MINUTES OF THE FIRST TUG MEETING

Robert Morris

Below follows what I heard at the first TEX Users Group meeting in Palo Alto, February 22, 1980. If I've abused the speakers I hope they will write to the newsletter to correct me. Occasionally I have inserted notes cross-referencing other speakers or sources. These are in italics with my initials, *...ram*, appended.

A rather broad spectrum of TEX expertise was assumed on the part of the listeners, even within the remarks of a given speaker. I have made no particular attempt to sort this out. Thus TEXperts will be bored by parts of some paragraphs and novices mystified by others. Sorry.

coffee(sic)

People introduced themselves. About 50 were in attendance. Miscellaneous remarks were made, all of which were amplified later except the location of the TEX and **METAFONT** files for those desiring to get them from SU-AI via the Arpanet. To do this, get the files FILES.INF[TEX,DEK] FILES.INF[MF,DEK] and FILES.INF[FNT,DEK] for TEX, **METAFONT**, and the SU TEX font files respectively. Each of these tells which files you must further import. In order to avoid pain induced by the SAIL character set, be sure to FTP them in ASCII mode. Failure to do this will cause grief at least about the right brace character, which is rather critical to TEX. Anyone without Arpanet access can get the SAIL sources described above by inquiring of

Dr. Luis Trabb Pardo
Dept. of Computer Science
Stanford University
Stanford CA 94305

Sources obtained in this way will not have the SAIL character set problem because they are generated on TOPS-20 at SCORE. At the moment, only SAIL sources are released. See below about Pascal.

SPEAKER: Donald Knuth,
Stanford University.

Don Knuth's opening remarks.

Don opened with some observations about the things he's learned from the TEX project. He did not anticipate the full power of TEX until he had used it extensively himself. For example, until he made macros to pretty-print Pascal programs, he had not thought much about text which was ragged on both left and right. And, early on he thought an interactive TEX would be useful, but finds now that TEX users internalize what TEX will do to such an extent that they usually know what TEX is going to do about their input and so have no pressing need to see it displayed on a screen immediately after the input is finished. Knuth's conversations with other typographical software users confirm this, e.g., with people setting classified ads in a newspaper.

TEX is rather stable now: No reported bugs from October to December, one in January, and one rumored yesterday. (*The SAIL version has been in use at Stanford since August 1978 and at other sites since September 1978. ...ram*)

Radical changes in TEX are to be discouraged, because they might destroy the stability of the system and the interchangeability of TEX input files. An attempt will be made to keep a single common denominator TEX in circulation. Don feels reliability is far more important than allowing everyone to add "missing features" to TEX. When the Pascal version is released there may be some last minute changes of this radical nature, but they should be in all released versions of TEX.

One flaw in this regard was the use of floating point arithmetic in TEX. The first Pascal release will use floating point, as does the SAIL version, with the attendant risk of machine dependencies due to rounding (e.g. in extreme cases, a paragraph might have 11 lines in one implementation and 10 in another if two line breakings had very similar "badness" value). The second Pascal release will use only fixed point arithmetic.

METAFONT.

METAFONT has a few dozen users. Don took 3–4 months to create about 60 fonts necessary to set his books. The eminent type designer Hermann Zapf visited, learned **METAFONT**, taught Don some type design, and helped beautify those fonts. The fonts are available for use with TEX. A Stanford C.S. Report "The Computer Modern Family of Fonts" is forthcoming. (*Don explained in a TUG Steering Committee meeting the next day, that one of the accomplishments of Zapf's visit was to write* **METAFONT** *programs incorporating some of Zapf's craft, especially techniques involving changes of pen pressure during letter construction. ...ram*)

Output devices.

The Alphatype CRS is fully functional under TEX. It sets about 3 inches/sec per baseline, requiring about 3–4 minutes per page of moderately complex math. Don rewrote the Alphatype internal software for TEX suitability when using **METAFONT** fonts.

In general, output on most devices is straightforward in principle. (*See report of David Fuchs' talk below. ...ram*) This includes known phototypesetters and even vector devices, although more software would be involved.

Audience requests.

What follows are Don's responses to questions from the audience about specific topics.

General Organization of TEX.

TEX consists of 5 modules:

(1) TEXSYS, about 10% of the code. Contains the storage management and error recovery routines. For complicated TEXing it uses around 30K words (36-bit), but for simple things as little as 8K has been used.

(2) TEXSYN, about 30% of the code. Processes macros, scans input tokens, and otherwise handles TEX syntax.

(3) TEXSEM, about 50% of the code. Responsible for switching modes (math, display, vertical, horizontal), contains the page and paragraph builders, math setting routines, rules. This module has as its main documentation the source code itself. (*But see report below on the Pascal version. ...ram*)

(4) TEXOUT, about 10% of the code. The output modules, responsible for recursively processing a list of boxes and making a list of output instructions for the device driver.

(5) TEXEXT, presently 0% of the code. The TEX extension module is the place the user defined extensions to TEX are placed. Examples of such will be supplied with the Pascal version, because an indexing and cross-referencing facility has been implemented that way for the time being. Also, researchers at XEROX have implemented color descriptions as an extension. Hooks are placed in TEXSYS to trap the TEXEXT. The feature is powerful and dangerous, providing the user with the ability to clobber any internal TEX data structure. (*Readers should not confuse this feature with the macro feature. The TEX extension feature is for the addition of things which TEX can not do, whereas the macro feature parameterizes or otherwise makes easy the things that require many or arcane TEX commands to do. Perhaps aside from the indexing feature, most TEX users will never encounter TEX extensions. ...ram*)

Indexing and document management features.

The initial problem with document management facilities was TEX's inability to make its activities known to anything but its output file. This was the difficulty of interfacing it to CMU Scribe, for example, or other software which needs to know what's going on while TEX is working. Although not all cases are covered, the TEXEXT solution indicated above seems to address many of the resulting problems. The index output thus generated may need some sorting or other trivial post-processing. (*See report below of Richard Zippel's MIT document preparation macro package for non-TEXEXT solution to some of these problems. ...ram*)

The macro facility.

The macro facility is designed to allow users who know nothing about typography to set papers well. With it, one source can produce many different output forms. This approach is similar to IBM's General Markup Language, and is being pursued by the American Mathematical Society.

Last minute changes.

A few are contemplated for inclusion in the Pascal version. The only one posing any incompatibility with existing TEX input files is the improved syntax of font definitions, which presently is similar to that for font changes, being distinguished by context. The new syntax would require existing input files to have their font definition statements changed. The ability to set equation numbers at the left may also be added.

TEX as a rough draft refinement tool.

TEX provides the same flexibility and ease of change that any text maintenance use of the computer does.

TEX as a general typesetting tool.

TEX was originally designed by Don to set the *Art of Computer Programming*. However, many of his solutions turned out to be appropriate to many other forms of typesetting. The line breaking and paragraph building mechanism is particularly successful, producing text with astonishingly few hyphenations. Widow avoidance is less successful: in mathematics, adding lead to a display is feasible to defeat widows, but in straight text non-uniform interline spacing is frowned upon. Rivers (continuous streams of white space dribbling down a page due to accidents of interword spacing) is not specifically addressed in TEX but seems to be a rare occurrence. Most of Don's effort in design was focused on mathematics spacing.

Chemical formulas and other scientific typesetting.

In principle, TEX should be good for most of this work. Don suggests that duplication of effort be avoided by discouraging exploration of non-math typesetting until the American Mathematical Society experience is greater. Some of their solutions should apply immediately to physics and chemistry publishing.

SPEAKER: Luis Trabb Pardo,
Stanford University.

The Pascal Version.

Luis sketched the problems associated with the Pascal project and what the final product will look like. The big question: when will it be released. The answer, not before it has been run successfully not only on the DECSystem 20, but also been transported to an altogether different architecture, the IBM 370. Both tasks are moving along and within sight of completion, but Luis is (*understandably! ...ram*) reluctant to name a date out of concern for being unable to meet any particular deadline. (*If you press the Stanford team privately they will admit that they hope release is months away. ...ram*) Both Knuth and Luis are greatly concerned that a high quality product as bug free as possible be produced. The requirement that the parallel 370 development be complete will help insure that no machine dependent traps will be sprung on those who attempt to move TEX. The installation of the 370 version will, moreover, be done by people who are not TEX wizards, thereby further simulating the environment released versions will run in. Installation on CDC, VAX, and Univac machines is also in development. Finding suitable Pascal compilers is a big stumbling block. Pascal does not seem to be amenable to large portable software projects because of many unspecified things even in "standard" Pascal.

The structure of the Pascal Version.

Pascal TEX as released by Stanford will consist of two pieces: the TEX module and the system dependent module. The former, comprising 90% of the code, should need almost no local tuning save for specifying a few parameters, such as word size. Further, this should be valid for an entire family, and not site specific. The system dependent module contains the code which allows TEX to communicate with the host operating system. It deals, for example, with the file system and the input character set. This is where most of the implementor's work will lie.

The Stanford team has been striving to make the Pascal TEX a model of transportable software. To this end, it is to be released in the form of a "documented description" and a support package. The documented description is a topdown description of how TEX works. The support package contains two programs for dealing with it. TEXDOC is a program which produces a typeset description of the fully commented TEX program. A paper copy of that will also be supplied. UNDOC is a program which converts the documented description into a Pascal program. Usually humans will never read this, studying instead the TEXDOCed version. The support package also includes some sample device driver software, a set of 200 dot/inch fonts and a sample TEXEXT extension.

What the implementor must do.

To implement TEX do this:

(1) First find a suitable compiler. (*Recipe for rabbit stew: first catch a rabbit! ...ram*) This is not entirely trivial because of the lack of Pascal standards. In fact it was the major delaying factor so far in the project. (*I assume a future issue of the newsletter will have an article detailing just what makes a compiler suitable. ...ram*)

(2) Redefine some of the data allocation policies and a few of the low level macros.

These tasks are constant for an entire architecture and need not be repeated by each site.

(3) Modify certain system dependent calls, typically the names of TEX calls to the host operating system. This may be quite site dependent. Luis estimated it might take one month of a system programmer's time for each operating system, but no special knowledge of TEX would be needed.

Resources.

How greedy is TEX? It occupies 75K 36-bit words on a DEC 20 and requires 2–3 CPU seconds per page to set mathematics. Something similar ought to hold on an IBM 370/138. Some investigations into making it work in smaller environments are taking place. Luis believes that such efforts should evolve in the following directions (*see the talk of Bob McClure below ...ram*):

(1) Overlay parts of the program and reduce the size of some of the data structures, e.g., reduce the number of fonts which can be handled at once (currently 64) and the complexity of math which can be set. Such efforts could result in 50% memory reduction.

(2) Make a multipass TEX. Natural separation: (a) macro expansion; (b) line breaking and box making; (c) page breaking and output. This might force abandoning some of the interaction between these tasks, e.g., macro definitions in output routines, but it would probably leave a highly useful TEX anyway.

(3) Memory management. For portability, TEX makes no assumptions about the ability of the host to manage memory, rather simply asking to be given a big block then left alone. An implementation which used the host's memory management might not need such a big chunk.

SPEAKER: Bob McClure, private consultant.

TEX in C.

Bob is working on a C version, using many of the smallifying techniques suggested above by Luis. The first target is a Z8000 system running version 7 UNIX, but the programming is not making use of UNIX features tempting as some of them might be. The aim is to have a TEX which will run in 48Kbytes of memory, thereby fitting on 16-bit machines. Heavy virtualizing of some of the tasks is envisioned. Roughly, the vast resources get stretched out in time and in space (on a disk). Bob isn't prepared to guess about speed of the resulting product which will be proprietary, but for sale at "a reasonable price". (*I hope a rumor mongering section of the newsletter will be devoted to others' efforts at making small* TEX, *or better still that those doing so will write brief descriptions for us. ...ram*)

SPEAKER: DAVID FUCHS, STANFORD UNIVERSITY.

Output devices.

David spoke on the structure of TEX output and the technical requirements of device drivers. Since he is devoting an article in this issue to it, I omit details. Roughly, implementors need to write a spooler and a program to convert the TEX output to a form suitable for the output device. This involves decisions about how much of the task to leave to the host, which in turn depends on the communications bandwidth between host and output device. The details thus depend on the host-device combination and may be somewhat site dependent.

SPEAKER: RICHARD ZIPPEL, MIT.

Richard spoke about a loosely organized document macro package available at MIT. It has many of the features one wants in document preparation and has some overlap with AMS-TEX. It is reported separately below.

SPEAKER: MIKE SPIVAK, PUBLISH OR PERISH PRESS.

AMS-TEX.

Mike is a developer of the AMS version of TEX and the author of "The Joy of TEX". The purpose of these two ventures is outlined by Mike elsewhere in this issue.

SPEAKER: DICK PALAIS, BRANDEIS UNIVERSITY.

The final session of the afternoon was chaired by Dick. It consisted of a discussion of existing, contemplated, and desired output devices and a discussion of the TEX Users Group.

Output Devices.

Presently running with TEX: Versatec 3211; Varian 9000; Alphatype CRS.

Under development or consideration: Canon laser printer.

Desired: Mergenthaler Linotron 202, 2000 and VIP; GSI C/A/T; APS-5; Compugraphic 8600 and UNISETTER; AMI; Sanders infinite matrix printer; vector devices in general.

Dick Friday (Digital Equipment) said he had experimented with driving a Diablo and found it not very difficult.

Rumors were mentioned about dramatically falling prices to be announced this June at the National Computer Conference. Others were mentioned that the internal coding of Mergenthaler devices were now sufficiently understood that they might be useful. (*The general problem with phototypesetters is finding out exactly what information they need so that* TEX *can provide it. ...ram*)

TUG organization.

A Steering Committee was elected by acclamation. Its first meeting is reported below. General discussion to guide it centered on the function of a users group. Among the roles proposed were the following (*When it met the next day, the Steering Committee took a much narrower view than the spectrum represented here. ...ram*): organize TEX research; organize "Birds of a Feather" sessions; organize a newsletter; supervise or perform TEX software support; distribute TEX; formulate a test suite to validate programs claiming to implement TEX.

Several people suggested seeking low cost commercial support along the lines of IMSL. Most agreed that an institutional membership fee on the order of $100 would be acceptable. (*The Steering Committee subsequently decided that such a fee is premature. ...ram*) It was generally agreed that the Users Group would initially be concerned with short term problems so no attempt need be made to find solutions which would be cast in stone. Dick Palais told us that the AMS is applying for trademark protection of the TEX logo and would thus keep administrative control over what could be called TEX.

* * * * * * * * * *

REPORT OF THE STEERING COMMITTEE MEETING OF FEBRUARY 23, 1980

The Committee met jointly with the AMS Font Subcommittee of the Committee on Composition Technology.

Richard Palais and Robert Morris agreed to be chair and secretary pro tem.

Robert Welland agreed to edit the newsletter. The first newsletter will have a report of the meeting and will be distributed free by the AMS upon inquiry about TEX. Subsequent newsletters will be by subscription only.

There will not be institutional membership until TUG has to play a larger role than circulation of information. Individual membership will be $10 annually and will cover newsletter expenses. All memberships and fees will be reconsidered at the next Steering Committee meeting.

Richard Friday will work up a proposal for a validation suite for programs whose authors desire to call them TeX. Presumably, the Users Group would pass recommendations to the AMS Board of Trustees about a given request to use the TeX logo.

The role of the Users Group in distribution will be re-examined after the Pascal release is made. In about six months the Steering Committee will decide when to meet again. At this time it is hoped that Pascal distribution will be under way from Stanford and alternate development sites.

Respectfully submitted,

Robert Morris,
UMASS/Boston

* * * * * * * * * *

Interface Software

* * * * * * * * * *

THE STATUS OF THE PASCAL IMPLEMENTATION OF TeX

September 9, 1980

Ignacio Zabala
Luis Trabb-Pardo

This document (PTEX.TXT[TEX,IAZ]%SAIL) is intended as a public, up-to-date report on the status of the PASCAL implementation of TeX. A file PTEX.BBD[TEX,IAZ]%SAIL is maintained that contains all the mail and new events related to PTEX.

System organization:

The TeX-PASCAL system consists mainly of three modules:

- the TEXPRE module implements the preprocessor that generates the data structures employed by TEX.
- the SYSDEP module contains routines that are very much dependent on the particular host system. It is used both by TEXPRE and TEX.
- the main TEX module.

Compatibilities and incompatibilities:

Neither TEX nor TEXPRE should need modification at any installation, but, surely, SYSDEP must be adjusted for each host site. The three modules are programmed in PASCAL though some installations may find it convenient to reprogram SYSDEP in assembler language for the sake of execution speed.

As the default case in the CASE statement is a non-standard feature of PASCAL, it has been given different names by different compilers. Compile time initialization is not standard either, but available under different names in most compilers. These are the only reasons why TEX and TEXPRE may have to be modified, and the modifications are straightforward.

PTEX installations:

TeX-PASCAL has been running in the PDP-10 (SAIL) machine in our CS Dept., here at Stanford, since April 1980. Since then, all changes made by Don Knuth in the SAIL program have been incorporated immediately into the PASCAL program, which has also undergone modifications as more information was obtained on the characteristics of widely used PASCAL compilers. The program has been in-house tested. It has already processed the whole TeX manual and several chapters of Knuth's *Art of Computer Programming.*

Stanford's CIT has an IBM machine of the 370 family. Eagle Berns is in charge of the installation of TeX there. He has obtained a copy of the new IBM PASCAL VS compiler, and has tried to compile PTEX with it. As of September 9, both TEXPRE and SYSDEP (in PASCAL, only slightly modified) had compiled and run successfully to generate the table of data structures employed by the main TEX module. For these trials, Berns used 36 bpw font information files from SAIL. There was a problem with two procedures in the main TEX module which were still too large for the VS compiler. This module has been modified accordingly and we are waiting for feedback from Berns.

Charles Lawson is installing PTEX on the UNIVAC 1100 of the Jet Propulsion Lab at Caltech using the University of Wisconsin PASCAL compiler. He has made a good programming effort to get PTEX up. (For instance, he has coded the "environment" modules required by his compiler.) This installation seems to be already past the compilation phase.

At the University of Minnesota, Mike Frisch is installing PTEX on a CDC-Cyber.

George Otto is in charge of the installation at Wharton. The Moore school has a UNIVAC 90-VS/9 where they use the PASCAL-8000 compiler.

David Kashtan has compiled everything successfully on the VAX (VMS) at SRI.

Richard Friday has also compiled everything on a VAX (UNIX) at DEC.