The Design of \LaTeX{} and METAFONT: A Retrospective

Nelson H. F. Beebe

Department of Mathematics
University of Utah
Salt Lake City, UT 84112-0090
USA
Prehistory (1452–1970)

- 500-year-long tradition of typesetting
- Expert human typographers with decades of experience
- Hand setting of type in lines and racks
- Letters stored into upper and lower cases (bins)
- Hot-lead process
- Proprietary handmade punch-cut fonts
- Typesetting on spread of two facing pages
- Publishers have editors and proofreaders
- Typesetting and book binding done by job shops
Typesetting on computers (1970–)

- expert human typographers, but now hampered by technology
- typographically substandard quality
- expensive and proprietary typesetting computer hardware and software
- optical font scaling
- proprietary optical fonts


see DEK’s *Digital Typography* (1999)
Knuth’s sabbatical year (1977–1978)

- improve typesetting of *The Art of Computer Programming* books
  
  I didn’t know what to do. I had spent 15 years writing those books, but if they were going to look awful I didn’t want to write any more. How could I be proud of such a product? — DEK (1996 Kyoto Prize address)

- reproduce look of *Linotype Modern 8a* fonts of earlier editions

- 0.x-MIPS departmental computers (notably, 16-bit PDP-11 and 36-bit PDP-10)

- computer use still cost $$$$ for many people
Computers in 1977

- mainframes: IBM and the BUNCH (Burroughs, UNIVAC, NCR, CDC, and Honeywell), clones (Amdahl, Russian ES, A Fujitsu, Hitachi, NEC, RCA, Siemens, Wang), ICL, Phillips, Texas Instruments, Xerox

- minicomputers: Data General, DEC PDP-\( n \), GE, Harris, Interdata, Perkin-Elmer, Prime, SDS, Varian, ...

- Xerox PARC: first workstations

- microcomputers based on Intel 8080, MOS 6502, Texas Instruments TMS1000, Zilog Z80, ...
PDP-10 computers

DEC PDP-10 ran several different operating systems:

- BBN TENEX
- Berkeley’s PROJECT GENIE SDS 940
- COMPUSERVE 4S72
- DEC TOPS-10
- DEC TOPS-20
- MIT ITS (Incompatible Time Sharing System)
- Stanford WAITS (Westcoast Alternative to ITS)
- TYMSHARE AUGUST and TYMCOM-X
PDP-10 contributions

PDP-10 systems hosted many important developments:

- **ETHERNET**, TCP/IP, and ARPANET backbone [SRI, UCB, UCLA, UCSB, Utah]
- Brian Reid’s document-formatting and bibliographic system, **SCRIBE** [CMU]
- Richard Stallman’s **EMACS** editor [MIT]
- Ralph Gorin’s **SPELL** [Stanford]
- Mark Crispin’s mail client, **MM** [Stanford]
- Frank da Cruz’s **KERMIT** [Columbia]
- Bill Gates and Paul Allen simulate Intel 8080 to develop MS-DOS
PDP-10 programming languages

- **Algol 60**
- **Basic**
- **Bliss** [DEC and Carnegie-Mellon University (CMU)]
- **Cobol 74**
- **C** (early 1983)
- **Forth**
- **Fortran 66 and Fortran 77**
- several dialects of **Lisp**, including **MacLisp** [MIT], **Interlisp** [BBN and Xerox], and **PSL** (Portable Standard Lisp) [Utah]
PDP-10 programming languages (cont.)

- MACRO, MIDAS, and FAIL assemblers
- MACSYMA [MIT], REDUCE [Utah] and MAPLE [Waterloo]
- PASCAL (mid-1982)
- shell-scripting language PCL (Programmable Command Language) [DEC and CMU] (early 1980s)
- SAIL (Stanford Artificial Intelligence Language) [ALGOL 60 with zillions of extensions]
- SIMULA 67
- SNOBOL
PDP-10 editors

- TECO (Text Editor and Corrector) [DEC]
  The most powerful and dangerous programming language and text editor ever invented. . . . advanced TECO addiction has been known to cause nightmares about infinite loops four characters long. . . . Not recommended for use via modem connections in bad weather, since at first glance many TECO programs are indistinguishable from line noise.

- TV (screen editor derived from TECO) [DEC]

- E (WAITS): with TV, DEK’s editor until his switch to EMACS and UNIX about 1990

- EDIT [DEC]

- EMACS (EDitor MACroS) [built on TECO] [MIT]
PDP-10 document-formatting systems

- DIGITAL STANDARD RUNOFF [TEX later used as a backend for VAX VMS manuals]
- Brian Reid's SCRIBE [model for LATEX and BIBTEX, but licensed and proprietary] [CMU]
PDP-10 architecture

- large, but clean, instruction set
- 744 instructions, augmented at XEROX PARC with 472 9-bit instructions for INTERLISP
- 36-bit words [octal notation: 7777777„765432]
- 18-bit address (262,144 words, 1.25MB), later extended to 30-bit (5GB), but only 23-bit addresses ever implemented in hardware (8,388,608 words, 40MB)
- external symbols stored in RADIX50 encoding with characters [A–Z0–9% .$] [4 bits of flags, 32 bits with six characters: $2^{32} > 40^6$ and $40_{10} = 50_8$
- bytes of any size from 1 to 36 (thus, efficient access to packed fields in records and structures)
PDP-10 architecture (cont.)

- Filesystem records byte count and byte size

`@vdir hello.*`

```
TOPS20:<BEEBE.C>
HELLO.C.1;P777700  1  99(7)  12-Jan-2005  07:09:41 BEEBE
.FAI.2;P777700    1 1870(7) 12-Jun-2005  08:11:40 BEEBE
.PRE.2;P777700    1  12(7)  12-Jun-2005  08:11:40 BEEBE
.REL.1;P777700    1 113(36) 12-Jun-2005  08:11:16 BEEBE
```

Total of 4 pages in 4 files

- Text files normally 7-bit ASCII, with low-order bit set to 1 to mark a line number in EDIT files
- 8-bit bytes allow sharing files with UNIX via NFS
PDP-10 architecture (cont.)

- largest signed integer: \(2^{35} - 1 = 34,359,738,367\)
- single-precision floating-point precision: 27 bits (8D)
- double-precision floating-point precision: 62 bits (18D)
- floating-point range: \(1.17e-38 \ldots 1.70e+38\)
- much later: UTF-9 and UTF-18 Unicode support
PDP-10 architecture (cont.)

- stack-based architecture (thus, recursion trivial)
- clean system call interface (JSYS)
- set trap jsys /all
- DDT (Dynamic Debugging Tool) sits in high address space and can debug any program written in any programming language
- DDT is the default command processor on MIT ITS
TOPS-20 features

- MONITOR (kernel) and EXEC (command processor) programmed in efficient assembly language
- supports 50 to 100 simultaneous users on terminal connections, thanks to PDP-11 front end
- command-line help

@? Command, one of the following:

ACCESS     ADVISE     APPEND
ARCHIVE    ASSIGN     ATTACH
BACKSPACE  BLANK      BREAK

... UNATTACH UNDECLARE UNDELETE
UNKEEP     UNLOAD     UNMAP
VDIRECTORY WDIRECTORY
TOPS-20 features (cont.)

- command-line completion and prompt [KERMIT & MM]

@comPILE (FROM) ? confirm with carriage return
or one of the following:
/10–BLISS   /36–BLISS   /68–COBOL
/74–COBOL   /ABORT      /ALGOL
...
/RELOCATABLE /SAIL       /SEARCH
/SIMULA      /SNOBOL     /STAY
/SYMBOLS     /WARNINGS

- tree-structured file system PS:<BEEBE.MF.CM>

- file ownership; 18-bit protection code (user, group, other)

- append, delete, execute, list, read, write access bits
TOPS-20 features (cont.)

- case-insensitive filenames
- Ctrl-V quotes special characters in filenames
- optional quotas in directories
- file generation numbers

@vDIRECTORY (VERBOSE, OF FILES) pdp10.c.*
  TOPS20:<BEEBE.HOC36>
  PDP10.C.3;P777752  8 19892(7) 21-Jan-2005 09:03:35 BEEBE
  .4;P777752  8 19897(7) 21-Jan-2005 10:38:55 BEEBE
  .5;P777752  8 19899(7) 21-Jan-2005 10:52:40 BEEBE

- tape archives with online directory entries
- DELETE, UNDELETE, and EXPUNGE
- ATTACH and DETACH
TOPS-20 features (cont.)

- user and system logical names
  
  @define TEXINPUTS: TEXINPUTS:,
  ps:<jones.tex.inputs>
  $^Edefine TEXINPUTS: ps:<tex.inputs>,
  ps:<tex.new>

- search path support built-in to MONITOR, so all programs and programming languages can use it

  @iNFORMATION (ABOUT) lOGICAL–NAMES (OF) sys:
  Job-wide:
  sys: => SYS:,TEX:
  System-wide:
  sys: => PS:<SUBSYS>,DOMAIN:,UNS:,SAI:,FUN:,HLP:,DSK:
Choosing a programming language

- assembly code tedious, would not survive hardware
- BLISS expensive and tied to DEC systems
- C not yet available
- COBOL awful: \texttt{MULTIPLY A BY B GIVING C}.
- FORTRAN most portable, but no recursion, no data structures beyond arrays, no low-level byte I/O, no decent character string support, six-character names
- LISP great, but inefficient and Babel of dialects
- PASCAL first available in mid-1982
- SAIL won
internal saf string array fname[0:2] # file name, extension, and directory;
internal simp procedure scanfilename # sets up fname[0:2];
begin integer j,c;
fname[0]_fname[1]_fname[2]_null;
j_0;
while curbuf and chartype[curbuf]=space
  do c_lop(curbuf);
loop begin c_chartype[curbuf];
case c of begin
  [pnt] j_1;
  [lbrack] j_2;
  [comma][wxy][rbrack][digit][letter];
else done
  end;
fname[j]_fname[j]&lop(curbuf);
end;
end;
SAIL conditional compilation

# changed to ^P^Q when debugging METAFONT;
define DEBUGONLY = ^Pcomment^Q

...# used when an array is believed to require
# no bounds checks;
define saf = ^Psafe^Q

# used when SAIL can save time implementing
# this procedure;
define simp = ^Psimple^Q

# when debugging, belief turns to disbelief;
DEBUGONLY redefine saf = ^P^Q

# and simplicity dies too;
DEBUGONLY redefine simp = ^P^Q
### Stanford extended ASCII character set

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>·</td>
</tr>
<tr>
<td>001</td>
<td>↓</td>
</tr>
<tr>
<td>002</td>
<td>α</td>
</tr>
<tr>
<td>003</td>
<td>β</td>
</tr>
<tr>
<td>004</td>
<td>∧</td>
</tr>
<tr>
<td>005</td>
<td>¬</td>
</tr>
<tr>
<td>006</td>
<td>ε</td>
</tr>
<tr>
<td>007</td>
<td>π</td>
</tr>
<tr>
<td>010</td>
<td>λ</td>
</tr>
<tr>
<td>011</td>
<td>γ</td>
</tr>
<tr>
<td>012</td>
<td>δ</td>
</tr>
<tr>
<td>013</td>
<td>∫</td>
</tr>
<tr>
<td>014</td>
<td>±</td>
</tr>
<tr>
<td>015</td>
<td>⊕</td>
</tr>
<tr>
<td>016</td>
<td>∞</td>
</tr>
<tr>
<td>017</td>
<td>∇</td>
</tr>
<tr>
<td>020</td>
<td>⊂</td>
</tr>
<tr>
<td>021</td>
<td>⊃</td>
</tr>
<tr>
<td>022</td>
<td>∩</td>
</tr>
<tr>
<td>023</td>
<td>∪</td>
</tr>
<tr>
<td>024</td>
<td>∀</td>
</tr>
<tr>
<td>025</td>
<td>∃</td>
</tr>
<tr>
<td>026</td>
<td>⊗</td>
</tr>
<tr>
<td>027</td>
<td>↔</td>
</tr>
<tr>
<td>030</td>
<td>_</td>
</tr>
<tr>
<td>031</td>
<td>→</td>
</tr>
<tr>
<td>032</td>
<td>~</td>
</tr>
<tr>
<td>033</td>
<td>≠</td>
</tr>
<tr>
<td>034</td>
<td>≤</td>
</tr>
<tr>
<td>035</td>
<td>≥</td>
</tr>
<tr>
<td>036</td>
<td>≡</td>
</tr>
<tr>
<td>037</td>
<td>∨</td>
</tr>
<tr>
<td>040–135</td>
<td>as in standard ASCII</td>
</tr>
<tr>
<td>136</td>
<td>↑</td>
</tr>
<tr>
<td>137</td>
<td>←</td>
</tr>
<tr>
<td>140–174</td>
<td>as in standard ASCII</td>
</tr>
<tr>
<td>175</td>
<td>◊</td>
</tr>
<tr>
<td>176</td>
<td>}</td>
</tr>
<tr>
<td>177</td>
<td>^</td>
</tr>
</tbody>
</table>
SAIL limits affect METAFONT

- 19 buffers for disk files
- no more than 150 characters/line
- initialization handled by a separate program module to save memory (INIMF, INITEX, VIRMF, and VIRTEX)
- bias of 4 added to case statement index to avoid illegal negative cases
- character raster allocated dynamically to avoid 128K-word limit on core image
- magic TENEX-dependent code to allocate buffers between the METAFONT code and the SAIL disk buffers because there is all this nifty core sitting up in the high seg... that is just begging to be used
PDP-10 address space affects \TeX

<table>
<thead>
<tr>
<th>Table</th>
<th>1984</th>
<th>2004</th>
<th>Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>strings</td>
<td>1819</td>
<td>98002</td>
<td>53.9</td>
</tr>
<tr>
<td>string characters</td>
<td>9287</td>
<td>1221682</td>
<td>131.5</td>
</tr>
<tr>
<td>memory words</td>
<td>3001</td>
<td>1500022</td>
<td>499.8</td>
</tr>
<tr>
<td>control sequences</td>
<td>2100</td>
<td>60000</td>
<td>28.6</td>
</tr>
<tr>
<td>font info words</td>
<td>20000</td>
<td>1000000</td>
<td>50.0</td>
</tr>
<tr>
<td>fonts</td>
<td>75</td>
<td>2000</td>
<td>26.7</td>
</tr>
<tr>
<td>hyphen. exceptions</td>
<td>307</td>
<td>1000</td>
<td>3.3</td>
</tr>
<tr>
<td>stack positions (i)</td>
<td>200</td>
<td>5000</td>
<td>25.0</td>
</tr>
<tr>
<td>stack positions (n)</td>
<td>40</td>
<td>500</td>
<td>12.5</td>
</tr>
<tr>
<td>stack positions (p)</td>
<td>60</td>
<td>6000</td>
<td>100.0</td>
</tr>
<tr>
<td>stack positions (b)</td>
<td>500b</td>
<td>200000</td>
<td>400.0</td>
</tr>
<tr>
<td>stack positions (s)</td>
<td>600</td>
<td>40000</td>
<td>66.7</td>
</tr>
</tbody>
</table>
PDP-10 address space and \TeX

- compact table storage with limit number of indexing bits
- table sizes determined at compile time (fixed in 1990s)
- font and DVI files: compact, and complex, binary format
- roman and Greek letters crammed into text fonts
- Computer Modern fonts designed with only 128 glyphs in a font
- although 256 characters/font, only 16 different widths and heights, one of which must be zero
- hundreds of text fonts, but only 16 math fonts
- before 1989, only one preloaded hyphenation table
fixed-length buffer limits input line length

trip and trap tests apply only to \texttt{initex} and \texttt{initex}, not \texttt{virtex} and \texttt{virtex}, which are compiled separately and used untested as \TeX\ and METAPOST

word boundaries known to \TeX, but not recorded in DVI file

cryptic error messages: \textit{you can't do that in horizontal mode!}
Reimplement \TeX\ and \METAFONT

- increasing interest by user community

- \textit{American Mathematical Society} needs archival, extensible, low-cost, portable, reliable, solid, and very-long-lasting, typesetting and font design systems that authors can use too

- typesetting of many technical documents by different authors on PDP-10s exposes design deficiencies and font infelicities of SAIL-coded \TeX\78 and \METAFONT78

- wider use outside PDP-10 world needs a more portable implementation language

- coding must be of superb quality, and published for anyone to read, use, and reuse
Switching languages: 1980–1982

- C still not available
- MAINSAIL (MAchine INdependent SAIL) (1979) had not been ported much, and was commercial product
- **PASCAL** has many flaws

  **PASCAL**, at least in its standard form, is just plain not suitable for serious programming. . . . This botch [confusion of size and type] is the biggest single problem in **PASCAL**. . . . I feel that it is a mistake to use **PASCAL** for anything much beyond its original target. In its pure form, **PASCAL** is a toy language, suitable for teaching but not for real programming.

  — Brian Kernighan: *Why PASCAL is not my favorite programming language* (1981)
Switching languages (cont.)

- **PASCAL** language is small and available on several other systems, and thus, only viable choice
  - write in subset of **PASCAL**, avoiding awkward parts (fixed-length strings, poor I/O, nested procedures, useless sets, dynamic memory allocation without freeing on some systems)
  - hide the mess with **TANGLE** and **WEAVE** preprocessors
  - use *literate programming*: interleaved fragments of prose and code, with automatically-generated name indexes: see DEK’s *TEX: The Program*, **METAFONT: The Program** (1986), and *Literate Programming* (1992)

- **TEX** and **METAFONT** (20K lines each) were *severe* stress tests for almost all **PASCAL** compilers
Filename scanning in PASCAL

PROCEDURE Scanfilename;
LABEL 30;
BEGIN
beginname;
WHILE buffer[curinput.locfield] = 32 DO
curinput.locfield := curinput.locfield+1;
WHILE true DO
BEGIN
IF (buffer[curinput.locfield] = 59) OR
(buffer[curinput.locfield] = 37) THEN
GOTO 30;
IF NOT morename(buffer[curinput.locfield])
THEN GOTO 30;
curinput.locfield := curinput.locfield+1;
END;
30:
endname;
END;
TEX and METAFONT ports

- Thea Hodge ports early \TeX\ in \texttt{PASCAL} to CDC Cyber (1980)
- Monte Nichols: VAX VMS (1981)
- Lance Carnes and David Fuchs independently port \TeX\ and METAFONT in \texttt{PASCAL} to 16-bit \texttt{INTEL 8086} on IBM PC (1981–1982)
- Sao Khai Mong translates METAFONT from SAIL to \texttt{FORTRAN} for HARRIS systems (1982)
- Irene Bunner and John Johnson: HP-1000 (1983)
- Susan Plass: IBM mainframe (EBCDIC charset) (1983)
TEX and METAFONT ports (cont.)

- others: PDP-11, Z8000, APOLLO, M68000 (1983)

- Bart Childs brings \TeX{} to DATA GENERAL (1983), PRIME (1984), and CRAY supercomputer (1988)

- Pavel Curtis and Howard Trickey spend months patching UNIX PASCAL compiler to finally get \TeX{} and METAFONT on Berkeley UNIX (1983)

- Pierre Mackay and Rick Furuta make complete UNIX distribution of \TeX{} and METAFONT (1983)

- Barry Smith and David Kellerman, PASCAL compiler developers at OREGON SOFTWARE, bring \TeX{} and METAFONT to VAX VMS and new APPLE MACINTOSH (1984)
TEX and METAFONT ports (cont.)

- Pat Monardo at Berkeley produces COMMON TEX, a translation of TEX from PASCAL to C (1986–87)
- Klaus Guntermann: ATARI ST (1987)
- WEB2C community project now source of TEXlive and most other TEX implementations
Thanks to 664 TUGboat authors

14 Poppelier 21 Clark 21 MacKay 21 Rahtz 39 Childs 9 Braams 9 Furuta 9 12 Carnes 12 Damrau 12 Whidden 112 Beeton 5 Berry 5 DeCorte 5 Dietsche 5 Doob 5 3 Appelt 3 Bartlett 3 Beck 3 Cameron 3 DeLand 3 Doumont 3 Dunn 3 Fries 3 Fox 3 Girou 3 Glendown 3 Greene 3 Grätzer 3
27 Goossens 5 Berry 5 DeCorte 5 Dietsche 5 Doob 5 39 Childs 9 Braams 9 Furuta 9 12 Carnes 12 Damrau 12 Whidden 112 Beeton 5 Berry 5 DeCorte 5 Dietsche 5 Doob 5
28 Bzyl 28 Downes 28 Guenther 28 Lammarsch 28 Morris 28 Nichols 28 Schöpf 28 Whitney 7 Berdnikov 7 Carlisle 7 Crawford 7 Fox 7 Jackowski 7 Lamport 7 Milligan 7 Plaise 7 von Bechtolsheim 7 Wonneberger 7 Yap 6 Abbott 6 Ferguson 6 Hendrickson 6 Kellerman 6 Kelly 6 Ogawa 6 4 Adams 4 Alexander 4 Barnhart 4

Practical \TeX\ 2005 talk... – p.35/36
The End

THE BEATLES
JULY/AUGUST 1969