TurboMETAFONT: A New Port in C for UNIX and MS-DOS

Richard J. Kinch

We have recently released TurboMETAFONT, a new, TRAP-certified implementation of METAFONT in the C language. Like our TurboTEX software, TurboMETAFONT is portable to computers having C language compilers meeting ANSI or Kernighan & Ritchie standards and sufficient memory. We certified the software for MS-DOS PC's and AT&T UNIX System V in January. Certification for other targets is in progress.

Our earlier TUGboat article, "TurboTEX: A New Port in C for UNIX and MS-DOS" (TUGboat, vol. 9, no. 1, April 1988, pp. 48–52), described how we created TurboTEX from the Knuth autograph `tex.web`. The techniques and tools included `Paschal`, our Pascal-to-C translator program; a large TurboTEX change file; a few hand-written C functions to connect to the operating system environment; and utilities for splitting the large amount of C code generated into smaller, separately-compiled pieces. The earlier article describes these items in detail, so we will not repeat the particulars now.

To create TurboMETAFONT, we took the same approach as for TurboTEX, since the WEB programs for TEX and METAFONT are similar. A few items in METAFONT required some extra effort, but once the TurboMETAFONT change file was complete the same tools we perfected for the TurboTEX translation worked well for TurboMETAFONT.

METAFONT Yields One Surprise.

One substantial translation issue did arise which was absent from TurboTEX but critical for TurboMETAFONT. METAFONT uses integer arithmetic for all its calculations, while TEX uses some integer and some floating point. While the METAFONT syntax thus appears to be a subset of TEX's, in fact METAFONT makes a semantically broader use of integer arithmetic in ways that TEX never does.

For example, METAFONT takes differences here and there between *integer* variables (32 bits) and *halfword* variables and constants. Pascal prodigiously promotes all subrange integral types to the largest integral type when computing expressions. Thus, mixing of integral types in Pascal expressions is not prone to error, but is often inefficient. C economizes with an elaborate set of promotional protocols which attempt to minimize the word length of the arithmetic instructions in the generated code. Thus, mixing of integral types in C can be optimally efficient for the machine's instruction set.

However, in C certain combinations of integral types are dangerous, and the programmer is expected to be wary of them. For example, an *unsigned int* in C is a variable which can only take a non-negative integer value. Subtracting one C *unsigned int* from another always yields an *unsigned int*-expression result. If the result should have been negative, the effect is tacit nonsense. The similar case in Pascal, namely using the `0..`**maxint** subrange type, works because Pascal first promotes the subrange components of the difference to signed integers before using them to compute expressions.

We did not want to translate the METAFONT Pascal code to C and completely impose the Pascal integer promotion on the C code; we prefer to retain the C efficiency. But there is no way for a translator to know from syntax or semantics whether a given expression in Pascal will have problems in C of the kind illustrated above. So we had to make a painstaking manual examination of each suspect expression (fortunately, these are not too prevalent in METAFONT), and analyze whether any possible run-time behavior required C type casting for correct results. The change file implements C type casts where needed.

MS-DOS PC's: The Final Frontier.

Getting a 23,000-line program like METAFONT or TEX to run on a personal computer with limited memory is a big job. The portability traps of the IBM PC/Intel 8086 architecture and the MS-DOS operating system make the task still more imposing. We feel that Microsoft has done penance for inflicting MS-DOS upon the computer industry (and producing a 32-year-old billionaire as a side effect) by producing their C compiler, which makes MS-DOS look almost like UNIX to the programmer. This compiler significantly simplifies the task of porting to the PC.

The earlier TurboTEX software contained two features peculiar to the PC version: overlays and virtual memory simulation. At the time of the earlier article we had not completed these features, and so we will describe them now since they are complete, and also relate to TurboMETAFONT.

Both TurboTEX and TurboMETAFONT require overlays for the `INITEX` and `INIMF` versions of the executable programs. These versions are slightly larger than the production versions, since they contain initialization code otherwise absent. We implemented optimally overlayed code by a method of run-time tracing. By adding a run-time trace feature to the Paschal translator, we obtain function-call histograms for the anticipated uses, namely, running INITEX to obtain a Plain TEX or LATEX format file, or running INIMF to obtain a Plain METAFONT or Computer Modern METAFONT base file. By overlaying the largest modules which are called only once or never during these runs, we obtain executables with near the performance of a non-overlayed version.

We developed a virtual memory simulator, released May 1987, for TurboTEX. The virtual memory feature allows a mainframe-sized TEX to run on PC's, which have an architectural and operating system limit of less than $600K$ bytes of usable program memory. By putting TurboTEX's largest array, *mem*, into simulated virtual memory we free enough space for a large version to run in the available PC memory. This technique succeeds with large macro packages and complex documents that cause other PC implementations of TEX to fail.

We do not now see the need to include the virtual memory simulator in the TurboMETAFONT programs. Since the TurboMETAFONT programs are smaller than TurboTEX, TurboMETAFONT will run on any PC with $640K$ memory installed. Furthermore, the enterprise of generating fonts with METAFONT does not seem to encourage the use of enormous macros or tables that accumulate from character to character, the way some applications of TEX quickly gobble memory for index entries or bit map manipulation. We may still offer a virtualized version of TurboMETAFONT in the future for PC users, if commercial demand proves a need.

Portable Preloading.

Another TurboTEX feature not needed by TurboMETAFONT is the portable preloading. This method in TurboTEX uses our utility program, FMT2INIT, to convert TurboTEX format files into C variable initializers, thus creating executable programs that start quicker with formats preloaded. Having such a preloaded executable is desirable for TurboTEX, because the TEX user typically makes many short runs of the program, so that the program load time dominates. The typical use of METAFONT, however, is for a lengthy run to generate an entire, existing, debugged font in a new magnification. Thus the value of preloading a TurboMETAFONT executable is small, and we have not carried over this feature.

Graphics Support.

A desirable feature of METAFONT is the online graphical output. This allows the user to view each character in a font, or even each stroke in a character, as METAFONT generates it. For the font developer, this interactive mode is critical to effective debugging of character programs. For the rest of us, it is just fun to watch the artistry in action.

On the PC, TurboMETAFONT supports the most common graphics adapters: the Hercules monochrome graphics adapter (HGA), the IBM color graphics adapter (CGA), and the CGA-compatible modes of the IBM extended graphics adapter (EGA) and virtual graphics adapter (VGA). We wrote the HGA driver code ourselves, but used the CGA/EGA/VGA drivers in the Microsoft C compiler's graphics library. We provide a separate executable program for each graphics adapter type to minimize memory use and increase execution speed. If there is sufficient demand we may incorporate Tektronix 4010 graphics output as an option for applications on larger (non-PC) computers.

Distributing the Product and Enlarging the TEX Community.

We have drawn much encouragement from the market acceptance of TurboTEX, having shipped over 500 copies in the first six months of publication with only TUGboat advertising. We will continue our approach of one low price for a complete package, now to include both TurboTEX *and* TurboMETAFONT, with complete source code available.

The TurboTEX emphasis on the pure C language and portability has served to significantly enlarge the number of TEX users. There is a large and growing body of potential TEX users who share a set of characteristics: They are programmers who work with C, they own or use a PC, they know and admire Knuth's *The Art of Computer Programming* series, they need a technical typesetting system, they have heard about TEX, and they know the shortcomings of other "desktop publishing" programs. TurboTEX, we have found, appeals to such people as a good fit in all these ways to get them started as serious TEX users. We plan to broaden our approach to appeal to these unTEX'ed masses, and perhaps they will soon be the new generation of TEX users.