

AmigaTeX . . . or How Envy Was Resisted and Knowledge Found on the Road to Ööç

Kim Kubik

THEY SAY THAT ENVY is one of the seven deadly sins. In early 1986 I finally decided I had envied others long enough, it was time to spend my own money on a personal computer to run TeX. I made a list of the pros and cons of the 8086 and 68000 based computers for which TeX was available and would vacillate from week to week on which one I thought best. Then the June TUGboat arrived and a short notice announced that TeX now ran on the Commodore Amiga, further stating that due to the multitasking operating system one could build “an impressive TeX environment.” I knew little about the Amiga but worked near Stanford so called the author, Tom Rokicki, and spent part of my lunch hour getting a quick overview of AmigaTeX. When I returned to work I got out that computer list, looked it over a last time, and tossed it in the garbage. And now, a year and a half after I bought an Amiga, I’m still finding out just what a gross understatement that phrase “an impressive TeX environment” really was.

◇ ◇ ◇

DEVELOPMENT of what would eventually become the Amiga began in the early 1980’s, a time when computer games were still the rage and many “real” computer companies in Silicon Valley were lamenting that they couldn’t get good engineers because the game companies were offering such lucrative salaries. A group formed out of this already select pool of talent to design a chip set for the ultimate game machine. Essentially they would tightly couple a Motorola 68000 cpu to custom RISC-style coprocessors to offload many of the cpu intensive chores necessary for sound and color graphics and animation. These components were designed to be modular with a multitasking executive at the core to coordinate their interactions.

Thus, when the bottom dropped out of the game market and the startup was refinanced by Commodore under the stipulation that the system be configured as more “business-like” the basis for a full-function workstation had already been implemented. A real-time message-passing operating system known as Tripos was ported to this hardware by a group of computer scientists in England. Tripos had been developed as a systems software research project at Cambridge (UK) with vision towards a

day when single users would have networked desktop workstations capable of running multiple programs simultaneously with sophisticated interprocess communication. Both a mouse-driven icon-based shell and a windowed Unix-style command line interpreter were added to Tripos and AmigaDOS was born. Suddenly the little game machine vaulted from the realm of single-tasking PCs and WYSIWYG Wonders toward league with the Suns and Apollos.

AMIGATEX makes extensive use of the multiwindow multitasking capabilities of the computer. Any text editor, the system command interpreter, the TeX compiler, and the AmigaTeX screen previewer may all be active in separate windows during a session. As soon as e.g. ‘tex myFoo.tex’ is typed at the prompt in the TeX window the user can slide back to the editor window and work on something else; as soon as page one of myFoo.dvi is output the user can preview it on the hi-res screen while the rest of myFoo.tex continues to be processed.

Or while the compiler continues to run the user can preview or print any other .dvi file, or score and listen to computer generated music, build objects for a color 3-D ray trace animation, or simultaneously download and unarc files from a bbs. The user is not forced into a linear one-dimensional mode of interacting with the computer. If there is sufficient memory to load a program the Amiga can generally run it no matter what else is active.

While the Amiga operating system does not support virtual memory like the big workstations it also does not burden the buyer with the expensive necessity of hardware memory management and a hard disk just for operating system overhead. As all executable programs are RAM bound a completely functional system can be run from floppies. This leads to one great advantage of the Amiga over any other true multitasking workstation: price. I purchased a single-floppy A1000 with 2.5 Mbytes of RAM in late 1986 for \$1600. I ran AmigaTeX on this with no hitches for a year before I even bought a second drive. The A1000 has been discontinued (but *not* obsoleted!) but the new models are low-end and high-end repackages of the same system.

For the TeX user on a severe budget a one Mbyte two-floppy A500 can be purchased discount for about \$1200; for someone needing maximum performance at minimal cost an add-in 68020 cpu increases the price only another \$1000. The Amiga model B2000 adds greater expandability and IBM-PC compatibility at a greater cost, but it is still priced far below any workstation with similar capabilities.

AmigaTeX ... or How Envy Was Resisted and Knowledge Found on the Road to Ööç

Kim Kubik

THEY SAY THAT ENVY is one of the seven deadly sins. In early 1986 I finally decided I had envied others long enough, it was time to spend my own money on a personal computer to run TeX. I made a list of the pros and cons of the 8086 and 68000 based computers for which TeX was available and would vacillate from week to week on which one I thought best. Then the June TUGboat arrived and a short notice announced that TeX now ran on the Commodore Amiga, further stating that due to the multitasking operating system one could build "an impressive TeX environment." I knew little about the Amiga but worked near Stanford so called the author, Tom Rokicki, and spent part of my lunch hour getting a quick overview of AmigaTeX. When I returned to work I got out that computer list, looked it over a last time, and tossed it in the garbage. And now, a year and a half after I bought an Amiga, I'm still finding out just what a gross understatement that phrase "an impressive TeX environment" really was.

◇ ◇ ◇

DEVELOPMENT of what would eventually become the Amiga began in the early 1980's, a time when computer games were still the rage and many "real" computer companies in Silicon Valley were lamenting that they couldn't get good engineers because the game companies were offering such lucrative salaries. A group formed out of this already select pool of talent to design a chip set for the ultimate game machine. Essentially they would tightly couple a Motorola 68000 cpu to custom RISC-style coprocessors to offload many of the cpu intensive chores necessary for sound and color graphics and animation. These components were designed to be modular with a multitasking executive at the core to coordinate their interactions.

Thus, when the bottom dropped out of the game market and the startup was refinanced by Commodore under the stipulation that the system be configured as more "business-like" the basis for a full-function workstation had already been implemented. A real-time message-passing operating system known as Tripos was ported to this hardware by a group of computer scientists in England. Tripos had been developed as a systems software research project at Cambridge (UK) with vision towards a

day when single users would have networked desktop workstations capable of running multiple programs simultaneously with sophisticated interprocess communication. Both a mouse-driven icon-based shell and a windowed Unix-style command line interpreter were added to Tripos and AmigaDOS was born. Suddenly the little game machine vaulted from the realm of single-tasking PCs and WYSIWYG Wonders toward league with the Suns and Apollos.

AMIGA TeX makes extensive use of the multiwindow multitasking capabilities of the computer. Any text editor, the system command interpreter, the TeX compiler, and the AmigaTeX screen previewer may all be active in separate windows during a session. As soon as e.g. 'tex myFoo.tex' is typed at the prompt in the TeX window the user can slide back to the editor window and work on something else; as soon as page one of myFoo.dvi is output the user can preview it on the hi-res screen while the rest of myFoo.tex continues to be processed.

Or while the compiler continues to run the user can preview or print any other .dvi file, or score and listen to computer generated music, build objects for a color 3-D ray trace animation, or simultaneously download and unarc files from a bbs. The user is not forced into a linear one-dimensional mode of interacting with the computer. If there is sufficient memory to load a program the Amiga can generally run it no matter what else is active.

While the Amiga operating system does not support virtual memory like the big workstations it also does not burden the buyer with the expensive necessity of hardware memory management and a hard disk just for operating system overhead. As all executable programs are RAM bound a completely functional system can be run from floppies. This leads to one great advantage of the Amiga over any other true multitasking workstation: price. I purchased a single-floppy A1000 with 2.5 Mbytes of RAM in late 1986 for \$1600. I ran AmigaTeX on this with no hitches for a year before I even bought a second drive. The A1000 has been discontinued (but *not* obsoleted!) but the new models are low-end and high-end repackages of the same system.

For the TeX user on a severe budget a one Mbyte two-floppy A500 can be purchased discount for about \$1200; for someone needing maximum performance at minimal cost an add-in 68020 cpu increases the price only another \$1000. The Amiga model B2000 adds greater expandability and IBM-PC compatibility at a greater cost, but it is still priced far below any workstation with similar capabilities.

One also gets something else for the modest investment: access to the end products from a community of some of the most avid computer enthusiast/programmers in the world. There are literally hundreds of public domain software tools for the Amiga. The open operating system allows these utilities to work together so that a user can tailor a truly flexible personal application environment.

This flexibility can yield a productivity increase a few orders of magnitude greater than any benchmark of straight cpu performance could possibly indicate. Perhaps a better “benchmark” of both the developer community’s sophistication and the Amiga’s capabilities is the fact that already next to my Amiga are five software product manuals typeset with AmigaTeX.

These capabilities are not without price: users have to learn AmigaDOS on their own as documentation of the Amiga as a workstation is all but nonexistent. Most dealers are completely ignorant of its potential, and it can take many (enjoyable) hours of testing new utilities to optimize an Amiga for specific work habits. Without a hardware deinterlacer (available only for the model B2000) there is an intolerable screen flicker in the hi-res mode that requires a very stable balanced room light and careful choice of colors to reduce.

Like TeX, AmigaDOS can initially seem frustrating because it offers so many choices: modular extensible software exacts a different kind of toll in the freedom it affords. When the Rexx language became available for the Amiga it was as if the last piece of a puzzle fell into place. A common vocabulary can now be used both to write macros within programs as well as to mediate interaction between programs.

As every capability of the system is available at all times the overlapping window paradigm becomes a true asset rather than mere dressing. The user who understands why TeX is not Desktop Publishing, and understands that ease of use is not enviable when the tradeoff is quality, might check out the Amiga. Like TeX, it can reward effort.

◇ ◇ ◇

ENVY is a transgression in that it focuses attention to surface appearance, diverting one from the true substance, *the kernel*, of a thing. The heretic might suggest that all one needs for an optimum TeX environment is a fast 32-bit cpu and a large screen. But when the system beneath such glitz is basically an 8-bit style program loader the suggestion is ludicrous. Absolution, at least in this

one case, came easily: before I got an Amiga, I *really* envied people who had those big Sun workstations to run TeX.

I don’t any more.

Using TeX and LaTeX with WordPerfect 5.0

Michael F. Modest
Pennsylvania State University

0.1 Introduction

While TeX is an extremely powerful typesetting language that can make your publications look as close to professional as you could possibly hope for, this beauty comes at a price: TeX by itself is somewhat tedious to use, and it is not WYSIWYG (what you see is what you get). There appears to be no truly WYSIWYG word processor available for the IBM PC or compatibles. The good news is that — when combined with WordPerfect, which allows customization of printer drivers — the tediousness of TeX can be overcome. I have written a TeX-LaTeX driver for WordPerfect which allows you to use WordPerfect normally while typing ordinary text, and uses simple macros when typing equations. I have written the preprocessor in such a way that it may be used for LaTeX documents or for straight TeX documents as long as the keys/macros specifically geared towards LaTeX are not employed. The user prepares his or her WordPerfect file and sees centering, indented paragraphs, Greek letters, mathematical symbols, subscripts, etc., actually displayed (and entered with single keystrokes). Then, rather than sending this file directly to a printer, “prints” it to a diskfile using the preprocessor which converts the WordPerfect format into a standard TeX or LaTeX file. If only the macros activated by the WordPerfect keyboards are used, the preprocessor even assures matching of braces, dollar signs and `\begin` and `\end` of LaTeX environments.

0.2 WordPerfect Fonts

I have programmed the driver so that — using the ordinary WordPerfect font change commands — a number of different types of fonts are available, as well as a predefined math displaymode, as indicated below. Depending on the graphics card and monitor present in your PC, different fonts and/or attributes may or may not be displayed:

- EGA card and monitor: choosing the 1 font/512 character option the full Greek alphabet