

Additional references and notes for \TeX : A branch in desktop publishing evolution, Part 2

In the following, the number at the beginning of a note is a page number; words or a topic from that printed text page then appear indicating the position on the page to which the note or reference relates; then comes the note or reference itself.

- 30 “ **\LaTeX had come on the scene in 1983**” One brief overview of \LaTeX is Open Source Documentation Software: An Overview, Shubhashree Savant and Sonal Sarnaik, International Conference on Advances in Information Technology and Management ICAIM, 2016, tinyurl.com/latexbriefoverview
- 30 “**surely contributed to their popularity**” In *TUGboat* \LaTeX creator Leslie Lamport said, “I don’t think \TeX and \LaTeX would have become popular had they not been free. Indeed, I think most users would have been happier with Scribe. Had Scribe been free and had it continued to be supported, I suspect it would have won out over \TeX . On the other hand, I think it would have been supplanted more quickly by Word than \TeX has been.”; *TUGboat*, vol. 22, no. 1/2, 2001, pp. 20–22, tug.org/TUGboat/tb22-1-2/tb70lamp.pdf
- 30 “**local user groups continued to expand**” Eric Frambach, \TeX user groups worldwide—what’s cooking?, *MAPS*, Autumn 2003, pp. 6–9, ntg.nl/maps/29/03.pdf
- 30 “**local user groups continued to expand**” Christina A. L. Thiele, The Future of \TeX and TUG, *TUGboat*, vol. 14, no. 3, 1993, pp. 162–166, tug.org/TUGboat/tb14-3/tb40thiele-future.pdf
- 31 “**DVI output to PostScript**” Tomas Rokicki’s *dvips* DVI post-processor, tug.org/dvips.
- 31 **$j\TeX$ and $p\TeX$** ctan.org/pkg/ptex, <https://ctan.org/pkg/uptex>; also Haruhiko Okumura, $p\TeX$ and Japanese Typesetting, *The Asian Journal of \TeX* , vol. 2, no. 1, April 2008, pp. 43–51, ajt.ktug.org/2008/0201okumura.pdf
- 31 **Con \TeX t format** pragma-ade.com;contextgarden.net
- 31 “**specification of PostScript programs**” While PostScript is often thought of in terms of text fonts, it can specify any sort of drawing, and indeed, is a general-purpose programming language.
- 32 “**in the \TeX Live collection**” Con \TeX t is also widely distributed independent of the \TeX Live collection.
- 32 “**David Grier has said**” Page 11 in the transcript of the Computer History Museum “PC Software Workshop: Marketing and Sales,” recorded May 6, 2004, CHM reference number X4621.2008.
- 33 “ **\TeX ’s initial development**” Donald E. Knuth, *Digital Typography*, CSLI Publications, Stanford, CA, 1999.
- 33 “**Ikarus developed by Peter Karow**” Peter Karow, *Digital Formats for Typefaces*, second edition, URW Verlag, 1987.
- 34 “**in various other systems**” tinyurl.com/ms-blog-use-of-tex
- 34 “**not as many as once did**” Word has become a preferred manuscript submission format, and other systems (e.g., InDesign) are now good at fine typesetting.
- 34 “**disproportionately used in less-developed countries**” Glyn Moody, *Rebel Code: The Inside Story of Linux and the Open Source Revolution*, Perseus Publishing, 2001, p 317.
- 34 “**algorithms developed for \TeX** ” \TeX has so much computer science in it that Victor Eijkhout built a university computer science course around \TeX : Victor Eijkhout, *The Computer Science of \TeX and \LaTeX* , based on CS 594, fall 2004, University of Tennessee, pages.tacc.utexas.edu/~eijkhout/Articles/TeXLaTeXcourse.pdf; no doubt other word processors and desktop publishing systems also had lots of embedded computer science—lexing, parsing, semantic interpretation, optimization of searches, etc.—but their source files may not have been so thoroughly documented or may not be available for study.

- 34 **“line-break algorithm uses”** Donald E. Knuth and Michael F. Plass, *Breaking Paragraphs into Lines*, reprinted in Knuth’s *Digital Typography*, CSLI Publications, Stanford, CA, pp. 67–155; `TEXDR.AFT`, Chapter 24 of *Digital Typography*.
- 34 **dynamic-programming** \TeX ’s line-breaking algorithm is routinely used as an example in computer science algorithms courses in explaining dynamic programming.
- 34 **international collaboration on hyphenation** See hyphenation.org; also Mojca Miklavc and Arthur Reutenauer, *Hyphenation in \TeX and elsewhere, past and future*, *TUGboat*, vol. 37, no. 2, 2016, pp. 209–213, tug.org/TUGboat/tb37-2/tb116miklavc.pdf
- 34 **“Knuth’s boxes-and-glue model”** Nelson Beebe, *Using boxes and glue in \TeX and \LaTeX* , math.utah.edu/~beebe/reports/2009/boxes.pdf
- 34 **“hz micro-typesetting method”** Peter Karow has described the development of the *hz* in *Digital Typography* with Hermann Zapf, *TUGboat*, vol. 36, no. 2, 2015, pp. 95–99, tug.org/TUGboat/tb36-2/tb113zapf-karow.pdf; Karow describes the approach as “a justification per paragraph system — as described by Donald Knuth”. Hermann Zapf described his ideas on micro typography in his paper about micro-typography and the *hz*-program, *Electronic Publishing*, vol. 6 no. 3, September 1993, pp. 283–288, cajun.cs.nott.ac.uk/compsci/epo/papers/volume6/issue3/zapf.pdf. We know the details of Hàn Thế Thành’s $\pdf\TeX$ development, but not of the development in InDesign. Both implementations got help from Zapf or his work: wikipedia.org/wiki/Hermann_Zapf; Hàn Thế Thành email of 2017-09-10; Hàn Thế Thành, *An Experience from a Digitization Project*, cahiers.gutenberg.eu.org/cgi-bin/article/CG_1998___28-29_197_0.pdf; Hàn Thế Thành, *Micro-typographic extensions to the \TeX typesetting system*, dissertation, Masaryk University Brno, Faculty of Informatics, October 2000, reprinted in *TUGboat*, vol. 21, no. 4, December 2000, pp. 317-434, tug.org/TUGboat/tb21-4/tb69thanh.pdf
- 34 **Math layout algorithm** Knuth’s math layout algorithm (Appendix G of *The \TeX book?*?) was also an innovation, but it has not been as widely copied.
- 35 **\TeX and PostScript** Knuth created his own font format before PostScript was available. He developed his fonts with Metafont which generated bitmaps of characters, and \TeX used the fonts out of Metafont, the characters of which could then be passed to a printer driver. This was in keeping with Knuth’s desire to control his digital typography work down to the level of the pixels. The (later dominant) PostScript approach from Adobe provided outlines of characters to its raster image processor that sits between the PostScript interpreter and a printer. The \TeX and Adobe approaches had little in common, and people other than Knuth later provided ways for \TeX to use PostScript and PDFs.
- 35 **“embedded Lua language interpreter”** In several instances, someone has developed a way to escape from \TeX to a programming language (e.g., Perl, Scheme), but these have not been widely used. There was a DANTE e.V.-sponsored effort to reimplement \TeX in Java, but it was not practically useful. Now the Lua \TeX team has created a useful version of \TeX that keeps \TeX ’s macro capability and adds the ability to escape to the Lua programming language.
- 35 **John Warnock interview** Knowledge@Wharton, Adobe Co-founder, John Warnock on Competitive Advantages of Aesthetics and the “Right” Technology, January 20, 2010, tinyurl.com/wharton-warnock
- 36 **“easier to use and more productive than WYSIWYG systems”** This two-part paper, including the figures and the Webnotes, was composed and revised with \LaTeX before being converted to Word for submission to the journal’s prepress process.
- 36 **“variety of (typically rather small)” companies** Early instances of these were Micro \TeX , developed by David Fuchs, and PC \TeX , developed by Lance Carnes. David Kellerman and Barry Smith provided commercial support of \TeX on VAX/VMS systems. Barry Smith later moved on to the commercial \TeX tures distribution for the Mac. Berthold and Blenda Horn’s Y&Y \TeX was another small commercial distribution of \TeX for Windows.
- 36 **Overleaf and ShareLaTeX** overleaf.com
- 37 **“a few small companies are making money”** The Wikipedia article on “comparison of \TeX editors” (accessed in November 2017) lists 49 editors for \LaTeX , with eight of them requiring payment. Most of the 49 editors work at the source code level; five claim a mix of source-code and WYSIWYG editing; two claim to be WYSIWYG; four claim What-You-See-Is-What-You-Mean editing. (LyX, for instance, lets one edit graphically with menu commands to declare what lines of text are, e.g., title line, in-line equation, etc.; and the program then turns the text into a \LaTeX document out of view of the user.)
- 37 **“advertising section of *TUGboat*”** tug.org/TUGboat/Contents/listkeyword.html#CatTAGAdvertisements

- 37 **LuaTeX project** luatex.org
- 38 **“Eplain is an example”** Originally created by Karl Berry, tug.org/eplain
- 38 **“reproducing many of the macro definitions available in plain TeX”** In answer to the question of why didn’t Lamport implement L^ATeX on top of plain TeX, we are not aware of any explication by Lamport, and could merely speculate; and it is good that he did replicate plain TeX’s macro definitions as it allows users to escape out of L^ATeX to do things not readily possible in L^ATeX.
- 38 **“sustained effort of Hàn Thế Thành”** Interview of Hàn Thế Thành, tug.org/interviews/thanh.html
- 38 **“notable example of how”** Hàn Thế Thành, The PDFTeX Program, *Cahiers GUTenberg*, no. 28–29, 1998, pp. 197–210, cahiers.gutenberg.eu.org/cg-bin/article/CG_1998__28-29_197_0.pdf
- 39 **“by including a high-level language”** Knuth clearly believes in creating tools appropriate to the job at hand. For instance, one way of thinking of the Metafont typeface design system is as a programming language. Yet for that Knuth also developed a macro capability—more powerful in some ways and significantly different, as people writing mathematical character definitions have vastly different needs than people typesetting documents. Also, when Knuth created his literate programming system WEB which uses Pascal for the code, he included a macro capability at the WEB level, partly to get around some of Pascal’s shortcomings for systems programming; when literate programming systems targeting C were later developed, C’s own macro processor could be used.
- 39 **“primitive typesetting operations”** Primitive TeX is really basic. In addition to macros for adding new commands, special characters used by lexical analysis and character sets can be specified by users to be different than Knuth specified with plain TeX.
- 39 **“kicking and screaming”** Peter Seibel, *Coders at Work*, Apress, 2009, p. 597.
- 39 **“user extension of editors and other text processing systems”** For example, later versions of RUNOFF, TECO, troff, Script, and Pub—a prevalent text formatting system in the Stanford AI Lab where Knuth initially developed TeX. Indeed the SAIL language in which TeX was originally written supported macro definitions, and Knuth used them for TeX.
- 39 **“world of user groups”** en.wikipedia.org/wiki/History_of_free_and_open-source_software
- 40 **“Gaudeul did an extensive study on TeX”**
- Alexandre Gaudeul, The (L^A)TeX project: A case study in open-source software, working paper, September 17, 2004 (quite different text than in the 2003 *TUGboat* publication).
 - Alexandre Gaudeul, Competition between open-source and proprietary software: the (L^A)TeX case study, working paper, January 6, 2005.
 - Alex Gaudeul, Do Open Source Developers Respond to Competition? The (L^A)TeX Case Study, March 27, 2006 (a different paper than the next paper with the same name).
 - Alexia Gaudeul, Do Open Source Developers Respond to Competition? The (L^A)TeX Case Study, working paper, March 2007, perhaps a preprint of a paper of the same name in *Review of Network Economics*, vol. 6, no. 2, June 2007, pp. 239–263.