The net effect is, if your C compiler knows how to handle casts (up to now, every C compiler we could test did), that this source runs on both types of machines.

Passing the Trip test

The important milestone after the banner is the Trip test. Needless to say, it didn't run at once. In fact, there were some very subtle bugs introduced while rewriting. It took another half a year to succeed with this test. One of the main problems lay in the input routine, where we didn't use Knuth's raw version, but the optimized version that happened to be in the change file (ever heard of a change file?). Needless to say, the algorithms and data structures in TEX are computer proof; that means, that if you have a compiler that deserves this name, you get this program running.

Sure enough, people learned about our project and asked for a port, if we ever got it running. Most of the time these people were more optimistic about a possible success than we were. But then we could make the (ultimate) test for portability.

The one thing we learned is, that TEX (whether in Pascal or in C or whatever) is not only a typesetting system, but also a compiler test system. There were some problems compilers introduced with the input of TEX, but if you wanted to demonstrate the bug to the computer or compiler distributor, you couldn't reproduce the error with a normal sized program. We encountered the fact that fixed array locations like `mem[32760]`, as happen to be used as kind of register in the typesetting processor TEX, will be translated to anything, but definitely not to the locations you would expect. Also you have to cope with the most tricky optimizers, which try to keep the program small enough by optimizing procedures away, or deleting the index in `z = mem[z]` before using it. But these problems were not too often encountered, and after tuning some I/O statements not for efficiency but for portability, we now have the following ports:

Cromemco V.0 and V.2
PCS Cadmus 32Bit UNIX Systems
ALTOS UNIX and XENIX Systems
Convex with 4.2 UNIX
AT&T 3b2 running V.0 and V.2/V.3
HP Series 9500 and 93XX under HP-UX
IBM RT under AIX

The only preconditions we pose are that we have a true 32-bit CPU (not an 80286) and we prefer UNIX or UNIX look-alikes, but we don't insist on this (as people insist on a VMS version).

Extensions

With this TEXinC version we started a cooperation with a German typesetter. In this project we designed an extended TEX program, which we call PhotoTEX, to cope with the possibilities available with phototypesetting machines, and made some adaptations for German respectively European environments. The PhotoTEX Program understands two additional keywords, `setsize` and `slantsize`, so that we are capable of handling dynamic fonts in the typesetting machine. Also we had to create metric files (tfm-files) for the fonts that are resident in phototypesetting machines, as there are machines that are not able to download fonts. Another hard problem was to find the right kernings, as typesetters need them.

An additional problem for the German environment is hyphenation of words with Umlaute (and other special characters you encounter in a European environment). At the moment, there are two solutions we know for hyphenation, both coming from the University in Bonn, Germany. One is to fool the hyphenation routine, while the other, and by far better, solution requires a minor change in the METAFONT description, recreating the fonts, and an addition in the dvi driver. We preferred the second approach for our German version of TEXinC.

TEX Adapted to CWEB

David Kennedy
Micro Publishing Systems, Inc.

This article announces TEX in CWEB, a new starting point for TEX ports. We have recently completed the translation of TEX to CWEB, a version of Don Knuth's WEB system of structured documentation, entirely rewritten in C, with many changes to take advantage of features found in C, but not in Pascal. (For a more complete description of CWEB refer to the TUGboat article: *WEB Adapted to C, Another Approach* by Silvio Levy, April, 1987).

Although this is a commercial venture, and the TEX translation is proprietary, we are offering a copy of the binary and/or source code for a reasonable license fee. We are also planning a fall 1988 commercial release of our fully TRIP-certified version of TEX for the PC and plan to release UNIX

System V, XENIX and OS/2 versions in the near future.

Given the current availability of TEX implementations on microcomputers, a TEX site needs to acquire a different implementation for each of its systems, assuming that a version exists. Our objective is to provide an implementation of TEX which is easy to port to various microcomputers and operating systems and preserve the documentation features of CWEB so that a system administrator can maintain one source to TEX for various small systems.

The original development work was done in MS-DOS on an AT compatible computer using the Microsoft C Compiler version 5.0. We are interested in hearing from institutions and individuals who would like to port TEX in CWEB to other unique operating systems, computers, and/or languages.

Adapting TEX to CWEB provides simultaneously a language like TEX for formatting and a language like C for programming. Once the WEB is translated to CWEB, a program called TANGLE is used to produce the C source code. Another program called WEAVE is used to produce a structured document, so that the large system can be understood entirely in terms of small modules and their local interrelationships. Any changes such as splitting the C source into multiple files is handled automatically within TANGLE (the entire C source produces an object file greater than the 64K limit for the Microsoft C Compiler). Any system specific changes such as those necessary to accomodate the 64K byte address space limitations imposed by the segmented architecture of the 8086 processor can be made in the change file included by TANGLE.

Now that the original translation of the TEX WEB to CWEB has been completed, and the major port for MS-DOS has been done, ports to other C environments will follow quickly. The most important thing is maintaining the source to TEX in a generic form while allowing for the smooth integration of system-dependent changes.

Fonts

Some Useful Variations of Standard Fonts

Glenn L. Vanderburg
Texas A&M University

The old "Almost Computer Modern" font set had a font called `amssmc40`. It was used for the chapter titles in the first edition of *The TEXbook*. It also turned out to be very popular for making signs. When the Computer Modern fonts were released, there was no equivalent font in the standard set. It was replaced by `cmssdc10 scaled 4000`. This is essentially the same font, with some minor improvements and one major disadvantage: the unusual magnification. Unfortunately, many DVI drivers have difficulty dealing with fonts which are not magnified by one of the standard "magstep" quantities.

To remedy this, I added the following line to `cmssdc10.mf` just after `input cmbase`:

```
numeric Pt#; 1Pt#=4pt#;
```

and changed all occurrences of '`pt#`' in the rest of the file to '`Pt#`'. I called the new file `cmssdc40.mf`. The font produced by it is not exactly equivalent to `cmssdc10 scaled 4000`, but the differences are negligible.

A similar modification to `cminch.mf` yields another extremely useful font. I noticed that a couple of signs I made with `cminch` had a harsh tone because of the capital letters, even when the wording was very friendly. Changing the '`generate title;`' line at the end of `cminch.mf` to '`generate roman;`' creates a `cminch` font with lowercase characters (among other things). It takes up a lot more disk space, but many people use it here at Texas A&M. It should not be called `cminch`, because it's not the same font. I call it `cmssbx104` (when the capital letters are one inch high, the font size is 104 points). This will cause problems for systems (like CMS) which have severe restrictions on filename length, but I have been assured by "people who know" that there is a convention for handling this problem.

Modifications like this are extremely simple to make, and can be quite useful. A new version of `cminch` with characters one centimeter high would be nice; it could be called `cmcm`. (Along the same lines, Tom Reid once observed that `cminch scaled\magstep1` might reasonably be called '`cmdecifoot`'.) I've seen several people requesting `cmssdc40`; it turns out to be very easy to create, and I think people will enjoy using it.