

Channels

- [Infotech](#)
- [Biotech](#)
- [Nanotech](#)
- [Extra](#)

Inside

- [Magazine](#)
- [Forums](#)
- [Newsletter](#)
- [Scorecards](#)

Special Events

- [Panel Series](#)
- [Nominate the Next TR100!](#)

Newsletter

Enter your email to receive our weekly newsletter.

TR on the air

JOHN BENDITT
ON EMERGING
TECHNOLOGIES



CNBC
DOW JONES
BUSINESS VIDEO

My Tools



2
free
trial issues

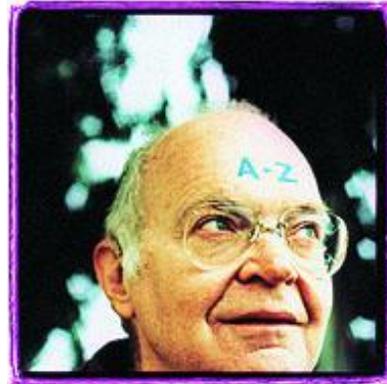
Rewriting the Bible in 0's and 1's

September/October 1999

By **Steve Ditlea**

Since the 1960s, Donald Knuth has been writing the sacred text of computer programming. He's a little behind schedule, but he has an excuse: he took time out to reinvent digital typography.

When you write about Donald Knuth, it's natural to sound scriptural. For nearly 40 years, the now-retired Stanford University professor has been writing the gospel of computer science, an epic called *The Art of Computer Programming*. The first three volumes already constitute the Good Book for advanced software devotees, selling a million copies around the world in a dozen languages. His approach to code permeates the software culture.



ANNE HAMERSKY

And lo, interrupting his calling for nine years, Donald Knuth wandered the wilderness of computer typography, creating a program that has become the Word in digital typesetting for scientific publishing. He called his software TeX, and offered it to all believers, rejecting the attempt by one tribe (Xerox) to assert ownership over its mathematical formulas. "Mathematics belongs to God," he declared. But Knuth's God is not above tricks on the faithful. In his TeX guide, *The TeXbook*, he writes that it "doesn't always tell the truth" because the "technique of deliberate lying will actually make it easier for you to learn the ideas."

Now intent on completing his scriptures, the 61-year-old Knuth (ka-NOOTH) leads what he calls a hermit-like existence (with his wife) in the hills surrounding the university, having taken early retirement from teaching. He has unplugged his personal e-mail account, posting a [Web page](#) to keep the software multitudes at bay by answering frequently asked questions such as, "When is Volume 4 coming out?"

About once a month during the academic year, Knuth comes down from the heights to a basement lecture room in the Gates Computer Science Building at Stanford to deliver one of his "Computer Musings" lectures, usually about some aspect of his current work on *The Art of Computer Programming*. These talks draw computer science students,

visiting professors, software engineers from nearby companies and an occasional CEO. On a balmy day earlier this year, the topic and listeners are different. To celebrate the publication of the third volume of his collected papers, *Digital Typography*, the associates of the Stanford University Libraries have invited an audience of fans of the printed word to hear Knuth talk about creating the TeX system for scientific and mathematical publication. Wearing a black T-shirt over a long-sleeve black shirt, his bald pate glistening in the overhead lights, he appears suitably monkish before about 70 acolytes and colleagues.

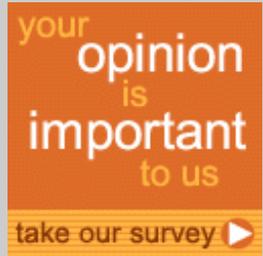
Hesitatingly, his words fighting his innate Lutheran modesty, he begins: "My main life's work and the reason that I started this whole project is to write a series of books called *The Art of Computer Programming*—for which I hope to live another 20 years and finish the project I began in 1962. Unfortunately, computer programming has grown over the years and so I've had to write a little more than I thought when I sketched it out." The faithful laugh knowingly.

Knuth relates his detour into digital typography during the 1970s. This was a time of enormous change in the typesetting industry, as computer systems replaced the hot type that had been used since the day of Gutenberg. Computer typography was less expensive, but also less esthetically pleasing—especially for complex mathematical notation. Recalls Knuth: "As printing technology changed, the more important commercial activities were treated first and mathematicians came last. So our books and journals started to look very bad. I couldn't stand to write books that weren't going to look good."

Knuth took it upon himself to write every line of code for software that yielded beautiful typography. He drew the name of his typesetting program from the Greek word for art—the letters are tau epsilon chi (it rhymes with "blech"). Says Knuth: "Well over 90 percent of all books on mathematics and physics" are typeset with TeX and with its companion software, Metafont, a tool Knuth developed to design pleasing type fonts.

He is quick to acknowledge the contribution of the type designers, punch cutters, typographers, book historians and scholars he gathered at Stanford while developing TeX. Some are in the audience. He tells them: "TeX is what we now call open-system software—anybody around the world can use it free of charge. Because of this, we had thousands of people around the world to help us find all the mistakes. I think it's probably the most reliable computer program of its size ever."

Anyone who doubts this claim by the decidedly unboastful Knuth can find confirmation from Guy Steele, one of TeX's first users and now a distinguished engineer at Sun Microsystems. TeX, says Steele, was one of the first large programs whose source code was published openly. Steele says Knuth's publication of the TeX code in a book, along with full comments, made it so that "anyone could understand how it works and offer bug fixes." With academe's top scientists and mathematicians as beta-testers, an extraordinary quality control team helped perfect TeX. (The TeX development effort was a model for today's open-source software movement, which has given the world Linux—an operating system that is beginning to compete with Microsoft Windows.)



Perfectability is a major preoccupation of Knuth's. The only e-mail address Knuth maintains gathers reports of errata from readers of his books, offering \$2.56 for each previously unreported error. (The amount is an inside joke: 256 equals 2 to the 8th power—the number of values a byte can represent.) Knuth's reward checks are among computerdom's most prized trophies; few are actually cashed.

He takes this error business very seriously. Engraved in the entryway to his home are the words of Danish poet Piet Hein:

The road to wisdom?
Well it's plain
and simple to express:

Err
and err
and err again
but less
and less
and less.

In a variation on this theme of perfectability, Knuth's contribution to computer science theory in the pages of *The Art of Computer Programming* has been his rigorous analysis of algorithms. Using methods in his book, the operations used to translate machine instructions into equations can be tested to determine whether they are optimal. Improving a program then becomes a question of finding algorithms with the most desirable attributes. Not that theoretical proofs can replace actually running software on a computer. In an often-cited remark he mentions on his Web page, he once warned a colleague: "Beware of the above code; I have only proved it correct, not tried it."

Bad Breaks

In Knuth's Stanford talk, perfectability was again a theme. He followed the pages in his volume on *Digital Typography* beyond its introductory chapters to the longest section in the book, which attacks a crucial problem in typography. He calls his listeners' attention to "one of the main technical tricks in the TeX system: the question of how to break up paragraphs so that the lines are approximately equal and good."

Poor spacing between words and ugly choices for line breaks had been among the major computer typography gaffes that launched Knuth on his TeX crusade. Odd word chasms, ladders of hyphens, and orphaned bits of text resulted from the rigid algorithms used to program line breaks without regard for visual elegance. Knuth's solution: have the computer use trial-and-error methods to test how each paragraph of text can best be broken up. Instead of "greedy" algorithms packing in the most words on a line—standard in computer typography before and after TeX—Knuth's computation-intensive method evaluates beauty.

Knuth seems born to the task of promoting beauty on the printed page—via computational methods. “I had a love of books from the beginning,” he tells his audience. “In my mother’s collection, we found the first alphabet book I had. I had taken the letters and counted all the serifs.” He is proud of his early literacy, telling a writer that he was the youngest member of the Book Worm Club at the Milwaukee Public



Library. His interest in typographic reproduction also came early in life. One of his earliest memories of pre-desktop publishing was helping his father, Ervin, with the mimeograph stencils for printing the church newsletter in the basement. Like his father’s newsletter, TeX was meant to be a homebrew project, on a manageable scale. “The original intent was that it would be for me and my secretary,” he tells *TR* in an interview in his home’s second-floor study. Leaning back in the black lounge chair, Knuth acknowledges that the long journey into TeX was intended to be a quick side trip: “I was going to finish it in a year.”

Events took a different turn. In 1978, Sun’s Steele—then an MIT grad student visiting Stanford—translated TeX for use on MIT’s mainframe computer. Suddenly, Knuth recalls, “I had 10 users, then 100. Each time it went through different levels of error. In between the 1,000th and 10,000th user, I tore up the code and started over.” Knuth says he realized then that TeX wasn’t just a digression, it was itself part of the vision. “I saw that this fulfilled a need in the world and so I better do it right.”

A key turning point in the spread of TeX was a lecture Knuth gave before the American Mathematical Society (AMS). Barbara Beeton, a staff specialist in composition systems for AMS and a longtime official of the Portland, Ore.-based TeX User’s Group, remembers the occasion: “He was invited to deliver the Josiah Willard Gibbs Lecture. Albert Einstein and John von Neumann had been among the previous speakers. Knuth talked about his new typesetting system for the first time in public.” Knuth was preaching to the choir; the assembled mathematicians were familiar with how printing quality had declined. Adds Beeton: “TeX was the first composition system meant to be used by the author of an article or book” as opposed to a publishing house. Soon after, AMS became the original institutional user of TeX, employing Knuth’s system to publish all of its documents and journals.

As word spread and more users took advantage of his free software (written for an academic mainframe computer but soon made available for PCs), Knuth found himself studying the history of printing to find solutions for narrow applications. Often as not, his research proved fruitless and he would have to come up with his own answer. For ceremonial invitations, he created new fonts; for musical typesetting he solved difficult alignment problems. “I had so many users,” he recalls. “From wedding invitations and programs for the local symphonic orchestra to computer programs.”

For nearly nine years, Knuth’s foray into typography occupied him full

time—pulling him away from work on the programming book that he considered his true calling. “I had to think of the endgame,” he says. “How could I responsibly finish TeX and say: This is not going to change anymore? I had to work out a four-year strategy to extricate myself” and return to *The Art of Computer Programming*.

Knuth’s solution: with the release of TeX version 3.0 in 1990, he declared his work complete. Disciples will have to maintain the system. Knuth says he will limit his work to repairing the rare bugs brought to his attention; with each fix he assigns one more digit to the version number so that it tends to pi (the current version is 3.14159).

One result of Knuth’s decision to stop making major changes to TeX is that the TeX file format has remained unchanged. “It’s the only software where you can take the file for your paper from 1985 and not have to convert it to print out the same way today,” notes David Fuchs, a senior researcher with Liberate Technologies (formerly Network Computer Inc.), who was a grad student at Stanford during the development of TeX. Fuchs estimates that there are 1 million TeX users worldwide; many employ special-purpose commercial packages built around the TeX “kernel,” such as L^ATeX (a command-oriented macro language) and TeXDoc (optimized for software documentation).

“On the downside, TeX is limited in its appeal because it’s not WYSIWYG,” Fuchs admits, employing the acronym for “what you see is what you get”—the standard term describing text processing software that displays formatting on screen as it will appear on the printed page. Rather than offering real-time onscreen interactivity, TeX requires a markup language typed into a document and interpreted by the computer; you see what you get only after it is in print. Despite its unintuitive user interface, TeX has developed a dedicated core of production professionals who will accept no substitute. “Why would anyone want anything else?” asks Paul Anagnostopolis, a Carlisle, Mass.-based publishers’ consultant and author of TeX-based software for book composition. “A lot of people don’t care about WYSIWYG.”

Opus in Progress

Spending nine years instead of one to create tex is the same kind of epic miscalculation that led Knuth to the monumental scale of *The Art of Computer Programming*. After earning his undergraduate degree at Case Institute (now Case Western Reserve), he was studying for his PhD and teaching at the California Institute of Technology in 1962 when he was contracted by textbook publisher Addison-Wesley to write a volume on computer compilers. (Compilers are special programs that convert the text typed in by programmers into instructions in a computer’s native binary language.)

In his book-lined study, Knuth recounts the history of the project. From 1962 to 1966, he wrote the first draft in pencil. The manuscript was 3,000 pages long. “I was thinking it was one volume of maybe 600 pages. I just figured type in books was smaller than my handwriting. Then I typed up chapter one and by itself it was 450 pages. I sent it to the publisher and they said: Don, do you have any idea how long your book will be?”

Faced with such an unwieldy manuscript, many publishers would have dumped the project. Instead, Addison-Wesley worked out a publication schedule for what could eventually stretch out to seven volumes. Volume 4 is supposed to be ready in 2004 and Volume 5 by 2009. Then Knuth may finish Volumes 6 and 7—if what he has to say on his chosen topics is still instructive. Peter Gordon, publishing partner at Addison-Wesley and Donald Knuth's editor for the last 20 years, explains that the success of the first three volumes of *The Art of Computer Programming* has allowed the publisher to build its entire computer science line around Knuth's work. "Don has his own life plan and his own sense of timing," he notes. "He's such a creative and gifted author, the best any editor can do is stay out of his way and let him follow his plan."

It helps that the book continues to draw praise from other seers in the digital realm. In his syndicated newspaper column, Bill Gates once responded to a reader: "If you think you're a really good programmer, or if you want to challenge your knowledge, read *The Art of Computer Programming*, by Donald Knuth." Gates described his own encounter with the book: "It took incredible discipline, and several months, for me to read it. I studied 20 pages, put it away for a week, and came back for another 20 pages. If somebody is so brash that they think they know everything, Knuth will help them understand that the world is deep and complicated. If you can read the whole thing, send me a resume."

What sustains Knuth through his epic project is his fundamental love of the subject. "People who work on the analysis of algorithms have double happiness," he says, sounding Yoda-like. "You are happy when you solve a problem and again when people smile using your solution in their software."

Before he can rest in the promised land, Knuth faces one last mountain. He must redesign the generalized computer used in his book for programming examples and exercises from a 50-year-old von Neumann-style machine with inefficient commands to a more modern RISC (reduced instruction set computer) system permitting faster operation. (Intel processors in most PCs are of the older variety; PowerPC chips in recent Macintosh models are RISC.) "I'm trying to design it so it's 10 years ahead of its time," says Knuth. "I've studied all the machines we have now and tried to take their nicest features and put them all together." This super RISC machine, which he calls MMIX, is essentially a teaching concept. But he says he "would love to see it built. I'm spending a lot of time documenting it so someone could build it. The design will be in the public domain." In the midst of his "Computer Musings" series of introductory talks on MMIX, Knuth is mere months away from completing this phase of his work.

And then? "I start charging away on Volume 4 at top speed. I can write about one publishable page a day," he says. On another book he once wrote at a rate of two pages a day "but that was too much. I couldn't be a good husband and a good father and that was not good. So, I'm just promising 250 pages a year for Volume 4." For devotees of Knuth's software Bible, Bill Gates included, those pages can't come soon enough.

As for Knuth's unwavering confidence in pursuing his long-term goal,

biology seems to be on his side. His mother is 87, in good health—still working, he says, in real estate office management. His father, who started him on the path to computer typography, died at 62. Still, this digital patriarch concludes: “My father’s father lived to be 97, so I’m hoping to take after him.”

Before he does reach the advanced age required to complete all his publishing plans, Knuth may have to face the temptations that come with fame. A two-page magazine ad for fatbrain.com proclaims, over an edgy collage: “Presenting the only bookstore on earth where Donald Knuth outsells John Grisham a billion to one.” This may be the first commercial acknowledgment of Knuth’s iconic status among the digerati. Will such recognition lead to a plague of attention just when the sage of software is about to resume his journey toward completion of his life’s mission? It makes you wonder how much longer Moses might have wandered the desert had today’s media been around.

Steve Ditlea is a contributing writer to *Technology Review*.

Related Links

[Donald Knuth's home page](#)

[MMIX](#)

[TeX Users Group home page](#)

[Answers to frequently asked questions about TeX from the UK TeX Users Group](#)

[TeX Usenet newsgroup](#)

[Listing of TeX consulting services and commercial TeX vendors](#)

[Return to Top](#)

[About Us](#) [Contact Us](#) [Privacy](#) [Terms of Use](#) [Advertise](#) [Subscribe](#)