

Die T_EXnische Komödie

dante

Deutschsprachige
Anwendervereinigung T_EX e.V.

24. Jahrgang Heft 2/2012 Mai 2012

2/2012

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 Autoren wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nicht gestattet. Alle Rechte zur weiteren Verwendung außerhalb von DANTE e.V. liegen bei den jeweiligen Autoren.

Beiträge sollten in Standard- $\text{\LaTeX}$ -Quellcode unter Verwendung der Dokumentenklasse dtk erstellt und per E-Mail oder Datenträger (CD/DVD) an untenstehende Adresse der Redaktion geschickt werden. Sind spezielle Makros, $\text{\LaTeX}$ -Pakete oder Schriften notwendig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden. Weitere Informationen für Autoren findet man auf der Projektseite <http://projekte.dante.de/DTK/AutorInfo> von DANTE e.V.

Diese Ausgabe wurde mit Lua $\text{\TeX}$, beta-0.70.1-2011061410 (rev 4277) ($\text{\TeX}$ Live 2011), erstellt. Als Standard-Schriften kamen Linux Libertine, Linux Biolinum, Luxi Mono und Latin Modern Math zum Einsatz.

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
dtkred@dante.de (Redaktion)

Druck: Konrad Triltsch Print und digitale Medien GmbH
Johannes-Gutenberg-Str. 1–3, 97199 Ochsenfurt-Hohstadt

Redaktion: Herbert Voß (verantwortlicher Redakteur)

Mitarbeit:	Andreas Eder	Rudolf Herrmann	Gert Ingold
	Rolf Niepraschk	Heiko Oberdiek	Günter Partosch
	Henning Hraban Ramm	Christine Römer	Volker RW Schaa
	Martin Sievers	Dominik Waßenhoven	

Redaktionsschluss für Heft 3/2012: 15. Juli 2012

ISSN 1434-5897

Editorial

Liebe Leserinnen und liebe Leser,

diesmal decken wir den gesamten Bereich der neueren Entwicklungen ab; Marco Daniel erklärt Ihnen am Beispiel des Paketes `xparse` die Philosophie, die hinter dem Projekt $\text{\LaTeX}$ 3 steht. `xparse` ist Teil der Reihe `l3packages`, die mittlerweile aus dem experimentellen Zweig von CTAN in den der *normalen* Pakete verschoben wurde. Bei einem aktuellen $\text{\TeX}$ Live – alternativ $\text{\MiKTeX}$ – kann man davon ausgehen, dass die jeweils neueste Version installiert ist. Einige häufig verwendete Pakete wie `fontspec`, `unicode-math` oder `siunitx` setzen bereits vollständig auf die Syntax von $\text{\LaTeX}$ 3 und verwenden das Paket `xparse`.

Patrick Gundlach setzt die Lua-Serie fort und zeigt, wie man mit Lua in die durch Vor- und Rückwärtszeiger verkettete Liste der $\text{\TeX}$ -Knoten eingreifen kann. Damit sind weitreichende Modifikationen der Textausgabe möglich.

Nach einer großen Tagung finden Sie hier immer auch den formalen Teil und einen Tagungsbericht, diesmal von Roland Geiger verfasst.

Neben den bekannten und häufig verwendeten KOMA-Script-Klassen, gibt es unter anderem noch die Dokumentenklasse `memoir`, welche vielfach im englischsprachigen Bereich benutzt wird. Die Unterschiede zeigt Christine Römer in ihrem Beitrag. Dominik Waßenhoven beschäftigt sich in zwei Beiträgen mit typografischen Fragen, während Jürgen Hanneder die Veränderungen von $\text{\TeX}$ aus der Sicht eines an Kontinuität interessierten Anwenders zeigt. Das Prinzip von Knuth, dass jedes Dokument zu jeder Zeit fehlerfrei übersetzbar sein sollte, gilt eben nur, wenn man den Neuerungen der $\text{\TeX}$ -Welt entsagt. Um alle Akzente, die in diesem Beitrag verwendet werden, korrekt zu setzen, wurde als Standardschrift die Linux Libertine verwendet; die TG-Pagella hat hier noch einige Probleme.

Ich wünsche Ihnen wie immer viel Spaß beim Lesen und verbleibe mit $\text{\TeX}$ nischen Grüßen,

Ihr Herbert Voß

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

dies ist mein erstes Grußwort als Vorsitzender von DANTE e.V. Daher zunächst ein paar kurze Worte zu mir: Ich bin 34 Jahre alt, Diplom-Mathematiker und arbeite derzeit als wissenschaftlicher Mitarbeiter an der Universität Trier im Bereich der »Digital Humanities«. Seit etwa zehn Jahren beschäftige ich mich mit $\text{\TeX}$ und habe dabei schon früh begonnen, es in verschiedenen wissenschaftlichen Disziplinen bekannt zu machen und zu verankern.

Im Vergleich zu den vergangenen Vorstandswahlen gab es diesmal größere Veränderungen. Nach vielen Jahren als (stellv.) Vorsitzender wollte Volker RW Schaa etwas kürzer treten. Gleichwohl bleibt er dem Vorstand als Beisitzer erhalten und wird seine wertvollen Kontakte, insbesondere auch die internationalen, weiterhin einbringen. Die Position von Adelheid Grob, die sich leider nicht mehr zur Wahl stellte, hat mit Herbert Voß ein prominentes Mitglied unseres Vereins übernommen. Ich bin sehr froh, mit ihm einen erfahrenen Mitstreiter zu haben, der die gewachsenen Strukturen innerhalb des Vereins und die handelnden Personen gut kennt. Für den ebenfalls ausgeschiedenen Bernd Raichle, dem ich an dieser Stelle nochmals herzlich für sein langjähriges Engagement im Vorstand danken möchte, wurde Patrick Gundlach zum Beisitzer gewählt. Er verstärkt damit die »junge« Fraktion. Klaus Höppner, Manfred Lotz sowie Uwe Ziegenhagen komplettieren das Vorstandsteam auf ihren vorherigen Positionen.

Auch wenn es kein klassisches Wahlprogramm umzusetzen gilt, so möchte ich doch Schwerpunkte benennen, mit denen aus meiner Sicht in den nächsten Jahren Weichen für die Zukunft gestellt werden. Das Thema »alt und jung« ist ein solches Feld. Dabei geht es in meinen Augen weniger um die seit einigen Jahren fortschreitende Erhöhung des Anteils der Rentnerinnen und Rentner unter den Mitgliedern bei gleichzeitiger Abnahme des Studierendenanteils. Vielmehr ist eine thematische und methodische Verschiebung zu beobachten. Die Probleme und Lösungen eines $\text{\LaTeX}$ 2_ε-Nutzers der ersten Stunde sind natürlich nicht mehr diejenigen eines $\text{\LaTeX}$ -Anwenders; 7-Bit und Type-1-Fonts nicht mehr notwendigerweise Teil des

heutigen T_EX-Alltags. Doch lebt T_EX, lebt DANTE e.V. auch genau von dieser Vergangenheit, von seiner Entwicklung und Geschichte, über die viele Mitglieder sehr anschaulich berichten können. Daher ist es aus meiner Sicht wichtig, darauf zu achten, auch in Zukunft für die »alten T_EX-Hasen« wie für die »jungen Wilden« etwas anzubieten.

Für Letztere gehört dazu heutzutage auch eine geänderte »Zielgruppenansprache«. T_EX kann heute jeder nutzen, sofern er einen Internetzugang hat. Dokumentationen und viele Hilfsmöglichkeiten finden sich im weltweiten Netz, so dass eine sorgenfreie T_EX-Nutzung gewährleistet ist – ganz ohne DANTE-Mitgliedschaft. Ich danke deshalb Uwe Ziegenhagen und Patrick Gundlach dafür, dass sie die Verantwortung für die DANTE-Auftritte bei Facebook (<https://www.facebook.com/dante.tex>) bzw. Twitter (https://twitter.com/#!/dante_ev) übernommen haben und dadurch helfen, eine wichtige neue Brücke nach außen aufzubauen.

Diese beiden Kanäle gehören heute genauso zur modernen »T_EX-Kommunikation« wie das Angebot des sehr erfolgreichen Netzwerks »StackExchange« (<http://tex.stackexchange.com/>). Dort bieten u. a. einige unserer Mitglieder wertvolle Unterstützung, so dass in Zukunft da hoffentlich ebenfalls eine Verbindung zwischen T_EX und DANTE etabliert werden kann. Last, but not least, hat Herbert Voß einen Online-Mitgliedsantrag sowohl für die Voll- als auch die Schnuppermitgliedschaft aufgesetzt, der gut angenommen wird. All dies sind aus meiner Sicht erste wichtige Schritte, um den Kontakt zwischen den (neuen) Anwendern und unserem Verein zu erhalten und zu intensivieren.

Grundlage für den Erfolg von T_EX ist sicherlich auch CTAN, das allzu oft als »gottgegeben« angesehen wird. Doch steckt in der Betreuung der Server – und der Anwender – eine Menge Arbeit. Insofern gebührt Rainer Schöpf und Joachim Schrod, die »nebenbei« auch noch unseren Vereinsserver betreuen, und Robin Fairbairns an dieser Stelle großer Dank für ihren langjährigen und nicht immer einfachen Einsatz. Für die »Außendarstellung« hat sich Gerd Neugebauer der Entwicklung einer neuen CTAN-Website angenommen, die sich derzeit zwar noch in der Testphase befindet, doch schon sehr vielversprechend aussieht (<http://www-new.ctan.org/>).

Eine weitere Brücke in die Außenwelt stellen die DANTE-Tagungen dar. Leider ist hier in den letzten Jahren sowohl die Zahl der Teilnehmer wie auch die der angebotenen Vorträge stetig zurückgegangen. Wir werden daher im Vorstand für kommende Tagungen Änderungen diskutieren, um den unterschiedlichen Wünschen und Anforderungen Rechnung zu tragen und unsere Veranstaltungen dadurch wieder attraktiver zu machen. Ich möchte an dieser Stelle bereits herzlich zur 47. Mitgliederversammlung von DANTE e.V. einladen. Sie findet voraussichtlich am 8. Oktober um 17 Uhr statt, diesmal im Rahmen der EuroT_EX in Breskens (Niederlande) vom 8.–12. 10. 2012 (<http://www.ntg.nl/eurotex2012/>). Da zeitgleich auch das sechste

internationale ConT_EXt-Treffen dort stattfindet, ist Breskens auf jeden Fall eine Reise wert. DANTE-Mitglieder erhalten zudem einen zusätzlichen Preisnachlass von 50 € auf das angebotene »Konferenz-Komplettpaket« aus Unterkunft und Verpflegung. Schließlich benötigt auch die Redaktion der DTK immer wieder neue Artikel – kleine wie große – sowie redaktionelle Unterstützung, um DANTE e.V. und T_EX abwechslungsreich und modern zu verkörpern. Dies ist umso wichtiger, als dass auf Beschluss der Mitgliederversammlung nunmehr alle DTK-Ausgaben bereits nach einem Jahr einer möglichst breiten Öffentlichkeit über <http://www.dante.de/DTK/Ausgaben.html> in digitaler Form zur Verfügung gestellt werden.

Das Ziel insgesamt ist klar: Die regelmäßig anfallenden Aufgaben sollten auf möglichst viele Freiwilligenschultern verteilt werden. Für viele von Ihnen ist sicherlich etwas dabei, für jedes »Zeitbudget« gibt es Möglichkeiten, sich einzubringen. Nicht alle Aufgaben bedingen einen langfristigen und/oder hohen Zeiteinsatz. Vielleicht haben Sie ja auch eine konkrete (Projekt-)Idee, wo Sie sich einbringen wollen und können – lassen Sie es uns wissen.

In diesem Sinne danke ich allen Mitgliedern für ihren bisherigen und künftigen Einsatz bei den vielfältigen Aufgaben des Vereins und wünsche Ihnen viel Spaß bei der Lektüre dieser Ausgabe der DTK.

Herzlichst Ihr/Euer
Martin Sievers

Beschlüsse der 46. Mitgliederversammlung von DANTE e.V. am 8. März 2012 in Leipzig

Manfred Lotz

Zeit: 8. März 2012, 9:15 Uhr – 12:00 Uhr

Ort: HTWK Leipzig
Lipsius-Bau
Raum Li 211
Karl-Liebknecht-Straße 145
04277 Leipzig

Teilnehmer: 34 (anhand der ausgegebenen Stimmkarten)

Leitung: Volker RW Schaa (Vorsitzender von DANTE e.V.)

Protokollant: Manfred Lotz (Schriftführer von DANTE e.V.)

Die Mitgliederversammlung wurde satzungsgemäß eingeladen und ist beschlussfähig.

TOP 1: Begrüßung, Tagesordnung und Vorstellung des Vorstands

TOP 1.1: Begrüßung und Vorstellung

Volker RW Schaa begrüßt die Teilnehmer der 46. Mitgliederversammlung von DANTE e.V. in Leipzig und stellt die Tagesordnung vor:

1. Begrüßung, Tagesordnung und Vorstellung des Vorstands

- Begrüßung und Tagesordnung
- Vorstellung des Vorstands

2. Bericht des Vorstands

- Ehrenpreis 2012
- Vergangene Tagungen
- Kommende Tagungen
- CTAN & Co
- FAQ, Mitarbeit, DANTE-Edition
- Projekte
- Trolling

3. Finanzbericht
4. Bericht der Kassenprüfer
5. Entlastung des Vorstands
6. Wahl eines Wahlleiters
7. Neuwahl des Vorstands
8. Nachwahl eines Kassenprüfers
9. Abstimmung über den Antrag »Zeitnahe allgemeine Veröffentlichung von ›Die T_EXnische Komödie‹ im Web«
10. Sonstiges

Die Tagesordnung wird ohne Einspruch akzeptiert.

TOP 1.2: Vorstellung des Vorstands

Bis auf Adelheid Grob, Klaus Höppner und Bernd Raichle sind die derzeitigen Vorstandsmitglieder anwesend und werden von Volker RW Schaa vorgestellt: Volker RW Schaa (Vorsitzender), Manfred Lotz (Schriftführer), Martin Sievers (Beisitzer), Uwe Ziegenhagen (Beisitzer), Herbert Voß (Beisitzer).

Der Verein unterhält in Heidelberg ein Büro, das von Frau Karin Dornacher geleitet wird.

TOP 2: Bericht des Vorstands

TOP 2.1: Ehrenpreis 2012

Der Ehrenpreis für 2012 geht an Ulrike Fischer für ihr langjähriges Engagement im Bereich T_EX. Über lange Jahre hat sie sich als unermüdliche Helferin bei Problemen, die auf den deutsch- und englischsprachigen T_EX-spezifischen Newsgroups behandelt werden, einen Namen gemacht. Sie wird zur nächsten Tagung von DANTE e.V. eingeladen.

TOP 2.2: Vergangene Tagungen

- TUG 2011 in Trivandrum, Indien vom 19. 10.–21. 10. 2011

TOP 2.3: Kommende Tagungen

- 20. BachoT_EX in Bachotek, Polen vom 29. 4.–3. 5. 2012
- TUG 2012 in Boston, Massachusetts, USA vom 16. 7.–18. 7. 2012
- Herbsttagung DANTE e.V. und EuroT_EX 2012 sowie 6. ConT_EXt User Meeting in Breskens, Niederlande vom 8. 10.–12. 10. 2012

Die Tagung wird gemeinsam von der NTG und DANTE e.V. organisiert.

TOP 2.4: CTAN & Co

- Nach dem Artikel von Joachim Schrod und Rainer Schöpf in *Die T_EXnische Komödie 2/2011* haben sich einige Personen gemeldet. Es soll ein Treffen in naher Zukunft stattfinden.
- CTAN-Webseite
Gerd Neugebauer hat die neue Seite <http://www-new.ctan.org/> entworfen und programmiert. Obwohl sie zur Zeit nach seinen eigenen Angaben in einem pre-alpha Stadium ist, sieht sie schon sehr vielversprechend aus.
- T_EX Collection 2012
Die T_EX Collection 2012 mit ihren Komponenten T_EXLive, MacT_EX, proT_EXt und CTAN-Abzug soll bis Ende Juni fertiggestellt sein und dann in den USA produziert werden. Damit wäre die DVD auf der TUG-Tagung in Boston im Juli verfügbar. Mit der Auslieferung an die einzelnen Benutzergruppen ist dann im August zu rechnen.

TOP 2.5: FAQ, Mitarbeit, DANTE-Edition

- Die FAQ wurde vor zwei Jahren ins Wiki gestellt. Seitdem ist die Zahl der Mitarbeiter auf Null gefallen.
- Mitarbeit
Wir wünschen uns mehr Aktive, allerdings sollte es sich eher um ein längerfristiges Engagement (mit geringem Arbeitsaufwand), als um kurze, intensive Mitarbeit handeln.
Beispiele für längerfristiges Engagement: Mitarbeit in der Redaktion der DTK mit einem geschätzten Aufwand von ca. 3 Tagen alle 3 Monate oder die Pflege der DTK-Klasse.
- dante-Edition bei lehmanns media
Herbert Voß weist daraufhin hin, dass die bei der lehmanns media erscheinende Buchreihe DANTE-Edition offen ist für alle Autoren.

TOP 2.6: Projekte

Genehmigte Projekte

- Extended Euler Font
Konvertierung AMS Euler Type 1 Fonts in OpenType und Erweiterung
 - ▷ Rahmen: 3.100 €
 - ▷ Basisversion vorhanden, zur Zeit ruht das Projekt wegen anderer Verpflichtungen.

Mehr voraussichtlich zur EuroT_EX

- Gyre Math Fonts

Hans Hagen erhielt eine Testversion der Gyre Math Fonts und ist am ausprobieren. Bei der Bacht_EX ist dann bereits mit der Präsentation einer Vorabversion zu rechnen.

Beantragte Projektförderung

- OrientalT_EX BIDI

- ▷ A Bidirectional Layout Model for Critical Editions
- ▷ Rahmen: 7.500 USD

- OrientalT_EX FONT

- ▷ Unicode script LM-Arabic Mono
- ▷ Rahmen: 7.500 USD

- Satzmodul für rabbinische Literatur im Talmud-Stil

Harald Vajkonny hat die Anforderungen an ein Satzmodul für rabbinische Literatur im Talmud-Stil beschrieben. Hier geht es darum, ein komplexes Abschnittslayout nachzubilden, wie es sich in der älteren rabbinischen Literatur und insbesondere im Talmud findet. Es geht hier zuerst um eine Machbarkeitsstudie, danach erst um eine Förderung.

TOP 2.7: Trolling

- Äußerungen auf der vereinsinternen Mailingliste

In den letzten Monaten hat es erheblichen Unmut auf der dante-ev-Mailingliste über die Äußerungen eines DANTE-Mitglieds gegeben.

- Reaktionen

In mehreren Mails wurde der Vorstand aufgefordert, das Mitglied zu reglementieren und gegebenenfalls auszuschließen.

- Aktion

Der Vorstand hat beschlossen, Mitglieder, die durch ihre Äußerungen auf den Mailinglisten den Vereinsfrieden stören, zweimal zu ermahnen und dann von der Liste auszuschließen.

Über eine solche Maßnahme soll dann bei der jeweils nächsten Mitgliederversammlung berichtet werden.

TOP 3: Finanzbericht

Herbert Voß stellt in Vertretung für Klaus Höppner den Finanzbericht vor.¹

TOP 4: Bericht der Kassenprüfer

Die Kassenprüfer Doris Behrendt und Martin Wilhelm Leidig haben am 25.2.2012 im Büro von DANTE e.V. eine Kassenprüfung vorgenommen. Thomas Ratajczak, der bei der Kassenprüfung wegen Krankheit verhindert war, stellt den Kassenbericht vor. Er bescheinigt der Kassenführung einen hervorragenden Zustand und stellt fest, dass die Vorschläge der vergangenen Jahre umgesetzt wurden.

TOP 5: Entlastung des Vorstands

Thomas Ratajczak beantragt die Entlastung des Vorstands durch die Mitgliederversammlung. Der Antrag wird einstimmig angenommen. Der Vorstand nimmt nicht an der Abstimmung teil.

TOP 6: Wahl des Vorstands

Zunächst bestimmt die Mitgliederversammlung Thomas Ratajczak einstimmig zum Wahlleiter.

Adelheid Grob und Bernd Raichle stellen sich nicht mehr zur Wahl.

Für den Vorstand stellen sich zur Wahl: Martin Sievers (Vorsitzender), Herbert Voß (stellvertretender Vorsitzender), Klaus Höppner (Schatzmeister), Manfred Lotz (Schriftführer), Volker RW Schaa (Beisitzer), Patrick Gundlach (Beisitzer) und Uwe Ziegenhagen (Beisitzer).

Auf Antrag von Volker RW Schaa werden die Wahlen geheim durchgeführt. Im Einzelnen ergeben sich die folgenden Abstimmungsergebnisse²:

		Ja	Enth.	Nein	ungültig
Martin Sievers	Vorsitzender	28	5	–	–
Herbert Voß	stellv. Vorsitzender	30	3	–	–
Klaus Höppner	Schatzmeister	30	3	–	–
Manfred Lotz	Schriftführer	32	2	–	–
Patrick Gundlach	Beisitzer	29	1	3	–
Volker RW Schaa	Beisitzer	31	0	2	–
Uwe Ziegenhagen	Beisitzer	28	3	2	–

¹ Anmerkung des Protokollanten: Der Finanzbericht erscheint in »Die T_EXnische Komödie« als Anhang zum Protokoll.

² Die Beisitzer werden in einem Wahlgang, bei dem auf dem Wahlzettel die drei Kandidaten genannt werden müssen, gewählt.

Damit sind Martin Sievers (Vorsitzender), Herbert Voß (stellvertretender Vorsitzender), Klaus Höppner (Schatzmeister), Manfred Lotz (Schriftführer), Patrick Gundlach (Beisitzer), Volker RW Schaa (Beisitzer) und Uwe Ziegenhagen (Beisitzer) gewählt. Alle Gewählten nehmen die Wahl an.³

Thomas Ratajczak gibt die Versammlungsleitung an Martin Sievers als neuen Vorsitzenden ab.

TOP 7: Nachwahl eines Kassenprüfers

Die Amtszeit von Martin Wilhelm Leidig ist abgelaufen. Er steht für eine erneute Kandidatur zur Verfügung und hatte, da er verhindert ist, vorab erklärt, die Wahl anzunehmen, falls er gewählt würde.

Mit 32 Ja-Stimmen und einer Enthaltung wird Martin Wilhelm Leidig für die nächsten zwei Jahre als Kassenprüfer gewählt.

TOP 8: Abstimmung über den Antrag »Zeitnahe allgemeine Veröffentlichung von ›Die T_EXnische Komödie‹ im Web«

Patrick Gundlach stellt den folgenden Antrag.

»Die MV möge beschließen:

- Die vollständigen Hefte der DTK werden langfristig nach dem Erscheinen kostenlos der Allgemeinheit zum Download im PDF-Format zur Verfügung gestellt.
- Den Mitgliedern werden die vollständigen Hefte spätestens mit Auslieferung des Druckwerks zum Download zur Verfügung gestellt.
- Der Vorstand hat das Recht, die Zeitspanne zwischen Drucklegung einer DTK und dem kostenlosen Download für die Allgemeinheit auf ein Jahr zu verlängern.«

Nachdem Patrick Gundlach seinen Antrag erläutert hat, entsteht eine lebhafte Diskussion. Er sieht vor allem einen Werbeeffect für den Verein, wenn die vollständigen Hefte der DTK der Allgemeinheit zur Verfügung stehen. Außerdem haben dann auch die Mitglieder, die im Sinne einer langen Postlaufzeit ungünstig wohnen, einen zeitnahen Zugriff auf die DTK. Volker RW Schaa bezweifelt den Werbeeffect.

Harald König ist dafür, dass die DTK nach einem Jahr veröffentlicht wird, aber dann auf Monatsbasis gerechnet. Herbert Voß erklärt, dass er nicht bereit ist, die DTK sofort auszuliefern.

Er weist darauf hin, dass die Rechte der in der DTK veröffentlichten Artikel nach einem Jahr verfallen. Gerd Neugebauer erklärt, dass dies seinerzeit so gedacht war,

³ Klaus Höppner hatte vorab erklärt, dass er, falls er gewählt würde, die Wahl annähme.

dass der Autor nach einem Jahr seinen Artikel selber veröffentlichen kann. An den Vorstand ergeht der Auftrag, die rechtliche Situation zu klären.

Falk Hohlfeld stellt den modifizierten Antrag,

- dass die »Die TeXnische Komödie« ein Jahr nach Erscheinen nicht nur für alle Mitglieder, sondern auch für die Allgemeinheit zur Verfügung gestellt wird.

Bei den folgenden Abstimmungen wird der Antrag Patrick Gundlachs mit 4 Ja-Stimmen, 9 Enthaltungen und 19 Nein-Stimmen abgelehnt.

Der Antrag Falk Hohlfelds wird mit 20 Ja-Stimmen, 10 Enthaltungen und 2 Nein-Stimmen angenommen.

TOP 9: Sonstiges

- Martin Schröder erinnert daran, dass das 25-jährige Bestehen von DANTE e.V., welches 2014 in Heidelberg begangen werden soll, vorbereitet werden müsse.
- Winfried Neugebauer weist darauf hin, dass zur Zeit auf <http://www.dante.de/> die Frühjahrstagung mit z. B. *DANTE 2012* und die Herbsttagung dann mit *Herbst 2012* angekündigt wird. Er bittet darum, diese Ankündigung zu vereinheitlichen.
- Martin Schröder schlägt vor, das Protokoll der Mitgliederversammlung früher vereinsöffentlich zugänglich zu machen.

Es wird darauf hingewiesen, dass das Protokoll, da vereinsintern, nicht öffentlich ins Web gestellt werden kann. Da auch nicht jeder die dante-ev-Mailingliste abonniert hat, wäre dann dafür eventuell eine eigene Mailingliste geeignet.

Martin Sievers schließt die Versammlung um 12:00 Uhr.

Volker RW Schaa
(Versammlungsleiter,
1. Hälfte)

Martin Sievers
(Versammlungsleiter,
2. Hälfte)

Thomas Ratajczak
(Wahlleiter)

Manfred Lotz
(Schriftführer)

Bericht des Schatzmeisters 2011

Klaus Höppner

Bankkonten

Neben einigen Konten bei der Volksbank (VR Bank Rhein-Neckar eG) bestehen weiterhin ein PayPal-Konto und ein Firmentagesgeldkonto bei der Degussa Bank.

Konto	Stand	1.1.2011 €	31.12.2011 €	Saldo €
001	Barkasse	332,05	20,08	-311,97
011	VoBa Giro 2310007	8.475,22	7.462,13	-1.013,09
012	VoBa Cash (Tagesgeld)	35.288,82	43.010,99	7.722,17
014	VoBa Wachstum Festgeld	11.001,66	11.386,72	385,06
015	VoBa Extrazins II	20.170,94	20.513,33	342,39
031	PayPal	348,11	330,65	-17,46
041	Degussa	49.138,24	49.730,75	592,51
Summe		124.755,04	132.454,65	7.699,61

Gegebenüber dem Vorjahr ist ein Mitte 2010 ausgelaufenes Wachstumskonto weggefallen. Ob der Erhalt des Kontos bei der Degussa-Bank weiterhin sinnvoll ist, muss noch entschieden werden. Die zum Zeitpunkt der Kontoeröffnung für ein Firmentagesgeldkonto sehr guten Zinskonditionen sind nur noch minimal besser als bei der Volksbank. Da das Konto kostenlos ist, kann aber noch etwas abgewartet werden, ob sich die Konditionen wieder verbessern.

Kostenstellenübersicht

Die einzelnen Einnahmen und Ausgaben werden auf unterschiedliche Kostenstellen verbucht, die im folgenden dargestellt sind. Diese sind in Kostenstellen für Einnahmen (800 und folgende) sowie für Ausgaben (400 und folgende) aufgeteilt. Zum Vergleich sind die Werte aus den Vorjahren mit angegeben.

Einnahmen

Kst.	Bezeichnung	Saldo (€)		
		2011	2010	2009
810	Beiträge	73.942,55	82.354,19	72.793,87
812	Beiträge Vorjahr	80,00	40,00	140,00
813	Beiträge Vorauszahlung	6.265,50	9.478,50	1.670,00
	Zwischensumme Beiträge	80.288,05	91.872,69	74.603,87
815	TUG-Beitrag	1.412,00	1.176,50	1.082,00
820	Spenden	1.667,20	1.691,38	1.407,07
822	Verkauf sonstiges	290,25	227,50	75,70
830	Verkauf Bücher	4.273,89	4.944,30	7.426,35
841	DANTE Frühjahrstagung	975,00	580,00	725,00
842	DANTE Herbsttagung			
844	Auslandstagungen			
850	Zinsen	2.042,13	2.754,71	2.651,00
860	WinEdt	265,50	642,50	120,00
890	Einnahmen sonstiges	-114,92	-75,48	-118,83
	Einnahmen gesamt	91.099,10	103.814,10	87.972,16

Ausgaben

Demgegenüber stehen die Ausgaben, die ebenfalls auf verschiedene Kostenstellen verbucht werden:

Kst.	Bezeichnung	Saldo (€)		
		2011	2010	2009
410	DTK	-18.467,33	-27.493,14	-13.361,73
415	TUG Beitrag / Zahlung	-1.144,79	-1.398,00	-1.041,00
420	Einkauf Bücher	-1.483,93	-2.593,44	-3.231,03
422	Einkauf sonstiges	-367,00		-46,95
423	Einkauf für Mitglieder (DVD)	-2.140,88		-904,82
425	WinEdt		-1.358,83	
430	Vorstand	-8.520,47	-14.288,99	-9.382,40
441	DANTE Frühjahrstagung	-1.959,59	-819,85	-1.396,44
442	DANTE Herbsttagung	-131,80	-244,99	
443	Bursary andere Tagungen	-1.039,63	-820,00	-948,29
444	Tagung Ausland			
445	EuroTeX	-500,00		
446	Linuxtag	-1.744,33	-2.620,50	-3.757,35

Forts. auf der nächsten Seite

Kst.	Bezeichnung	Saldo (€)		
		2011	2010	2009
451	Büro Miete+Nebenkosten	-8.957,91	-8.802,52	-8.851,88
452	Personal	-29.153,92	-28.302,32	-29.232,38
453	Verbrauchsmaterial	-251,69	-486,97	-236,70
454	Inventar, Anschaffungen	-678,04		
455	Porto	-1.887,33	-2.767,95	-1.085,10
460	Web-Server	-2.086,99	-2.263,75	-2.741,02
480	Spesen	-2.030,16	-1.323,70	-898,39
485	Geschenke	-51,92	-423,79	-173,86
499	Sonstiges	-162,00	-132,73	-191,83
6474	Projektfonds	-639,78	-2.648,28	-13.502,81
Ausgaben gesamt		-83.399,49	-98.789,75	-90.983,98

Saldo der Einnahmen und Ausgaben

	2011	2010	2009	2008
Einnahmen gesamt	91.099,10	103.814,10	87.972,16	88.611,94
Ausgaben gesamt	-83.399,49	-98.789,75	-90.983,98	-85.889,95
Gesamtsaldo	7.699,61	5.024,35	-3.011,82	2.721,99

Bewertung der Zahlen

Der Abschluss weist für das Jahr 2011 einen Gewinn von etwa 7.700 € aus. Dies ist nach der im Vorjahr vorgestellten (sehr konservativen) Finanzplanung überraschend, da eigentlich mit einem kleinen Verlust geplant wurde. Tatsächlich war die Höhe der eingenommenen Mitgliedsbeiträge deutlich niedriger als 2010, da – wie im letzten Kassenbericht dargestellt – die Beitragsrechnungen für das Jahr 2011 schon sehr früh vor dem Jahreswechsel verschickt und daher viele Beitragszahlungen noch 2010 verbucht wurden. Es sei an dieser Stelle daran erinnert, dass unsere Buchhaltung kameralistisch, d. h. nicht periodengerecht ist.

Dem stehen aber deutlich niedrigere Ausgaben als kalkuliert entgegen. So waren die Druckkosten für »Die T_EXnische Komödie« deutlich zu hoch angesetzt, insbesondere da 2011 keine Extrakosten anfielen, wie es beispielsweise in früheren Jahren bei Farbdrucken wie den EuroT_EX-Ausgaben der Fall war. Weiterhin flossen kaum Mittel aus dem Projektfonds ab, weil bei einigen geförderten Fonds-Projekten aus persönlichen Gründen der Geförderten die Weiterarbeit unterbrochen wurde. Da kein Mitglied des Vorstands an der TUG 2011 in Indien teilnehmen konnte, lagen auch die Spesen des Vorstands etwas niedriger als eingeplant.

Die Höhe unseres Vermögens von 120.000 € liegt über einer Jahreseinnahme, dem Richtwert, was das Finanzamt bei gemeinnützigen Vereinen als Rücklagen akzeptiert. Es ist zu hoffen, dass die 2010 unterbrochenen Projekte wieder in Schwung (oder neue hinzu) kommen, denn wir sollten einen Abbau der Rücklagen anstreben oder zumindest verhindern, dass diese weiter wachsen. Andererseits tun sich durch die hohen Rücklagen große Spielräume für die Finanzierung der von DANTE gemeinsam mit der NTG veranstalteten EuroT_EX in Breskens, Belgien, und des mit 2014 nicht mehr allzuweit entfernten 25-jährigen Jubiläums unseres Vereins auf.

Mitgliederzahlen

Die Mitgliederzahlen sind relativ stabil knapp über 2.000 Mitgliedern. In der folgenden Tabelle sind jeweils die Mitgliederzahlen zur ersten Ausgabe von »Die T_EXnische Komödie« des entsprechenden Jahres angegeben. Die Tendenz ist leicht fallend. Besonders auffallend ist die Zahl der in der Zeile »ermäßigt« enthaltenen studentischen Mitglieder. Dieser Anteil ist seit 2007 von 200 auf 132 gesunken.

Mitglieder	2012	2011	2010	2009	2008	2007
Schnupper	15	14	18	21	11	11
Schüler	10	9	9	4	6	8
Ermäßigt	227	239	246	256	286	283
Privat	1681	1673	1684	1696	1674	1626
Institut	107	106	107	108	109	113
Firma	27	26	26	29	26	25
Ehren	8	8	8	8	8	8
Gesamt	2075	2075	2098	2122	2120	2074

Projekt-Förderung

Projekt	Betrag (€)
Regionaltreffen Süd	139,78
Libertine Fonts	500,00
Gesamt	639,78

WinEdt-Lizenzen

DANTE e.V. bietet seinen Mitgliedern einen günstigen Bezug der WinEdt-Lizenzen an. Nach Erscheinen der neuen Version wurden im Jahr 2010 wieder verstärkt

Lizenzen nachgefragt. Hierdurch wurde es notwendig, neue Lizenzen vom Hersteller zu erwerben (jeweils 50 normale und ermäßigte Lizenzen):

Jahr	Verkauf	Einkauf	Saldo
2002	958,98	-1.646,44	-687,46
2003	835,00	-349,89	485,11
2004	402,50	0,00	402,50
2005	570,00	-1.367,76	-797,76
2006	642,50	-960,26	-317,76
2007	222,50	0,00	222,50
2008	217,50	0,00	217,50
2009	120,00	0,00	120,00
2010	642,50	-1.358,83	-716,33
2011	265,50	0,00	265,50
Gesamt	4.876,98	-5.683,18	-806,20

Bücher für Mitglieder

Der Ein- und Verkauf von Büchern wird seit 2007 extra in der Finanzbuchhaltung protokolliert. Hieraus ergeben sich seitdem die folgenden Daten:

Jahr	Verkauf	Einkauf	Saldo
2007	2.737,00	-2.400,10	336,90
2008	3.655,00	-2.736,15	918,85
2009	7.426,35	-3.231,03	4.195,32
2010	4.944,30	-2.593,44	2.350,86
2011	4.273,89	-1.483,93	2.789,96
Gesamt	23.036,54	-12.444,65	10.591,89

Der Buchverkauf stellt inzwischen eine stabile Einnahmequelle dar. Auf Anregung früherer Kassenprüfer wird seit 2011 eine einfache Inventarliste von Büchern geführt. Diese ergab bei der letzten Kassenprüfung, dass es keine Diskrepanz zwischen Inventarliste und den gebuchten Ein- und Verkäufen gibt.

Finanzplan 2012

Das tatsächliche Ergebnis am Ende des Jahres hängt von einigen Unwägbarkeiten ab, angefangen bei den Terminen des Rechnungsversands und den Buchungsterminen »großer« Rechnungen (insbesondere Druckkosten), aber auch von persönlichen

Faktoren, wie es sich beim Projektfonds im letzten Jahr gezeigt hat. Solche Einflussfaktoren sind im folgenden Finanzplan naturgemäß nicht einkalkulierbar. Eine weitgehend konstante Fortschreibung der letztjährigen Ein- und Ausgabesituation zeigt, dass die Finanzsituation im Verein recht ausgeglichen ist. Nicht berücksichtigt sind z. B. zusätzliche Kosten für die EuroT_EX über die normale Unterstützung internationaler Konferenzen hinaus. Diese können nun entstehen, da DANTE e.V. die EuroT_EX mitorganisiert. Auch ein Tagungsband der EuroT_EX, der dann als Ausgabe von »Die T_EXnische Komödie« erscheint, würde wegen des Farbdrucks höhere Kosten verursachen, wegen der Zeit zur Erstellung des Tagungsbandes wahrscheinlich aber erst 2013. Wie vorher dargestellt sind solche zusätzlichen Ausgaben in Anbetracht der hohen Rücklagen sogar wünschenswert.

Finanzplan 2012	
Beiträge	85.000,00
Spenden	1.600,00
Zinsen	2.000,00
DANTE-Tagungen	-1.000,00
Verkauf / Einkauf	3.000,00
Einnahmen	90.600,00
DTK	-20.000,00
DVD	-2.000,00
Büro	-40.000,00
Vorstand	-12.000,00
Inventar	-500,00
Porto	-2.500,00
Web-Server	-2.500,00
Förd. Int. Tagung.	-1.500,00
Veranst. (LinuxTag)	-1.000,00
Projektfonds	-5.000,00
sonst. Ausgaben	-4.000,00
Ausgaben	-91.000,00
Saldo	-400,00

Bericht der Kassenprüfer zum Vereinsjahr 2011

Doris Behrendt, Martin Wilhelm Leidig

Einführung

Am Samstag, den 25. Februar 2012, trafen sich die von den Vereinsmitgliedern ordentlich gewählten Kassenprüfer *Doris Behrendt* und *Martin Wilhelm Leidig* zur Kassenprüfung im Vereinsbüro in Heidelberg. Neben der Büroleiterin Frau *Karin Dornacher* war auch der Schatzmeister *Klaus Höppner* anwesend. *Thomas Ratajczak* als dritter Kassenprüfer war krankheitsbedingt entschuldigt.

Die Prüfung des Rechnungsjahres 2011 begann um 10.30 Uhr und endete am selben Tag gegen 17.00 Uhr. Nach ersten allgemeinen Informationen über das vergangene Rechnungsjahr wurden alle zur Prüfung notwendigen Akten nebst Kontenplänen von Frau *Karin Dornacher* zur Überprüfung vorgelegt.

Rechtsgrundlage

Die Kassenprüfung wurde nach den GOB¹ und den Zielen des Vereins nach der derzeit gültigen Vereinssatzung durchgeführt.

Die Ergebnisse der Prüfung wurden unmittelbar protokolliert.

Prüfgegenstände

- Barkasse
- Girokonto VR-Bank 2310007
- VR-Bank Cash 502310011
- VR-Bank Wachstum 3700083887
- VR-Bank Extrazins II 3700459070
- PayPal-Konto
- Degussa-Bank 730025726
- Kreditoren (Rechnungen)

Freistellung

Die bisherige Freistellung von der Körperschaftsteuer gilt weiterhin bis 31.12.2012.

¹ Kaufmännische Grundsätze der ordnungsgemäßen Buchführung

Eröffnungsbuchungen zum 1.1.2011

Nach Abschluss des Vereinsrechnungsjahres 2010 ergaben sich zum 1.1.2011 nachfolgende Eröffnungsbuchungen.

Konto	Bezeichnung	Betrag in €
001	Barkasse	332,05
011	VR-Bank Giro	8.475,22
012	VR-Bank Cash	35.288,82
014	VR-Wachstum	11.001,66
015	VR-Extrazins	20.170,94
031	PayPal	348,11
041	Degussa-Bank	49.138,24

Barkasse

Die Barbelege wurden vollständig geprüft, es ergaben sich keine Beanstandungen.

Konten

Neben der Barkasse unterhielt der Verein im Jahr 2011 folgende Bankkonten:

- **VR-Bank** Girokonto, Unterkonto 011
- **VR-Bank** Cash, Unterkonto 012
- **VR-Bank** Wachstum, Unterkonto 014
- **VR-Bank** Extrazins II, Unterkonto 015
- **PayPal** Internetkonto, Unterkonto 031
- **Degussa-Bank**, Unterkonto 041

Die Salden der Buchführung stimmen mit den Bankauszügen überein; die Anfangsstände 2011 entsprechen den Endständen 2010. Die jeweiligen SBK²- und EBK³-Buchungen wurden ordnungsgemäß durchgeführt.

Die Belege wurden vollständig geprüft. Alle Konten der Buchführung sind in den Jahresabschluss eingeflossen. Nach mündlicher Erklärung des Schatzmeisters existieren keine weiteren Konten auf den Namen des Vereins.

Ausgaben erfolgten satzungskonform und stets nach dem Grundsatz der Wirtschaftlichkeit.

Als Jahresabschluss für das Rechnungsjahr 2011 ergeben sich folgende Bestände:

² Schluss-Bilanz-Konto = Abschlussbuchung eines Geschäftsjahres

³ Eröffnungs-Bilanz-Konto = Erstbuchung eines Geschäftsjahres (Übertragsbuchung)

Bezeichnung	Endbestand in €
Barkasse (001)	20,08
VR-Bank Giro (011)	7.462,13
VR-Bank Cash (012)	43.010,99
VR-Wachstum (014)	11.386,72
VR-Extrazins (015)	20.513,33
PayPal (031)	330,65
Degussa-Bank (041)	49.730,75

Abweichungen

Beim Kauf des TAN-Generators für das Onlinebanking über Ebay wurde der Kaufbetrag von 10,71 € einmal direkt an den Versteigerer überwiesen, anschließend aber versehentlich noch ein zweites Mal als Ersatz einer Barauslage an den Schatzmeister überwiesen. Der Betrag wird vom Schatzmeister zurückerstattet.

Feststellung

Die Umsetzung des Vorschlags der letztjährigen Kassenprüfer, den Buchbestand und die Buchzugänge jährlich zum Kassenprüfungstermin zu erfassen, wurde von den Kassenprüfern kontrolliert. Es wurden keine Unregelmäßigkeiten festgestellt.

Abschluss

Die Anfangs- und Endbestände der Konten stimmen mit den Kontoauszügen, welche lückenlos und vollständig vorhanden sind, überein. Auch die Rechnungen und Journale der Unterkonten sind vollständig. Sämtliche Unterlagen des Rechnungsjahres sind zudem ordentlich aufbewahrt.

Alle Geschäftsfälle wurden sowohl im Buchhaltungsprogramm als auch über die Kontoauszüge und Rechnungen richtig kontiert. Die Buchungen wurden konsistent, richtig und nachvollziehbar ausgeführt.

Dem Schatzmeister *Klaus Höppner* und der Büroleiterin *Karin Dornacher* wird eine vorbildliche Buch- und Kassenführung bescheinigt. Bei allen Geschäftsvorfällen blieben die Vereinsinteressen gewahrt.

Heidelberg, den 25. Februar 2012

Doris Behrendt Martin Wilhelm Leidig

T_EX-Theatertage

DANTE 2012

Roland Geiger

Die Frühjahrstagung und 46. Mitgliederversammlung von DANTE e.V. fand vom 7. bis 9. März 2012 in Leipzig statt. Seit 2002 bin ich Mitglied bei DANTE und hatte noch nie an einer Tagung teilgenommen. Als im Herbst 2011 die Nachricht gepostet wurde, dass die Frühjahrstagung in meiner Heimatstadt Leipzig stattfinden wird, stand für mich fest, diesmal werde ich dabei sein.

Dienstag, 6. März

Vorabend

Wie üblich begann die Tagung mit einem Vorabendtreff, diesmal im »Alfredo«, einem Lokal in der Leipziger Südvorstadt. Das Lokal war bereits gut mit Tagungsteilnehmern gefüllt. Ich fand Platz an einem freien Vier-Mann-Tisch, blieb aber nicht lange alleine: »Ist noch frei? Ich heiße Dominik.« Schon war der erste Kontakt geknüpft. Der Name sagte mir etwas aus den Foren, und so war bei italienischer Küche kein Mangel an Gesprächsstoff vorhanden, außerdem hatten wir ja $\text{\TeX}$ als gemeinsames Hobby.

Auch die weiteren Plätze am Tisch blieben nicht lange leer, mit Harald erweiterten sich die Themen um Linux im Allgemeinen und Ubuntu im Besonderen. Meine Tischnachbarn konnten mir außerdem die Teilnehmer zuordnen, deren Namen mir von der Homepage von DANTE, aus der DTK oder der deutschsprachigen Newsgroup `de.comp.text.tex` bekannt waren.

Mittwoch, 7. März

Die Tagung fand im Lipsius-Bau der Hochschule für Technik, Wirtschaft und Kultur Leipzig (HTWK), in der Karl-Liebknecht-Straße statt. Von mir aus war das mit einer Wochenkarte der LVB bequem zu erreichen. Auch die Gaststätten aller Abendveranstaltungen lagen in Nähe dieser Magistrale im Leipziger Süden.



Abbildung 1: Vorabendtreff im »Alfredo«

Abbildung 2: Hochschule für Technik, Wirtschaft und Kultur (HTWK)



Wir hatten zwei Seminarräume zur Verfügung, einen Pausenraum und einen für die Vorträge. Der Vortragsraum war mit Beamer und Wandtafel ausgestattet, außerdem konnte von den Teilnehmern ein WLAN-Anschluss genutzt werden. Im Pausenraum standen, von le-tex organisiert, während der Tagung Kaffee, Tee, Wasser und Säfte sowie Obst und Kekse bereit. Dort erfolgte auch die Anmeldung zur Tagung.

Der Vormittag begann mit der Registrierung der Teilnehmer, und nun war jeder Dank des Schildes am Schlüsselband namentlich zu erkennen. Es hatten sich nur 24 Teilnehmer vorangemeldet, es gab aber viele Nachmeldungen, sodass zur Mitgliederversammlung 33 Stimmberechtigte anwesend waren. Dazu kamen noch Mitarbeiter von le-tex, dem Mitveranstalter und Sponsor, und natürlich am ersten Tag Studenten der HTWK.

Begrüßung

Die Tagung wurde von Volker RW Schaa (DANTE e.V.) als Vorsitzenden und von Gerrit Imsieke (le-tex) eröffnet.



Abbildung 3: Anmeldung



Abbildung 4: Eröffnung der Tagung

ℒ_{TeX} ist nicht barrierefrei – Anforderungen an ein barrierearmes ℒ_{TeX}

Günter Partosch hielt den ersten Vortrag. Die UN-Behinderten-Konvention und deren Umsetzung in nationales Recht fordert einen barrierefreien Zugang, auch zu Dokumenten. Das betrifft in Deutschland Vorschriften auf Bundes- und Landesebene. Während bei Dokumenten Anforderungen an Schriftform und -größe, Kontrast zwischen Schriftfarbe und Hintergrund einfach zu erfüllen sind, stellen Bilderläuterungen allgemein und insbesondere mit ℒ_{TeX} unüberwindbare Probleme dar. Der Vortrag wies auf etwas hin, das wohl viele von uns noch nie gehört hatten. Anschließend setzte Axel Strübing von le-tex das Thema mit den Problemen bei der praktischen professionellen Umsetzung fort. Er musste für das Ziel »Barrierearmut« sogar Änderungen im ℒ_{TeX}-Quellcode vornehmen.

Einsatz des hyperref-Pakets zum Setzen der PDF-Dokumenteneigenschaften

Thomas Ferber stellte in seinem ersten Vortrag Eigenschaften des hyperref-Pakets vor. Diese sind zwar alle in der Paket-Dokumentation beschrieben; aber die ist sehr umfangreich. Thomas sprach über die Spezialbefehle des Paketes, z. B. die Befehle der `\pdfinfo`-Gruppe (Title, Author, Keywords, ...), der `\pdfcatalog`, in dem beispielsweise der Pagemode beim Öffnen festgelegt werden kann, die Linkbox Farben durch das `\hypersetup` und auf `\label` und `\ref`, womit auch eine Beziehung zum Vortrag über Barrierefreiheit herstellbar war.

Animationen

Herbert Voß zeigte animierte Anwendungen innerhalb einer Beamer-Präsentation. Als Beispiel diente die 3-dimensionale Darstellung eines Torus, um den ein senkrecht geführter zweiter Torus bewegt wurde. Das Beispiel sollte insbesondere die Möglichkeiten verdeutlichen, sich schwierig vorzustellende 3-dimensionale Objekte in Bewegung darzustellen.



Abbildung 5: Günter Partosch hielt den ersten Vortrag



Abbildung 6: Zuhörer am Nachmittag

$\text{\TeX}$ im 21. Jahrhundert – wo wir sind und wo geht es hin?

lautete der Vortrag von Martin Schröder, mit dem die Nachmittagsvorträge begonnen wurden. Er ging auf die Entwicklung von $\text{\TeX}$ 78 über $\text{\TeX}$ 82 bis zur Version 3 im Jahre 1990 ein, die bekanntlich eingefroren wurde. Die Entwicklung durch die Community geht aber weiter und heute werden mit den Distributionen die drei Engines pdf $\text{\TeX}$, X $\text{\TeX}$ und Lua $\text{\TeX}$ ausgeliefert. Seit 1994 nutzen wir $\text{\LaTeX}$ 2 ϵ und sehen auf die Entwicklung von $\text{\LaTeX}$ 3. Eine neuere unabhängige Entwicklung ist Con $\text{\TeX}$ t, die als Alternative zu $\text{\LaTeX}$ nicht unerwähnt bleiben soll. Heute treibt uns die Internationalisierung der Sprachen an. $\text{\TeX}$ 78 beherrschte den ASCII-Zeichensatz (7 Bit), seit $\text{\TeX}$ 90 können 256 Zeichen direkt gesetzt werden (8 Bit), heute stehen Unicode-Zeichensätze auf der Agenda (32 Bit). Um OpenType-Fonts nutzen zu können, brauchen wir freie Fonts in diesem Format. Seit Jahren stehen mit »Latin Modern« und neuerdings mit »TeX Gyre« solche Fonts zur Verfügung. Erstmals in der über 30jährigen Geschichte von $\text{\TeX}$ stehen auch freie Mathematik-Fonts im OpenType-Format zur Anwendung bereit.

Zusammenfassend stellte Martin fest, dass an und für $\text{\TeX}$ weiterhin intensiv weiterentwickelt wird.

biblatex und MIBibTeX

Er hatte diesmal die weiteste Anreise, Jean-Michel Hufflen aus Frankreich. Das Paket biblatex kann mit BibTeX, Biber oder MIBibTeX benutzt werden. Jean-Michel stellte insbesondere sein MIBibTeX und dessen Besonderheiten wie neue Syntax und Typprüfung (z. B. für year) vor. Das Paket ist in Entwicklung, in der Diskussion gab er einen Link zu seiner Homepage an: <http://lifc.univ-fcomte.fr/~jmhufflen/> Dieser gilt so lange das Paket noch nicht auf CTAN abgelegt ist.

Referenzen mit L^AT_EX - eine Übersicht

Die drei Makrobefehle kennt doch jeder auswendig: \label, \ref und \pageref. Das dachten viele, bevor sie den Vortrag von Martin Sievers gehört hatten. Deren Anwendung ist ja durchaus umständlich und eine Eingabe für eine Referenz »siehe \ref auf Seite \pageref« auch nicht so richtig anwenderfreundlich. Um das zu vereinfachen und zu verbessern, gibt es eine Vielzahl von Paketen, allein neun auf CTAN, und Martin stellte deren Möglichkeiten und Anwendungen vor (cleveref, fancyref, varioref, zref, titleref, nameref, ...).

L^AT_EX an der Freien Universität Berlin – ein Erfahrungsbericht

Dieser Vortrag von Herbert Voß sollte den Schlusspunkt hinter den ersten Tagungstag setzen. An der FU Berlin gibt es eine Seminarserie zu L^AT_EX im Bachelor-Studiengang. Eine angepasste Fassung der aktuellen T_EXLive wird auf Linux aufgesetzt. (Angepasst wegen der Zugriffsrechte des T_EX-Standardverzeichnisbaumes.)

Herbert erzählte sehr interessant von der Herangehensweise, den bisher ausschließlich mit der Maus agierenden Studenten ein völlig anderes Betriebssystem und die Kommandozeile sowie die Grundlagen von L^AT_EX beizubringen. Die Teilnehmer sollten anschließend befähigt sein, auf ihren heimischen Laptops ein lauffähiges T_EX-System benutzen zu können.

Abendtreff im »Bayerhaus«

Natürlich kann ein Tagungstag nicht mit dem letzten Vortrag zu Ende sein. Er endet erst mehr oder weniger spät in der Gaststätte. Dort war es möglich, Kontakte zu pflegen oder neue zu knüpfen, den Tag »sacken zu lassen« oder einfach nur zu entspannen. Laut Programm ein »Restaurant«, kennen die Leipziger Studenten das »Bayerhaus« als Mensa und Studentenkneipe, die es auch ist. Billard haben wir dort nicht gespielt. Wir hatten einen eigenen Bereich für uns und einen gemütlichen Abend mit Essen und Trinken direkt vom Tresen.

Donnerstag, 8. März

Der Tag begann mit der Registrierung zur Mitgliederversammlung von DANTE e.V. und dem Empfang der Stimmzettel zur Wahl der neuen Vorstandsmitglieder.

Mitgliederversammlung von DANTE e.V.

Die Tagung wurde durch Volker RW Schaa, Vorsitzender von DANTE e.V., und einem Grußwort von Uwe Kulisch, Dekan der Fakultät Medien an der HTWK Leipzig, in dessen Räumen wir zu Gast waren, eröffnet.



Abbildung 7: Die 46. Mitgliederversammlung von DANTE e.V.

Über die Mitgliederversammlung wird in der DTK ausführlich berichtet, deshalb hier nur soviel: Der langjährige Vorsitzende Volker RW Schaa kandidierte nicht wieder für dieses Amt, wird aber weiterhin als Beisitzer im Vorstand mitarbeiten. Zum neuen Vorsitzenden wurde Martin Sievers gewählt.

An die Wahl schloss sich eine lebhaft, emotional und kontrovers geführte Diskussion zum Antrag »Zeitnahe allgemeine Veröffentlichung von ›Die Technische Komödie‹ im Web« an. Dieser Vorschlag fand keine Mehrheit. Die Mehrzahl der Mitglieder stimmte für einen in der Versammlung eingebrachten Vorschlag, die DTK quartalsweise ein Jahr nach ihrem Erscheinen kostenlos und für alle zugänglich auf der Homepage von DANTE ins Netz zu stellen.

Stammtisch Leipzig (geplant)

Während dieser Vormittagssitzung ließ Jörg Reschke eine Liste herumgehen mit der Frage: »Wer hätte Interesse an einem ›Leipziger Stammtisch‹?« Etliche Teilnehmer fanden die Idee gut. Wer Interesse hat, sollte deshalb die Seite <http://projekte.dante.de/Stammtische/WebHome> beobachten, auf der von Jörg nähere Einzelheiten bekanntgegeben werden.



Abbildung 8: Der amtierende
Vorsitzende Volker RW Schaa



Abbildung 9: Der neugewählte
Vorsitzende Martin Sievers

Vom $\text{\LaTeX}$ -Dokument zum EPUB – ein steiniger Weg

Als Einstieg in den zweiten Teil des Tages verteilte Dominik eine Runde »Leipziger Lerchen«¹. Die Nachmittagssitzung begann mit dem einzigen Gemeinschaftsvortrag der Tagung. Thomas Ferber und Dominik Wagenführ stellten den Weg zu EPUB allgemein und aus ihrer praktischen Erfahrung heraus vor, Thomas aus der Eigenproduktion eines Buches, Dominik von der Erstellung der Online-Zeitschrift »freies Magazin« (<http://www.freiesmagazin.de/>), welches neuerdings ein EPUB mit anbietet.

Ein (La) $\text{\TeX}$ -Lauf erzeugt einen Absatz mit definiertem Font, Unterscheidung und Zeilenumbruch, der unverändert auf Papier oder im PDF ausgegeben wird. In einem E-Book sollen die Schriftart und -größe geändert werden können, wobei diese Änderungen auch zu einem anderen Absatz und damit zu einem geänderten Umbruch führen. Eine Einteilung in Seiten und Kopf- und Fußzeilen ist nicht nötig und auch nicht gewünscht, Hyperlinks dagegen sehr.

E-Books haben unterschiedliche Geometrien, unterschiedliche Auflösungen und können zum einen sowohl Farbe, zum anderen nur Grautöne darstellen. Ergänzend können Weblinks, Sound und Videoclips eingebunden werden. Von den Autoren wurden verschiedene Wege mit unterschiedlichen Paketen ausprobiert. Der Weg über HTML/XML brachte die besten Ergebnisse.

¹ Diese bekannte Leipziger Spezialität bestand bis zum Fangverbot 1876 aus Feldlerchen im Teigmantel. Heute ist sie ein Mürbeteigtörtchen, gefüllt mit einer Marzipanmasse über einer dünnen Marmeladenschicht, mit zwei überkreuzten Teigstreifen verziert.

Database Publishing mit Lua $\TeX$

Patrick Gundlach war mit zwei Vorträgen auf der Tagung vertreten (siehe auch Seite 32), und in beiden ging es um Lua $\TeX$. Im ersten Vortrag stellte Patrick »datengestütztes Publizieren« vor. Dazu entwickelte er im Vortrag die Erstellung eines (mehrsprachigen) Katalogs, dessen Elemente in einer Datenbank abgelegt sind und der mit $\TeX$ gesetzt werden soll. XML wird als Eingabesprache benutzt, das Design hat Layoutregeln zu folgen, die sowohl statisch (z. B. neue Artikelgruppe) als auch dynamisch (z. B. Platz optimieren) sein können. Als Programmiersprache wird Lua benutzt und im speeddata Publisher realisiert.

OpenType-/TrueType-Schriften für Text und Mathematik mit X $\LaTeX$ und Lua $\LaTeX$

Das Nachmittagsprogramm ging weiter mit Lua $\LaTeX$, Herbert Voß hielt den Vortrag »Mathematische Fonts mit Lua $\LaTeX$ «. Seit nunmehr etwas über 30 Jahren werden Schriften im METAFONT-Format, in Text- und Mathematiksschriften getrennt, aber aufeinander abgestimmt, verwendet (Computer Modern). Seit ca. zehn Jahren gibt es diese Textschriften als Type-1- bzw. als OpenType-Variante (Latin Modern), eine analoge Mathematiksschrift fehlte bisher.

Nachdem über 2 000 mathematische Zeichen und ihre Positionen in Unicode standardisiert worden sind, werden aktuell mindestens vier freie Mathematiksschriften entwickelt: »XITS Math«, »Asana Math«, »Latin Modern Math« und »Neo Euler«, dazu kommen die kommerzielle »Cambria Math« von Microsoft und die »Lucida Math«.

Alle auf einem Rechner installierten OpenType-Fonts können sowohl mit X $\TeX$ als auch mit Lua $\TeX$ genutzt werden. Zur Einbindung wird fontspec benutzt. Der Anwender hat damit sowohl deutlich mehr Freiheiten, als er bei Aufruf eines Font-Stylefiles hätte, er hat aber auch die Arbeit und muss Parameter selbst definieren, falls er vom Standard abweicht. Die Mathematiksschriften werden über das Paket mathspec bzw. unicode-math eingebunden.

Das Dokument ist Programm – Anwendung von Lua-Bibliotheken in $\TeX$ -Dokumenten

Mit dem Vortrag von Christof Meigen wurde der Tagungsnachmittag beendet.

Für die Skriptsprache Lua gibt es zahlreiche Bibliotheken, die frei benutzt werden können. Für Lua ist »LuaRocks« (<http://www.lua-rocks.org/>) das, was CTAN für $\TeX$ ist. Mit Lua $\TeX$ und wxLua ist es problemlos möglich, Dokumenten eine grafische Oberfläche zu geben. Mit lua-curl können Webdienste abgefragt werden und mit luasql kann man auf Datenbanken zugreifen.



Abbildung 10: Patrick Gundlach und Christof Meigen mit Vorträgen zu Lua $\TeX$

Als Vorführung ließ Christof von den Teilnehmern Daten in einen im Web abgelegten Evaluationsbogen eintragen. Diese Einträge zeigte er als Online-Abfrage als ausgefüllte Tabelle auf seinem Laptop. Die Abfrage erfolgte mittels Lua-Bibliotheken und wurde mit Lua $\TeX$ formatiert. Von den Teilnehmern wurde der Wunsch geäußert, dass Christof über dieses interessante Gebiet einen Artikel in der DTK veröffentlichen möge. Die Präsentation liegt inzwischen auf <http://www.dante.de/>.

Tagungssessen im »Pilot« (Centraltheater)

Als letzter wichtiger Programmpunkt stand das Tagungssessen an. »Centraltheater«, wo ist denn das? Als »Café Schauspielhaus« kennt es jeder Leipziger, zentral gelegen und vom Tagungsort aus bequem zu erreichen.



Abbildung 11: Abendveranstaltung im »Pilot« des Centraltheaters

Für DANTE war der Raum in der ersten Etage reserviert und ein Menü zur freien Auswahl vorgemerkt. Hier konnten wir in entspannter Atmosphäre den Tag ausklingen lassen und interessante Gespräche weiterführen. Ich benutze $\text{\LaTeX}$ seit dem Jahre 2000 in der Version $\text{\LaTeX}$ 2_ε. Da war es schon interessant von Zeiten zu hören, als $\text{\TeX}$ noch auf Magnetband aus Stanford kam. Es zeigte sich auch, dass viele aktive Mitglieder unseres Vereins wirklich Nutzer der ersten Stunde sind, ohne deren Einsatz eine solche Tagung nicht stattfinden würde.

Es war immer Zeit, die Plätze und die Gesprächspartner zu wechseln. So kam ich mit Harald Vajkonny über semitische Sprachen ins Gespräch. Als Physiker bin ich eher mit dem Formelsatz vertraut, die Anforderungen eines Philologen sind interessant und für mich so ganz anders.



In der Vormittagssitzung waren seine Vorstellungen zur Sprache gekommen, hebräische Texte mit kritischem Fußnotenapparat mit $\text{\TeX}$ setzen. Das gezeigte Beispiel bestand aus einem zentralen Textblock der Größe A6 auf einer A4-Seite, um den die Kommentare angeordnet sind. Dazu kommen bei einem hebräischen Text die Schreibrichtung von rechts nach links, Fonts, die so noch nicht existieren, und ein Blocksatz, der feste Wortabstände fordert, den Dehnungsausgleich über eine Verlängerung bzw. Verkürzung der waagerechten Buchstabenteile erzeugt (nichts mit glue). Es gibt für $\text{\TeX}$ offenbar zunehmend Herausforderungen aus dem philologischen Bereich.

Freitag, 9. März

Lua $\text{\TeX}$ Interna

Der Freitagmorgen begann, wie der Donnerstag endete, mit einem Vortrag zu und über Lua $\text{\TeX}$. Gehalten wieder von Patrick Gundlach, sein Name taucht auch im Web und im Blog zu den Fragen um diese neue Engine immer wieder auf.

Der Vortrag zeigte, wie sich Lua $\text{\TeX}$ auf der Ebene von Nodelisten programmieren lässt. Das Beispiel »lua-visual-debug – Das Unsichtbare sichtbar machen« stand bereits ein paar Tage im Web, aber erst Patricks Vortrag und seine Erläuterungen

schafften (bei mir) Klarheit über den Mechanismus. Das Programm klinkt sich in die Nodelisten ein und macht beispielsweise gedehnte Längen, Unterschneidungen, Trennstellen, ... grafisch sichtbar. Anhand der Dokumentation zu `luatexref-t` zeigte er noch, welche Möglichkeiten es für Programmierer gibt, sich mit Lua in den $\TeX$ -Code einzuhängen.

$\TeX$ Online Communities

Stefan Kottwitz hielt den Vortrag über mögliche Kommunikationsformen im Netz. Die Diskussionsforen begannen mit Mailing-Listen. Um 1990 entstand die Usenet-Gruppe `comp.text.tex`, seit 1996 ist die deutsche Gruppe `de.comp.text.tex` aktiv. Es existieren auch entsprechende Gruppen französisch oder spanisch sprechender Anwender. Weithin bekannt ist <http://goLaTeX.de/>. Aktuell gibt es Web-Foren und Blogs, die eine zahlreiche passive und aktive Gemeinde haben. Heute werden auch Informationen über Twitter oder auf Facebook ausgetauscht.

Stefan betreibt den $\TeX$ blog »Typography with $\TeX$ and $\LaTeX$ «. Hier fand ich im »TUG 2011 Report« den Verweis auf die ersten und neuesten Informationen der Konferenz in Kerala. Und der »DANTE 2012 Report« steht auch schon seit dem 11. März, einen Tag nach Tagungsende, im Blog.

Er ging weiter allgemein auf Q&A Sites ein, und hier besonders auf <http://tex.stackexchange.com/> (`TeX.sx`), den Community Blog, den er mit anderen moderiert. Der Blog hat aktuell mehr als 10 000 registrierte Nutzer, die auf ca. 16 000 Fragen mehr als 28 000 Antworten gegeben haben. Interessant waren weiter seine Ausführungen zur Selbstorganisation des Blogs durch die Teilnehmer. Den interessierten Lesern kann der Einstieg in die Blogwelt über <http://planet.dante.de/> empfohlen werden.

Bild- und Briefautomation mit $\LaTeX$

Uwe Ziegenhagen zeigte in einer Vorführung nicht nur die Erstellung von Briefumschlägen mit $\LaTeX$, er führte auch vor, wie die Grafik einer elektronisch erworbenen Briefmarke zugefügt werden kann. Diese Leistung der Post erwirbt man als Bogen. Als Erweiterung kann sich Uwe die Verwaltung der gekauften und noch vorhandenen Briefmarken in einer Datenbank vorstellen.

Projektbericht »Libertine Open Fonts Projekt«

»Libertine«, auf diesen Vortrag war ich gespannt. Ich finde die Schrift wunderschön und benutze sie als Alternative zur Standardschrift mit $\LaTeX$, mit $\pdf\LaTeX$, um genau zu sein. Philipp Poll, der den Vortrag hielt, entwickelt die freie Schrift »Libertine«. Sie kann sowohl unter Linux, als auch unter Windows mit den freien oder kommerziellen Office-Paketen genutzt werden.

Von Michael Niedermair wird das Paket `libertineotf` zum Einbinden der Fonts in $\text{Xe}_\text{L}^{\text{A}}\text{TeX}$ und $\text{Lua}_\text{L}^{\text{A}}\text{TeX}$ gepflegt, Bob Tennent führt das Paket in der ursprünglichen Version als `libertine-legacy` für die Nutzung in $\text{pdf}_\text{L}^{\text{A}}\text{TeX}$ fort.

In Philipps Vortrag ging es nicht um die vielen Schriftschnitte der Serifenschrift (mit der »Biolinum« gibt es auch eine zur »Libertine« passende Groteske), sondern um den Bericht über die von DANTE geförderte Entwicklung einer passenden »Nichtproportionalen Schrift«.

Er stellte die Anforderungen an einen Monospace Font vor und die Schwierigkeiten, schmal laufende (i) und weit laufende (m) Buchstaben auf eine gemeinsame Breite, aber immer noch optisch ansprechend zu gestalten. Als Vergleich zeigte er Beispiele mit seinem Schriftentwurf, und parallel dazu mit der »Courier« gesetzt.

Ein Aufruf von `Fontforge` gab einen Einblick in die Arbeit eines Fontgestalters und die vielen zu beachtenden Parameter eines OpenType-Fonts.

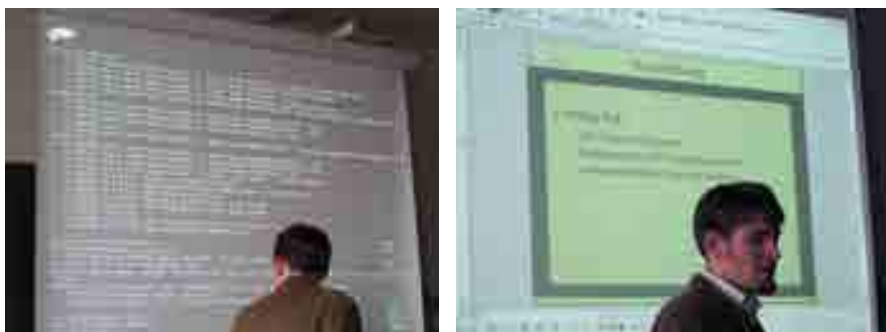


Abbildung 12: Philipp Poll und die »Libertine«

In der Diskussion wurde von den Teilnehmern an Philipp der Wunsch nach einem Monospaced-Condensed-Font herangetragen, um Listings mit einer Zeilenbreite von 80 Zeichen ohne Umbruch setzen zu können. Philipp zeigte sich sowohl für diesen Wunsch, als auch für eine »Libertine-Math« aufgeschlossen, für die er mangels eigener Vorstellungen definitiv Unterstützung benötigen würde.

Ich würde gerne einmal einem Vortrag über die »Libertine« als Gesamtpaket hören. Im Augenblick kann ich allen an diesem Font Interessierten nur empfehlen, einmal die Homepage <http://www.linuxlibertine.org/> zu besuchen.

Das $\text{T}_\text{E}\text{X}$ in le-tex. $\text{T}_\text{E}\text{X}$ im Unternehmenseinsatz

Freitagnachmittag. Den letzten Vortrag der Leipziger DANTE-Tagung hielt Martin Kraetke von der Firma le-tex über den Einsatz von $\text{T}_\text{E}\text{X}$ in der Firma.



Abbildung 13: »TeX bei le-tex« beendete die Vortragsserie

Wie der Firmenname verrät, spielt T_EX eine wichtige Rolle beim professionellen Zeitschriften- und Buchsatz im Unternehmen. Es gibt immer noch Verlage wie Springer, die für ihren Einsatz dieses Satzsystems bekannt sind. Es muss sich aber gegen die Konkurrenz der DTP-Programme wie »Adobe InDesign« oder »Quark Express« behaupten. Da spielt oft die Firmenphilosophie des Kunden die entscheidende Rolle. Zunehmend wird der Einsatz von »XML First« für zukünftige elektronische Bücher gefordert. Martin gab Beispielpublikationen zur Ansicht für die Tagungsteilnehmer in Umlauf.

Verabschiedung und Abschluss

Damit war die Tagung (fast) zu Ende.

Der neue Vorsitzende von DANTE e.V. Martin Sievers sprach die Schlussworte, den Dank an alle Vortragenden und verabschiedete die Teilnehmer. Sein besonderer Dank galt der Firma le-tex für deren Unterstützung der Tagung. Er konnte, wie wir alle, auf erfolgreiche und interessante Tage in Leipzig verweisen. Dieser positiven Einschätzung schloss sich Gerrit Imsieke an. Es standen für diesen Nachmittag und Abend noch ein Museumsbesuch, eine Firmenbesichtigung und ein Abendtreff an und am Sonntabend noch eine Stadtführung.

Führung durch das Museum für Druckkunst Leipzig

Das »Museum für Druckkunst Leipzig« ist für alle an klassischen Druckerzeugnissen und ihrer Herstellung Interessierten einen Besuch wert. Leipzig, ist eine Buchstadt mit seinen Großverlagen wie Reclam, Brockhaus, Insel oder Kiepenheuer, aber auch mit seinen unzähligen mittleren und kleinen Verlagen, die nach 1945 die Stadt verlassen haben. Doch auch in der Zeit der DDR blieb Leipzig eine Stadt

Abbildung 14: Schlussworte von Martin Sievers



des Buchdrucks, mit seinen Verlagen, mit der »Deutschen Bücherei« (Deutsche Nationalbibliothek sei 1913), mit dem »Institut für Graphische Technik« und mit der »Hochschule für Grafik und Buchkunst«.

Als zwischen 1980 und 1990 die elektronische Satztechnik Einzug hielt, gab es noch genug alte funktionsfähige Satz- und Druckmaschinen in der Stadt. Eine Stiftung kaufte sie auf und stellt sie seitdem im von uns besuchten Museum aus. Es ist eine Kombination aus Museum und Werkstatt, Fachleute führten während unserer Führung die Maschinen vor. Außerdem werden die Handsatz- und Druckmaschinen immer noch von Künstlern für lithografische Arbeiten genutzt.

Dr. Susanne Richter, die Direktorin des Museums, führte uns durch die Ausstellung. Dabei konnten wir die Druck- und die Satzmaschinen in Aktion erleben.

Besichtigung der Firma le-tex

Einmal im Leipziger Stadtteil Lindenau, gingen wir zu Fuß zur Firma le-tex, um sie auf einem kurzen Rundgang zu besichtigen.

Abendtreff im »Nora Roman« (Schaubühne Lindenfels)

Letzter Punkt des Tagungsprogramms und letzter Spaziergang durch Lindenau war dann der Weg zur Abendgaststätte. Für mich war es nur ein Aufenthalt zum Abendessen. Eine Tagung ist auch eine anstrengende Veranstaltung.

Sonnabend, 10. März

Stadtführung und Mittagessen in »Auerbachs Keller«

Treffpunkt war um 10 Uhr auf dem Augustusplatz vor dem »Gewandhaus zu Leipzig«, dem dritten Konzerthaus in seiner 231jährigen Geschichte. Gerrit Imsieke führte uns durch die Innenstadt, vorbei an der »Nikolaikirche«, dem Ausgangspunkt



Abbildung 15: Hier stehen wirklich interessante Bücher



Abbildung 16: Der Verfasser mit der einzigen Teilnehmerin



Abbildung 17: Treffpunkt »Gewandhaus« zur Stadtführung

der Leipziger Montagsdemonstrationen, durch die Passagen dieser so typischen Bauweise für die Innenstadt hin zum »Alten Markt« mit dem »Alten Rathaus« (im 14. Jahrhundert von Hieronymus Lotter umgebaut), einem wunderschönen Renaissancebau.

Weiter führte der Weg durch das Barfußgässchen, vorbei am »Coffe Baum« – hier spielte schon J. S. Bach, der das Getränk in seiner Kaffeeantate lobte (BWV 211) – weiter die Hainstraße zum nordwestlichen Ausgang aus der Innenstadt, mit Blick auf die Pleiße ins Waldstraßenviertel, einem der baulich besterhaltensten zusammenhängenden Vierteln Europas mit Gründerzeithäusern.

Zurück ging es im Bogen durch den Clara-Zetkin-Park, am Internat der Thomaschule (die Thomaner feierten am 20. März den 800. Jahrestag ihrer Gründung) vorbei mit dem Ziel »Neues Rathaus«. Von da ging es weiter mit Blick in die »Tho-

maskirche«, um dann auf kurzem Wege im durch Goethes »Faust« weltberühmten »Auerbachs Keller« zum Mittagessen und damit zur endgültig letzten Veranstaltung der 46. Tagung von DANTE e.V. zu kommen.



Abbildung 18: Mittagessen in »Auerbachs Keller«

Und mit einem letzten Abschied nach dem Mittagessen war die Tagung nun wirklich beendet.

Schlussbemerkung

Ich habe zum ersten Mal an einer Veranstaltung von DANTE e.V. teilgenommen und werde das wieder tun. Den Besuch solch einer Tagung kann ich nur weiterempfehlen.

Die kleinen Gaststätten waren unserem Ansturm in den Mittagspausen kaum gewachsen. Bei zukünftigen Tagungen sollte diese Pause deshalb auf zwei Stunden verlängert werden. Die Pausenversorgung im Hause, die Abendveranstaltung und die Organisation der Veranstaltung kann ich nur loben.

Im vorliegenden Tagungsbericht habe ich meine persönlichen Eindrücke geschildert. Ich bin bewusst auf alle Vorträge eingegangen, um den Lesern der DTK einen Eindruck vom Ablauf der Leipziger Veranstaltung zu geben und sie neugierig auf weitere Informationen zu machen. Viele der Vorträge sind als Präsentationen, erstellt mit der $\text{\LaTeX}$ -Klasse beamer, auf der Homepage von DANTE (<http://www.dante.de/events/dante2012/Programm/Vortraege.html>) zu finden. Bei den Vortragenden bitte ich um Nachsicht, wenn ich Teile ihrer Präsentation nicht so verstanden und wiedergegeben habe, wie sie gedacht waren.

Ich möchte mich hiermit bei allen bedanken, die bei und für DANTE e.V. tätig sind.

Bretter, die die Welt bedeuten

Das Paket xparse – Dokumentenmakros auf Basis expl3

Marco Daniel

Das Schreiben größerer Arbeiten in $\text{\TeX}$ führt bei jedem Autor schnell dazu, sich eigene Makros und Umgebungen zu definieren. Die bekannteste Methode für den Endanwender ist die Nutzung der Makros `\newcommand` oder `\newenvironment`. Das Paket xparse stellt Möglichkeiten bereit, Makros zu definieren, bei denen der Anwender verschiedene optionale Argumente, Sternversionen oder Kombinationen von Makros kreieren kann.

Einleitung

Das Paket xparse stellt eine auf expl3 basierende Möglichkeit dar, Dokumentenanweisungen zu definieren [3]. Ein Parser (engl. to parse, »analysieren«, bzw. lateinisch pars, »Teil«; im Deutschen gelegentlich auch »Zerteiler«) ist ein Programm, das in der Informatik für die Zerlegung und Umwandlung einer beliebigen Eingabe in ein für die Weiterverarbeitung brauchbares Format zuständig ist. Häufig werden Parser eingesetzt, um im Anschluss an den Analysevorgang die Semantik der Eingabe zu erschließen und daraufhin Aktionen durchzuführen (<http://de.wikipedia.org/wiki/Parser>).

Die Entwicklung des Paketes xparse ist noch nicht abgeschlossen. So gibt es Anweisungen, die als experimentell gelten. Die hier vorgestellten Möglichkeiten sind aber stabil und können durch den Endanwender genutzt werden. Gleichzeitig gibt es die Zusage, dass sich die Funktionsnamen nicht ändern werden.

Wie bereits angesprochen, stellt das Paket Alternativen zu `\newcommand` bereit. Natürlich sind auch Definitionen von Umgebungen möglich, die aber hier nicht näher erläutert werden sollen. Es sei aber so viel gesagt, dass die dort vorgestellte Argumentenspezifikation äquivalent zu den hier vorgestellten ist.

Neben den Standardverhalten der $\text{\LaTeX} 2_{\epsilon}$ -Makros, wie `\newcommand`, stellt das Paket eine signifikante Verbesserung der Funktionalität bereit. Es wird, wie üblich, mittels `\usepackage{xparse}` eingebunden.

Definition neuer Funktionen

Zur Definition von Nutzermakros stehen folgende Anweisungen zur Verfügung:

```
1 \DeclareDocumentCommand <Funktionsname> { Argumente } { Code }
2 \NewDocumentCommand   <Funktionsname> { Argumente } { Code }
3 \RenewDocumentCommand <Funktionsname> { Argumente } { Code }
4 \ProvideDocumentCommand <Funktionsname> { Argumente } { Code }
```

Die Unterschiede zwischen den einzelnen Anweisungen sind im Folgenden aufgelistet:

`\DeclareDocumentCommand` definiert eine neue Anweisung. Sollte bereits ein gleichnamiges Makro existieren, so wird es ohne Warnung überschrieben. Es ist somit mit der $\text{\TeX}$ -Primitiven `\def` vergleichbar.

`\NewDocumentCommand` definiert eine neue Anweisung. Sollte bereits ein gleichnamiges Makro existieren, wird eine Fehlermeldung ausgegeben. Es ist somit mit dem $\text{\LaTeX}$ -Kommando `\newcommand` vergleichbar.

`\RenewDocumentCommand` redefiniert eine neue Anweisung. Sollte kein gleichnamiges Makro existieren, wird eine Fehlermeldung ausgegeben. Es ist somit mit dem $\text{\LaTeX}$ -Kommando `\renewcommand` vergleichbar.

`\ProvideDocumentCommand` definiert eine neue Anweisung, falls kein gleichnamiges Makro existiert. Es ist daher mit dem bekannten $\text{\LaTeX}$ -Befehl `\providecommand` vergleichbar.

Mit Hilfe der genannten Funktionen lassen sich neue Makros definieren. Entgegen der herkömmlichen Makrodefinition nutzt $\text{\LaTeX} 3$ einen alternativen Algorithmus zur Auszeichnung der Argumente. Diese Möglichkeit wird im nächsten Abschnitt vorgestellt.

Argumentenauszeichnung

Die Definition von Funktionen mit `\newcommand` oder `\def` sind ausführlich in [1] erläutert. Üblicherweise werden Anweisungen in der folgenden Form definiert:

```
1 \newcommand\foo[1]{ Code }
2 \def\foo#1{ Code }
```

Hierbei kann die Anzahl der Argumente variieren. Eine Limitierung dieser Vorgehensweise ist vermutlich schon jedem untergekommen. Beispielsweise ist die Definition eines Makros mit zwei optionalen Argumenten für den Anfänger alles

andere als leicht. Es gibt Pakete wie `twoopt`, die dies erleichtern, jedoch wird eine Anweisung mit der Syntax `\foo[Opt1]{Pflicht}[Opt2]` schon schwieriger. Dies ist nur eine Einschränkung, die durch das Paket `xparse` eliminiert wird.

Um die Begrenzung in der Zahl der Argumente aufzuheben, nutzt `xparse` keine Ziffern für die Definition der Argumente, sondern Buchstaben. Diese werden später noch vorgestellt. Wie bereits in $\TeX$ oder $\LaTeX$ wird hierbei zwischen verpflichtenden und optionalen Argumenten unterschieden.

Die Umsetzung der *long*-Argumente ist ebenfalls optimiert worden. Um dies mittels `xparse` umzusetzen, kann *jedes einzelne* Argument mit dem Präfix `+` versehen werden, wodurch die Steuerung durch den Nutzer besser umgesetzt wird. Wird das Präfix nicht gesetzt, sind keine Umbrüche in den Argumenten zulässig.

Das Paket `xparse` stellt insgesamt 14 Argumentenbezeichner bereit, wobei sechs von ihnen zur Kategorie der Pflicht-Argumente zählen. Ohne an dieser Stelle eine Wertung der Relevanz abzugeben, werden alle Argumentenbezeichner aufgelistet und erläutert. Zu ausgewählten Argumentenauszeichnern wird ein Beispiel präsentiert, das das Verhalten veranschaulicht. Hierbei wird folgende Syntax zur Unterstützung genutzt: Die definierte Funktion hat den Namen `Arg` und wird jeweils um den entsprechenden Argumentenauszeichner ergänzt.

Verpflichtende Argumente

- m** Ein Standardargument, das ein einzelnes Token oder eine geklammerte Tokenliste sein kann. Es handelt sich hierbei um ein normales $\TeX$ -Argument, wie es bei `\newcommand` Anwendung findet. Beispiel:

```
1 \NewDocumentCommand \Argm { m }
2   {\textcolor{gray}{#1}}
3 \Argm{Die \TeX nische Komödie!}
```

Die $\TeX$ nische Komödie!

- l** Ein Argument, das alles übernimmt, bis das erste öffnende Gruppentoken auftritt. In $\LaTeX$ ist es normalerweise eine Klammer `{`:

```
1 \NewDocumentCommand \Argl { l }
2   {\textcolor{gray}{#1}}
3 \Argl Die \TeX nische Kom{ödie!}
```

Die $\TeX$ nische Komödie!

- r** Ein begrenztes Argument, wobei die Begrenzungstoken durch den Nutzer angegeben werden. Die Syntax ist `r<token1><token2>`. Soll beispielsweise das verpflichtende Argument in runden Klammern übergeben werden, ist die Deklaration `r()`. Sollte das öffnende Token fehlen, so wird ein Fehler generiert und im Text erscheint `\NoValue`:

```

1 \NewDocumentCommand \Argr { r<> }
2   {\textcolor{gray}{#1}}
3 \Argr<Die \TeX nische> Komödie!

```

Die $\text{\TeX}$ nische Komödie!

- R** Verhält sich wie der Typ `r`, allerdings mit dem Unterschied, dass ein Defaultwert anstelle von `\NoValue` übergeben wird. Das heißt, die Syntax ist `R<token1><token2><default>`.
- u** Liest als Argument alles ein, bis ein/e vorgegebene/s Token/-liste erkannt wird, wobei das begrenzende Token nicht mehr zum Argument zählt. Die Syntax ist `u<token(s)>`:

```

1 \NewDocumentCommand \Argu { u{n} }
2   {\textcolor{gray}{#1}}
3 \Argu Die \TeX nische Komödie!

```

Die $\text{\TeX}$ Xische Komödie!

- v** Liest die Argumente als *verbatim*-Material zwischen zwei definierten Token. Vergleiche zum $\text{\LaTeX}$ 2 _{ϵ} -Kommando `\verb` sind zulässig, aber stimmen nicht hundertprozentig, wie in späteren Beispielen deutlich wird. Die Syntax ist `v<token1><token2>`, wobei folgende Token nicht erlaubt sind: `%`, `\`, `{`, `}`, `^` oder `_`. Eine Begrenzung durch `{ }` ist hierbei zulässig. Eine weitere Limitierung besteht in der Anwendung. Eine Funktion mit einem *verbatim*-Argument kann nicht innerhalb einer anderen Funktion wie `\footnote` genutzt werden.

Optionale Argumente

- o** Einem optionalen Standardargument in eckigen Klammern, das den Wert `\NoValue` liefert, sollte kein optionales Argument übergeben werden. Sollten die eckigen Klammern des optionalen Argumentes gegeben sein (egal, ob ein Token enthalten ist oder nicht), so wird nicht `\NoValue` übergeben:

```

1 \NewDocumentCommand \Argo { o }
2   {\textcolor{gray}{#1}}
3 \Argo Die \TeX nische Komödie!\\
4 \Argo[] Die \TeX nische Komödie!\\
5 \Argo[NEU] Die \TeX nische Komödie!

```

-NoValue-Die $\text{\TeX}$ nische Komödie!
 Die $\text{\TeX}$ nische Komödie!
 NEU Die $\text{\TeX}$ nische Komödie!

- O** Verhält sich wie der Typ `o`, erlaubt aber das Setzen eines Defaultwertes. Die Syntax ist `O<default>`.
- d** Verhält sich wie der Typ `o`, erlaubt aber das Setzen der Begrenzungstoken. Die Syntax ist `d<token1><token2>`.
- D** Verhält sich wie der Typ `O`, erlaubt aber das Setzen der Begrenzungstoken. Die Syntax ist `D<token1><token2><default>`.

- s Testet, ob die Sternvariante der Funktion genutzt wird. Der Typ liefert entweder den Wert `\BooleanTrue` oder `\BooleanFalse`, der mittels `\IfBooleanTF` abgefragt werden kann.
- t Ist die Basis des optionalen Argumententyps `s`, mit dem Unterschied, dass das optionale *einzelne* Token selbst bestimmt werden kann. Die Syntax ist `t<token>`.
- g Dieser Argumententyp verhält sich wie der Typ `o`. Hierbei wird das optionale Argument nicht in eckigen Klammern eingelesen, sondern in $\TeX$ -Gruppentoken `{ }`:

```

1 \NewDocumentCommand \Argg { g }
2   {\textcolor{gray}{#1}}
3 \Argg Die \TeX nische Komödie!\\
4 \Argg{} Die \TeX nische Komödie!\\
5 \Argg{NEU} Die \TeX nische Komödie!
```

-NoValue-Die $\TeX$ nische Komödie!
 Die $\TeX$ nische Komödie!
 NEU Die $\TeX$ nische Komödie!

- G Verhält sich wie der Typ `g`, erlaubt aber das Setzen eines Defaultwertes. Die Syntax ist `G<default>`.

Die Vielzahl der verschiedenen Typen wirkt auf den ersten Blick erdrückend. Die Nützlichkeit wird an einfachen Beispielen im übernächsten Abschnitt verdeutlicht. Zunächst werden die bereits erwähnten Makros `\NoValue` und `\BooleanTrue`/`\BooleanFalse` vorgestellt.

Implementierte Tests des Paketes xparse

Die speziellen Makros `\NoValue` und `\BooleanTrue`/`\BooleanFalse` sind bereits bei der Vorstellung der Argumentenauszeichnung eingeführt worden. Die Bedeutung erschließt sich aus der Namensgebung. `\NoValue` steht für die fehlende Übergabe eines Argumentes und `\BooleanTrue`/`\BooleanFalse` setzt eine entsprechende Abfrage auf den Wert *Wahr* oder *Falsch*.

Um die entsprechenden Werte abzufragen, bietet `xparse` die Anweisungen `\IfNoValueTF` und `\IfBooleanTF`. Die Syntax der Tests ist identisch.

```

1 \IfNoValueTF{<Argument>}{<Wahr>}{<Falsch>}
2 \IfBooleanTF{<Argument>}{<Wahr>}{<Falsch>}
```

Neben den vorgestellten zwei Abfragen sind Varianten verfügbar, die jeweils nur den Wahr- oder Falschteil als verpflichtendes Argument haben. Somit erweitert sich die Liste der Tests zu

```

1 \IfNoValueT{<Argument>}{<Wahr>}
2 \IfNoValueF{<Argument>}{<Falsch>}
3 \IfBooleanT{<Argument>}{<Wahr>}
4 \IfBooleanF{<Argument>}{<Falsch>}
```

Verarbeitung der Argumente

Die Übergabe der Argumente mittels *Buchstaben* und deren Bedeutung wurde im vorangegangenen Abschnitt erläutert. Die gezeigten Beispiele geben einen ersten Einblick in die Verarbeitung der Argumente. Bevor auf das Thema der Verarbeitung eingegangen wird, noch ein wichtiger Hinweis. Wie bereits in $\TeX$, ist das Maximum der Argumente auf neun begrenzt.

Um die einzelnen Argumententypen zu verarbeiten, wird auf die $\TeX$ -Syntax zurückgegriffen. Das bedeutet, jedem Argument ist entsprechend der Chronologie der *Buchstaben* ein $\#<\text{Ziffer}>$ zugeordnet. Ein einfaches Beispiel soll dies verdeutlichen. Nehmen wir an, wir wollen eine Anweisung mit zwei optionalen und einem verpflichtenden Argument definieren, wobei wir die Syntax $\backslash\text{Ueberschrift}[\text{Opt1}]\{\text{Pflicht}\}[\text{Opt2}]$ fordern. Basierend auf den gewonnenen Erkenntnissen der vorangegangenen Ausführungen ist dies lösbar:

```

1 \DeclareDocumentCommand \Ueberschrift { O{\bfseries} m O{black} }
2 {%
3   Meine~Überschrift~ist
4   \begin{center}%
5     \color{#3} % Farbe der Überschrift -- zweites optionales Argument
6     #1         % Schriftattribut -- erstes optionales Argument
7     #2         % Eingabe -- erstes (einziges) verpflichtendes Argument
8   \end{center}%
9 }
10
11 Dieser Code führt bei der Eingabe von \Macro{Ueberschrift[\textbackslash
12   itshape]\{Ein Beispiel\}} zu:
13 \Ueberschrift[\itshape]{Ein Beispiel}
14 Wohingegen die Eingabe \verb\Ueberschrift{Ein Beispiel}[gray!80]+ zu
   folgender Ausgabe führt:
   \Ueberschrift{Ein Beispiel}[gray!80]
```

Dieser Code führt bei der Eingabe von
 $\backslash\text{Ueberschrift}[\text{\itshape}]\{\text{Ein Beispiel}\}$ zu: Meine Überschrift ist
Ein Beispiel

Wohingegen die Eingabe $\backslash\text{Ueberschrift}\{\text{Ein Bei-}$
 $\text{spiel}\}[gray!80]$ zu folgender Ausgabe führt: Meine Überschrift ist
Ein Beispiel

Abschließende Beispiele

Ein erstes Beispiel haben wir im vorangegangenen Abschnitt gesehen. An diesem Punkt sollte deutlich sein, wie die Deklaration von Argumenten funktioniert und wie diese verarbeitet werden. An dieser Stelle noch zwei kleinere Beispiele.

Die Syntax der Funktionsdeklaration ist bereits im Vorfeld erläutert worden. Hierbei wird die Frage aufgeworfen, wie eine Funktion ohne Argument definiert wird. Dies ist durch ein leeres zweites Argument der entsprechenden Funktion gegeben.

```
1 \NewDocumentCommand \Info { }
2 { \par\noindent Hier~beginnt~die~Information\par}
```

Zum Schluss noch ein kombiniertes Beispiel, das die wichtigsten Argumentenbezeichner nutzt.

Nehmen wir an, wir benötigen eine Anweisung, deren Argument im Text formatiert werden soll. Gleichzeitig soll eine Sternvariante existieren, die das Wort in den Index schreibt und bei Bedarf zusätzlich durch ein zweites optionales Argument mit den Begrenzern <> formatiert. Insgesamt sind also vier Argumentenbezeichner zu nutzen. Entsprechend der Wünsche sind diese s, o, d und m.

```
1 \NewDocumentCommand \SpezialWort { s O{\itshape} d<> m}
2 {%
3   {%#2{#4}}%           % Ausgabe des formatierten Argumentes
4   \IfBooleanT #1        % Teste ob Sternvariante
5   {%                    % Wahr Teil
6     \IfNoValueTF{#3}    % Teste Formatierung Indexeintrag
7     {%                  % Wahr Teil
8       \index{#4}%
9     }%
10    {%                  % Falsch Teil
11      \index{#4|#3}%
12    }%
13  }%
14 }
```

Mit Hilfe der neuen Funktion `\SpezialWort` ist die folgende Eingabe möglich, wobei die Sternvarianten einen Eintrag im Index erzeugen:

```
1 Text \SpezialWort{Index} Text
2 \clearpage
3 Text \SpezialWort*{Index} Text
4 \clearpage
5 Text \SpezialWort*[\bfseries]{Index} Text
6 \clearpage
7 Text \SpezialWort*[\bfseries]<textit>{Index} Text
```

Eine noch nicht vorgestellte Möglichkeit des Paketes ist die Funktion `\SplitList` in Kombination mit der *noch* experimentellen Funktion `\ProcessList`. Diese gehören zu den Erweiterungen, die das Paket bietet. Die Namen lassen schon einen Rückschluss auf die Bedeutung zu. Die Funktion `\SplitList` trennt eine gegebene Liste an einem definierten Token. Die daraus resultierende Verarbeitung erfolgt hierbei durch die Funktion `\ProcessList`, die jedes Element als *Schleife* abarbeitet. Um die Nützlichkeit zu demonstrieren, werden zusätzlich $\LaTeX$ 2_ε Anweisungen genutzt.

Dies zeigt gleichzeitig, dass es keine Einschränkungen in der Zusammenarbeit mit $\text{\TeX}$ 2_ε gibt, obwohl das Paket *xparse* auf Basis von *expl3* programmiert wurde.

```

1 \newif\ifnameeven \nameevenfalse
2 \NewDocumentCommand \Listenverarbeitung%
3   { > { \SplitList { , } } m }%
4   {%
5     \ProcessList {#1}{\VerarbeiteListe}%
6     \nameevenfalse%
7   }
8 \NewDocumentCommand \VerarbeiteListe%
9   { m }%
10  {%
11    \ifnameeven
12      \textbf{#1}\quad%
13      \nameevenfalse%
14    \else
15      \textit{#1}\quad%
16      \nameeventrue%
17    \fi%
18  }
19
20 Mit Hilfe der Funktion \texttt{Listenverarbeitung}\ lässt sich folgendes
21 Ergebnis erzielen:
22 \begin{center}
23 Ich möchte mich bei folgenden Personen bedanken:\\
24 \Listenverarbeitung{Frank Mittelbach, Joseph Wright, Herbert Voß, Rolf
25 Niepraschk}
26 \end{center}

```

Mit Hilfe der Funktion *Listenverarbeitung* lässt sich folgendes Ergebnis erzielen:
 Ich möchte mich bei folgenden Personen bedanken:
Frank Mittelbach **Joseph Wright** *Herbert Voß* **Rolf Niepraschk**

Anmerkungen

Das Paket beginnt seinen Siegeszug langsam, aber sicher. Die Häufigkeit der Empfehlung ist besonders auf <http://tex.stackexchange.com> spürbar. Das Paket selbst und die dazugehörigen Anwendungen sind, soweit nicht anders in der Dokumentation angemerkt, stabil.

Eine persönliche Anmerkung sei gestattet. Als ich zum ersten Mal von dem *verbatim*-Argument las, war ich begeistert. Es stellte sich allerdings heraus, dass neben der genannten Einschränkung der Schachtelung eine weitere existiert, die ich hier nicht vorenthalten möchte. Wir nutzen ein einfaches Beispiel.

```

1 \NewDocumentCommand\MyVerb{v}{\texttt{\color{gray!70}#1}}

```

Das Kommando arbeitet wie das $\text{\LaTeX 2}_{\epsilon}$ -Makro `\verb`, allerdings erlaubt es eine flexiblere Formatierung. So erzeugt die Anweisung `\MyVerb|\LaTeXe|` die Ausgabe `\LaTeXe`. Auf den ersten Blick funktioniert es problemlos. Allerdings gibt es Probleme bei Ligaturen. Die Eingabe `\MyVerb|!'|` erzeugt die Ausgabe `!'`. Natürlich gibt es Lösungsmöglichkeiten, aber keine ist hundertprozentig zufriedenstellend. Eine Möglichkeit wäre für den Schriftstil `\ttfamily` alle Ligaturen zu deaktivieren. Detaillierter möchte ich allerdings an dieser Stelle nicht auf diese Problematik eingehen.

Abschließend ein kleiner Hack in Verbindung mit *verbatim*-Argumenten. Sollte es einmal zu der Situation kommen, dass Anweisungen mit einem *verbatim*-Argument innerhalb einer anderen Anweisung genutzt werden, so kann folgender Hack helfen:¹

```
1 \footnote{\scantokens{\MyVerb|\LaTeX|\noexpand}}
```

Zusammenfassung

Der vorliegende Artikel gab eine erste Einführung in die Nutzung des Paketes *xparse*. Der Schwerpunkt sollte hierbei auf die Möglichkeiten des Paketes selbst liegen, ohne auf die Möglichkeiten der zugrundeliegenden Sprache `exp13` einzugehen. Es ist gezeigt worden, dass das Paket mit $\text{\LaTeX 2}_{\epsilon}$ sowie mit $\text{\TeX}$ -Makros arbeiten kann. Die Vielzahl der Argumentenbezeichner ist begrüßenswert und erlaubt dem Anwender Makrodefinitionen mit komplexerer Struktur.

Die im vorangegangenen Abschnitt getätigte Danksagung sei hier noch einmal wiederholt. Ohne die hervorragende Zusammenarbeit mit Frank Mittelbach, Joseph Wright, Herbert Voß und Rolf Niepraschk wäre dieser Artikel nicht möglich gewesen.

Literatur

- [1] Michel Goossens, Frank Mittelbach und Alexander Samarin: *The $\text{\LaTeX}$ Companion*; Addison-Wesley, Reading, Massachusetts; 1. Aufl.; 1994.
- [2] The $\text{\LaTeX 3}$ Project: *The xparse package – Document command parser*; <http://ctan.org/pkg/xparse>; 2012.

¹ In dieser Fußnote findet der Hack Anwendung: `\LaTeX` .

biblatex mit tufte-latex verwenden

Dominik Waßenhoven

Durch eine Frage auf $\text{\TeX}$.sx¹ bin ich auf das Problem gestoßen, biblatex mit den Tufte-Klassen zu verwenden. Glücklicherweise gibt es dafür bereits eine Lösung, die hier kurz vorgestellt werden soll.

Die Dokumentklassen `tufte-handout` und `tufte-book` bilden den Stil der Bücher von Edward Tufte² nach. Das hervorstechendste Merkmal seiner Bücher ist, dass es eine breite Marginalspalte gibt, in der Anmerkungen und kleinere Grafiken Platz finden. In der Version 3.5.0, die derzeit auf CTAN verfügbar ist,³ werden bibliographische Angaben mit Hilfe des Pakets `natbib[1]` gesetzt, wobei einige Umdefinierungen des `\cite`-Befehls dazu führen, dass die Verweise mit vollständigen bibliographischen Angaben ausgegeben werden. Will man die Möglichkeiten von `biblatex[2]` nutzen, braucht man mindestens Version 3.5.2 von `tufte-latex`, die derzeit nur im SVN-Repository zu finden ist.⁴ Mit dieser Version wird die Option `nobib` eingeführt, so dass die angesprochenen Umdefinierungen des `\cite`-Befehls nicht ausgeführt werden. Stattdessen kann man nun `biblatex` mit all seinen Optionen nutzen, wie im folgenden Beispiel demonstriert wird:

Listing 1: Beispiel mit dem Standard-biblatex-Stil »numeric«

```
\documentclass[nobib]{tufte-handout}
\usepackage[ngerman]{babel}
\usepackage{hyphenat}
\usepackage[backend=biber]{biblatex}
\addbibresource{biblatex-examples.bib}

\begin{document}
Literaturverweise im numerischen Stil.\cite{springer}
\printbibliography
\end{document}
```

Das Paket `hyphenat[4]` wird lediglich geladen, um einen Konflikt mit `biblatex` zu umgehen, denn der Befehl `\nohyphenation` wird sowohl in `tufte-common.def` wie auch von `biblatex` definiert. Lädt man `hyphenat`, von dem der Befehl in `tufte-common.def` übernommen wurde, gibt es keine Fehlermeldung.⁵

¹ »Can I use biblatex with Tufte classes?« auf <http://tex.stackexchange.com/questions/45934/can-i-use-biblatex-with-tufte-classes>.

² <http://www.edwardtufte.com>.

³ CTAN:macros/latex/contrib/tufte-latex/.

⁴ <http://code.google.com/p/tufte-latex/source>; wichtig ist vor allem die Datei `tufte-common.def`.

⁵ Ob sich dadurch irgendwelche Beeinträchtigungen ergeben, kann ich nicht beurteilen.

Statt des numerischen Stils kann man natürlich auch jeden anderen biblatex-Stil nutzen. Man sollte nur darauf achten, dass die Literaturverweise dann auch in eine `\footnote` gepackt werden, was im folgenden Beispiel durch die biblatex-Option `autocite=footnote` gewährleistet ist (statt `\autocite` kann man natürlich auch `\footcite` verwenden):

Listing 2: Beispiel mit dem biblatex-Stil »verbose«

```
\documentclass[nobib]{tufte-handout}
\usepackage[ngerman]{babel}
\usepackage{hyphenat}
\usepackage[
  style=verbose,
  autocite=footnote,
  backend=biber
]{biblatex}
\addbibresource{biblatex-examples.bib}

\begin{document}
Bei einem ersten Literaturverweis erfolgt ein Vollzitat,\autocite{springer}
weitere Verweise werden
mit einem Kurzzitat ausgegeben.\autocite{springer}
\printbibliography
\end{document}
```

Es kann auch sinnvoll sein, bei Folgezitationen auf die Anmerkung mit der ersten, vollständigen Literaturangabe zu verweisen. Damit kann man ganz auf ein Literaturverzeichnis verzichten. Das lässt sich beispielsweise mit dem Stil »footnote-dw« aus dem Paket `biblatex-dw`[3] erreichen:

Listing 3: Beispiel mit dem Stil »footnote-dw« aus dem Paket `biblatex-dw`

```
\documentclass[nobib]{tufte-handout}
\usepackage[ngerman]{babel}
\usepackage{hyphenat}
\usepackage[
  style=footnote-dw,
  backend=biber
]{biblatex}
\addbibresource{biblatex-examples.bib}

\begin{document}
Beim ersten Literaturverweis erfolgt ein Vollzitat,\cite{springer} direkt
nachfolgende Verweise werden mit
"ebenda" wiedergegeben.\cite{springer} Weitere Zitate werden mit einem
Hinweis auf das erste Zitat
ausgegeben.\cite{brandt,springer}
\end{document}
```

Literaturverweise im numerischen Stil.[1] <i>Literatur</i> [1] Otto Springer. "Mediaeval Pilgrim Routes from Scandinavia to Rome". In: <i>Mediaeval Studies</i> 12 (1950), S. 92–122.
--

Abbildung 1: Ausgabe von Listing 1

Bei einem ersten Literaturverweis erfolgt ein Vollzitat,¹ weitere Verweise werden mit einem Kurzzitat ausgegeben.² <i>Literatur</i> Springer, Otto. "Mediaeval Pilgrim Routes from Scandinavia to Rome". In: <i>Mediaeval Studies</i> 12 (1950), S. 92–122.	¹ Otto Springer. "Mediaeval Pilgrim Routes from Scandinavia to Rome". In: <i>Mediaeval Studies</i> 12 (1950), S. 92–122. ² Springer, "Mediaeval Pilgrim Routes".
--	--

Abbildung 2: Ausgabe von Listing 2

Beim ersten Literaturverweis erfolgt ein Vollzitat,¹ direkt nachfolgende Verweise werden mit »ebenda« wiedergegeben.² Weitere Zitate werden mit einem Hinweis auf das erste Zitat ausgegeben.³	¹ Otto Springer: Mediaeval Pilgrim Routes from Scandinavia to Rome, in: <i>Mediaeval Studies</i> 12 (1950), S. 92–122. ² Ebd. ³ Ahasver von Brandt/Erich Hoffmann: Die nordischen Länder von der Mitte des 11. Jahrhunderts bis 1448, in: Ferdinand Seibt (Hrsg.): <i>Europa im Hoch- und Spätmittelalter</i> (Handbuch der europäischen Geschichte 2), Stuttgart 1987, S. 884–917; Springer: <i>Mediaeval Pilgrim Routes</i> (wie Anm. 1).
--	---

Abbildung 3: Ausgabe von Listing 3

Literatur

- [1] Patrick W. Daly: *Natural Sciences Citations and References (Author–Year and Numerical Schemes)*; Sept. 2010; CTAN:macros/latex/contrib/natbib.
- [2] Philipp Lehman: *The biblatex package. Programmable bibliographies and citations*; Nov. 2011; CTAN:macros/latex/contrib/biblatex.
- [3] Dominik Waßenhoven: *biblatex-dw*; Nov. 2011; CTAN:macros/latex/contrib/biblatex-contrib/biblatex-dw.
- [4] Peter Wilson: *The hyphenat package*; Apr. 2004; CTAN:macros/latex/contrib/hyphenat.

Aufrechte Klammern in kursivem Text

Dominik Waßenhoven

In *The Elements of Typographic Style* empfiehlt Robert Bringhurst, Klammern immer aufrecht zu setzen, auch wenn Sie innerhalb von kursivem Text stehen.¹ Er gibt dazu folgende Erklärung:

Runde und eckige Klammern sind keine Buchstaben, und deshalb ist es kaum sinnvoll, sie als »normal« oder »kursiv« zu bezeichnen. Es gibt aufrechte und geneigte Klammern; die Klammern kursiver Schriftschnitte sind fast immer geneigt, aber aufrechte Klammern sind generell zu bevorzugen. Das heißt, sie müssen aus dem aufrechten Schriftschnitt stammen, und sie brauchen gegebenenfalls zusätzlichen Abstand, wenn man sie mit kursiver Schrift einsetzt. [...] Die geneigten eckigen Klammern, die normalerweise in kursiven Schriftschnitten vorkommen, sind – wenn überhaupt – sogar noch weniger brauchbar als geneigte runde Klammern. Falls es – Gott bewahre – ein Buch oder einen Film mit dem Titel *The view from My [sic] Bed* gäbe, könnten geneigte Klammern sinnvoll sein, um auszudrücken, dass die eckigen Klammern und ihr Inhalt tatsächlich zum Titel gehören. Ansonsten sollten aufrechte eckige Klammern verwendet werden, egal ob der Text aufrecht oder kursiv ist: »The View from My [sic] Bed« und »*the view from my [sic] bed.*«²

Will man diese Vorgaben mit $\text{\LaTeX}$ umsetzen, muss man die Klammern immer aus dem `\emph`-Befehl heraushalten.

```
1 The View from My [\emph{sic}] Bed;\n
2 \emph{the view from my \emph{[{}sic\emph{}}] bed.}\n
3 \emph{The view from My [sic] Bed.}
```

Das ist vor allem dann umständlich, wenn in kursiviertem Text aufrechte Klammern gesetzt werden sollen, wie die zweite Zeile des obigen Code-Beispiels zeigt. Diese Zeile ließe sich alternativ folgendermaßen schreiben:

¹ [1, 85]: »Use upright (i.e., »roman«) rather than sloped parentheses, brackets and braces, even if the context is italic.«

² [1, 85]: »Parentheses and brackets are not letters, and it makes little sense to speak of them as roman or italic. There are vertical parentheses and sloped ones, and the parentheses on italic fonts are almost always sloped, but vertical parentheses are generally to be preferred. That means they must come from the roman font, and may need extra spacing when used with italic letterforms. [...] The sloped square brackets usually found on italic fonts are, if anything, even less useful than sloped parentheses. If, perish the thought, there were a book or film entitled *The view from My [sic] Bed*, sloped brackets might be useful as a way of indicating that the brackets and their contents are actually part of the title. Otherwise, vertical brackets should be used, no matter whether the text is roman or italic: ›The View from My [sic] Bed‹ and ›*the view from my [sic] bed.*«

```
1 \emph{the view from my} [\emph{sic}] \emph{bed.}
```

Diese Lösung ist aber ebenfalls nicht sehr schön. Außerdem besteht das Problem, dass die genannten Lösungen nicht mehr ohne Weiteres funktionieren, sobald man sie innerhalb von eigenen Befehlen einsetzt – etwa einem Befehl zum Setzen von lateinischen Zitaten in kursiver Schrift. Deshalb drängt sich die Frage auf, wie sich das gewünschte Verhalten möglichst automatisch mit $\TeX$ -Mitteln bewerkstelligen lässt. Diese Frage wurde auf $\TeX$.sx gestellt³ und erhielt mehrere Antworten, von denen zwei im Folgenden kurz dargestellt werden.

Eine recht einfache Methode ist, einen eigenen Befehl zu definieren, der kursiven Text innerhalb aufrechter Klammern setzt. Dazu benötigt man zunächst einen Befehl für die aufrechten Klammern:

```
1 \newcommand\notemph[1]{\textup{#1}}
```

Für die kursiven Texte in aufrechten oder runden Klammern definiert man dann jeweils einen Befehl:

```
1 \newcommand\ep[1]{\notemph{({#1\notemph{}})}% ep=emph in parentheses
2 \newcommand\eb[1]{\notemph{[#1\notemph{[]]}% eb=emph in square brackets
```

Der Code könnte dann folgendermaßen aussehen (wie dieser Code ausgegeben wird, ist am Ende dieses Beitrags zu sehen):

```
1 The View from My \emph{\eb{sic}} Bed;\n
2 \emph{the view from my \eb{sic} bed.}\n
3 \emph{The view from My [sic] Bed.}
```

Das Ganze funktioniert dann auch in verschachtelten Umgebungen (und damit auch in eigenen Befehlen, die `\emph` verwenden). Der Vorteil dieser Methode ist, dass man überall volle Kontrolle über die Klammern hat und sie auch leicht wieder undefinieren kann. Nachteilig ist dagegen, dass der Code recht unübersichtlich wird (so in der ersten Zeile des obigen Beispiels) und man jedes Mal daran denken muss, die Befehle einzusetzen. Es ist also nur eine halbautomatische Lösung.

Mit Hilfe von $\TeX$ 3 lässt sich jedoch auch eine vollautomatische Lösung erreichen, indem man die Definition der Klammern anpasst, so dass sie immer aufrecht gesetzt werden – unabhängig vom Kontext. Dazu werden die beiden Pakete `expl3` [2] und `xparse` [3] benötigt. Bei der Umdefinierung der Klammern in der Präambel des Dokuments muss zunächst die Syntax von `expl3` angeschaltet werden:

```
1 \ExplSyntaxOn
```

Anschließend wird die alte Definition von `\emph` kopiert:

³ Siehe <http://tex.stackexchange.com/q/13048/1376>.

```
2 \cs_new_eq:Nc \emph_old:n { \emph~ }
```

Nun wird festgelegt, wie die Klammern gesetzt werden sollen, nämlich aufrecht (`\textup`).

```
3 \cs_new_eq:NN \emph_braces:n \textup
```

Jetzt werden alle vorkommenden Klammern durch den neuen Klammern-Stil ersetzt:

```
4 \cs_new:Npn \emph_new:n #1 {
5   \tl_set:Nn \l_emph_tl {#1}
6   \tl_replace_all:Nnn \l_emph_tl {{}}{\emph_braces:n{}}
7   \tl_replace_all:Nnn \l_emph_tl {}){\emph_braces:n{}}
8   \tl_replace_all:Nnn \l_emph_tl {[[]]{\emph_braces:n{[[]]}}
9   \tl_replace_all:Nnn \l_emph_tl {}]{\emph_braces:n{[]]}}
10  \exp_args:NV \emph_old:n \l_emph_tl
11 }
```

Bei der jetzt noch nötigen Umdefinierung von `\emph` wird eine Sternvariante eingeführt, die es erlaubt, die alte Definition von `\emph` zu verwenden:

```
12 \RenewDocumentCommand \emph {sm} {
13   \IfBooleanTF {#1} {\emph_old:n {#2}} {\emph_new:n {#2}}
14 }
```

Zum Schluss muss lediglich noch die `expl`-Syntax wieder abgeschaltet werden:

```
15 \ExplSyntaxOff
```

Mit dieser Umdefinierung sind im Text jetzt keine gesonderten Befehle mehr zu verwenden, `\emph` wird ganz gewöhnlich eingesetzt. Wenn man doch einmal geneigte Klammern braucht, hat man dafür `\emph*` zur Verfügung:

The View from My <i>[sic]</i> Bed;	1 The View from My \emph{[sic]} Bed;\
the view from my <i>[sic]</i> bed.	2 \emph{the view from my [sic] bed.}\
The view from My <i>[sic]</i> Bed.	3 \emph{The view from My [sic] Bed.}

Literatur

- [1] Robert Bringhurst: *The Elements of Typographical Style*; Hartley & Marks; Point Roberts and Vancouver; 5. Aufl.; 2008.
- [2] The $\text{\LaTeX}$ Project: *The expl3 package and $\text{\LaTeX}$ 3 programming*; Febr. 2012; CTAN:macros/latex/contrib/l3kernel.
- [3] The $\text{\LaTeX}$ Project: *The xparse package. Document command parser*; Febr. 2012; CTAN:macros/latex/contrib/l3packages.

Der T_EXnische Fortschritt und seine Tücken

Jürgen Hanneder

Nur zur Klarstellung: Wenn ich im folgenden über die Entwicklung von T_EX et al. klage, dann ist dies die Klage eines ausschließlichen T_EX-Benutzers, der immer noch nicht weiß oder wissen will, wie Formatvorlagen in Textverarbeitungsprogrammen funktionieren. Meine DANTE-Mitgliedsnummer ist sehr niedrig und meine Leidensfähigkeit, wenn es um Satz geht, ist ausgeprägt: Als ich als Student mit einem XT-Computer (Betriebssystem DOS) meine Abschlussarbeit formatierte, dauerte der T_EX-Lauf meiner kritischen Edition eines Sanskrit-Textes mit edmac heute unvorstellbare drei Stunden (für edmac benötigt man aber drei Läufe!). Dennoch habe ich seither nichts anderes als dieses geniale Programm benutzt, denn wenn man neun Stunden gewartet hatte, bekam man zumindest ein richtig gesetztes Buch, während die Konkurrenz sich mit Schreibmaschinenästhetik zufriedengeben musste. Manches hat sich zum Glück seither geändert. Derselbe T_EX-Lauf – ich habe es kürzlich mit der uralten Datei versucht – dauert nun unter einer Minute.

Auch anderes ist seither sehr viel komfortabler geworden. Anfangs mussten, wenn man indische Texte in Transkription setzen wollte, für diakritische Zeichen T_EX-Kommandos eingeben werden. Den indischen Gott Kṛṣṇa schrieb man als T_EXie `K\{r}\{s}\{n}`a. Man musste sich schließlich behelfen, indem man – für jedes System neu – einen Bildschirmfont, eine Tastaturbelegung und die T_EX-Eingabekodierung schrieb, ein Problem, welches jede T_EX-Implementation individuell löste. Doch die Vorteile lagen auf der Hand. Während die Redmond-hörigen Kollegen immer genervt nach einem Font mit allen Diakritika suchten, konnte man sich als T_EX-Anwender entspannt zurücklehnen. Wollte man seine Spuren verwischen und nicht ein von weitem als solches erkennbares T_EX-Dokument setzen, sondern ein normales Buch, dann musste man Computer Modern und die L^AT_EX-Standardstile meiden.¹ Wenn man sich noch einen hochwertigen kommerziellen Font kaufte, der schon in T_EX integriert war,² konnte man eigentlich sehr zufrieden sein.³

Das größte Problem im Bereich der indischen Sprachen war das der Originalschriften. Hier – es geht im Folgenden aber nur um Sanskrit und Devanāgarī – hatte ein Indologe⁴ schon längst eine gute Lösung entwickelt, die T_EX sehr lange einen Wettbewerbsvorteil verschaffte. Zwar brauchte man einen Präprozessor, den man für

¹ Siehe die Beispiele auf S. 57 (*Abhinavagupta's Philosophy of Revelation*, Groningen 1998.) und S. 58 (*Mokṣopāya. Kritische Edition*. Wiesbaden 2011).

² Siehe <http://home.vr-web.de/was/fonts>.

³ Siehe die Probeseiten aus indologischen Publikationen mit Aldus (S. 57) und Sabon (S. 58).

⁴ Siehe CTAN:/language/devanagari/velthuis.

jedes Betriebssystem kompilieren musste, und für die Eingabe wurde ein neues Transkriptionsschema geschaffen (Kṛṣṇa musste nun K.r.s.na geschrieben werden), doch das System wurde so beliebt, dass sich die vermutlich aus der Not erfundene Transkription sogar in der Indologie etablierte! Denn man konnte das, was die Konkurrenz heute noch nicht kann: Kritische Editionen in indischen Schriften drucken. Aber eigentlich nur in einem einzigem Schriftschnitt, denn man musste die Grenzen von $\TeX$ auf den verschiedensten Ebenen umgehen und alle Elemente auf die Struktur des Fonts zuschneiden. Da indische Schriften, wenn der Font nicht viele Hundert Zeichen haben soll, aus Einzelelementen zusammengesetzt werden, mussten alle diese Strichelemente möglichst effektiv in Fonts konventioneller Größe untergebracht und dann vom Programm zusammengesetzt werden. Man benötigte also für jeden Font einen nur darauf abgestimmten Präprozessor, da niemand versucht hatte, den Font mit $\TeX$ -Makros zusammenzusetzen, was sicher die für $\TeX$ -Fanatiker bessere Lösung gewesen wäre. Es gibt eine solche, wenn mich nicht alles täuscht, für die südindische Schrift Tamil.

Doch schien dies alles nicht mehr nötig, denn es winkte wieder der Fortschritt in Form neuer $\TeX$ -Varianten (Omega, Aleph etc.), die für Multilingualisten alles konnten, und für die man folgerichtig immer wieder Anpassungen produzierte. Der Haken war, dass Forscher dies in ihrer Freizeit neben ihrer eigentlichen Forschung bewerkstelligen mussten. Doch diese Arbeit wurde immer wieder obsolet, da sich die $\TeX$ -Gemeinde nie auf *ein* besseres $\TeX$ konzentrieren wollte, und ich muss gestehen, dass ich mit steigendem Arbeitsaufwand die Geduld verlor, als Omega von Aleph, dann Lua $\TeX$ abgelöst wurde; sobald man mühevoll eine Anpassung an die aktuellen $\TeX$ -Programme erstellt hatte, gab es ein neues und die Entwicklung des alten stockte. Der gemeinsame Nenner schien die gegenüber dem Urprogramm immer weiter reduzierte Halbwertszeit zu sein, eine Entwicklung, die eigentlich den großen Wettbewerbsvorteil von $\TeX$, seine spektakuläre Stabilität, zunichte machte. Natürlich: Wenn man englischen Text druckte, war alles kein Problem, aber wenn man Sanskrit drucken wollte, fing alles immer wieder von Neuem an.

Ich muss für das Folgende die Perspektive daher noch weiter einengen: Es geht hier darum, $\TeX$ zu verwenden, um in einem begrenzten Zeitfenster ein Dokument zur Druckreife zu bringen. Vertröstungen auf die nächste Programmversion, welche die bisherigen Probleme elegant löst, werden plötzlich irrelevant. Was zählt, ist das, was zu einem fest definierten Zeitpunkt vollständig funktioniert – und zwar auch solche Probleme, die einem beim Herumprobieren nicht gleich auffallen und die wahrscheinlich dem Entwickler, der seine Makros nie zur Buchproduktion verwendet hatte, ebenfalls nicht bewusst waren. Steht man vor dieser Aufgabe, so merkt man, dass die $\TeX$ -Gemeinde mehr entwickler- als anwenderorientiert ist. Es gibt viele Lösungen für Probleme, jedoch vergleichsweise wenige gelungene Anwendungsbeispiele, zumindest für die Geisteswissenschaften. Es gibt ja nicht

einmal eine Dokumentklasse, die standardmäßig ein Seitenformat nahe am Goldenen Schnitt produziert, was natürlich auch daran liegt, dass die meisten $\TeX$ -Anwender vermutlich aus den Naturwissenschaften kommen und dort die DIN-Formate beliebt sind. Es ist kein Problem DIN B5 zu drucken, aber ein schönes Format für die Wissenschaftliche Buchgesellschaft muss man sich selbst schreiben. Natürlich ist so etwas nicht schwierig, doch der erzieherische Effekt wird vertan. Wir wollen ja auch vermeiden, dass sich falsche Kapitälchen aufgrund der Sehgewohnheiten als Standard einprägen. Dass auch viele Wissenschaftsverlage nur noch DIN B5 zu kennen scheinen, erschwert die Sache erheblich, aber ein Gang in die Belletristikabteilung einer größeren Buchhandlung kann hier heilsam sein. Dort gibt es keine DIN-Formate, doch das nur am Rande.

Die entscheidende Entwicklung war die Erweiterung der Zeichensätze. Die vermutlich englischsprachigen Fontdesigner wussten zwar, dass man in Europa kleine Punkte und Striche über bestimmte Buchstaben setzt, die heute gelegentlich der OCR-Software zum Opfer fallen, haben den Sinn dieser alten europäischen Sitte aber lange nicht eingesehen. Jetzt geht man davon aus, dass moderne Unicode-Zeichensätze alle Diakritika beherrschen, doch ist dies eben nur fast richtig, denn der Großteil der Unicode-Fonts kann zwar einen Teil der für die Umschrift von Sanskrit verwandten Zeichen (unproblematisch ist $\bar{a}$, $\bar{u}$ und $\bar{i}$), aber nicht alle (viele scheitern an $\ddot{t}$, $\ddot{n}$ oder $\ddot{r}$). Stellen Sie sich also die Situation vor, dass die meisten Zeichensätze zwar $\ddot{a}$ und $\ddot{u}$ können, aber kein $\ddot{o}$. Die theoretisch sehr hohe Zahl an Möglichkeiten schwindet. Auch Hinweise auf Fonts, die entweder nicht weiterentwickelt und nicht mehr bereitgestellt werden (Code2000) oder solche, deren Copyright eine kommerzielle Verwendung ausschließen (TitusCyberbit), erweisen sich für den Praktiker als Sackgassen, so interessant sie zum Experimentieren im einzelnen auch sein mögen.

Haben Sie sich bei dem Gedanken ertappt: Aber es gibt doch jetzt $\XeTeX$, das löst doch solche Probleme schon längst! Die natürliche Reaktion auf Probleme mit dem alten Programm ist bekanntlich, eine neue Version zu installieren. Wir sind und bleiben wohl technikbegeisterte Kinder des Fortschritts. Und natürlich habe auch ich immer die letzte Linux-Version auf allen meinen Rechnern und teste seit längerem $\XeTeX$, drucke Devanāgarī damit, habe mich aber erst jetzt, sozusagen unter Sachzwang, dazu durchringen müssen, ein Buch damit zu setzen. Der Grund liegt in der zwiespältigen Natur der Neuerungen dieses Programmes. Einerseits ist es phantastisch, dass nun alle Systemschriften mit $\TeX$ kooperieren (oder umgekehrt). Aber die Zahl der Schriften, die sich für den wissenschaftlichen Buchdruck in der Indologie wirklich eignen, vergrößert sich weitaus weniger spektakulär.

Die Ausgangslage war, dass in $\TeX$ seit jeher die Eingabe $\backslash d\{d\}$ das richtige Zeichen ($\ddot{d}$) ergibt und man den $\TeX$ -Code unter einer besser lesbaren Oberfläche verbergen

kann. Man hat einen Bildschirmfont mit den richtigen Zeichen, die Eingabekodierung setzt dies in đ um, ein Tastaturmakro erleichtert die Eingabe. Und der zum Druck verwandte Font steht unter der Kontrolle von $\TeX$, so dass die große Zahl der mit $\TeX$ kooperierenden Zeichensätzen keinerlei Probleme macht. Dieses Verfahren wurde in DOS-Zeiten noch mit $\text{em}\TeX$ realisiert, dann mit Omega etc. Eingabeseitig hat sich nicht viel geändert, man benötigt immer noch einen geeigneten Bildschirmzeichensatz – viele, die behaupten, unicode-fähig zu sein, können đ nicht darstellen – und eine Eingabemethode. Doch mit $\text{X}_{\text{f}}\TeX$ – und dies ist der entscheidende Unterschied – hat man die Kontrolle über die Diakritika den Fontdesignern übertragen, die selten Indologen sind. Was bedeutet dies für den Praktiker? Ganz einfach: Die theoretische Vielfalt an Fonts, die $\text{X}_{\text{f}}\TeX$ so reizvoll macht, schmilzt im Nu auf die bisherige Größe zusammen. Die meisten Fonts, die mit $\text{X}_{\text{f}}\TeX$ zusammenarbeiten, stellen die Umschrift des Sanskrit nicht korrekt dar. Bei manchen fehlen Zeichen (đ ist wieder beliebt), bei anderen stimmt das Kerning nicht usw. Sicher kleine, vielleicht auch leicht behebbare Fehler (wenn man Fontdesigner ist), aber solche, die den Font für eine Buchproduktion ausschließen. Da die hochwertigen kommerziellen Fonts, die man sich für die alten $\TeX$ -Versionen gekauft hatte (Type 1, kein OpenType), nicht mehr funktionieren, fängt man wieder neu an, kauft Fonts und hofft, dass sie gut programmiert sind.

Ich werde für die genannte Publikation dennoch $\text{X}_{\text{f}}\TeX$ verwenden, denn erstmals mit $\TeX$ eine Auswahl aus mehreren Devanāgarī-Schriften zu haben, ist ein durchaus erhebendes Gefühl. Aber auch hier gilt: Nur ein Bruchteil funktioniert wirklich, und ich bin – um es positiv auszudrücken – immer wieder fasziniert, wie wenig Anspruch und Realität zusammenpassen. Manche haben typographische Mängel, andere sind vermutlich unzureichend programmiert und verschlucken ganze Ligaturen. Kein einziger der anlässlich der World Sanskrit Conference in New Delhi im Januar 2012 von indischen Entwicklern verteilten Zeichensätze tut, was er soll (zumindest mit $\text{X}_{\text{f}}\TeX$)! Kommerzielle Devanāgarī-Fonts für Sanskrit sind bisher ebenfalls Fehlanzeige, wobei URW kurz vor der Fertigstellung eines solchen steht. Das Beispiel auf S. 59 zeigt die acht Zeichensätze, die nach einer langwierigen Sichtung einer großen Zahl übrig geblieben sind.

Versucht man nun in einer Publikation, europäische und indische Schriften in ein und demselben Dokument beziehungsweise in derselben Zeile zu setzen, dann müssen »Ost« und »West« auch noch bei Schriftdicke und Stil zusammenpassen. Dadurch fällt einer der beliebtesten Sanskrit-Fonts (Sanskrit2003) wieder heraus, da er eigentlich die fette Variante einer nicht existierenden Normalschrift ist. Es bleiben also weitaus weniger Möglichkeiten, als man denkt, aber am Ende – und daher will ich mit einer guten Nachricht enden – lässt sich mit $\text{X}_{\text{f}}\TeX$ das Buch dann doch so produzieren, wie ich es mir vorstelle, denn der Devanāgarī-Font Siddhanta

passt glücklicherweise ganz gut zum T_EX-Font Libertine. Aber ich experimentiere noch.

1. INTRODUCTION

1.1 READING THE *Yogavāsiṣṭha*

Everyone who has attempted to read the printed *Yogavāsiṣṭha* (YV)¹ in two volumes will remember the beginning with its set of frame stories which reminds one of the introduction to the or similar narrative works. Since few people today and presumably also few in former times used to read the whole text, but were content with an impression that especially in a partly repetitive work as the YV could be gained very soon, it is quite probable that the perspective in which the work is put at its inception would prefigure one's expectation of what the rest contains and in what manner one would have to interpret it.

In order to explain this point I shall briefly comment upon the dialogue between Sūtikṣṇa and Agastī that starts the work:²

*bhagavan dharmatattvajña sarvaśāstraviniścita /
saṁśayo 'sti mahān ekas tvam etaṁ kṛpayā vada (5)
mokṣasya kāraṇaṁ karma jñānaṁ vā mokṣasādhanaṁ /
ubhayaṁ vā viniścitya ekaṁ kathaya kāraṇam (6)*

O Lord, you know the truth of *dharma*, you have determined [the sense of] all Śāstras. There is one large doubt, please speak about this out of compassion: Is action the cause of liberation, or is knowledge the means to liberation? Or both? Tell me with certainty the one cause.

Agastī answers:

*ubhābhyām eva pakṣābhyāṁ yathā khe pakṣiṇāṁ gatiḥ /
tathaiva jñānakarmabhyāṁ jāyate paramaṁ padam (7)
kevalāt karmaṇo jñānān na hi mokṣo 'bhijāyate /
kiṁtūbhābhyāṁ bhaven mokṣaḥ sādhanaṁ tūbhayaṁ viduḥ (8)
asminn arthe purāvṛttam itihāsaṁ vadāmi te /*

¹ See the bibliography for the editions of this version.

² The text is quoted from the printed YV, Sarga 1.5ff.

PRATHAMAS SARGAḤ

vasiṣṭhaḥ

vāgbhābhīr brahma vid brahma bhāti svapna ivātmani yad idaṃ tat svaśabdārthair yo yad vetti sa vettu tat	I.1
nyāyenānena sarvasmin sarge brahmāmbare sati kim idaṃ kasya vakṣīti codyacañcur nirākṛtaḥ	I.2
ahaṃ tāvad yathājñānaṃ yathāvastu yathākramam yathāsvabhāvaṃ vacmīdaṃ tat sarvaṃ śrūyatāṃ budhāḥ	I.3
svapnavat paśyati jagac cinnabho dehavinmayam svapnasamsāradṛṣṭāntā ihaivāntas samanvitāḥ	I.4
mumukṣuvyavahāroktimayāt prakaraṇād anu athotpattiprakaraṇaṃ mayedaṃ parikathyate	I.5
bandho 'yaṃ dṛśyasadbhāvo dṛśyābhāve na bandhanam na sambhavati dṛśyaṃ tu yathedaṃ tac chṛṇu kramāt	I.6
utpadyate yo jagati sa eva kila vardhate sa eva mokṣam āpnoti svargaṃ vā narakaṃ ca vā	I.7
atas te svavabodhārthaṃ tat tāvat kathayāmy aham utpattiṃ saṃśṛtāv eti pūrvam eva hi yo yathā	I.8
idaṃ prakaraṇārthaṃ tvaṃ saṅkṣepāc chṛṇu rāghava tataḥ prakathayiṣyāmi vistaraṃ te yathepsitam	I.9
yad idaṃ dṛśyate sarvaṃ jagat sthāvarajaṅgamam tat suṣupta iva svapnaḥ kalpānte pravinaśyati	I.10
tatas stimitagambhīraṃ na tejo na tamas tatam anākhyam anabhivyaktaṃ yat kiñcid avaśiṣyate	I.11

1a Ś₁₄ 58^r 1ab brahma bhāti svapna ivātmani cett.] *bra*[...]*ivātmani* Ś₁₄ 1d vettu cett.] *vetti* Ś₇
2ab sarvasmin sarge cett.] [...] *rge*[...] Ś₁₄ 2b brahmāmbare sati cett.] *brahmākāmbare*[...] *ati*
Ś₁₄^{pc.s.m.} 2c vakṣīti Ś₁^{pc} Ś₃ Ś₇ Ś₉ N₁₂ N₂₀] *vakṣeti* Ś₁^{pc}, l.n. Ś₁₄ 2d codya cett.] [...] Ś₁₄ 2d cañcur
cett.] *cuñcur* Ś₉, [...] *uñcur* Ś₁₄ 3c yathāsvabhāvaṃ cett.] *yathā*[...] Ś₁₄ 3c vacmīdaṃ
Ś₁ Ś₃ Ś₇ Ś₉ N₁₂] *vaśmīdaṃ* N₂₀, [...] *idaṃ* Ś₁₄ 4a paśyati Ś₁ Ś₇ Ś₉ Ś₁₄^{pc} N₁₂ N₂₀] *paśyata* Ś₃, *yūṣanta*
Ś₁₄^{pc.s.m.} 4c svapnasamsāradṛṣṭāntā cett.] *sva*[...] *ṣṭāntā* Ś₁₄ 4d ihaivāntaḥ Ś₁ N₁₂ N₂₀ Ś₉ Ś₁₄] *ihaivāntas* Ś₃ Ś₇
4d Ś₉ 98^r 5b anu Ś₇ Ś₁₄] *param* Ś₁ Ś₃ Ś₉ N₁₂ N₂₀ 5c athot cett.] [...] *t* Ś₁₄ 6d
tac chṛṇu Ś₇ Ś₉ Ś₁₄ N₁₂] *śṛṇu tat* Ś₁ Ś₃ N₂₀ 7d N₂₀ 2^r 8a te svava Ś₁ Ś₃ Ś₇ Ś₉ Ś₁₄^{pc} N₂₀] *teṣu ava* N₁₂,
te svāva Ś₁₄^{pc.s.m.} 8c pūrvam cett.] *p*[...] *rvam* Ś₁₄ 9a idaṃ Ś₇ Ś₉ Ś₁₄ N₁₂] *imaṃ* Ś₁ Ś₃ N₂₀ 10d
pravinaśyati cett.] *pravinaśyati* Ś₁^{pc}

Devanāgarī unter X_YT_EX

Die folgende Übersicht zeigt die mir bekannten Devanāgarī-Zeichensätze, die bisher in der Praxis mit X_YT_EX zusammenarbeiten.

फेरेङ्गितीर्थसङ्काशे काशति पुरी सुन्दरी मारदुर्गाभिधाना सा लानानूपे व्यवस्थिता	<i>Siddhanta</i>
फेरेङ्गितीर्थसङ्काशे काशति पुरी सुन्दरी मारदुर्गाभिधाना सा लानानूपे व्यवस्थिता	<i>Samyak</i>
फेरेङ्गितीर्थसङ्काशे काशति पुरी सुन्दरी मारदुर्गाभिधाना सा लानानूपे व्यवस्थिता	<i>Nakula</i>
फेरेङ्गितीर्थसङ्काशे काशति पुरी सुन्दरी मारदुर्गाभिधाना सा लानानूपे व्यवस्थिता	<i>Sahadeva</i>
फेरेङ्गितीर्थसङ्काशे काशति पुरी सुन्दरी मारदुर्गाभिधाना सा लानानूपे व्यवस्थिता	<i>Sanskrit2003</i>
फेरेङ्गितीर्थसङ्काशे काशति पुरी सुन्दरी मारदुर्गाभिधाना सा लानानूपे व्यवस्थिता	<i>Jaipur</i>
फेरेङ्गितीर्थसङ्काशे काशति पुरी सुन्दरी मारदुर्गाभिधाना सा लानानूपे व्यवस्थिता	<i>JanaSanskrit</i>
फेरेङ्गितीर्थसङ्काशे काशति पुरी सुन्दरी मारदुर्गाभिधाना सा लानानूपे व्यवस्थिता	<i>shiDeva</i>

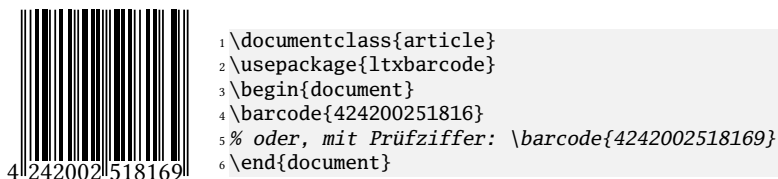
Strichcodes erzeugen mit Lua $\TeX$

Patrick Gundlach

$\TeX$ bietet hervorragende Typographie, mit Lua kann man einfach programmieren. Wenn man diese beiden kombiniert, kann man komplexe Satzanforderungen einfach erfüllen. In diesem Beitrag werde ich einen Weg beschreiben, GTIN-13 Strichcodes (auch unter dem Namen EAN-13 bekannt) mit Lua $\TeX$ zu erzeugen. Der Artikel zeigt neben grundlegender Lua-Programmierung auch, wie Lua aus $\TeX$ heraus aufgerufen wird und wie Lua und $\TeX$ zusammenarbeiten.

Einführung

Es gibt mehrere Wege, Barcodes von $\TeX$ aus zu erzeugen. Ein Weg ist das PSTricks-Paket `pst-barcode`, einige Pakete benötigen spezielle Fonts, und ein Paket, das ich gefunden habe, basiert auf vertikalen Linien (`\rule`), aber der Quellcode ist nicht besonders zugänglich für Anfänger und daher schwer zu erweitern. Die Lua-basierte Lösung, die ich hier vorstelle, soll einfach auch für nicht $\TeX$ -Programmierer zu verstehen sein, was natürlich immer subjektiv ist. Damit der Artikel kurz bleibt, werden nur EAN-13-Strichcodes gezeichnet. Optional wird noch die Prüfziffer berechnet, wenn nur 12 Ziffern angegeben werden. Folgendes ist eine Beispielausgabe des Programms:



Die Paketdatei `ltxbarcode.sty` ist ebenfalls kurz. Diese kann im selben Verzeichnis liegen wie die $\TeX$ -Datei.

```

1 \ProvidesPackage{ltxbarcode}
2
3 \directlua{require("ltxbarcode")}
4
5 \newcommand\barcode[1]{%
6   \directlua{ ltxbarcode.generate_barcode("#1") }}

```

Das Paket lädt die Datei `ltxbarcode.lua`, die weiter unten gezeigt und erklärt wird. `require` fügt die Dateierweiterung `.lua` von selbst an. Anschließend wird das Kommando `\barcode` definiert, dessen einzige Aufgabe es ist, in den Lua-Modus zu springen (`\directlua`) und die Lua-Funktion `generate_barcode()` aufzurufen.

Das einzige Argument der Funktion ist der Inhalt von #1, also das, was dem $\TeX$ -Befehl `\barcode` übergeben wurde. Der Namensraum `ltxbarcode` wird durch das `require()` automatisch hinzugefügt.

Hier gibt es eine kleine Falle, denn normalerweise wird der Code folgendermaßen geschrieben:

```
1 \newcommand\barcode[1]{%
2   \directlua{
3     ltxbarcode.generate_barcode("\luatexluaescapestring{#1}")}}
```

Durch das `\luatexluaescapestring` werden »gefährliche« Zeichen in harmlose umgewandelt. Das sind beispielsweise doppelte Anführungszeichen, denn diese würden das Ende der Zeichenkette signalisieren und Lua kann nichts mit den restlichen Argumenten anfangen. Da wir aber nur normale Ziffern übergeben, sparen wir uns die Sicherheit, nichtsdestotrotz sind solche Vorsichtsmaßnahmen immer gut. Der lange Name `\luatexluaescapestring` kommt daher, dass alle Lua $\TeX$ -Befehle in $\TeX$ ein `luatex` vorangestellt bekommen, damit es keine Namensüberschneidungen mit schon existierenden Paketen gibt. Es gibt nur eine Ausnahme: Der Befehl `\directlua` bleibt, wie er im Handbuch beschrieben ist. Bevor wir das Lua-Modul genauer anschauen, möchte ich zeigen, wie der Lua-Modus und Lua $\TeX$ kommunizieren.

Von Lua nach $\TeX$

Die Übergabe von Argumenten von $\TeX$ nach Lua ist leicht, wie oben im Funktionsaufruf `generate_barcode()` gezeigt. Die andere Richtung, also von Lua nach $\TeX$, ist interessanter, weil man wissen muss, wann der Lua-Code ausgeführt wird. Lua-Befehle werden nämlich genau dann ausgeführt, wenn $\TeX$ die schließende Klammer von `\directlua` findet. Dieser Aufruf wird dann durch die Ausgabe von dem Codeblock »ersetzt«:

```
1 \directlua{
2   tex.sprint("\hbox{%)")
3   tex.sprint("Hallo Welt%")
4   tex.sprint("}")
5 }
```

und der $\TeX$ -Code

```
1 \hbox{%
2 Hallo Welt%
3 }
```

sind mehr oder weniger äquivalent. Das bedeutet, dass Zeichenketten (strings) in kleinere Einheiten geteilt werden können. Diese werden am Ende vom `\directlua`-Aufruf wieder zusammengefügt. Was aber nicht möglich ist:

```

1 \directlua{
2   tex.sprint("\setbox0\hbox{Hallo Welt!}")
3   % geht nicht, weil Box 0 noch nicht bekannt ist:
4   tex.sprint(
5     string.format("Die Breite von Box 0 ist nun %d", tex.box[0].width))
6 }

```

Innerhalb von `\directlua` kann man $\TeX$ - und Lua-Berechnungen, wie diese, nicht vermischen, da der $\TeX$ -Wert bis zum Ende des `\directlua`-Aufrufs nicht bekannt ist. Daher kann man nicht auf die Größen einer Box zurückgreifen, bevor $\TeX$ sie setzt. Wenn wir die beiden Ansätze für die Barcodes miteinander vergleichen, wird das Problem eine Rolle spielen.

Erste Lösung: `tex.sprint()`

Im ersten Ansatz wird der Strichcode berechnet (das ist eine einfache Funktion) und mithilfe von `\hbox-`, `\vrule-` und `\kern-`Befehlen mit `tex.sprint()` erzeugt. Bevor wir diese Funktion anschauen, fangen wir mit dem Kopf der Lua-Datei an (`ltxbarcode.lua`). Die meisten Lua-Module starten mit einem Aufruf von `module()`. Alle, bis auf die Haupt-Funktionen, werden als *lokal* deklariert, das heißt, dass diese nur innerhalb der Datei sichtbar sind.

```

1 module(...,package.seeall)
2
3 local add_checksum_if_necessary, mkpattern, split_number, calculate_unit,
  pattern_to_wd_dp

```

Nun kommt das Herzstück des Moduls. Wir benutzen die Methode, wie oben beschrieben, um eine Sequenz von $\TeX$ -Befehlen zu erzeugen, die direkt nach `\directlua` ausgeführt werden. Die Idee ist, den Strichcode mit den Befehlen `\vrule` und `\kern` zu zeichnen und die Zahlen in einer separaten Box unterhalb des Codes einzufügen.

```

1 function generate_barcode(str)
2
3   -- Wenn nur 12 Ziffern übergeben werden, fügen wir die 13. hinzu
4   str = add_checksum_if_necessary(str)
5
6   -- Die kleinste Lücke/Strich ist ein siebtel der Breite einer Ziffer und damit
7   -- abhängig von der Schriftart
8   local u = calculate_unit()
9
10  -- Wir fangen mit der hbox für die Striche an:
11  tex.sprint([\newbox\barcodebox\setbox\barcodebox\hbox{%%}])
12
13  -- Das Muster (pattern) ist eine Zeichenkette von Ziffern, die die Breite eines
14  -- Strichs bzw einer Lücke repräsentieren. 0 ist eine spezielle Markierung, die
15  -- einen längeren Strich der Breite 1 bezeichnet. Die Breiten werden mit 1/7

```

```

16 -- der Ziffernbreite multipliziert, da die Summe der Striche/Lücken für eine
17 -- Ziffer insgesamt 7 Einh. breit sind. Ein Beispiel-Muster fängt an mit
18 -- 80103211112312132113231132111010132...
19 local pattern = mkpattern(str)
20
21 -- Für jeden Eintrag im Muster erzeugen wir eine Lücke oder einen Balken mit
22 -- der Breite, die im Eintrag angegeben ist. Eine Tiefe > 0 wird für die Balken
23 -- am Rand und in der Mitte des Strichcodes benutzt. Es gibt keine technische
24 -- Notwendigkeit dafür, aber es ist üblich, den Strichcode so darzustellen.
25 local wd,dp -- Breite und Tiefe eines Strichs
26 for i=1,string.len(pattern) do
27   wd,dp = pattern_to_wd_dp(pattern,i)
28   -- gerade Einträge sind die Länge eines Strichs(vrule), ungerade Lücken(kern).
29   if i % 2 == 0 then
30     tex.sprint(
31       string.format([[\\vrule width %dsp height 2cm depth %s]],wd * u,dp))
32   else
33     tex.sprint(string.format([[\\kern %dsp]],wd * u))
34   end
35 end
36 -- Wir haben die hbox mit den Strichen und erzeugen die hbox mit den Ziffern.
37 tex.sprint(
38   [[\\vbox{\\hsize\\wd\\barcodebox\\box\\barcodebox\\kern -1.7mm\\hbox{\\%}}
39 ])
40
41 -- Die Ziffern unterhalb werden in drei Teile getrennt. Eine vor dem ersten
42 -- Strich, die erste Hälfte links und die zweite rechts von dem mittleren.
43 local first,second,third = split_number(str)
44 tex.sprint(
45   string.format(
46     [[\\%s\\kern %dsp %s\\kern %dsp%s]],first, 5 * u, second, 4 * u, third
47 ))
48 end

```

Wir müssen die Striche in eine eigene Box packen, damit die umschließende vertikale Box genau diese Breite hat. Da wir, wie oben beschrieben, nicht innerhalb des `\directlua`-Aufrufs auf die Breite zurückgreifen können, müssen wir erst die horizontale Box mit den Strichen erzeugen und anschließend ihre Breite in `\hsize` benutzen. Dafür erzeugen wir entsprechende $\TeX$ -Befehle.

Die Hauptfunktion benutzt mehrere Hilfsfunktionen. Eine davon berechnet die Breite des schmalsten Balkens und der schmalsten Lücke. Das ist genau $1/7$ der Breite einer Ziffer. Wir benutzen die Fontbibliothek von Lua $\TeX$, um auf die aktuelle Schriftart zugreifen zu können. Die Nummer des Zeichens 0 ist 48, aber es ist kodierungsabhängig. In der Praxis funktioniert es aber in allen Fällen, aber zumindest sollte man das im Hinterkopf behalten, falls es doch mal Schwierigkeiten geben sollte.

```

1 function calculate_unit()
2   -- Die relativen Breiten der Striche und Lücken in einer Ziffer summieren sich
   zu 7
3   local currentfont = font.fonts[font.current()]
4   local digit_zero = currentfont.characters[48]
5   return digit_zero.width / 7
6 end

```

Die nächste Funktion berechnet die Breite und die Tiefe einer vertikalen Linie. Die Höhe ist fest (natürlich könnte man die auch veränderbar machen, aber dann hätte der Leser dieses Artikels nichts mehr zu tun).

```

1 function pattern_to_wd_dp(pattern,pos)
2   local wd,dp
3   wd = tonumber(string.sub(pattern,pos,pos))
4   if wd == 0 then
5     dp = "2mm"
6     wd = 1
7   else
8     dp = "0mm"
9   end
10  return wd,dp
11 end

```

Die Berechnung der Prüfziffer ist recht einfach. Es werden alle Ziffern addiert, jede zweite Ziffer wird mit 3 multipliziert (von hinten gezählt) und die Prüfziffer ist der Betrag, den man zum nächsten Vielfachen von 10 addieren muss.

```

1 function add_checksum_if_necessary( str )
2   if string.len(str) == 13 then
3     return str
4   end
5
6   local sum = 0
7   local len = string.len(str)
8   for i=len,1,-1 do
9     if (len - i ) % 2 == 0 then
10      sum = sum + tonumber(string.sub(str,i,i)) * 3
11    else
12      sum = sum + tonumber(string.sub(str,i,i))
13    end
14  end
15  local checksum = (10 - sum % 10) % 10
16  return str .. tostring(checksum)
17 end

```

Die folgende Mustererstellung ist ein wichtiger Teil des Algorithmus. Der Strichcode wird in kleine Teile unterteilt, in dem zwei Striche und zwei Lücken eine Ziffer repräsentieren. Die Breiten dieser Striche und Lücken variieren zwischen *eins* und *vier*, multipliziert mit einer passenden Einheit. Die Breite einer Ziffer summiert

sich zu sieben dieser Einheiten, und diese werden durch ein einfaches Muster, wie 2221 für die Ziffer 1, dargestellt. Die erste Ziffer in einem Strichcode wird nicht durch Striche und Lücken dargestellt, sondern durch die Kodierung der nächsten sechs Ziffern. Wenn beispielsweise die erste Ziffer in einem Strichcode eine 1 ist, dann werden die dritte, fünfte und sechste Ziffer umgekehrt. Dafür wird das Feld `mirror_t` im Code unten benutzt. Im Beispiel oben wäre das umgekehrte Muster 1222. Wir fügen etwas Leerraum links vom Barcode hinzu, um Platz für die erste Ziffer zu reservieren. Ebenfalls wird die zwingend erforderliche Zeichenfolge 111 links und rechts vom Strichcode eingefügt. In unserem Beispiel ist das 010, da die beiden Striche etwas länger dargestellt werden sollen. Dieses Muster wird später im Code erkannt.

```

1 function mkpattern(str)
2   -- Die Ziffern 0-9 werden durch diese Striche/Lücken dargestellt. 3211
3   -- bedeutet: Lücke von 3 Einheiten, Strich von 2 Einheiten Breite und eine
4   -- Lücke und ein Strich von jeweils einer Einheit.
5   local digits_t = {"3211", "2221", "2122", "1411", "1132", "1231", "1114", "1312", "1213", "3112"}
6
7   -- Die erste Ziffer wird durch die Darstellung der nächsten sechs kodiert.
8   -- Eine 1 bedeutet, dass die erzeugten Striche/Lücken umzukehren sind.
9   local mirror_t = {"-----", "--1-11", "--11-1", "--111-", "-1--11", "-11--1", "-1-1-1", "-1-11-", "-11-1-"}
10
11  -- Die Zeichenkette aus Ziffern wird in ein Feld konvertiert.
12  local number = {}
13  for i=1,string.len(str) do
14    number[i] = tonumber(string.sub(str,i,i))
15  end
16
17  -- Die erste Ziffer bestimmt, wie die nächsten sechs dargestellt werden.
18  local prefix = table.remove(number,1)
19  local mirror_str = mirror_t[prefix + 1]
20
21  -- Die Variable pattern speichert das erzeugte Muster. Es wird mit der Lücke
22  -- für e die erst Ziffer angefangen, anschließend kommt die linke Begrenzung
23  -- 111 bzw. hier als 010 für die Kodierung der längeren Striche angegeben.
24  local pattern = "010"
25  local digits_str
26
27  for i=1,#number do
28    digits_str = digits_t[number[i] + 1]
29    if string.sub(mirror_str,i,i) == "1" then
30      digits_str = string.reverse(digits_str)
31    end
32    pattern = pattern .. digits_str
33    -- Die mittleren beiden Striche
34    if i==6 then pattern = pattern .. "10101" end
35  end

```

```

36 -- Das Muster 111 rechts anfügen.
37 return pattern .. "010"
38 end

```

Die letzte Funktion teilt den Zifferncode in drei Teile, so dass wir diese unterhalb des Strichcodes mit Lücken darstellen können.

```

1 function split_number(str)
2   return string.match(str, "(%d)(%d%d%d%d%d%d)(%d%d%d%d%d%d)")
3 end

```

Das Ergebnis dieses Codes ist ein $\TeX$ -String

```

1 \newbox\barcodebox\setbox\barcodebox\hbox{%
2 \kern 374492sp
3 \vrule width 46811sp height 2cm depth 2mm
4 \kern 46811sp
5 ...
6 \vrule width 46811sp height 2cm depth 2mm
7 }\vbox{\hsize\wd\barcodebox\box\barcodebox\kern -1.7mm\hbox{%
8 8\kern 234057sp 008940\kern 187246sp027004}}

```

Das ist, was $\TeX$ nach der schließenden Klammer von `\directlua` sieht. Diese Lösung funktioniert für unser kleines Beispiel sehr gut, aber für komplexere Programme ist das nicht mehr das Mittel der Wahl. Die Übergabe der Zeichenketten ist anfällig für fehlerhaftes Maskieren der Sonderzeichen wie % und &. Außerdem muss man sich immer merken, welche Teile der Zeichenketten wie `\hbox{` schon geschrieben wurden. Zum Glück haben wir mit Lua $\TeX$, dem neuen Universalwerkzeug in der $\TeX$ -Welt, eine andere Möglichkeit, um das Problem zu lösen.

Zweite Lösung: Direktes Setzen mit Low-Level-Nodes

Der andere Ansatz, das Problem zu lösen, sieht aus, als ob man einen Vorschlaghammer benutzt, um eine Nuss zu knacken. Sobald man aber mit dieser Methode vertraut ist, eignet sie sich für viele Aufgaben, wenn man sowieso Lua als Programmiersprache benutzt. Die Idee besteht darin, die grundlegenden Datenstrukturen zu erstellen, die $\TeX$ selbst intern erzeugt, bevor die Inhalte in PDF ausgegeben werden. Die kleinsten Einheiten dieser Datenstruktur sind sogenannte *nodes*, also Knoten. Ein Knoten kann ein Zeichen, eine dehnbare Länge, ein Strich oder ein anderes Element sein, das wir von $\TeX$ kennen. Die Ziffer 0 wird beispielsweise von der folgenden Tabelle beschrieben:

Eintrag	id	char	font	lang
Wert	37	48	15	0

Es gibt weitere (optionale) Einträge in dieser Tabelle. Die *prev* und *next* Einträge hingegen sind notwendig, um komplexere Datenstrukturen zu erzeugen. Der Knoten,

der durch die oben dargestellte Tabelle repräsentiert wird, kann mit folgenden Lua-Anweisungen erzeugt werden.

```
1 n = node.new("glyph") -- interne id: 37
2 n.char = 48
3 n.font = 15
4 n.lang = 0
```

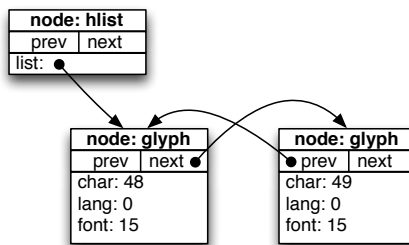
Um eine horizontale box (hbox) mit dieser Ziffer zu erzeugen, reicht ein Aufruf von `node.hpack()`.

```
1 hbox = node.hpack(n)
```

Das ist dasselbe wie `\hbox{0}`, mit dem Unterschied, dass die Box nur im Speicher von $\TeX$ und nicht im PDF ist. Es wird etwas komplexer, wenn mehr als ein Element in einer Box gespeichert werden soll. Dafür muss man mehrere Nodes erzeugen und in eine Liste *verketteten*. Jeder Knoten hat einen *prev*- und einen *next*-Zeiger in der Tabelle, der auf den Vorgängerknoten bzw. den Nachfolgerknoten zeigen muss. Werden beispielsweise die Ziffer 0 und 1 in einer hbox gespeichert, sieht das folgendermaßen aus:

```
1 digit_0 = node.new("glyph")
2 digit_1 = node.new("glyph")
3 -- nicht gezeigt: Tabellen wie oben füllen
4 digit_0.next = digit_1
5 digit_1.prev = digit_0
6 hbox = node.hpack(digit_0)
```

Das Ergebnis ist eine Datenstruktur, die wie folgt visualisiert werden kann:



Der Eintrag *list* der *hlist* (hbox) zeigt auf die *node list*, die mit der Ziffer 0 anfängt. Die Idee für unseren zweiten Ansatz ist nun, eine solche Liste zu erzeugen, die die vertikalen Linien und Lücken beinhaltet, und eine zweite Liste für die Ziffern. Wir erzeugen daher zusätzliche Hilfsfunktionen sowie eine neue Hauptfunktion.

```
1 local add_to_nodelist, mkrule, mkkern, mkglyph
2
3 function generate_barcode_lua(str)
4   str = add_checksum_if_necessary(str)
```

```

5
6 local u = calculate_unit()
7 local nodelist
8
9 -- Die geraden Einträge sind die Linien, die ungeraden die Lücken
10 local pattern = mkpattern(str)
11 local wd,dp
12 for i=1,string.len(pattern) do
13     wd,dp = pattern_to_wd_dp(pattern,i)
14     if i % 2 == 0 then
15         nodelist = add_to_nodelist(
16             nodelist,mkrule(wd * u,tex.sp("2cm"),tex.sp(dp)))
17     else
18         nodelist = add_to_nodelist(nodelist,mkkern(wd * u))
19     end
20 end
21 -- barcode_top wird die vbox aus der ersten Lösung
22 local barcode_top = node.hpack(nodelist)
23 barcode_top = add_to_nodelist(barcode_top,mkkern(tex.sp("-1.7mm")))
24
25 -- Die folgende Liste enthält die dargestellten Ziffern
26 nodelist = nil
27 for i,v in ipairs({split_number(str)}) do
28     for j=1,string.len(v) do
29         nodelist = add_to_nodelist(nodelist,mkglyph(string.sub(v,j,j)))
30     end
31     if i == 1 then
32         nodelist = add_to_nodelist(nodelist,mkkern(5 * u))
33     elseif i == 2 then
34         nodelist = add_to_nodelist(nodelist,mkkern(4 * u))
35     end
36 end
37 local barcode_bottom = node.hpack(nodelist)
38 -- barcode_top hat nun drei Einträge: die hbox mit den Linien und Kerns, das
39 Kern
40 -- mit der Größe -1.7mm und die hbox mit den Ziffern unterhalb der Striche
41 barcode_top = add_to_nodelist(barcode_top,barcode_bottom)
42 local bc = node.vpack(barcode_top)
43
44 -- node.write() schreibt die vbox in die PDF Datei
45 node.write(bc)
46 end

```

Die Struktur ist dieselbe wie in der ersten Lösung. Der Hauptunterschied liegt in den Hilfsfunktionen `mkrule()`, `mkkern()` und `mkglyph()`, um Linien, Unterschneidungen und Zeichen zu erzeugen und dem Aufruf von `add_to_nodelist()`. Die erzeugte Liste wird mithilfe des Funktionsaufrufs `node.write()` in die PDF-Datei geschrieben.

```

1 function add_to_nodelist( head,entry )
2     if head then

```

```

3  -- Den Eintrag 'entry' an die Nodeliste anhängen und die prev/next Zeiger
    anpassen.
4  local tail = node.tail(head)
5  tail.next = entry
6  entry.prev = tail
7  else
8  -- Keine Nodeliste vorhanden, also wird einfach der Eintrag zur Nodeliste
9  head = entry
10 end
11 return head
12 end

```

Falls die Nodeliste existiert, wird der neue Eintrag hinter den letzten Knoten eingefügt. Man könnte auch zum Ende der Nodeliste gelangen, in dem man die *next*-Zeiger so lange weiter geht, bis man auf den »leeren« Wert *nil* stößt, mit der Funktion `node.tail()` ist das aber einfacher. Anschließend passen wir die *next*- und *prev*-Zeiger der beiden letzten Knoten an und geben den Listenkopf zurück.

```

1 function mkrule( wd,ht,dp )
2   local r = node.new("rule")
3   r.width = wd
4   r.height = ht
5   r.depth = dp
6   return r
7 end
8
9 function mkkern( wd )
10  local k = node.new("kern")
11  k.kern = wd
12  return k
13 end
14
15 function mkglyph( char )
16  local g = node.new("glyph")
17  g.char = string.byte(char)
18  g.font = font.current()
19  g.lang = tex.language
20  return g
21 end

```

Diese drei Funktionen benötigen keine große Erklärung. Sie erzeugen einen Knoten vom gewünschten Typ. Es mag auf den ersten Blick überraschen, dass der Glyph-Knoten einen Eintrag für die Sprache und die Schriftart benötigt, weil man normalerweise auf der $\TeX$ -Ebene diese Werte nicht angibt. Da aber diese Nodeliste die Datenstruktur ist, die $\TeX$ kurz vor der Ausgabe in PDF benutzt, müssen die Werte jedem Zeichen zugeordnet sein.

Fazit

Es gibt zwei Wege, um Satzinformationen von Lua an T_EX zu übergeben: Der erste Weg ist mithilfe von `tex.sprint()`-Anweisungen und der zweite ist mithilfe einer Nodeliste. Sobald man in der Lua-Welt unterwegs ist, fühlt es sich falsch an, Informationen an T_EX mit `tex.sprint()`-Anweisungen zu übergeben. Man muss noch immer Kategorie-Codes (catcodes) beachten, ebenso Gruppen (`{` und `}`) und Schutzzeichen wie den Schrägstrich. In der prozeduralen Welt von Lua liegt der richtige Weg darin, die zu setzenden Elemente auf niedriger Ebene aus Nodes zusammenzubauen. Mit Hilfsfunktionen wird diese Aufgabe sehr einfach. Sobald man anfängt, in diesen Datenstrukturen zu denken, kann man seinen Fokus darauf richten, Dinge auf der Seite anzuordnen. Dann kommt einem auch die Eingabesprache von T_EX nicht in die Quere. Heutzutage mutet die Programmiersprache von T_EX schon sehr anachronistisch an, wenn man mal mit Sprachen wie JavaScript in Berührung gekommen ist. T_EXs Algorithmen sind aber immer noch unübertroffen. Wenn man also die Mächtigkeit einer einfachen Programmiersprache mit den Textsatzqualitäten von T_EX kombiniert, wird eine neue Generation von Anwendungen für schönen Textsatz möglich.

Literatur

- [1] Patrick Gundlach: *T_EX without T_EX*; http://wiki.luatex.org/index.php/TeX_without_TeX; 2011.
- [2] Patrick Gundlach: *Quellcode zu diesem Artikel*; <https://gist.github.com/2035612>; 2012.
- [3] Taco Hoekwater: *LuaT_EX reference manual*; [luatexref-t.pdf](#); 2011.
- [4] Herbert Voß: *The current state of the PSTricks project*; *TUGBoat*; 31(1); 2010.

Bücher schreiben mit der $\text{\LaTeX}$ -Dokumentenklasse memoir

Christine Römer

Memoir ist eine moderne und leistungsfähige Dokumentenklasse, die bisher im deutschsprachigen Raum wenig bekannt ist, was sicher auch dem mächtigen Konkurrenten KOMA-Script und der nicht optimalen Dokumentation geschuldet ist. Eventuell ist auch die irreführende Benennung der Dokumentenklasse eine Ursache. Es ist beim Vorhandensein von zwei modernen $\text{\LaTeX}$ -Dokumentenklassen für das Setzen von komplexen Publikationen (scrbook und memoir) nicht einsehbar, dass in einigen Einführungsbüchern für das Schreiben von Büchern immer noch die Vorgängerklasse book empfohlen wird.

Einführung

Die Textklasse memoir wurde 2001 von ihrem Schöpfer Peter R. Wilson öffentlich gemacht. Zur Zeit ist die Version 3.6j aktuell, die von Lars Madsen gepflegt wird. Eingangs wurde sie als Weiterentwicklung der Standardklassen book und report »mit viel Drum und Dran« ([7]) für das Schreiben von umfangreichen Texten entwickelt; sie kann heute jedoch auch mit Gewinn für kleine Textsorten verwendet werden. Sie hat gegenüber book nicht nur viel mehr vordefinierte Funktionen zur Textgestaltung, sondern auch die Funktionen zahlreicher Zusatzpakete in die Klasse integriert und vereinfacht so die Textgestaltung deutlich. Beispielsweise sind für Tabellen Codes aus den Paketen dcolumn, delarray, tabularx und booktabs eingeschlossen und in die drei angebotenen Tabellenumgebungen integriert:

```
1 \begin{tabular}[<pos>]{<format>} Reihen \end{tabular}
2 \begin{tabular*}[<width>][<pos>]{<format>} Reihen \end{tabular*}
3 \begin{tabularx}[<width>][<pos>]{<format>} Reihen \end{tabularx}
```

Natürlich können weitere Pakete hinzu geladen werden. Wie die Dokumentation zeigt, hat es in der Entwicklung Probleme mit der Integration von hyperref gegeben. Es muss jetzt nicht mehr zusätzlich mit memhfixc geladen werden. Nachfolgend ist auch zu sehen, wie memoir zum Schreiben von deutschsprachigen Texten in Unicode-fähigen Editoren, die für alle Betriebssysteme zur Verfügung stehen, eingebunden wird (pdf $\text{\LaTeX}$ -Variante):

```
1 \documentclass[12pt,ngerman,a4paper]{memoir}
2 \usepackage[T1]{fontenc}
```

```

3 \usepackage[utf8]{inputenc}
4 \usepackage{babel}
5 \usepackage[colorlinks,linktoc=all]{hyperref}
6 \pagestyle{empty}
7
8 \begin{document}
9 \url{http://mirror.ctan.org/macros/latex/contrib/memoir/}
10 \end{document}

```

Im README (2011/03/06) wird zum Leistungsumfang ausgeführt:

Memoir is a flexible class for typesetting poetry, fiction, non-fiction and mathematical works as books, reports, articles or manuscripts. Documents can use 9pt, 10pt, 11pt, 12pt, 14pt, or 17pt as the normal fontsize and, if you have scalable fonts, 20pt, 25pt, 30pt, 36pt, 48pt, or 60pt sizes, or even larger. Many methods are provided to let you create your particular design. The class incorporates over 30 of the more popular packages.

Die sehr ausführliche Dokumentation (memman.pdf, fast 600 Seiten) beschreibt auch, wie man die vordeklarierten Textgestaltungselemente konfigurieren kann. Sie nimmt jedoch manchmal Züge einer allgemeinen Einführung in $\text{\LaTeX}$ und die Typografie an. Dabei hebt sie auch nicht immer ab, was nun speziell von den Standardklassen stammt und was von memoir hinzugefügt bzw. modifiziert wurde. Im 3. Kapitel »Text and fonts« werden beispielsweise die für $\text{\LaTeX}$ zur Verfügung stehenden Schriften charakterisiert und in Kapitel 11 »Rows and columns« werden allgemeine Prinzipien einer Tabellengestaltung abgehandelt. Diese Weitschweifigkeit macht es schwierig, die zahlreichen in der Dokumentenklasse enthaltenen Innovationen wahrzunehmen und sich in die Benutzung einzuarbeiten.

Vergleich mit KOMA-Script

Es kann hier kein ausführlicher Vergleich mit KOMA-Script erfolgen, dieser soll nur in Bezug auf einige Aspekte für das Setzen eines Buches (scrbook) erfolgen. Zum Vergleich mit der älteren Klasse book siehe [1].

Das komplexe Paket KOMA-Script von Markus Kohm erschien 1994 als Nachfolger von Script-2 (Frank Neukam 1993), 1994 kam die Briefklasse von Axel Kielhorn hinzu. Es wurde ständig weiterentwickelt und existiert heute (08.03.2012) als KOMA-Script 3.10a. Es handelt sich dabei um eine Stilfamilie, die den Regeln der europäischen Typografie folgt und im Layout zurückgeht auf Arbeiten von Jan Tschichold, der eher der traditionellen Typografie folgte. memoir scheint nach meinem Eindruck kein spezielles typografisches Konzept zugrunde zu liegen. An allen möglichen Stellen kann über programmierte Befehle das Layout verändert werden. Dies ist zum einen positiv, da es kreativen Buchgestaltungen Raum gibt, kann aber auch negativ

betrachtet werden, da es zu unschönen, nicht zusammenpassenden typografischen Stilverbindungen führen kann.

scrbook wurde als verbesserte Alternative zu der $\text{\LaTeX}$ -Standardklasse book geschaffen. Neben einem veränderten vordefinierten Seitenstil zur Ausgangsklasse wurden weitere Funktionen hinzugefügt, beispielsweise die Aufnahme des Literaturverzeichnisses und anderer Verzeichnisse ins Inhaltsverzeichnis oder die einfache Satzspiegelberechnung mit typearea.

Während in memoir die Standardeinstellung für den englischen Sprachraum festgelegt ist (letterpaper wird automatisch eingestellt), erfolgt die Standardeinstellung von KOMA-Script für den deutschen Sprachraum (DIN-Papierformat a4paper). Beide Pakete sind jedoch nicht darauf festgelegt. scrbook ist ohne Zusatzfestlegungen u. a. auf die Schriftgröße 11 pt, das Format a4paper, nummerierte, serifenlose Überschriften ohne den Zusatz Kapitel sowie die traditionellen Absatzeinrückungen, eine eigene, unnummerierte Titelseite, doppelseitigen Druck mit jeweiligem Kapitelanfang auf einer eigenen ungeraden Seite, Seitenzahlen in der Fußzeile außen und mit Kopfzeilen ausgelegt. Dass KOMA-Script etliche Möglichkeiten enthält, mit denen das Aussehen verändert werden kann, soll nur nochmals erwähnt werden. Dies ist jedoch nicht im gleichen Umfang wie bei memoir der Fall.

Mit scrbook und im Zusammenspiel mit den Paketen makeidx, biblatex und csquotes baut sich ein deutschsprachiges, wissenschaftliches Fachbuch im Default-Fall aus folgenden Komponenten auf, wobei auf nicht nötige Zusatzdefinitionen verzichtet werden soll:

```

1 \documentclass[a4paper,ngerman]{scrbook}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage{babel}
5 \usepackage{blindtext}% nur zu Demonstrationszwecken
6 \usepackage{makeidx}
7 \makeindex
8 \usepackage[backend=biber,style=authoryear]{biblatex}
9 \usepackage[babel,german=guillemets]{csquotes}
10 \addbibresource{Bibliografie.bib}
11
12 \begin{document}
13 \begin{titlepage}
14 Titel, etc.
15 \end{titlepage}
16 \tableofcontents
17 \chapter{Dies und Das}
18 \blindtext
19 \section{Jenes}
20 \blindtext
21
22 \printbibliography

```

```

23 \printindex
24 \end{document}

```

Standardwerte von memoir

Ein deutschsprachiges, wissenschaftliches Fachbuch kann mit memoir analog zu der Dokumentenklasse book strukturiert werden.

```

1 \documentclass[ngerman,a4paper]{memoir}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage{babel}
5 \usepackage{blindtext}
6 \usepackage{makeidx}
7 \makeindex
8 \usepackage[backend=biber,style=authoryear]{biblatex}
9 \usepackage[babel,german=guillemets]{csquotes}
10 \addbibresource{Bibliografie.bib}
11
12 \aliaspagestyle{chapter}{empty}% leere Kopfzeile, keine Seitennummer
13
14 \begin{document}
15 \title{Einbandtitel}
16 \begin{titlingpage}
17   \title{Buchtitel}
18   \author{Alpha\thanks{ABC},
19         Beta\thanks{XYZ} and
20         Omega\thanksmark{2}}
21   \maketitle
22   \begin{abstract} Text\end{abstract}
23 \end{titlingpage}
24
25 \frontmatter
26 \tableofcontents
27
28 \chapter{Vorbemerkungen}
29 \blindtext
30
31 \mainmatter
32 \chapter{Dies}
33 \section{Das}
34 \blindtext
35 \section{Jenes}
36 \blindtext
37
38 \appendix
39 \addappheadtotoc
40 \chapter{Und noch etwas}
41
42 \backmatter
43 \printbibliography

```

```

44 \printindex
45 \end{document}

```

Ohne das Einfügen abstrakter Gliederungselemente (`\frontmatter`=Vorspann, `\mainmatter`=Hauptteil und `\backmatter`=Nachspann), die auch für `book` und für `scrbook` zur Verfügung stehen, wird der Text durchgängig unten in der Mitte mit Seitenzahlen versehen. Nach dem Ergänzen der Strukturierungsbefehle sind die Seitenzahlen oben rechts bzw. links außen. `\frontmatter` führt zu römischer Seitennummerierung, die Sternchenversion verhindert dagegen die Änderung der Nummerierungsart. Nach `\mainmatter` wird mit arabischen Seitenzahlen begonnen. Wenn der `Appendix`-befehl (`appendix`) eingefügt wird, wechselt die Kapitelnummerierung im Anhang in eine alphabetische Form (in unserem Beispiel erscheint dann »Anhang A«). Nach dem Strukturierungsbefehl `\backmatter` findet bei `memoir` im Standardfall keine Kapitelnummerierung mehr statt. Mit der Anweisung `\addappheadtotoc` werden die Anhänge in das Inhaltsverzeichnis eingetragen und mit `\appendixtocname` kann der Name angepasst werden:

```

1 \usepackage{babel}
2 \addto\captionsngerman{\renewcommand*{\appendixtocname}{Anhang}}

```

Ohne spezielle Angabe nimmt `memoir` die Schriftgröße 10 pt, Computer Modern im OT1-Encoding. Der Text wird wie bei `scrbook` auch mit Absatzeinzug und ohne Absatzabstand gesetzt, was ich schade finde. Auch wenn u. a. in [4] zahlreiche einleuchtende Argumente für den Absatzeinzug vorgetragen werden, stimme ich jenen zu, die meinen, dass Einzüge nicht unbedingt sein müssen [5, S. 176]. Ihre Ordnungs- und Gliederungsfunktion kann auch durch kleine Zeilenabstände übernommen werden. Mit `scrbook` kann dies einfach mit bereitgestellten Optionen bei den fakultativen Angaben zur Dokumentenklasse eingestellt werden (`parskip`, ...). Wie bei der Klasse `book` kann man auch in `memoir` mit $\TeX$ -Befehlen für vertikale und horizontale Abstände und zusätzlichen Anpassungen der Umgebungsparameter durch das `parskip`-Paket den Absatzeinzug auf 0 und Abstand zwischen aufeinanderfolgenden Absätzen auf 1 ex einrichten. Die Plus- und Minuswerte legen die erlaubte Ausdehnung und Stauchung fest.

```

1 \setlength{\parindent}{0pt}
2 \setlength{\parskip}{1ex plus 0.2ex minus 0.1ex}

```

Da dies jedoch laut der Dokumentation eine problematische Lösung ist, die zu unangenehmen Seiteneffekten führen kann, weil $\TeX$ den Abstand `\parskip` auch an anderen Orten nutzt, stellt `memoir` die Befehle `\abnormalparskip` (setzt Zeilenzwischenraum) und `\nonzeroparskip` bereit, um ähnlich dem Paket `parskip` diese Effekte zu reduzieren.

```

1 \setlength{\parindent}{0pt}
2 \nonzeroparskip

```

Mit `\abnormalparskip{<length>}` kann ein spezifischer Wert gewählt werden. Mit der Deklaration `\traditionalparskip` kann auf die ursprünglichen Werte zurückgesetzt werden.

Die Kapitelüberschriften fallen ohne Abänderung relativ mächtig aus und sind in Serifenschrift und mit dem Zusatz *Kapitel* abgefasst (vgl. Abbildung 1).

Abbildung 1: Standardkapitelüberschriften



Sowohl für *scrbook* als auch bei *memoir* ist der Seitenstil *headings* (mit standardisiertem Seitenkopf, der in der Regel die aktuelle Überschrift enthält) der Normalfall. Während bei *memoir* im Defaultfall sich die Seitenzahl auch im Seitenkopf befindet, steht sie bei KOMA-Script in der Fusszeile.

Besondere Bestandteile von *memoir*

Von Madsen werden in [6, S. 23] für die Dokumentenklasse *memoir* zusammenfassend folgende Spezifika hervorgehoben:

- Flexible Wege für das Einrichten der Seite, beispielsweise beim Setzen von Marginalien oder dem Bestimmen der Größe von Textblöcken und deren Platzierung auf der Seite oder beim Kombinieren von Beidem.
- Ein umfassendes System zum Einrichten von Fuß- und Kopfzeilen (ist nach Madsen viel enger integriert, als das jemals ein Add-on-Paket könnte).
- Schnittstellen zur Konfigurierung des Aussehens und der Handhabung der verschiedenen Listen (wie das Inhaltsverzeichnis und Verwandte) sowie Schnittstellen, um neue zu kreieren.
- Methoden zur Anpassung der Erscheinungsform von Kapitelüberschriften und den abhängigen Abschnittstiteln.
- Außerdem weitere Konfigurationsmöglichkeiten und Tools.

Mir sind als kleine, nützliche Spezialitäten aufgefallen, dass neben den Standardfontgrößen Befehle für eine ganz kleine (`\miniscule`), und eine weitere ganz große (`\HUGE`) Zeichenform geschaffen wurden. Während die Standardgrößen von 5 pt bis 25 pt gehen, reicht bei `memoir` die Spanne von 4 pt bis 132 pt.

Es existiert eine Vielfalt an Befehlen für Fußnoten, z. B. Fußnoten mit verbatim-Text `\verbfootnote{}`, außerdem `\twocolumnfootnotes` oder `\paragraphfootnotes`. Mit dem Makro `\fancybreak{...}` lassen sich selbstdefinierte Unterbrechungslinien einfügen, beispielsweise kleine Asterisks wie in Abbildung 5.

```
1 \fancybreak{*}\{* * * * *}\{*}
```

Nachfolgend soll speziell auf den vierten Punkt von Madsen, das Anpassen von Kapitelüberschriften und auf einige Aspekte der Seitenstile eingegangen werden. Damit wird m. E. die Funktionsweise von `memoir` sichtbar.

Kapitel- und Seitenstile

`memoir` hat über zwanzig vordefinierte Kapitelstile, die sich vor allem in den Kapitelüberschriften unterscheiden. Außerdem können Kapitelstile neu definiert werden. Etliche der eingebundenen Stile sind von Nutzern geschaffen worden. Sie werden dann in der Dokumentation auch teilweise mit kritischen Kommentaren und Verbesserungsvorschlägen versehen. Zahlreiche Befehle stehen zur Verfügung, um die vorhandenen Stile zu modifizieren.

Den Default-Kapitelstil sehen Sie in der Abbildung 1, die weiteren findet man in der Dokumentation abgebildet. Der `\chapterstyle{article}` ist, wie der Name schon sagt, für die »Simulation der Artikel-Textklasse« [7] vorgesehen (Abbildung 2), die bei den Optionen von `memoir` geladen werden kann:

```
1 \documentclass[11pt,article,oneside]{memoir}
```

Er kann, wie die anderen vordefinierten Kapitelstile, in die Präambel eingebunden werden.

Es existiert auch ein Kapitelstil »komalike«. Um noch ein expressives Beispiel zu nennen, der Stil `pedersen`: Einzeilige Überschriften in kursiven Buchstaben mit der Nummerierung im rechten Seitenrand. Die Überschrift und/oder die Nummerierung können zusätzlich eingefärbt werden. Neben dem `graphicx`-Paket wird dafür auch `color` bzw. `xcolor` benötigt (wie in Abbildung 3).

```
1 \usepackage{graphicx,xcolor}
2 \definecolor{nicered}{rgb}{.647,.129,.149}
3 \renewcommand{\colorchapnum}{\color{nicered}}
4 \renewcommand{\colorchaptitle}{\color{nicered}}
5 \chapterstyle{pedersen}
```

1 Forschungsstand

1.1 Vergangenheit

1.2 Gegenwart

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen

Abbildung 2: article-Kapitelstil

Forschungsstand

1

1.1 Vergangenheit

1.2 Gegenwart

Abbildung 3: pedersen-Kapitelstil

Ohne Einfärbungen genügt `\chapterstyle{pedersen}` in der Präambel.

Überschriften unter der Kapitelebene (bis zu Subparagraphen) können hinsichtlich ihres vertikalen Abstandes, der Einrückung vom linken Seitenrand und der Schriftgröße konfiguriert werden.

So beschreibt der Wert des `length`-Arguments die Größe des Einschubs der Überschrift (Nummer und Titel) vom linken Seitenrand. Er beträgt normalerweise 0 pt.

```
1 \set<S>indent{<length>}%      allgemein
2 \setsubsecindent{<length>}% für eine Subsectionüberschrift
```

So kann man auch das Aussehen vom Kapitelstil `pedersen` auf der ersten Unter-ebene mit der folgenden Definition in kursiver Schrift linksbündig setzen (siehe Abbildung 4).

```
1 \setsecheadstyle{\itshape\raggedright}
```

Mit dem folgenden Makro kann man die Schriftart für die Nummerierung und den Titel modifizieren. `<S>` steht beispielsweise für `subsection`.



Abbildung 4: Veränderte Unterüberschrift

```
1 \set<S>headstyle{<font>}
```

Neben dem üblichen Setzen der Nummerierungstiefe mit `\setcounter{secnumdepth}{...}` kann dies auch zusätzlich für die subsections erfolgen, sowohl generell in der Präambel als auch temporär im `mainmatter`-Bereich des Buches:

```
1 \setsecnumdepth{<secname>}
2 \maxsecnumdepth{<secname>}
```

`\setsecnumdepth{<secname>}` und `\maxsecnumdepth{<secname>}` haben ein Argument, das entweder eine Zahl oder ein Gliederungsname sein muss:

```
1 \setsecnumdepth{<secname>|<number>}
2 \setsecnumdepth{2}
3 \setsecnumdepth{section}
```

So ergäbe der Befehl `\setcounter{secnumdepth}{2}`, dass die folgenden Subsections nummeriert sein würden, und `\setsecnumdepth{section}` alle Sections. Die allgemeine Nummerierungs- und Gliederungstiefe zeigt die folgende Tabelle 1 (genauer in [8, Kap. 6]).

Der Stil der Standardkopfzeile in Versalien entspricht nicht jedermanns Geschmack (vgl. Abbildung 5) und lässt sich mit definierten Makros bzw. den eingebundenen Paketen abändern: So kann man den Seitenkopf mit Bezug auf einen Seitenstil umdefinieren. Beispielsweise ergibt der nachfolgende Code eine Gestaltung wie in [3] (vgl. Abbildung 6).

```
1 \makeatletter
2 \addtopsmarks{companion}{}{ }%
3 \createmark{chapter}{left}{shownumber}{\@chapapp\ }{. \ }%
4 }
5 \pagestyle{companion}
6 \makeatother
```

Tabelle 1: Zusammenhang zwischen Gliederungsbefehl und interner Nummerierung der Überschriftsebenen

Gliederungsteil	Ebene
<code>\book</code>	-2
<code>\part</code>	-1
<code>\chapter</code>	0
<code>\section</code>	1
<code>\subsection</code>	2
<code>\subsubsection</code>	3
<code>\paragraph</code>	4
<code>\subparagraph</code>	5

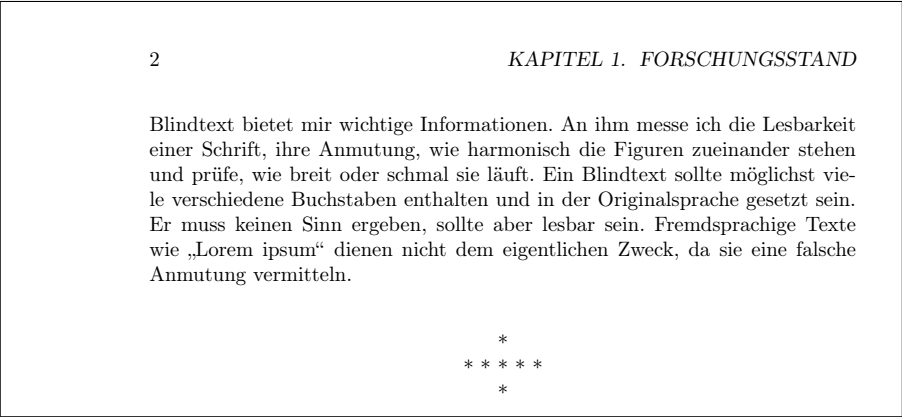


Abbildung 5: Standardkopfzeile

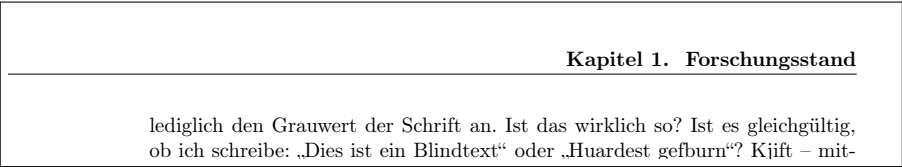


Abbildung 6: Kopfzeile nach dem LATEX-Begleiter

Bei Stilen ohne Linie unter der Kopfzeile kann mit dem folgenden Makro eine Linie hinzufügen.

```
1 \makeheadrule{headings}{\textwidth}{\normalrulethickness}  
2 \pagestyle{...}
```

Welche Makros zu einem Seitenstil gehören und wie man sich diese einrichten kann, wird sehr gut in [2] beschrieben. Für das Erzeugen eines eigenen Seitenstils existieren drei Makros. Es kann also auch mit Bezug auf einen schon vorhandenen erfolgen (`\aliaspagestyle`):

```
1 \makepagestyle{<name>}
2 \coppagestyle{<new name>}{<old name>}
3 \aliaspagestyle{<alias>}{<original>}
```

Zusammenfassend zeigt die Beschäftigung mit memoir, dass damit eine Dokumentenklasse geschaffen wurde, die es sowohl $\text{\LaTeX}$ -Anfänger/innen als auch fortgeschrittenen $\text{\TeX}$ ies ermöglicht, ansprechende komplexe Textformen zu setzen. Veränderungen am Gesamtlayout setzen aber ein gewisses Stilempfinden voraus.

Literatur

- [1] Andreas Heinze: *Memoir Eine Klasse für sich*. PowerPoint-Präsentation 15.06.2004, http://www2.informatik.hu-berlin.de/~heinze/latex_memoir/index.html.
- [2] Lars Madsen: *Page styles on steroids or memoir makes page styling easy*. In: The $\text{\PracTeX}$ Journal, 2008, No. 2, Article revision 2008/7/24, <http://tug.org/pracjourn/2008-2/madsen/madsen.pdf>.
- [3] Frank Mittelbach und Michel Goossens: *Der $\text{\LaTeX}$ -Begleiter*. Pearson Studium: München, 2. überarbeitete und erweiterte Auflage, 2005.
- [4] Markus Kohm und Jens-Uwe Morawski: *KOMA-Script ein wandelbares $\text{\LaTeX} 2_{\epsilon}$ -Paket*, CTAN:macros/latex/contrib/koma-script/scrguide.pdf.
- [5] Ralf Turttschi: *Praktische Typografie*. Niggli: Zürich, 4. Auflage, 2000.
- [6] Herbert Voss: *$\text{\LaTeX}$ Quick Reference*. UIT: Cambridge, England, 2012.
- [7] Peter R. Wilson/Lars Madsen: *The memoir class*. In: The $\text{\PracTeX}$ Journal, 2006, No. 3, Article revision 2006/08/19, <http://www.tug.org/pracjourn/2006-3/wilson/wilson.pdf>.
- [8] Peter R. Wilson/Lars Madsen: *The Memoir Class for Configurable Typesetting. User Guide*. The Herries Press, Normandy Park, WA 2011, CTAN:macros/latex/contrib/memoir/memman.pdf.

Tipps und Tricks

Anwendung aktiver Zeichen

Herbert Voß

T_EX kennt bekanntermaßen als Grundelemente Primitive und auf Anwenderebene die Makros, die nach vollständiger Expansion nur noch aus Primitiven bestehen. Sowohl Makros als auch Primitive werden mit einem Backslash eingeleitet. Eine Sonderrolle spielen die aktiven Zeichen, die ohne diesen vorangehenden Backslash sich wie ein Makro verhalten können.

Einleitung

Man kann sich einen zu verarbeitenden Quelltext als zweidimensionale Matrix vorstellen; in der ersten Spalte steht ein Zeichen oder Befehl (Token) und in der zweiten Spalte die zugehörige Kategorie, in T_EX-Notation: Der »catcode«. Normalerweise sind die einfachen Buchstaben der Kategorie 11 und die Ziffern der Kategorie 12 zugeordnet. Die Kategorie 13 kennzeichnet dagegen ein aktives Zeichen, welches nicht notwendigerweise ein Buchstabe sein muss. Eines der bekannten aktiven Zeichen ist die Tilde ~, die man ohne vorangehenden Backslash als Befehl verwenden kann, denn sie ist formal wie ein Anwendermakro definiert (siehe `latex.ltx`):

```
\catcode '~=13           % Zeichen als aktiv definieren
\def~{\nobreakspace{}}   % entspricht prinzipiell \def~{...}
```

Dieses aktive Zeichen lässt sich auch undefinieren:

```
\def~{\textvisiblespace}
Ein~kleiner~Versuch~mit~einem~aktiven~Zeichen
```

Ein_kleiner_Versuch_mit_einem_aktiven_Zeichen

Anwendung

Die Möglichkeit, jedes beliebige Zeichen der Kategorie 13 zuordnen zu können und ihm dadurch Makroeigenschaften zu verleihen, kann in bestimmten Anwendungsfällen hilfreich sein. Im Folgenden soll eine Matrix, die formal aus lauter Nullen und

Einsen besteht, zur besseren Darstellung statt der Ziffern mit farbigen Rechtecken markiert werden, um so die Unterscheidung zwischen 0 und 1 deutlicher werden zu lassen. Dazu könnte man in Analogie zu obigem Beispiel einfach diese beiden Ziffern aktiv machen:

```
\catcode '\0=13
\catcode '\1=13
```

Da die Tabelle aber jeweils eine Überschriftenzeile und -spalte enthält, deren Zellen auch diese Ziffern enthalten, kann man diese Methode nicht ohne besondere Vorkehrungen verwenden, denn die Überschriften sollen unverändert bleiben. Das folgende Beispiel zeigt das unbrauchbare Ergebnis.

```
\[ \begin{pmatrix}% Normale Ausgabe
x\backslash y & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6\\
0.1 & 1 & 0 & 1 & 1 & 0 & 1\\
0.2 & 0 & 1 & 1 & 1 & 0 & 1\\
0.3 & 1 & 1 & 0 & 1 & 1 & 0\\
\end{pmatrix} \]
```

```
%
\begingroup
\catcode '\0=13 \catcode '\1=13
\def0{\colorbox{black}{\phantom{I}}}
\def1{\colorbox{black!35}{\phantom{I}}}%
%
```

```
\[ \begin{pmatrix}% Manipulierte Ausgabe
x\backslash y & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6\\
0.1 & 1 & 0 & 1 & 1 & 0 & 1\\
0.2 & 0 & 1 & 1 & 1 & 0 & 1\\
0.3 & 1 & 1 & 0 & 1 & 1 & 0\\
\end{pmatrix} \]
```

```
\endgroup
```

$$\begin{pmatrix} x \backslash y & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 \\ 0.1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0.2 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0.3 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} x \backslash y & \text{■} \cdot \text{■} & \text{■} \cdot \text{■} & \text{■} \cdot \text{■} & \text{■} \cdot \text{■} & \text{■} \cdot \text{■} & \text{■} \cdot \text{■} \\ \text{■} \cdot \text{■} & \text{■} & \text{■} & \text{■} & \text{■} & \text{■} & \text{■} \\ \text{■} \cdot \text{■} & \text{■} & \text{■} & \text{■} & \text{■} & \text{■} & \text{■} \\ \text{■} \cdot \text{■} & \text{■} & \text{■} & \text{■} & \text{■} & \text{■} & \text{■} \end{pmatrix}$$

Die Ziffern 0 und 1, die ihre ursprüngliche Bedeutung behalten sollen, müssen maskiert werden, indem man beispielsweise Makros `\N` und `\E` definiert und dabei auf das Primitiv `\string` zurückgreift. Dadurch wird jedes aktive Zeichen durch sich selbst ersetzt und verliert dabei seine aktive Eigenschaft:

```

\begin{group}
\catcode '\0=13
\catcode '\1=13
\def\N{\string0}% Null -> 0
\def\E{\string1}% Eins -> 1
\def0{\colorbox{black}{\phantom{I}}}\def1{\colorbox{black!35}{\phantom{I}}}
%
\[ \begin{pmatrix}
x\backslash y & \N.\E & \N.2 & \N.3 & \N.4 & \N.5 & \N.6\\
\N.\E & 1 & 0 & 1 & 1 & 1 & 0 & 1\\
\N.2 & 0 & 1 & 1 & 1 & 0 & 1\\
\N.3 & 1 & 1 & 0 & 1 & 1 & 0\\
\end{pmatrix} \]
\end{group}

```

$$\begin{pmatrix} x \backslash y & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 \\ 0.1 & \text{gray} & \text{black} & \text{gray} & \text{gray} & \text{black} & \text{gray} \\ 0.2 & \text{black} & \text{gray} & \text{gray} & \text{gray} & \text{black} & \text{gray} \\ 0.3 & \text{gray} & \text{gray} & \text{black} & \text{gray} & \text{gray} & \text{black} \end{pmatrix}$$

Soll dieses Verfahren wiederholt angewendet werden, so ist es etwas mühselig, jedesmal alles innerhalb einer lokalen Gruppe neu definieren zu müssen. Es bietet sich daher an, eine Umgebung `makeActive` zu erstellen, die diese Definition vornimmt. Soll das Verfahren auch für eine Tabelle bei einer erweiterten Spaltendefinition möglich sein, so könnte man auf die folgende Idee kommen:

```

\begin{tabular}{c*6>{\makeActive}c}
x\textbackslash y & \N.\E & \N.2 & \N.3 & \N.4 & \N.5 & \N.6\\
\N.1 & 1 & 0 & 1 & 1 & 0 & 1\\
\N.2 & \relax0 & \relax1 & \relax1 & \relax1 & \relax0 & \relax1\\
\N.3 & \multicolumn{2}{c}{1} & 0 & 1 & 1 & 0\\
\end{tabular}

```

$x \backslash y$	0.1	0.2	0.3	0.4	0.5	0.6
0.1	1	0	1	1	0	1
0.2						
0.3	1		0	1	1	0

Dem Beispiel kann entnommen werden, dass dies nur für die mittlere Tabellenzeile funktioniert. Man muss bei einer Tabelle beachten, dass $\TeX$ den Inhalt einer jeden Zelle prophylaktisch einliest, um das erste nichtexpandierbare Token zu finden. Dies ist in der ersten und zweiten Tabellenzeile ab Spalte 2 immer die 0 oder die 1, womit deren Kategorie aber schon festliegt, bevor `\makeActive` in Aktion tritt. In der mittleren Zeile wird durch $\TeX$ dagegen erst das nicht expandierbare `\relax`

eingelassen, dann `\makeActive` ausgeführt und abschließend die Ziffer gelesen und als aktives Zeichen entsprechend ausgeführt.

Man muss bei der Definition einer Umgebung sicherstellen, dass bereits vor dieser die beiden Ziffern 0 und 1 aktiv sind. Andererseits darf dies nicht für das ganze Dokument gelten, sodass eine etwas trickreiche Definition nötig ist, die auf einem Vorschlag von Heiko Oberdiek beruht:

```
\begingroup
\makeatletter
\catcode'\0=\active
\catcode'\1=\active
\@firstofone{\endgroup
\newenvironment{makeActive}
{
\catcode'\0=\active \catcode'\1=\active
\def0{\colorbox{black}{\phantom{I}}}%
\def1{\colorbox{cyan}{\phantom{I}}}%
}%
\makeatother
\newcommand\N{\string0}
\newcommand\E{\string1}
```

Die Catcodes werden zuerst innerhalb einer Gruppe verändert, sodass sie nach dieser Gruppe wieder ihren alten Wert haben. Mit `\@firstofone` erreicht man hier, dass das folgende Argument mit den gerade gültigen, beziehungsweise richtigen, Catcodes eingelesen wird. Innerhalb der Definition der Umgebung `makeActive` müssen die Catcodes noch einmal auf den Wert 13 gesetzt werden, was hier über das durch $\TeX$ definierte Macro `\active` erfolgen kann; denn 0 und 1 sind bereits aktive Zeichen. Damit die Catcodes auch mit jeder Anwendung der Umgebung verändert werden, muss dies hier noch einmal explizit erfolgen.

Diese Definition mag auf den ersten Blick umständlich erscheinen, ist aber der Tatsache geschuldet, dass man die Catcodes in diesem speziellen Fall nicht so einfach um- und dann wieder redefinieren kann. So wird beispielsweise

```
\catcode'\0=\active
\catcode'\1=\active
\newenvironment{makeActive}
{
\def0{\colorbox{black}{\phantom{I}}}%
\def1{\colorbox{cyan}{\phantom{I}}}%
}%
\catcode'\0=12
\catcode'\1=12
```

nicht funktionieren, weil ein Zurücksetzen mit `\catcode'\0=12` nicht möglich ist, da 1 noch immer ein aktives Zeichen ist und entsprechend durch seine Definition ersetzt wird. Es gibt mehr oder weniger umständliche Möglichkeiten, dies dennoch zu

erreichen, die aber hier nicht weiter diskutiert werden sollen. Mit der oben angegebenen Umgebung kann jetzt auch für Tabellen, die zudem das Makro `\multicolumn` enthalten dürfen, die Umsetzung vorgenommen werden:

```
\begin{makeActive}
\begin{tabular}{*7c}
x\textbackslash y & \N.\E & \N.2 & \N.3 & \N.4 & \N.5 & \N.6\\
\N.\E & 1 & 0 & 1 & 1 & 0 & 1\\
\N.2 & & 0 & 1 & 1 & 1 & 0 & 1\\
\N.3 & & & \multicolumn{2}{c}{1} & 0 & 1 & 1 & 0\\
\end{tabular}
\end{makeActive}
```

x\y	0.1	0.2	0.3	0.4	0.5	0.6
0.1						
0.2						
0.3						

Man kann auch mit den Abständen und der Breite spielen, um die Rechtecke ohne Abstand zueinander zu setzen. Dabei muss allerdings immer beachten werden, dass die beiden Ziffern 0 und 1 aktiv sind:

```
\begin{makeActive}
\makeatletter
\def0{\fboxsep=\string0pt\relax \colorbox{black}{\@arstrut\hphantom
{\,9.9\,}}}%
\def1{\fboxsep=\string0pt\relax \colorbox{red}{\@arstrut\hphantom
{\,9.9\,}}}
\makeatother
\begin{tabular}{c@{\kern2pt} *6{c@{}}}
x\textbackslash y & \N.\E & \N.2 & \N.3 & \N.4 & \N.5 & \N.6\\
\N.\E & 1 & 0 & 1 & 1 & 0 & 1\\
\N.2 & & 0 & 1 & 1 & 1 & 0 & 1\\
\N.3 & & 1 & 1 & 0 & 1 & 1 & 0\\
\end{tabular}
\end{makeActive}
```

x\y	0.1	0.2	0.3	0.4	0.5	0.6
0.1						
0.2						
0.3						

Von fremden Bühnen

Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit der letzten Ausgabe bis zum Redaktionsschluss vor. Die Liste folgt der umgekehrten chronologischen Reihenfolge. Reine Updates werden nicht aufgeführt. Die Angaben können auf der *ctan-ann*-Mailingliste verfolgt werden, die auch über Twitter und Identi.ca als @ctanannounce verfügbar sind.

tui von *Nicolas Vaughan* ist eine Klasse für Arbeiten an der *Universidad de los Andes* in Bogotá.

CTAN:macros/latex/contrib/tui

longnamefilelist von *Uwe Lück* erweitert `\listfiles`, so dass auch Dateinamen mit mehr als acht Zeichen verarbeitet werden können.

CTAN:macros/latex/contrib/longnamefilelist

checkcites von *Paulo Roberto Massa Cereda* ist ein Lua-Skript, das undefinierte oder nicht benutzte Zitationen aus aux- und bib-Dateien ausgibt.

CTAN:support/checkcites

computer-typesetting-using-latex von *Evgeny Baldin* ist eine russische Einführung zu $\TeX$.

CTAN:info/russian/Computer_Typesetting_Using_LaTeX

sepfootnotes von *Eduardo C. Lourenço de Lima* dient zum Sammeln von Fußnotentexten in einer separaten Datei, um den Haupttext zu entlasten.

CTAN:macros/latex/contrib/sepfootnotes

messagebubbles von *Simon Harrer* erlaubt es, Text und Daten in »Sprechblasen« zu setzen.

CTAN:macros/latex/contrib/messagebubbles

media9 von *Alexander Grahn* dient zum Einfügen von Audio- und Videoinhalten in diversen Formaten in PDF-Dateien (kompatibel zu Acrobat-9/X).

CTAN:macros/latex/contrib/media9

lmake von *Shengjun Pan* erleichtert das Setzen von Listen nach einem bestimmten Muster.

CTAN:macros/latex/contrib/lmake

issuulinks von *Boris Veytsman* ändert `hyperref` zur Verwendung mit ISSUU, so dass externe anstelle von internen Hyperlinks erzeugt werden.

CTAN:macros/latex/contrib/issuulinks

bibleref-lds von *Chad Parry* ist eine Erweiterung von *bibleref-mouth*, um Schriften der *Church of Jesus Christ of Latter-day Saints (LDS)* zu zitieren.

CTAN:macros/latex/contrib/bibleref-lds

menukeys von *Tobias Weh* erlaubt die Darstellung von Pfaden in Aufklapp-Menüs oder von Tastenkombinationen mit mehreren Themen.

CTAN:macros/latex/contrib/menukeys

lua-visual-debug von *Patrick Gundlach* dient zum »visuellen Debugging« mit Lua $\TeX$.

CTAN:macros/luatex/generic/lua-visual-debug

footnoterange von *H.-Martin Münch* fasst Fußnoten, die mehrfach auftreten und die innerhalb einer Umgebung *footnoterange* stehen, zusammen (z. B.: 1–4 statt 1, 2, 3, 4) und unterdrückt auf Wunsch Hyperlinks auf die Fußnoten.

CTAN:macros/latex/contrib/footnoterange

droit-fr von *Yves de Saint-Pern* ist eine $\LaTeX$ -Dokumentenklasse und ein Bibliographiestil für juristische Arbeiten an französischen Universitäten.

CTAN:macros/latex/contrib/droit-fr

aeb_mobile von *D. P. Story* formatiert die Ausgabe eines PDF für ein Smartphone.

CTAN:macros/latex/contrib/aeb_mobile

mattex von *Romeo Van Snick* erlaubt den Export von Matlab-Variablen zur weiteren Verwendung mit $\LaTeX$.

CTAN:macros/latex/contrib/mattex

hausarbeit-jura von *Martin Sievers* bietet eine neue Klasse zum Schreiben juristischer Hausarbeiten auf Basis von *jurabook* und *jurabib*, insbesondere für $\LaTeX$ -Anfänger.

CTAN:macros/latex/contrib/hausarbeit-jura

zhmcjk von *Leo Liu* erleichtert das Setup von CJK-Schriften.

CTAN:language/chinese/zhmcjk

japanese-otf-uptex von *Norbert Preining* enthält den up $\LaTeX$ -Support des Japanischen OTF-Pakets *otfbeta*.

CTAN:language/japanese/japanese-otf-uptex

einfuehrung von *Herbert Voß* enthält die Beispieldateien zu dem Buch »Einführung in $\LaTeX$ « aus der Edition $\alpha\alpha\alpha\alpha$.

CTAN:info/examples/Einfuehrung

jfontmaps von *Norbert Preining* enthält die $\LaTeX$ -Unterstützung für japanische Kanji-Schriftzeichen.

CTAN:language/japanese/jfontmaps

pst-pulley von *Thomas Söll* dient zum Zeichnen von Flaschenzügen mithilfe von *PSTricks*.

CTAN:graphics/pstricks/contrib/pst-pulley

texlive-dummy von *Rolf Niepraschk* ist ein Dummy-RPM, das anstelle der $\TeX$ Live-Pakete von openSUSE alle Abhängigkeiten auf dem System einrichtet, ohne die dazugehörigen Dateien zu installieren, für alle Anwender, die ein original $\TeX$ Live verwenden möchten.

CTAN:support/texlive/texlive-dummy

fixltxhyph von *Claudio Beccari* behebt einen Fehler bei der Silbentrennung in $\TeX$, der in den romanischen Sprachen auftritt, wenn ein Apostroph von einem `\emph`-Befehl gefolgt

wird.

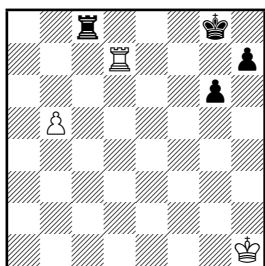
CTAN:macros/latex/contrib/fixltxhyph

gtrcrd von *Riccardo Bresciani* dient zum Setzen von Akkorden über einem Liedertext.

CTAN:macros/latex/contrib/gtrcrd

xpatch von *Enrico Gregorio* erweitert die Befehle aus dem Paket *etoolbox*.

CTAN:macros/latex/contrib/xpatch



Weiß gewinnt in drei Zügen^a

1. b6 Kf8 2. b7 Tb8 3. Tc7

^a nach L.Pachmann:

Moderne Schachstrategie.

ARTIA: Prag 1958, S. 171.

```
%% Christine Römer
\usepackage{texmate}
\setchessfontfamily{leipzig}

\begin{minipage}[t]{0.4\linewidth}
\position{2r3k/3R3p/6p/1P/8/8/8/7K}

\shortstack{\showboard \\[5pt]
Weiß gewinnt in drei Zügen\footnote{nach L.Pachmann:\newline
Moderne Schachstrategie.\newline ARTIA: Prag 1958, S.\,171.}}
\end{minipage}

\begin{minipage}[b]{0.3\linewidth}
\footnotesize\rotatebox{90}{1.\,b6 Kf8 2.\,b7 Tb8 3.\,Tc7}
\end{minipage}
```

Bücher und Rezensionen

Edition dante – Neuauflagen

Markus Kohm und Jens-Uwe Morawski:
KOMA-Script – Eine Sammlung von Klassen
und Paketen für $\text{\LaTeX}$ 2_ε; 4. Auflage;
DANTE e.V., Lehmanns Media, 2012
592 Seiten; ISBN 978-3-86541-459-5;
19,95 € (Ladenpreis) bzw. 15,- € für Mitglieder
von DANTE e.V., jeweils versandkostenfrei.



Herbert Voß:
Mathematiksatz mit $\text{\LaTeX}$; 2. Auflage;
DANTE e.V., Lehmanns Media, 2012
384 Seiten; ISBN 978-3-86541-485-4;
19,95 € (Ladenpreis) bzw. 15,- € für Mitglieder
von DANTE e.V., jeweils versandkostenfrei.



Bestellung

Bitte schicken Sie eine E-Mail an office@dante.de mit Angabe von *Name*, *Anschrift*, *Mitgliedsnummer* und *Anzahl der Exemplare*, und überweisen Sie den Betrag auf das Konto von DANTE e.V. oder bezahlen Sie per PayPal. Die Kontonummer finden Sie am Ende dieses Heftes und Informationen zu PayPal auf <http://www.dante.de/index/Intern/Zahlung.html>.

Bitte beachten Sie für Bestellungen bei DANTE e.V. folgende Informationen zum Widerrufsrecht: Käufer können bei Bestellungen per E-Mail, Internet, Brief oder Telefon den Kaufvertrag innerhalb einer Frist von 14 Tagen ab Erhalt der Ware per Brief, Fax oder E-Mail oder durch Rücksendung der Ware widerrufen (siehe Kontaktadresse). Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs oder der Ware. Bei einem Bestellwert bis 40,- € hat der Besteller die Rücksendekosten zu tragen. Bei Verschlechterung der Ware, die über die übliche Prüfung der Ware hinausgeht, hat der Besteller gegebenenfalls Wertersatz zu leisten.



DANTE 2012 – Museum für Druckkunst (Fotos: Uwe Ziegenhagen)

Leserbriefe

Anmerkungen zu »Texte und Tabellen mit Lua $\text{\LaTeX}$ «, Heft 2/2011

Annette Huck

Vielen Dank für den ausführlichen und instruktiven Artikel zu Lua $\text{\LaTeX}$ in Heft 2/2011. Mit Hilfe dieses phantastischen Beitrags ist es mir gelungen, ohne größere Einarbeitungszeit in Lua, meine bisherigen $\text{\LaTeX}$ -Vorlagen zum Rechneschreiben zu überarbeiten und die entsprechenden Beträge nun automatisch berechnen zu lassen. Seither geht das Erstellen von Rechnungen begeisternd schnell und effizient.

Neulich bin ich auf einen kleinen Rundungsfehler gestoßen. Bei einem Nettobetrag von beispielsweise 1399,99 ergibt sich bei einem Umsatzsteuersatz von 19% der Betrag der Umsatzsteuer zu 265,9981. Von der im Heft vorgestellten Funktion wird dieser Betrag nun fehlerhaft auf 265,00 gerundet, anstatt korrekt auf 266,00.

Das Problem taucht immer auf, wenn eine Zahl der Form $\ast,99i\ast$ mit $5 \leq i \leq 9$ kaufmännisch gerundet werden soll.

Statt Code-Zeile 45/46 auf Seite 43 in Heft 2/2011

```
MWSt = string.format( "%d,%02d", mwst, ( mwst * 100 + 0.5 ) % 100 )
```

müsste es

```
MWSt = string.format( "%d,%02d", ( mwst * 100 + 0.5 ) / 100, ( mwst * 100 + 0.5 ) % 100 )
```

lauten, damit beim Runden ein möglicher Übertrag in die Vorkommastellen berücksichtigt wird. Eine alternative Lösung wäre, dass man das kaufmännische Runden und das Ersetzen von Punkt durch Komma nacheinander von Lua-Funktionen erledigt:

```
MWSt = string.gsub( string.format( "%.02f", mwst ), "%.", ", " )
```

Man lässt Lua die Variable `mwst` mit »%.02f« auf zwei mit Nullen aufgefüllte Nachkommastellen runden. Das `%f` im Format-String steht wie bei `printf` in der Sprache C als Platzhalter für eine Fließkommazahl. Der Punkt zwischen `%` und `f` dient als

Trennzeichen. Links vom Punkt kann man die Mindestfeldbreite angeben (hier nicht geschehen), und rechts davon die Genauigkeit (hier »02«). Dabei sorgt 0 für das Auffüllen mit Nullen und 2 gibt die Anzahl der Nachkommastellen an, welche glücklicherweise gerundet und nicht abgeschnitten werden. Danach wird mit `string.gsub` jeder Punkt „%.“ durch ein Komma „.“ ersetzt. Dem Punkt muss in diesem Fall ein umschaltendes % vorangestellt werden, weil der Punkt allein ein Sonderzeichen in einem regulären Ausdruck wäre.

Allerdings taucht dieser Rundungsfehler bei der vorgeschlagenen Benutzung doch recht selten auf, weil ja nur bei einem von 100 möglichen Cent-Werten, nämlich bei *,99* €, überhaupt eine Veränderung der Euro-Stellen beim Runden möglich ist, und dann auch nur beim Aufrunden. Ich denke, es interessiert auch andere Leserinnen und Leser, diese Funktion zu korrigieren, aber man muss deshalb noch lange keinen Steuererklärungsalptraum für die bereits erstellten Rechnungen befürchten; wahrscheinlich ist dieser Fall bisher bei den meisten einfach noch gar nicht vorgekommen.



Kassenprüfer bei der Arbeit (Fotos: Herbert Voß)

Spielplan

Termine

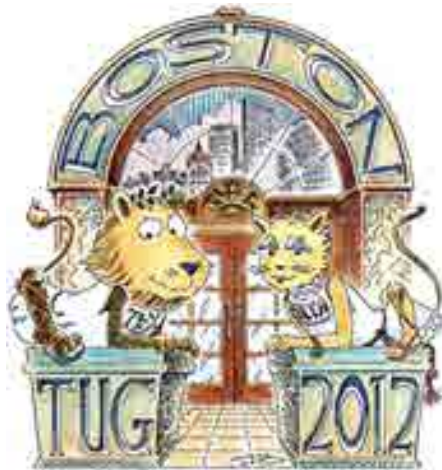
2012

29. 4. – 3. 5. **20. BachoT_EX-Konferenz 2012**
Bachotek, nahe Brodnica, Polen
<http://www.gust.org.pl/bachotex/2012/>

23. 5. – 26. 5. **Linuxtag Berlin**
Messegelände
14055 Berlin
<http://www.linuxtag.org/2012/>

16. 7.–18. 7. **TUG 2012**
Boston, Massachusetts (USA)
<http://tug.org/tug2012/>

8. 10. – 12. 10. **EuroT_EX 2012**, 6th International ConT_EXt meeting und
DANTE-Herbsttagung mit 47. Mitgliederversammlung
Breskens (Niederlande)
<http://www.ntg.nl/eurotex2012/>



Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßige Treffen von T_EX-Anwendern statt, die für jeden offen sind. Im WWW gibt es aktuelle Informationen unter <http://projekte.dante.de/Stammtische/WebHome>.

Aachen

Torsten Bronger, bronger@physik.rwth-aachen.de

Gaststätte Knossos, Templergraben 28, 52062 Aachen

Zweiter Donnerstag im Monat, 19.00 Uhr

Berlin

Michael-E. Voges, Tel.: (03362) 50 18 35,

mevoges@t-online.de

Ort derzeit wechselnd

Zweiter Donnerstag im Monat, 19.00 Uhr

Bremen

Winfried Neugebauer, Tel.: 0176 60 85 43 05, tex@wphn.de

Wechselnder Ort

Erster Donnerstag im Monat, 18.30 Uhr

Darmstadt

Karlheinz Geyer, geyer.k.fv.tu@nds.tu-darmstadt.de, <http://www.da-tex.org>

Wechselnder Ort

Erster Freitag im Monat, ab 19.30 Uhr

Erlangen

Walter Schmidt, Peter Seitz, w.a.schmidt@gmx.net

Gaststätte »Deutsches Haus«, Luitpoldstraße 25, 91052 Erlangen

Dritter Dienstag im Monat, 19.00 Uhr

Frankfurt

Harald Vajkonny, <http://wiki.lug-frankfurt.de/TeXStammtisch>

Restaurant »moschmosch«, Wilhelm-Leuschner-Straße 78, 60329 Frankfurt

Vierter Donnerstag im Monat, 19.30 Uhr

Hamburg

Lothar Fröhling, lothar@thefroehlings.de

Restaurant Sandstuv, Neue Straße 17, 21073 Hamburg-Harburg

Letzter Dienstag im Monat, 19.00 Uhr

Hannover

Mark Heisterkamp, heisterkamp@rrzn.uni-hannover.de

Seminarraum RRZN, Schloßwender Straße 5, 30159 Hannover

Zweiter Donnerstag im Monat, 18.30 Uhr

Heidelberg

Martin Wilhelm Leidig, Tel.: (06203) 40 22 03, moss@moss.in-berlin.de

Anmeldeseite zur Mailingliste: <http://mailman.moss.in-berlin.de/mailman/listinfo/stammtisch-hd-moss.in-berlin.de>

Wechselnder Ort

Letzter Freitag im Monat, ab 19.30 Uhr

Karlsruhe

Klaus Braune, Tel.: (0721) 608-4 40 31, klaus.braune@kit.edu,

SCC (Steinbuch Centre for Computing) des KIT (vormals Universität Karlsruhe, Rechenzentrum),

Zirkel 2, 2. OG, Raum 203, 76131 Karlsruhe

Erster Donnerstag im Monat, 19.30 Uhr

Köln

Helmut Siegert

Institut für Kristallographie, Zülpicher Straße 49b, 50674 Köln

Letzter Dienstag im Monat, 19.30 Uhr

Leipzig

Erhard Pross, erhard.pross@gmx.de, Gerrit Imsieke, <http://projekte.dante.de/Stammtische/>

Leipzig

vierteljährliche Treffen, Ankündigung auf der Webseite

München

Uwe Siart, uwe.siart@tum.de, <http://www.siart.de/typografie/stammtisch.xhtml>

Erste Woche des Monats an wechselnden Tagen, 19.00 Uhr

Stuttgart

Bernd Raichle, bernd.raichle@gmx.de

Bar e Ristorante »Valle«, Geschwister-Scholl-Straße 3, 70197 Stuttgart

Zweiter Dienstag im Monat, 19.30 Uhr

Trier

Martin Sievers, stammtisch-trier@texberatung.de

nach Vereinbarung

Wuppertal

Andreas Schrell, Tel.: (02193) 53 10 93, as@schrell.de

Restaurant Croatia »Haus Johannisberg«, Südstraße 10, 42103 Wuppertal

Zweiter Donnerstag im Monat, 19.30 Uhr

Würzburg

Bastian Hepp, LaTeX@sning.de

nach Vereinbarung

Adressen

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

Tel.: (0 62 21) 2 97 66 (Mo., Mi.–Fr., 10.00–12.00 Uhr)

Fax: (0 62 21) 16 79 06

E-Mail: dante@dante.de

Konto: VR Bank Rhein-Neckar eG

BLZ 670 900 00

IBAN DE67 6709 0000 0002 3100 07

Kontonummer 2 310 007

SWIFT-BIC GENODE61MA2

Präsidium

Präsident: Martin Sievers president@dante.de

Vizepräsident: Herbert Voß vice-president@dante.de

Schatzmeister: Klaus Höppner treasurer@dante.de

Schriftführer: Manfred Lotz secretary@dante.de

Beisitzer: Patrick Gundlach

Volker RW Schaa

Uwe Ziegenhagen

Ehrenmitglieder

Peter Sandner	22. März 1990	Klaus Thull	22. März 1990
Yannis Haralambous	5. Sep. 1991	Barbara Beeton	27. Feb. 1997
Luzia Dietsche	27. Feb. 1997	Donald E. Knuth	27. Feb. 1997
Eberhard Mattes	27. Feb. 1997	Hermann Zapf	19. Sep. 1999

Server

DANTE: <http://www.dante.de/> (Rainer Schöpf, Joachim Schrodtt)

CTAN: <http://mirror.ctan.org/>

FAQ

DTK: <http://projekte.dante.de/DTK/WebHome>

T_EX: <http://projekte.dante.de/DanteFAQ/WebHome>

Autoren/Organisatoren

Doris Behrendt Pfortleinsgasse 1 97318 Biebelried Doris.Behrendt@me.com	[19]	Martin Wilhelm Leidig Nadlerstraße 6 68526 Ladenburg moss@moss.in-berlin.de	[19]
Marco Daniel Beerentaltrift 90g 21077 Hamburg marco.daniel@mada-nada.de	[39]	Manfred Lotz manfred@dante.de	[7]
Jürgen Fenn Friedensallee 174/20 63263 Neu-Isenburg juergen.fenn@gmx.de	[88]	Christine Römer Institut für germanistische Sprachwissenschaft FSU Jena Christine.Roemer@uni-jena.de	[72, 90]
Roland Geiger Heiterblickallee 4 04329 Leipzig	[23]	Martin Sievers Im Treff 8 54296 Trier martin@dante.de	[4]
Patrick Gundlach Eisenacher Straße 101 10781 Berlin Patrick@gundla.ch	[60]	Herbert Voß Wasgenstraße 21 14129 Berlin herbert@dante.de	[3, 83, 91]
Jürgen Hanneder Deutschhausstraße 12 35032 Marburg Juergen.Hanneder@staff.uni-marburg.de	[53]	Dominik Waßenhoven Marsweg 93 90763 Fürth dominik@wassenhoven.info	[47, 50]
Annette Huck annette@hopfenwiesen.de	[93]		

Die T_EXnische Komödie

24. Jahrgang Heft 2/2012 Mai 2012

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 7 Beschlüsse der 46. Mitgliederversammlung von DANTE e.V.
- 14 Bericht des Schatzmeisters 2011
- 20 Bericht der Kassenprüfer zum Vereinsjahr 2011

T_EX-Theatertage

- 23 DANTE 2012

Bretter, die die Welt bedeuten

- 39 Das Paket xparse – Dokumentenmakros auf Basis expl3
- 48 biblatex mit tufte-latex verwenden
- 51 Aufrechte Klammern in kursivem Text
- 54 Der T_EXnische Fortschritt und seine Tücken
- 61 Strichcodes erzeugen mit LuaT_EX
- 72 Bücher schreiben mit der L^AT_EX-Dokumentenklasse memoir

Tipps und Tricks

- 83 Anwendung aktiver Zeichen

Von fremden Bühnen

- 88 Neue Pakete auf CTAN

Bücher und Rezensionen

- 91 Edition *dante* – Neuauflagen

Leserbriefe

- 93 Anmerkungen zu »Texte und Tabellen mit LuaL^AT_EX«, Heft 2/2011

Spielplan

- 95 Termine
- 96 Stammtische

Adressen

- 99 Autoren/Organisatoren