* * * * * * * * * * *

Site Reports

* * * * * * * * * * *

NEWS FROM THE TEX PROJECT

David Fuchs

As of mid-August, the state of the world is as follows: All of the WEB listings have been re-printed in red covers. This is the final printing of TEX82 before it becomes a real book. The TEXbook was also re-printed in a red cover, missing only Appendices D and E; we will wait a few weeks to get people's reviews, then we'll phototypeset it in September, and off to the publisher it goes. The whole set of preprints is available through our distribution service, along with the latest distribution tapes.

TEX is frozen. The only changes being made are bug fixes. We are currently on version 0.9999 (no kidding). It is slightly different than the red manuals, but all changes made to the system are listed in a file called RED.ALERT on the distribution tapes, and they will also be published in TUGboat (see below for the changes that have happened so far). When the TEXbook is available, we'll declare version 1.0, and there will be much rejoicing and throwing out of old TEX manuals. By the way, bugs in the code are now worth $5.12, while bugs in the TEXbook will get you $2.56.

Tapes formats available are generic ASCII, generic EBCDIC, Tops-20, VAX/VMS, VM/CMS (by the time you read this, we hope) and MVS (likewise). Tapes for Berkeley VAX/Unix are available from R. Furuta, as mentioned in his note on page 74. There are also some new font tapes for 200, 240 and 300 dot/inch devices. Mike Spivak's first shot at $\mathcal{AMS}$-TEX for TEX82 is also included on the distribution tapes.

The Vax/VMS port was done by me, using the new Pascal 2.1 compiler from Digital. The distribution tape is in standard BACKUP format, and has all the sources as well as EXE files. You do not need to have the compiler to run the EXE files, since they were produced with LINK/NOSYSSHR. If you do have the compiler, you'll want to re-link everything so that the runtimes will be shared at execution time. If you get this tape, a good way to start is to look at the file [.TEX.DOC]VMS.DOC for further details.

The Tops-20 port was done by me, using the Hedrick Pascal compiler from Rutgers. The distribution tape is in standard DUMPER format, and has all the sources as well as EXE files. You'll need to have the compiler to run the EXE files, since the runtimes are mapped in from a common file at runtime. This shouldn't be a problem, since the compiler is virtually free. If you get this tape, a good way to start is to look at the file <TEX.WEB>TOPS20.DOC for further details.

The VM/CMS port was a combined effort, using the IBM Pascal/VS compiler. The distribution tape (by the time you read this) is in standard TAPEDUMP format. You shouldn't need to have the compiler, since the executable programs are included. If you get this tape, a good way to start is to look at the file VMCMS DOC for further details.

At present, there are no device drivers on the distribution tapes. We would be happy to add any drivers that people see fit to contribute. Other, related software can also be included where appropriate.

Actually, a bit more needs to be said about the font distribution. First, we have totally moved over to the new "AM" series of Computer Modern. For a while we were using a mix of some old "CM" and some incomplete "AM" fonts, but now the "AM" series is complete. The older "AM" and "CM" fonts are now out of date, because some characters have changed position, and all of the empty slots have been filled in (lots of new characters, including a proportionality sign). PLAIN.TEX knows about the layout and the \skewchar features in the new "AM" fonts, so you'll need them to run TEX82. The new "AM" fonts are also frozen in the sense that characters won't be moving around any more, but there will be improved versions of the characters themselves some months down the road. Finally, some of the font names have changed a little, so that the whole system is more logical and self-consistent (the old CMATHX is now AMEX10, while CMDUNH is now AMDUNH10, etc.).

As discussed at the TUG meeting, the latest TEX has a few interesting notions about magnifying fonts. In particular, a font can be magnified by a factor of 1.2 by saying `\font\foo=amr10 scaled \magstep1`. An entire document can be magnified 1.2 times by saying `\magnification\magstep1`. Similarly, `\magstep2` is a factor 1.44. In fact, you can use up to `\magstep5`, each one being a factor of 1.2 bigger than the previous. There's also `\magstephalf`, which is almost 1.1 (actually, `\sqrt{1.2}`). Anyway, what we're distributing is all of our AM fonts in the lower `\magsteps`, plus the main ones in the higher `\magsteps`. (The "main ones" are exactly the ones from PLAIN.TEX that are not named `\preloaded`.)

One interesting result of doing things this way is that there need be only one series of pixel files for 200 and 240 dots/inch devices. Consider: the PXL file for `\magstep2` of a 200 dot/inch font is the same as the PXL file for `\magstep1` of the same font at 240 dots/inch. Indeed, we now provide a single distribution tape for both 200 and 240 dots/inch. The only complication is that `\magstephalf` fonts from each series don't fit into the other series, so recipients of the distribution tape may delete the inappropriate fonts if desired.

The main problem with providing so many magnifications is that the amount of font data is getting too large to fit onto the distribution tapes. For this reason, we are forced to change the way that fonts are stored on the tapes. Now, each byte from the file is a single byte on the tape. The block size is 8000 bytes, and the last block of a file is only as long as it needs to be.

One item discussed at the TUG meeting was that it would be nice if TUG members could dial up a Stanford computer and read a bulletin board with recent TEX news. As soon as we get our Sun machines going, we'll try to provide this service. This should allow folks without CSnet access to read TEXhax between TUGboat issues.

* * * * * * * * * * *

TEX82 ON CP-6

Rick Mallett
Computing Services
Carleton University
Ottawa, Ontario, Canada.
Tel: (613) 231-7145

TEX82 version 0.95 is up and running on the Honeywell CP-6 computer system here at Carleton University, and thanks to the folks at Stanford, it took very little effort to get it going. From an implementer's point of view TEX has certainly come a long way from the days of TEX in Pascal, which took us more like 6 months to get working, although I must admit that a good portion of that time was spent debugging our Pascal compiler. Getting TEX to run efficiently is quite another matter though — we've been "changing" TEX all summer and we only finished about 3 hours ago.

Our biggest problem stemmed from the fact that, although our operating system allows a core image to be saved and restored, the Pascal runtime is such that the I/O streams are not properly re-initialized. Therefore, we had to take the other obvious approach and block the FMT and TFM files and change TEX to handle them. This turned out to be more work than we had expected, mainly because we kept tripping over the hacks in the I/O routines to get around the problem that the DEC-20 handles terminal and file I/O differently.

Before further describing our conversion difficulties, however, I would like to tell you a bit about who we are and what we have been doing with TEX. I'd also like to comment briefly on the CP-6 operating system, since I suspect that very few people have ever heard of it.

We first implemented TEX in 1981, when we used it to help debug the Pascal compiler that we were writing for CP-6, but without an output device it did not see a lot of use. However, in the spring of 1982 we set to work to build a controller for the Canon LBP-10 laser printer, and by September we were finally printing pages formatted by TEX. Since that time we have printed over 19000 pages, from course notes, reports, and theses to an article for a Ukrainian hiking journal, and more recently a linguistics text that required the entire phonetic alphabet.

Now for CP-6. CP-6 is a relatively new (ca. 1980) operating system, which runs on the large Honeywell 36 bit machines, and it was designed to be similar to the CP-V operating system which ran the Xerox Sigma and 560 series computers. Some of you may recall that Xerox dumped support for its large scale computers in 1976; fortunately for us, Honeywell Information Systems hired the same team of people who had created CP-V and put them to work writing a similar operating system to run on their 36 bit machines. The result is a "state-of-the-art", large-scale, time-sharing system that seems optimal for someone who needs something larger than a VAX 11/780, with equivalent or better functionality, but who does not want to pay the exorbitant prices that IBM likes to charge for a "large" machine.

Actually, there isn't all that much to say about our implementation of TEX on CP-6, as most of the problems we encountered have been reported in the last issue of TUGboat. We had to change all the file names, of course, and modify *InputIn* so that the first character of the first line of terminal input wasn't lost, and set *name_length* before opening the pool file, but for the most part it was pretty straightforward. Unfortunately, we wasted a bit of time writing PXL to CHR and CHR to PXL conversion programs, as we did not realize that these routines had already been written and are now on the distribution tape, but it was not that hard anyway. Like the people porting TEX

to VAX/UNIX, we had to modify our compiler to support `CASE OTHERWISE`, but, since we had written the Pascal compiler, it was fairly easy to change it, once we had convinced ourselves that it really was acceptable to provide features not included in the "proposed" standard. (The experience of putting up TEX80 without `CASE OTHERWISE` was a rather convincing argument.)

A more serious problem arose from the fact that we had decided long ago to store all reals as 72 bit quantities, and we did not want to go back and change the code generator to use single precision reals. Although the address space on our machine is such that TEX fit easily even with the double-sized MEM array, our experience with TEX80 had shown us that we could not afford to make heavy use of TEX without seriously degrading system performance—it's pretty hard to run over 100 users with only 6 million words of real memory. Fortunately, the last issue of TUGboat arrived just at the moment that we were agonizing over what to do about the problem, and we learned from Pavel Curtis' and Howard Trickey's comments that modifying all references to the MEM array so that only the most significant bits of all reals are stored there isn't as hard as we had imagined. It turned out that we were able to make the modification in one afternoon. Of course, it should be even easier the next time around with the *float* and *unfloat* routines that have now been added to TEX.

Other than that there were no major problems to the initial conversion. As I pointed out earlier though, our latest round of performance improvements which involves replacing all the Pascal I/O was quite another story.

Finally, there is one problem that we have encountered with TEX that I suspect is widespread, but I have not seen it reported elsewhere. It turns out that TEX is so useful that everyone wants to use it, and we are still trying to figure out how we can afford to make TEX available without swamping the system. The solution that we are looking at is to charge for all TEX services, but we have not been forced to charge academic users before, and we are not looking forward to setting up the necessary bureaucracy. It sure seems stupid to have to penalize those students who go out of their way to create nice, neat, legible reports and theses. Nevertheless, it looks as though that's the way things will have to go around here for the time being.

* * * * * * * * * * *

IBM SITE REPORT

Susan Plass

TUG members from 13 IBM-architecture sites met in a Birds-of-a-Feather session at the July 1983 annual meeting. About half of these sites currently run a version of TEX; most of the rest are in the process of installing TEX on their systems. Of these sites, 6 run MVS, 5 run VM/CMS, 1 runs both MVS and VM, and 1 runs TSS. We explored the idea of breaking the IBM group into two groups, one for MVS and a second for VM. The consensus seemed to be that there was no need yet to break the group formally, but Alan Spragens of SLAC has volunteered to serve as a focal point for a VM-oriented information exchange. Give Alan a call if you have VM/TEX questions and especially if you have VM/TEX answers!

* * * * * * * * * * *

UNIX TEX SITE REPORT

Richard Furuta[†]
Department of Computer Science
University of Washington

Work on Unix TEX for Berkeley Unix, 4.1 bsd, has been proceeding quite uneventfully since our last report in these pages. Well, perhaps not completely uneventfully—we ran comparative benchmarks against the Unix and VMS TEX82, Version 0.97, implementations and were quite surprised to discover that the VMS version was several times faster than the Unix version. Thanks to Howard Trickey's Herculean efforts, and replacement of the critical inner loop input routines with C language routines, the Unix TEX82 has been sped up quite noticeably and we feel that the Unix running speed is now quite acceptable.

Several other people have contributed to TEX on Unix recently. Steven Correll of the Lawrence Livermore National Labs sent along a patch to allow an editor escape from TEX into vi. Howard added an escape for Gosling's **emacs** and this code is now in the distributed version of TEX82. Mark Senn sent a BBN Bitgraph DVI page previewer he wrote while at Purdue. This will be included in the Unix distribution when the necessary fonts become available. Further contributions to the Unix TEX distribution are invited.

[†] This work is funded, in part, by a grant from Northern Telecom.

The current version of TEX82 on Berkeley Unix, 4.1 bsd, is now 0.9999. Mike Harrison at Berkeley has verified that this version of TEX will also run on 4.2 bsd, and a number of sites are running it on 4.1c, as well. Some work has been done to get this version of TEX working on the Sun workstation, also running 4.2 bsd. We are eagerly awaiting news of this conversion effort.

Ordering TEX82 for 4.1 bsd

TEX82 is now available for sites running Berkeley Unix, versions 4.1, 4.1c, and 4.2. We are now making available a "beta test" distribution which includes sources for TEX and WEB, libraries, fonts, and whatever device drivers we can obtain. At the time of writing, we have drivers for the Versatec, thanks to Carl Binding of our Department, for the Imagen laser printer, thanks to Pavel Curtis of Cornell, and for the Symbolics LGP-1 laser printer. If you have DVI 2 drivers for other devices, please send them to us and we'll happily include them in future distributions. The font bit maps included in the distribution are for devices with resolutions of 200 pixels/inch, 240 pixels/inch, and 300 pixels/inch.

This distribution will be the latest version of TEX82 available to us (presently 0.9999). Note that versions before 1.0 are considered to be pre-releases, and hence the "beta test" designation. TEX82 has been frozen, however, and the only changes made between the present version and Version 1.0 will be bug fixes. If you want to wait for Version 1.0, please let us know when you send in your order and we'll hold it. Version 1.0 may be out by the time this report is published.

As the implementation uses a modified version of Berkeley's **pc** Pascal compiler, we will only be able to provide a complete distribution to those sites with source licenses for 4.1 or 4.2 bsd. We will, however, try to accommodate those with 4.1 or 4.2 bsd binary licenses. Please make sure that you indicate clearly whether you have a source or a binary license.

The size of the distribution has increased to about 20 megabytes, due mostly to increases in the size of the fonts area. However, most sites can get away with using quite a bit less disk space since only those font magnifications which are actually used on the intended output device need be loaded.

Tapes will be in **tar** format, blocked 20, and written at 1600 bpi, unless otherwise specified (we can also write 800 bpi).

To order, send a check for $50 (U.S. Funds) made to the University of Washington, a copy of your 4.1 or 4.2 bsd license, and your address to:

Richard Furuta
Computer Science, FR-35
University of Washington
Seattle, WA 98195

We would appreciate it if foreign sites could increase the amount of their check as appropriate to pay the added postal costs necessary for mailing the tape. If you have a CSNet, Arpanet, or uucp mail address, please include it and we'll add you to our electronic mailing list.

Please note that it is not necessary for you to send any of your Western Electric Unix Licenses—only the 4.1 or 4.2 bsd license. But please *do* remember to include the 4.1 or 4.2 bsd license—we've had to write or call many sites asking for it which delays things considerably. Please do not send purchase orders as we have no facilities for handling them.

TEX82 on Other Versions of Unix

Several people have expressed interest in porting TEX to varieties of Unix other than 4.1 bsd. We have heard from people who either have begun ports or are contemplating beginning ports to Amdahl's UTS Operating System and to 2.8 bsd. We will keep *TUGboat* readers up to date on progress on these ports. However, since a certain amount of time passes between *TUGboat*s, I would like to ask those of you who are interested in one or another of these efforts to drop me a note. In this fashion, I'll be able to judge the amount of interest in each of these ports and I'll be able to send out announcements should they be completed before the next *TUGboat*.

If anyone is interested in attempting ports to other versions of Unix (say System III or V), we want to hear from you. We're quite anxious to see TEX become available on all versions of Unix and will try to help in whatever ways we can.

* * * * * * * * * * *

VAX/VMS

Monte C. Nichols
Sandia National Laboratory
Livermore, California

There's good news and not-so-good news to report for VAX/VMS. Through the good graces of David Fuchs (see page 72), TEX82 has been ported to VAX/VMS where it is up and running well. Unfortunately, there is no spooler available on the distribution tape at the present time. Hopefully, there will be soon.

The most recent version of TEX82 (.9999) for VAX/VMS is available from Maria Code (see page

129) in BACKUP format. Be sure to specify that you are on a VAX/VMS system. Hopefully, the .PXL files will also soon be available in BACKUP format.

Anyone with a spooler for TEX82 who is willing to have it put on the distribution tape, please call me. The interfaces listed in the table on page 71 are not on the distribution tape; information should be obtained from the sites listed.

* * * * * * * * * * *

REVISED VARIAN OUTPUT DRIVER IN VAX/VMS FORTRAN

Jim Mooney
West Virginia University

The VAX output driver for the Varian Plotter, written in VMS FORTRAN, has been updated to process Version 1 DVI files, and has received some other improvements as well. The first version of this program was reported in TUGboat, V2 No. 3, pp. 14–15.

The revised package was developed as a consulting project for Science Applications, Inc., McLean, VA, with the assistance of Buff Miner.

The new system consists of two programs, DVITOVAR and OUTTOVAR. DVITOVAR can produce raster files like its predecessor, which can be plotted separately by OUTTOVAR; alternately, it can drive the plotter directly.

Other improvements include better use of VMS logical names and command line parsing; an option to rotate the output 90 degrees; and control of magnification and page margins.

DVITOVAR accepts Version 1 DVI files. Alas, the stability of this version has been somewhat less than originally predicted. TEX has moved on to Version 2. A DVITOVAR for Version 2 is a possibility for the future.

The set of files comprising the DVITOVAR package has been provided to Oregon Software and will be included on future releases of their VAX/VMS TEX tapes.

A document describing the package is available from me on request. Technical questions may be addressed to me at:

Dept. of Statistics and Computer Science
West Virginia University
Morgantown, WV 26506
(304) 293-3607

Note: I cannot supply copies of the programs themselves, as I have no facility for making tapes. We do not even have a system capable of running TEX available to us at the present time.

* * * * * * * * * * *

Fonts

* * * * * * * * * * *

TEX FOR ARABIC SCRIPT

Pierre MacKay
University of Washington

The principal focus of our work with TEX at the University of Washington is the development of a capacity to typeset Arabic Script texts. For this we must extend and modify the basic program, but we are making every effort to ensure that our Arabic Script version of TEX will be an enhancement, and will leave all the basic features of TEX82 intact. The TRIP.TEX file will give us a good test of our success in this regard.

Nevertheless, our eventual DVI output will be distinct from that produced by other sites in the following ways:

1. The ID number will be 102. Our special driver programs will be set so that a driver for DVI102 will accept DVI2 files, but a standard driver will complain about an Arabic Script file. On the UNIX distribution tapes, incidentally, you will always get the standard DVI2 drivers unless you specifically ask for a DVI102 driver.
2. DVI codes 250 and 251, which are unimplemented in "vanilla" TEX, will be used to mark off right-to-left text in a horizontal list. These codes will be balanced at any completed level of \hbox{} before the surrounding \vbox{} is built. Code 250 marks the beginning or right-to-left text and code 251 marks the return to a normal TEX environment.
3. The " (double-quote) character is activated as a new primitive in TEX text input. It operates as a toggle in the way the $ (dollar-sign) math-mode toggle works. The environment between balanced pairs of " toggles is known as "reversed text mode" and may be useful for Hebrew text as well as for Arabic. Math-mode may be nested within reversed text mode and will be governed by the normal rules of math mode. A double quote character in math mode retains its normal TEX significance. If you want Arabic Script text inside math mode, you will have to \hbox{} it separately and to \unhbox{} it at the desired place.

None of this actually works yet, but we are getting very close. The TEX for Arabic Script project is supported in part by grants from Northern Telecom and BNR Inc.