In addition there are several problems with \send files. The text to be sent must have length less than the LRECL for that file. And in line 1540 change

rwritefile(fil,'DDNAME ='||trim(fname));

to *rwritefile(fil,fname);*

Change line 1584 from **ELSE IF** *fj* <= 8 to **ELSE IF** *fj*<=8

The development effort for TEX at CIT is now directed at installing TEX82. I expect to report on our progress at the next meeting, and I encourage others who are also installing TEX82 to report back on your efforts either to me before the meeting or at the IBM Birds of a Feather session at the March meeting.

* * * * * * * * * * * *

TEX INSTALLATION AT THE UNIVERSITY OF MICHIGAN

Paul Grosso

ABSTRACT. Work has been done at the University of Michigan to convert parts of the TEX system to run under MTS. Currently we have the capability to run TEX and produce text output on a Linotron 202 typesetter in standard, bold, and italic fonts in a variety of point sizes. Further work needs to be done on input macro packages and the user interface as well as the support of other output devices.

Over the past year, the joint IBM/University of Michigan Word Processing Project has been experimenting with parts of the TEX system. Having gained much experience with it as well as some related software, the University of Michigan (UM) ordered a Pascal/VS copy of TEX from the Stanford Center for Information Technology (CIT) designed to run under IBM's MVS operating system.

The Word Processing Project (WPP) has an IBM 370/148 running VM/CMS, MVS, and MTS (Michigan Terminal System) with a network connection to UM's Amdahl 470V/8 running MTS (MTS-UM). When the tape came from Stanford (in record time—a week after we mailed out our order we had a tape in our hands), we read most of the files directly into files under VM/CMS; one file, that containing the font file partitioned data set, was written in "unloaded format", so we read it under MVS and transfered the individual font files to VM.

TEX compiled with no problems; with a few VM file definitions (to reassign files to DDNAMEs), TEXPRE ran generating TEXINTBL. A few more file definitions and TEX generated a DVI file for a sample input file. We then transfered the files to the MTS operating system on the 148 (MTS-WPP) and made several changes to the SYSDEP module to convert TEX to run on MTS.

File naming conventions were the major changes including the removal of most file extensions, allowing file names of up to 20 characters including periods, altering the option specifications of the RESET and REWRITE calls for MTS usage, and changing the PDSIN call used to open font files into a standard RESET call.

Some values in the ASCII↔EBCDIC tables had to be altered to allow for differences between the EBCDIC codes that IBM and MTS uses for some of the characters; we had some hard to trace problems before these differences were noticed—would you believe the characters in question were \ (backslash), ` (grave), { (left curly brace), and } (right curly brace). Then there were the differences in character sets caused by the discrepancies between standard ASCII and Stanford SUAI code. Presumably because the right curly brace is ´175 in ASCII and ´176 for SUAI, the CHRX array in SYSDEP had both the '7D'x and '7E'x mapped into 'D0'x (right curly). Normally in ASCII (and to be consistent with its inverse in the ORDX array), '7E'x should get mapped into '5F'x (not-sign or tilde)

We also had some problems with ASCIITBL. Line 34 (character ´042) was a vertical bar (| or '4F'x) and should have been an exclamation point (! or '5A'x). Line 95 (character ´136) was a not-sign (¬ or '5F'x) which we changed to a cent-sign (¢ or '4A'x) to agree with SYSDEP. Line 125 (character ´174) was a broken vertical bar (¦ or '6A'x) rather than the vertical bar expected by SYSDEP (| or '4F'x). Finally to maintain consistency with the right curly brace change to SYSDEP mentioned above, lines 126 and 127 (characters ´175 and ´176) were changed from ALT and right curly to right curly (} or 'D0'x) and not-sign (¬ or '5F'x) respectively. (Of course, the chcode declaration in the BASIC macro file making character ´176 a close delimiter had to be changed accordingly.

We encountered a Pascal/VS problem running TEXPRE on MTS. Due to its large memory requirements (and some glitches in the MTS Pascal/VS), we had to increase the STACK and HEAP Pascal/VS run-time parameters to get TEXPRE to run properly. TEX itself bombed reading font files that had lines that had been padded with a blank (introduced because files are shipped from VM/CMS to MTS-WPP through a "virtual card reader," and MTS trims all but one blank when trimming file lines). When this was corrected, we ran across several problems with the code in INLN in SYSDEP.

A logical test for end-of-line failed at the end of a file and an end-of-line test had to be added to the logical expression. (Technically, in Pascal EOLN is not defined for a text input file when EOF is true. On page 115 of IBM's Pascal/VS Language Reference Manual, EOLN is to be true if the file is positioned to the EOLN character and false otherwise. However, under MVS and VM/CMS, EOLN is true at the end of the file, and TEX depended on this. In the Pascal/VS at UM, EOLN was false when EOF was true.) Also, we got a Pascal/VS run time error when reading input lines that were too long and needed to be broken. The code in INLN failed to assign a value to the last character of the buffer; we assigned "buffer[bufptr]:=c" after the REPEAT...UNTIL. All in all, the initial installation went smoothly and took less than a week of part time work.

Noting that TEX reads and writes four byte file lines (most notably TEXINTBL which is over 44,000 lines and the DVI file itself) and that Pascal/VS does no blocking of such files, we added some blocking to TEX's I/O routines. We noticed about a threefold improvement in TEX's execution speed on MTS by blocking TEXINTBL and the DVI files into 4000 byte records. Furthermore, since MTS does its own dynamic memory allocation (and charges for memory for just the time it is allocated), we made several changes to TEX's memory management process.

Initially we allocate a small part of the MEM array (by using the REF feature of Pascal/VS and allocating the space dynamically with our operating system) and then get and free blocks of extra memory as required. We still would like to make some additions and changes to SYSDEP so that the interaction between TEX and the user at the terminal works a bit better on MTS.

With the help of David Fuchs, a program was developed to convert DVI files to a format that the Linotron 202 can handle. Of course, much of the display mathematics and more interesting capabilities of TEX are not available since we are restricted to the fonts available on the Linotron, but the output looks good. Next we hope to investigate cheaper proof quality output devices so the expensive Linotron copy need only be produced for the final copy. In the future we may consider purchasing a phototypesetter such as an Autologic APS-Micro 5.

VAX/VMS

Monte C. Nichols
Sandia Laboratories
Livermore, CA

David Kellerman
Oregon Software
Portland, Oregon

The most recent Stanford version of TEX-Pascal along with the improved spooler for the Versatec is now available from Oregon Software. This is of course the version having magnification capability, etc., and is the last version prior to TEX82. The new version should get rid of most of the remaining bugs and keep the VAX/VMS community running in a superior fashion until a version of TEX82 becomes available for VMS. For those of you new to TUG, Oregon Software has volunteered to distribute TEX for the VAX/VMS community. For $50 they will send you a tape with all the VAX/TEX related files on it. See TUGboat Vol. 2, No. 2 and Vol. 3, No. 1 for further information.

A rather large number of VMSer's attended the most recent TUG meeting at Stanford as well as the classes for TEX82 that followed. Several individuals have gotten WEB, TANGLE and WEAVE running on the VAX/VMS system, so we should be able to see a preliminary version of TEX82 at some reasonable date. Our best guess for the availability of a production version of TEX82 would be Nov–Dec 1982, but it could happen sooner.

The newest version of TEX as well as TEX82 use .tfm and .pxl font files. Unfortunately the 200 dot/in font files made available from Stanford in this new format did not include some of the old font files we had come to know and love; especially noted by their absence were the larger fonts that had proven useful for a number of purposes. One of us (MCN) has obtained METAFONT files for some of these missing files and plans to find a DEC 10 or 20 soon where the missing 200 dpi fonts can be generated.

Anyone who has developed spoolers for devices other than Versatec is encouraged to send them to Oregon Software for inclusion on the next distribution tape.