

#SATZ und #FORMATIERE

Lieber Herr Trauth,

wenn ich mich in die von Ihnen entfachte Diskussion einmische, so tue ich es eigentlich unter den denkbar ungünstigsten Voraussetzungen, denn meine TUSTEP-Sozialisierung begann um 1974, also im Lochkartenzeitalter, mit dem Satzprogramm. Codes und Aufruf sind mir also seit langem vertraut, weit besser als die Formatiere-Codes. Womit ich keineswegs sagen will, daß ich alles, was man überhaupt braucht, präsent hätte, aber die wichtigsten Codes sind mir in Fleisch und Blut übergegangen, und Formatiere habe ich zum schnellen Ausdrucken von Satzdaten benutzt, als es *SASIM und *PSAUS noch nicht gab. Mein großes Interesse an TUSTEP bewegt mich, trotzdem in die Diskussion einzugreifen.

Diese Diskussion begann ja, trotz Ihrer krankheitsbedingten Abwesenheit von Ihnen ausgelöst, schon Anfang Januar in Blaubeuren. Die Herren, die TUSTEP-Kurse geben, beklagten vor allem den Umstand, daß sie ihren Kursteilnehmern keine hinreichend einfache Möglichkeit anbieten können, Seminar- oder Magisterarbeiten mit TUSTEP zu erstellen, weil das Satzprogramm zu kompliziert sei, Formatiere zwar bequem, heute aber nicht einmal mehr für Seminararbeiten zu gebrauchen sei, denn es bietet keine Proportionalschriften an.

Eine – in Blaubeuren auch von mir vehement vertretene – Meinung war, man solle ein Kommandomakro programmieren, das das Satzprogramm mit reduzierten Möglichkeiten auf einfache Weise aufzurufen gestattet.

Heute denke ich etwas anders. Mit einem solchen Kommandomakro könnte man zwar erreichen, daß der Benutzer *relativ* einfach eine formal schlichte Arbeit setzen, in ghostview ansehen und drucken kann. Allerdings müßte er doch einen einfachen Codesatz – für Überschriften, Absatz, Fußnoten, Auszeichnungen u. a. – lernen und benutzen, seien es nun die originären Satzcodes oder eine vordefinierte Menge von tags. Die rechte Einsicht in die Funktionsweise des Satzprogramms einschl. PSAUS und FUNO wäre mit einem solchen Verfahren, das sich ja der black box nähert, eben nicht zu gewinnen.

Die Diskussion der letzten beiden Wochen hat mich überzeugt, daß eine Weiterentwicklung des Formatiere in der von Ihnen angedeuteten Richtung – Unterstützung der Standard-PostScriptschriften, Grundschrift, Einschaltungen, 3 Überschriftstufen, mehrere Textsorten (Literaturteil, Titelei, lebende Kolumnentitel) eine sehr nützliche Sache wäre, die jedenfalls TUSTEP-Neulinge davon überzeugen könnte, ihre Seminararbeiten nicht im Word schreiben zu müssen. Andere Diskutierer verlangen zusätzlich mehrspaltigen Satz, Tabellen, PDF-Ausgabe; Sie selbst schlagen noch die Unterstützung

nicht-Postscriptfähiger Drucker vor (dazu unten). Zusätzlich würde ich dafür plädieren, inhaltliche Auszeichnungen für Überschriften, Einschaltungen u. dgl. wie im Satzprogramm vorzusehen, die man sich jetzt nur mit Makros zusammenstellen kann (Handbuch S. 592). Offen bleibt die Frage, ob Schriftgrade für die Textsorten – wie im Satzprogramm – angebar oder (für Seminararbeiten ausreichend?) fest vorgegeben sein sollten, was den Abstand zum Satzprogramm vergrößern würde, dem Sie ja ausdrücklich keinen Konkurrenten heranzüchten wollen. Der Programmieraufwand dürfte in jedem Fall erheblich sein; beurteilen kann ihn außer Herrn Ott sicher am besten Herr Schälkle, wie Herr Küster richtig bemerkt.

Oft wird in der laufenden Diskussion die Einfachheit der Codes des Formatiere gegenüber der angeblich umständlichen des Satzprogramms gerühmt. Aber man täusche sich nicht: eine Formatieren mit mehr Möglichkeiten wird auch mehr Parameter und mehr Steuerzeichen haben.

Die Diskussion geht ja in mehrere Richtungen: es wird ein Tutorial zu anderen Programmen und hier insbesondere zum Satzprogramm vorgeschlagen. (Im Internet findet man leicht ein Kochbuch zu LaTeX; so ähnlich könnte auch ein Satz-Kochbuch aussehen, wenn auch die Küchenanalogien nicht ganz so weit getrieben werden müßten.) Also eine einfache Einführung. Einfach müßte bedeuten: eine Auswahl mit Andeutung weitergehender Möglichkeiten, Erläuterung (nicht Vermeidung) satztechnischer Begriffe (die man übrigens teilweise sogar im Brockhaus erklärt findet) in einem Glossar, Bereitstellen einiger Parametersätze für typische Aufgaben. [Wer die Beschreibung der Satzcodes für kryptisch erklärt, möge Beispiele nennen.] Die Frage ist nur, wie meist, »wer macht's?«

Auch die Forderung nach einer völligen Neuprogrammierung des Satzprogramm wird erhoben. Hier bin ich skeptisch. Das Programm wird ja ständig weiterentwickelt, nach dem Erscheinen des Handbuchs z. B. neue Schriften einschl. Fraktur, PDF-Marks, Verknüpfung mit WWW-Dateien, zweispaltiger Satz von Fußnoten unter einspaltigem Text und vieles andere. Natürlich würde auch ich mich freuen, wenn überlange Apparate automatisch auf die Folgeseite übertragen würden, Zeilennummerierung abschnitts- und nicht nur seitenbezogen angegeben werden und dann natürlich in Apparaten benutzt werden könnte [aber hier wird's kompliziert: bei mehrerer kurzen Texten auf einer Seite müßte eine Textnummer oder -überschrift hinzukommen!] oder – darüber bin ich schon gestolpert – wenn der Bezugsende-Code nicht spätestens auf der übernächsten Seite stehen müßte.

Es mag ja sein, daß solche Wünsche bei einer völligen Neuprogrammierung – wenn sie denn finanziert würde – vor vornherein berücksichtigt würden. Es mag auch sein, daß eine neu konzipierte Codierung einleuchtender wäre als die historisch gewachsene. Sicher scheint mir, daß ein neues Satzprogramm sich anfangs und über lange Zeit durch

eine Vielzahl von Fehlern »auszeichnen« würde, was bei einem Programm mit dieser Fülle von Möglichkeiten kaum zu vermeiden wäre und eine professionelle Nutzung für den Anfang jedenfalls ausschließen würde. Wahrscheinlich müßte das jetzige Satzprogramm mehrere Jahre lang parallel angeboten werden, bis das neue sich stabilisiert hätte. Eine voraussehbare, aber kaum abwendbare Schwierigkeit kommt hinzu: das Satzprogramm könnte nur mit eigens eingestelltes Personal neu programmiert werden – Projektpersonal, das Abschluß des Projekts verschwunden wäre. Wer könnte dann das Programm pflegen? Also plädiere ich stattdessen, wie oben gesagt, für eine Einführung in Standardfälle und im übrigen für organische Weiterentwicklung wie bisher.

Ungern würde ich auf die Postscript-Ausgabe verzichten. Man kann es im ghostview, der viel besser steuerbar ist als der Adobe Reader, bekanntlich sehr bequem ansehen und auf beliebigen Druckern ausdrucken, was die Forderung nach Unterstützung nicht-postscriptfähiger Drucker relativiert. Man kann in Postscript-Code auch manuell eingreifen, was bei PDF dem Vernehmen nach praktisch unmöglich sein soll. Programme, die aus Postscript PDF erzeugen, gibt es genügend: neben dem nicht ganz billigen Distiller gibt es kostenlose Programme (z. B. WordToPDF, das nicht nur Word-Dateien verarbeitet, sondern eben auch Postscript-Dateien), und neue Versionen von ghostview können es auch. Nebenbei: ersetzt ghostview nicht auch das immer wieder geforderte Satz-preview? Das Satzprogramm + *psaus läuft doch auch für hunderte von Seiten in Sekunden ab (wenn man das Zweitprotokoll einschaltet), und man kann danach ghostview mit \$\$ browse in die Kommandofolge einbeziehen. Also WY-SIWYG mit einer Minute Verzögerung.

Ich plädiere also für

- eine knappe Einführung in das Satzprogramm, vielleicht im Internet (Satz-Kochbuch)
- eine Aufwertung des Formatiere
- Weiterentwicklung, aber bis auf weiteres nicht Neuprogrammierung des Satzprogramms

In diesem Sinn grüßt Sie herzlich

Ihr Friedrich Seck