luametateX-1.htm>), für die es aber eine im Entwicklungsstadium befindliche L^AT_EX-Anpassung gibt: <https://github.com/zauguin/luametalatex>.



1.4.2 Aktuelle Programme

Die jeweiligen Programmversionen kann man sich auf der Kommandozeile ausgeben lassen:

T_EX Das Original...

```
voss@shania:~> tex -v
TeX 3.141592653 (TeX Live 2024)
kpathsea version 6.4.0
Copyright 2024 D.E. Knuth.
There is NO warranty. Redistribution of this software is
covered by the terms of both the TeX copyright and
the Lesser GNU General Public License.
For more information about these matters, see the file
named COPYING and the TeX source.
Primary author of TeX: D.E. Knuth.
```

pdfT_EX Die zur Zeit am häufigsten verwendete Variante, kann Schriften im Format METAFONT und Type-1 handhaben.

```
voss@shania:~$ pdftex -v
pdfTeX 3.141592653-2.6-1.40.26 (TeX Live 2024)
kpathsea version 6.4.0
Copyright 2024 Han The Thanh (pdfTeX) et al.
There is NO warranty. Redistribution of this software is
covered by the terms of both the pdfTeX copyright and
the Lesser GNU General Public License.
For more information about these matters, see the file
named COPYING and the pdfTeX source.
Primary author of pdfTeX: Han The Thanh (pdfTeX) et al.
Compiled with libpng 1.6.43; using libpng 1.6.43
Compiled with zlib 1.3.1; using zlib 1.3.1
Compiled with xpdf version 4.04
```

X_ET_EX Die Unicode Version, kann Schriften im Format OpenType, TrueType und Type-1 handhaben.

```
voss@shania:~> xetex -v
XeTeX 3.141592653-2.6-0.999996 (TeX Live 2024)
kpathsea version 6.4.0
Copyright 2024 SIL International, Jonathan Kew and Khaled Hosny.
There is NO warranty. Redistribution of this software is
covered by the terms of both the XeTeX copyright and
the Lesser GNU General Public License.
For more information about these matters, see the file
named COPYING and the XeTeX source.
Primary author of XeTeX: Jonathan Kew.
Compiled with ICU version 74.2; using 74.2
Compiled with zlib version 1.3.1; using 1.3.1
Compiled with FreeType2 version 2.13.2; using 2.13.2
Compiled with Graphite2 version 1.3.14; using 1.3.14
Compiled with HarfBuzz version 8.3.0; using 8.3.0
Compiled with libpng version 1.6.43; using 1.6.43
```

```
Compiled with pplib version v2.2
Using Mac OS X Core Text and Cocoa frameworks
```

LuaTeX Faktisch der Nachfolger von pdfTeX, unterstützt die Skriptsprache Lua (<https://www.lua.org/>) und kann Schriften in allen gängigen Formaten handhaben.

```
voss@shakira:~> luatex -v
This is LuaTeX, Version 1.18.0 (TeX Live 2024)
Development id: 7611

Execute 'luatex --credits' for credits and version details.

There is NO warranty. Redistribution of this software is covered by
the terms of the GNU General Public License, version 2 or (at your option)
any later version. For more information about these matters, see the file
named COPYING and the LuaTeX source.
```

LuaHBTeX Nachfolger von pdfTeX mit Unterstützung für HarfBuzz.

```
voss@shakira:~> luahtex -v
This is LuaHBTeX, Version 1.18.0 (TeX Live 2024)
Development id: 7611

Execute 'luahtex --credits' for credits and version details.

There is NO warranty. Redistribution of this software is covered by
the terms of the GNU General Public License, version 2 or (at your option)
any later version. For more information about these matters, see the file
named COPYING and the LuaTeX source.

LuaTeX is Copyright 2022 Taco Hoekwater and the LuaTeX Team.
```

In der obigen Zusammenstellung erscheint kein L^AT_EX, denn dieses ist, wie bereits erwähnt, lediglich eine Ansammlung von Befehlen zur einfacheren Handhabung von T_EX. Mit anderen Worten: L^AT_EX ist ein spezielles Format, welches beim Aufruf von L^AT_EX oder pdfL^AT_EX automatisch von dem eigentlichen Programm pdfTeX oder LuaHBTeX geladen wird. Seit 2006 ist zumindest für T_EXLive die eigentliche Maschine pdfTeX, sodass auch ein Aufruf von L^AT_EX das Programm pdfTeX startet, lediglich mit der Anweisung das L^AT_EX-Format zu laden und eine DVI-Ausgabe zu erstellen. Standardmäßig wäre das Ausgabeformat PDF.

1.5 Die Distribution

Zu einem lauffähigen T_EX-System gehören mehrere tausend Dateien, was man als Distribution bezeichnet; eine Vollinstallation von T_EXLive 2021 umfasst ungefähr 75.000 Dateien. Diese sind zum einen abhängig vom verwendeten Betriebssystem und zum anderen ganz profan abhängig von Personen, die sich die Arbeit machen und eine solche Distribution erstellen. Es gibt momentan nur zwei Distributionen, die nahezu alle bedeutenden Betriebssysteme wie Linux, macOS, Windows, Solaris, usw. unterstützen. Dies sind T_EXLive (<https://www.tug.org/texlive>), welche gemeinsam

von den internationalen Benutzergruppen herausgegeben wird und MiKTeX (<https://www.miktex.org/>), welche von Christian Schenk herausgegeben und hauptsächlich für Windows genutzt wird. Für macOS wird wegen der speziellen Voraussetzungen die Installation von MacTeX (<https://www.tug.org/mactex/>) empfohlen, welches selbst das komplette T_EXLive enthält. Die anderen Varianten wie fpT_EX oder XeM_TE_X gelten mittlerweile als völlig veraltet und sind bedeutungslos.

1.5.1 T_EXLive-Installation



Zur Zeit der Drucklegung dieses Buches ist die aktuelle Version TL2024. Die Installation von T_EXLive kann von verschiedenen Quellen erfolgen.

- Mit einem ISO-Image von der offiziellen Webseite von T_EXLive (<https://www.tug.org/texlive/acquire-iso.html>),
- oder einem Komplettabzug eines Spiegelservers (≈2.4GiB) (<https://www.tug.org/texlive/mirmon/>),
- oder mit dem »Netinstaller« (≈6.6MiB), was die zu empfehlende Variante ist (<https://tug.org/texlive/acquire-netinstall.html>),
- oder für Linux mithilfe des integrierten Paketmanagers, was aber in diesem Fall nicht zwingend die neueste Version sein muss,
- oder für Ubuntu mit dem Skript von <https://github.com/scottkosty/install-tl-ubuntu>, womit man eine aktuelle Version von T_EXLive bekommt, im Gegensatz zum Ubuntu-Paket.

Vor dem Beginn der Installation sollte man sich vergewissern, ob nicht eine bereits vorhandene T_EX-Installation existiert, die schon Umgebungsvariablen entsprechend gesetzt hat. Insbesondere unter Windows kann es hier zu Problemen kommen. Sicherheitshalber sollte man eventuell vorher eine existierende T_EX-Installation entfernen.

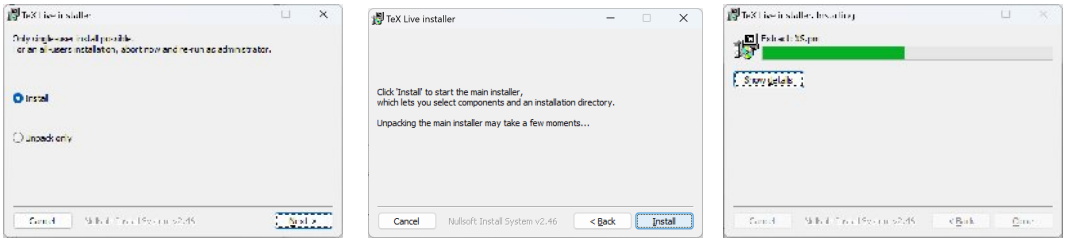
Bei einem Ein-Personen-System (single user system) besteht für die eigentliche Installation keine Notwendigkeit diese als Systemadministrator (Windows) oder root (Linux/Unix) vorzunehmen. Man sollte sich jedoch für eine Möglichkeit entscheiden und diese Variante dann auch bei Updates berücksichtigen.

1.5.2 Installation unter Windows

Bei einer permanenten Netzanbindung empfiehlt sich die Installation über den so genannten »Netinstaller« (Abbildung 1.6 auf der nächsten Seite).

Startet man den »Netinstaller« `install-tl-windows.exe` von tug.org/texlive/acquire-netinstall.html, wobei dieser in der Regel automatisch entpackt wird, so erscheint nach dem Start das in Abbildung 1.6c auf der nächsten Seite gezeigte Startfenster. Unter neueren Windowsversionen ist eventuelle vorher ausdrücklich zu bestätigen, dass es sich um eine korrekte aus dem Netz geladene Datei handelt.

Im einfachsten Fall kann man auf »Install« klicken; die Einstellungen sind im Wesentlichen an europäische Verhältnisse angepasst. Ohne weitere Änderungen werden dann ca. 2.4 GigaByte an Daten auf den Rechner übertragen. Möchte man beispielsweise den gesamten CJK-Bereich (Chinesisch, Japanisch und Koreanisch) nicht installiert haben,



(a) Auswahl der Aktion für heruntergeladene Datei `install-tl-windows.exe`

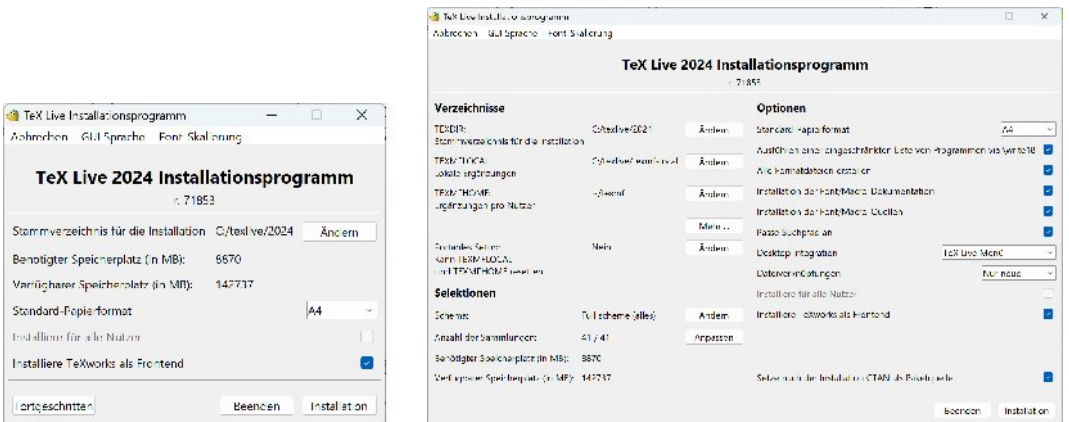
(b) Start zum Entpacken der Installationsdatei.

(c) Entpacken und Installieren der Installationsdateien.

Abbildung 1.6: Herunterladen, Entpacken und anschließendes automatisches Starten der Installationsdatei.

so kann man dies über den Button »Fortgeschritten« einstellen (Abbildung 1.6c). Es erscheint dann das in Abbildung 1.7b dargestellte Fenster. Die Auswahl der Sprachen und der sogenannten »Collections« erreicht man über den Button »Customize«. Kann man eine Internetverbindung nur über einen Proxy-Server erreichen, so sind vorher die beiden Umgebungsvariablen `http_proxy` und `ftp_proxy` zu setzen.

```
set http_proxy=proxy.server:8080
set ftp_proxy=proxy.server:8080
```



(a) Installationsmenü für TeXLive.

(b) Der sogenannte »Expertenmodus« für die Installation.

Abbildung 1.7: Mögliche Einstellungen im »Expertenmodus« und Start der Installation.

Da inzwischen alle Systeme eine ausreichend große Festplattenkapazität besitzen, sollte man die Auswahl aus Abbildung 1.7a übernehmen und alles installieren. Ist dies nicht möglich, so deaktiviert man die entsprechenden Bereiche, die nicht installiert werden sollen.

Bei einer vollständigen Installation mit »scheme-full« werden automatisch alle vorhandenen Pakete ausgewählt. Möchte man einzelne Gruppen oder auch nur Pakete abwählen, so ist dies im Menüpunkt »Makropakete« möglich (Abbildung 1.7a). Für »scheme-full« sind immer alle Sprachen ausgewählt. Das Deaktivieren einzelner Sprachen bringt keine wesentliche Ersparnis bei der Speicherkapazität und ist daher nicht

zu empfehlen.

Mit der Einstellung »TEXDIR« wird das Hauptverzeichnis ausgewählt. Standardmäßig wird hier `C:\texlive\<Jahr>` vorgeschlagen. Alle anderen Verzeichnisse, wie `TEXMFLOCAL` passen sich dann automatisch an. Weitere Einstellungen kann man für eine Desktop-Installation so lassen. Wie man `TEXLive` für den USB-Stick als portable Version installiert, wird in Abschnitt [A.9](#) erklärt. Nach Abschluss der Einstellungen wählt man »Installieren« und die Kopieraktion beginnt, wobei die Installation protokolliert wird (Abbildung [1.11b](#)). Je nach Internetverbindung und Rechnerleistung kann dies unter Umständen sehr lange dauern.

Eine ausführliche Anleitung für die Installation der Distribution unter Windows beschreibt [Zimmer \(2021\)](#).

Pfade unter Windows

Der Windows-Installer nimmt bereits einige wichtige Konfigurationen vor:

- Hinzufügen des Verzeichnisses für die Binärdateien zum aktuellen Suchpfad für Programme. Sollte dabei ein anderer Pfad einer existierenden `TEX`-Installation entdeckt werden, so wird dieser gelöscht. Zur Kontrolle kann man das Logfile zu Rate ziehen: `C:\texlive\2024\install-tl.log`.
- Hinzufügen eines Menüs für `TEXLive` bei der Programmauswahl.
- Hinzufügen einiger Desktop-Shortcuts, die jedoch vom Anwender geändert werden können.

Update

Der Updatemanager kann nach der Installation über das normale Startmenü geladen werden, welcher dieselbe grafische Oberfläche aufweist, wie der Manager unter Linux (Abbildung [1.12 auf Seite 21](#)). Im Ablauf des Updates vom System und den Paketen ist ebenfalls kein Unterschied, sodass hier auf [Abschnitt 1.5.3 auf Seite 20](#) verwiesen wird.

Ein globales Update aller Pakete kann auch bei Windows über die Kommandoebene erreicht werden. Im Befehlsfenster erfolgt dies durch Eingabe von `cmd` (Eingabeaufforderung) und anschließender Auswahl. [Abbildung 1.8 auf der nächsten Seite](#) zeigt die Eingabe des Befehls

```
tlmgr --self --all update
```

und die Ausgabe für einen typischen Updateverlauf.

Hierbei steht `--self` für ein Update des `TEXLive`-Managers selbst und `--all` für ein Update aller Pakete, so überhaupt Updates vorhanden sind. Als Repository wird automatisch ein zuletzt benutztes oder zufällig ausgewähltes benutzt.