

Die T_EXnische Komödie

DANTE
Deutschsprachige
Anwendervereinigung T_EX e.V.

13. Jahrgang Heft 3/2001 September 2001

3/2001

Impressum

„Die T_EXnische Komödie“ ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Schreibenden wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nur im nicht-kommerziellen Rahmen gestattet. Verwendungen in größerem Umfang bitte zur Information bei DANTE e.V. melden.

Beiträge sollten in Standard-L^AT_EX-Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und an untenstehende Anschrift geschickt werden (entweder per E-Mail oder auf Diskette). Sind spezielle Makros, L^AT_EX-Pakete oder Schriften dafür nötig, so müssen auch diese mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden.

Diese Ausgabe wurde mit Hilfe folgender Programme erstellt: e-TeX, Version 3.14159-2.1 (Web2c 7.3.2x), L^ATeX2_ε (2000/06/01), xdvik 22.05d-k und windvi 0.67 für die Bildschirmdarstellung und dvips(k) 5.86d für Korrektur und Belichtung. Die Schriften zur Belichtung wurden mit dem METAFONT-Modus `linoone` (1270 dpi) berechnet.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2700

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

E-Mail: dante@dante.de
dtk-redaktion@dante.de (Redaktion)

Druck: Konrad Triltsch Print und digitale Medien GmbH
Johannes-Gutenberg-Straße 1–3, 97199 Ochsenfurt-Hohe Stadt

Redaktion: Gerd Neugebauer (verantwortlicher Redakteur)

Luzia Dietsche	Rolf Niepraschk	Volker RW Schaa
Rudolf Herrmann	Günter Partosch	
Uwe Münch	Bernd Raichle	

Redaktionsschluss für Heft 4/2001: 6. Oktober 2001

ISSN 1434-5897

Editorial

Liebe Leserinnen und Leser,

dieses Mal muss ich mich mit meinen normalerweise hier zu lesenden Klagen leider – oder besser glücklicherweise – zurückhalten. Einigermassen rechtzeitig haben wir eine bunte Palette von interessanten Beiträgen für „Die T_EXnische Komödie“ zusammen bekommen.

Der Schwerpunkt in dieser Ausgabe liegt bei den Nutzeroberflächen. Da ist auf der einen Seite Emacs, der inzwischen für sehr viele Betriebssysteme verfügbar ist. In vieler Hinsicht hat Emacs einiges mit T_EX gemeinsam. Zu Emacs gibt es zwei Beiträge, die sich damit beschäftigen, wie man Emacs mit MiK_TE_X zusammenbringt und Emacs zur Nutzung von diversen Schriftsystemen und der Zusammenarbeit mit (A)T_EX überreden kann.

Auf der anderen Seite der Nutzungsoberflächen ist LyX angesiedelt. LyX bietet eine grafische Oberfläche, die die Interna von L^AT_EX vor Nutzern verbirgt. In dieser Ausgabe startet die Vorstellung von LyX, die in der nächsten Ausgabe fortgesetzt werden wird.

Passmarken, Schnittmarken, Farbbalken, Farbseparation und weitere Stichworte dieser Art kennzeichnen das Ende der Produktionskette, wenn das Ergebnis unserer Arbeit schließlich zur Druckerei geht. Auch hierfür gibt es Unterstützung in Form eines Paketes, das in einem Beitrag beschrieben wird.

xindy ist ein Index-Prozessor, der als Nachfolger von makeindex angetreten ist. Viele Dinge, die man in makeindex nur mit vielen Tricks erreichen kann, lassen sich mit xindy einfacher bewerkstelligen. Ein Beitrag in diesem Heft beschreibt die Installation, die doch nicht ganz einfach ist.

Es scheint also doch so, dass die Themen rund um (L^A)T_EX noch nicht ausgehen. Ich wünsche allen Lesern, dass das auch in Zukunft so weitergehen wird.

Mit T_EXnischen Grüßen

Ihr Gerd Neugebauer

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

wie Sie sicher erfreut bemerkt haben, liegt dieser Ausgabe von „Die T_EXnische Komödie“ die neueste Version der T_EXLive-CD-ROM bei. Das Bemühen um ein möglichst fehlerfreies Produkt hat dieses Jahr den Erscheinungstermin von Mai auf August verschoben. Die Hauptbeteiligten des T_EXLive-Projekts, Fabrice Popineau und Sebastian Rahtz, benötigten noch zusätzliche Testzeit. Nun liegt Ihnen mit der T_EXLive6b ein System vor, das die Installation und Konfiguration eines T_EX-Systems noch weiter vereinfacht hat. Unix- und PC-Benutzer müssen nun schon etwas tiefer im Gedächtnis kramen, um die Erinnerungen an die aufwändigen Arbeiten und den Frust nach einer misslungenen Installation eines „von Hand“ aufgesetzten Systemes wiederzufinden. Die Benutzer anderer Betriebssysteme mögen diese Euphorie bitte entschuldigen – dem Vorstand ist bewusst, dass nicht alle Systeme einen derart guten Service erfahren, und leider steht es auch nicht in unserer Macht, die Entwicklung für andere Betriebssysteme entscheidend zu beeinflussen, da wir alle auf freiwillige Helfer angewiesen sind. Wir möchten hiermit allen Beteiligten danken, die sich die Mühe gemacht haben, dieses System zusammenzustellen, zu testen und zu dokumentieren, und die Gestaltung des CD-Covers und die Produktion des CD-ROMs in die Hand zu nehmen.

Alle diese Bemühungen sind nur ein Aspekt der Weiterentwicklung von T_EX in den letzten Jahren. Viele neue Pakete, Fonts, sowie Erweiterungen und Neuerungen bereits bekannter Software können Sie auf der CD-ROM finden. Da auf der CD-ROM nur sehr begrenzter Platz ist, lohnt sich auch die Suche auf dem CTAN-Server von DANTE e.V.

Hier soll nur erwähnt werden, dass Hàn Thế Thành seine Doktorarbeit abgeschlossen hat, deren Ergebnis nun als pdfT_EX in der Version 1.0 zur Verfügung steht. Im Rahmen einer kleinen Feier während der TUG 2001-Konferenz in Newark (Delaware, USA) durfte ich, als Vizepräsident von DANTE e.V., in

einer kleinen Rede Hàn Thế Thành für seine Verdienste um T_EX danken und ihm eine Graphik von Duane Bibby überreichen, die Hermann Zapf mit der folgenden kalligraphischen Würdigung versehen hatte:

“The T_EX community thanks Hàn Thế Thành for his innovative extensions to T_EX The Program. His research and development, resulting in pdfT_EX, take T_EX users to new heights in typography and document production.”

Desweiteren freuen wir uns darüber, dass in diesen Tagen die Version 1.0 von $\mathcal{N}\mathcal{T}\mathcal{S}$ angekündigt werden wird.

Im Rahmen der EuroT_EX in Kerkrade wird die Diskussion über die weitere Entwicklung all der Komponenten wie $\mathcal{N}\mathcal{T}\mathcal{S}$, Omega und T_EXlib ein wesentlicher Bestandteil der Tagung sein.

Die „andere“ CD-ROM, der Abzug des CTAN, wird zur Zeit erstellt und soll der vierten Ausgabe von „Die T_EXnische Komödie“ in diesem Jahr beiliegen. Zudem planen wir für dieses Jahr wieder eine Ausgabe von „Die T_EXnischen Kontakte“ zu erstellen, mit den aktuellen Mitgliederlisten, Adressen, der Satzung und weiteren Informationen für Mitglieder.

Wir freuen uns sehr, bereits jetzt den Tagungsort für die DANTE 2002 mitteilen zu können. Wir werden beim Regionalen Rechenzentrum der Universität Erlangen-Nürnberg zu Gast sein. Die Tagung wird vom Mittwoch, den 20. Februar 2002, bis Freitag, den 22. Februar 2002, stattfinden.¹ Die 26. Mitgliederversammlung von DANTE e.V. findet direkt in Anschluss an die Tagung, am Samstag, den 23. Februar 2002 in Erlangen statt. Wir danken Hans Cramer und dem Rechenzentrum in Erlangen für die angebotene Gastfreundschaft.

Wir würden uns freuen, möglichst viele Mitglieder sowohl auf der EuroT_EX-Tagung als auch auf der Mitgliederversammlung begrüßen zu können.

Mit freundlichen Grüßen,

Thomas Koch Volker RW Schaa
(Präsident) (Vizepräsident)

¹ Siehe auch Seite 66

Bretter, die die Welt bedeuten

Emacs, AucTeX und MiKTeXs Yap unter Windows ¹

Rainer Thiel

Wer die Vorteile eines Satzsystems wie (L^A)TeX nutzen will, muss auf die direkte Begutachtung seiner Arbeit auf dem Bildschirm verzichten. Möglich ist dagegen ein bequemer Editor-Compiler-Zyklus, der es erlaubt, vom DVI-Viewer an die entsprechende Stelle im Quelltext zu springen und umgekehrt. Der vorliegende Beitrag beschreibt aus der Sicht eines mit den Interna von (L^A)TeX kaum vertrauten Anwenders, wie dies unter Windows mit GNU-Emacs und MiKTeX zu realisieren ist.

Einleitung

Der Anwender, der mit modernen, für graphische Benutzeroberflächen entwickelten Textverarbeitungsprogrammen vertraut ist, ist daran gewöhnt, die Ergebnisse seiner Eingriffe, jedenfalls ungefähr, unmittelbar am Bildschirm verfolgen zu können (*What You See Is – more or less – What You Get*). Dies ist bei der Verwendung von (L^A)TeX konzeptionell bedingt unmöglich. Möglich ist dagegen der Aufbau eines komfortablen Zyklus, der es erlaubt, den Text in einem Editor zu bearbeiten und aus dem Editor heraus zu kompilieren, die resultierende DVI-Datei in einem Viewer zu betrachten und wieder in den Quelleditor zurückzuwechseln.

Heute, zu Zeiten schneller Rechner, ist dies ein durchaus praktikables, ja bequemes Verfahren: Als ich meine ersten Gehversuche mit (L^A)TeX, damals L^ATeX 2.09 machte, erforderte die Kompilation auch kleinerer Dokumente auf

¹ Ich danke den hilfreichen Geistern der TeX-D-L-Liste, insbesondere Christian Kuhn und Stefan Ulrich, für ihre wertvollen Beiträge zur Lösung des hier besprochenen Problems. Gerd Neugebauer gilt mein Dank für zahlreiche redaktionelle Hinweise; er hat mich außerdem vor einigen Fehlern bewahrt.

einem Heimcomputer beträchtliche Geduld. Auch wer diese Zeiten – es handelt sich um die zweite Hälfte der 80er Jahre – selbst erlebt hat, muss heute Mühe aufwenden, um sich die damaligen Verhältnisse zu vergegenwärtigen: Ich arbeitete damals auf einem mit 8 MHz getakteten Atari 1024 ST, in dem ein Motorola-68 000-Prozessor seinen Dienst tat. 1 MByte RAM – nicht wenig zu dieser Zeit – mussten genügen, und die 20-MB-Festplatte (heute braucht eine MiKTEX-Installation allein mindestens etwa 100 MB) war für damalige Verhältnisse außerordentlich großzügig bemessen – hatte ich doch kurz zuvor noch ganz ohne Festplatte mit nur einem 720 KB-Diskettenlaufwerk gearbeitet. – Unter diesen Bedingungen war die Kompilation eines (L)TEX-Dokuments stets eine lähmend langsame Prozedur, und insbesondere, wenn Schriften generiert werden mussten, war minutenlanges Warten angesagt. Heute dagegen ist die Übersetzung eines 200 Seiten starken Buches eine Sache weniger Sekunden.

Um nun die Korrektur eines Dokuments im Editor und die Begutachtung von deren Ergebnissen im Viewer möglichst effektiv zu gestalten, sollte beim Aufruf des Viewers aus dem Editor und umgekehrt die jeweils entsprechende Stelle im Dokument bzw. im Quelltext angesprungen werden.

Dieser Beitrag stellt eine Möglichkeit vor, dieses Ziel unter Windows bequem zu erreichen. Dabei ist eine Konfiguration aus Emacs Version 20.7 als Editor (mit AucTEX Version 9.9) und MiKTEX Version 2.0 up 1 (mit Yap Version 0.98 k) oder MiKTEX Version 2.1 (mit Yap 0.98 n) unter Windows 98 SE (deutsch) vorausgesetzt. Dies sind die Konfigurationen, unter denen das hier Ausgeführte getestet wurde. Es ist jedoch unwahrscheinlich, dass sich unter anderen neueren Windows-Systemen und Emacs-Versionen wesentliche Abweichungen ergeben könnten.

Anforderungen

Die Anforderungen, die sich auf diese Weise stellen, sollen hier noch einmal zusammengefasst und um einige Punkte erweitert werden, die sich beim Versuch der Umsetzung schnell als problematisch herausstellen:

1. Ein Aufruf von Yap soll, wenn das DVI-File einmal existiert, aus Emacs heraus möglich sein; dabei sollte an die Stelle gesprungen werden, an der der Cursor im Editor steht.
2. Die „inverse search“ des Editors Yap soll das Quelldokument in Emacs öffnen, falls es noch nicht offen ist, und zu der entsprechenden Stelle im

Quelltext springen (in Yap ist die entsprechende Stelle die, auf die man doppelklickt). Dabei soll zweierlei vermieden werden,

- (a) dass bei jedem Aufruf eine neue Kopie von Emacs im Arbeitsspeicher (oder mehr als ein Buffer für eine Datei) geöffnet wird. Diese Forderung dürfte in jedem Falle sinnvoll sein,
- (b) dass bei einem Aufruf immer ein neuer Frame geöffnet wird. Diese Forderung entspricht meiner Arbeitsweise und meinem Geschmack. Andere mögen hier andere Präferenzen haben.

Lösungen

Emacs mit AucTeX

Emacs ist zwar – gerade für Windows-Anwender – ein höchst gewöhnungsbedürftiger Texteditor. Er lässt sich jedoch gerade für (L^A)TeX so komfortabel konfigurieren, wie kaum ein anderes geeignetes Programm. Emacs fällt unter die GNU General Public License [7] und darf damit von jedem Anwender kostenlos verwendet und ggf. auch verändert und erweitert werden. Erhältlich ist es von jedem der zahlreichen FTP-Mirror-Server der GNU-Organisation, die unter [6] verzeichnet sind. Für den Windows-Anwender ist es wichtig, die Version für sein System (und nicht etwa eine Version für Unix-Systeme) zu erhalten. Windows-Versionen erkennt man an dem Namensteil „i386“ unmittelbar vor der Extension; gegenwärtig wird für die meisten Anwender die Datei `emacs-20.7-bin-i386.tar.gz` auf einem der Server, etwa auf dem der Rheinisch-Westfälischen Technischen Hochschule in Aachen [10], wohl die richtige Wahl sein.

Wer ein TeX-System wie MiKTeX unter Windows benutzt und als Editor Emacs verwendet, ist gut beraten, das Paket AucTeX zu nutzen, das unter anderem einen gegenüber Standard-Emacs erheblich verbesserten L^ATeX-Modus bereit stellt. Die neueste Version von AucTeX steht immer auf der offiziellen AucTeX-Seite [1], aber auch auf dem Server von DANTE e.V. [3] zum Download bereit. Die aktuelle Version ist derzeit 10.0 g und wird ebenfalls in einer `.tar.gz`-Datei bereit gestellt. Das Auspacken solcher Archive ist unter Windows zwar mit Bordmitteln nicht möglich; doch gibt es verschiedene Windows-Programme, die mit ihnen problemlos umgehen können. Eines der bequemsten ist das Freeware-Programm PowerArchiver [4], das auch eine weitgehende Unterstützung des Kontextmenüs bietet.

Die Installation von Emacs und AucTeX unter Windows verläuft gutmütig. Eine detaillierte Beschreibung der Installation findet sich in einem Artikel der Fachzeitschrift c't [11]. Der Artikel ist zwar schon etwas betagt, aber nach wie vor nützlich und gut auf die Bedürfnisse weniger erfahrener Anwender zugeschnitten. Hier sollen nur einige wenige unverzichtbare Punkte erörtert werden:

1. Emacs selbst kann an einer beliebigen Stelle in der Verzeichnisstruktur der Festplatte entpackt werden. Bei mir ist das `D:\emacs-20.7`. Dieses Verzeichnis wird im Folgenden `$EMACS$` genannt.
2. Ein Aufruf des Hilfsprogramms `addpm.exe`, das mit Emacs mitgeliefert wird und in `$EMACS$\bin` steht, macht die Installation perfekt. Ruft man es aus dem Explorer mit einem Doppelklick auf, so muss man keine weiteren Maßnahmen treffen, sodass dieses Verfahren vorzuziehen ist. In der MS-DOS-Eingabeaufforderung dagegen müsste man den Installationspfad explizit im Argument angeben, also etwa eintippen: `$EMACS$\bin\addpm $EMACS$`, wobei natürlich `$EMACS$` durch den Installationspfad zu ersetzen ist, in meinem Falle also durch `D:\emacs-20.7`. `addpm.exe` trägt dann den Emacs-Installationspfad in die Registry ein und legt eine Verknüpfung zum Aufruf von Emacs im Startmenü an.
3. Wichtig ist weiterhin, Emacs über eine Umgebungsvariable mitzuteilen, in welchem Verzeichnis die Dokumente liegen, auf die man üblicherweise zugreifen möchte, bei vielen Windows-Anwendern `C:\Eigene Dateien`. Unter Windows 98 oder ME muss man dazu in die Datei `autoexec.bat` eine Zeile wie `SET HOME=C:\Eigene Dateien` einfügen, wobei natürlich der tatsächlich gewünschte Pfad einzutragen ist. Unter Windows NT man die Umgebungsvariable `HOME` stattdessen auch einfacher setzen, indem man in der Systemsteuerung „System“ aufruft, die Registerkarte „Umgebung“ anklickt und dort die Variable `HOME` durch Eintrag des Namens bzw. Pfades in den Feldern „Variable“ und „Wert“ setzt. Dieser Pfad, der erst nach einem Neustart wirksam wird, muss auch die Emacs-Konfigurationsdatei `_emacs` enthalten, aus der Emacs seine Einstellungen liest. Eine Beispieldatei, die ein geeigneter Ausgangspunkt für eigene Experimente sein kann, steht, als Ergänzung zu [11], auf dem FTP-Server des Heise-Verlags zum Download bereit [9]. Diese Konfigurationsdatei muss nicht wie früher `_emacs` heißen, sondern kann sowohl unter Windows wie nahezu allen anderen Betriebssystemen `.emacs` genannt werden. Um die Verwendung der unveränderten Beispieldatei zu ermöglichen, gehe ich hier dennoch

von dem Namen `_emacs` aus. Wer den Namen `.emacs` bevorzugt, sollte, um die in Tabelle 1 angegebenen Tastenkombinationen voll nutzen zu können, in Zeile 132 der Datei „`_emacs`“ durch „`.emacs`“ ersetzen. Zudem enthält das Archiv unter anderem die Datei `tex-site.el`, die in den Unterverzeichnis `site-lisp` im Emacs-Verzeichnis gehört. In dieser Datei sind die Variablen `TeX-lisp-directory` und `TeX-macro-global` der eigenen Konfiguration anzupassen. Dadurch wird in Verbindung mit AucTeX der Aufruf von L^AT_EX direkt aus Emacs heraus ermöglicht.

4. `$EMACS$` enthält das Verzeichnis `site-lisp`. Darin ist AucTeX in einem Unterverzeichnis namens `auctex` (ohne weitere Angaben wie etwa die Versionsnummer) zu installieren. Die zu AucTeX gehörigen Dateien und Unterverzeichnisse stehen also sämtlich in dem Verzeichnis `$EMACS$\site-lisp\ auctex`.

Hat man in der genannten Weise Emacs mit AucTeX installiert, so ist nicht nur ein Anfang gemacht, sondern mit dem Anfang auch sprichwörtlich schon mehr als die Hälfte des Ganzen erreicht [2]. Konkret heißt das in unserem Falle, dass die als Ergänzung zu [11] verfügbare Konfigurationsdatei `tex-site.el` bereits dafür sorgt, dass L^AT_EX mit Source-Specials aufgerufen wird, die den DVI-Viewer Yap anweisen, an die Stelle zu springen, die der Cursor-Position beim Aufruf entspricht. Für den Aufruf von L^AT_EX, von Yap und für einige andere Zwecke stellen die Beispiel-Konfigurationsdateien `_emacs` und `tex-site.el` aus [11] eine Reihe bequemer Tastenkombinationen zur Verfügung, die in Tabelle 1 zusammengestellt sind und die Einträge von AucTeX im „Command“-Menü ergänzen.

Gnuserv und Yap

Es bleibt noch die Aufgabe, dafür zu sorgen, dass auch der umgekehrte Weg offen steht, dass also ein Doppelklick auf eine Stelle des formatierten Dokuments im DVI-Viewer Yap die entsprechende Stelle im Quelltext in Emacs anspringt, ohne eine neue Instanz von Emacs im Arbeitspeicher anzulegen oder einen neuen Emacs-Frame zu öffnen, falls der Quelltext bereits in Emacs geöffnet ist. Um dieses Ziel zu erreichen, empfiehlt sich die Verwendung von Gnuserv, über dessen Beschaffung und Installation [5] umfassend informiert. Besonders aufschlussreich ist jedoch ein Erfahrungsbericht von David Vanderschel [12], aus dem man auch zahlreiche Hinweise auf Anwendungsbeispiele von Gnuserv entnehmen kann. Diese Veröffentlichung geht allerdings über

F1	gehe zu Zeile Nr. ...
F2	aktuellen Puffer speichern
F3	Datei laden
C-F3	_emacs laden
S-F3	tex-site.el laden
F4	Speedbar
S-F4	in das nächste Fenster schalten
M-F4	Datei schließen
F5	aktuelle Zeile einrücken
C-F5	L ^A T _E X interaktiv
F6	Rest der Zeile löschen
C-F6	T _E X-Viewer Yap
F7	aktuelle Zeile verdoppeln
F8	nächstes Wort löschen
F9	Lesezeichen setzen
C-F9	zum Lesezeichen springen
F10	Menüleiste
C-F12	Sofortdruck mit Notepad
S-F12	interaktiver Druck mit Wordpad
Alt-F12	PostScript-Druck mit Ghostscript

Tabelle 1: Tastenkombinationen für die Arbeit mit Emacs und L^AT_EX.

den Zweck einer Verwendung in einem T_EX-System hinaus. So kann man vieles über die Zuordnung von Dateien zu Gnuserv unter Windows (die dann nach einem Doppelklick mit Emacs geöffnet werden), über die Aufnahme von Emacs-Optionen in die Kontextmenüs und Ähnliches lernen.

Auch für die Installation würde ich weniger kenntnisreichen Anwendern, zu denen ich mich selbst zähle, eher den Erfahrungsbericht von D. Vanderschel [12] als die Hauptseite [5] empfehlen. Der Bericht ist sehr klar geschrieben und auch für Leute, die keine größere Erfahrung mit der Konfiguration von Emacs haben, sondern den Editor nur anwenden, gut verständlich.

Die Installation von Gnuserv, beschrieben in [12], vollzieht sich im Wesentlichen in vier Schritten:

1. Auspacken von `gnuserv-2.1p1.zip` (erhältlich zum Beispiel auf [8]).
2. Verschieben der so extrahierten `.exe`-Dateien in das `\bin`-Verzeichnis von Emacs (bei mir zum Beispiel `d:\emacs-20.7\bin`).

3. Verschieben von `gnuserv.el` in ein Verzeichnis, aus dem Emacs Konfigurationsdateien lädt (das Verzeichnis `site-lisp` ist ein guter Kandidat) und Kompilieren der Datei zu `gnuserv.elc`. Diese Kompilation (von der ich nicht ganz sicher bin, ob sie unabdingbar ist; zumindest beschleunigt sie die Arbeit mit Gnuserv) hat mir etwas Kopfzerbrechen bereitet. Eigentlich geht es jedoch ganz einfach: Man ruft aus Emacs (mit `M-xdired`) `dired` auf, wechselt in das Verzeichnis, in dem `gnuserv.el` steht (bei mir etwa `d:\emacs-20.7\site-lisp`), setzt den Cursor auf den Eintrag „`gnuserv.el`“ und drückt `Shift-B`. Es wird dann eine Datei `gnuserv.elc` in demselben Verzeichnis erzeugt.
4. Schließlich kopiert man den von Vanderschel ziemlich am Anfang unter „Installation“ gelisteten Code in die eigene Emacs-Konfigurationsdatei `_emacs`.

Anpassen von Yap und der Emacs-Konfigurationsdatei

Damit ist Gnuserv grundsätzlich funktionsfähig, und ich kann nur noch einmal summarisch auf die Anwendungsbeispiele in [12] verweisen. Hier bleibt die Frage zu klären, wie man

1. Yap so konfiguriert, dass es Gnuserv in geeigneter Weise aufruft, und
2. verhindert, dass jeder Aufruf einen neuen Frame öffnet.

Die Konfiguration von Yap ist verhältnismäßig banal. Sie erfolgt dadurch, dass man unter „View|Options: Inverse Search“ folgende Kommandozeile einträgt:

```
d:\emacs-20.7\bin\gnuclient.exe -q -F +"%l" "%f"
```

Dabei muss statt `d:\emacs-20.7\bin\` der Pfad stehen, in dem die Programmdatei `gnuclient.exe` tatsächlich steht, also in der Regel der `\bin`-Pfad von Emacs.

Dies garantiert allerdings nur, dass bei wiederholten Aufrufen nicht mehrere Kopien von Emacs im Arbeitsspeicher (und auch nicht mehr als ein Buffer für eine Datei) geöffnet werden. Es wird jedoch nach wie vor bei jedem Aufruf ein neuer Emacs-Frame geöffnet. Dies mag der Arbeitsweise mancher Anwender durchaus entgegenkommen. Wer dies jedoch vermeiden will, braucht lediglich Folgendes an das Ende der Konfigurationsdatei `_emacs` zu setzen:

```
(setq gnuserv-frame (selected-frame))
```

Dies hat zur Folge, dass einzig der zu diesem Zeitpunkt offene und in der Regel beim Start von Emacs geöffnete Frame von Gnuserv verwendet wird. Nachteilig ist, dass Gnuserv keinen dafür zuständigen Frame findet, falls dieser irgendwann geschlossen wurde; dem lässt sich jedoch ggf. durch Neustart von Emacs abhelfen.

Bei der Wahl der Dateinamen gibt es eine Einschränkung, deren Ursache ich nicht ermitteln konnte: Während Dateinamen, die Leerzeichen enthalten, nicht zu Problemen führen, ist unbedingt darauf zu achten, dass solche, die mit Ziffern beginnen, vermieden werden. Derlei Dateinamen führen dazu, dass die Übergabe der Stelle, an die Yap springen soll, misslingt.

Zusammenfassung

Die vorgestellte Konfiguration erlaubt einen perfekten Zyklus aus *Bearbeitung des Quelltextes – Formatierung und Begutachtung des Dokuments – Rücksprung in den Editor*. Der Aufruf von Yap erfolgt dabei einfach aus AucTEXs L^AT_EX-Menü in Emacs heraus oder durch Betätigung einer entsprechenden Tastenkombination (C-F6). Yap springt dann an die Stelle im formatierten Dokument, die der Stelle im Quelltext entspricht, an der bei seinem Aufruf der Cursor stand. Umgekehrt springt Emacs auf einen Doppelklick in das Dokument in Yap hin an die entsprechende Stelle im Quelltext.

Dies funktioniert selbst dann perfekt, wenn man aus Yap an eine Stelle in einem mit `\input` oder `\include` eingebundenen Teildokument springen will, das zuvor in Emacs noch gar nicht geöffnet war. Emacs öffnet in diesem Fall automatisch die entsprechende Datei und springt an die verlangte Stelle.

Literatur

- [1] Per Abrahamsen: *AUC_TE_X*; <http://mirrors.sunsite.dk/auctex/www/auctex/>; offizielle AucTEX-Site.
- [2] Aristoteles: *Politik*; I 7, 1098^b7.
- [3] *AucTEX*; <ftp://ftp.dante.de/tex-archive/support/auctex/>; AucTEX auf dem Server von DANTE e.V.
- [4] Cone Xware, Inc: *PowerArchiver – The Ultimate Archiving Utility*; <http://www.powerarchiver.com>; PowerArchiver Homepage.

- [5] Claus Dethlefsen: *L^AT_EX, Emacs etc. for your PC – Tuning*; <http://www.math.auc.dk/~dethlef/Tips/tuning.html>.
- [6] Free Software Foundation: *FTP Mirror List*; <http://www.gnu.org/order/ftp.html>; Liste der Mirrors des GNU-FTP-Servers.
- [7] *Gnu General Public Licence*; <http://www.gnu.org/copyleft/gpl.html>.
- [8] *gnuserv*; <http://www.gnu.org/software/emacs/windows/ntemacs/contrib/gnuserv-2.1p1.zip>; GnuServ Download Site.
- [9] *Heise c't T_EX-Konfigurationsdateien*; <ftp://ftp.heise.de/pub/ct/listings/9922-228.zip>.
- [10] Andrew Innes: *Precompiled Distribution of Emacs for Windows NT and Windows 95/98/2000*; <ftp://ftp.informatik.rwth-aachen.de/pub/gnu/windows/emacs/20.7/emacs-20.7-bin-i386.tar.gz>; Emacs erhält man beispielsweise hier.
- [11] Dr. Justus Noll und Harald Bögeholz: *Wordlos glücklich. Umstieg auf das Textsatzsystem T_EX, Teil 1; c't*; 22, S. 228–235; 1999.
- [12] David Vanderschel: *gnuserv: Various Tips for Fully Exploiting this Functionality*; <http://www.math.auc.dk/~dethlef/Tips/gnuserv.txt>.

cjk-enc – eine Schnittstelle zwischen Emacs und L^AT_EX

Werner Lemberg

Im folgenden Artikel wird das Lisp-Programm `cjk-enc.el` beschrieben, eine Schnittstelle zwischen Emacs und L^AT_EX, welche die verschiedenen Zeichensätze von Emacs in für L^AT_EX verständliche Befehle konvertiert. Ursprünglich für ostasiatische Sprachen entwickelt, wurde es inzwischen um europäische Zeichensätze (auch Russisch und Neugriechisch) erweitert. `cjk-enc.el` ist Teil des CJK-Pakets.

Einleitung

Emacs ist einer der mächtigsten Editoren, die frei verfügbar sind.¹ Seit einigen Jahren bereits unterstützt er die simultane Verwendung von mehreren Zeichensätzen, im besonderen asiatische für Chinesisch oder Japanisch, aber auch solche für Thai, Vietnamesisch, Neugriechisch und Russisch. Auch Zeichensätze, die auf der lateinischen Schrift beruhen (latin-1, latin-2, usw.) können problemlos gleichzeitig verwendet werden.

Da `cjk-enc.el` eine temporäre Datei erzeugt und das eigentliche Dokument nicht verändert, ist die tatsächliche Kodierung des Dokuments (z. B. ISO-2022 oder UTF-8) irrelevant. Diese temporäre Datei kann direkt mit L^AT_EX verarbeitet werden.

Installation

Um `cjk-enc.el` zu installieren, genügt es, folgende Zeile in die Konfigurationsdatei von Emacs einzufügen (normalerweise heißt diese `.emacs`):

```
(load-library "cjk-enc")
```

Die Datei `cjk-enc.el` selbst muss sich in einem Verzeichnis befinden, in dem Emacs Lisp-Dateien sucht, z. B. `/usr/local/share/emacs/site-lisp`.

Nach dem Laden ist eine neue Ausgabekodierung (`'cjk-coding'`) zur Konversion in das L^AT_EX-Zwischenformat definiert.

Die L^AT_EX-Schnittstelle

Normalerweise braucht man nichts Spezielles tun. Zeichen wie “ 8u” oder “ 51c” werden automatisch in L^AT_EX-Makros konvertiert. Wie auch bei anderen mehrsprachigen Dokumenten müssen jedoch sprachenspezifische Trennmuster ausgewählt werden, sodass die Verwendung des Babel-Pakets ratsam ist. Ebenso sollte man T₁-kodierte Schriften verwenden, da T₁ die meisten europäischen Zeichen abdeckt.

Tabelle 1 listet die unterstützten Zeichensätze auf (auch die ostasiatischen, auf die aber hier nicht näher eingegangen wird). Zu beachten ist, dass nicht

¹ Ich gehe hier nicht weiter auf die beiden wichtigsten Inkarnationen (FSF Emacs und XEmacs) ein, da `cjk-enc.el` beide Versionen unterstützt.

latin-iso8859-1	ISO/IEC 8859-1 (Latin 1)
latin-iso8859-2	ISO/IEC 8859-2 (Latin 2)
latin-iso8859-3	ISO/IEC 8859-3 (Latin 3)
latin-iso8859-4	ISO/IEC 8859-4 (Latin 4)
latin-iso8859-9	ISO/IEC 8859-9 (Latin 5)
cyrillic-iso8859-5	ISO/IEC 8859-5
greek-iso8859-7	ISO/IEC 8859-7
thai-tis620	TIS 620-2533
vietnamese-viscii- <code>\{lower,upper\}</code>	VISCII 1.1
latin-jisx0201	JIS X 0201-1976
katakana-jisx0201	JIS X 0201-1976
chinese-gb2312	GB 2312-1980
japanese-jisx0208	JIS X 0208:1997
japanese-jisx0212	JIS X 0212-1990
korean-ksc5601	KS X 1001:1992 (KS C 5601-1992)
chinese-cns11643-[1234567]	CNS 11643-1992 Ebenen 1-7
chinese-big5-[12]	Big 5

Tabelle 1: Eine Liste der von `cjk-enc.el` unterstützten Zeichensätze in Emacs Version 20.7 – Emacs 21 wird auch neuere ISO-8859-Varianten unterstützen, welche das Euro-Symbol enthalten. Auf der linken Seite ist der Zeichensatzname in Emacs aufgeführt, rechts der Standard, welcher den Zeichensatz definiert.

59	<code>\kra</code>	U+0138 LATIN SMALL LETTER KRA (Grönländisch)
0F	<code>\TBAR</code>	U+0166 LATIN CAPITAL LETTER T WITH STROKE (S20ami)
0t	<code>\tbar</code>	U+0167 LATIN SMALL LETTER T WITH STROKE (S20ami)

Tabelle 2: Diese Zeichen gibt es weder in T1, noch haben sie einen vordefinierten Namen in der Standard-L^AT_EX-Distribution. Sie werden in latin-4 benötigt (fehlen aber in der Datei `latin4.def` der neuesten L^AT_EX-Version). Auf der rechten Seite ist der Kodepunkt und Zeichenname in Unicode angegeben. Die hier gezeigten Makronamen sind in `MULEenc.sty` definiert.

alle Zeichen in L^AT_EX standardmäßig vorhanden sind (siehe Tabellen 2 und 3).


```

0H \textmalteseH U+0166 LATIN CAPITAL LETTER H WITH STROKE
0h \textmalteseH U+0167 LATIN SMALL LETTER H WITH STROKE

```

Tabelle 3: Diese Zeichen gibt es nicht in T₁, sind aber bereits in L^AT_EX definiert (für latin-3). Sie werden für Maltesisch gebraucht. Aus technischen Gründen benutzt MULEnc.sty das Makro \TEXTMALTESEH anstelle von \textmalteseH.

Alle benötigten Makros befinden sich in der Datei MULEnc.sty, welche automatisch inkludiert wird und daher nicht explizit mittels \usepackage geladen werden muss. Technische Details werden weiter unten beschrieben.

Für Vietnamesisch wird die Fontkodierung T₅ benötigt, welche im vnr-Paket enthalten ist [2]. Thailandisch wurde bereits in einer früheren Ausgabe der T_EXnischen Komödie beschrieben [3].

Die derzeitige Implementation für die kyrillische Schrift benötigt viel temporären Speicherplatz (für einen einzigen Buchstaben bis zu acht Bytes zusätzlich, da Makronamen wie \CYRSHCH für ‘0ш’ oder \cyrhrdsn für ‘0ъ’ verwendet werden). Es muss explizit eine der T₂-Kodierungen aktiviert werden – Russisch braucht T_{2A}. Leider ist die Unterstützung von Emacs für Kyrillisch auf ISO 8859-5 beschränkt, sodass beispielsweise Ukrainisch nicht komplett dargestellt werden kann (es fehlen die Zeichen ‘І’ und ‘і’, U+0490/U+0491 CYRILLIC CAPITAL/SMALL LETTER GHE WITH UPTURN, die als Makros eingegeben werden müssen).

Neugriechisch wird durch ISO 8859-7 in Emacs abgedeckt, Altgriechisch dagegen nicht. Jedoch kann mit Hilfe von cjk-enc bei Altgriechisch auf die Verwendung einer Transliteration verzichtet und direkt griechische Buchstaben für die Ligaturen mit den diakritischen Zeichen verwendet werden. cjk-enc verwendet die LGR-Fontkodierung; z.B. erzeugt ‘>\textasciitilde\textgreek{0h}’ die Ausgabe “~0η”.

Die Emacs-Schnittstelle

cjk-enc stellt eine Reihe von Funktionen zur Verfügung, die am besten bestimmten Tasten zugeordnet werden. Die wichtigste Funktion ist cjk-write-file, um eine temporäre Datei zu erzeugen. Normalerweise wird die Endung .tex durch .cjk ersetzt, für Bib_TE_X-Dateien jedoch durch -cjk.bib, da Bib_TE_X stets .bib als Endung benötigt. Außerdem wird dadurch eine Na-

menkollision vermieden, da ja L^AT_EX selbst Hilfsdateien mit dieser Endung erzeugt. Der spezielle Befehl `\CJKbibliography\{foo\}` l^adt `foo-cjk.bib`.

8Ahnliche L^AT_EX-Makros sind `\CJKinclude\{bar\}` und `\CJKinput\{foo.bar\}`; ersterer inkludiert `bar.cjk` anstelle von `bar.tex`, letzterer dagegen l^adt die Datei `foo.bar` – nur wenn die Datei keine Endung hat, wird `.cjk` angeh^angt.

Die Emacs-Funktion `cjk-write-all-files` erkennt diese drei Makros. Angenommen, die Dateien `a.tex` und `b.tex` enthalten Kapitel eines Dokumentes, welche mehrere Zeichens^atze von Emacs ben^utzen, dann w^urden sie mit

in der Mutterdatei `master.tex` inkludiert. Durch den Aufruf der Funktion `cjk-write-all-files` werden diese drei Dateien konvertiert, und das Dokument kann mit

```
latex master.cjk
```

bearbeitet werden.

Um z. B. `cjk-write-file` der Taste F11 zuzuordnen, muss folgende Lisp-Anweisung in `.emacs` eingefügt werden:

```
(global-set-key [f11] 'cjk-write-file)
```

In der Programmiersprache Lisp hat der Apostroph die gleiche Aufgabe wie `\noexpand` in T_EX; es verhindert die Expansion des folgenden Tokens.

Wichtig ist es, dass die Kodierung der Datei für Emacs erkenntlich ist. In vielen Fällen funktioniert der Algorithmus zum automatischen Erkennen, doch gibt es Fälle, die garantiert fehlschlagen (latin-1 ist von latin-2 nicht zu unterscheiden, um ein Beispiel zu nennen). Die beste Lösung ist es, eine Dateivariabel² namens `coding` zu verwenden, um die Dateikodierung anzugeben. Es gibt zwei Möglichkeiten, Dateivariablen in einer Datei zu setzen:

- Zu Beginn einer Datei (innerhalb eines Kommentars) sehen Dateivariablen so aus:

```
-*- var1: val1; ... -*-
```

Beispiel:

- Die zweite Möglichkeit ist eine Dateivariablenliste am Ende der Datei, ebenfalls innerhalb eines Kommentars:

```
Local Variables:
var1: val1
...
End:
```

Beispiel:

Der momentane Wert der Dateikodierung kann mit der Tastenkombination `Ctrl-h C` abgefragt und mit `Ctrl-x RET f` gesetzt werden.

²Im englischen Handbuch bzw. in den Info-Dateien von Emacs werden Dateivariablen unter dem Eintrag *local variables in files* dokumentiert.

Technische Details

Der interne Mechanismus wurde bereits einmal vor l8angerer Zeit in „Die T_EXnische Komödie“ beschrieben [4], doch haben sich seitdem einige Details ge8andert. Kernidee ist es, den ASCII-Eingabekode 0x7F (<delete>) zu einem aktiven Zeichen zu machen.³ Dem Autor ist kein Paket bekannt, welches diesen Kode ben8otigt – in T_EX ist es ja standardm8a0ßig als ung8ultiges Zeichen mit dem Catcode 15 deklariert und somit ohne Funktion. Vom typographischen Standpunkt ist <delete> ebenfalls vollkommen nutzlos. Ein weiterer Vorteil ist, dass in allen Zeichens8atzen, die ASCII erweitern (und genau solche ben8utzt Emacs), dieses Zeichen stets den gleichen Kode hat.

0x7F ist als Multiplex-Makro definiert, welches anhand seiner Parameter weitere Makros aufruft⁴; es funktioniert daher auch in verbatim-Umgebungen. Zuerst allerdings muss verhindert werden, dass es unkontrolliert expandiert:

`\protect` ist in L^AT_EX entweder als `\@typeset@protect` oder als `\string` definiert; trifft letzteres zu, werden Daten in Hilfsdateien oder 8ahnliches geschrieben und Makros d8urfen nicht expandieren. Der Befehl `\expandafter` bewirkt, dass zuerst `\fi` gelesen wird (um den `\if`-Block zu beenden), bevor `\mule@arg` selbst behandelt wird.

Damit die Argumente von `\mule@arg` vor `\uppercase` (und auch `\lowercase`) gesch8utzt sind, werden nur Zahlen als Parameter verwendet bzw. Glyphnamen wie `\cyra`, deren gro0ßgeschriebene Formen (`\CYRA`) ebenfalls g8ultige Glyphnamen sind. 0x7F dient gleichzeitig als Parameterbegrenzer.

Ein Beispiel: Das Zeichen “8a” wird durch cjk-enc als folgende Byte-Sequenz repr8asentiert: `<0x7f, 8, 0x7F, a, 0x7F>`. Die Zahl 8 wird als Index f8ur Makros der Gestalt `\mule@n` benutzt; im konkreten Fall expandiert `\mule@8` zu `\}`.

Um Unterschnitte und Ligaturen nicht zu zerst8oren, darf nur Makroexpansion erfolgen, daher die seltsam anmutende Definition von `\mule@arg`, wodurch das zweite Argument f8ur einen Test verdoppelt wird.

³ 0x7F wird erst seit der Version 4.4.0 ben8utzt, deren Betaversion als aktueller Schnappschuss von <ftp://ftp.ffii.org/pub/cjk/devel/cjk-current.tar.gz> erh8altlich ist; es wird ebenfalls Bestandteil der CD-ROM „T_EX Live 6“ sein.

⁴ Alle folgenden Beispiele setzen die Anweisung `\endlinechar -1` voraus, damit Zeilenenden nicht durch “%” gesch8utzt werden m8ussen.

Das Makro `\mule@@arg` testet nun, ob der zweite Parameter von `0x7F` mit `'\'` beginnt, das einen regulären Catcode 0 hat oder dessen Catcode 12 ist (das kann in verbatim-Umgebungen vorkommen: z.B. wird `"0æ"` in `<0x7F, 0, 0x7F, \, a, e, 0x7F>` konvertiert). Im ersteren Falle wird `\mule@@@arg` ausgeführt.

Hier die Definition von `\mule@ifbslash`.

```
@@
```

```
\endVerbatim
```

```
\beginVerbatim \endVerbatim
```

```
\beginVerbatim
```

Der Catcode von `'|'` wird lokal auf 0 gesetzt, damit `'\'` zu einem *other character* gemacht werden kann. `\mule@ifbslash` ruft `\mule@@ifbslash` auf und fügt `'\'` in die Parameterliste ist. Nur wenn #1 (von `\mule@ifbslash`) das Zeichen `'\'` gefolgt von anderen Zeichen enthält, ist #2 von `\mule@@ifbslash` nicht leer. Anderenfalls enthält #1 die gesamte Zeichenkette. Es gibt in MULEenc.sty keine Makros der Gestalt `\xxxmule@@@`; nur `\mule@@@` ist definiert, mithin liefert der Test mit `\ifx` das gewünschte Resultat. Da ja nur Makroexpansion erlaubt ist, darf nicht innerhalb von `\mule@@bslash` ein temporäres Makro für den Test definiert werden, was die nächstliegende Idee wäre.

Auch in die Parameterliste von `\mule@exec` wird `"\"` eingefügt, um dieses Zeichen zu schlucken. Durch zweimaligen Aufruf von `\csname` wird jetzt der gleiche Effekt erzielt wie mit `\mule@@@arg`.

Wie schon vorher erwähnt, werden Indexnummern (in willkürlicher Ordnung) für Makros verwendet:

```
\mule@def{0}{\@firstofone}
...
\mule@def{8}{\"}
...
\mule@def{33}{\~}
\mule@def{34}{\~}
\mule@def{48}{\k}
\mule@def{49}{\u}
```

...

`\mule@0` ruft einfach den zweiten Parameter von `0x7F` auf.⁵ Es ist gedacht f8ur Glyphnamen, die sowohl gro0ß- als auch kleingeschrieben vorhanden sind, z.B. `\aa` und `\AA`.

`MULEenc.sty` enth8alt mehr als 200 solcher Definitionen; ein Gro0ßteil davon ist f8ur den asiatischen Teil des CJK-Pakets notwendig.

Ausblick

Das simultane Ben8utzen mehrerer Zeichens8atze ist aus heutiger Sicht veraltet. Der notwendige Standard, ISO 2022, ist hochkompliziert und nicht einmal in Emacs vollst8andig implementiert. Unicode [1] bietet einen viel besseren und weniger problembehafteten Ansatz zum Erstellen multilingualer Dokumente. Auch ist die Anzahl der Unicode-Editoren inzwischen bedeutend gr8o0ßer als solche, die ISO 2022 unterst8utzen. Tatsache ist allerdings, dass weder die derzeitige noch die n8achste Version von Emacs Unicode direkt implementiert hat.⁶

Dasselbe Problem hat $\text{\TeX}$. Obwohl mit Ω bereits ein Unicode-Nachfolger verf8ugbar ist, wird $\text{\TeX}$ aus verschiedensten Gr8unden weiterhin f8ur mehrsprachige Dokumente “missbraucht”. F8ur einfache Anwendungen gibt es mindestens zwei Unicode-Implementationen direkt f8ur $\text{\TeX}$ ([6], [7]), doch leiden auch sie unter der Begrenzung auf 256 Zeichen in einem Zeichensatz, sodass Unterschnitte und Ligaturen nur begrenzt m8oglich sind. Sprachen wie Thai, die eine linguistische Analyse zur korrekten Worttrennung benötigen, um korrekte Worttrennungen zu finden - es gibt keinen Zwischenraum zwischen W8ortern - k8onnen ohne externe Unterst8utzung (z.B. durch einen Pr8aprozessor wie `cjk-enc`) 8uberhaupt nur beschr8ankt unterst8utzt werden.

Emacs ben8utzt weitaus mehr Zeichens8atze, als in `cjk-enc` implementiert sind, so z.B. Devanagari oder Tibetisch. Geplant ist, bereits existierende Pakete f8ur diese Sprachen in `cjk-enc` zu integrieren.

⁵`@firstofone` ist in `latex.ltx` definiert als

```
\long\def\@firstofone#1{\#1}
```

⁶Es gibt f8ur Emacs allerdings ein Paket namens `mule-ucs`, welches eine gute Unicode-Emulation bietet.

Diese Arbeit ist keine Zeitverschwendung, da bei einer späteren Umstellung auf Unicode sich bloß die Eingabekodierung ändert, die Schriften und dafür eventuell benötigte Prozessor nicht. Der Autor wäre für weitere Vorschläge und Kritik sehr dankbar.

Literatur

- [1] *The Unicode Standard*: <http://www.unicode.org>.
- [2] *Die vnr-Schriftfamilie*, entwickelt von H32an Th20ê Th32anh, dem Autor von pdf_EX.
- [3] Werner Lemberg: *Thai-Zeichensätze*. In: Die T_EXnische Komödie, Heft 4/2000.
- [4] Werner Lemberg: *Das CJK-Paket für L^AT_EX 2_ε*. In: Die T_EXnische Komödie, Heft 4/1995.
- [5] *Das CJK-Paket*: <http://cjk.ffii.org>.
- [6] *ucs.sty*, ein inputenc-Modul für UTF-8, entwickelt von Dominique Unruh: <http://www.unruh.de/DniQ//latex/unicode/>.
- [7] *Das utf8-Paket*, entwickelt von (Vladimir Volovich): <ftp://pub/tex/macros/latex/contrib/supported/t2/etc/utf-8/>.

LyX – Open Source Document Processor, Teil 1

Laura E. Jackson, Herbert Voß

LyX als sogenanntes Frontend zu $\text{\LaTeX}$, welches wiederum ein Frontend zu $\text{\TeX}$ ist, versucht etwas zu optimieren, was $\text{\LaTeX}$ selbst gar nicht leisten will: eine hochwertiges Satzsystem wie $\text{\TeX}$ einem an tiefsinnigen typographischen Fragen relativ uninteressierten Benutzer zugänglich zu machen. LyX versteht sich daher auch nicht als Textprozessor, sondern als Dokument-Prozessor, denn es verfügt selbst über keinerlei eigene Formatierungsmöglichkeiten, die über die Bildschirmdarstellung hinausgehen.

Alle beschriebenen Eigenschaften von LyX beziehen sich auf die derzeit offizielle Version LyX 1.1.6fix3. Eine erheblich erweiterte Version 1.2 ist für dieses Jahr zu erwarten.

Die Geschichte

Unter Federführung von Donald Knuth wurde ausgehend von den Prinzipien des herkömmlichen Drucksatzes ein System entwickelt, welches Anwendern einen professionellen Satz wissenschaftlicher Publikationen ermöglichte. In der ersten Version nannte Knuth dieses System $\tau\epsilon\chi$, sehr bald nur noch $\text{\TeX}$ geschrieben, aber immer „Tech“gesprochen.

1977 Erste veröffentlichte Version von $\text{\TeX}$

1982 Zweite Version

1990 Dritte und letzte Version

Eine weise Entscheidung traf Knuth, indem er beschloss, $\text{\TeX}$ *nicht* weiterzuentwickeln! Das Problem in der Software-Entwicklung ist keinesfalls eine in ihren Eigenschaften mehr oder weniger begrenzte Anwendung zu entwickeln, sondern zu erkennen, dass man sich mit einer ständigen Weiterentwicklung auf dem Weg der „eierlegenden Wollmilchsau“ befindet, der sogenannten universellen Anwendung. Diese fordert ab einem bestimmten Punkt einen immer größer werdenden

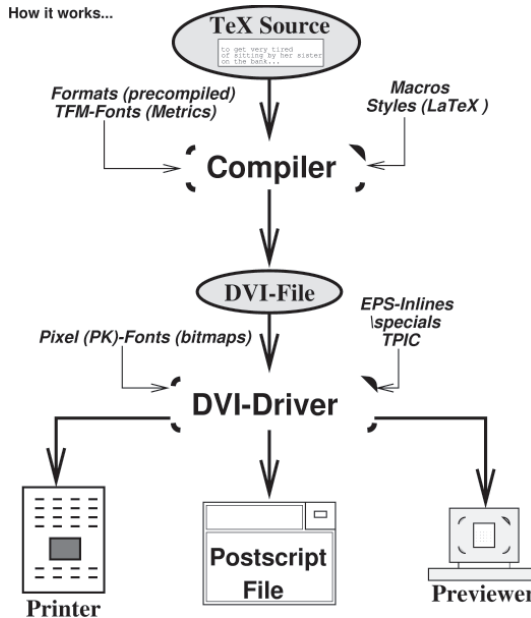


Abbildung 1: Der formale Aufbau eines $\text{\TeX}$ -Systems
(<http://www.rhrk.uni-kl.de/~knopper/tex/howitworks.gif>)

Entwicklungs- und Wartungsaufwand bei einem immer kleiner werdenden Fortschritt in der Anwendung. $\text{\TeX}$ kann auch als erstes wegweisendes Open-Source-Projekt verstanden werden, denn Knuth stellte von Anfang an den gesamten, in der Sprache WEB erstellten Quellcode von $\text{\TeX}$ zur Verfügung. Dies führte dazu, dass $\text{\TeX}$ faktisch für jedes System angepasst wurde.

$\text{\TeX}$ in seiner Ursprungsversion kann nicht nur aus heutiger Sicht als kryptisch bezeichnet werden, was Leslie Lamport bereits Anfang 1980 veranlasste, eine Sammlung von Makropaketen zu entwickeln, welche als $\text{\TeX}2_{\epsilon}$ den heutigen aktuellen Stand darstellen [4]. Daneben gibt es eine Vielzahl von anderen Entwicklungen, die alle auf $\text{\TeX}$ basieren, hier aber nicht weiter erwähnt werden sollen. Information gibt es reichlich im Comprehensive $\text{\TeX}$ Archiv Network (CTAN - <http://www.ctan.org>; gespiegelt bei DANTE, Deutschsprachige

Anwendervereinigung T_EX e.V.). Seit einigen Jahren wird intensiv an einer Fortentwicklung gearbeitet, die unter dem Projektnamen $\LaTeX$ 3 läuft, während L^AT_EX selbst die Arbeit an $\LaTeX$ eingestellt hat. Den formalen Aufbau eines T_EX-Systems zeigt Abbildung 1, woraus deutlich zu erkennen ist, dass $\LaTeX$ nur eine untergeordnete Bedeutung in diesem Gesamtsystem spielt.

Das Fehlen einer grafischen Benutzeroberfläche für $\LaTeX$ kann grundsätzlich als unwesentlich betrachtet werden, stellte aber in der heutigen „Alles-so-schön-bunt-Welt“ ein wesentliches Kriterium dar, um 1994/95 im Rahmen einer Diplomarbeit eine grafische Oberfläche zu entwickeln, die in Anlehnung an „Lyrik“ über „LyriX“ nun LyX genannt wird. Ziel der Entwicklung war und ist es immer noch, eine Umgebung für das in Abbildung 1 dargestellte T_EX-System zu entwickeln, die den Anwender auch von tiefergehenden $\LaTeX$ -Kenntnissen befreien sollte. Bleibt also festzustellen, dass man eigentlich *immer* von T_EX reden müsste, denn alle anderen Programmumgebungen sind nur aufgesetzt. Für das Verständnis der Philosophie von LyX ist dies jedoch unerheblich, insbesondere für den Nutzer, der LyX gerade wegen der gewissen T_EX- bzw. $\LaTeX$ -Ferne anwenden möchte. Neben LyX gibt es noch verschiedene andere grafische Oberflächen für $\LaTeX$, wie beispielsweise *ε:doc* (<http://members.magnet.at/hfbuch/edoc/>).

Wysiwyg versus Wysiwym

Ein Schlagwort der 90er Jahre in der Software-Entwicklung war das Wysiwyg-Prinzip (*What You See is What You Get*), was dem Anwender eine schnelle Kontrolle seiner Arbeit ermöglichen sollte. Einerseits bereitet diese Forderung erhebliche Probleme bei der Software-Entwicklung und andererseits ist festzustellen, dass im Bereich der wissenschaftlich-technischen Publikationen ein wesentlicher Teil des Layouts durch DIN-Normen oder Verlagsvorgaben festliegt; der Autor hat sich in diesem Rahmen zu bewegen. Daher entschied man sich, bei der Entwicklung von LyX das Wysiwym-Prinzip (*What You See is What You Mean*) anzuwenden, d. h. der Anwender hat nur eine globale Ähnlichkeit zwischen dem, was die grafische Oberfläche anzeigt, und der späteren Ausgabe. Dabei hängt die Ähnlichkeit von der Komplexität des Textes ab. Sie liegt im Bereich

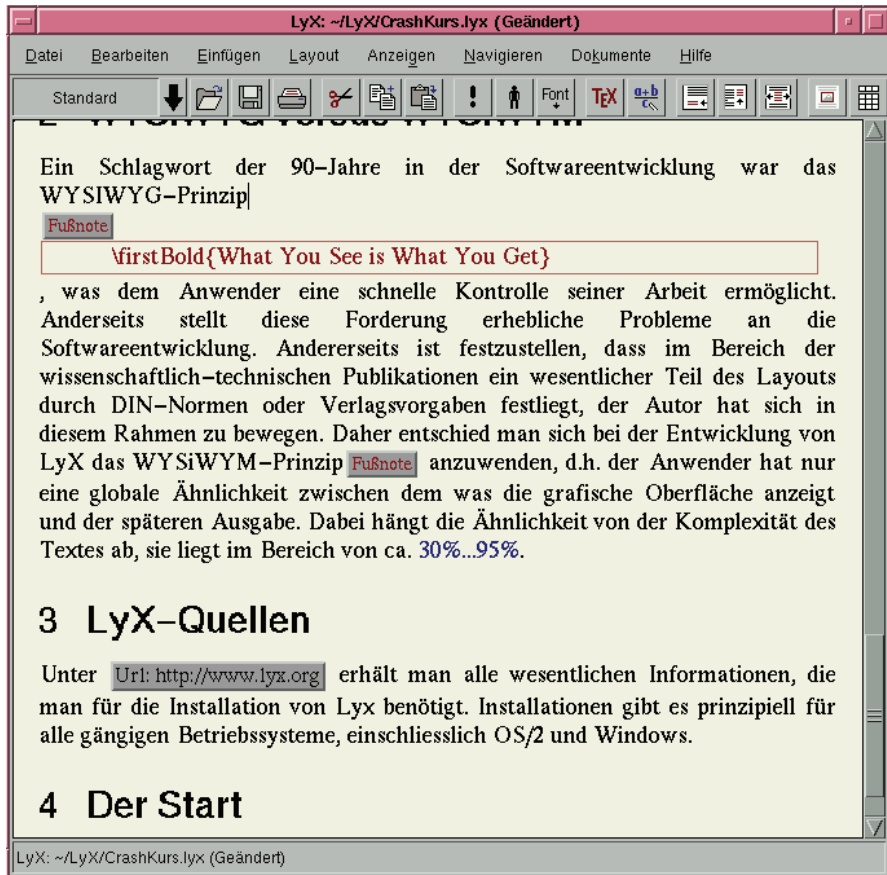


Abbildung 2: Die LyX-Oberfläche

von ca. 30%...95%, was exemplarisch durch Abbildung 2 gezeigt wird.

Damit erst gar kein Missverständnis auftritt, soll noch einmal klar gestellt werden, dass Wysiwym ein Kompromiss ist und dass gegen die Kombination von $\text{\TeX}$ und Wysiwyg nichts, aber auch gar nichts spricht.

$$\begin{aligned}
 A &= \lim_{n \rightarrow \infty} \Delta x \left(a^2 + \left(a^2 + 2a\Delta x + (\Delta x)^2 \right) \right. \\
 &\quad + \left(a^2 + 2 \cdot 2a\Delta x + 2^2 (\Delta x)^2 \right) \\
 &\quad + \left(a^2 + 2 \cdot 3a\Delta x + 3^2 (\Delta x)^2 \right) \\
 &\quad \quad \quad + \dots \\
 &\quad \left. + \left(a^2 + 2 \cdot (n-1) a\Delta x + (n-1)^2 (\Delta x)^2 \right) \right) \\
 &= \frac{1}{3} (b^3 - a^3)
 \end{aligned}$$

Abbildung 3: LyX (L^AT_EX-Formelsatz)

Warum LyX (L^AT_EX) und nicht ???

Es gibt hierfür keine überzeugende Antwort, wenn man dem typographischen Layout nicht sonderlich viel Aufmerksamkeit schenkt. So ist beispielsweise der Formelsatz in WinWord überhaupt nicht mit dem von L^AT_EX zu vergleichen, was die folgenden Abbildungen verdeutlichen sollen. Die WinWord-Formel (Abbildung 4) wurde mit Hilfe des Adobe PostScript-Treibers für Windows als PostScript-Datei ausgegeben und hier einfach eingebunden. Es trat somit keine nachträgliche Veränderung der Formel auf. Bei der Erstellung der Formel wurde der Standard-Zeichensatz von WinWord 2000 angewendet.

Der wesentliche Unterschied liegt in den verwendeten Fonts, denen man unter T_EX weitaus mehr Aufmerksamkeit bei der Erstellung gewidmet hat als den bekannten TrueType-Fonts unter Windows. Diese können im Allgemeinen unter L^AT_EX verwendet werden, wodurch sie allerdings auch nicht besser werden.

Unter Linux wird sehr häufig das Software-Paket StarOffice von SUN empfohlen, da es zum einen kostenlos und zum anderen professionell aufgebaut ist. Im Gegensatz zu L^AT_EX kann es als echtes Office-Paket mit Textverarbeitung, Tabellenkalkulation, Datenbank usw. gelten, während L^AT_EX bekanntermaßen nur eins kann: Textverarbeitung und -bearbeitung. Dies allerdings so gut wie wohl kein anderes Software-Paket

$$\begin{aligned}
 A = \lim_{n \rightarrow \infty} \Delta x & \left(a^2 + \left(a^2 + 2a\Delta x + (\Delta x)^2 \right) \right. \\
 & + \left(a^2 + 2 \cdot 2a\Delta x + 2^2 (\Delta x)^2 \right) \\
 & + \left(a^2 + 2 \cdot 3a\Delta x + 3^2 (\Delta x)^2 \right) \\
 & + \dots \\
 & \left. + \left(a^2 + 2 \cdot (n-1)a\Delta x + (n-1)^2 (\Delta x)^2 \right) \right) \\
 & = \frac{1}{3} (b^3 - a^3)
 \end{aligned}$$

Abbildung 4: WinWord-2000 Formelsatz

Jede Textverarbeitung hat bekanntermaßen ihre Stärken und Schwächen, sodass hierauf nicht weiter eingegangen zu werden braucht. Die Stärken von $\text{\LaTeX}$ liegen zweifelsohne im Bereich der technisch-naturwissenschaftlichen Literatur. Mittlerweile sind der Anwendung von $\text{\LaTeX}$ durch die Vielzahl an kostenlos zur Verfügung stehenden Paketen praktisch keine Grenzen mehr gesetzt. Am Schluss werden einige Beispiele dazu angegeben. Der Umfang eines kompletten installierten $\text{\TeX}$ / $\text{\LaTeX}$ -Pakets steht zwar einer Installation eines Office-Pakets in nichts nach, dennoch ergibt sich ein extremer Geschwindigkeitsvorteil für $\text{\TeX}$ bei der Formatierung von großen Dokumenten.

LyX-Layout

Das für den $\text{\LaTeX}$ -Anfänger größte Problem liegt sehr häufig in der Erkenntnis, dass das Layout des $\text{\LaTeX}$ -Fensters überhaupt nichts mit dem späteren $\text{\LaTeX}$ -Layout zu tun haben muss, was schon daran zu erkennen ist, dass die Bildschirmausgabe über keinerlei Trennungsmöglichkeiten verfügt (Abbildung 2). Innerhalb der Wysiwy-Philosophie war man dennoch bestrebt, viele Ähnlichkeiten zum späteren Endlayout zu erreichen. Die $\text{\LaTeX}$ -Layouts legen lediglich fest, wie etwas auf dem Bildschirm erscheinen soll und welches das dazugehörige

$\TeX$ -Layout ist. Für das Absatzlayout *Title* sieht das entsprechende LyX-Layout wie folgt aus:

Listing 1: Titel-Layout

```
# LyX Title-Layout
Style Title
  Margin          Static
  LatexType       Command
  InTitle         1
  LatexName       title
  ParSkip         0.4
  ItemSep         0
  TopSep          0
  BottomSep       1
  ParSep          1
  Align           Center
  AlignPossible   Center
  LabelType       No_Label
# standard font definition
Font
  Family          Sans
  Series          Bold
  Size            Largest
EndFont
End #title
```

Hierbei stellt `LatexName title` den direkten Zusammenhang zu $\TeX$ her, denn dadurch wird festgelegt, was mit dem Text passieren soll, der im LyX-Fenster das Absatzlayout *title* aufweist (vgl. Abbildung 5). Andererseits kann man seine eigenen Layouts schreiben und umfangreichen $\TeX$ -Code dabei berücksichtigen, wie das folgende Layout zeigt, welches ein Absatzlayout mit einem grauen Hintergrund definiert:

Listing 2: Color-Umgebung

```
#
# Standard color-style definition
#
Style myStandardColor
  Margin          Static
  LatexType       Environment
  LatexName       cminipage
  ParIndent       MM
  ParSkip         0.4
  Align           Block
```

```

AlignPossible      Block, Left, Right, Center
LabelType          No_Label
# standard font definition
Font
  color             blue
  family            typewriter
EndFont
Preamble
\usepackage{color,calc}
\definecolor{myColor}{rgb}{0.9,0.9,0.9}%    rgb color model
\newenvironment{cminipage}{%
  \noindent%
  \begin{lrbox}{\@tempboxa}%
  \begin{minipage}{\linewidth-2\fbboxsep}%
    \begin{list}{}{}% LyXcode
      \setlength{\rightmargin}{\leftmargin}
      \raggedright
      \setlength{\itemsep}{0pt}
      \setlength{\parsep}{0pt}
      \ttfamily}%
    \item[]}%
  {\end{list}}%
  \end{minipage}%
\end{lrbox}%
\fcolorbox{myColor}{myColor}{\usebox{\@tempboxa}}%
}%
EndPreamble
End # myStandardColor

```

Intern wird der LyX-Layoutname Standard-Color verwendet, der dem ebenfalls selbstdefinierten $\TeX$ -Layout cminipage entspricht.

Der Start

Die grafische Oberfläche zeigt sich für heutige Verhältnisse eher spartanisch in ihrer Aufmachung (Abbildung 2), enthält aber alle wesentlichen Elemente, die mehr oder weniger zu jeder Textverarbeitung gehören und in Tabelle 1 zusammengefasst sind. Dem Benutzer sind hierbei keine Einschränkungen vorgegeben, denn er kann diese Leiste nach Belieben seinen eigenen Vorstellungen anpassen. Einige zusätzliche Symbole sind bereits in dem Standardsystem dabei, können aber auch selbst erstellt und natürlich der LyX-Community zur Verfügung gestellt werden (siehe Beispiele am Ende des Artikels).

LyX hat keine eigenen Formatierungseigenschaften, d. h. baut vollständig auf $\TeX$ auf. Somit kann LyX auch nur seine Arbeit beginnen, wenn als Grundvoraussetzung eine $\TeX$ -Dokumentenklasse ausgewählt wurde.

Symbol	Bedeutung
	Dokument öffnen
	Speichern
	Drucken
	Ausschneiden
	Kopieren
	Einfügen
	Italic
	Kapitälchen
	frei belegbar
	TeX-Mode
	Mathematischer Modus
	Fußnote einfügen
	Randnotiz einfügen
	Umgebungstiefe ändern
	Grafik einfügen
	Tabelle einfügen

Tabelle 1: Bedeutung der Buttons

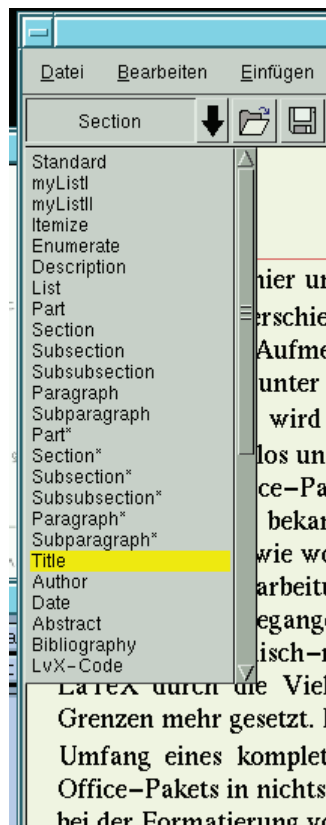


Abbildung 5: Das LyX-Layout-Menü

Die Zahl der zur Verfügung stehenden Klassen hält sich in Grenzen, da in diesen nur die grundsätzlichen Dinge festgelegt werden. Im Wesentlichen gibt es folgende Klassen:

- article
 - ▷ article (IEEEtran)
 - ▷ article ($\mathcal{AMS}$)
 - ▷ article (Springer)
 - ▷ article (Koma-Script)
 - ▷ article (exam)

- ▷ article (Hebrew)
- ▷ article (dtk)
- book
 - ▷ book ($\mathcal{AMS}$)
 - ▷ book (Koma-Script)
- report
 - ▷ report (Koma-Script)
- letter
 - ▷ letter (g-Brief) german
 - ▷ letter (g-Brief) french
 - ▷ letter (german)
 - ▷ letter (Koma-Script)
- seminar
- slide
- APA style
- CD-Box Cover

Standardmäßig wählt LyX die Klasse `article`, wenn nichts anderes angegeben wird. Grundsätzlich kann sich jeder Anwender seine eigenen Klassen schreiben, da $\TeX$ als offenes System praktisch keine Grenzen hat [2, 1]. So haben viele Universitäten ihre eigenen Klassen für Dissertationen, was vor allem in den USA üblich ist, oder einzelne Institute für ihre Diplomarbeiten. Will man eine derartige neue $\TeX$ -Klasse für LyX zur Verfügung stellen, so sollte zuerst ein entsprechendes LyX-Layout (siehe Seite 29) erstellt werden.

Textparameter

Wie bei jeder Textverarbeitung sind auch bei $\TeX$ -Anwendungen global wirksame Parameter festzulegen, von denen hier nur einige exemplarisch angegeben werden:

- ein-/zweispaltig,
- Papiergröße,
- Seitenränder,
- Kopfzeilen,
- Hauptsprache des Dokuments,
- Art der Anführungsstriche,
- usw.

$$a = \arg(z) = \begin{cases} 0 & (a < 0 \wedge b = 0) \\ \pi & (a < 0 \wedge b = 0) \\ \arctan \frac{\operatorname{Im}(z)}{\operatorname{Re}(z)} & (a \in \mathbb{R} \wedge b > 0) \\ \arctan \frac{\operatorname{Im}(z)}{\operatorname{Re}(z)} & (a \in \mathbb{R} \wedge b < 0) \end{cases} \quad (\#)$$

Abbildung 6: LyX im Formelmodus

Durch die Verwendung von PostScript als Standardausgabeformat kann die Druckvorlage auch noch nachträglich in jeder beliebigen Weise verändert werden. So kann beispielsweise ein erstelltes Buch-Cover von der genormten Größe 37,1cm × 23cm für eine Kontrollausgabe mit dem folgenden Befehl im DIN-A4-Format auf dem Drucker ausgegeben werden:

```
psresize -W37.1cm -H23cm -pa4 Cover.ps | lpr
```

Der Formelgenerator

Es wurde bereits auf die hervorragende Qualität des Formelsatzes von $\text{\TeX}$ verwiesen. LyX unterstützt dies in einem grafischen Modus, der nahezu Wysiwyg-Qualitäten erreicht (Abbildung 6). Man vergleiche auch das entsprechende Beispiel im letzten Abschnitt des zweiten Teils.

$$a = \arg(z) = \begin{cases} 0 & (a < 0 \wedge b = 0) \\ \pi & (a < 0 \wedge b = 0) \\ \arctan \frac{\operatorname{Im}(z)}{\operatorname{Re}(z)} & (a \in \mathbb{R} \wedge b > 0) \\ \arctan \frac{\operatorname{Im}(z)}{\operatorname{Re}(z)} & (a \in \mathbb{R} \wedge b < 0) \end{cases} \quad (1)$$

Im Gegensatz zu dem ebenfalls relativ komfortablen Mathematikeditor von WinWord existieren unter $\text{\TeX}$ wesentlich mehr vertikale und horizontale Möglichkeiten der Ausrichtung, wohingegen WinWord im Gegensatz zu LyX das bessere Auswahlmenü im Editor hat. Das LyX-Math-Panel hat einen mehr als spartanischen Charakter, wie Abbildung 7 hinreichend verdeutlicht.

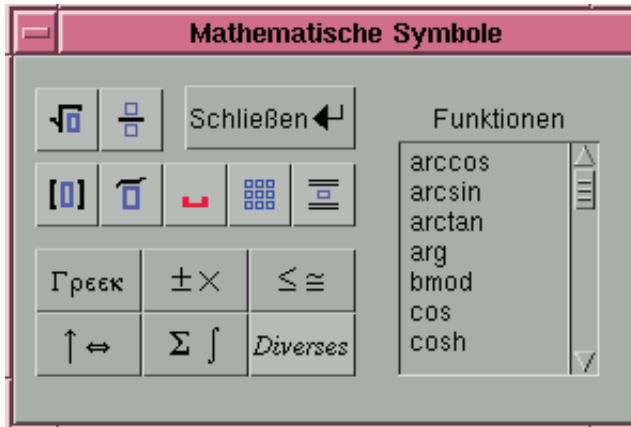


Abbildung 7: LyX-Math-Panel

Das von der *American Mathematical Society* unterstützte Paket *amsmath* bietet umfangreiche Funktionen in Zusammenhang mit dem Formelsatz an, von denen LyX in der kommenden Version wesentliche berücksichtigen wird [1]. Unterschieden wird immer zwischen einem Inline-Modus und einer abgesetzten Formel (Display-Math). Im ersten Fall ist der gesamte mathematische Ausdruck Teil einer regulären Zeile, im zweiten Fall ein eigener Absatz mit der Möglichkeit der Gleichungsnummerierung. Dabei ist ein Wechsel zwischen diesen beiden Modi für einen mathematischen Ausdruck per Tastenbefehl möglich, wobei automatisch eine Anpassung der Anordnung und Größe von Indizes und Exponenten erfolgt.

Als Beispiel betrachte man den folgenden mathematischen Ausdruck, der einmal direkt in der aktuellen Zeile ($\prod_{i=1}^{n-1} e^{-iq_2} = e^{-\frac{n(n-1)}{2}q_2}$) und zum anderen als abgesetzte Formel geschrieben wurde [5].

$$\prod_{i=1}^{n-1} e^{-iq_2} = e^{-\frac{n(n-1)}{2}q_2} \quad (2)$$

Fließobjekte (Floats)

Per Definition können Abbildungen und Tabellen in sogenannte Fließobjekte eingebunden werden, die dann von $\text{\TeX}$ in günstige Positionen gesetzt werden. Dies unterliegt gewissen Kriterien, wie

- minimaler Textanteil auf einer Seite,
- maximale Zahl von Floats auf einer Seite,
- Abstand zwischen Floats auf einer Seite,
- Abstand zwischen Float und Text.

Es sind in jedem Fall erheblich mehr Steuerungsmöglichkeiten vorhanden als sonst bei Textverarbeitungen üblich. Probleme gibt es lediglich im Spaltenmodus, wenn mehr als zwei Spalten benötigt werden. In diesem Fall muss man sich eigene „Float“-Umgebungen definieren. Zu Problemen kommt es immer dann, wenn man sehr viele Fließumgebungen innerhalb eines Kapitels hat, denn $\text{\TeX}$ setzt erst den gesamten Text und versucht dann die fließenden Objekte möglichst sinnvoll einzufügen. Dazu müssen diese natürlich erst einmal zwischengespeichert werden, was zu Speicherproblemen führen kann. Hier ist es dann notwendig, manuell einzugreifen.

Neben den standardmäßig vorgegeben Objekttypen Bild und Tabelle (ein- oder zweispaltig) gibt es unter LyX noch den Algorithmus als vordefiniertes Fließobjekt. Dem Anwender sind jedoch grundsätzlich keine Grenzen in der Definition neuer, eigener Objekte gesetzt. Für sämtliche Fließobjekte existiert ein Befehl `Listof...`, der ein Verzeichnis gleicher Fließobjekttypen an beliebiger Stelle des Textes ausgeben kann.

Formal sind Float-Objekte für LyX Insets und erscheinen alternativ im geschlossenen oder offenen Zustand. Im ersten Fall erscheint im Text nur ein kurzes Symbol, welches Auskunft über die Art des Objekts gibt, während der zweite Fall der Standard für eine Bearbeitung des Floats darstellt. Das Öffnen und Schließen eines Float-Objekts geschieht durch einfaches Anklicken des Symbols oder für alle Objekte über einen Menüpunkt.

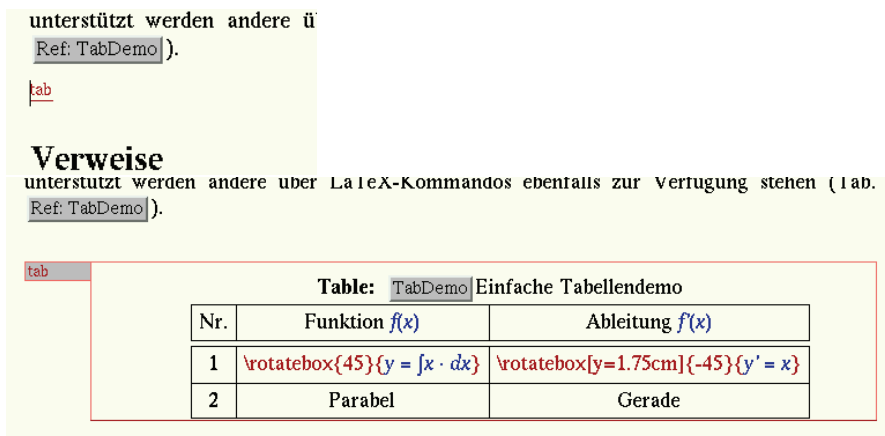


Abbildung 8: Float-Objekt (entspricht Tabelle 2) im geschlossenen und geöffneten Zustand

Bilder und Tabellen

In der Regel befinden sich diese innerhalb von Floats, denn ihre Anordnung beeinflusst wesentlich das Layout des Dokuments. Grundsätzlich kann man sie auch unabhängig davon in den Text einfügen. Sämtliche Grafiken müssen entweder das PostScript-Format (Suffix „ps“) oder das Encapsulated-PostScript-Format (Suffix „eps“) aufweisen. Da dieses Format grundsätzlich völlig ungeeignet für „real pictures“ ist, sind die Dateien schon mal mehrere Megabyte groß. Es empfiehlt sich daher, mit den „gezippten“ Dateiformaten zu arbeiten, was allerdings voraussetzt, dass man eine weitere Hilfsdatei von wenigen Byte erzeugt, die die Abmessungen der sogenannten Bounding-Box enthält:

```
%%BoundingBox: 0 0 376 166
```

Damit ist $\text{\LaTeX}$ in der Lage, den von der eingebundenen Grafik benötigten Platz bei der Formatierung zu berücksichtigen. Das „Entzippen“ der Grafik wird automatisch mit Hilfe des Programmes gunzip durchgeführt.

Nr.	Funktion $f(x)$	Ableitung $f'(x)$
1	$y = \int x \cdot dx$	$\frac{d}{dx} \int x \cdot dx = x$
2	Parabel	Gerade

Tabelle 2: Einfache Tabellendemo

Für das Einfügen von Tabellen gibt es mehrere umfangreiche Pakete, die eine Erweiterung der standardmäßigen Unterstützung von $\LaTeX$ vornehmen, wobei einige direkt von LyX unterstützt werden, andere über $\LaTeX$ -Kommandos ebenfalls zur Verfügung stehen (Tabelle 2).

Verweise

Diese sind auf vielfältige Art möglich, unterscheiden sich aber nicht prinzipiell von denen anderer Textverarbeitungsprogramme, sodass hier darauf nicht weiter eingegangen werden muss. Das $\LaTeX$ -Paket `hyperref` unterstützt die Markierung und Hervorhebung derartiger Verweise als Link für die Umwandlung in eine PDF-Datei. Dieses muss in der derzeitigen Version noch über die $\LaTeX$ -Präambel (vergleiche Teil 2) eingebunden werden, denn LyX unterstützt momentan nur das Paket `prettyref`, sodass man per Menü die Art des Querverweises bestimmen kann (Abbildung 9). Für die Version 1.2.0 ist jedoch eine vollständige Integration des Pakets `hyperref` vorgesehen.

Verweise beziehen sich grundsätzlich auf sogenannte referenzierbare Label, die an jeder beliebigen Stelle des Dokuments mit einem frei wählbaren Namen eingetragen werden können. Da dies bei umfangreichen Texten ohne weiteres mehr als 100 Labels sein können, bietet LyX hier eine Unterstützung, indem es beispielsweise für Überschriften-Labels automatisch den Typ („sec:“ für Section) vor die eigentliche Marke setzt.

LyX kann die Labels nach ihrem Erscheinen im laufenden Text oder aber sortiert anzeigen. Auch Verweise auf Fußnoten sind kein

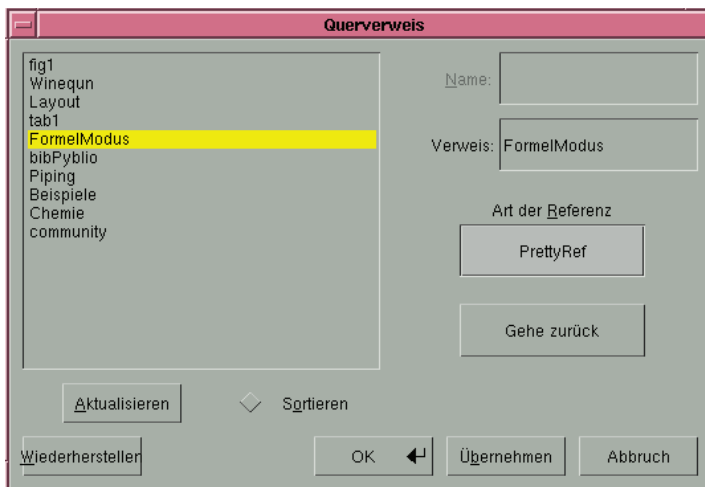


Abbildung 9: Menü zum Eintragen eines Verweises

Problem, müssen jedoch analog zu $\text{\LaTeX}$ mit einem kleinen Trick „erzwungen“ werden.

Minipages

Minipages bilden innerhalb von $\text{\LaTeX}$ eine wesentliche Voraussetzung für diffizile Layout-Fragen und werden daher auch von LyX direkt unterstützt. Die Minipages, als „Seite in einer Seite“ sind prinzipiell eigenständige Dokumente mit all den Eigenschaften, die eben für einzelne Seiten gelten. Andererseits liegt damit fest, dass eine Minipage nicht größer als eine Seite werden kann und auch vollständig in dieser liegen muss, also keinen Seitenumbruch aufweisen kann.

Typische Anwendungsfälle für Minipages sind das Nebeneinandersetzen zweier Objekte (Abbildungen, Tabellen, Algorithmen etc.) mit eigenen, fortlaufend nummerierten Bildunterschriften in einer Fließumgebung.

Dies¹ soll jetzt an einem willkürlichen Beispiel gezeigt werden, bei dem dieser Text wiederholt in der Minipage² erscheint, wobei eine Breite von 45% für die linke und 40% für die rechte Minipage vorgesehen wurden. Auch das Einfügen von Formeln ist kein Problem:

$$f(n) = \prod_{i=1}^{n-1} e^{-iq_2} \quad (3)$$

Und Fußnoten sind auch möglich!

¹ Initialen lassen sich leicht mit dem Paket `dropping` erstellen.

² Eine Minipage kann nicht größer als eine Seite selbst sein und darf keinen Seitenumbruch aufweisen!

Dies¹ soll jetzt an einem willkürlichen Beispiel gezeigt werden, bei dem dieser Text wiederholt in der Minipage² erscheint, wobei eine Breite von 45% für die linke und 40% für die rechte Minipage vorgesehen wurden. Auch das Einfügen von Formeln ist kein Problem:

$$f(n) = \prod_{i=1}^{n-1} e^{-iq_2} \quad (4)$$

Und Fußnoten sind auch möglich!

¹ Initialen lassen sich leicht mit dem Paket `dropping` erstellen.

² Eine Minipage kann nicht größer als eine Seite selbst sein und darf keinen Seitenumbruch aufweisen!

Eingerichtet werden Minipages über `Layout>Absatzformat>Extras`, was sich allerdings in der neuen LyX-Version 1.2 etwas ändern wird, sodass weitere Optionen im Zusammenhang mit Minipages erleichtert werden. Abbildung 10 zeigt die in der aktuellen Version möglichen Einstellungen für Minipages. In dem hier vorliegenden Beispiel wurde `\hfill` zwischen einzelne Absätze eingefügt, so dass die linke Minipage linksbündig und die rechte eben rechtsbündig erscheint.

Eine Minipage kann natürlich auch beliebig rotiert werden, was durch den ~~TeX~~TeX-Befehl `\rotatebox` unterstützt wird.

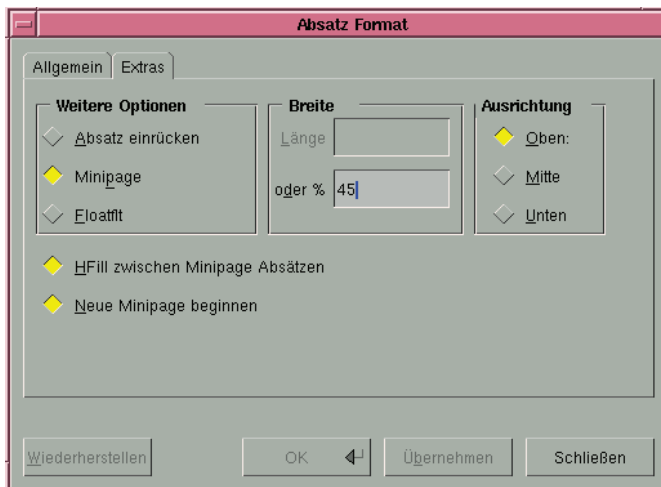


Abbildung 10: Absatzlayout für Minipages

Dies¹ soll jetzt an einem willkürlichen Beispiel gezeigt werden, bei dem dieser Text wiederholt in der Minipage² erscheint, wobei eine Breite von 45% für die linke und 40% für die rechte Minipage vorgesehen wurden. Auch das Einfügen von Formeln ist kein Problem:

$$f(n) = \prod_{i=1}^{n-1} e^{-iq_2} (5)$$

Und Fußnoten sind auch möglich!

¹Initialen lassen sich leicht mit dem Paket **dropping** erstellen.

²Eine Minipage kann nicht größer als eine Seite selbst sein und darf keinen Seitenbruch aufweisen!

Weitere Möglichkeiten, insbesondere hinsichtlich der vertikalen Positionierung von Minipages, kann man [3, 2] entnehmen.

Verteilte Dokumente

Das Schreiben von Büchern kann sehr schnell eine Frage der Übersicht werden, wenn mehrere hundert Seiten zu organisieren sind. Hier bietet LyX das Arbeiten mit verteilten Dokumenten an, sodass das

gesamte Dokument in beliebig kleine Einheiten unterteilt werden kann.

Der Aufbau des Hauptdokuments könnte dann entsprechend Abbildung 11 aussehen. Es enthält nur die Angaben über die einzufügenden Kapitel und Verzeichnisse (Inhalt, Index), sowie Informationen über die Art der Seitenzahlen, die im Bereich `\frontmatter` (Titel, Vorwort,...) eine römische Seitennummerierung, im Bereich `\mainmatter` und `\backmatter` eine arabische Nummerierung aufweisen. Das Hauptdokument muss selbst also keinerlei Text aufweisen.

LyX öffnet spätestens bei der Übersetzung des Dokuments automatisch alle benötigten Dateien. Dies geschieht aber auch schon dann, wenn man versucht, einen Querverweis von einem Teildokument in ein anderes einzufügen, denn LyX erstellt aus allen verteilten Dokumenten ein gemeinsames Label-Verzeichnis, sodass Querverweise zwischen verteilten Dokumenten keinerlei Probleme bereiten.

Bibliographie

Dies stellt wieder einen wichtigen Schwerpunkt von $\TeX$ dar, denn es existieren derartig viele Software-Pakete, die alle möglichen Variationen und spezielle Sonderwünsche unterstützen. Das völlig eigenständig ablaufende Programm $\BibTeX$ setzt prinzipiell keine Grenzen in der Form der Zitate, da diese durch ein eigenes Style-File festgelegt sind, welches vom Anwender beliebig erweitert werden kann. LyX überwacht den gesamten Mechanismus und startet bei Bedarf von sich aus das $\BibTeX$ -Programm; der Anwender kann sich um andere Dinge kümmern. Das Einfügen der Literaturzitate erfolgt über ein Menü, welches sämtliche in einer angegebenen Literaturdatenbank oder im Text durch ein Schlüsselwort festgelegte Literaturangaben enthält (Abbildung 12).

Grundsätzlich kann man sich eine einzige Literaturdatenbank anlegen, in die *sämtliche* Literatur wie Bücher, Zeitschriftenartikel, Tagungsberichte, Technische Berichte, Sammlungsbände usw. eingetragen werden kann und zwar unabhängig vom Fachgebiet. Der Aufbau eines derartigen Eintrages ist relativ einfach, wie Listing 3 zeigt [3].

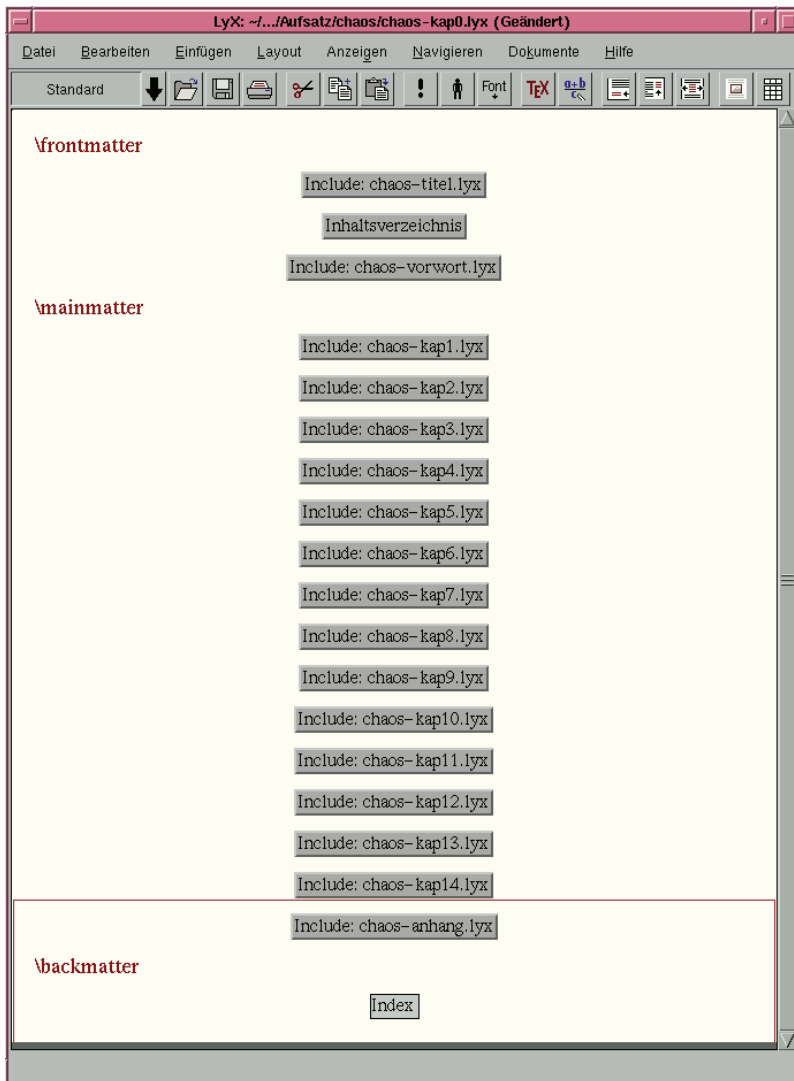


Abbildung 11: Aufbau eines Hauptdokuments der Dokumentenklasse `book` mit den im „Eval Red Text“ angegebenen L^AT_EX-Steuerungskommandos

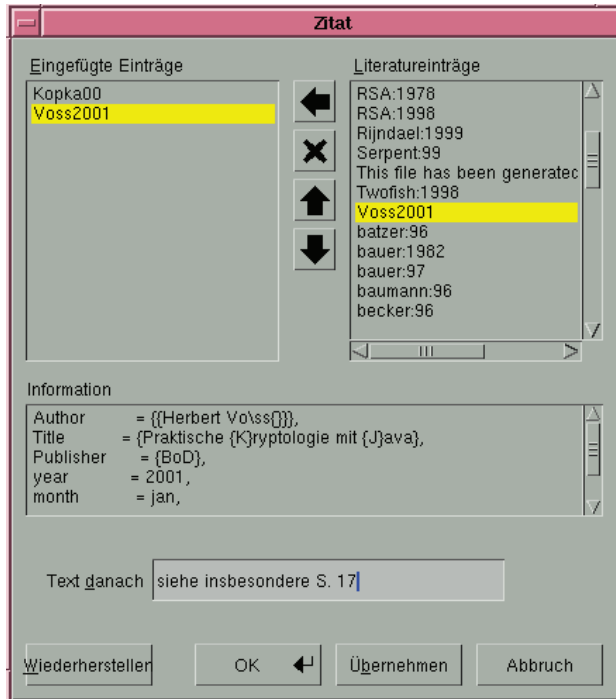
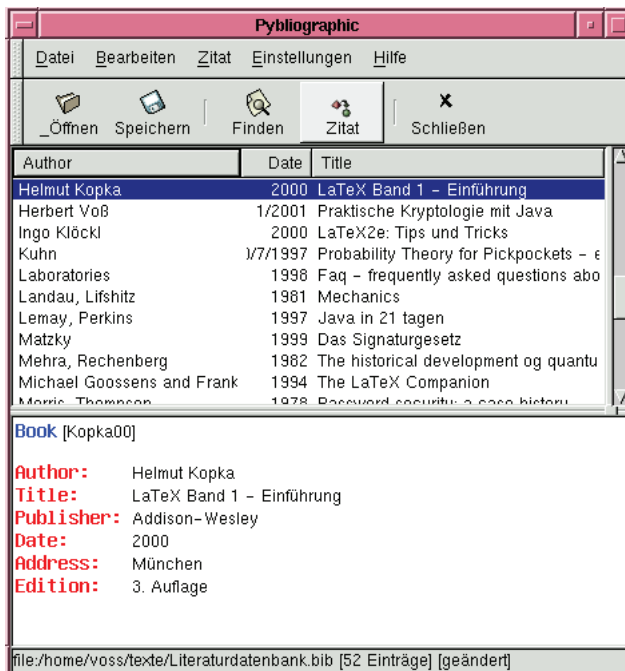


Abbildung 12: Das Menü zum Einfügen von Literaturzitaten

Listing 3: Beispiel-Eintrag aus einer Literaturdatenbank

```
@ARTICLE{article-full,
  author = {L[eslie] A. Lamport},
  title = {The Gnats and Gnus Document Preparation System},
  journal = {\mbox{G-Animal's} Journal},
  year = 1986,
  volume = 41,
  number = 7,
  pages = "73+",
  month = jul,
  note = "This is a full ARTICLE entry",
}
```

Dazu existieren verschiedene Programme, die diese Verwaltung der Datenbank erleichtern. Mit LyX braucht dann nur aus dieser

Abbildung 13: *Pybliographic* zum Verwalten einer Literaturdatenbank

Datenbank der entsprechende Eintrag ausgewählt zu werden, der als Literaturzitat in den laufenden Text eingefügt wird. Insbesondere kann das von der American Mathematical Society herausgegebene Abkürzungsverzeichnis für Namen von Zeitschriften die Tipparbeit erheblich erleichtern. Das Verzeichnis ist Teil einiger $\text{\TeX}$ -Distributionen oder kann von <http://www.ams.org/tex> bezogen werden. Eine sehr gute Web-Seite ist <http://liinwww.ira.uka.de/bibliography/>. Mit LyX ist aber auch ein paralleles Arbeiten mit einem Literaturdatenbankprogramm wie *Pybliographer* (Abbildung 13) möglich, welches das *Piping* unterstützt und so direkt Zitate über den sogenannten LyX-Server in den Text einfügt. Diese Eigenschaft weisen mehrere der gängigen Bib $\text{\TeX}$ -Programme auf, wie TkBib $\text{\TeX}$ oder Sixpack.

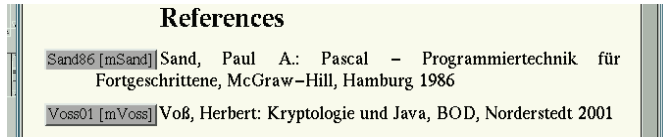


Abbildung 14: Absatzlayout Bibliography

Will man auf eine externe Literaturdatenbank verzichten, so kann man mit dem direkt von LyX unterstützten Bibliography-Layout arbeiten, wobei die Schlüssel sozusagen „online“ vergeben werden (Abbildung 14). Diese Vorgehensweise ist aber ausdrücklich nicht zu empfehlen, denn derartige Literaturstellen können dann nur noch mit Hilfe der „Drag and Drop“-Methode in andere Dokumente eingefügt werden. Dazu muss man aber in jedem Fall immer den Namen des alten Dokuments parat haben.

Zusammenfassung

In diesem ersten Teil wurden die wesentlichen Eigenschaften von LyX gezeigt, die insbesondere Anfängern den Einstieg in das professionelle $\text{\LaTeX}$ -System erleichtern soll. So ist das Erstellen einfacher Texte ohne jegliche Kenntnis von $\text{\LaTeX}$ möglich. Im zweiten Teil werden dann insbesondere die Rechtschreibkorrektur, Ausgabemöglichkeiten sowie Beispiele behandelt.

Literatur

- [1] Michael Goossens, Frank Mittelbach und Alexander Samarin: *The $\text{\LaTeX}$ Companion*; Addison Wesley; 1994.
- [2] Ingo Klöckl: *$\text{\LaTeX}2\epsilon$: Tips und Tricks*; Heidelberg; dpunkt Verlag; 2000.
- [3] Helmut Kopka: *$\text{\LaTeX}$ Band 1 - Einführung*; Addison-Wesley; München; 3. Aufl.; 2000.
- [4] Leslie Lamport: *Das $\text{\LaTeX}$ Handbuch*; Addison-Wesley; München; 1995.

- [5] Herbert Voß: *Ly_X and Mathmode*; <http://www.perce.de/lyx/equations.pdf>; Mai 2001.

„Poligraf“ oder „Zwischen T_EX und der Druckerei“ -- der zweite Versuch

Janusz Marian Nowacki

Während der IV. Konferenz „BachoT_EX ’96“ der polnischen T_EX-Benutzergruppe habe ich zum ersten Mal das Makro-Paket POLIGRAF vorgestellt. Jetzt möchte ich eine neue Version veröffentlichen, in die Rückmeldungen und Verbesserungsvorschläge eingeflossen sind.

Einführung

Ich war sehr erfreut darüber, dass POLIGRAF auf großes Interesse bei vielen T_EX-Benutzern stieß, die sich mit Textsatz und den Bedürfnissen des Mehrfarbdrucks beschäftigen. Leider muss ich gestehen, dass die Version von 1996 noch zahlreiche Mängel aufwies.

POLIGRAF funktionierte nur mit Plain-T_EX, das ich selbst benutze, korrekt und in vollem Umfang. Obwohl M_T_EX-Benutzer die Mehrheit der T_EX-Benutzer stellen, haben meine fehlenden M_T_EX-Kenntnisse dazu geführt, dass diese Gruppe bei der Verwendung auf Schwierigkeiten gestoßen ist.

Desweiteren wollte ich zu viele Probleme auf einmal lösen. Beispielsweise war es nicht nötig, sich mit der Seitenmontage auf dem Druckbogen zu beschäftigen, da ich mittlerweile weiß, dass es hierfür bessere Methoden gibt.

Schließlich habe ich den Mechanismus der Farbseparation zu sehr ausgebaut. Das hatte zur Folge, dass er zu kompliziert wurde. Inzwischen ist, parallel zu meiner Arbeit, von den Kollegen aus der Firma BOP in Gdańsk das Paket cmyk-hax.tex entwickelt worden.

Ich bin zu dem Schluss gekommen, dass diese Lösung die bessere zur Farbseparation ist.

Teile des Pakets

Zur Zeit besteht das Paket POLIGRAF aus drei Dateien¹:

- `poligraf.sty` - die Paketdatei mit allen Makros,
- `crops.pro` - Header-Datei für `dvips`,
- `separate.pro` - optionale Header-Datei für `dvips`.

Das Paket finden Sie auf dem FTP-Server `ftp.gust.org.pl` im Verzeichnis `GUST/contrib/poligraf.zip` und auf der CD-ROM „T_EX Live 6“.

In der Praxis reicht die Datei `crops.pro`

Die meisten Dokumente, die mit T_EX oder anderen DTP-Systemen gesetzt werden, werden auf Laserdruckern ausgegeben. Viele Publikationen werden aber wegen der Auflagenhöhe oder der Qualität in professionelle Offset-Druckereien gegeben. Zu diesem Zweck sollen die Seiten des Dokumentes oder die Druckbögen zusätzliche Elemente enthalten, die von den Druckereien verlangt werden. Hierzu gehören beispielsweise: Passmarken (registration marks), Schnittmarken (crop marks), Farbsättigungsstufen (color steps) und Farbbalken (color bars).

In der neuen Version des POLIGRAF-Paketes wird diese Aufgabe von der Header-Datei `crops.pro` erfüllt, die beim Aufruf von `dvips`² angegeben werden muss.

Die mit einem beliebigen T_EX-Format erzeugte Datei `foo.dvi` wird mit folgendem Aufruf

```
dvips -h crops.pro foo.dvi
```

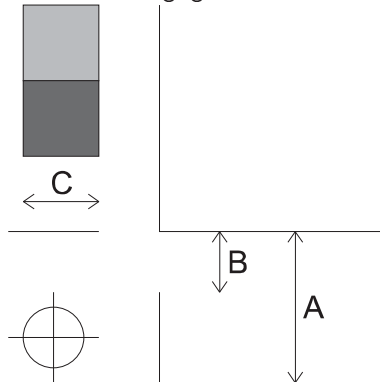
in die PostScript-Datei `foo.ps` übersetzt, die alle notwendigen Elemente enthält.

¹ Zur Farbseparation ist zusätzlich das Paket `cmk-hax.tex` notwendig.

² Die Header-Dateien `crop.pro` bzw. `crop.lpr` werden zusammen mit dem Programm `dvips` verteilt. Leider stellen diese Dateien nur einen Teil der benötigten Elemente zur Verfügung.

Der Benutzer hat die Möglichkeit, mit Hilfe des POLIGRAF-Pakets die in der Datei crops.pro voreingestellten Werte der folgenden Parameter zu ändern.

- cropmarksize - mit dieser Größe wird die Seitenlänge (A) des Quadrats in Millimeter angegeben, in das die Schnittmarke gesetzt wird. Der Vorgabewert beträgt 10mm.
- cropmarkdistance - der Abstand der Schnittmarken von der Seitenfläche in Millimeter. Dieser Wert bestimmt zusammen mit dem vorherigen Parameter gleichzeitig die Größe der eingeschriebenen Passmarke. Der Vorgabewert beträgt 3mm.
- barsize - die Breite (C) der quadratischen Flächen in Millimeter, aus denen die Farbsättigungsstufen und -balken zusammengesetzt sind. Hierfür sind 5mm vorgegeben.



- colorbars - Schalter zur Steuerung von Farbsättigungsstufen und -balken:
 - ▷ 0 - ohne Farbsättigungsstufen und -balken,
 - ▷ 1 - mit Farbsättigungsstufen und -balken,
 - ▷ 2 - nur mit Farbsättigungsstufen,
 - ▷ 3 - nur mit Farbbalken.
- mirror - Spiegelung der gesamten Seiten einschließlich der Pass- und Schnittmarken sowie der Farbsättigungsstufen und

-balken. Wird dieser Schalter benutzt, dürfen keine anderen Methoden der Seitenspiegelung verwendet werden.

- `labeloff` - unterdrückt die Ausgabe der T_EX-Seitennummer, der Nummer der PostScript-Seite und des Namens der separierten Farbe am unteren Seitenrand.
- `xoffset` und `yoffset` - verschiebt die ganze Seite einschließlich der Pass- und Schnittmarken sowie der Farbsättigungsstufen und -balken im PostScript-Koordinatensystem.

An dieser Stelle weise ich auf die Notwendigkeit hin, das Papierformat, auf dem die Publikation gedruckt werden soll, in der T_EX-Eingabedatei zu spezifizieren. Vergisst man dies, kann es Probleme verursachen. In der Regel ist für T_EX das Papierformat gleichgültig. Wesentliche Angaben sind die Textbreite, -höhe und die Breite der Ränder. Dagegen muss dvips das Papierformat kennen, um vom T_EX-Koordinatensystem in das PostScript-Koordinatensystem umrechnen zu können. In der T_EX-Eingabedatei ist das Papierformat daher festzulegen.

```
\special{papersize=xmm,ymm}
```

Wenn wir die Papiergröße nicht in der Eingabedatei festlegen, nimmt dvips den in der Datei `config.ps` angegebenen Wert. In Europa ist dies meist ISO/DIN A4, was nicht unbedingt mit unseren Erwartungen übereinstimmen muss.

Selbst wenn die Papiergröße in der Eingabedatei festgelegt wird, kann dvips noch für Überraschungen sorgen. dvips sucht nach einer in `config.ps` definierten Papiergröße und rundet, falls keine passende vorhanden ist, zur nächst passenden Papiergröße mit einer nicht näher festgelegten Genauigkeit. Dieses Verhalten kann man durch Angabe der Option

```
-t unknown
```

```
ausschalten.3
```

³ In einigen, vor allem älteren dvips-Versionen war diese Option nicht korrekt definiert.

Aus dem Inneren einer T_EX-Datei

Die Header-Dateien des dvips-Programms machen uns von den T_EX-Formaten unabhängig, aber sie sind nicht für alle Benutzer ein bequemes Werkzeug. Ein „reinrassiger“ T_EX-Benutzer bevorzugt es, alle Elemente der Publikation durch Deklarationen in der T_EX-Eingabedatei zu steuern. Dies soll das Paket `poligraf.sty` ermöglichen. Die Namenserverweiterung `*.sty` zeigt, dass nun L^AT_EX-Benutzer unterstützt werden. Um das neue POLIGRAF zu verwenden, genügt es, folgendes in die Eingabedatei zu schreiben:

```
\input poligraf.sty
```

oder für L^AT_EX

```
\usepackage{poligraf}
```

Dadurch bindet dvips automatisch die zuvor beschriebene Header-Datei `crops.pro` in die erzeugte PostScript-Datei ein. Zusätzlich bietet es uns die Möglichkeit, die Vorgabewerte durch gleichnamige T_EX-Befehle zu ändern. Hiermit werden die in der Datei `crops.pro` enthaltenen Werte überschrieben.

Die Verwendung von `poligraf.sty` ermöglicht es uns damit, nur die T_EX-Eingabedatei zu editieren. Darüberhinaus haben wir die einzelnen Parameter der Pass- und Schnittmarken sowie der Farbsättigungs- und -balken zusammen mit dem gesamten Dokument gespeichert, sodass das Dokument in der Zukunft immer gleich aussehen wird.

Die Farbseparation

Für die Bedürfnisse des Mehrfarbdrucks soll die Publikation in die im Druckergewerbe verwendeten Grundfarben, d. h. Cyan, Magenta, Gelb und Schwarz zerlegt werden.

Normalerweise wird diese Aufgabe in der Druckerei erfüllt. Viele Benutzer wollen jedoch die Separation selbst durchführen. Ein Grund kann darin liegen, dass sie die erhaltenen Effekte noch vor der Belichtung unter Kontrolle haben wollen (ohne die Kosten zu tragen). Ein anderer Grund wäre das eigene Erzeugen von Druckvorlagen mit Hilfe eines Laserdruckers und handelsüblichen Kopierfolien.

Wie ich schon erwähnt habe, können wir die Separation durchführen, indem wir das bewährte Makro-Paket `cmk-hax.tex` mit seinen beeindruckenden Möglichkeiten benutzen. Mit Hilfe dieses Pakets kann man nicht nur die Separation durchführen, sondern auch die Farben einzelner Objekte verändern.

Für die Farbseparation stellt `poligraf.sty` folgende Befehle zur Verfügung:

- `\separate\CYAN` - Erzeugen des Cyan-Auszuges,
- `\separate\MAGENTA` - Erzeugen des Magenta-Auszuges,
- `\separate\YELLOW` - Erzeugen des Gelb-Auszuges,
- `\separate\BLACK` - Erzeugen des Schwarz-Auszuges,
- `\NoOverPrintBlack` - standardmäßig druckt POLIGRAF die schwarze Farbe über die vorher gedruckten Farben, sodass an diesen Stellen in den Farbauszügen die jeweiliger Farbe in voller Sättigung vorhanden ist. Dieser Befehl schaltet dieses Verhalten ab, sodass die Farbauszüge an diesen Stellen frei bleiben.

Benutzen wir den Befehl `\separate`, liest POLIGRAF automatisch das Paket `cmk-hax.tex` ein.

Farbseparation per Kommandozeile

Die Farbseparation können wir auch mit der Header-Datei `separate.pro` durchführen. Diese Datei wurde aus `cmk-hax.tex` erzeugt. Sie enthält einige Parameter, mit denen wir Einfluss auf die Funktionen nehmen können. Unter anderem können wir die Farbe des zu erwartenden Farbauszuges festlegen. Mit dem folgenden Befehl erhalten wir die PostScript-Datei mit dem gewünschten Farbauszug:

```
dvips -h separate.pro foo.dvi
```

Wenn wir zusätzlich die Pass- und Schnittmarken sowie die Farbsättigungs- und -balken wünschen, benutzen wir den folgenden Befehl:

```
dvips -h separate.pro -h crops.pro foo.dvi
```

wobei wir darauf achten müssen, die beiden Header-Dateien in der gezeigten Reihenfolge anzugeben.

Um alle benötigten Farbauszüge zu erzeugen, ist die Datei `separate.pro` mehrmals zu editieren. Eine bequemere Lösung bieten die vier fertigen Separationsdateien: `cyan.pro`, `magenta.pro`, `yellow.pro` und `black.pro`. Mit einer Skript- oder `*.bat`-Datei kann dieses Verfahren abgekürzt werden.

Dokumente, die nur als `*.dvi`-Dateien vorliegen, können ohne `poligraf.sty` durch Benutzung der Header-Dateien `crops.pro` und `separate.pro` bearbeitet werden. Dies ist auch eine gute, manchmal die einzige Lösung für andere als die oben erwähnten T_EX-Formate. Das Paket `poligraf.sty` kann mit diesen Formaten funktionieren, aber das konnte ich nicht prüfen, weil ich sie nicht kenne.

Was fehlt im neuen Poligraf?

Das grundsätzliche Ziel, das ich bei der Neufassung von POLIGRAF verfolgt habe, war ein leicht zu benutzendes Paket, das mit vielen Formaten funktionieren sollte.

Im neuen POLIGRAF-Paket fehlen die folgenden Befehle der letzten Version:

- `\Language\Polski` und `\Language\English` - weil sie nur vereinzelt verwendet wurden,
- `\Twoside`, `\Landscape`, `\LeftMargin` und `\Topmargin` - weil an ihrer Stelle die Standardlösungen gut verwendbar sind,
- `\Hoffset`, `\Voffset` - weil sie durch `\Xoffset` und `\Yoffset` ersetzt wurden,
- `\ScrAngle`, `\ScrFrequency` und `\Rasterize` - ihre Funktionalität wird durch das eingebundene Paket `cmym-hax.tex` abgedeckt,
- `\Hline`, `\Vline`, `\ShowGrid`, `\Marglines` - diese Informationen können mit Hilfe der Programme wie `psview` und `gsview` ebenso erlangt werden,
- `\beginLocalRaster` und `\endLocalRaster` - gehören nicht zum Aufgabenbereich des POLIGRAF-Paketes.

Danksagung

Ich möchte allen bisherigen POLIGRAF-Benutzern herzlich dafür danken, dass sie mich auf meine Fehler hingewiesen und einige Veränderungen vorgeschlagen haben. Ich bedanke mich auch herzlich für die Hilfe beim Schreiben der neuen Version, die mir Piotr Pijanowski, Piotr Strzelczyk, Marcin Woliński und Staszek Wawrykiewicz geleistet haben.

Ebenso möchte ich mich bei der Übersetzerin Iwona Nowacka und Volker RW Schaa sowie Bernd Raichle bedanken, die mir bei der Erstellung der deutschen Version dieses Artikels halfen.

Indexverzeichnisse mit xindy und xindy.sty – eine Einführung

Christopher Creutzig

Indexverzeichnisse werden beim Arbeiten mit $\text{\TeX}$ traditionell mit `makeindex` gesetzt. Dieses Programm hat leider viele Einschränkungen, die das Erstellen der Indizes kompliziert machen. `xindy` ist ein Programm, das einen Teil der Einschränkungen umgeht. Mit `xindy.sty` gibt es nun auch eine einigermaßen komfortable Möglichkeit, diese Funktionen direkt aus $\text{\LaTeX}$ heraus zu nutzen.

Einleitung

Gute technische Bücher haben gute Indexverzeichnisse, denn diese sind ein Teil eines Buches, der dem Leser hilft, wenn er etwas nachschlagen muss. Über die inhaltlichen Probleme bei der Erstellung eines guten Indexes wollen wir hier gar nicht reden; im Rahmen dieses Artikels soll lediglich eine Möglichkeit aufgezeigt werden, wie ein Index technisch realisiert werden kann.

Dieser Artikel beschreibt nach einer allgemeinen Übersicht hauptsächlich die Installation von `xindy`, die leider (noch?) etwas aufwändiger

ist als die Installation eines $\TeX$ -Pakets. Es folgt ein kurzes Beispiel; tiefergehende Anwendungen bleiben zukünftigen Artikeln vorbehalten.

makeindex und xindy – ein Vergleich

Die traditionelle Methode, einen Index zu erzeugen, besteht darin, die Einträge mit dem Befehl `\index` in den Text zu schreiben, $\TeX$ über den Text laufen zu lassen und anschließend `makeindex` aufzurufen, das eine beim $\TeX$ -Lauf entstandene Datei mit der Endung `.idx` einliest und eine Datei mit der Endung `.ind` generiert, die bei weiteren $\TeX$ -Läufen eingelesen wird. Hierbei werden die Einträge alphabetisch sortiert und gleichlautende Einträge werden zusammengefasst. Die Seitenzahlen werden – wie vom Inhaltsverzeichnis gewohnt – automatisiert gesetzt. Untereinträge, wie „Spielkarten/-mischen sind (bis zu zwei Ebenen tief) möglich.

Mit speziellen Zeichen in der Eingabe ist es auch möglich, die Seitenzahl im fertigen Verzeichnis kursiv, fett, unterstrichen oder sonstwie gesondert zu setzen, Querverweise einzusetzen, z.B.: „DTK: *siehe* $\TeX$ nische Komödie, Die“, oder auch die Sortierung zu beeinflussen. Letzteres ist beispielsweise immer dann nötig, wenn der Indexeintrag Formatierungsanweisungen oder sonstige $\TeX$ -Kommandos enthält, wie in `\texttt{README.TXT}`. Eine weitere wichtige Möglichkeit besteht darin, ganze Abschnitte in einem Indexeintrag zusammenzufassen, die dann als „von-bis“eingetragen werden („Speichern: 7, 10-14, 26“).

`makeindex` kennt vier Sortierreihenfolgen, wovon nur zwei mit Umlauten zurechtkommen. Bei `xindy` hingegen kann der Anwender die Sortierung nahezu beliebig einstellen, womit man auch für andere Sprachen als Deutsch und Englisch ohne größere Verrenkungen einen korrekten Index erhalten kann. Sortierregeln für Englisch, Französisch, Spanisch und Deutsch sind in der Verteilung enthalten. Außerdem kann man `xindy` beibringen, verschiedene Schreibweisen eines Buchstabens (`"s`, `\3`, `\ss{}`, `^~df`) für die Sortierung und das Zusammenfassen gleichlautender Einträge als identisch anzusehen. Mit demselben Mechanismus lässt sich `xindy` dazu bringen, beispielsweise „,

|test|“direkt hinter (oder vor) „test“einzuordnen, ohne dass die Autorin hierfür (wie bei makeindex) einen eigenen „Sortierschlüssel“ an müsste – was natürlich weiterhin möglich ist. Auch Sortierungen wie beispielsweise im Ungarischen, wo es Buchstaben wie „ll“oder „ly“gibt, die für unsereins nach zwei aufeinanderfolgenden Buchstaben aussehen, sind kein Problem – xindy lässt sich mit wenigen Zeilen dazu bringen, die korrekten Abschnitte „... , K, L, Ll, Ly, M, ...“anzulegen.

Für die meisten Anwender sind Seitenzahlen sicherlich genau das Richtige – allerdings hat makeindex mit komplizierteren Seitenzahlen, wie „Kapitel-Seite“, Probleme beim Sortieren. Diese Seitennummerierung ist bei Referenzwerken, bei denen Teile für sich aktualisierbar sein müssen, recht verbreitet. In xindy kann man beliebige „Verweistypen“ (beispielsweise also auf Seitenzahlen, auf Positionen innerhalb der Seite, auf ganze Kapitel oder auf Bibelstellen verweisen; mit der entsprechenden Einstellung sortiert xindy „Genesis 29,19“korrekt vor „Exodus 14,12“ein. Natürlich können verschiedene Verweistypen innerhalb eines Eintrags stehen. Es ist auch möglich, für jeden Verweistyp anzugeben, wann und wie mehrere Einträge zusammengefasst werden sollen. Im Beispiel der Bibelstellen lässt sich so eine Sortierung „Hiob 34,32; Psalm 37,16; 63,4; 69,32; 84,11“erreichen.

xindy installieren

xindy hat Vorteile, in Ordnung. Es hat aber auch einen gravierenden Nachteil gegenüber makeindex: Letzteres ist Bestandteil nahezu jeder $\text{\TeX}$ -Installation, wohingegen xindy im Normalfall erst noch nachinstalliert werden muss. Die beiden folgenden Abschnitte zeigen, wie man das auf (neueren) Windows-Systemen und auf Unix-Derivaten macht. Zusätzlich ist noch xindy.sty zu installieren, den Sie von der URL <ftp://xindy.sourceforge.net/pub/xindy/xindystyle/> bekommen können. Ich kann hier nicht auf die Installation eines $\text{\TeX}$ -Paketes eingehen. Dafür sind die verschiedenen $\text{\TeX}$ -Distributionen zu unterschiedlich.

Windows

Für die Installation unter Windows (für diesen Artikel getestet: Windows 98) benötigt man zunächst einmal Entpacker für .zip-

und .tar.gz-Dateien. Für .zip finden Sie unzip auf der CD-ROM 1 des CTAN-Abzugs von DANTE e.V. in der Datei tools\zip\info-zip\WIN32\unz542xN.exe, gzip in demselben Verzeichnis in der Datei gzip124xN.zip und tar im Verzeichnis tools\tar\msdos. Ich gehe im Folgenden davon aus, dass diese Programme an einem Ort installiert sind, der von der Kommandozeile aus gefunden wird – die Entpack-Schritte lassen sich natürlich auch mit einem der zahlreichen Programme mit grafischer Nutzungsoberfläche durchführen.

Zunächst entpackt man von der CD-ROM 1 oder dem CTAN-Server aus dem Verzeichnis indexing\xindy die Datei xdydos20.zip, und zwar direkt an den endgültigen Ort. Von der Kommandozeile aus dient dazu der Befehl unzip F:\indexing\xindy\xdydos20.zip, wenn Ihr CD-ROM-Laufwerk den Laufwerksbuchstaben F: hat. Hierbei wird automatisch ein Verzeichnis mit dem Namen xindydos angelegt. In diesem Verzeichnis liegt nach dem Entpacken ein weiteres Archiv mit dem Namen emxrsx.zip, das Sie direkt vor Ort auspacken müssen (cd xindydos und pkunzip emxsrc). Anschließend können Sie dieses Archiv bedenkenlos löschen; es wird nicht mehr benötigt.

Als zweiter Schritt muss aus demselben Verzeichnis der CD-ROM beziehungsweise des FTP-Servers die Datei xindy-2.0d.tar.gz entpackt werden. Der Ort ist wiederum beliebig, in dieser Datei finden Sie die systemunabhängigen Teile von xindy. Zum Entpacken kopieren Sie die Datei zunächst in ein Verzeichnis Ihrer Wahl, es wird später von dort aus ein Unterverzeichnis xindy\xindy-2.0d angelegt werden, und rufen Sie gunzip xindy-2.0d.tar.gz, gefolgt von tar xvf xindy-2.0d.tar auf. Anschließend können Sie xindy-2.0d.tar wieder löschen.

Als dritten Schritt müssen Sie nun noch in dem Verzeichnis, das Sie im ersten Schritt angelegt haben, zwei Dateien editieren: xindy.bat und makindx4.bat. Sie enthalten Pfadangaben, die vermutlich auf Ihrem System nicht stimmen. Diese sind glücklicherweise leicht zu finden: Ersetzen Sie einfach alle F: durch den Pfad, unter dem Sie im ersten Schritt das Verzeichnis xindydos angelegt haben. Zusätzlich sollten Sie in xindy.bat eine weitere Zeile einfügen, die in etwa so aussieht:

```
set XINDY_SEARCHPATH=.;c:\tex\xindy\xindy-2.0D\modules
```

Hierbei muss der Teil nach dem Zeichen „,

;“in das Verzeichnis zeigen, das Sie im zweiten Schritt angelegt haben – dort findet xindy seine mitgelieferten Style-Files.

Wenn Sie es gewohnt sind, $\LaTeX$ und/oder $\TeX$ von der Kommandozeile aus aufzurufen, bietet es sich an, xindy.bat und makindx4.bat in das Verzeichnis zu legen, in dem auch latex und tex gefunden werden. Rufen Sie hingegen $\TeX$ von einer Oberfläche wie WinEdt auf, können Sie dort direkt den Pfad zu xindy.bat eintragen.

Unix/Linux

Ich gehe im Folgenden davon aus, dass Sie auf dem System, in dem Sie xindy installieren wollen, *root*-Rechte haben. Man kann das Programm auch ohne diese Rechte installieren. Meistens ist es aber besser, in einem solchen Fall die Systembetreuung um die Installation zu bitten, einfach damit nicht 20 verschiedene Anwender dasselbe Paket installieren müssen.

Bei dem Versuch, das xindy-Paket vom CTAN-Abzug 09/2000 zu installieren, habe ich beim automatisierten Test Fehlermeldungen erhalten. Die Version 2.1, die ich von <ftp://xindy.sourceforge.net/pub/xindy/> bekommen habe, hatte dieses Problem ebenfalls; ein Update hilft hier nicht. Das Problem ist offensichtlich die Funktion „mapc“, die in den beiden fehlgeschlagenen Tests (ex1 und ex2) verwendet wird. Sie wird in der Dokumentation nicht erwähnt und funktioniert offensichtlich auch nicht. Von daher habe ich die Fehlermeldungen zu diesen beiden Tests ignoriert. Übrigens war bei der Windows-Installation keine Rede von einem solchen Problem, da für Windows nur fertig kompilierte Programme zur Verfügung stehen und die Test-Suite wohl nur für diejenigen vorgesehen ist, die das Programm übersetzen.

Das nun folgende Installations-Beispiel ist auf einem GNU/Linux-System durchgeführt worden, sollte aber mit minimalen Änderungen auch auf anderen Unix-Derivaten laufen. Das abschließende „,

make install“muss mit *root*-Rechten ausgeführt werden.

```
$ cd /usr/src/
$ tar xzf ~/incoming/xindy-2.1.tar.gz
```

```
$ cd xindy/
$ ln -s xindy-2.1 current
$ tar xzf ~/incoming/xindy-2.1-ix86-linux-glibc21.tar.gz
$ cd ix86-linux-glibc21
$ make all
...
Compilation finished!
```

You can run the testsuite by typing ‘make testsuite’.

```
$ make gzipped
gzip xindy.mem
mv xindy.mem.gz xindy.mem
$ make testsuite
...
-----
Checking tests
-----

Test "ex1" failed!
Test "ex2" failed!
make[1]: Leaving directory ‘/usr/src/xindy/xindy-2.1/tests’
$ su -c ‘make install’
Password:
./install.sh -c -d /usr/local/bin
...
```

Ein Beispiel

Wir wollen ein ganz einfaches Beispiel betrachten: Ein kurzer Text soll mit Indexeinträgen versehen werden, die ohne weitere Auszeichnung nur Seitenzahlen verwenden.

Als Erstes brauchen wir eine Stil-Datei, die sagt, was xindy eigentlich mit den Daten tun soll (der Preis der Flexibilität). Diese Dateien sind in einer einfachen Lisp-Syntax gehalten; die einzelnen Anweisungen werden in runden Klammern geschrieben. Wir verwenden folgende Datei:

```
; Ein Beispiel-Style aus der Distribution:
(require "tex/makeidx.xdy")
```

```
; TeX-Umlaute werden behandelt:
(require "tex/isolatin1m.xdy")

; Sortiere Umlaute nach Duden-Regelungen, "case-insensitiv"
(require "lang/german/duden.xdy")
(require "lang/latin/caseisrt.xdy")

; Abschnittsüberschriften
(require "lang/latin/letgroup.xdy")
```

Die letzten beiden einzulesenden Dateien sind schon in makeidx.xdy enthalten, aber es schadet nichts, sie mehrfach einzubinden, wodurch klarer wird, was passiert. require liest eine Datei aus /usr/local/lib/xindy/modules/ ein, als wäre sie an dieser Stelle Bestandteil der ursprünglichen Datei gewesen. Wenn Sie das Verhalten ein klein wenig anders haben wollen, können Sie die Dateien auch direkt in eine andere Datei einbinden und entsprechend bearbeiten.

Nun brauchen wir noch einen Beispieltext:

```
\documentclass{article}
\usepackage{makeidx}
\usepackage{xindy}
\usepackage{german}
\usepackage[latin1]{inputenc}
\makeindex
\begin{document}
\section{Eins}
\indexindy{Indexeinträge}%
\indexindy[open-range]{Beispiel}%
\indexindy[open-range]{Beispiel;weiteres}%
\indexindy[attr=see][Indexeinträge]{Indexeintrag}%
\indexindy{Test}
In diesem Text stehen Indexeintr"age.
\clearpage
\section{Zwei}
Hier ist der Eintrag "'Beispiel"' zu Ende.
\indexindy[close-range]{Beispiel}%
Au"serdem kommt noch ein Eintrag Indexeinträge,
dieser mit einer anderen Schreibweise für den Umlaut.
```

```

\indexindy{Indexeintr\ "age}
\indexindy{Test}
\clearpage
\section{Drei}
Hier ist nun auch der Indexeintrag
"Beispiel;weiteres" beendet. Der Indexeintrag
"Test" taucht auf allen drei Seiten auf.
\indexindy{Test}
\indexindy[close-range]{Beispiel;weiteres}%
\printindex
\end{document}

```

Diese Datei übersetzen wir nun ganz normal mit $\TeX$. Dabei wird eine Datei xindy_test.idx angelegt. Anschließend rufen wir xindy auf:

- Unix: xindy xindy_test.xdy xindy_test.idx
- Windows: xindy xindy_test.xdy xindy_test.idx xindy_test.ind

Hierbei entsteht die Datei xindy_test.ind, ganz analog zu dem Vorgehen mit makeindex. Diese Datei wird durch den Befehl `\printindex` am Ende unserer Testdatei eingelesen, wenn $\TeX$ ein weiteres Mal aufgerufen wird. Das Ergebnis sieht in etwa so aus:

Index

Beispiel, 1, 2
weiteres, 1-3

Indexeintrag, *siehe* Indexeinträge
Indexeinträge, 1, 2

Test, 1-3

Man sieht hieran eine Reihe von Dingen:

- Die Trennung mit dem Semikolon „;“ erzeugt einen Untereintrag. Das Trennzeichen ist in xindy.sty fest eingeebrannt, man muss Texte, die schon Indexeinträge mit einem Zeichen „!“ enthalten, bearbeiten. xindy selbst bekommt dieses Zeichen „;“ übrigens niemals zu sehen.

- Die Seitenzahlen hinter „Beispiel“ stehen einzeln, weil 1-2 etwas merkwürdig wirken würde. Bei dem Untereintrag ist aber zu sehen, dass die Bereichsangabe funktioniert.
- Verschiedene Umlaut-Schreibweisen werden als gleich erkannt.
- Wie man am letzten Eintrag sieht, werden Seitenzahlen, die direkt aufeinanderfolgen, zu einem Bereich zusammengefasst.

Die drei zuletzt genannten Punkte lassen sich durch die Konfiguration einstellen, wenn man sie gerne anders haben möchte. Wenn ausreichendes Interesse besteht, kann ich gerne in späteren Ausgaben auf Konfigurationsmöglichkeiten eingehen. Rückmeldungen bitte an christopher@creutzig.de oder an die in „Die T_EXnischen Kontakte“ angegebene Adresse.

Zusammenfassung

xindy bietet eine flexible Alternative zu makeindex. Die Konfiguration ist etwas komplizierter, bietet aber dafür wesentlich mehr Möglichkeiten. Eine Installation lohnt sich für jeden, der technische Bücher setzt, da die Flexibilität bei der Indexerstellung schnell zu besseren Indizes führt, die dem Leser den Umgang mit der besprochenen Materie erleichtern.

Von fremden Bühnen

Der \year=2002 T_EX-Kalender

T_EX Merchandising Project, Martin Schröder

Das T_EX Merchandising Project präsentiert den \year=2002 T_EX-Kalender:

Die Titelseite



the \year=2002
T_EX
calendar

Mai



April April Avril						
Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
14	1	2	3	4	5	6
15	8	9	10	11	12	13
16	15	16	17	18	19	20
17	22	23	24	25	26	27
18	29	30				

Mai May Mai						
Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
18			1	2	3	4
19	6	7	8	9	10	11
20	13	14	15	16	17	18
21	20	21	22	23	24	25
22	27	28	29	30	31	

Juni JUNE Junin						
Week	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
22						1
23	4	5	6	7	8	9
24	10	11	12	13	14	15
25	17	18	19	20	21	22
26	24	25	26	27	28	29
27	30					

Die deutsche Bibliothek - CIP-Einheitsaufnahme:

the \year=2002 T_EX calendar

Jena: IKS Garamond, 2001

ISBN 3-934601-28-6

- Format ISO A4
- 26 Seiten (12 Monate + Titel + Kolophon)
- Jeder Monat mit einem Bild von Duane Bibby; elf davon aus den Büchern von Donald Knuth
- Titelbild von Duane Bibby exklusiv für diesen Kalender
- Spiralbindung zum Umschlagen mit Bügel zum Aufhängen
- Kalendarium mit T_EX-Gedenktagen und jeweils drei Monaten auf einen Blick
- Kalendarium in deutsch/englisch/französisch
- Preis: 10,00 €

Bezugsquellen:

- Eine Buchhandlung ihrer Wahl (der Kalender sollte dank ISBN problemlos zu besorgen sein und ist im VLB gelistet)
- Lehmanns Fachbuchhandlung,
Abt. Versand,
Hardenbergstraße 11,
10623 Berlin,
Tel. 0800 / 2 66 26 65,
Fax (+49) 030 / 61 79 11 33,
E-Mail bestellung@lehmanns.de,
<http://www.lob.de>
- T_EX Merchandising Project,
p. A. Martin Schröder,
Crüsemannallee 3,
28213 Bremen,
E-Mail tm@tm.oneiros.de,
<http://tm.oneiros.de>

Spielplan

Termine

- September und Oktober 2001 Calligraphic Type Design in the Digital Age: An Exhibition in Honor of the Contributions of Hermann and Gudrun Zapf
San Francisco Public Library, Skylight Gallery
San Francisco, USA
<http://www.zapfest.com>
- 23.9.-27.9.2001 EuroT_EX 2001 - “T_EX and Meta: the Good, the Bad, and the Ugly Bits”
Kerkrade/Niederlande
<http://www.ntg.nl/eurotex>
- 28.9.2001 25. Mitgliederversammlung, DANTE e.V.
Kerkrade/Niederlande
<http://www.dante.de/dante/events/mv25/>
- 10.-25.10.2001 Frankfurter Buchmesse 2001
Frankfurt am Main
<http://www.frankfurter-buchmesse.de/>
Kontakt: Ausstellungs- und Messe GmbH Frankfurter Buchmesse
- 24.-26.10.2001 CIDE 2001
4th International Conference on The Electronic Document
Toulouse, Frankreich
<http://www.irit.fr/CIDE2001/>
- 20.-23.2.2002 DANTE 2002 und 26. Mitgliederversammlung,
DANTE e.V.
Erlangen
<http://www.dante.de/dante2002/>

T_EX-Tagung DANTE 2002 in Erlangen – Ankündigung und Call for Papers

Hans Cramer, Klaus Höppner

Die T_EX-Tagung DANTE2002 findet

vom 20. bis 23. Februar 2002

im Regionalen Rechenzentrum Erlangen (RRZE) der Friedrich-Alexander-Universität Erlangen-Nürnberg statt. Veranstalter sind gemeinsam das Regionale Rechenzentrum Erlangen und DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.

Am Mittwoch sind Tutorien geplant, Donnerstag und Freitag sind für Vorträge, Diskussionen und Präsentationen vorgesehen, am Samstag wird voraussichtlich die Mitgliederversammlung von DANTE e.V. stattfinden.

Die Teilnahme an den Tutorien ist kostenlos. Der Tagungsbeitrag für die Vorträge und das Beiprogramm (inkl. Tagungssessen) beträgt für Mitglieder von DANTE e.V. 25 € und für Nichtmitglieder 50 €. Der Besuch der Mitgliederversammlung von DANTE e.V. ist unabhängig von der Teilnahme an der Tagung; auch Nichtmitglieder sind wie üblich als Gäste gerne willkommen.

- Alle, die einen Vortrag oder ein Tutorium halten oder eine Diskussion leiten möchten, werden gebeten, dies mit dem Formular „Anmeldung von Beiträgen“

quote <http://www.dante.de/dante2002/>

oder per E-Mail

dante2002@dante.de

an die Organisatoren bis zum

15. November 2001

anzumelden. Die Annahme von verspäteten Anmeldungen ist nur unter Vorbehalt möglich.

Zu einem Vortrag oder Tutorium ist ein *extended abstract* einzureichen. Richtlinien dafür sind auf der oben genannten WWW-Seite zu finden oder bei den Organisatoren zu erfragen.

Alle Firmen und Institutionen, die ihre Produkte präsentieren bzw. die Tagung finanziell unterstützen wollen, werden gebeten, sich möglichst frühzeitig an dieselben Adressen zu wenden.

Insbesondere ist, wie auf den letzten Tagungen, ein Büchertisch mit einer Präsentation von Neuerscheinungen aus dem Bereich T_EX & Friends geplant.

Die Homepage der Tagung findet sich unter

<http://www.dante.de/dante2002/>

Hier finden Sie die jeweils aktuelle Fassung des Programms, Hinweise zum Tagungsort, zur Anreise und zu Übernachtungsmöglichkeiten sowie Formulare zur Tagungsanmeldung.

Mit Fragen, Wünschen und Anregungen wenden Sie sich bitte an

Hans Cramer, Sigmar Lingner
DANTE2002
Universität Erlangen-Nürnberg
Regionales Rechenzentrum Erlangen
Martensstr. 1
D-91058 Erlangen

E-Mail: dante2002@dante.de (bevorzugt)

oder an DANTE e.V. in Heidelberg.

Die Veranstalter hoffen, dass möglichst viele T_EX-Interessierte die Veranstaltung in Erlangen besuchen werden, und freuen sich auf einen erfolgreichen Tagungsverlauf.

Da auf der Mitgliederversammlung am 23. Februar 2002 die Amtszeit des Vorstandes endet und Neuwahlen stattfinden, würde sich der Vorstand von DANTE e.V. insbesondere über einen zahlreichen Besuch der Mitgliederversammlung freuen.

Mit freundlichen Grüßen

Hans Cramer (RRZE)

Klaus Höppner (DANTE e.V.)

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von T_EX-Anwendern statt, die für Jeden offen sind. Im WWW gibt es aktuelle Informationen unter <http://www.dante.de/events/stammtische/>.

Berlin

Rolf Niepraschk
Tel.: 0 30/3 48 13 16
niepraschk@ptb.de
*Gaststätte „Bärenschenke“
Friedrichstr. 124
Zweiter Donnerstag im Monat, 19.00 Uhr*

Bremen

Martin Schröder
Tel.: 04 21/2 23 94 25
martin@oneiros.de
*Wechselnder Ort
Erster Donnerstag im Monat, 18.30 Uhr*

Chemnitz

Ralf König
Tel.: 03 71/5 90 54 75
ralf.koenig@1998.tu-chemnitz.de
*Universitätsteil 1, Strasse der Nationen 62,
Raum 1/068
Dritter Mittwoch im Monat, 18.00 Uhr*

Dortmund

Stephan Lehmke
Stephan.Lehmke@cs.uni-dortmund.de
*Cafe Durchblick
Universität Dortmund, Campus Nord
Zweiter Mittwoch im Monat, 20.00 Uhr*

Dresden

Hilmar Preuße
hille42@gmx.de
*Medien- und Kulturzentrum Pentacon,
Schandauer Str. 64
Erster Mittwoch im Monat, 19.00 Uhr*

Erlangen

Walter Schmidt, Peter Seitz
was@VR-Web.de,
p.seitz@koehler-seitz.de
*Gaststätte „Gambrinus“
Vierzigmannstr. 7
Dritter Dienstag im Monat, 19.00 Uhr*

Freiburg

Heiko Oberdiek
Tel.: 07 61/4 34 05
oberdiek@ruf.uni-freiburg.de
*Gaststätte „Aquila“
Sautierstr. 19
Dritter Donnerstag im Monat, 19.30 Uhr*

Hamburg

Volker Hüttenrauch
volker_huettenrauch@hh.maus.de
*Vereinsheim der Hamburger
Microcomputer-Hochschulgruppe
Grindelallee 143 (Hinterhof)
Letzter Donnerstag im Monat, 18.00 Uhr*

Hannover

Mark Heisterkamp
Seminarraum RRZN, Regionales
Rechenzentrum für Niedersachsen,
Schloßwender Str. 5
Tel.: 05 11/7 62 51 34
heisterkamp@rrzn.uni-hannover.de
*Seminarraum RRZN
Schloßwender Str. 5
Zweiter Mittwoch von geraden Monaten,
18.30 Uhr*

Heidelberg

Luzia Dietsche
Tel.: 0 62 21/54 45 27
luzia.dietsche@urz.uni-heidelberg.de
China-Restaurant „Palast“

Adressen

*DANTE, Deutschsprachige
Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg*

Tel.: 0 62 21/2 97 66 (Mo, Mi–Fr, 10⁰⁰–12⁰⁰ Uhr)

Fax: 0 62 21/16 79 06

*E-Mail: **dante@dante.de***

Konten: Volksbank Rhein-Neckar eG

BLZ 670 900 00

Kontonummer 2 310 007

Postbank Karlsruhe (Auslandsüberweisungen)

BLZ 660 100 75

Kontonummer 213 400 757

Präsidium

*Präsident: Thomas Koch **president@dante.de***

*Vizepräsident: Volker RW Schaa **vice-president@dante.de***

*Schatzmeister: Horst Szillat **treasurer@dante.de***

*Schriftführer: Günter Partosch **secretary@dante.de***

*Beisitzer: Klaus Höppner **adviser@dante.de***

Server

*ftp: **ftp.dante.de** [134.93.8.251]*

*E-Mail: **ftpmail@dante.de***

*WWW: **http://www.dante.de/***

Autoren/Organisatoren

- T_EX Merchandising Project** [63]
p. A. Martin Schröder
Crüsemannallee 3
28213 Bremen
tm@tm.oneiros.de
- Ausstellungs- und Messe GmbH**
Frankfurter Buchmesse [65]
Reineckstr. 3
D-60313 Frankfurt am Main
visitor@book-fair.com
- Conference CIDE 2001** [65]
l'Institut de Recherche en Informatique de Toulouse (IRIT)
UPS-118, route de Narbonne
F-31062 Toulouse Cedex
France
- Hans Cramer** [65]
Universität
Erlangen-Nürnberg
Regionales Rechenzentrum Erlangen
Martensstr. 1
D-91058 Erlangen
- Christopher Creutzig** [54]
Im Samtfelde 19
33098 Paderborn
- Klaus Höppner** [65]
siehe Seite 70
- Laura Elizabeth Jackson** [24]
Raleigh, North Carolina (USA)
lejacks2@unity.ncsu.edu
- Thomas Koch** [4]
siehe Seite 70
- Werner Lemberg** [14]
Kl. Beurhausstr. 1
44137 Dortmund
- Janusz Marian Nowacki** [47]
al. 23 Stycznia 54d,
86-300 Grudziądz
Polen
j.nowacki@gust.org.pl
- Volker RW Schaa** [4]
siehe Seite 70
- Priv.-Doz. Dr. Rainer Thiel** [6]
Ernst-Lemmer-Str.
14
35041 Marburg
- Herbert Voß** [24]
Wasgenstr. 21
14129 Berlin
voss@lyz.orgBerlin

Die $T_{E}X$ nische Komödie

13. Jahrgang Heft 3/2001 September 2001

Impressum

Editorial

Hinter der Bühne

4 Grußwort

Bretter, die die Welt bedeuten

6 *Emacs, Auc $T_{E}X$ und MiK $T_{E}X$ s
Yap unter Windows*

14 *cjk-enc – eine Schnittstelle
zwischen Emacs und $\LaTeX$*

24 *L γ X – Open Source Document
Processor, Teil 1*

47 *„Poligraf“ oder „Zwischen $T_{E}X$
und der Druckerei“ – der zweite
Versuch*

54 *Indexverzeichnisse mit xindy
und windy.sty – eine
Einführung*

Von fremden Bühnen

63 *Der \year=2002 $T_{E}X$ -Kalender*

Spielplan

65 *Termine*

66 *$T_{E}X$ -Tagung DANTE 2002 in
Erlangen – Ankündigung und
Call for Papers*

69 *Stammtische*

Adressen

71 *Autoren/Organisatoren*