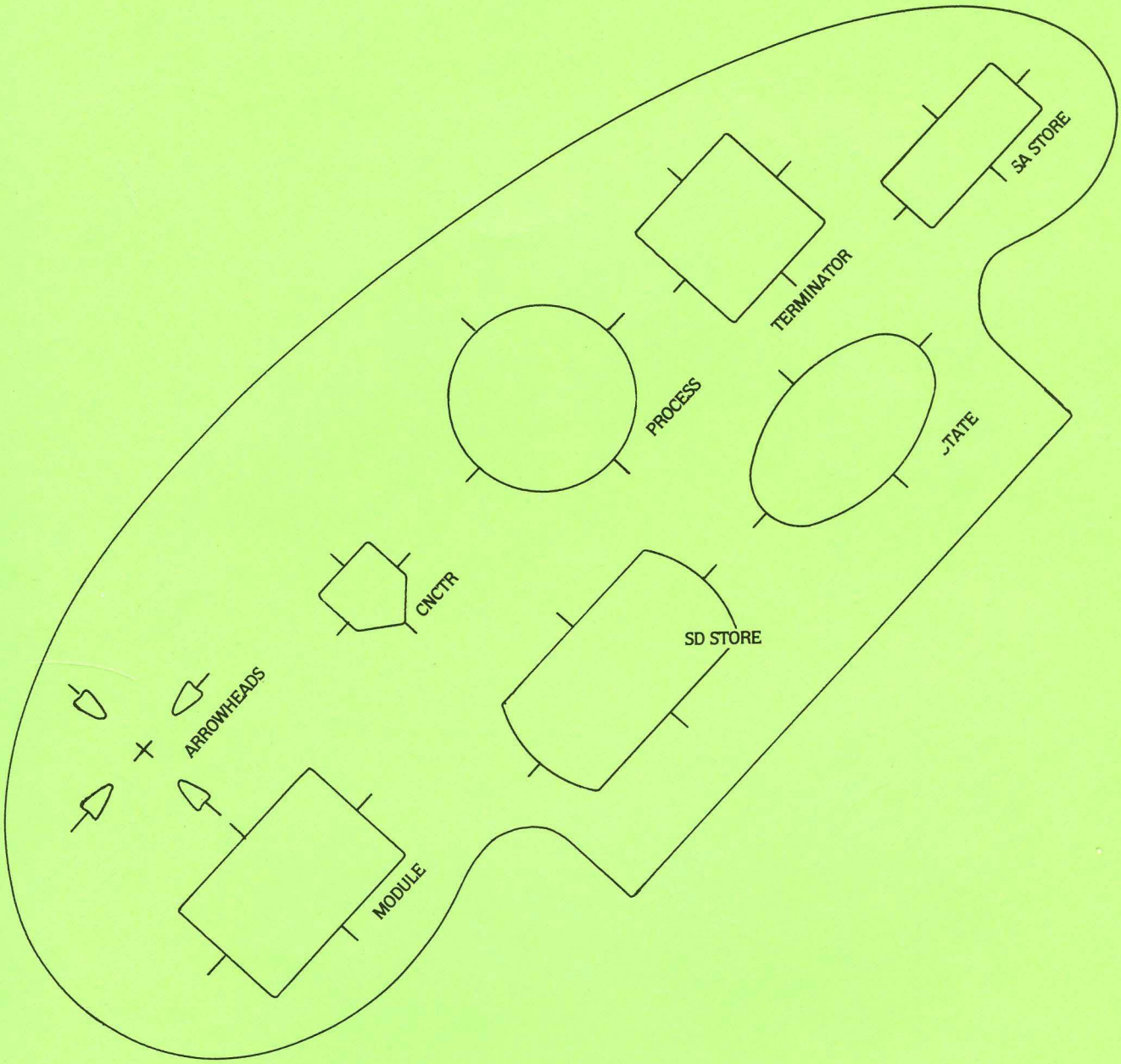
"THE HEAP"

DECUS

JULY 1985 ISSUE



LANGUAGES AND TOOLS SIG

Table of Contents

(*) ADA is a trademark of the Department of Defense

Editor's Notes

This issue of "The Heap" contains numerous small items, which I hope will be of some interest to you. As usual, I have included some items from the last symposium, in New Orleans, for those of you who couldn't get there. These include VAX Ada field test reports, and the slides from a talk on TPU programming techniques.

Also in this issue is a list of errors from the TeXBook, second edition. If you are a user of TeX, this should prove valuable, and if you're not a user of TeX, you should be.

Kathy Hornback, our esteemed SIG Chair, has submitted an article summarizing the results of the last Wishlist survey, and DEC's response to the top vote getters. Along with this, Alan Rizzuto has submitted the form for starting the next iteration of the wishlist survey. Alan has kindly taken over this function from me, I hope you will help him with the process by submitting items. I know he will be more organized about the whole affair than I was.

Finally, I have included a number of letters which may be of interest to our membership, including a response to the Fortran 8X material. That material was probably our hottest topic in New Orleans; watch this space for future developments. In particular, we will probably be including a questionnaire sometime soon; at the request of the Fortran 8X standards group.

This will probably be the final issue of "The Heap" as you now know it. The DECUS Management Council has voted to combine the various SIG newsletters into a combined publication. I and several others have been adamantly opposed to this, but the old saying about fighting City Hall holds true. There comes a time when you've got to admit you've lost, and make the best of the situation. If all the rosy predictions for the combined newsletter come true, you should be better served in the future, so lets hope for the best. In any case, I hope that over the past year I have provided you with useful and interesting information. I will continue to try to do so under the new format also. As usual, I welcome any comments or suggestions. My address is:

Alan L. Folsom, Jr.
Dept 431, Fischer & Porter Co.
E. County Line Road
Warminster, Pa. 18974

AI Note

Special SIGART Issue on AI in Engineering

The SIGART newsletter, a quarterly publication of the ACM special interest group on Artificial Intelligence, continues to be an excellent source for information on "Real World" AI projects. The April issue (No. 92), contains a ninety page survey of the present status of Artificial Intelligence as applied to engineering. The authors of the survey are D. Sriram and R. Joobbani of CMU. Dr. Sriram was the compiler of the extensive AI bibliography which was published in the last issue of "The Heap". Quoting from their introduction,

"The papers in this special issue were compiled from responses to the annouucement in the July 1984 issue of the SIGART newsletter and notices posted over the ARPAnet. The interest being shown in this area is reflected in the sixty papers received from over six countries."

Expert systems form the largest group of systems in the survey, but causal knowledge representation and natural language processing are also surveyed. There is a section describing tools for the applicattion of AI to engineering, as well as a special section devoted to groups developing more than one application. Here is the complete list of application areas covered:

Causality
Control
Databases/Management
Design (this was the largest group, reporting on 23 design systems)
Diagnosis
Education
Groups involved in several projects
Interpretation
Manufacturing
Miscellaneous
Natural Language
Repair/Maintenance
Simulation
Tools

The authors promise a future update to the survey, and draw attention to the "First International Conference on Applications of Artificial Intelligence to Engineering Problems" which will be held April 15-18, 1986, at Southampton University, England. For more information about the conference, contact:

D. Sriram, Technical Chairman First AIEP
Civil Engineering
  and Construction Robotics Laboratory
Department of Civil Engineering
Carnegie-Mellon University
Pittsburgh, PA 15213
sriram@cmu-ri-cive.arpa

Working ... .

kms fusion inc. A Subsidiary of KMS Industries, Inc.
3621 South State Road, P.O. Box 1567, Ann Arbor, Michigan 48106-1567 313/769-8500

June 4, 1985

The L&T Newsletter
C/O Al Folsom
Fisher and Porter
East County Lane Road
Warminster, Pa 18974

Dear Al:

VAXintosh Futures

James Downward
VAXintosh Working Group
C/O KMS Fusion, Inc
P.O. Box 1567
Ann Arbor, Mich. 48106
(313)-769-8500

At the Spring DECUS meeting the VAX SIG and the Languages and Tools SIG seem to have jointly formed a VAXintosh Working Group. Yes, you heard right, a "VAXintosh". Not surprisingly, there is a lot of interest among the DECUS membership in providing MACINTOSH like functionality for VAX systems and DEC systems in general.

At the "Just a Modest Proposal" session on Monday night, I presented a proposal that DEC needed a VAXintosh-like machine. The proposal was well received by the VMS community. That night I talked long into the morning hours with the software engineer in charge of the DEC user/machine interface for the VAXstation. It was quite clear from this conversation that DEC was very interested in the issue and wanted significant user input to help define what the needs, requirements, and market place for such product were.

Consequently on Wednesday, A BOF session on "VAXintosh Futures" was held with close to 100 attendees. It was soon realized that a VAXintosh represents a rather complicated, multi-faceted issue. Consequently, we propose to develop a working paper for presentation to DEC outlining in some detail what we mean by a VAXintosh. Topics which must be addressed include both hardware and software requirements and the general philosophy for the user/machine interface. Topics which we feel need to be addressed include:

- Market segments to be addressed by the VAXintosh. Why is it needed? What users would benefit most from a VAXintosh and why?

- Specific hardware requirements. Would just a cheap desk-VAX do, or is more needed? This includes graphic resolution, graphic display speed, ergonomics, input device (mouse?), local intelligence, and computer port I/O speed.

- User/Machine Interface. Is a MAC-like, icon driven interface what is needed? If so, must it mimic all of DCL or just a subset?

- User software development tools for a VAXintosh. Is an extension of SMG adequate to provide the functionality needed? Is a "Quick-Draw"-like interface needed. What about a special CLI to map MAC-like icon commands into DCL commands, or possibly extensions of the CDU to include ICON commands so that utilities could work both from an icon driven interface or from a DCL interface?

- DEC-supplied VAXintosh software. Are DEC equivalents of software packages like MACwrite, MACdraw, etc. needed for the VAXintosh or will user developed software suffice (if the interface and development tools exist)?

- MAC-Specific issues. Should DEC have a DECnet-MAC software product (they have a DECnet/MS-DOS product) or an interconnect to AppleTalk? Should DEC offer additional support for the MACINTOSH as a VAX terminal?

As you can see the issues are rather complex and we have a significant amount of work to do if we are to address these issues adequately to make an impact on DEC within the appropriate time window. Numerous attendees at the BOF session, have volunteered to write mini-working papers on one or more of the above subjects. I have volunteered to gather these papers and try to hammer them into a coherent whole (throwing in my own ideas also, of course).

However, we don't want to limit the input to this effort to the people who attended the BOF. Input from as many sources and points of view as possible is needed. Consequently, if you are interested in "VAXintosh Futures", write down your thoughts, proposals, or design goals and send them to me.

If we get enough input and support from the VMS community, by Fall, the VAX and Languages and Tools SIGs can jointly present DEC with a detailed working paper proposing what we would want in a VAXintosh. Currently, we have some support within DEC for a VAXintosh in some form. They have a "gut level" feeling that something is needed but are not sure quite what. This gives the VMS community a significant "window of opportunity" to influence DEC software and hardware product management when it counts.

Sincerely,

James Downward

AMAX ENGINEERING AND MANAGEMENT SERVICES COMPANY

A Subsidiary of AMAX Inc.
1707 COLE BOULEVARD • GOLDEN, COLORADO 80401 • (303) 234-9020

June 13, 1985

DECUS Languages and Tools SIG
Mr. Alan L. Folsom, Jr., Editor
Dept. 431
Fischer & Porter Company
County Line Road
Warminster, PA 18974

A Programmer's Response to May 1984 FORTRAN-8X FIB-1 and the DEC and IBM Objections to Publication Therof

I am deeply disappointed by the negative responses to FORTRAN-8X which were expressed by the vendors, and which probably are representative of their customers. In a nutshell, the FORTRAN Committee is being asked to preserve nearly all of old FORTRAN in any new FORTRAN. What a shame. In the name of "upward compatibility", we are to be shackeled to dinosaurous language constructs that hobble the progress of programming practice and profession. Today's and tomorrow's programmers are to be mired in the tar pits of decades-old source code, and denied the ability to exploit and even propel the astonishing progress in hardware technology all around. As a programmer with over fifteen years' use of the language, I'm ready for a new, even radical if need be, departure from this venerable-but-musty old syntax. To the preservationists, I say "Your insistence on fossil-code compatibility is holding us down!"

It's very clear, for instance, that the WHERE construct and the array operations proposed in FORTRAN-8X FIB-1 facilitate some safe, simple, and unambiguous compilations into parallel-processor code. (Is it sinister, or mere coincidence that neither vendor is vanguard in parallel processing?) If WHERE and arrayops are written off as too "not demonstrably compatible" with existing FORTRAN, we will lose our best hope for orders-of-magnitude performance improvements for the sake of programs punched on 026's.

Computer vendors tend to see language changes in terms of extensions. That is how we get tied to particular brands in spite of standardized languages. Every vendor's FORTRAN-77 has its own set of extensions, and any program using those extensions suffers in portability to another computer model. As a result, the vendor is more familiar with tacking on incremental enhancements than writing a whole new compiler. Such methodology is costly: in the name of old-syntax compatibility, DEC-20 FORTRAN-77 is only this summer finally fleshed out; I would have been much happier to have a complete DEC-20 FORTRAN-77 implementation as early as the new VAX product came out, even if compatibility was compromised. (I understand that compatibility was what DEC-20 user representatives asked for; I think some leadership should have steered them away from that tar.)

Ultimately, compatibility is ridiculous, and doomed. Where would we be if a standards body had decreed that the next generation desktop hardware had to be 6502-compatible? Or, if no VAX were permitted that could not boot-up from a PDP-11 disk? The competition to FORTRAN is PASCAL; if FORTRAN cannot incorporate modern language constructs, it will end up abandoned, untaught, and unbought. I would much rather convert to a new, strong FORTRAN than have to abandon it all and go to another language. Yet there I am headed, if preservationists have their way.

Source-level compatibility is an issue for the myopic. Given an adequate filter program (which should be a part of the standard), I could even continue to write in FORTRAN-77 and use FORTRAN-8X. Surely it's easier to write a 77-to-8X filter than it is to write an 8X-and-77-and-66-and extensions compiler! Let the languages be incompatible; just give us the bridge out of one language and into the next.

Where are we going? New languages are incorporating good features of others. New TrueBASIC resembles PASCAL more than BASIC; I think that's to the better. If I'm not mistaken, FORTRAN-8X is learning a few tricks from APL; I would love to gain APL power without having to relearn operator precedence. New computer language constructs seem to be converging on psuedocode structure and higher-level abstractions. If we can just slough off some of the old baggage, our new languages can also become less complex and costly, more simple, effective, and economical.

Peter F. Klammer
Mineral Systems Development

/sm

This is a list of all corrections made to *The TEXbook* since the second printing. If your copy doesn't say '*Second printing (October 1984)*' on the copyright page, you should also look at the previous bug list. In fact, the most important corrections to the first printing were discovered first, so they've already been made.

Page 23, line 16 (10/13/84)

```
This is TeX, Version 1.0 (preloaded format=plain 83.7.15)
```

Page 33, line 32 (10/21/84)

The bottom line shows how far TEX has gotten until now in the story

---

Page 67, append a new exercise (1/19/85)

EXERCISE 11.6
Construct a `\frac` macro such that '`\frac1/2`' yields '$1/2$'.

Page 130, line 15 (4/17/85)

`$y'''_3+g'^2$` $y_3''' + g'^2$

Page 194, lines 13–15 should be centered better (10/22/84)

$$x \equiv x; \tag{1}$$

$$\text{if } x \equiv y \text{ then } y \equiv x; \tag{2}$$

$$\text{if } x \equiv y \text{ and } y \equiv z \text{ then } x \equiv z. \tag{3}$$

Page 215, lines 9 and 10 from the bottom (12/23/84)

general format is the same as for `\def` and `\gdef`, but TEX blindly expands the tokens of the replacement text according to the expansion rules above. For example, consider

---

Page 233, lines 15–19 (1/19/85)

Weight	*Servings*	*Approximate Cooking Time**
8 lbs.	6	1 hour and 50 to 55 minutes
9 lbs.	7 to 8	About 2 hours
$9\frac{1}{2}$ lbs.	8 to 9	2 hours and 10 to 15 minutes
$10\frac{1}{2}$ lbs.	9 to 10	2 hours and 15 to 20 minutes

Page 236, lines 18–21 (1/19/85)

Squab	*Poussin*	2	$\frac{3}{4}$ to 1	Broil, Grill, Roast
Broiler	*Poulet Nouveau*	2 to 3	$1\frac{1}{2}$ to $2\frac{1}{2}$	Broil, Grill, Roast
Fryer	*Poulet Reine*	3 to 5	2 to 3	Fry, Sauté, Roast
Roaster	*Poularde*	$5\frac{1}{2}$ to 9	Over 3	Roast, Poach, Fricassee

[This change should also be made at the bottom of page 237.]

Page 236, fifth-last line (1/19/85)

```
$quab&Poussin&2&\frac3/4$ to 1&Broil, Grill, Roast\cr
```

Page 237, line 25 (10/10/84)

saying '\tabskip=⟨glue⟩'. For example, let's do the poultry table again, but with the

Page 280, lines 7 and 8 (1/8/85)

⟨4-bit number⟩. The specified output stream is opened or closed, for use in \write commands, as explained in Chapter 21.

Page 300, lines 5–10 [changed for version 1.3] (11/25/84)

what part of TEX's memory has become overloaded; one of the following fourteen things will be mentioned:

```
number of strings      (names of control sequences and files)
pool size      (the characters in such names)
main memory size      (boxes, glue, breakpoints, token lists, characters, etc.)
```

Page 300, lines 23–29 [changed for version 1.3] (11/25/84)

If you have a job that doesn't overflow TEX's capacity, yet you want to see just how closely you have approached the limits, just set \tracingstats to a positive value before the end of your job. The log file will then conclude with a report on your actual usage of the first eleven things named above (i.e., the number of strings, ..., the save size), in that order. Furthermore, if you set \tracingstats equal to 2 or more, TEX will show its current memory usage whenever it does a \shipout command. Such statistics are broken into two parts; '490&5950' means, for example, that 490 words are being used for "large" things like boxes, glue, and breakpoints, while 5950 words are being used for "small" things like tokens and characters.

Page 305, line 26 (12/24/84)

sentable as ^^M. Asking TEX to \show\^^M produces the response '> \^^M=macro:->\␣'.

Page 308, line 25 (3/25/85)

```
\def\appendroman#1#2#3{\edef#1{\def\noexpand#1{\csname
```

Page 311, insert a new answer (1/19/85)

11.6.
```
\def\frac#1/#2{\leavevmode\kern.1em
\raise.5ex\hbox{\the\scriptfont0#1}\kern-.1em
/\kern-.15em\lower.25ex\hbox{\the\scriptfont0#2}}
```

[This causes answer 12.8 to move to page 312; answer 12.16 also moves to page 313.]

Page 327, lines 26–33 (10/22/84)

```
19.16. $$\displaylines{\hfill x\equiv x;\hfill\llap{(1)}\cr
        \hfill\hbox{if}\quad x\equiv y\quad\hbox{then}\quad
          y\equiv x;\hfill\llap{(2)}\cr
        \hfill\hbox{if}\quad x\equiv y\quad\hbox{and}\quad
          y\equiv z\quad\hbox{then}\quad
          x\equiv z.\hfill\llap{(3)}\cr}$$
```

There's also a trickier solution, which begins with

```
$$\displaylines{x\equiv x;\hfil\llap{(1)}\hfilneg\cr
```

Page 332, lines 17–24 (1/19/85)

```
\settabs\+\indent&10\frac1\2 lbs.\qquad&\it Servings\qquad&\cr
\+&\negthinspace\it Weight&\it Servings&
  {\it Approximate Cooking Time\/}+\cr
\smallskip
\+&8 lbs.&6&1 hour and 50 to 55 minutes\cr
\+&9 lbs.&7 to 8&About 2 hours\cr
\+&9\frac1/2 lbs.&8 to 9&2 hours and 10 to 15 minutes\cr
\+&10\frac1/2 lbs.&9 to 10&2 hours and 15 to 20 minutes\cr
```

Page 332, lines 33–35 (1/19/85)

proofs. (You weren't supposed to think of this, but it has to be mentioned.) See exercise 11.6 for the '\frac' macro; it's better to say '1/2' than '$\frac12$', in a cookbook.

Another way to treat this table would be to display it in a vbox, instead of including a first column whose sole purpose is to specify indentation.

Page 357, lines 35 and 36 (1/8/85)

```
\def\+{\discretionary{\thinspace\the\textfont2\char2}{}{}}
```

Page 357, last two lines (4/17/85)

```
\def\pr@m@s{\ifx'\next\let\next\pr@@@s \else\ifx^\next\let\next\pr@@@t
  \else\let\next\egroup\fi\fi \next}
\def\pr@@@s#1{\prim@s} \def\pr@@@t#1#2{#2\egroup}
```

Page 358, lines 8–12 (1/23/85)

```
\def\hbar{{\mathchar'26\mkern-9mu h}}
\def\surd{{\mathchar"1270}}
\def\angle{{\vbox{\ialign{$\m@th\scriptstyle##$\crcr
      \not\mathrel{\mkern14mu}\crcr \noalign{\nointerlineskip}
      \mkern2.5mu\leaders\hrule height.34pt\hfill\mkern2.5mu\crcr}}}}
```

Page 359, lines 7–8 (1/22/85)

```
\def\ddots{\mathinner{\mkern1mu\raise7pt\vbox{\kern7pt\hbox{.}}\mkern2mu
    \raise4pt\hbox{.}\mkern2mu\raise1pt\hbox{.}\mkern1mu}}
```

Page 360, line 22 (1/22/85)

```
  \mkern5mu \raise.6\dimen0\copy\rootbox \mkern-10mu \box0}
```

Page 361, line 3 (3/27/85)

```
\def\buildrel#1\over#2{\mathrel{\mathop{\kern0pt#2}\limits^{#1}}}
```

Page 361, lines 19–20 (1/22/85)

```
\def\bmod{\mskip-\medmuskip \mkern5mu
  \mathbin{\rm mod} \penalty900 \mkern5mu \mskip-\medmuskip}
```

Page 361, line 27 (5/1/85)

```
\def\matrix#1{\null\,\vcenter{\normalbaselines\m@th
```

Page 361, bottom line (5/1/85)

```
  \null\;\vbox{\kern\ht1\box2}\endgroup}
```

Page 362, line 9 (5/1/85)

```
\def\eqalign#1{\null\,\vcenter{\openup1\jot\m@th
```

Page 364, line 3 (3/23/85)

```
\def\plainoutput{\shipout\vbox{\makeheadline\pagebody\makefootline}%
```

Page 399, eighth-last line (2/11/85)

```
  \baselineskip=\footnotebaselineskip\noindent\unhbox0\par}
```

Page 401, line 5 (1/29/85)

```
\fontdimen parameters to qualify as a math symbol font). (2) Set all the font identifiers
```

Page 414, line 10 (12/17/84)

```
\font\titlefont=cmssdc40          % titles in chapter openings
```

Page 444, bottom line (1/10/85)

depth $d(z) + v$, consisting of box x followed by an appropriate kern followed by box z.

Page 466, left column (1/19/85)

fractions, 67, 139–143, 152, 170, 179, 186, 444–445.
huge, 196.
slashed form, 67, 139–140, 233, 236.

Page 470, index entries for `\longleftarrow` thru `\Longrightarrow` (10/5/84)

The reference to page 358 should be underlined (seven times).

Page 475, index entry for punctuation in formulas (4/29/85)

Add a reference to page 161.

Page 478, first and last lines (10/11/84)

Delete the last line in the right-hand column (since it appears on page 479), and add the following line at the top of the left-hand column (since it was dropped by mistake from the second printing):

styles of math formatting, 140–141, 441–447.

Page 478, tabskip entries (3/25/85)

Instead of '237–239' and '237–238' it should say '237–239' twice.

Page 483, lines 16–17 (1/19/85)

P.O. Box 9506
Providence RI 02940-9506, USA.

Page 483, lines 22–23 (1/19/85)

P.O. Box 9506
Providence RI 02940-9506, USA.

TPU
PROGRAMMING
TECHNIQUES

SEARCHING

- USE *EXACT* SEARCHES INSTEAD OF *NO_EXACT* SEARCHES WHERE POSSIBLE
- FORWARD SEARCHES ARE SLIGHTLY FASTER THAN REVERSE SEARCHES
- USE *SEARCH*(PATTERN_VARIABLE) INSTEAD OF *SEARCH*(PATTERN EXPRESSION)
- USE *ANY*('ABC') INSTEAD OF 'A'|'B'|'C'

SEARCHING

- USE *ANCHORED* PATTERN SEARCH TO LOOK FOR TEXT ON CURRENT LINE INSTEAD OF LOOKING AT EACH CHARACTER
- USE PATTERNS ALONG WITH OTHER TPU FUNCTIONS

SEARCHING

```
    ! check if a particular pattern lies within col 1 to 80)
PROCEDURE TEST

whitespace := " " + "	";  ! space + tab
page_pat := "!" & SPAN(whitespace) & "Page " & REMAIN;
search_range:= SEARCH(page_pat,forward);

    POSITION(search_range);

IF (GET_INFO(current_buffer,"offset_column") >= 1)
   THEN
    POSITION(END_OF(search_range));
     IF  (GET_INFO(current_buffer,"offset_column") <= 80)
      THEN
        POSITION(search_range);
        RETURN
    ENDIF;
ENDIF;

ENDPROCEDURE
```

USING THE LANGUAGE

- USE *IF* (EXP) INSTEAD OF *IF* (EXP) = 1
- USE *IF NOT* (EXP) INSTEAD OF *IF* (EXP) = 0
- USE *ON_ERROR* TO CATCH ERRORS INSTEAD OF CHECKING RESULTS OF *FUNCTIONS*

Slide 5

USING THE LANGUAGE

```
Procedure STRIP
        LOCAL searchpat,search_range;
on_error
return ! all done (when EOB or search fails)
endon_error;
  searchpat := SPAN( '    ' ) & LINE_END; ! match trailing spaces or tabs
LOOP
   search_range := SEARCH (searchpat,forward);
  position(beginning_of(search_range));
  erase_character(length(search_range));
ENDLOOP;
ENDPROCEDURE
```

USING THE LANGUAGE

- MOVE MARK AND *CREATE_RANGE* FUNCTIONS OUT OF *LOOPS* WHERE POSSIBLE
- MOVE *SUBSTR* OPERATIONS OUT OF *LOOPS* WHERE POSSIBLE
- USE THE *CASE* STATEMENT INSTEAD OF NESTED *IF-THEN-ELSE* WHERE POSSIBLE

ORDER OF MARKER VARIABLES IS IMPORTANT FOR RELATIONAL OPERATORS

USING THE LANGUAGE

- DELETE CONTENTS OF GLOBAL VARIABLES WHEN NO LONGER IN USE
- AVOID RUNNING THE COMPILER INADVERTENTLY
- USE ACTUAL VARIABLE NAMES INSTEAD OF *CURRENT_X* BUILT-INS

USING THE LANGUAGE

- AVOID NUMEROUS LOCAL DECLARATIONS IN FREQUENTLY CALLED PROCEDURES
- STORE STRING VARIABLES (IF USED REPEATEDLY) IN GLOBAL VARIABLES

PROCEDURE CALL OVERHEAD IS MINIMAL FOR PROCEDURES WITHOUT ARGUMENTS

BUFFER MANAGEMENT

- DELETING A BUFFER CAN BE VERY COMPUTE BOUND
- CREATE BUFFERS ONLY WHEN NECESSARY AND ERASE BUFFER CONTENTS WHEN NO LONGER NEEDED
- USE ***COPY*** OR ***MOVE TEXT*** TO MOVE LARGE BLOCKS OF DATA INSTEAD OF A ***SELECT*** RANGE
- (USE ***MAX_LINES*** BUFFER SETTING TO CONTROL BUFFER GROWTH)

EXTENDING EVE OR EDT INTERFACE

- READ CHAPTER 5 (REFERENCE), CHAPTER 6 (EVE), CHAPTER 10(EDTEM)
- NO PERFORMANCE PENALITY FOR HEAVILY COMMENTED PROGRAMS
- CONVERT A GROWING COMMAND FILE INTO SECTION FILE
- REMOVE UNUSED PROCEDURES FROM YOUR SECTION FILE

EXTENDING EVE OR EDT INTERFACE

- MOVE INFREQUENTLY USED PROGRAMS OUT OF YOUR SECTION FILE
- DYNAMICALLY COMPILE AND EXECUTE PROGRAMS WHEN NEEDED
- VM FOR PROCEDURES IN A SECTION FILE CANNOT BE RE-USED IF THE PROCEDURE IS REDEFINED
- DELETE PROCEDURES IF USED ONLY ONCE WHEN BUILDING SECTION FILE

EXTENDING EVE OR EDT INTERFACE

- PUT GLOBAL VARIABLE ASSIGNMENTS IN PROCEDURE AT THE BEGINNING OF SECTION FILE
- FEEL FREE TO "REDUCE" INTERFACE BASED ON PERSONAL PREFERENCE

LOW-SPEED TERMINALS (< 2400 BAUD)

- SHRINK YOUR MAIN WINDOW SIZE FOR SLOW SPEED LINE USAGE

```
DEFINE_KEY ("adjust_window(current_window,0,-10)",
    key_name("s",shift_key)
   )
```

LOW-SPEED TERMINALS
(< 2400 BAUD)

- ***USE OVERSTRIKE*** **EDITING IN THE MIDDLE OF TEXT WHERE POSSIBLE**

```
  PROCEDURE toggle
IF get_info (current_buffer, "mode") = overstrike
THEN
    set (insert, current_buffer);
ELSE
    set (overstrike, current_buffer);
ENDIF;
ENDPROCEDURE
DEFINE_KEY('toggle',ctrl_a_key);
```

LOW-SPEED TERMINALS
(< 2400 BAUD)

- **TO INSERT LINE OF TEXT, "OPEN" UP A NEW LINE FIRST, RATHER THAN PUSH EXISTING LINE OVER**
- **USE SET*(MESSAGE_FLAGS,5)* WHERE POSSIBLE**
- **USE *NONE* ATTRIBUTE FOR SELECT OPERATIONS OR HIGHLIGHT ONLY THE STARTING CHARACTER OF THE SELECT RANGE**

WINDOWING

- USE MINIMUM NUMBER OF WINDOWS
- AVOID OVERLAPPING WINDOWS
- USE ***NONE*** ATTRIBUTE FOR MARKERS AND RANGES WHERE POSSIBLE

WINDOWING

- KEEP ONE BOUNDARY OF YOUR MAIN WINDOW AT THE TOP OR BOTTOM EDGE OF SCREEN
- USE ***UPDATE,SCROLL,REFRESH,SHIFT*** WINDOW ONLY WHEN NECESSARY
- USE ***TIMER*** MESSAGE ONLY WHEN NECESSARY
- USE "SECTION" MOVES INSTEAD OF AUTOREPEATING ARROW KEYS WHERE POSSIBLE

READING INPUT

- USE ***READ_LINE*** INSTEAD OF PROGRAMMABLE COMMAND WINDOW WHERE POSSIBLE
- USE ***READ_KEY*** INSTEAD OF ***READ_CHAR*** WHERE POSSIBLE
- USE DEFINED KEYS INSTEAD OF COMMAND LINE MODE

Slide 19

MISCELLANEOUS

- USE TPU AS "***KEPT***" EDITOR INSTEAD OF INVOKING TPU MULTIPLE TIMES TO EDIT MULTIPLE FILES
- USE ***EDIT/TPU/NOCOMMAND*** IF A COMMAND FILE IS NEVER USED
- DO ***CREATE_PROCESS*** ONCE , NOT ONCE PER DCL COMMAND !

VAX™ Ada®
FIELD TEST REPORT

Bevin Brett
VAX Ada Project
DIGITAL

85APR26

I. FIELD TEST - DURATION AND MAGNITUDE

Duration

FT1 officially started	01-Nov-1984
kits sent out	10-Nov-1984 (Approx)
FT2 started	08-Feb-1985
SDC code freeze	08-Mar-1985

85APR26

I.A FIELD TEST - DURATION AND MAGNITUDE

Magnitude

20 FT sites,

4_000_000 lines of Ada code

85APR26

II. QAR SUMMARY

Number of QARS

Total Closed:	**287**
Total Answered:	**9**
Total Open:	**11**
Total QARs:	**307**

85APR26

II.A QAR SUMMARY (CONT'D)

Breakdown of Closed QARs by Category

	Number	Total %
Fixed in next release:	107	34
Inquiry:	68	22
User error:	38	12
Suggestion:	25	8
Documentation error:	19	6
Not reproducible:	11	3
Informational:	4	1
Temporary workaround:	3	0
Design change:	2	0
Unsupported:	2	0
Temporary patch:	0	0

Total: 287

85APR26

II.c QAR SUMMARY -

USER ERRORS

User errors

- Overloading resolution problems, the compiler is MUCH better than humans!
- Exact reading of manual on "intuitively obvious" issues

85APR26

II.c1 QAR SUMMARY -

USER ERRORS (CONT'D

- People confuse run-time exceptions and compile-time errors
- Unconstrained records causing storage error

85APR26

II.c2 QAR SUMMARY -

USER ERRORS (CONT'D)

- Absence of Volatile makes things that look like optimizer errors
- Misunderstandings of I/O and Tasking

85APR26

III. FT SITE COMMENTS

Comments from users

- Improvements
- Documentation
- Conversions from other compilers
- Other

85APR26

III.A FT SITE COMMENTS -

IMPROVEMENTS

- Resources consumed
  - cpu
  - memory
  - I/O
- Variety of ideas for improving

85APR26

III.A2 FT SITE COMMENTS -

IMPROVEMENTS (CONT'D)

Diagnostics

- "some error messages seemed a little obscure"
- "good diagnostics provided by the VAX Ada compiler"

Enhancements

- Recognize more special cases

85APR26

III.A3 FT SITE COMMENTS -

IMPROVEMENTS (CONT'D)

Generated code quality

- Few actual suggestions
- Already similar to Pascal, PL/I, C
- Depends on program
- Recognize more special cases

85APR26

III.A4 FT SITE COMMENTS -

IMPROVEMENTS (CONT'D)

ACS enhancements

- Better compilation management
- Ada command to compile a file for which a unit does not yet exist
- Tool to determine the correct order to compile a set of files

85APR26

III.A5 FT SITE COMMENTS -

IMPROVEMENTS (CONT'D)

More programming environment tools

- Cross reference utility
- Pretty printer

85APR26

III.B ' FT SITE COMMENTS

DOCUMENTATION

- "best documentation of any Ada product we've seen"
- "Didn't use very much"

85APR26

III.B1 FT SITE COMMENTS -

DOCUMENTATION (CONT'D)

More examples, especially

- Common Language Environment
- I/O

85APR26

IV GATHERED STATISTICS

- Nature of source programs
  - size
  - number of lines after generics and inlining
- Compiler performance
  - working set
  - page faults
  - cpu time
  - elapsed time
- Compilation library characteristics
  - traffic analysis

85APR26

IV. CONCLUSIONS

@ 1500 page working set seems about right

@ Speeds between 250 and 2500 lines per minute

@ Room for improvement

@ More data needed

85APR26

V FIELD TEST CONCLUSIONS

- Duration adequate because of limited range of applications available
- Not as many "user errors" as expected
- It's amazing how many bugs you can have in a validated compiler!
- Very few known bugs in the V1 compiler

VAX Ada® Field Test Panel

Bob Gable
Lear Siegler/Instrument Division
4141 Eastern Avenue SE MS 121
Grand Rapids, Michigan 49508

Spring 1985 DECUS (New Orleans)

®Ada is a registered trademark of the U.S. Government (Ada Joint Program Office).

CONCLUSIONS

- VAX Ada Right Choice for
  - VAX-Hosted Support Software
  - Ada Training (with LSE)
  - Embedded Software
    - Initial Development
    - Module and Function Test
- Well Integrated with CMS, LSE
- VAX Ada, VAX LISP ==> 2 8600's in Cluster
- The Ada Language
  - Time-Consuming to Learn
  - Efficiency Problems for Real-Time, Embedded Computers
  - Promising Features

2

Lear Siegler Ada Work
Current and Near-Term

- Previous Languages
  - FORTRAN, JOVIAL, Pascal, assembly lang. ==>,Ada
- Configuration Management Tools
- MIL-STD-1750A, Z8002
- Embedded Flight Computer Software
  - Module, Functional Development and Verification
  - Cross-Targeted Compiler for Integration
  - Support Software (Linker, Assembler)

VAX Ada
Callable-CMS

- CMS$SHOW_GENERATION
  - Passing Procedure Addresses by Value
    - ROUTINE'ADDRESS attribute
  - DEC provided IMPORT_VALUED_PROCEDURE pragma
  - EXPORT_VALUED_PROCEDURE Pragma Needed
    - CMS Calls User Function with In Out Argument; Expects User to Return Status Value; Resorted to FORTRAN call
- Defining Interface Packages Time Consuming

VAX Ada Field Test Problems

1. Ada Language Itself
2. Many Bugs in Separate Compilation
3. Using VAX Debugger with Ada More Difficult than with VAX FORTRAN (esp tasking)
4. Compiling Ada Programs in ADA$BATCH from LSE Awkward
5. Fewer Problems than VAX FORTRAN Field Test

5

VAX Ada Compilation Speed

Compiler	Max. Working Set Size	CPU Secs.	Page Faults	Source Lines/Min.
VAX Ada	2048	11	1087	502
Lear FORTRAN	1024	31	285	387

- VAX Ada Working Set Size

< 1000	Too Slow
1000-2000	Tolerable
2500	DEC's Recommended Value

- VAX Ada Runs Quickly, Uses Large Amounts of Memory

6

VAX Ada Field Test Mechanics

- Ada Team Responds Quickly!
- Daily Review of QAR Data Base
- Learn from Others' Problems
  - Dynamic Strings
  - Working Set Tuning
- Transfer of Files - Not Ideal

7

ACM SigAda Ada Run-Time Environment Working Group

Establish Conventions, Criteria, Guidelines for Ada Run-Time Environments; Provide Framework to Evaluate and Effectively Use Ada RTS

- Group is Mixture of Ada Implementers, Users
- Can Real-Time, Embedded Applications Use Full Ada? even small systems?
- Working Group Consensus
  - DEC Documentation Superb - Especially Implementation Dependent Features of Language, Run-Time System
  - Ada LRM with Interspersed with VAX Ada Features
  - If Only the Cross-Targeting Ada Compiler Vendors Did As Well

8

Language-Sensitive Editor Field Test

- Significantly Aids Programming in New Languages (e.g. Ada)
- Customized LSE for Other Languages
  - JOVIAL
  - Microtec/8086 Pascal
  - SLiTeX, LaTeX
  - Commonly-Used System Service Calls (from FORTRAN)
- "Semi-Compatible" with EDT

9

Performance Coverage Analyzer Field Test

- Embedded Computer Software Functional Test
  - For Path Coverage/Statement Coverage
  - Replaces FORTRAN/JOVIAL/PASCAL-Specific Tools
  - Only Tool To Do This for Ada
- Support Software
  - For Performance Improvement

10

Languages and Tools Wishlist
Results and DEC Responses

– K. Hornbach
Chair, Languages & Tools

The Languages & Tools SIG published a wishlist in the February issue of the newsletter. The results were tabulated by Al Folsom and Al Rizzuto, and forwarded on to DEC. DEC responded to the Wishlist items at the New Orleans Symposium. The following are their responses, based on notes I took in the session. I have attempted to make them as accurate as possible, but please remember that all the usual disclaimers apply.

1. **DEC should make all VAX manuals available online, and provide a structured way to access them.**(126 votes)

   DEC Response: We are actively looking at it. We have worked with some of these capabilities internally. The new CDROM (laser disk system) offers interesting possibilities. The HELP files under Version 4.0 VMS are greatly expanded, and are a step in that direction.

2. **DEC should provide a sophisticated test formatter, with things such as multiple fonts, proportional spacing, math/-Greek, and macros (along the lines of formatters like TEX and Scribe).** (117 votes)

   DEC Response: We are evaluating a course of action in this important marketing area. We have worked with a number of such systems (like TEX and Scribe) internally, and are tracking the technology.

3. **DEC should add the complete "Block DO" and "Block CASE" constructs to VAX-Fortran.** (92 votes)

   DEC Response: Portions of these constructs are already implemented in VAX-Fortran V4.0. We may implement further features in V5.0, and would expect to comply with the Fortran standard in this area, once it becomes finalized.

4. **All VAX compilers should support a `/check=argument_count` qualifier (that checks to make sure the number and types of arguments passed match).** (92 votes)

   DEC Response: Should all languages support this? Would it be something that should be done at link time, rather than compile time, so that all calling sequences can be checked? We are interested in such a capability but have questions about how it could be effectively implemented in the Common Language Environment.

5. **DEC should market a general programmer's workstation.** (79 votes)

   DEC Response: We are evolving towards that, with the VAXstation II. We need to know more what you think a programmer's workstation should consist of.

6. **DEC should provide an automated Documentation Control Manager.** (79 votes)

   DEC Response: We're not sure what this means. CMS will store any ASCII file, and we use that internally to control our documents.

7. **DEC should provide a "Lint"-like tool for VAX C.** (78 votes)

   DEC Response: We are very aware of the need for a Lint-like capability. We are interested in C portability issues. The `/std=port` compile option does a lot of checking. We will take a look at doing more, but will probably approach the solution from the DEC philosophy of providing tools that are multi-lingual, vs. a separate Lint-like tool for every different compiler.

8. **DEC product brochures should contain more technical information about actual capabilities and requirements for a language/tool.** (76 votes)

DEC Response: We are aware of the need and interested in solving it. We have many different types of product literature; one of our biggest problems is how to ensure that we have a complete and up-to-date mailing list, so that the information actually reaches you. The Electronic Store is starting to make tool demos available on line.

9. **DEC should provide on-line CAI (Computer Aided Instruction) classes for all new software tools.** (71 votes)

   DEC Response: CAI courses are produced by Educational Services at DEC; they feel that CAI courses are not appropriate for technical people; they should be aimed more towards non-technical types. They are expensive to develop, and must justify themselves as a product.

10. **DEC should provide more complete implementation of the C runtime library and closer compatibility of functions such as vfork() to the UNIX implementation.** (69 votes)

    DEC Response: Version 2 of C has a lot more of the capability you are requesting. We are aware of the need and try to do as much as possible.

11. **DEC should provide software tool support for software design.** (68 votes)

    DEC Response: We are very interested in this topic, and are formulating ideas about how to meet this need. We would be very interested in hearing about the methodologies you use and would like to see supported.

12. **DEC should provide more emulation of the UNIX environment on VMS.** (68 votes)

    DEC Response: We are continuously working to enhance the C run time library and DEC/Shell, and plan to continue this activity.

13. **DEC should provide a problem report database tool that**

is integrated with DEC/CMS and DEC/Test Manager. (67 votes)

DEC Response: This is something we are interested in. It relates to the concept of an overall "Configuration Management" tool. We have given sessions in this area at the last few Symposia. We are still working out all the details of what a configuration management tool would cover; but problem reports would obviously be one aspect.

14. **DEC should provide a tool to generate the dependancies file for MMS.** (65 votes)

    DEC Response: We're like to solve this problem - it is the most requested enhancement for MMS. We would like such a tool to also work on software outside the Common Language Environment, which makes it a much more complicated problem. However, we do have some ideas about how it might be solved.

15. **DEC should provide a configuration control manager.** (63 votes)

    DEC Response: We are very interested in this, and are working on concepts for it in advanced development right now.

16. **DEC should support GKS to level 2.** (63 votes)

    DEC Response: We have plans to do this.

17. **DEC should provide a description of all tools in on-line help, even if they're not licensed on the system, so people know what's available and what the general capabilites for each tool are.** (63 votes)

    DEC Response: It's a good idea, but we have some questions about implementation details. It would require quite a bit of logistical manuevering to get it to happen. We plan on exploring this idea further.

LTSIG Wishlist
for Anaheim

Hello - My name is Alan Rizzuto, I am your new wishlist coordinator for the LTSIG. Some of you my have seen my name in the newsletter or may have met me at New Orleans. I am replacing Al Folsom who is publishing this newsletter now.

I would like to take this opportunity to acquaint you with what a wishlist is. If you already know what the wishlist is, I would like to explain the importance of it to you and the other members of the SIG.

The wishlist is our (leaders of the SIG) method of finding out the needs of each of the members of the SIG. The two areas of help that the wishlist provides is: what can the leadership of the SIG do for you the member, and what can DEC do to help you accomplish your task more efficiently.

The most important part of the wishlist is YOUR involment in the wishlist. Each time we present the wishlist for voting (twice a year, before each symposia), we will be asking YOU for your ideas on what should be on the wishlist. The only way that the wishlist can be totally effective is if YOU the member put your ideas into the wishlist.

The wishlist is then voted upon by the complete membership, which gives us a view of what all of the members feel about a given issue. The items that recieved the most votes (top ten) from the membership are then responded to by the leadership of the SIG for SIG related issues and most importantly DEC responds to the DEC related issues. This allows you the member input to both DEC and the leadership of the SIG.

On the following page you will find a form that should be used to write your wishlist entry. The wishlist will be published in the next issue of the newsletter. Please feel free to copy the form and send me as many items that you would like to see on the wishlist.

If you have any questions or would like to contact me for any reason you can reach me at the following address or phone number between the hours of 0830 and 1700 Eastern Standard Time.

Alan Rizzuto
EMC Systems Inc.
P.O. Box 534
Cockeysville, Md. 21030
(301) 628-8167

Please send the forms in as soon as possible so that your item will appear in the wishlist.

Thankyou,
Alan Rizzuto

L&T Spring 1985 Wishlist Questionaire

Name: _____

Address: _____

City, State ZIP: _____

Phone: _____

Check the one most appropriate choice:

- ☐ Interested in Languages and Tools
- ☐ User of Languages and Tools
- ☐ User of Languages and Tools, and interface with DEC developers

Wishlist Question is directed toward:

- ☐ Languages and Tools SIG
- ☐ DEC

Type(s) of Operating Systems: _____

_____

Type(s) of Hosts: _____

_____

Wishlist Question: _____

_____

_____

_____

_____

_____

Printed in the U.S.A.

DECUS SUBSCRIPTION SERVICE
DIGITAL EQUIPMENT COMPUTER SOCIETY
219 BOSTON POST ROAD, (BP02)
MARLBORO, MA 01752

Bulk Rate
U.S. Postage
PAID
Permit No. 18
Leominster, MA
01453

STATUS CHANGE

Please notify us immediately to guarantee continuing receipt of DECUS literature. Allow up to six weeks for change to take effect.

( ) Change of Address
( ) Please Delete My Membership Record
(I Do Not Wish To Remain A Member)

DECUS Membership No: _____

Name: _____

Company: _____

Address: _____

_____

State/Country: _____

Zip/Postal Code: _____

Mail to: DECUS - Attn: Subscription Service
219 Boston Post Road, BP02
Marlboro, Massachusetts 01752 USA

bel
not
old
ide
on,
ity,