

Textverarbeitungen: Dumm und Ineffizient

von Allin Cottrell

(cottrell@wfu.edu)

übersetzt von Hannes Keil und Claudia Thormann

(keil@in.tum.de und thormann@in.tum.de)

Inhaltsverzeichnis

1 Die Behauptung	2
2 Gedruckte Dokumente	2
2.1 Schreiben versus Typographie	3
2.2 Die garstigen Seiten von WYSIWYG	4
2.3 Die Struktur eines Dokuments	5
2.4 Texteditoren	5
2.5 Die Vorzüge von ASCII	6
2.6 Das Satzprogramm	7
2.7 Es zusammensetzen	9
3 Digitale Verbreitung	9
3.1 Einfache Schriftstücke	10
3.2 Kompliziertere Dokumente	10
4 Voraussetzung	11
5 Schimpf, schimpf	11
6 Verweise	13

1 Die Behauptung

Ein Textverarbeitungsprogramm ist ein dummes und äußerst ineffektives Werkzeug, um Texte für die Kommunikation mit anderen aufzubereiten. Das ist die Behauptung, die ich im Folgenden darlegen werde. Es mag Ihnen zuerst seltsam vorkommen, wenn ich gegen Textverarbeitungen bin. Was schlage ich dann vor: Dass wir mit der Hand schreiben oder mechanische Schreibmaschinen verwenden? Nein. Obwohl einige Punkte für diese Vorgehensweisen sprechen, gehe ich davon aus, dass die meisten Leser dieses Aufsatzes den überwiegenden Teil ihrer Schreibarbeit mit dem Computer erledigen – so wie ich. Meine Behauptung ist, dass es sehr viel bessere Wege gibt, einen Text mit dem Computer aufzubereiten, als mit einem Textverarbeitungsprogramm.

Der Wortlaut meiner Behauptung ist provokant formuliert, aber lassen Sie mich eins klarstellen: Wenn ich sage, *Textverarbeitungsprogramme* sind dumm, dann sage ich nicht, dass *Sie*, wenn Sie so ein Programm benützen, dumm sind. Ich ziehe über eine Technologie her, allerdings eine, die eifrig von den großen Softwarehäusern gefördert wird und die dadurch ein *de facto* Standard geworden ist. Wenn Sie nicht zufällig zur rechten Zeit am rechten Ort waren, ist Ihnen wahrscheinlich gar nichts von Alternativen bekannt. Diese Alternativen werden von den großen Softwarehäusern keineswegs unterstützt und gefördert – aus gutem Grund: Wie Sie sehen werden, sind diese Alternativprogramme frei erhältlich.

Lassen Sie uns beginnen, indem wir uns erst einmal vom Endprodukt entfernen. Text, der geschrieben wurde, um Ideen und Informationen an andere weiterzureichen, wird auf zwei Arten verbreitet:

1. Als „Hardcopy“, also in der traditionellen gedruckten Form auf Papier, oder
2. elektronisch. Das bedeutet, via E-Mail, Internetseiten oder als Dokumente, die dazu gedacht sind, am Bildschirm betrachtet zu werden.

Es gibt hier eine Überlappung. Zum Beispiel kann ein Dokument, das dazu gedacht ist, ausgedruckt zu werden, auch elektronisch verbreitet werden, in der Hoffnung, dass der Empfänger die Möglichkeit hat, die Datei auszudrucken. Aber lassen Sie uns diese beiden Arten der Verbreitung im Einzelnen betrachten.

2 Gedruckte Dokumente

Sie wollen einen Text an Ihrer Computertastatur schreiben und ihn dann auf Ihrem Drucker schön ausdrucken. Natürlich wollen Sie nicht, dass das in Echtzeit geschieht (dass also sofort etwas gedruckt wird, nachdem Sie es geschrieben haben). Sie wollen den Text erst schreiben und ihn dann in digitaler Form auf irgend einem Speichermedium speichern. Sie wollen in der Lage sein, den Text wieder zu laden, ihn nach Belieben zu verändern und ihn dann zu drucken, wenn es soweit ist. Sicherlich ist eine Textverarbeitung – wie der Marktführer MS Word – der „natürliche“ Weg, all dies zu tun? Nun, es ist ein Weg, aber nicht der beste. Warum nicht?

2.1 Schreiben versus Typographie

Um einen druckbaren Text mit einer Textverarbeitung zu erstellen, werden Sie effektiv gezwungen, zwei Aufgaben auf einmal zu bewältigen, die *inhaltlich* verschieden sind und die – um sicherzustellen, dass die investierte Zeit bestmöglich genutzt wird und die schlussendliche Kommunikation möglichst erfolgreich verläuft – auch *praktisch* verschieden sein sollten. Diese beiden Aufgaben sind

1. das Schreiben des Textes selbst. Damit meine ich die jeweilige Wortwahl, um die Ideen und die logische Struktur des Textes klar auszudrücken. Letzteres kann unterschiedliche Formen annehmen, je nachdem, um welchen Text es sich handelt. In der Struktur enthalten sind Punkte wie die Einteilung des Textes in Absätze, Abschnitte oder Kapitel, die Entscheidung, ob eine Bemerkung als Fußnote oder im Fließtext erscheint, das Hervorheben bestimmter Textpassagen, die Markierung von Zitaten gegenüber den Worten des Autors.
2. die Typographie des Textes. Darunter versteht man Aspekte wie die Wahl der Schriftart, in der der Text gedruckt wird und die Art, wie strukturelle Elemente dargestellt werden. Sollen Abschnittsüberschriften in breiter Schrift oder mit Kapitälchen gedruckt werden? Sollen sie linksbündig oder zentriert sein? Soll der Text in Blocksatz erscheinen oder nicht? Sollen Anmerkungen am Seitenende oder nach jedem Kapitel stehen? Soll der Text ein- oder zweispaltig gesetzt werden? Und dergleichen mehr.

Der Autor eines Textes sollte sich – zumindest in erster Linie – voll und ganz auf die erste dieser beiden Aufgaben konzentrieren. Das ist die Aufgabe des Autors. Adam Smith¹ wies bekanntermaßen auf die großen Vorteile hin, die durch Arbeitsteilung entstehen. Das Erstellen und die logische Strukturierung sind der Beitrag des Autors zur Erstellung eines gedruckten Textes. Typographie ist die Arbeit des Typographen. Diese Arbeitsteilung wurde natürlich bei der traditionellen Erstellung von Büchern und Artikeln in der Prä-Computer-Ära eingehalten. Der Autor schrieb und markierte für den Herausgeber die logische Struktur des Textes durch verschiedene Anmerkungen. Der Typograph übersetzte den Text des Autors in ein gedrucktes Werk indem er die Anmerkungen des Autors über die logische Struktur in ein konkretes typographisches Design übertrug. Man muss sich nur einmal vorstellen, dass sich zum Beispiel Jane Austen überlegt, in welcher Schrift sie die Kapitelüberschriften von „*Stolz und Vorurteil*“² schreiben möchte, um zu sehen, wie lächerlich diese Vorstellung ist. Jane Austen war eine großartige Schriftstellerin; sie war keine Typographin.

Sie werden jetzt denken, dies sei etwas abseits vom Problem. Jane Austens Werke waren gut genug, um veröffentlicht zu werden; professionelle Typographen hatten ein Interesse daran, diese Werke zu setzen und zu drucken. Sie und ich sind nicht in dieser glücklichen Lage. Wenn wir einen gedruckten Artikel haben wollen, müssen wir ihn

1 Anm. d. Ü.: Adam Smith (1723–1790) war schottischer Volkswirtschaftler und schuf das erste geschlossene System einer Volkswirtschaftslehre, auf das alle späteren Theoretiker Bezug nahmen.

2 Anm. d. Ü.: im Original „Pride and Prejudice“

uns selbst drucken (und nebenbei wollen wir das noch etwas schneller haben als mittels traditioneller Typographie). Nun, ja und nein. Wir müssen es irgendwie schon selbst machen (mit Hilfe unserer Computer), aber uns stehen eine Menge Hilfsmittel zur Verfügung – insbesondere ein professionelles Satzprogramm. *Dieses Programm (oder besser diese Programmsammlung) wird für uns genau das tun, was traditionelle Typographen für Shakespeare, Jane Austen, Sir Walter Scott und all die anderen getan haben – umsonst und in wenigen Sekunden oder Bruchteilen von Sekunden.* Wir müssen das Programm nur mit geeigneten Anmerkungen versehen (so genannten Markup-Befehlen), wie es auch die früheren Autoren getan haben.

Ich schlage deshalb vor, dass es in der Erstellung von gedruckten Texten mit dem Computer zwei unterscheidbare „Augenblicke“ geben sollte. Zuerst schreibt man seinen Text und erstellt seine logische Struktur korrekt, indem man diese mittels geeigneter Anmerkungen im Text beschreibt. Dies erreicht man mit einem *Texteditor*, einem kleinen Programm, das nicht mit einem Textverarbeitungssystem verwechselt werden sollte (ich werde die Unterschiede weiter unten behandeln). Dann „übergibt“ man seinen Text an ein Satzprogramm, das in sehr kurzer Zeit eine schön gesetzte Kopie des Textes zurückliefert.

2.2 Die garstigen Seiten von WYSIWYG

Diese beiden Arbeitsschritte werden mit den modernen WYSIWYG-Textverarbeitungen („What You See Is What You Get“)³ zu einem einzigen Schritt vermischt. Sie schreiben Ihren Text und *während Sie fortfahren* erhält der Text am Bildschirm eine konkrete typographische Erscheinung, die eng an das angelehnt ist, was Sie sehen, wenn Sie das Dokument ausdrucken (trotz allem sieht es aus verschiedenen Gründen nicht immer so aus). In der Tat wird der Text direkt gesetzt, während Sie ihn eintippen. Auf den ersten Blick scheint das eine große Annehmlichkeit zu sein; bei näherer Betrachtung ist es ein Fluch. Hierzu gibt es drei Gründe:

1. Der Autor wird von seiner eigentlichen Arbeit – einen Text zu schreiben – abgelenkt und statt dessen vor typographische Entscheidungen gestellt, zu deren Beantwortung er oder sie nicht genügend Hintergrundwissen hat (das „sich durch Schriftarten und Seitenränder kämpfen“ anstatt sich auf den Inhalt zu konzentrieren).
2. Die Textsatzalgorithmen, die in WYSIWYG-Textverarbeitungen Verwendung finden, opfern Qualität zu Gunsten von Geschwindigkeit, die sie für das Setzen und immer wieder neu Setzen der Benutzereingaben in Echtzeit benötigen. Das Endprodukt ist sehr viel schlechter als das eines echten Textsatzprogrammes.
3. Der Benutzer von Textverarbeitungssystemen unterliegt der starken Versuchung, die logische Struktur des Textes aus den Augen zu verlieren und dies durch oberflächliche typographische Elemente auszugleichen.

³ Anm. d. Ü.: Sinngemäß: Was Sie am Bildschirm sehen ist genau das, was Sie am Drucker erhalten. Ursprünglich kommt dieser Begriff aus der frühen Mac-Welt, als Monitore und Drucker noch die gleiche Auflösung von 75 dpi verwendeten.

Die beiden ersten Punkte sollten selbsterklärend sein. Aber lassen Sie mich den dritten etwas ausführlicher darlegen. Seine Wichtigkeit hängt von der Art des Dokuments ab.

2.3 Die Struktur eines Dokuments

Nehmen Sie zum Beispiel eine Abschnittsüberschrift. Solange die logische Struktur eines Dokuments betrachtet wird, ist das Einzige, was von Bedeutung ist, dass ein bestimmtes Textstück irgendwie als Abschnittsüberschrift „markiert“ ist. Man könnte zum Beispiel tippen `\section{Text der Überschrift}`. Wie diese Überschrift dann in der fertigen Version aussieht, ist eine andere Frage. Wenn Sie eine Textverarbeitung verwenden, ist das, was Sie sehen, auch das (und nur das!), was Sie bekommen. Sie werden gezwungen, über die exakt festgelegte typographische Erscheinung Ihrer Überschrift zu entscheiden, sobald Sie sie schreiben.

Mal angenommen, Sie entscheiden sich dafür, dass Sie Ihre Überschriften in fetter Schrift und etwas größer als den restlichen Text schreiben wollen. Wie erreichen Sie dieses Aussehen? Es gibt mehrere Möglichkeiten, aber für die meisten Menschen ist der offensichtlichste und intuitivste Weg (mal eine komplette WYSIWYG-Umgebung vorausgesetzt), den Text der Überschrift zu schreiben, ihn zu markieren, auf den Knopf für fette Schrift zu klicken, den Auswahldialog für die Schriftgrößen zu öffnen und eine etwas größere Schrift zu wählen. Die Überschrift ist jetzt fett und etwas größer.

Super! Aber was sagt jetzt noch, dass das eine Überschrift ist? Es gibt nichts in Ihrem Dokument, das dieses kleine Stückchen Text logisch als Überschrift markiert. Angenommen, Sie hätten später die Überschrift doch lieber in Kapitälchen geschrieben, römisch durchnummeriert oder zentriert – oder was auch immer. Was Sie jetzt sagen würden, ist „Bitte ändere alle Abschnittsüberschriften so und so“. Aber falls Sie die Formatierungen wie oben durchgeführt haben, müssen Sie das gesamte Dokument durchforsten und jede Überschrift einzeln verbessern.

Nun, es *gibt* einen Weg, die logische Bedeutung von Textstücken in (zum Beispiel) MS Word festzulegen. Sie *können* – wenn Sie sorgfältig vorgehen – erreichen, dass Sie mit einem einzigen Befehl das Aussehen aller Abschnittsüberschriften ändern können. Aber nur wenige Benutzer von Word nutzen dies konsequent, was auch nicht weiter verwunderlich ist: Der WYSIWYG-Ansatz ermutigt nicht gerade, sich mit der Struktur zu beschäftigen. Man kann einfach – eigentlich zu einfach – logische Struktur durch geeignete Formatierung vortäuschen. Auf der anderen Seite ist die Notwendigkeit, die Struktur des Textes zu markieren sofort vorhanden, wenn man mit einem Texteditor schreibt.

2.4 Texteditoren

OK, es ist wahrscheinlich an der Zeit, zu erklären, was ein Texteditor ist und wie er sich von einem Textverarbeitungssystem unterscheidet. Ein moderner Texteditor sieht ein wenig aus wie eine Textverarbeitung. Er hat alle gängigen Einrichtungen, wie eine Menüleiste und/oder anklickbare Symbole für Funktionen wie Speichern und Laden, Suchen und Ersetzen, Rechtschreibprüfung und so weiter. Aber ihm fehlt die Fähigkeit,

den Text zu setzen. Der Text, den Sie tippen, erscheint klar lesbar am Bildschirm, aber ohne Anspruch darauf, den Text in seiner endgültigen Form darzustellen.

Wenn Sie Ihren Text speichern, wird er als *reiner Text* gespeichert, was in der westlichen Welt gewöhnlich „ASCII“ genannt wird (American Standard Character-Set for Information Interchange). Der ASCII-Zeichensatz besteht aus 128 Zeichen (und wird deshalb manchmal als „7-Bit“ Zeichensatz bezeichnet, da er sieben Bit pro Buchstabe benötigt: $2^7 = 128$). Er enthält die Ziffern 0 bis 9, die lateinischen Buchstaben als Groß- und Kleinbuchstaben, die gängigsten Satzzeichen und einige Spezialzeichen.⁴ ASCII ist der kleinste gemeinsame Nenner für die schriftliche Kommunikation auf elektronischer Basis. Eine Nachricht in ASCII wird von jedem Computer auf der Welt „verstanden“. Wenn Sie eine solche Nachricht verschicken, können Sie sicher sein, dass der Empfänger genau das sieht, was Sie getippt haben.

Im Gegensatz dazu enthält eine Datei, die Sie mit einem Textverarbeitungssystem speichern, diverse „Steuerzeichen“, die außerhalb des ASCII-Standards liegen. Diese Zeichen geben an, welche Formatierung Sie angegeben haben (fett oder kursiv) sowie diverse Arten von internen „Verwaltungsdaten“, die die Textverarbeitung benötigt. Sie sind nicht allgemein „verständlich“. Um aus ihnen etwas sinnvolles zu erhalten, benötigen Sie eine Kopie des Textverarbeitungssystems, mit dem der Text erstellt wurde (oder ein geeignetes Umwandlungsprogramm). Wenn Sie die Datei eines Textverarbeitungsprogramms mit einem Texteditor öffnen, werden Sie (neben dem Text oder Teilen von ihm) viel „lustiges Zeug“ entdecken: das ist der binäre Formatierungscode.

Nachdem ein Texteditor keinerlei binären Formatierungscode einfügt, müssen Sie Text-eigenschaften (wie kursiv) dadurch markieren, dass Sie *Markup* verwenden. Das bedeutet, dass Sie eine Anmerkung schreiben (und dabei nichts als ASCII verwenden), die dem *Typographen* sagt, dass der entsprechende Text kursiv gesetzt werden soll. Zum Beispiel würden Sie für das \LaTeX -Satzprogramm (mehr darüber weiter unten) schreiben `\textit{ein wenig kursiver Text}`. Tatsächlich ist es so, dass Sie das nicht einmal selber schreiben müssen, wenn Sie einen Texteditor verwenden, der dafür ausgelegt ist, mit \LaTeX zusammenzuarbeiten. Sie tippen einfach ein Tastenkürzel, wählen etwas aus einem Menü aus oder klicken auf ein Symbol und die passende Notation wird für Sie eingefügt. Die Vorgehensweise, mit der Sie ein ASCII-Dokument erstellen, das Sie dann \LaTeX übergeben können, unterscheidet sich nicht sehr von der, wie Sie mit einer modernen Textverarbeitung arbeiten.

2.5 Die Vorzüge von ASCII

Der Ansatz, Ihren Text mit einem Texteditor in reinem ASCII zu schreiben und ihn dann mit einem gesonderten Programm zu setzen, hat mehrere einleuchtende Vorteile:

1. Übertragbarkeit. Wie schon oben dargelegt, kann *jeder* auf jedem Computersystem Ihren „markierten“ Text lesen, sogar wenn derjenige nicht die nötigen Hilfsmittel

⁴ Anm. d. Ü.: Der ASCII-Standard enthält keine Umlaute. Für Umlaute benötigt man z. B. den ANSI- bzw. Latin1-Standard (dies sind 8-Bit Zeichensätze, die den ASCII-Zeichensatz als Untermenge enthalten).

hat, um den Text in der fertigen (gesetzten) Fassung anzusehen. Im Gegensatz dazu kann eine Datei der Textverarbeitung „Schick 9.0“ vollkommen unverständlich für einen Empfänger sein, der nicht die gleiche Sorte und Version der Textverarbeitung hat wie Sie – außer er oder sie kennt sich mit Computern gut aus und ist in der Lage, den wirklichen Text von dem binären „Müll“ zu trennen. Diese Versionsunverträglichkeit kann für Sie mit der Zeit genauso zu einem Problem werden wie für einen Gesprächspartner. Sie könnten sehr gut auf Schwierigkeiten stoßen, wenn Sie eine „Schick 8.0-Datei“ mit „Schick 9.0“ lesen wollen und umgekehrt. Aber Sie werden niemals irgendwelche Probleme haben, gutes altes ASCII zu lesen.

2. Hohe Dichte. Eine ASCII-Datei beschreibt Ihre *Ideen* und nicht viel „Textverarbeitungsinterna“. Bei kleineren Texten kann die Datei eines Textverarbeitungssystems bis zu zehn Mal größer sein als eine entsprechende ASCII-Datei, die nur die relevanten Informationen enthält.
3. Sicherheit. Der „Texteditor und Typograph“-Ansatz bietet Ihnen die größtmögliche Sicherheit, dass Sie niemals irgendwelche Probleme mit beschädigten Dokumenten haben werden (solange nicht Unannehmlichkeiten, wie einen Festplattencrash oder ähnliches auftreten). Der Quelltext wird immer da sein, selbst wenn das Satzprogramm aus irgendeinem Grunde abbricht. Sie hatten Sie bisher großes Glück, wenn Sie regelmäßig mit Textverarbeitungsprogrammen arbeiten und noch *nie* Probleme mit kaputten Dateien hatten!

Weiterführende Literatur hierzu: Sam Steingolds Seite „No Proprietary Binary Data Formats“ (<http://www.podval.org/~sds/data.html>).

2.6 Das Satzprogramm

Hier schulde ich Ihnen noch ein paar Details zu dem Satzprogramm, das zur von mir vertretenen Strategie passt. Ich werde dabei nicht in technische Details eintauchen, aber doch versuchen, genug zu sagen, um Ihnen zu vermitteln, wovon ich rede.

Das meiner Strategie zugrunde liegende Textsatzprogramm nennt sich $\text{T}_{\text{E}}\text{X}^5$ und wurde von Donald Knuth von der Stanford University geschrieben. $\text{T}_{\text{E}}\text{X}$ ist frei erhältlich (mittels Download von vielen Internetseiten), in Formaten für so gut wie jedes Computersystem (wenn Sie wollen, können Sie für einen sehr moderaten Preis eine CD-ROM bestellen, die sämtliche für $\text{T}_{\text{E}}\text{X}$ erforderliche Dateien enthält). Knuth begann mit $\text{T}_{\text{E}}\text{X}$ im Jahr 1977; 1990 gab er dann bekannt, dass er nicht mehr länger daran arbeiten wird – jedoch nicht aus mangelndem Interesse, sondern, weil zu dieser Zeit das Programm im Wesentlichen perfekt war. Es ist so fehlerlos, wie ein Computerprogramm nur sein kann und es leistet großartige Arbeit darin, Texte jeglicher Couleur zu setzen, vom einfachen Fließtext bis zur höheren Mathematik.

⁵ Anm. d. Ü.: Gesprochen wird $\text{T}_{\text{E}}\text{X}$ übrigens ähnlich wie Blech. Das große X am Schluss ist ein griechisches Chi.

Ich sprach oben von \LaTeX . Während \TeX das zugrunde liegende Textsatzprogramm ist, ist \LaTeX ein großer Satz an Makros⁶, die ursprünglich von Leslie Lamport nach 1980 entwickelt wurden und die jetzt von einer internationalen Expertengruppe gepflegt werden. Diese Makros erleichtern dem gewöhnlichen Anwender das Leben um Einiges.

\LaTeX ist noch immer in Entwicklung, da stets neue Fähigkeiten und Zusatzpakete dem zugrunde liegenden Satzprogramm hinzugefügt werden. Verschiedene „Zusätze“ für \TeX sind ebenfalls in Entwicklung, zum Beispiel ein System, das es einem erlaubt, PDF-Dateien (Adobes „Portable Document Format“) direkt aus der ASCII-Quelldatei zu erzeugen. Ich sage „in Entwicklung“, aber damit meine ich, dass diese Programme stetig verbessert werden. Sie sind bereits sehr stabil und reich an Optionen.

Wie oben erwähnt, vermitteln Sie \LaTeX die gewünschte Struktur und Formatierung Ihres Dokuments in Form einer Reihe von Anmerkungen. Es gibt viele Bücher (und Internet-basierte Anleitungen), die diese Anmerkungen im Detail erklären und ich werde hier nicht näher darauf eingehen. Die häufigsten Anmerkungen sind einfach und leicht zu merken, darüber hinaus erleichtern die \LaTeX -freundlichen Texteditoren (von denen es viele gibt) die Sache noch zusätzlich.

Ein sehr verlockendes Merkmal von \LaTeX ist die Fähigkeit, den Schriftsatz Ihres Textes mit wenigen Befehlen textübergreifend und einheitlich zu ändern. Die grundlegende Erscheinung wird festgelegt durch

1. die „Dokumentklasse“, die Sie wählen (z.B. Bericht, Brief, Artikel, Buch).
2. die „Pakete“ oder Stildateien, die Sie laden wollen.

Sie können beispielsweise vollständig die Schriftfamilie ändern (durchgehend in Text, Abschnittsüberschriften, Fußnoten und allem) und/oder die Größe der benutzten Schrift nur durch das Verändern eines oder zweier Parameter im „Vorspann“ Ihrer ASCII-Quelldatei variieren. Auf ähnliche Art und Weise können Sie alles zweispaltig setzen oder es von Hochformat in Querformat umwandeln. Es ist möglich, etwas Vergleichbares mit einem Textverarbeitungsprogramm zu erhalten, aber damit sind derartige Änderungen im Allgemeinen viel unangenehmer und es ist viel wahrscheinlicher, dass Sie etwas durcheinander bringen und unbeabsichtigt Widersprüchlichkeiten in der Formatierung auftreten.

Sie können das Setzen mit \LaTeX so kompliziert machen, wie Sie wollen. Sie können zum Beispiel eine „Hände-weg“-Einstellung wählen: Geben Sie nur eine Dokumentklasse an und überlassen Sie den Rest den Standardmakros. Im Allgemeinen erzeugt dies gute Ergebnisse, bei denen der Textsatz von viel höherer Qualität ist als bei irgendeinem Textverarbeitungsprogramm. (Natürlich werden Dinge wie die Nummerierung von Kapiteln, Abschnitten und Fußnoten, Verweise und so weiter automatisch erledigt.) Oder Sie können eine mehr „eingreifende“ Haltung einnehmen und diverse Pakete laden (oder sogar Ihr eigenes schreiben), um das unterschiedliche Erscheinungsbild des Schriftsatzes zu beeinflussen. Wenn Sie diese Einstellung bevorzugen, können Sie wirklich schöne und einzigartige Ausgabedateien erzeugen.

⁶ Anm. d. Ü.: Ein Makro (macro code) ist aus mehreren kleinen Befehlen aufgebaut.

2.7 Es zusammensetzen

Lassen Sie mich Ihnen einen kurzen Überblick darüber geben, wie dies alles funktioniert. Wenn Sie ein gut eingerichtetes T_EX haben, geht es wie folgt: Sie tippen Ihren Text in einen T_EX-fähigen Editor. Dabei können Sie die nötigen Befehle direkt eingeben oder sie vom Editor über Menüs oder Knöpfe einfügen lassen. Wenn Sie den Punkt erreichen, an dem Sie einen Blick auf die gesetzte Version werfen möchten, wählen Sie im Menü den entsprechenden Punkt oder drücken einen Knopf im Editor, um den Satzvorgang zu starten. Ein anderer Menüpunkt oder Knopf öffnet eine Voransicht, in der Sie den Text so sehen können, wie er gedruckt aussehen wird. Und normalerweise ist dies wirkliches „WYSIWYG“ – das Vorschau-Programm zeigt eine sehr genaue Darstellung der zu druckenden Datei. Sie können vergrößern und verkleinern, herumbliättern und so weiter. Mit einem anderen Menüpunkt schicken Sie die Datei an den Drucker oder fahren mit dem Editieren fort.

An einem späteren Punkt im Verlauf möchten Sie die aktualisierte Datei ansehen. Drücken Sie wieder auf den Setzer-Knopf. Dieses Mal müssen Sie das Programm für die Vorschau nicht erneut aufrufen: Wenn Sie es im Hintergrund haben laufen lassen, wird es jetzt automatisch die aktualisierte und gesetzte Version anzeigen. Wenn Sie die Datei zu Ende geschrieben haben, können Sie die gesetzte Version der Datei löschen, um Plattenplatz zu sparen. Sie müssen nur die ASCII-Quelldatei weiterhin speichern; die gesetzte Version kann auf einfache Art und Weise wieder erstellt werden, wann immer Sie sie brauchen.

3 Digitale Verbreitung

Der vorangehende Abschnitt hat sich hauptsächlich mit der Ausgabe gut aussehender Erzeugnisse auf den Drucker beschäftigt. Einige andere Überlegungen ergeben sich, wenn Sie ein Schriftstück für die digitale Verbreitung vorbereiten (E-Mail, Internetseiten und so weiter).

Nehmen wir zuerst E-Mail. Wenn Leute nur eine kurze, schnelle Nachricht schicken wollen, tippen sie normalerweise die Nachricht direkt in ein E-Mail-Programm, sei es ein „herkömmliches“ textbasiertes Programm wie Pine oder ein GUI-Programm (GUI: Graphische Benutzeroberfläche) wie Netscape oder Eudora. In jenem Fall wird die Nachricht wahrscheinlich in Form von ASCII verschickt (oder vielleicht in HTML, d.h. Hypertext Markup Language, der Sprache der Webseiten, die selber größtenteils in ASCII verfasst ist). Aber was, wenn Sie ein längeres Stück Text verschicken wollen, das Sie schon unabhängig von einem E-Mail-Programm verfasst haben?

Zu diesem Zweck wird es immer üblicher, ein Dokument als Datei eines Textverarbeitungsprogrammes anzuhängen. Wie lautet die alternative Handlungsabfolge in diesem Fall?

Wir müssen zwischen zwei Fällen unterscheiden: Ist besagter Text relativ kurz und unkompliziert (eine Mitteilung, ein Brief, Notizen aus einem Treffen, eine Aufzählung der Tagesordnung, ein Zeitplan für einen Besuch) oder ist er komplizierter (eine wissenschaftliche Abhandlung – vielleicht mit viel Mathematik, ein bebildeter Bericht, ein

Buchmanuskript)? Die „ASCII plus Textsatzprogramm“-Herangehensweise führt zu verschiedenen Vorschlägen in diesen zwei Fällen.

3.1 Einfache Schriftstücke

Bei einfachen Schriftstücken müssen wir uns fragen: Brauchen wir die Formatierung, die Schriftinformationen und all das wirklich? Ist es nicht effizienter, mehr im Sinne von wirkungsvoller und wirtschaftlicher Verständigung, einfach den reinen ASCII Text zu verschicken, mit der wenigen Formatierung, die ASCII erlaubt? Dies spart einerseits Bandbreite (denken Sie daran, dass Dateien von Textverarbeitungsprogrammen *viel* größer sein können als ASCII-Dateien, die den selben Text beinhalten) und stellt außerdem sicher, dass niemand aus der Kommunikation ausgeschlossen wird, weil er zufällig nicht „Schick 9.0“ auf seinem Rechner laufen hat. Sie können eine ASCII-Datei, die mit einem Texteditor erzeugt wurde, auf dieselbe Art und Weise anhängen, mit der Sie die Datei eines Textverarbeitungsprogramms anhängen, oder Sie fügen sie einfach in den Textteil der E-Mail ein (denn es handelt sich um nichts anderes als reinen Text). Da die \TeX -Quelldateien nichts anderes sind als ASCII – und wenn wir von einem einfachen Dokument reden, beinhaltet es nicht allzu viele Befehle, und diese sind ziemlich selbsterklärend – können sie genauso behandelt werden.

3.2 Kompliziertere Dokumente

Längere und kompliziertere Dokumente sind wohl in gesetzter Form leichter zu lesen. Mathematik kann schwer in ASCII konvertiert werden und komplizierte Diagramme und Bilder natürlich erst recht nicht. Also wie steht es mit \TeX in diesem Zusammenhang? Ich habe dargelegt, dass Dateien von Textverarbeitungsprogrammen problematisch sein können, weil Ihr Gesprächspartner möglicherweise nicht „Schick 9.0“ hat so wie Sie. Aber gilt das andersherum nicht genauso? Selbst wenn Ihre Neugierde ausreichend geweckt wurde, um \TeX auszuprobieren – wie viele Ihrer Gesprächspartner haben \TeX installiert? Dies ist eine vernünftige Frage, aber sie ist beantwortbar. Wenn Sie möchten, dass Ihr Gesprächspartner in der Lage ist, eine gesetzte Version Ihrer Datei anzusehen, bei ihm aber kein \TeX installiert ist, haben Sie diese Möglichkeiten:

1. Konvertieren Sie die \TeX -Quelldatei nach HTML. Es gibt gute Konvertierungsprogramme zu diesem Zweck. (HTML und \TeX haben tatsächlich eine starke Familienähnlichkeit, sie verwenden nämlich beide logisches Markup, so dass beide Formate problemlos ineinander umgewandelt werden können⁷) Dann kann Ihr Gesprächspartner Ihren Text mit Hilfe eines Internetbrowsers lesen.
2. Hat Ihr Gesprächspartner Zugang zu einem postscriptfähigen Drucker? Im akademischen Umfeld oder in der Geschäftswelt ist dies sehr wahrscheinlich. In diesem

⁷ Die Binärcodierung, die von Textverarbeitungsprogrammen verwendet wird, ist ein ganz anderes Ding, weshalb Konvertierungen zwischen den Formaten, die \TeX und Textverarbeitungsprogramme verwenden, nicht einfach sind. Da \TeX ein überlegenes Textsatzsystem ist, ist es darüber hinaus prinzipiell unmöglich, ein \TeX -Dokument, sagen wir nach Word zu konvertieren, ohne Informationen zu verlieren.

Fall könnten Sie eine gesetzte Version Ihres Dokuments in Form einer PostScript-Datei, die er einfach an den Drucker schicken kann, verschicken. Und/oder er kann es auf dem Bildschirm ansehen, wenn er das Programm „ghostview“ installiert (das es im Internet kostenlos zum Herunterladen gibt).

3. Hat Ihr Gesprächspartner den „Acrobat-Reader“ für PDF-Dateien installiert? (Auch diesen kann man umsonst herunterladen.) Wenn ja, können Sie eine PDF-Variante Ihres gesetzten Dokuments verschicken.

In der Diskussion um die Möglichkeiten des Verschickens von Text mittels E-Mail haben wir das Thema, Text für Webseiten aufzubereiten, bereits gestreift. Sie haben die Möglichkeit, HTML direkt zu schreiben. Wenn Sie das nicht tun möchten, können Sie indirekt HTML erzeugen, indem Sie einen passenden GUI-Editor benutzen, Netscape Communicator beispielsweise. Sicherlich können Sie auch HTML erzeugen, indem Sie MS Word benutzen (nebenbei gesagt schreckliches HTML, voller fremdartiger Befehle („Tags“), die es schwierig machen, es mit einer anderen Anwendung zu bearbeiten). Wenn Sie $\text{T}_{\text{E}}\text{X}$ benutzen, ist es einfach, Ihre Dokumente nach (sauberem, dem Standard entsprechenden) HTML zu konvertieren.

4 Voraussetzung

Ich habe versucht, eine Lanze zu brechen für die Alternative „ASCII plus Textsatzprogramm“ gegenüber den Textverarbeitungsprogrammen. Ich gebe aber zu, dass es *einige wenige* Arten von Dokumenten gibt, für die ein WYSIWYG-Textverarbeitungsprogramm in der Tat das Werkzeug der Wahl ist. Ich denke an kurze, schnelle Dokumente, die ein hohes Verhältnis von „Formatierungsangaben“ zu Text haben: Flyer, Poster, Partyeinladungen usw. Sie könnten dies in $\text{T}_{\text{E}}\text{X}$ machen, aber es wäre nicht effizient. Die Standard- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Dokumentklassen (Bericht, Artikel usw.) würden von geringem Nutzen sein. Und während $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ sehr geschickt ist im automatischen Umgang mit einer Palette von Schriften, ist es nicht eingestellt auf so etwas wie das „Mischen und Anpassen“ von verspielten Schriften, die man bei solchen Gelegenheiten verwendet. Logische Struktur ist nicht wirklich ein Anliegen: Sie sind interessiert an „purem Formatieren“. Sie wollen beispielsweise wissen: „Wenn ich *diese* Zeile in einer 36-Punkt-Schrift setze, verschiebt das meine letzte Zeile auf die nächste Seite, was ich nicht will?“ WYSIWYG ist dein Mann.

Wenn ein Großteil Ihrer Textverarbeitung von dieser Art ist, haben Sie wahrscheinlich vor langer Zeit mit dem Lesen aufgehört. Wenn ein Großteil Ihrer Textbearbeitung die Erstellung relativ formaler Dokumente umfasst, gilt meine Ausnahme nicht für Sie.

5 Schimpf, schimpf

Es mag Ihnen nicht entgangen sein, dass ich mich ein bisschen über dieses Thema erregte. Ja, das tue ich. Der Punkt ist, dass es nicht nur um eine wissenschaftliche Debatte zwischen alternativen Arten der Textbearbeitung geht. Es ist eine Waage, auf der sich

die Macht und der Reichtum der großen Softwareanbieter auf einer einzigen Waagschale befinden. Ganz konkret haben wir eine Situation vor uns, in der MS Word dabei ist, fast weltweit, *der* Standard für die Textverarbeitung am Computer zu werden. Aber Word ist ein Standard, der wenig als Empfehlung hat außer der Tatsache, dass es ein Standard *ist* (oder danach strebt, einer zu sein).

Es ist ein bisschen wie QWERTZ. Kennen Sie die Geschichte? Warum steht bei der Standard-Anordnung der Tasten auf der Tastatur von Schreibmaschinen (und deshalb auch bei Computertastaturen) QWERTZUIOP in der ersten Zeile? Das war nicht die ursprüngliche Anordnung der Schreibmaschinentasten. Es wurde zu einem einzigen Zweck entworfen, nämlich um *Schreibkräfte zu verlangsamen*. Das Problem war, dass die Geschicklichkeit der frühen Schreibkräfte schnell die Möglichkeiten der frühen mechanischen Schreibmaschinen überstieg: Ein schneller Schreiber konnte die Tasten festklemmen, wenn er sie schneller drückte, als sie zurückfallen konnten, nachdem sie das Farbband getroffen hatten. QWERTZ verteilte die Tasten, so dass sie nicht so schnell tippen konnten. Dies ist klarerweise eine verrückte Anordnung für die Tasten auf einer elektrischen Tastatur, aber es ist zu spät, um es rückgängig zu machen. QWERTZ ist Standard, und alle Versuche, die Tastatur vernünftiger zu gestalten, haben angesichts dieser Realität versagt.

Analog dazu behaupte ich, dass MS Word kein *Recht* hat, ein Standard für die Textverarbeitung zu sein, da es klar ist, dass es (für die meisten Zwecke) weniger effizienter ist als bereits verfügbare Alternativen. Ich hoffe, dass es in diesem Fall noch nicht zu spät ist, dass es immer noch eine Möglichkeit gibt, Nein zu Word zu sagen. Tatsächlich ist Word in gewisser Hinsicht schlimmer als QWERTZ: Es ist kein wirklicher Standard sondern eher eine Vereinbarung. Der Microsoft-„Standard“ für das interne Format der gespeicherten Datei ist etwas, das mit der Laune der Microsoft Corporation steht und fällt. Das MS-Word-Quasi-Monopol kommt huckepack mit dem MS-Windows-Quasi-Monopol (eine Angelegenheit, auf die ich hier nicht näher eingehen werde). Und solange sie nicht von konkurrierenden Firmen stark unter Druck gesetzt wird, hat Microsoft kein besonderes Interesse daran, irgendeine Form von langanhaltendem Standard für das interne Format der gespeicherten Datei zu etablieren. Im Gegenteil: Sie hat ein starkes Interesse daran, Sie zu zwingen, Ihr Word in regelmäßigen Abständen „upzupgraden“. Oh Gott, Word N.0 will das Dokument nicht öffnen, das Ihr Kollege Ihnen eben geschickt hat, geschrieben mit N+1? Nun ja, dann sollten Sie besser „upgraden“ nicht war? Selbst wenn es keine Funktionen in N+1 gibt, die es nicht auch schon in N gab und die von irgendeinem wirklichen Nutzen für Sie sind.⁸

⁸ Denn was aus meiner Sicht noch schlimmer ist – als jemand, der viele Jahre lang Word benutzt hat, bevor er zu T_EX wechselte und der ein lebhaftes Interesse an Typographie hat – ist die Tatsache, dass in MS Word für Windows keine lohnenden Funktionen eingebaut wurden seit der Version 2.0 von etwa 1990.

6 Verweise

Wenn Sie mir so weit gefolgt sind, haben Sie vielleicht Interesse an genaueren Informationen über gute Texteditoren, das $\text{T}_{\text{E}}\text{X}$ Satzsystem und so weiter.

Der beste Ort, um nach Informationen über $\text{T}_{\text{E}}\text{X}$ und Konsorten zu suchen, ist wahrscheinlich die TUG homepage (<http://www.tug.org>)⁹ (TUG ist die $\text{T}_{\text{E}}\text{X}$ Users Group). Diese wird Sie mit allen Links versorgen, die Sie brauchen könnten; einer der Hauptlinks ist der zu den Seiten des Comprehensive $\text{T}_{\text{E}}\text{X}$ Archive Network¹⁰, CTAN (<http://www.tug.org/ctan.html>), von denen Sie vollständige $\text{T}_{\text{E}}\text{X}$ -Distributionen für beinahe alle Computersysteme herunterladen können. Solche Distributionen beinhalten das eigentliche Textsatzsystem, eine große Sammlung von Makros, ein Vorschau-Programm und Software, um druckbare Dateien zu erzeugen.

$\text{T}_{\text{E}}\text{X}$ -Distributionen (auf jeden Fall die kostenlosen) beinhalten meist nicht den Texteditor, den Sie ebenfalls brauchen (es sei denn, Sie haben bereits einen, den Sie mögen). Es gibt viele Möglichkeiten, aber mein persönlicher Liebling für die Arbeit mit $\text{T}_{\text{E}}\text{X}$ -Dateien ist Emacs (<http://www.emacs.org>) zusammen mit dem AU $\text{T}_{\text{E}}\text{X}$ -Paket (<http://sunsite.auc.dk/auctex/>). Das letztgenannte macht Emacs sehr benutzerfreundlich: Es hebt die $\text{T}_{\text{E}}\text{X}$ -Syntax hervor, so dass Sie alle Fehler in Ihrem Markup auf den ersten Blick sehen können und es bietet eine Menge auf $\text{T}_{\text{E}}\text{X}$ bezogene Befehle in übersichtlichen Menüs.

Für den Fall, dass Sie interessiert sind, gibt es auf meiner Homepage eine Abbildung (<http://ricardo.ecn.wfu.edu/~cottrell/emacs-screen.jpg>), die zeigt, wie $\text{T}_{\text{E}}\text{X}$ und Emacs zusammen verwendet werden (jpeg, 145.345 Bytes).

⁹ Anm. d. Ü.: Das deutschsprachige Pendant ist die „Deutschsprachige Anwendervereinigung $\text{T}_{\text{E}}\text{X}$ e.V.“ (<http://www.dante.de>)

¹⁰ Anm. d. Ü.: „Vollständiges $\text{T}_{\text{E}}\text{X}$ Archiv-Netzwerk“. CTAN gilt aber als Eigenname und wird deshalb nicht übersetzt.