

fig-FORTH for PDP-11

ASSEMBLY SOURCE LISTING

WITH COMPILER SECURITY

AND

VARIABLE LENGTH NAMES

VERSION 1.3

JANUARY 1980

Provided through the courtesy of the



P.O. Box 8231 San Jose, CA 95155 (408)277-0668

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```
.TITLE F. I. G.
*****
PDP-11 FORTH          INTRODUCTION          PDP-11 FORTH
*****
PDP-11 FORTH          RT-11, RSX-11M, AND STAND-ALONE          JANUARY 1980

DEVELOPED BY THE
FORTH INTEREST GROUP / FORTH IMPLEMENTATION TEAM
P. O. BOX 1105
SAN CARLOS, CA. 94070

IMPLEMENTED BY
JOHN S. JAMES
P. O. BOX 348
BERKELEY, CA. 94701

THIS SYSTEM IS IN THE PUBLIC DOMAIN AND CAN BE USED
WITHOUT RESTRICTION. PLEASE CREDIT THE FORTH INTEREST
GROUP IF YOU REPUBLISH SUBSTANTIAL PORTIONS.

THE FORTH INTEREST GROUP / FORTH IMPLEMENTATION TEAM
ALSO HAS DEVELOPED NEARLY IDENTICAL VERSIONS OF THIS
SYSTEM FOR THE
8080          6502
6800          6809
PACE          ALPHA MICRO

FOR MORE INFORMATION, WRITE:
JOHN S. JAMES
P. O. BOX 348
BERKELEY, CA. 94701

OR

FORTH INTEREST GROUP
P. O. BOX 1105
SAN CARLOS, CA. 94070

'PDP' AND 'RSX' ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION.
```

NOTE: This system
is available on
diskette from the
author, John James
PO Box 348
Berkeley, CA 94701
(415) 526-8815

```

56      ,      THIS FORTH SYSTEM HAS
57      ,      - FULL LENGTH NAMES
58      ,      - EXTENSIVE COMPILE-TIME CHECKS AND ERROR MESSAGES
59      ,      - DOUBLE INTEGER I/O
60      ,      - A FORTH ASSEMBLER, PERMITTING STRUCTURED, INTERACTIVE
61      ,      DEVELOPMENT OF DEVICE HANDLERS, SPEED-CRITICAL
62      ,      ROUTINES, AND LINKAGE TO OPERATING SYSTEMS OR TO
63      ,      SUBROUTINE PACKAGES WRITTEN IN OTHER LANGUAGES.
64      ,      - STRING-HANDLING ROUTINES
65      ,      - A STRING-SEARCH EDITOR
66      ,      - LINKED VOCABULARIES
67      ,      - HOOKS FOR MULTITASKING/MULTIUSER (CURRENTLY SINGLE TASK)
68      ,      - AND AS CURRENTLY CONFIGURED IT RUNS IN A 24K BYTE
69      ,      TASK IMAGE (THIS INCLUDES BUFFERS, OPERATING-SYSTEM
70      ,      AREA, AND ROOM FOR SUBSTANTIAL ADDITIONAL FORTH
71      ,      PROGRAMMING) ON ANY PDP-11 OR LSI-11 CPU, WITH OR
72      ,      WITHOUT HARDWARE MULTIPLY/DIVIDE. THIS DISKETTE
73      ,      WILL BOOT AND RUN STAND-ALONE; ALSO IT CONTAINS A
74      ,      SOURCE PROGRAM WHICH CAN BE ASSEMBLED TO RUN
75      ,      UNDER RT-11, RSX-11M, OR STAND-ALONE. THIS SYSTEM
76      ,      CAN BE MODIFIED TO INTERFACE WITH ANY OTHER OPERATING
77      ,      SYSTEM WHICH SUPPORTS READ AND WRITE A CHARACTER,
78      ,      DETECT A CHARACTER (OPTIONAL), AND READ AND WRITE
79      ,      A DISK BLOCK.
80      ,
81      ,
82      ,      IT IS ALIGNED WITH THE 1978 STANDARD OF THE FORTH INTERNATIONAL
83      ,      STANDARDS TEAM.
84      ,
85      ,
86      ,
87      ,      RECOMMENDED DOCUMENTATION:
88      ,      - A FORTH LANGUAGE MANUAL. WE PARTICULARLY RECOMMEND EITHER
89      ,      (A) 'USING FORTH', BY FORTH, INC.
90      ,      OR
91      ,      (B) 'A FORTH PRIMER', BY W. RICHARD STEVENS, KIT
92      ,      PEAK NATIONAL OBSERVATORY.
93      ,      EITHER IS AVAILABLE THROUGH THE FORTH INTEREST GROUP,
94      ,      P.O. BOX 1105, SAN CARLOS, CA. 94070.
95      ,      - PDP-11 FORTH USER'S GUIDE, AVAILABLE FROM JOHN S. JAMES,
96      ,      ADDRESS ABOVE. Also available from FIG.
97      ,      - FORTH REFERENCE CARD FOR THE FORTH IMPLEMENTATION TEAM
98      ,      COMMON MODEL, AVAILABLE FROM FIG.
99      ,      - 'FIG-FORTH INSTALLATION MANUAL', ALSO FROM FIG.
100     ,
101     ,
102     ,
103     ,      ACKNOWLEDGMENTS:
104     ,      THIS FORTH SYSTEM (IN 'FORTH.MAC') IS A GROUP PRODUCT
105     ,      OF THE FORTH IMPLEMENTATION TEAM OF THE FORTH INTEREST
106     ,      GROUP (P.O. BOX 1105, SAN CARLOS CA. 94070). THE IMPLEMENTER
107     ,      IS RESPONSIBLE FOR THIS PDP-11 VERSION OF THE MODEL, AND FOR
108     ,      THE SOFTWARE ON SCREENS IN 'FORTH.DAT'. ALTHOUGH THE LATTER
109     ,      IS NOT AN OFFICIAL RELEASE OF THE F. I. O., THE CONTRIBUTIONS
110     ,      FROM MEMBERS OF THE GROUP ARE TOO NUMEROUS TO CITE
111     ,      INDIVIDUALLY.
112     ,      IN ADDITION WE APPRECIATE THE PDP-11 CODING

```

113
114
115

;
;
;

IMPROVEMENTS SUGGESTED BY STUART R. DOLE, DOLE & FARMER,
PO BOX 142, PETALUMA, CA. 94952; BY PAUL EDELSTEIN;
BY RICK STEVENS OF KITT PEAK; AND OTHERS.

```

117      / *****
118      /
119      / BRINGING UP THE SYSTEM
120      /
121      / *****
122      /
123      /
124      /
125      / TO RUN STAND-ALONE:
126      /
127      / - BOOT THE DISKETTE LIKE ANY OTHER SYSTEM DISK, FROM DX0.
128      / FORTH SHOULD COME UP AND TYPE 'FIO FORTH' AND THE VERSION
129      / NUMBER. TEST AS DESCRIBED FOR RT-11 BELOW.
130      /
131      / - MAKE A COPY OF THE DISK; THIS STAND-ALONE SYSTEM DOES NOT
132      / PROTECT AGAINST ACCIDENTALLY OVERWRITING THE SYSTEM OR THE
133      / SOURCE PROGRAMS. TO MAKE AN EXACT COPY OF THE ENTIRE DISK,
134      /
135      / 1. PUT A BLANK DISK INTO THE SECOND DRIVE (DX1). FOR
136      / SAFETY, SET THE WRITE-PROTECT SWITCH ON THE DRIVE
137      / WHICH CONTAINS THE ORIGINAL SYSTEM DISK.
138      /
139      / 2. TYPE '38 LOAD', AND CARRIAGE RETURN. THE SYSTEM SHOULD
140      / RESPOND 'OK'. THEN TYPE 'COPY' AND RETURN. EACH OF
141      / THE 77 TRACKS WILL BE READ FROM DX0 AND WRITTEN ON DX1.
142      /
143      / - NOTE THE LAYOUT OF THE DISKETTE. IT IS SET UP TO BOOT AND
144      / RUN STAND-ALONE, BUT IT ALSO CONTAINS AN RT-11 DIRECTORY,
145      / AND A MACRO-11 SOURCE PROGRAM 'FORTH.MAC' (WHICH PRODUCED
146      / THIS LISTING). THIS ALLOWS THE SAME DISK TO BE BOOTED
147      / AND RUN, OR TO PROVIDE SOURCE FOR MODIFICATION AND RE-ASSEMBLY.
148      / AS PROVIDED, THE FILE 'FORTH.DAT' CONTAINS FORTH SCREENS
149      / 1-70. YOU CAN USE LOCATIONS BEYOND 70, BUT THESE WILL
150      / OVERWRITE THE 'FORTH.MAC' SOURCE PROGRAM. STAND-ALONE USERS
151      / MAY NEVER NEED TO USE THIS SOURCE, AND MAY WANT TO REMOVE IT
152      / AND USE THE SPACE FOR SOMETHING ELSE. MAKE A COPY FIRST.
153      /
154      / - STAND-ALONE USERS CAN ADD THEIR OWN OPERATIONS AND THEN
155      / SAVE A BOOTABLE IMAGE OF THE NEW SYSTEM. THE NEW OPERATIONS
156      / WILL BE AVAILABLE WHEN THE DISK IS BOOTED IN THE FUTURE.
157      / THE LOADER WHICH IS USED WILL ONLY LOAD IMAGES UP TO 7.9K;
158      / THIS LEAVES SEVERAL HUNDRED BYTES FOR NEW OPERATIONS, WHICH
159      / CAN INCLUDE EXTENDING THE SYSTEM BY BRINGING IN SOURCE OR
160      / OBJECT CODE. TO SAVE THE CURRENT SYSTEM, EXECUTE 'FORTH DEFINITIONS'
161      / IF NECESSARY TO GET INTO THE FORTH VOCABULARY, THEN 'DECIMAL 34 LOAD'
162      / SOME WARNING MESSAGES WILL BE PRINTED (MSG #4); THEY CAN BE
163      / IGNORED.
164      /
165      / - IF YOU DO WANT TO RE-ASSEMBLE THE SYSTEM FOR STAND-ALONE
166      / USE (WHICH MOST USERS SHOULD NEVER FIND NECESSARY),
167      / YOU MUST USE RT-11 TO EDIT AND ASSEMBLE 'FORTH.MAC'. NOTE
168      / THAT ALTHOUGH THIS LISTING IS ASSEMBLED FOR STAND-ALONE,
169      / THE SOURCE PROGRAM SUPPLIED IS SET FOR RT-11 ASSEMBLY;
170      / COMMENT OUT THE 'RT11' DEFINITION, AND REMOVE THE COMMENTING
171      / ON 'ALONE'. ASSEMBLE, LINK, AND RUN, AND THE SYSTEM SHOULD
172      / COME UP STAND-ALONE. IMMEDIATELY REMOVE THE RT-11 SYSTEM DISK
173      / AND PLACE THE FORTH DISK IN DRIVE ZERO. TO REVISE
174      / THE BOOTABLE IMAGE ON THE FORTH DISK SO THAT YOUR NEW SYSTEM
175      / BOOTS STAND-ALONE, LIST SCREEN 34 (DECIMAL), AND FOLLOW THE
176      / INSTRUCTIONS THERE. THE RUN TAKES ABOUT ONE MINUTE.
177      /
178      / - THE BOOTABLE SYSTEM DOES NOT USE HARDWARE MULTIPLY AND DIVIDE.
179      / IF YOU DON'T HAVE RT-11 TO EDIT AND RECOMPILE WITH 'EIS'
180      / CONDITIONAL ASSEMBLY, THE MULTIPLY/DIVIDE ROUTINES CAN BE
181      / PATCHED. IF YOU PATCH FROM THE KEYBOARD MONITOR, THE

```

```

174      , RESTART ADDRESS IS 1000 OCTAL ( COLD START) OR 1004 (WARM
175      , START). SAVE THE NEW VERSION AS A BOOTABLE SYSTEM, AS
176      , DESCRIBED ABOVE.
177      ,
178      , - THE SKEWED DISK I/O OPERATIONS SKIP TRACK ZERO, FOR COMPATIBILITY
179      , WITH STANDARD PDP-11 SECTOR SKEWING. THE PHYSICAL READ
180      , OPERATIONS ('RTS', 'WTS', 'NRTS', 'NWTS') CAN READ ANY SECTOR,
181      , HOWEVER.
182      ,
183      , - ALSO THE SYSTEM AS DISTRIBUTED SKIPS THE FIRST 56 SECTORS
184      , (7 SCREENS) IN ORDER TO SKIP THE BOOT BLOCK AND AN
185      , RT-11 DIRECTORY. THIS CAUSES THE SCREEN POSITIONS TO BE THE
186      , SAME FOR STAND-ALONE AND FOR RT-11 (WHICH ACCESSES THE FILE
187      , 'FORTH.DAT'). YOU CAN CHANGE THIS BY CHANGING THE VALUE OF
188      , THE VARIABLES 'S-SKIP' (NUMBER OF SCREENS SKIPPED) AND
189      , 'S-USE' (NUMBER OF SCREENS USED BEFORE ACCESSING THE
190      , SECOND DISK). THESE VARIABLES CAN BE CHANGED AT ANY TIME.
191      , SO DISK SCREENS CAN BE READ INTO BUFFERS AND THEN FLUSHED
192      , TO DIFFERENT LOCATIONS ON THE DISK.
193      ,
194      , - ADVANCED USERS MAY NOTE THAT THIS SYSTEM IS DESIGNED TO
195      , ALLOW THE MEMORY LAYOUT - NUMBER AND LOCATION OF DISK
196      , BUFFERS, LOCATION OF THE STACK, ETC. - TO BE CHANGED
197      , DYNAMICALLY, WITHOUT REASSEMBLY.
198      ,
199      , TO BRING UP THIS SYSTEM UNDER RT-11:
200      ,
201      , - BE SURE THAT RT-11 IS SELECTED BELOW. THE LINES DEFINING
202      , 'RSX11M' AND 'ALONE' SHOULD BE COMMENTED OUT; 'RT11' SHOULD
203      , NOT BE. NOTE THAT THIS DISK IS DISTRIBUTED READY FOR RT-11
204      , ASSEMBLY (EVEN THOUGH THIS LISTING IS FOR STAND-ALONE).
205      ,
206      , - IF YOU HAVE HARDWARE MULTIPLY/DIVIDE, ALSO REMOVE THE
207      , SEMICOLON FROM THE LINE DEFINING 'EIS'.
208      ,
209      , - IF YOU ARE USING AN OLDER VERSION OF RT-11 (VERSION 2),
210      , YOU MAY NEED TO USE THE MACROS '..V2..' AND '.REODEF'.
211      ,
212      , - ASSEMBLE, LINK, AND RUN. THE SYSTEM SHOULD COME UP AND
213      , TYPE 'FIO-FORTH' AND THE VERSION NUMBER.
214      ,
215      , - TEST THAT IT IS UP BY TRYING SOME ARITHMETIC OR DEFINITIONS, E. G.
216      ,      88 88 * . (NOTE THAT THE '.' MEANS PRINT)
217      ,      : SQUARE DUP * ;
218      ,      25 SQUARE .
219      ,
220      , OR TYPE 'VLIST' FOR A LIST OF ALL THE FORTH OPERATIONS IN THE
221      , DICTIONARY.
222      ,
223      , - THE DISK SHOULD WORK IF THE DISKETTE IS IN DRIVE 'DK'.
224      , MAKE SURE THAT 'DK' IS ASSIGNED TO WHATEVER PHYSICAL
225      , DRIVE YOU ARE USING - OR CHANGE LINE 'RTFILE:' IN
226      , 'FORTH.MAC'. TEST THE DISK BY TYPING
227      ,      1 LIST
228      ,
229      , WHICH SHOULD LIST THE SCREEN WHICH LOADS THE EDITOR,
230      , ASSEMBLER, AND STRING ROUTINES.
231      ,
232      , - IN CASE YOU NEED TO GET A LISTING FROM THE ASSEMBLY OF
233      , 'FORTH.MAC' (NOT USUALLY NECESSARY), AND YOUR SYSTEM HAS
234      , ONLY DISKETTES (NO LARGER DISKS), THE 'ALLOCATE' OPTION
235      , IS NECESSARY BECAUSE OF THE SIZE OF THE '.LST' FILE
236      , (AROUND 230 BLOCKS). FIRST COPY 'FORTH.MAC' ONTO A
237      , SEPARATE DISKETTE BY ITSELF. THEN EXECUTE
238      ,      .MACRO /LIST:FORTH.LST /ALLOCATE:300. /NOOBJECT
239      , AND REPLY 'FORTH.MAC' WHEN ASKED FOR 'FILES?'.
240      ,

```

```

231      ,
232      , TO BRING UP THE SYSTEM UNDER RSX-11M:
233      , - THE DISKETTE PROVIDED IS IN RT-11 FILE FORMAT. THE TWO FILES
234      , MUST BE COPIED OFF THE DISKETTE INTO AN RSX DIRECTORY. THE
235      , 'FORTH.DAT' FILE MUST BE COPIED IN IMAGE MODE. ANY RSX
236      , DIRECTORY MAY BE USED. ASSUMING THE DISKETTE IS IN DRIVE 0,
237      , USE THE RSX COMMANDS:
238      , >FLX =DX:FORTH.MAC/RT
239      , >FLX =DX:FORTH.DAT/RT/IM
240      , INCIDENTALLY, 'FORTH.DAT' IS THE SYSTEM'S 'VIRTUAL MEMORY'
241      , FILE, USED FOR DISK I/O. THE REST OF THE SYSTEM (THIS
242      , PROGRAM ALONE) CAN RUN INDEPENDENTLY, EVEN IF 'FORTH.DAT'
243      , IS NOT AVAILABLE.
244      , - EDIT 'FORTH.MAC' TO SELECT RSX ASSEMBLY. CHANGE THE SEMICOLON
245      , TO COMMENT OUT 'RT11' NOT 'RSX11'. LET 'EIS' BE DEFINED IF
246      , YOU HAVE HARDWARE MULTIPLY/DIVIDE.
247      , - ASSEMBLE, TASK BUILD, AND RUN. TEST AS WITH RT11 ABOVE.
248      , - THE DISK I/O SHOULD WORK IF 'FORTH.DAT' IS IN THE DEFAULT
249      , DEVICE AND DIRECTORY. TEST AS ABOVE.
250      ,
251      ,
252      ,
253      , THE SYSTEM AS SUPPLIED RESERVES 8000. BYTES FOR YOUR FORTH
254      , PROGRAMMING AND STACK. THIS IS ENOUGH FOR SUBSTANTIAL PROJECTS.
255      , (NOTE THAT THE EDITOR, ASSEMBLER, AND STRING PACKAGE, IF LOADED,
256      , USE MORE THAN 5K OF THIS.) TO CHANGE THIS MEMORY SIZE, CHANGE
257      , THE '8000.' WHICH IS IN THE LINES FOLLOWING THE LABEL 'DP:',
258      , NEAR THE END OF THIS PROGRAM. INCIDENTALLY, VERY FEW JOBS
259      , (E.G. RECURSION) WILL EVER USE MORE THAN 100 WORDS OF THIS SPACE
260      , FOR THE STACK; THE REST OF THE SPACE IS AVAILABLE FOR A STRING
261      , STACK (IF USED) OR FOR YOUR PROGRAMS - AND FORTH OBJECT CODE IS
262      , CONSIDERABLY MORE COMPACT THAN ASSEMBLY.
263      ,
264      ,
265      ,
266      , THE FORTH VIRTUAL FILE 'FORTH.DAT' IS USED FOR STORING SOURCE
267      , PROGRAMS (OR DATA). THIS FILE HAS 70 1-K SCREENS (1-70),
268      , I.E. 140 PDP-11 DISK BLOCKS. SCREENS 4 AND 5 ARE USED BY THE
269      , SYSTEM FOR STORING ERROR AND WARNING MESSAGES. SCREENS 6-30
270      , CONTAIN A TEXT EDITOR, ASSEMBLER, STRING PACKAGE, AND MISCELLANEOUS
271      , EXAMPLES. SCREENS 40 THROUGH 47 CONTAIN A BINARY STAND-ALONE
272      , SYSTEM (NOT USED UNDER RT-11 OR RSX-11M). USERS MAY WANT
273      , TO SAVE THEIR SOURCE PROGRAMS AND DATA IN THE BLANK SCREENS.
274      , THE SIZE OF THIS FORTH SCREENS FILE ('FORTH.DAT') CAN BE INCREASED
275      , IF NEEDED. IF THE SYSTEM IS TO BE BOOTED STAND-ALONE, THE LOCATION
276      , OF THE SYSTEM BINARY IMAGE ON THE DISK MUST NOT BE CHANGED;
277      , THEREFORE, IF THE DISK IS TO BE USED TO RUN STAND-ALONE, DO NOT
278      , USE RT-11 TO MOVE 'FORTH.DAT' TO ANOTHER PLACE ON THE DISK.
279      ,
280      ,
281      ,
282      ,
283      ,
284      , NOTE THAT THE RT-11 AND RSX-11M SYSTEMS DO NOT ECHO CHARACTERS
285      , WHICH ARE INPUT FROM THE TERMINAL. INSTEAD, THEY LET THE OPERATING
286      , SYSTEM (RT-11 OR RSX-11M) ECHO THEM. THIS IS DONE SO THAT TYPING
287      , CONVENTIONS WILL BE THE SAME AS THE USER IS FAMILIAR WITH. ALSO,

```

```

288      ; TO AVOID SWAPPING DELAYS, THE RSX VERSION OF 'KEY' READS A LINE OF
289      ; CHARACTERS AT A TIME.
290      ;
291      ;
292      ;
293      ;
294      ; CHANGE THESE LINES TO CONTROL CONDITIONAL ASSEMBLY:
295      ;
296      ; RT11=1      ; COMMENTED OUT UNLESS RT-11
297      ; RSX11=1     ; COMMENTED OUT UNLESS RSX11M
298      000001 ALONE=1 ; COMMENTED OUT UNLESS STAND-ALONE
299      ; EIS=1      ; COMMENTED OUT UNLESS HARDWARE MULTIPLY-DIVIDE
300      ; LINKS=1     ; COMMENTED OUT UNLESS SUBROUTINE LINKAGE FROM
301      ;              ; FORTH TO OTHER LANGUAGES
302      ;
303      ;. PAGE
304      ; *****
305      ;
306      ; VARIATIONS FROM F. I. Q. MODEL
307      ;
308      ; *****
309      ;
310      ;
311      ; 'FIRST' AND 'LIMIT' HAVE BEEN MADE USER VARIABLES, NOT CONSTANTS.
312      ; THEREFORE WHEN THEY ARE USED, 'FIRST @' AND 'LIMIT @' ARE
313      ; REQUIRED.
314      ;
315      ; 'CODE' AND 'FORTH' ARE NOT PURE CODE, SO THEY WERE MOVED TO THE
316      ; END OF THE DICTIONARY. THIS IS SO THE BULK OF THE DICTIONARY
317      ; COULD BE PUT IN PROM OR USED RE-ENTRANTLY.
318      ;
319      ; THE MACHINE-INDEPENDENT I/O SECTION WAS MOVED TO NEAR THE END OF
320      ; THE DICTIONARY, BECAUSE IT IS NOT ALWAYS PURE CODE, AND ALSO TO
321      ; ALLOW THE I/O TO BE REDEFINED WITHOUT REASSEMBLY.
322      ;
323      ; THIS SYSTEM MUST TEST FOR FIRST-TIME-THROUGH TERMINAL AND DISK
324      ; I/O, TO AVOID ERRONEOUS ATTEMPT TO OPEN FILES TWICE AT LATER COLD
325      ; STARTS. IT CLEARS DISK BUFFERS AT COLD START.
326      ;

```

```

328      ; *****
329      ;
330      ;           SET UP REGISTERS AND MACROS.
331      ;
332      ; *****
333      ;
334      ;
335      000002      W=%2           ; TEMPORARY USED BY 'NEXT' MACRO (THE INNER INTERPRETER)
336      000003      U=%3           ; POINTER TO THE USER AREA
337      000004      IP=%4          ; FORTH INSTRUCTION COUNTER
338      000005      B=%5           ; FORTH STACK POINTER
339      000006      RP=SP          ; FORTH RETURN-STACK POINTER
340      ;
341      ; NOTE - CODE ROUTINES CAN USE REGISTERS 0, 1, 4, AND 5, WITHOUT
342      ; RESTORING THEM.
343      ;
344      ;
345      ; MACRO DEFINITIONS
346      ;
347      ;
348      ;
349      ; THE 'HEAD' MACRO CREATES A FORTH DICTIONARY HEADER. ITS ARGUMENTS ARE:
350      ; (1) LENGTH BYTE - THE LENGTH OF THE NAME BEING DEFINED. THE SIGN BIT
351      ; OF THE LENGTH BYTE MUST BE SET, SO THAT THE SYSTEM WILL RECOGNIZE
352      ; THE END OF A VARIABLE-LENGTH NAME FIELD; THEREFORE THE LENGTH BYTE
353      ; IS GIVEN AS 200 OCTAL PLUS THE LENGTH. IF THE OPERATION IS
354      ; IMMEDIATE, THE BIT NEXT TO THE SIGN BIT IS ALSO SET, SO THE LENGTH
355      ; BYTE IS GIVEN AS 300 OCTAL PLUS THE LENGTH.
356      ; (2) NAME - THE NAME OF THE OPERATION BEING DEFINED.
357      ; (3) LCHAR - THE ASCII VALUE OF THE LAST CHARACTER OF THE NAME, WITH THE
358      ; SIGN BIT SET. THE NAME FIELD MUST HAVE AN EVEN LENGTH (INCLUDING
359      ; THE LENGTH BYTE), SO IF THE NUMBER OF CHARACTERS IN THE NAME IS
360      ; EVEN, 'LCHAR' WILL BE GIVEN AS 240 (200 PLUS CODE FOR A SPACE).
361      ; (4) LABEL - THE ASSEMBLY-LANGUAGE LABEL ASSOCIATED WITH THE 'CODE FIELD'
362      ; OF THIS DICTIONARY HEADER. THESE LABELS ARE USED IN THE PRECOMPILED-
363      ; FORTH SECTION OF THE SYSTEM. WHEN POSSIBLE, THE FORTH OPERATION
364      ; NAME ITSELF IS USED AS THE ASSEMBLY LABEL; OTHERWISE AN ABBREVIATION
365      ; IS USED. BY CONVENTION, THESE NAMES ARE LIMITED TO FIVE CHARACTERS,
366      ; FOR CONSISTENCY AMONG VARIOUS ASSEMBLERS FOR DIFFERENT MICROPROCESSORS.
367      ; (THE FORTH IMPLEMENTATION TEAM USES THE SAME LABELS IN ALL OF ITS
368      ; VERSIONS.)
369      ; (5) CODE - POINTER TO THE MACHINE-LANGUAGE "CODE ROUTINE" ASSOCIATED
370      ; WITH THIS OPERATION TYPE OR DATA TYPE. E.G. FOR ANY COLON DEFINITION,
371      ; THIS ARGUMENT IS 'DOCOL', THE LABEL OF A FIVE-INSTRUCTION ASSEMBLY
372      ; ROUTINE WHICH USES THE RETURN STACK TO HANDLE THE NESTED EXECUTION
373      ; OF ANOTHER LEVEL OF FORTH OPERATIONS. FOR ANY CONSTANT, THIS CODE
374      ; ROUTINE IS 'DOCON', AND SIMILARLY FOR ALL OTHER DATA TYPES.
375      ; THE CODE ARGUMENT MAY BE OMITTED. IN THAT CASE, THE 'HEAD'
376      ; MACRO LEAVES THE CODE FIELD POINTING TWO BYTES BEYOND ITSELF, WHERE
377      ; MACHINE-LANGUAGE CODE MUST BEING - AND THE OPERATION SO DEFINED IS
378      ; CALLED A "PRIMITIVE". THE "NUCLEUS SECTION" OF THIS VERSION OF
379      ; FORTH CONTAINS ABOUT 45 PRIMITIVES, FROM WHICH THE WHOLE SYSTEM
380      ; IS BUILT; IN EFFECT, THESE PRIMITIVES DEFINE THE VIRTUAL FORTH
381      ; MACHINE. (A FEW OPERATIONS IN THE "PRECOMPILED FORTH" SECTION
382      ; OF THE SYSTEM HAVE BEEN REPLACED WITH PRIMITIVES, TO OPTIMIZE
383      ; EXECUTION SPEED. AND WHEN A FORTH ASSEMBLER IS ADDED TO THIS
384      ; SYSTEM, USERS WILL BE ABLE TO DEFINE THEIR OWN PRIMITIVES DIRECTLY

```

```

385      ,      IN FORTH, IMMEDIATELY READY FOR EXECUTION. )
386      ,
387      , THE 'HEAD' MACRO CREATES A FORTH HEADER CONSISTING OF
388      , LENGTH BYTE - SIGN BIT SET
389      , NAME OF THE OPERATION - VARIABLE LENGTH - SIGN BIT SET ON LAST CHAR.
390      , LINK FIELD, WHICH POINTS TO THE BEGINNING OF THE PREVIOUS DICTIONARY
391      , HEADER (USED AT COMPILE TIME)
392      , CODE POINTER.
393      ,
394      000000 LINK=0      , LAST LINK FIELD IS 0, INDICATING END OF THE DICTIONARY.
395      ,
396      ,
397      . MACRO HEAD, LENGTH, NAME, LCHAR, LABEL, CODE
398      LINK2=,
399      . BYTE LENGTH
400      . ASCII ^NAME^
401      . EVEN
402      . =. -1
403      . BYTE LCHAR      , LAST CHARACTER OF NAME (OR BLANK FILL),
404      , PASSED IN OCTAL, WITH HIGH BIT SET.
405      . WORD LINK
406      LINK=LINK2
407      LABEL: . IF NB CODE
408      . WORD CODE
409      . IFF
410      . WORD . +2
411      . ENDC
412      . ENDM
413      ,
414      ,
415      ,
416      , THE 'NEXT' MACRO TRANSFERS CONTROL FROM ONE FORTH OPERATION TO THE
417      , 'CODE ROUTINE' OF THE NEXT. NOTICE THAT ONLY TWO INSTRUCTION
418      , EXECUTIONS ARE REQUIRED TO TRANSFER CONTROL FROM USEFUL OPERATIONS
419      , OF ONE FORTH PRIMITIVE TO THOSE OF THE NEXT.
420      ,
421      . MACRO NEXT
422      MOV (IP)+, W
423      JMP @ (W)+
424      . ENDM
425      ,
426      ,
427      , MACRO CALLS
428      ,
429      ,
430      . IFDF RT11
431      . MCALL . RCTRLD, . TTYIN, . TTINR, . TTYOUT, . EXIT, . TRPSET
432      . MCALL . SETTOP, . DSTATUS, . FETCH, . LOOKUP, . READW, . WRITW
433      . ENDC
434      ,
435      ,
436      . IFDF RSX11
437      . MCALL GIOW*C, EXIT*S, ALUN*C, ASTX*S, SVTK*S
438      . MCALL FDBDF*, FDRC*A, FDBK*A, FDOP*A, FRSZ*
439      . MCALL OPEN*M, READ*, WRITE*, WAIT*, CLOSE*
440      . MCALL GIOW*
441      . ENDC

```

```

443
444
445
446
447
448
449
450
451
452
453
454
455
456
457 000000
458 000000 000167 007724
459 000004 000167 010002
460
461
462
463 000010 000011
464 000012 000000
465 000014 015412'
466 000016 000010
467 000020 043436'
468
469
470 000022 035126'
471 000024 043436'
472 000026 043146'
473 000030 000037
474 000032 000000
475
476 000034 015426'
477
478 000036 015426'
479
480 000040 015410'
481 000042 035132'
482 000044 043146'
483 000046 000000
484 000050 000000
485

```

```

; *****
;
; START-UP TABLE
;
; *****
;
; AT STARTUP, MOST OF THESE VALUES ARE MOVED INTO THE USER AREA
; (STARTING AT 'XDP:'); THEY ARE NORMALLY ACCESSED THERE. THE VALUES
; HERE ARE NOT USUALLY CHANGED, BUT THEY MAY BE CHANGED E.G. TO
; CONTROL WHAT HAPPENS AT COLD START. THIS TABLE COULD BE MOVED OUT OF
; LOW MEMORY IF NECESSARY FOR ROM SYSTEMS.
;
;
; OFORTH::
ORIGIN: JMP CENT ; GLOBAL LABEL - NORMALLY NOT USED
; COLD START ENTRY POINT
; JMP WENT ; WARM START ENTRY ADDRESS
;
; NOTE - COLD START WIPEs OUT ANY NEW DICTIONARY DEFINITIONS, AND
; THEN DOES A WARM START. WARM START CLEANS UP STACKS, TERMINAL
; BUFFER, ETC.
;
; .WORD 11 ; CPU
; .WORD 0 ; REVISION
; .WORD TASK-10 ; POINTER TO LATEST WORD DEFINED
; .WORD 10 ; BACKSPACE CHARACTER
; .WORD XUP ; POINTER TO USER AREA
;
; NOTE - THE USER AREA IS A HOOK IN THIS SYSTEM TO ALLOW MULTITASKING
; TO BE ADDED LATER.
;
; .WORD XSO ; POINTER TO BEGINNING OF THE STACK
; .WORD XRO ; POINTER TO BEGINNING OF RETURN STACK
; .WORD XTIB ; POINTER TO TERMINAL INPUT BUFFER
; .WORD 37 ; MAXIMUM NAME-FIELD WIDTH, NORMALLY 31
; .WORD 0 ; WARNING MODE; 0=ERROR #, 1=DISK MESSAGE
;
; NOTE - WARNING MODE INITIALIZED TO ZERO, IN CASE DISK ISN'T UP.
; .WORD XDP ; FENCE TO PROTECT AGAINST ACCIDENTAL
; 'FORGET' OF THE SYSTEM.
; .WORD XDP ; POINTER TO NEXT AVAILABLE DICTIONARY
; LOCATION (RETURNED BY 'HERE').
; .WORD XXVOC ; POINTER TO INITIAL VOCABULARY LINK
; .WORD DSKBUF ; INITIALIZE 'FIRST'
; .WORD ENDBUF ; INITIALIZE 'LIMIT'
; .WORD 0 ; AVAILABLE
; .WORD 0 ; AVAILABLE
;

```

```

487                                     ; *****
488                                     ;
489                                     ;       NUCLEUS
490                                     ;
491                                     ; *****
492                                     ;
493                                     ;
494                                     ;
495                                     ; THE NUCLEUS CONTAINS THE PRIMITIVES FROM WHICH THE SYSTEM IS BUILT.
496                                     ;
497                                     ;
498                                     ;
499                                     ;
500 000052                                HEAD    203,LIT,324,LIT                ; ***** LIT
501                                     ; USED ONLY BY COMPILER.  PUSH FOLLOWING LITERAL ONTO STACK.
502 000062 012445                        MOV     (IP)+,-(S)
503 000064                                NEXT
504                                     ;
505 000070                                HEAD    207,EXECUTE,305,EXEC          ; ***** EXECUTE
506                                     ; EXECUTE FORTH WORD WHOSE CODE ADDRESS IS ON STACK
507 000104 012502                        MOV     (S)+,W
508 000106 000132                        JMP     @ (W)+
509                                     ;
510                                     ;
511 000110                                HEAD    206,BRANCH,240,BRAN          ; ***** BRANCH
512                                     ; USED ONLY BY COMPILER.  FORTH BRANCH TO ADDRESS WHICH FOLLOWS.
513 000124 061404                        ADD     (IP),IP
514 000126                                NEXT
515                                     ;
516 000132                                HEAD    207,@BRANCH,310,ZBRAN        ; ***** @BRANCH
517                                     ; USED ONLY BY COMPILER.  FORTH BRANCH IF TOP OF STACK
518                                     ; IS ZERO (FALSE).
519 000146 005725                        TST     (S)+
520 000150 001003                        BNE     3*
521 000152 061404                        ADD     (IP),IP
522 000154                                NEXT
523 000160 062704 000002                3*:    ADD     #2,IP
524 000164                                NEXT
525                                     ;
526 000170                                HEAD    206,(LOOP),240,XLOOP          ; ***** (LOOP)
527                                     ; USED ONLY BY COMPILER.  INCREMENT LOOP INDEX BY 1, BRANCH
528                                     ; IF BELOW LIMIT.
529 000204 005216                        INC     (RP)
530 000206 021666 000002                CMP     (RP),2(RP)
531 000212 002003                        BGE     1*
532 000214 061404                        ADD     (IP),IP
533 000216                                NEXT
534 000222 062706 000004                1*:    ADD     #4,RP
535 000226 062704 000002                ADD     #2,IP
536 000232                                NEXT
537                                     ;
538 000236                                HEAD    207,(+LOOP),251,XPLOD        ; ***** (+LOOP)
539                                     ; USED ONLY BY COMPILER.  INCREMENT LOOP INDEX BY TOP OF STACK,
540                                     ; MAYBE BRANCH.
541 000252 061516                        ADD     (S),(RP)
542 000254 005725                        TST     (S)+
543 000256 002414                        BLT     2*

```

```

544 000260 026616 000002      CMP      2(RP), (RP)
545 000264 003403          BLE      1$
546 000266 061404          ADD      (IP), IP
547 000270          NEXT
548 000274 062706 000004      1$:      ADD      #4, RP
549 000300 062704 000002      ADD      #2, IP
550 000304          NEXT
551 000310 021666 000002      2$:      CMP      (RP), 2(RP)
552 000314 003767          BLE      1$
553 000316 061404          ADD      (IP), IP
554 000320          NEXT
555
556 000324          HEAD      204, (DO), 240, XDO
557          USED ONLY BY COMPILER. SET UP 'DO' LIMIT AND INDEX.
558 000336 016546 000002      MOV      2(S), -(RP)
559 000342 011546          MOV      (S), -(RP)
560 000344 062705 000004      ADD      #4, S
561 000350          NEXT
562
563 000354          HEAD      201, 1, 311, 1
564          RETURN CURRENT LOOP INDEX TO STACK.
565 000362 011645          MOV      (RP), -(S)
566 000364          NEXT
567
568 000370          HEAD      205, DIOIT, 324, DIOIT
569          USED BY COMPILER.
570          ( ( ASCII-DIOIT BASE ==> DIOIT-VALUE TRUE (OR FALSE))
571 000402 162765 000002      SUB      #60, 2(S)
572 000410 026527 000002      CMP      2(S), #11
573 000416 003407          BLE      1$
574 000420 162765 000007      SUB      #7, 2(S)
575 000426 026527 000002      CMP      2(S), #12
576 000434 002412          BLT      2$
577 000436 005765 000002      TST      2$
578 000442 002407          BLT      2$
579 000444 026515 000002      CMP      2(S), (S)
580 000450 002004          BGE      2$
581 000452 012715 000001      MOV      #1, (S)
582 000456          NEXT
583 000462 062705 000002      1$:      ADD      #2, S
584 000466 005015          CLR      (S)
585 000470          NEXT
586
587          HEAD      206, (FIND), 240, PFIND
588 000474          USED BY COMPILER. FIND A WORD IN THE DICTIONARY.
589          ( ( STRING-ADDRESS NFA ==> PFA LENGTH TRUE (OR FALSE)).
590          STRING-ADDRESS IS ADDRESS OF THE LENGTH BYTE OF THE
591          STRING BEING SOUGHT. NFA IS NAME-FIELD ADDRESS OF
592          WORD IN DICTIONARY WHERE SEARCH BEGINS. PFA IS
593          PARAMETER-FIELD ADDRESS OF THE DICTIONARY ENTRY
594          WHICH IS FOUND. IF WORD NOT FOUND, ONLY ONE RESULT
595          (0, FALSE) IS RETURNED.
596          SETUP - GET ARGS, PRESERVE NEEDED REGISTERS
597 000510 012500          MOV      (S)+, R0
598 000512 012501          MOV      (S)+, R1
599 000514 010546          MOV      R5, -(RP)
600          PRESERVE REGISTERS

```

```

601 000516 010446      MOV R4,-(RP)
602 000520 010346      MOV R3,-(RP)
603 000522 005046      CLR -(RP)
604 000524 011102      ; PREPARE R2 FOR FAST COMPARE
605 000526 042702 100200 MOV (R1),R2
606 000532 000532      BIC #100200,R2
607
608
609
610
611
612 000532 011003      ; FAST TEST TO ELIMINATE MOST WORDS
613 000534 042703      ; COMPARE FIRST WORD TO SPECIALLY PREPARED R2
614 000540 020203      ; THEN INCREMENT TO FIND END OF NAME.
615 000542 001406      FAST: MOV (R0),R3
616 000544 005720      BIC #100300,R3
617 000546 100376      CMP R2,R3
618
619 000550 005710      BEQ NOFAST
620 000552 001435      XMATCH: TST (R0)+
621 000554 011000      BPL XMATCH
622 000556 000765      ; R0 NOW POINTS TO LINK
623
624
625 000560 011016      ; NO FAST ELIMINATION POSSIBLE
626 000562 010105      ; BRANCH HERE IF NO MATCH THIS TIME
627 000564 000407      ;
628
629 000566 005725      ; END OF FAST ELIMINATION TEST
630 000570 011504      ;
631 000572 011003      NOFAST: MOV (R0),(RP)
632 000574 042703      MOV R1,R5
633 000600 020304      BR NOFAST1
634 000602 001360      ; NOW DO THE MAIN LOOP TO CHECK FOR MATCH
635 000604 032720      MLOOP: TST (R5)+
636 000610 001766      MOV (R5),R4
637
638 000612 012602      MOV (R0),R3
639 000614 012603      MOV (R0),R3
640 000616 012604      BIC #100000,R3
641 000620 012605      CMP R3,R4
642 000622 062700      BNE XMATCH
643 000626 010045      ; GET PARAMETER FIELD ADDRESS
644 000630 042702 177400 MOV #177400,R2
645 000634 010245      ; R2 CONTAINS LENGTH BYTE
646 000636 012745      MOV #1,-(S)
647 000642 005726      NEXT
648 000646 012603      ; POP LENGTH BYTE
649 000650 012603      ; RESTORE REGISTERS
650 000652 012604      MOV (RP)+,R4
651 000654 012605      MOV (RP)+,R5
652 000656 005045      CLR -(S)
653
654 000660      ; REPLACE LENGTH BYTE WITH
655      ; FAILURE FLAG.
656 000664      ; WE ARE DONE - FAILURE TO FIND
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

HEAD 207, ENCLOSE, 305, ENCL
 USED BY COMPILER. BREAK NEXT WORD OUT OF INPUT BUFFER. ***** ENCLOSE

```

658
659 000700 011500
660 000702 016501 000002
661 000706 162705 000004
662 000712 122100
663 000714 001776
664 000716 162701 000001
665 000722 010165 000004
666 000726 105711
667 000730 001420
668 000732 122100
669 000734 001374
670 000736 010115
671 000740 162701 000001
672 000744 010165 000002
673 000750 016501 000006
674 000754 160115
675 000756 160165 000002
676 000762 160165 000004
677 000766
678 000772 010115
679 000774 020165 000004
680 001000 001361
681 001002 062701 000001
682 001006 000756
683
684
685
686
687
688
689
690 001010
691
692 001022
693
694 001032
695
696 001050
697
698
699
700
701
702 001060
703
704 001072 005715
705 001074 001407
706 001076 016500 000002
707 001102 016501 000004
708 001106 112120
709 001110 005315
710 001112 001375
711 001114 062705 000006
712 001120
713
714

```

```

; ( START-ADDRESS DELIMITER ==> ADDRESS OFFSET END NEXT-CHARACTER)
      MOV      (S),R0      ; DELIMITER
      MOV      2(S),R1     ; STARTING ADDRESS
      SUB      #4,S        ; MAKE SPACE FOR RESULTS
ENC1:  CMPB    (R1)+,R0     ; SKIP OVER LEADING DELIMITERS
      BEQ      ENC1
      SUB      #1,R1
      MOV      R1,4(S)
ENC2:  TSTB    (R1)         ; TEST FOR NULL
      BEQ      ENC4
      CMPB    (R1)+,R0     ; NOT NULL, SO FIND END OF TOKEN
      BNE      ENC2
      MOV      R1,(S)
      SUB      #1,R1
ENC3:  MOV      R1,2(S)     ; FINISH UP AND RETURN
      MOV      6(S),R1
      SUB      R1,(S)
      SUB      R1,2(S)
      SUB      R1,4(S)
      NEXT
ENC4:  MOV      R1,(S)      ; HANDLE NULL CASE
      CMP      R1,4(S)
      BNE      ENC3
      ADD      #1,R1
      BR       ENC3

```

```

;
;
; THE NEXT 4 HEADERS POINT TO INSTALLATION-DEPENDENT TERMINAL I/O
; ROUTINES.
;

```

```

      HEAD     204,EMIT,240,EMIT,PEMIT      ; ***** EMIT
;
      HEAD     203,KEY,331,KEY,PKEY        ; ***** KEY
;
      HEAD     211,?TERMINAL,314,QTERM,PQTER ; ***** ?TERMINAL
;
      HEAD     202,CR,240,CR,PCR           ; ***** CR
;
;
;
      HEAD     205,CMOVE,305,CMOVE         ; ***** CMOVE
; MOVE BYTES IN MEMORY. ( FROM TO N ==>)
      TST      (S)
      BEQ      2*           ; NO MOVE
      MOV      2(S),R0
      MOV      4(S),R1
1*:  MOVB     (R1)+,(R0)+
      DEC      (S)
      BNE      1*
2*:  ADD      #6,S
      NEXT
;
;

```

```

715 001124      HEAD 202,U*,240,USTAR , ***** U*
716             ( N1 N2 ==> PRODUCT).  PRODUCT IS 32-BIT DOUBLE INTEGER,
717             , HIGH WORD TOP.
718             , THIS MUST BE UNSIGNED MULTIPLICATION.
719 001134      004767 000004      JSR PC,UMULT
720 001140      NEXT
721 001144
722
723 001144      012502      UMULT:
724 001146      012746      , THE VALUES TO MULTIPLY ARE ON THE STACK.
725 001152      005000      MOV (S)+,R2
726 001154      005001      MOV #20,-(RP)      , SET LOOP COUNT
727 001156      006101      CLR R0
728 001160      006100      CLR R1
729 001162      006102      ROL R0
730 001164      103002      ROL R1
731 001166      061501      ROL R2
732 001170      005500      BCC 1$
733 001172      005316      ADD (S),R1
734 001174      001370      ADC R0
735 001176      010115      DEC (RP)
736 001200      010045      BNE 2$
737 001202      005726      MOV R1,(S)
738 001204      000207      MOV R0,-(S)
739      RTS      , POP TEMPORARY
740 001206      PC
741      HEAD 202,U/,240,USLAS , ***** U/
742 001216      004767 000004      , THIS DIVISION MUST BE UNSIGNED.
743 001222      JSR PC,UDIV
744 001226      NEXT
745
746 001226      012502      UDIV:
747 001230      012500      , THE VALUES TO DIVIDE ARE ON THE STACK
748 001232      012501      MOV (S)+,R2 , DIVISOR
749 001234      012745      MOV (S)+,R0
750 001240      006301      MOV #20,-(S)      , LOOP COUNT
751 001242      006100      ASL R1
752 001244      001405      ROL R0
753 001246      160200      BEQ 2$      , NO NEED TO SUBTRACT
754 001250      005201      SUB R2,R0
755 001252      103002      INC R1
756 001254      060200      BCC 2$
757 001256      005301      ADD R2,R0
758 001260      005315      DEC R1
759 001262      001366      BNE 1$      , MUST RESTORE
760 001264      005725      TST (S)+      , LOOP SIXTEEN TIMES
761 001266      010045      R0,-(S) , POP TO DISCARD COUNT
762 001270      010145      MOV R1,-(S) , QUOTIENT
763 001272      000207      RTS
764
765 001274      ,
766      HEAD 203,AND,304,AND , ***** AND
767 001304      005115      ( N1 N2 ==> N3).
768 001306      042515      BITWISE AND.
769 001310      BIC (S), (S)
770      NEXT
771 001314      HEAD 202,OR,240,OR , ***** OR

```

```

772 001324 052515      BIS      (S)+, (S)
773 001326      NEXT
774
775 001332      HEAD     203, XOR, 322, XOR
776      . IFDF
777      MOV      (S)+, R0
778      XOR      R0, (S)
779      . IFF
780 001342 011546      MOV      (S), -(RP)
781 001344 046516      BIC      2(S), (RP)
782 001350 042515      BIC      (S)+, (S)
783 001352 052615      BIS      (RP)+, (S)
784      . ENDC
785 001354      NEXT
786
787 001360      HEAD     203, SP0, 300, SPAT
788 001370 010501      MOV      S, R1
789 001372 010145      MOV      R1, -(S)
790 001374
791
792 001400      HEAD     203, SP!, 241, SPSTO
793 001410 016305 000006  MOV      6(U), S
794
795 001414      NEXT
796
797 001420      HEAD     203, RP!, 241, RPSTO
798 001430 016706 176370  MOV      ORIGIN+24, RP
799 001434
800
801 001440      HEAD     202, <(S)>, 240, SEMIS
802 001450 012604      MOV      (RP)+, IP
803 001452
804
805 001456      HEAD     205, LEAVE, 305, LEAVE
806 001470 011666 000002  MOV      (RP), 2(RP)
807 001474
808
809 001500      HEAD     202, ^/OR/, 240, TOR
810 001510 012546      MOV      (S)+, -(RP)
811 001512
812
813 001516      HEAD     202, R>, 240, FROMR
814 001526 012645      MOV      (RP)+, -(S)
815 001530
816
817 001534      HEAD     201, R, 322, R
818 001542 011645      MOV      (RP), -(S)
819 001544
820
821 001550      HEAD     202, 0=, 240, ZEGU
822 001560 005715      TST      (S)
823 001562 001402      BEQ      1$
824 001564 005015      CLR      (S)
825 001566 000402      BR
826 001570 012715 000001  MOV      #1, (S)
827 001574
828 001574

```

, ***** XOR
 , ***** SP0
 , ***** SP!
 , OFFSET 6 IN USER AREA
 , ***** RP!
 , ***** IS
 , ***** LEAVE
 , ***** >R
 , ***** R>
 , ***** R
 , ***** 0=

1\$:

2\$:

```

829
830 001600      HEAD      202, 0<, 240, ZLESS      , ***** 0<
831 001610 005715      TST      (S)
832 001612 100402      BMI      1$
833 001614 005015      CLR      (S)
834 001616 000402      BR      2$
835 001620 012715 000001      1$: MOV      #1, (S)
836 001624      2$:
837 001624      NEXT
838
839 001630      HEAD      201, +, 253, PLUS      , ***** +
840 001636 062515      ADD      (S)+, (S)
841 001640      NEXT
842
843 001644      HEAD      202, D+, 240, DPLUS      , ***** D+
844 001654 066565 000002 000006      ADD      2(S), 6(S)      , ADD LOW
845 001662 005565 000004      ADC      4(S)
846 001666 061565 000004      ADD      (S), 4(S)      , ADD HIGH
847 001672 062705 000004      ADD      #4, S
848 001676      NEXT
849
850 001702      HEAD      205, MINUS, 323, MINUS      , ***** MINUS
851      CHANGE SIGN.
852 001714 005415      NEQ      (S)
853 001716      NEXT
854
855 001722      HEAD      206, DMINUS, 240, DMINU      , ***** DMINUS
856      CHANGE SIGN OF DOUBLE INTEGER WORD ON STACK.
857 001736 005415      NEQ      (S)
858 001740 005465 000002      NEQ      2(S)
859 001744 005615      SBC      (S)
860 001746      NEXT
861
862 001752      HEAD      204, OVER, 240, OVER      , ***** OVER
863      ( N1 N2 ==> N1 N2 N1)
864 001764 016545 000002      MOV      2(S), -(S)
865 001770      NEXT
866
867 001774      HEAD      204, DROP, 240, DROP      , ***** DROP
868 002006 062705 000002      ADD      #2, S
869 002012      NEXT
870
871 002016      HEAD      204, SWAP, 240, SWAP      , ***** SWAP
872 002030 016501 000002      MOV      2(S), R1
873 002034 011565 000002      MOV      (S), 2(S)
874 002040 010115      MOV      R1, (S)
875 002042      NEXT
876
877 002046      HEAD      203, DUP, 320, DUP      , ***** DUP
878 002056 011545      MOV      (S), -(S)
879 002060      NEXT
880
881 002064      HEAD      202, +!, 240, PSTOR      , ***** +!
882      ADD NUMBER SECOND ON STACK TO ADDRESS ON TOP.
883 002074 066575 000002 000000      ADD      2(S), 0(S)
884 002102 062705 000004      ADD      #4, S
885 002106      NEXT

```

```

886      002112      016545      0000002      0000000
887      002126      117515      0000000
888      002132      011546      0000002      0000000
889      002136      046516      0000002      0000000
890      002140      042515      0000002      0000000
891      002144      052615      0000002      0000000
892      002148      016545      0000002      0000000
893      002152      116575      0000002      0000000
894      002156      062705      0000006      0000000
895      002160      0000000
896      002164      0000000
897      002168      0000000
898      002172      017515      0000000
899      002176      0000000
900      002180      117501      0000000      0000000
901      002184      042701      177400
902      002188      010115
903      002192      0000000
904      002196      0000000
905      002200      0000000
906      002204      0000000
907      002208      0000000
908      002212      0000000
909      002216      0000000
910      002220      0000000
911      002224      016575      0000002      0000000
912      002228      062705      0000004
913      002232      0000000
914      002236      0000000
915      002240      0000000
916      002244      116575      0000002      0000000
917      002248      062705      0000004
918      002252      0000000
919      002256      0000000
920      002260      0000000
921      002264      0000000

```

; HEAD 206, TOGGLE, 240, TOGGLE ; ***** TOGGLE
 ; (BYTE-ADDRESS BIT-PATTERN ==>) EXCLUSIVE-OR INTO MEMORY BYTE.
 MOV 2(S), -(S) ; PUSH THE BYTE
 MOVB 2(S), (S) ; TO BE TOGGLED
 ; AVOID USING 'XOR' INSTRUCTION - NOT AVAILABLE ON ALL PDP-11
 MOV (S), -(RP)
 BIC 2(S), (RP)
 BIC (S)+, (S)
 BIS (RP)+, (S)
 MOV 2(S), -(S)
 MOVB 2(S), 6(S)
 ADD #6, S
 NEXT
 ; SET UP RETURN ADDRESS
 ; PUT THE TOGGLED BYTE BACK TO MEM.
 ; ADJUST STACK POINTER
 ; ***** @
 HEAD 201, 6, 300, AT
 MOV 6(S), (S)
 NEXT
 ; ***** C@
 HEAD 202, C@, 240, CAT
 MOV 6(S), R1
 BIC #177400, R1
 MOV R1, (S)
 NEXT
 ; ***** !
 HEAD 201, !, 241, STORE
 MOV 2(S), 6(S)
 ADD #4, S
 NEXT
 ; ***** C!
 HEAD 202, C!, 240, C@TOR
 MOV 2(S), 6(S)
 ADD #4, S
 NEXT

```

923                                     ; *****
924                                     ;
925                                     ;       PRE-COMPILED FORTH SECTION
926                                     ;
927                                     ; *****
928                                     ;
929                                     ;
930                                     ;
931                                     ; NOTE - A FEW OF THE FOLLOWING OPERATIONS HAVE BEEN
932                                     ; CONVERTED TO CODE FOR SPEED.  HOWEVER, THE WORD ORDER
933                                     ; IN THE DICTIONARY HAS NOT BEEN CHANGED.
934                                     ;
935 002310                                HEAD      301, , 272, COLON, DOCOL      ; ***** :
936 002316 004202' 004074' 003150'      . WORD   GEXEC, SCSP, CURR, AT, CONT, STORE, CREAT, RBRAC, PSCOD
937 002324 002176' 003132' 002242'
938 002332 006700' 004412' 004562'
939 002340 010446      DOCOL:  MOV      IP, -(RP)
940 002342 010204      MOV      W, IP
941 002344      NEXT
942                                     ;
943                                     ;
944 002350                                HEAD      301, < , >, 273, SEMI, DOCOL      ; ***** ;
945 002356 004256' 004346' 001446'      . WORD   GCSP, COMP, SEMIS, SMUDG, LBRAC, SEMIS
946 002364 004440' 004374' 001446'
947                                     ;
948                                     ;
949 002372                                HEAD      210, CONSTANT, 240, CON, DOCOL      ; ***** CONSTANT
950 002410 006700' 004440' 003416'      . WORD   CREAT, SMUDG, COMMA, PSCOD
951 002416 004562'
952 002420 011245      DOCON:  MOV      (W), -(S)
953 002422      NEXT
954                                     ;
955                                     ;
956 002426                                HEAD      210, VARIABLE, 240, VAR, DOCOL      ; ***** VARIABLE
957 002444 002406' 004562'      . WORD   CON, PSCOD
958 002450 010245      DOVAR:  MOV      W, -(S)
959 002452      NEXT
960                                     ;
961                                     ;
962                                     ;
963                                     ;
964                                     ;
965 002456                                HEAD      204, USER, 240, USER, DOCOL      ; ***** USER
966 002470 002406' 004562'      . WORD   CON, PSCOD
967 002474 011245      DOUSE:  MOV      (W), -(S)
968 002476 060315      ADD      U, (S)
969 002500      NEXT
970                                     ;
971                                     ;
972                                     ;
973                                     ;
974                                     ;
975                                     ;
976                                     ;
977                                     ;
978                                     ;
979                                     ;
980                                     ;
981                                     ;
982                                     ;
983                                     ;
984                                     ;
985                                     ;
986                                     ;
987                                     ;
988                                     ;
989                                     ;
990                                     ;
991                                     ;
992                                     ;
993                                     ;
994                                     ;
995                                     ;
996                                     ;
997                                     ;
998                                     ;
999                                     ;
1000                                    ;

```

```

976                                     ;
977 002544                             HEAD    202,BL,240,BL,DOCON          ; ***** BL
978                                     ; BLANK.
979 002554 000040                       .WORD    40
980                                     ;
981 002556                             HEAD    203,C/L,314,CL,DOCON          ; ***** C/L
982                                     ; # OF CHARACTERS PER LINE
983 002566 000100                       .WORD    100
984                                     ;
985                                     ; 'FIRST' AND 'LIMIT' MOVED TO USER AREA
986                                     ;
987 002570                             HEAD    205,B/BUF,306,BBUF,DOCON          ; ***** B/BUF
988                                     ; BYTES PER DISK-BLOCK BUFFER.
989 002602 002000                       .WORD    1024.
990                                     ;
991 002604                             HEAD    205,B/SCR,322,BSCR,DOCON          ; ***** B/SCR
992                                     ; DISK BLOCKS PER FORTH SCREEN.
993 002616 000001                       .WORD    1
994                                     ;
995 002620                             HEAD    207,+ORIGIN,316,PORIG,DOCOL          ; ***** +ORIGIN
996                                     ; RETURNS ADDRESS, GIVEN OFFSET FROM ORIGIN.
997 002634 000060' 000000' 001634'      .WORD    LIT,ORIGIN,PLUS,SEMIS
998                                     ;
999                                     ; USER VARIABLES
1000                                     ;
1001 002644                             HEAD    202,80,240,BZERO,DOUSE          ; ***** 80
1002                                     ; STACK ORIGIN.
1003 002654 000006                       .WORD    6
1004                                     ;
1005 002656                             HEAD    202,R0,240,RZERO,DOUSE          ; ***** R0
1006                                     ; RETURN STACK ORIGIN.
1007 002666 000010                       .WORD    10
1008                                     ;
1009 002670                             HEAD    203,TIB,302,TIB,DOUSE          ; ***** TIB
1010                                     ; TERMINAL INPUT BUFFER.
1011 002700 000012                       .WORD    12
1012                                     ;
1013 002702                             HEAD    205,WIDTH,310,WIDTH,DOUSE          ; ***** WIDTH
1014                                     ; MAXIMUM NAME LENGTH (DEFAULT, 31 CHARACTERS).
1015 002714 000014                       .WORD    14
1016                                     ;
1017 002716                             HEAD    207,WARNING,307,WARN,DOUSE          ; ***** WARNING
1018                                     ; WARNING MODE (DEFAULT, GIVE MESSAGE NUMBER AT ERROR
1019                                     ; OR WARNING CONDITION, DON'T GO TO DISK FOR MESSAGE).
1020 002732 000016                       .WORD    16
1021                                     ;
1022 002734                             HEAD    205,FENCE,305,FENCE,DOUSE          ; ***** FENCE
1023                                     ; PREVENTS 'FORGET' BELOW THIS 'FENCE' SETTING.
1024 002746 000020                       .WORD    20
1025                                     ;
1026 002750                             HEAD    202,DP,240,DP,DOUSE          ; ***** DP
1027                                     ; DICTIONARY POINTER TO NEXT AVAILABLE SPACE.
1028 002760 000022                       .WORD    22
1029                                     ;
1030 002762                             HEAD    210,VOC-LINK,240,VOCL,DOUSE          ; ***** VOC-LINK
1031                                     ; VOCABULARY LINK (MAINLY FOR FUTURE USE).

```

```

1032 003000 000024      . WORD  24
1033
1034 003002      ,      HEAD    205,FIRST,324,FIRST,DOUSE      , ***** FIRST
1035      ADDRESS OF BEGINNING OF DISK BUFFER.
1036 003014 000026      . WORD  26
1037
1038 003016      ,      HEAD    205,LIMIT,324,LIMIT,DOUSE      , ***** LIMIT
1039      ADDRESS JUST BEYOND END OF DISK BUFFERS.
1040 003030 000030      . WORD  30
1041
1042      , POSITIONS 32 AND 34 ARE AVAILABLE FOR EXPANSION.
1043      , THEY ARE INITIALIZED FROM BOOT-UP TABLE, AT COLD START.
1044
1045 003032      ,      HEAD    203,BLK,313,BLK,DOUSE      , ***** BLK
1046      CURRENT DISK BLOCK BEING LOADED (0=TERMINAL)
1047 003042 000036      . WORD  36
1048
1049 003044      ,      HEAD    202,IN,240,IN,DOUSE      , ***** IN
1050      OFFSET IN TERMINAL INPUT BUFFER.
1051 003054 000040      . WORD  40
1052
1053 003056      ,      HEAD    203,OUT,324,OUT,DOUSE      , ***** OUT
1054      OFFSET IN OUTPUT LINE.
1055 003066 000042      . WORD  42
1056
1057 003070      ,      HEAD    203,SCR,322,SCR,DOUSE      , ***** SCR
1058      CURRENT FORTH DISK SCREEN.
1059 003100 000044      . WORD  44
1060
1061 003102      ,      HEAD    206,OFFSET,240,OFFSET,DOUSE      , ***** OFFSET
1062      OFFSET TO GET TO ANOTHER DISK DRIVE.
1063 003116 000046      . WORD  46
1064
1065 003120      ,      HEAD    207,CONTEXT,324,CONT,DOUSE      , ***** CONTEXT
1066 003134 000050      . WORD  50
1067
1068 003136      ,      HEAD    207,CURRENT,324,CURR,DOUSE      , ***** CURRENT
1069 003152 000052      . WORD  52
1070
1071 003154      ,      HEAD    205,STATE,305,STATE,DOUSE      , ***** STATE
1072 003166 000054      . WORD  54
1073
1074 003170      ,      HEAD    204,BASE,240,BASE,DOUSE      , ***** BASE
1075 003202 000056      . WORD  56
1076
1077 003204      ,      HEAD    203,DPL,314,DPL,DOUSE      , ***** DPL
1078      OFFSET OF DECIMAL POINT AFTER DOUBLE-INTEGGER INPUT.
1079 003214 000060      . WORD  60
1080
1081 003216      ,      HEAD    203,FLD,304,FLD,DOUSE      , ***** FLD
1082      OUTPUT FIELD WIDTH.
1083 003226 000062      . WORD  62
1084
1085 003230      ,      HEAD    203,CSP,320,CSP,DOUSE      , ***** CSP
1086      USED BY COMPILER TO HOLD CURRENT STACK POSITION,
1087      FOR ERROR CHECKING.
1088 003240 000064      . WORD  64

```

```

1089
1090 003242      ,      HEAD      202, R#, 240, RNUM, DOUSE      , ***** R#
1091      ,      CURSOR POSITION (FOR SOME EDITORS).
1092 003252 000066      . WORD      66
1093
1094 003254      ,      HEAD      203, HLD, 304, HLD, DOUSE      , ***** HLD
1095      ,      POINTS TO LAST CHARACTER HELD IN 'PAD'
1096 003264 000070      . WORD      70
1097
1098 003266      ,      HEAD      203, USE, 305, USE, DOUSE      , ***** USE
1099 003276 000072      . WORD      72
1100
1101 003300      ,      HEAD      204, PREV, 240, PREV, DOUSE      , ***** PREV
1102 003312 000074      . WORD      74
1103
1104
1105      ,      END OF USER AREA
1106
1107
1108 003314      ,      HEAD      202, 1+, 240, ONEP      , ***** 1+
1109 003324 005215      INC      (S)
1110 003326      NEXT
1111
1112 003332      ,      HEAD      202, 2+, 240, TWOP      , ***** 2+
1113 003342 062715 000002      ADD      #2, (S)
1114 003346      NEXT
1115
1116 003352      ,      HEAD      204, HERE, 240, HERE, DOCOL      , ***** HERE
1117 003364 002756' 002176' 001446'      . WORD      DP, AT, SEMIS
1118
1119 003372      ,      HEAD      205, ALLOT, 324, ALLOT, DOCOL      , ***** ALLOT
1120 003404 002756' 002072' 001446'      . WORD      DP, PSTOR, SEMIS
1121
1122 003412      ,      HEAD      201, <, >, 254, COMMA, DOCOL      , ***** ,
1123 003420 003362' 002242' 002530'      . WORD      HERE, STORE, TWO, ALLOT, SEMIS
1124 003426 003402' 001446'
1124
1125      ,      THIS SYSTEM DOES NOT USE 'C,'
1126
1127 003432      ,      HEAD      201, -, 255, SUB      , ***** -
1128 003440 162515      SUB      (S)+, (S)
1129 003442      NEXT
1130
1131 003446      ,      HEAD      201, =, 275, EQUAL      , ***** =
1132 003454 026525 000002      CMP      2(S), (S)+
1133 003460 001402      BEQ      1#
1134 003462 005015      CLR      (S)
1135 003464 000402      BR      2#
1136 003466 012715 000001      1#: MOV      #1, (S)
1137 003472      2#: NEXT
1138
1139 003476      ,      HEAD      201, ^/</, 274, LESS      , ***** <
1140 003504 026525 000002      CMP      2(S), (S)+
1141 003510 002402      BLT      1#
1142 003512 005015      CLR      (S)
1143 003514 000402      BR      2#
1144 003516 012715 000001      1#: MOV      #1, (S)

```

```

1145 003522      20:  NEXT
1146      ,
1147 003526      HEAD  201, ^/>/, 276, GREAT      , ***** >
1148 003534 026525 000002      CMP  2(S), (S)+
1149 003540 003002      BGT   1#
1150 003542 005015      CLR   (S)
1151 003544 000402      BR    2#
1152 003546 012715 000001      10:  MOV  #1, (S)
1153 003552      20:  NEXT
1154      ,
1155 003556      HEAD  203, ROT, 324, ROT      , ***** ROT
1156 003566 011500      MOV  (S), R0
1157 003570 016515 000004      MOV  4(S), (S)
1158 003574 016565 000002 000004      MOV  2(S), 4(S)
1159 003602 010065 000002      MOV  R0, 2(S)
1160 003606      NEXT
1161      ,
1162 003612      HEAD  205, SPACE, 305, SPACE, DOCOL      , ***** SPACE
1163 003624 000060' 000040 001020'      . WORD  LIT, 40, EMIT, SEMIS
1164 003632 001446'
1165 003634      ,
1166 003646 005715      HEAD  204, -DUP, 240, DDUP      , ***** -DUP
1167 003650 001401      TST   (S)
1168 003652 011545      BEQ   1#
1169 003654      MOV  (S), -(S)
1170      10:  NEXT
1171      ,
1172      HEAD  210, TRAVERSE, 240, TRAV, DOCOL      , ***** TRAVERSE
1173      , MOVE (FORWARDS OR BACKWARDS) ACROSS A (VARIABLE LENGTH)
1174      , DICTIONARY NAME FIELD.
1175 003700 002026'      . WORD  SWAP
1176 003706 001762' 001634' 000060' XXN1: . WORD  OVER, PLUS, LIT, 177, OVER, CAT, LESS, ZBRAN, XXN1-.
1177 003714 000177 001762' 002216'
1178 003722 003502' 000144' 177760      . WORD  SWAP, DROP, SEMIS
1179 003722 002026' 002004' 001446'
1180      ,
1181      HEAD  206, LATEST, 240, LATES, DOCOL      , ***** LATEST
1182 003730 003150' 002176' 002176'      . WORD  CURR, AT, AT, SEMIS
1183 003744 001446'
1184      ,
1185      THE NEXT 4 OPERATORS CAN DEPEND ON COMPUTER WORD SIZE.
1186      THEY CONVERT ADDRESSES WITHIN THE NAME FIELDS OF FORTH
1187      DICTIONARY ENTRIES.
1188 003754      HEAD  203, LFA, 301, LFA, DOCOL      , ***** LFA
1189 003764 000060' 000004 003436'      . WORD  LIT, 4, SUB, SEMIS
1190 003772 001446'
1191      ,
1192 003774      HEAD  203, CFA, 301, CFA, DOCOL      , ***** CFA
1193 004004 002530' 003436' 001446'      . WORD  TWO, SUB, SEMIS
1194      ,
1195 004012      HEAD  203, NFA, 301, NFA, DOCOL      , ***** NFA
1196 004022 000060' 000005 003436'      . WORD  LIT, 5, SUB, LIT, -1, TRAV, SEMIS
1197 004030 000060' 177777 003674'
1198 004036 001446'
1199      ,
1200 004040      HEAD  203, PFA, 301, PFA, DOCOL      , ***** PFA

```

```

1195 004050 002520' 003674' 000060' . WORD ONE, TRAV, LIT, 5, PLUS, SEMIS
      004056 000005 001634' 001446'
1196
1197
1198
1199
1200 004064 HEAD 204, !CBP, 240, SCBP, DOCOL ; ***** !CBP
1201 004076 001366' 003236' 002242' . WORD SPAT, CBP, STORE, SEMIS
      004104 001446'
1202
1203 004106 HEAD 206, ?ERROR, 240, GERR, DOCOL ; ***** ?ERROR
1204 004122 002026' 000144' 000010 . WORD SWAP, ZBRAN, XXN2-, ERROR, BRAN, XXN3-.
      004130 006530' 000122' 000004
1205 004136 002004' XXN2: . WORD DROP
1206 004140 001446' XXN3: . WORD SEMIS
1207
1208 004142 HEAD 205, ?COMP, 320, GCOMP, DOCOL ; ***** ?COMP
1209 004154 003164' 002176' 001556' . WORD STATE, AT, ZEQV, LIT, 21, GERR, SEMIS
      004162 000060' 000021 004120'
      004170 001446'
1210
1211 004172 HEAD 205, ?EXEC, 303, GEXEC, DOCOL ; ***** ?EXEC
1212 004204 003164' 002176' 000060' . WORD STATE, AT, LIT, 22, GERR, SEMIS
      004212 000022 004120' 001446'
1213
1214 004220 HEAD 206, ?PAIRS, 240, GPAIR, DOCOL ; ***** ?PAIRS
1215 004234 003436' 000060' 000023 . WORD SUB, LIT, 23, GERR, SEMIS
      004242 004120' 001446'
1216
1217 004246 HEAD 204, ?CBP, 240, GCSP, DOCOL ; ***** ?CBP
1218 004260 001366' 003236' 002176' . WORD SPAT, CBP, AT, SUB, LIT, 24, GERR, SEMIS
      004266 003436' 000060' 000024
      004274 004120' 001446'
1219
1220 004300 HEAD 210, ?LOADING, 240, GLOAD, DOCOL ; ***** ?LOADING
1221 004316 003040' 002176' 001556' . WORD BLK, AT, ZEQV, LIT, 26, GERR, SEMIS
      004324 000060' 000026 004120'
      004332 001446'
1222
1223 004334 HEAD 207, COMPIL, 305, COMP, DOCOL ; ***** COMPIL
1224 COMPIL THE EXECUTION ADDRESS FOLLOWING.
1225 004350 004152' 001524' 002054' . WORD GCOMP, FROMR, DUP, TWOP, TOR, AT, COMMA, SEMIS
      004356 003340' 001506' 002176'
      004364 003416' 001446'
1226
1227 004370 HEAD 301, [, 333, LBRAC, DOCOL ; ***** [
1228 STOP COMPILATION, ENTER EXECUTION STATE.
1229 004376 002510' 003164' 002242' . WORD ZERO, STATE, STORE, SEMIS
      004404 001446'
1230
1231 004406 HEAD 201, ], 335, RBRAC, DOCOL ; ***** ]
1232 ENTER COMPILATION STATE.
1233 004414 000060' 000300 003164' . WORD LIT, 300, STATE, STORE, SEMIS
      004422 002242' 001446'
1234
1235 004426 HEAD 206, SMUDGE, 240, SMUDG, DOCOL ; ***** SMUDGE
1236 ALTER LATEST WORD NAME (SO THAT DICTIONARY SEARCH

```

```

1237      , WON'T FIND A PARTIALLY-COMPLETE ENTRY.
1238 004442 003742' 000060' 000040      . WORD  LATES, LIT, 40, TOQQL, SEMIS
      004450 002124' 001446'
1239      ,
1240 004454      HEAD  203, HEX, 330, HEX, DOCOL      ; ***** HEX
1241 004464 000060' 000020' 003200'      . WORD  LIT, 20, BASE, STORE, SEMIS
      004472 002242' 001446'
1242      ,
1243 004476      HEAD  207, DECIMAL, 314, DEC, DOCOL      ; ***** DECIMAL
1244 004512 000060' 000012' 003200'      . WORD  LIT, 12, BASE, STORE, SEMIS
      004520 002242' 001446'
1245      ,
1246 004524      HEAD  205, OCTAL, 314, OCTAL, DOCOL      ; ***** OCTAL
1247 004536 000060' 000010' 003200'      . WORD  LIT, 10, BASE, STORE, SEMIS
      004544 002242' 001446'
1248      ,
1249 004550      HEAD  207, <('CODE)>, 251, PSCOD, DOCOL      ; ***** ('CODE)
1250      USED ONLY BY COMPILER; COMPILED BY 'CODE'.
1251 004564 001524' 003742' 004046'      . WORD  FROMR, LATES, PFA, CFA, STORE, SEMIS
      004572 004002' 002242' 001446'
1252      ,
1253      ,
1254      , ***** THE DEFINITION OF 'CODE' WAS MOVED TO THE END OF
1255      , THE DICTIONARY, BECAUSE IT IS NOT PURE CODE (IT IS PATCHED
1256      , WHEN A FORTH ASSEMBLER IS LOADED).
1257      ,
1258      ,
1259 004600      HEAD  207, ^/<BUILDS/, 323, BUILD, DOCOL      ; ***** <BUILDS
1260      , CREATE NEW DATA TYPE WITH CODE ROUTINE IN HIGHER-LEVEL FORTH.
1261 004614 002510' 002406' 001446'      . WORD  ZERO, CON, SEMIS
1262      ,
1263 004622      HEAD  205, DOES>, 276, DOES, DOCOL      ; ***** DOES>
1264 004634 001524' 003742' 004046'      . WORD  FROMR, LATES, PFA, STORE, PSCOD
      004642 002242' 004562'
1265 004646 010446      DODOE:  MOV  IP, -(RP)
1266 004650 012204      MOV  (W)+, IP
1267 004652 010245      MOV  W, -(S)
1268 004654      NEXT
1269      ,
1270 004660      HEAD  205, COUNT, 324, COUNT, DOCOL      ; ***** COUNT
1271      , CONVERT STRING TO THE FORMAT USED BY 'TYPE'.
1272 004672 002054' 003322' 002026'      . WORD  DUP, ONEP, SWAP, CAT, SEMIS
      004700 002216' 001446'
1273      ,
1274 004704      HEAD  204, TYPE, 240, TYPE, DOCOL      ; ***** TYPE
1275 004716 003644' 000144' 000030      . WORD  DDUP, ZBRAN, XXL2-, , OVER, PLUS, SWAP, XDO
      004724 001762' 001634' 002026'
      004732 000334'
1276 004734 000360' 002216' 001020' XXL1: . WORD  I, CAT, EMIT, XLOOP, XXL1-, , BRAN, XXL3-,
      004742 000202' 177770 000122'
      004750 000004
1277 004752 002004'      XXL2: . WORD  DROP
1278 004754 001446'      XXL3: . WORD  SEMIS
1279      ,
1280 004756      HEAD  206, =CELLS, 240, ECELL, DOCOL      ; ***** =CELLS
1281      , NOTE - I NEED THIS, TO FORCE EVEN ADDRESS.
1282 004772 002054' 002520' 001302'      . WORD  DUP, ONE, AND, PLUS, SEMIS

```

```

005000 001634' 001446'
1283
1284 005004 HEAD 211,-TRAILING,307,DTRAI,DOCOL ; ***** -TRAILING
1285 005022 002054' 002510' 000334' . WORD DUP,ZERO,XDO
1286 005030 001762' 001762' 001634' XXW6: . WORD OVER,OVER,PLUS,ONE,SUB,CAT
005036 002520' 003436' 002216'
1287 005044 002552' 003436' 000144' . WORD BL,SUB,ZBRAN,XXW7-.,LEAVE,BRAN,XXWA-.
005052 000010 001466' 000122'
005060 000006
1288 005062 002520' 003436' XXW7: . WORD ONE,SUB
1289 005066 000202' 177740 001446' XXWA: . WORD XLOOP,XXW6-.,SEMIS
1290
1291 005074 HEAD 204,(.),240,PDOTQ,DOCOL ; ***** (.)
1292 USED ONLY BY COMPILER. COMPILED BY '. '
1293 005106 001540' 004670' 002054' . WORD R,COUNT,DUP,ONEP,ECELL
005114 003322' 004770'
1294 005120 001524' 001634' 001506' . WORD FROMR,PLUS,TOR,TYPE,SEMIS
005126 004714' 001446'
1295
1296 005132 HEAD 302,.,240,DOTQ,DOCOL ; ***** .
1297 TYPE ASCII MESSAGE.
1298 005142 000060' 000042 003164' . WORD LIT,34.,STATE,AT,ZBRAN,XXL6-.
005150 002176' 000144' 000026
1299 005156 004346' 005104' 005750' . WORD COMP,PDOTQ,WORD,HERE,CAT,ONEP,ECELL
005164 003362' 002216' 003322'
005172 004770'
1300 005174 003402' 000122' 000012 . WORD ALLOT,BRAN,XXL7-.
1301 005202 005750' 003362' 004670' XXL6: . WORD WORD,HERE,COUNT,TYPE
005210 004714'
1302 005212 001446' XXL7: . WORD SEMIS
1303
1304 005214 HEAD 206,?ALIGN,240,GALIO,DOCOL ; ***** ?ALIGN
1305 005230 003362' 002520' 001302' . WORD HERE,ONE,AND,ALLOT,SEMIS
005236 003402' 001446'
1306
1307 005242 HEAD 206,EXPECT,240,EXPEC,DOCOL ; ***** EXPECT
1308 READ N CHARACTERS TO MEMORY (AND TERMINATE WITH NULLS).
1309 ( ADDRESS N ==>).
1310 005256 001762' 001634' 001762' . WORD OVER,PLUS,OVER,XDO
005264 000334'
1311 005266 001030' 002054' 000060' XXK1: . WORD KEY,DUP,LIT,16,PORIO,AT,EQUAL,ZBRAN,XXK2-.
005274 000016 002632' 002176'
005302 003452' 000144' 000040
1312 005310 002004' 000060' 000010 . WORD DROP,LIT,10,OVER,I,EQUAL,DUP,FROMR
005316 001762' 000360' 003452'
005324 002054' 001524'
1313 005330 002530' 003436' 001634' . WORD TWO,SUB,PLUS,TOR,SUB,BRAN,XXK3-.
005336 001506' 003436' 000122'
005344 000060
1314 005346 002054' 000060' 000015 XXK2: . WORD DUP,LIT,15,EQUAL,ZBRAN,XXK4-.
005354 003452' 000144' 000016
1315 005362 001466' 002004' 002552' . WORD LEAVE,DROP,BL,ZERO,BRAN,XXK5-.
005370 002510' 000122' 000004
1316 005376 002054' XXK4: . WORD DUP
1317 005400 000360' 002270' 002510' XXK5: . WORD I,CSTOR,ZERO,I,ONEP,CSTOR,ZERO,I,TWOP,CSTOR
005406 000360' 003322' 002270'
005414 002510' 000360' 003340'

```

```

005422 002270'
1318                                     ; NOTE DIFFERENCE FOR STAND-ALONE, BELOW
1319                                     . IFDF ALONE
1320 005424 001020' 000202' 177636 XXK3: . WORD EMIT, XLOOP, XXK1-, , DROP, SEMIS
005432 002004' 001446'
1321                                     . IFF
1322 XXK3: . WORD DROP, XLOOP, XXK1-, , DROP, SEMIS
1323                                     . ENDC
1324
1325 005436 HEAD 205, QUERY, 331, QUERY, DOCOL ; ***** QUERY
1326 005450 002676' 002176' 000060' . WORD TIB, AT, LIT, 120, EXPEC, ZERO, IN, STORE, SEMIS
005456 000120' 005254' 002510'
005464 003052' 002242' 001446'
1327
1328 005472 HEAD 301, X, 200, NULL, DOCOL ; ***** THE NULL
1329 ; THE NULL OPERATION (ASCII 0) STOPS INTERPRETATION/COMPILATION
1330 ; AT END OF A TERMINAL INPUT LINE, OR A DISK SCREEN. ALL DISK
1331 ; BUFFERS MUST TERMINATE WITH NULLS, AND 'EXPECT' PLACES NULLS
1332 ; AFTER EACH TERMINAL INPUT LINE.
1333 ; NOTE THAT THE 'X' IN THE HEADER ABOVE WILL BE CHANGED TO A NULL.
1334 005500 003040' 002176' . WORD BLK, AT
1335 005504 000144' 000046' 002520' . WORD ZBRAN, XXJ2-, , ONE, BLK, PSTOR, ZERO, IN, STORE
005512 003040' 002072' 002510'
005520 003052' 002242'
1336 005524 003040' 002176' 002614' . WORD BLK, AT, BSCR, MOD, ZEGU, ZBRAN, XXJ1-, , GEXEC, FROMR, DROP
005532 010534' 001556' 000144'
005540 000010' 004202' 001524'
005546 002004'
1337 005550 000122' 000006 XXJ1: . WORD BRAN, XXJ4-.
1338 005554 001524' 002004' XXJ2: . WORD FROMR, DROP
1339 005560 001446' XXJ4: . WORD SEMIS
1340
1341 005562 HEAD 204, FILL, 240, FILL, DOCOL ; ***** FILL
1342 005574 002026' 001506' 001762' . WORD SWAP, TOR, OVER, CSTOR, DUP, ONEP, FROMR
005602 002270' 002054' 003322'
005610 001524'
1343 005612 002520' 003436' 001070' . WORD ONE, SUB, CMOVE, SEMIS
005620 001446'
1344
1345 005622 HEAD 205, ERASE, 305, ERASE, DOCOL ; ***** ERASE
1346 005634 002510' 005572' 001446' . WORD ZERO, FILL, SEMIS
1347
1348 005642 HEAD 206, BLANKS, 240, BLANK, DOCOL ; ***** BLANKS
1349 005656 002552' 005572' 001446' . WORD BL, FILL, SEMIS
1350
1351 005664 HEAD 204, HOLD, 240, HOLD, DOCOL ; ***** HOLD
1352 005676 000060' 177777 003262' . WORD LIT, -1, HLD, PSTOR, HLD, AT, CSTOR, SEMIS
005704 002072' 003262' 002176'
005712 002270' 001446'
1353
1354 005716 HEAD 203, PAD, 304, PAD, DOCOL ; ***** PAD
1355 005726 003362' 000060' 000104 . WORD HERE, LIT, 104, PLUS, SEMIS
005734 001634' 001446'
1356
1357 005740 HEAD 204, WORD, 240, WORD, DOCOL ; ***** WORD
1358 005752 003040' 002176' 000144' . WORD BLK, AT, ZBRAN, XXJ1-, , BLK, AT, BLOCK, BRAN, XXJ2-,
005760 000014 003040' 002176'

```

005766	011314'	000122'	000006			
1359 005774	002676'	002176'		XXI1:	. WORD	TIB, AT
1360 006000	003052'	002176'	001634'	XXI2:	. WORD	IN, AT, PLUS, SWAP, ENCL, HERE, LIT, 42, BLANK, IN
006006	002026'	000676'	003362'			
006014	000060'	000042'	005654'			
006022	003052'					
1361 006024	002072'	001762'	003436'		. WORD	PSTOR, OVER, SUB, TOR, R, HERE, CSTOR, PLUS
006032	001506'	001540'	003362'			
006040	002270'	001634'				
1362 006044	003362'	003322'	001524'		. WORD	HERE, ONEP, FROMR, CMOVE, SEMIS
006052	001070'	001446'				
1363						
1364						
1365 006056						
1366 006074	003322'	002054'	001506'	XXF3:	HEAD	210, (NUMBER), 240, PNUMB, DOCOL ; ***** (NUMBER)
006102	002216'	003200'	002176'		. WORD	ONEP, DUP, TOR, CAT, BASE, AT, DIQIT
006110	000400'					
1367 006112	000144'	000054'	002026'		. WORD	ZBRAN, XXQ4-, SWAP, BASE, AT, USTAR, DROP
006120	003200'	002176'	001132'			
006126	002004'					
1368 006130	003564'	003200'	002176'		. WORD	ROT, BASE, AT, USTAR, DPLUS
006136	001132'	001652'				
1369 006142	003212'	002176'	003322'		. WORD	DPL, AT, ONEP, ZBRAN, XXQ5-, ONE, DPL, PSTOR
006150	000144'	000010'	002520'			
006156	003212'	002072'				
1370 006162	001524'	000122'	177706	XXQ5:	. WORD	FROMR, BRAN, XXF3-
1371 006170	001524'	001446'		XXQ4:	. WORD	FROMR, SEMIS
1372						
1373 006174					HEAD	206, NUMBER, 240, NUMB, DOCOL ; ***** NUMBER
1374 006210	002510'	002510'	003564'		. WORD	ZERO, ZERO, ROT, DUP, ONEP, CAT, LIT, 55, EQUAL
006216	002054'	003322'	002216'			
006224	000060'	000055'	003452'			
1375 006232	002054'	001506'	001634'		. WORD	DUP, TOR, PLUS, LIT, -1
006240	000060'	177777				
1376 006244	003212'	002242'	006072'	XXF6:	. WORD	DPL, STORE, PNUMB, DUP, CAT, BL, SUB
006252	002054'	002216'	002552'			
006260	003436'					
1377 006262	000144'	000026'	002054'		. WORD	ZBRAN, XXF7-, DUP, CAT, LIT, 56, SUB
006270	002216'	000060'	000056'			
006276	003436'					
1378 006300	002510'	004120'	002510'		. WORD	ZERO, GERR, ZERO, BRAN, XXF6-
006306	000122'	177734				
1379 006312	002004'	001524'	000144'	XXF7:	. WORD	DROP, FROMR, ZBRAN, XXFA-, DMINU
006320	000004'	001734'				
1380 006324	001446'			XXFA:	. WORD	SEMIS
1381						
1382 006326					HEAD	205, -FIND, 304, DFIND, DOCOL ; ***** -FIND
1383 006340	002552'	005750'	003362'		. WORD	BL, WORD, HERE, COUNT, UPPER, HERE, CONT, AT, AT, PFIND
006346	004670'	006416'	003362'			
006354	003132'	002176'	002176'			
006362	000506'					
1384 006364	002054'	001556'	000144'		. WORD	DUP, ZEQU, ZBRAN, XXE3-, DROP, HERE, LATES, PFIND
006372	000012'	002004'	003362'			
006400	003742'	000506'				
1385 006404	001446'			XXE3:	. WORD	SEMIS
1386						
1387 006406					HEAD	205, UPPER, 322, UPPER, DOCOL ; ***** UPPER

```

1388      ; SETS STRINGS TO UPPER CASE - TO ALLOW
1389      ; LOWER AS WELL AS UPPER CASE FROM TERMINAL.
1390      .WORD      001762' 001634' 002026'
1391      .WORD      000334' 002216' 000060' XXE2:
1392      .WORD      000360' 000137' 000332' 000360'
1393      .WORD      002216' 000060' 000173
1394      .WORD      000452' 000350' 000144' 000012
1395      .WORD      000434' 001302' 000360' 000040
1396      .WORD      000462' 000360' 000060' 000040
1397      .WORD      000470' 002124' 177734 001446' XXE1:
1398      .WORD      000472' 000202' 000762' 001446'
1399      .WORD      000500' 000514 000762' 001446'
1400      .WORD      000520' 000532' 000540' 000557
1401      .WORD      000556' 000557 000556' 000557
1402      .WORD      000556' 000557 000556' 000557
1403      .WORD      000556' 000557 000556' 000557
1404      .WORD      000556' 000557 000556' 000557
1405      .WORD      000556' 000557 000556' 000557
1406      .WORD      000556' 000557 000556' 000557
1407      .WORD      000556' 000557 000556' 000557
1408      .WORD      000556' 000557 000556' 000557
1409      .WORD      000556' 000557 000556' 000557
1410      .WORD      000556' 000557 000556' 000557
1411      .WORD      000556' 000557 000556' 000557
1412      .WORD      000556' 000557 000556' 000557
1413      .WORD      000556' 000557 000556' 000557
1414      .WORD      000556' 000557 000556' 000557
1415      .WORD      000556' 000557 000556' 000557
1416      .WORD      000556' 000557 000556' 000557
1417      .WORD      000556' 000557 000556' 000557
1418      .WORD      000556' 000557 000556' 000557
1419      .WORD      000556' 000557 000556' 000557

```

; SETS STRINGS TO UPPER CASE - TO ALLOW
 ; LOWER AS WELL AS UPPER CASE FROM TERMINAL.
 .WORD OVER, PLUS, SWAP, XDO
 .WORD I, CAT, LIT, ¹⁴⁰137, GREAT, I, CAT, LIT, 173, LESS
 AND, ZBRAN, XXE1-, I, LIT, 40, TOOOL
 XLOOP, XXE2-, SEMIS
 207, (ABORT), 251, PABOR, DOCOL
 ABORT, SEMIS
 205, ERROR, 322, ERROR, DOCOL
 WARN, AT, ZLESS, ZBRAN, XXN4-, PABOR
 HERE, COUNT, TYPE, PDOTO
 .BYTE 3
 .ASCII / ? /
 .EVEN
 .WORD MESS, SPSTO, IN, AT, BLK, AT, QUIT, SEMIS
 203, ID., 256, IDDOT, DOCOL
 PAD, LIT, 40, LIT, 137, FILL, DUP
 PFA, LFA, OVER, SUB, PAD, SWAP, CHOVE
 PAD, COUNT, LIT, 37, AND, TYPE, SPACE, SEMIS
 206, CREATE, 240, CREAT, DOCOL
 DFIND, ZBRAN, XXD2-, DROP, NFA, IDDOT
 LIT, 4, MESS, SPACE
 HERE, DUP, CAT, WIDTH, AT, MIN, ONEP, ALLOT
 GALIQ, DUP, LIT, 240, TOOOL, HERE, ONE, SUB
 LIT, 200, TOOOL, LATES, COMMA, CURR, AT, STORE
 HERE, TWOP, COMMA, SEMIS
 311, [COMPILE], 335, BCOMP, DOCOL

; ***** (ABORT)
 ; ***** ERROR
 ; ***** ID.
 ; ***** CREATE
 ; ***** [COMPILE]

```

1420 007034 006336' 001556' 002510' . WORD DFIND, ZEGU, ZERO, GERR, DROP, CFA, COMMA, SEMIS
      007042 004120' 002004' 004002'
      007050 003416' 001446'
1421                                     ,
1422 007054                                     HEAD 307, LITERAL, 314, LITER, DOCOL ; ***** LITERAL
1423 007070 003164' 002176' 000144' . WORD STATE, AT, ZBRAN, XXD6-, , COMP, LIT, COMMA
      007076 000010' 004346' 000060'
      007104 003416'
1424 007106 001446' XXD6: . WORD SEMIS
1425                                     ,
1426 007110                                     HEAD 310, DLITERAL, 240, DLITE, DOCOL ; ***** DLITERAL
1427 007126 003164' 002176' 000144' . WORD STATE, AT, ZBRAN, XXN5-, , SWAP, LITER, LITER
      007134 000010' 002026' 007066'
      007142 007066'
1428 007144 001446' XXN5: . WORD SEMIS
1429                                     ,
1430 007146                                     HEAD 202, UC, 240, ULESS, DOCOL ; ***** UC
1431                                     , UNSIGNED LESS-THAN, NEEDED FOR '7STACK'
1432                                     , : UC >R 0 R> 0 DMINUS D+ SWAP DROP 0< ;
1433 007156 001506' 002510' 001524' . WORD TOR, ZERO, FROMR, ZERO, DMINU, DPLUS
      007164 002510' 001734' 001652'
1434 007172 002026' 002004' 001606' . WORD SWAP, DROP, ZLESS, SEMIS
      007200 001446'
1435                                     ,
1436 007202                                     HEAD 206, 7STACK, 240, GSTAC, DOCOL ; ***** 7STACK
1437                                     , ERROR CHECK.
1438 007216 002652' 002176' 002530' . WORD SZERO, AT, TWO, SUB, SPAT, ULESS, ONE, GERR
      007224 003436' 001366' 007154'
      007232 002520' 004120'
1439 007236 001366' 003362' 000060' . WORD SPAT, HERE, LIT, 200, PLUS, ULESS, TWO, GERR
      007244 000200' 001634' 007154'
      007252 002530' 004120'
1440 007256 001446' . WORD SEMIS
1441                                     ,
1442 007260                                     HEAD 211, INTERPRET, 324, INTER, DOCOL ; ***** INTERPRET
1443 007276 006336' XXE4: . WORD DFIND
1444 007300 000144' 000036' 003164' . WORD ZBRAN, XXEA-, , STATE, AT, LESS
      007306 002176' 003502'
1445 007312 000144' 000012' 004002' . WORD ZBRAN, XXE5-, , CFA, COMMA, BRAN, XXE6-,
      007320 003416' 000122' 000006
1446 007326 004002' 000102' XXE5: . WORD CFA, EXEC
1447 007332 007214' 000122' 000034 XXE6: . WORD GSTAC, BRAN, XXE7-,
1448 007340 003362' 006206' 003212' XXEA: . WORD HERE, NUMB, DPL, AT, ONEP, ZBRAN, XXF4-, , DLITE, BRAN, XXF5-,
      007346 002176' 003322' 000144'
      007354 000010' 007124' 000122'
      007362 000006
1449 007364 002004' 007066' XXF4: . WORD DROP, LITER
1450 007370 007214' XXF5: . WORD GSTAC
1451 007372 000122' 177702 XXE7: . WORD BRAN, XXE4-,
1452                                     ,
1453 007376                                     HEAD 211, IMMEDIATE, 305, IMMED, DOCOL ; ***** IMMEDIATE
1454 007414 003742' 000060' 000100 . WORD LATES, LIT, 100, TO00L, SEMIS
      007422 002124' 001446'
1455                                     ,
1456 007426                                     HEAD 212, VOCABULARY, 240, VOCAB, DOCOL ; ***** VOCABULARY
1457 007446 004612' 000060' 120201 . WORD BUILD, LIT, 120201, COMMA, CURR, AT, CFA, COMMA
      007454 003416' 003150' 002176'

```

```

007462 004002' 003416'
1458 007466 003362' 002776' 002176' . WORD HERE, VOCL, AT, COMMA, VOCL, STORE, DOES
007474 003416' 002776' 002242'
007502 004632'
1459 007504 003340' 003132' 002242' DOVOC: . WORD TWOP, CONT, STORE, SEMIS
007512 001446'

1460
1461
1462
1463
1464
1465
1466 007514
1467 007534 003132' 002176' 003150' . HEAD 213, DEFINITIONS, 323, DEFIN, DOCOL ; ***** DEFINITIONS
007542 002242' 001446' . WORD CONT, AT, CURR, STORE, SEMIS

1468
1469 007546
1470 007554 000060' 000051 005750' . HEAD 301, (, 250, PAREN, DOCOL ; ***** (
007562 001446' . WORD LIT, 51, WORD, SEMIS

1471
1472 007564
1473 007576 002510' 003040' 002242' . HEAD 204, QUIT, 240, QUIT, DOCOL ; ***** QUIT
007604 004374' . WORD ZERO, BLK, STORE, LBRAC

1474 007606 001426' 001056' 005446' XXB1: . WORD RPSTD, CR, QUERY, INTER, STATE, AT
007614 007274' 003164' 002176'
1475 007622 001556' 000144' 000010 . WORD ZEGU, ZBRAN, XXB2-, , PDOTQ
007630 005104'

1476 007632 003 . BYTE 3
1477 007633 040 117 113 . ASCII / OK/
1478 . EVEN
1479 007636 000122' 177746 XXB2: . WORD BRAN, XXB1-.
1480
1481 007642
1482 007654 001406' 004510' 003622' . HEAD 205, ABORT, 324, ABORT, DOCOL ; ***** ABORT
1483 007662 001056' 005104' . WORD SPSTD, DEC, SPACE
1484 007666 021 . WORD CR, PDOTQ
1485 007667 106 111 107 . BYTE 21
007672 055 106 117 . ASCII /FIQ-FORTH V 1.3 /
007675 122 124 110
007700 040 040 126
007703 040 061 056
007706 063 040

1486 . EVEN
1487 007710 015400' 007532' 007574' . WORD FORTH, DEFIN, QUIT
1488
1489
1490
1491 007716
1492 007730
1493 007730 016767 170060 005450 . HEAD 204, COLD, 240, COLD ; ***** COLD
1494 007736 016703 170056 CENT: . WORD COLD START ENTRY POINT
1495 . MOV ORIGIN+14, FORTH+6 ; SET 'FORTH' VOCABULARY FROM STARTUP TABLE
1496 . MOV ORIGIN+20, U ; INITIALIZE USER POINTER
1497 ; NOTE - FOR SMALLER STAND-ALONE BOOT, INITIALIZE AREAS IN
; HIGH MEMORY WHICH MUST BE INITIALIZED.
; CLEAR DISK BUFFERS ON FIRST TIME THROUGH
1498 007742 016700 170074 . MOV ORIGIN+42, R0 ; 'FIRST' - BEGINNING OF DISK BUFFERS
1499 007746 016701 170072 . MOV ORIGIN+44, R1 ; 'LIMIT' - JUST BEYOND DISK BUFFERS
1500 007752 005020 1#: . CLR (R0)+

```

```

1501 007754 020001          CMP      R0,R1
1502 007756 002775          BLT      1$
1503                          ; NOW INITIALIZE 'OUT', 'OFFSET', 'USE' AND 'PREV'
1504 007760 005063 000042      CLR      42(U)          ; CLEAR 'OUT'
1505 007764 005063 000046      CLR      46(U)          ; CLEAR 'OFFSET'
1506 007770 016763 170046 000072      MOV      ORIGIN+42,72(U) ; TO 'USE'
1507 007776 016763 170040 000074      MOV      ORIGIN+42,74(U) ; TO 'PREV'
1508                          ; END OF SPECIAL HIGH-MEMORY INITIALIZE
1509 010004 012701 000030      MOV      #30,R1          ; ON COLD START, MOVE 24. BYTES
1510 010010 000402          BR      W2
1511 010012          WENT:      ; WARM START ENTRY POINT
1512 010012 012701 000012      MOV      #12,R1          ; ON WARM START, MOVE TEN BYTES
1513 010016 012705 000022'      W2:      MOV      #ORIGIN+22,R5 ; START MOVING FROM HERE
1514 010022 016700 167772      MOV      ORIGIN+20,R0      ; MOVE TO THE USER AREA
1515 010026 062700 000006      ADD      #6,R0           ; PLUS 6
1516 010032 060501          ADD      R5,R1              ; COMPUTE LOOP STOP ADDRESS
1517 010034 012520      1$:      MOV      (R5)+,(R0)+
1518 010036 020501          CMP      R5,R1
1519 010040 002775          BLT      1$
1520 010042 016706 167756      MOV      ORIGIN+24,RP      ; INITIALIZE THE RETURN-STACK POINTER
1521                          ; NOW SET FORTH'S INSTRUCTION COUNTER, AND GO
1522 010046 012704 010056'      MOV      #00,IP          ; START EXECUTION WITH 'ABORT'
1523 010052          NEXT
1524                          ; NOTE - NORMALLY THE ABOVE INSTRUCTION WOULD BE 'MOV #ABORT+2,IP'.
1525                          ; IT HAS BEEN CHANGED HERE TO ALLOW USER TO PATCH A DIFFERENT
1526                          ; START-UP. BUT THE SYSTEM WON'T WORK UNTIL SOME OF THE WORK
1527                          ; OF 'ABORT' HAS BEEN DONE, SO THAT WORK IS REPEATED. THE USER
1528                          ; CAN PATCH OVER THE 'ABORT' AND THE ZEROS.
1529                          ;
1530 010056 001406' 004510' 015400' 00:      .WORD      SPSTD, DEC, FORTH, DEFIN, ABORT, 0, 0, 0
1531 010064 007532' 007652' 000000
1532 010072 000000 000000
1533
1534
1535
1536 010076          HEAD      204, S->D, 240, STOD          ; ***** S->D
1537 010110 005045          CLR      -(8)                  ; SIGN EXTEND WITH ZEROS
1538 010112 005765 000002      TST      2(8)                ; BUT IF NEGATIVE,
1539 010116 100001          BPL      1$
1540 010120 005315          DEC      (8)                    ; CHANGE THE ZEROS TO ONES
1541 010122      1$:      NEXT
1542
1543                          ; NOTE - THIS SYSTEM DOESN'T NEED THE OPERATIONS '+-' AND 'D+--',
1544                          ; BECAUSE 'M*' AND 'M/' ARE DEFINED IN CODE.
1545
1546 010126          HEAD      203, ABS, 323, ABS, DOCOL          ; ***** ABS
1547 010136 002054' 001606' 000144'      .WORD      DUP, ZLESS, ZBRAN, XXRS-, , MINUS
1548 010144 000004 001712'
1549 010150 001446'      XXRS:      .WORD      SEMIS
1550
1551 010152          HEAD      204, DABS, 240, DABS, DOCOL          ; ***** DABS
1552 010164 002054' 001606' 000144'      .WORD      DUP, ZLESS, ZBRAN, XXRB-, , DMINU
1553 010172 000004 001734'
1554 010176 001446'      XXRB:      .WORD      SEMIS
1555

```

```

1554 010200          HEAD      203, MIN, 316, MIN, DOCOL          ; ***** MIN
1555 010210 001762' 001762' 003532' . WORD      OVER, OVER, GREAT, ZBRAN, XXR7-. , SWAP
      010216 000144' 000004 002026'
1556 010224 002004' 001446' XXR7: . WORD      DROP, SEMIS
1557
1558 010230          HEAD      203, MAX, 330, MAX, DOCOL          ; ***** MAX
1559 010240 001762' 001762' 003502' . WORD      OVER, OVER, LESS, ZBRAN, XXR6-. , SWAP
      010246 000144' 000004 002026'
1560 010254 002004' 001446' XXR6: . WORD      DROP, SEMIS
1561
1562 010260          HEAD      202, M*, 240, MSTAR          ; ***** M*
1563          . IFDF      EIS          ; HARDWARE MULTIPLY/DIVIDE?
1564          MOV          (S)+, R0
1565          MUL          (S), R0
1566          MOV          R1, (S)
1567          MOV          R0, -(S)
1568          NEXT
1569          . IFF
1570 010270 016546 000002          MOV          2(S), -(RP)          ; USE RETURN STACK FOR SAVING SIGN
1571 010274 100002          BPL          1$
1572 010276 005465 000002          NEG          2(S)          ; GET ABSOLUTE VALUE
1573 010302 005715          1$: TST          (S)
1574 010304 100002          BPL          2$
1575 010306 005416          NEG          (RP)          ; ADJUST SIGN WHICH WAS SAVED
1576 010310 005415          NEG          (S)          ; GET ABSOLUTE VALUE
1577 010312 004767 170626          2$: JSR          PC, UMULT
1578 010316 005726          TST          (RP)+          ; NEGATIVE RESULT?
1579 010320 100007          BPL          3$          ; NO
1580          ; IF GET HERE, NEGATE THE DOUBLE-INTEGGER NUMBER ON THE STACK
1581 010322 005115          COM          (S)
1582 010324 005165 000002          COM          2(S)
1583 010330 062765 000001 000002          ADD          #1, 2(S)
1584 010336 005515          ADC          (S)
1585 010340          3$: NEXT
1586          . ENDC
1587
1588 010344          HEAD      202, M/, 240, MSLAS          ; ***** M/
1589          . IFDF      EIS          ; HARDWARE MULTIPLY/DIVIDE?
1590          MOV          2(S), R0
1591          MOV          4(S), R1
1592          DIV          (S)+, R0
1593          MOV          R1, 2(S)
1594          MOV          R0, (S)
1595          NEXT
1596          . IFF
1597 010354 016546 000002          MOV          2(S), -(RP)          ; SAVE DIVIDEND SIGN
1598 010360 001001          BNE          5$          ; ZERO WOULDN'T INDICATE
1599 010362 005216          INC          (RP)          ; A SIGN CHANGE.
1600 010364 011646          5$: MOV          (RP), -(RP)          ; DUPLICATE IT
1601 010366 100011          BPL          1$
1602          ; IF GET HERE, TAKE ABSOLUTE VALUE OF DOUBLE-INTEGGER DIVIDEND.
1603 010370 005165 000002          COM          2(S)
1604 010374 005165 000004          COM          4(S)
1605 010400 062765 000001 000004          ADD          #1, 4(S)
1606 010406 005565 000002          ADC          2(S)
1607 010412 005715          1$: TST          (S)          ; IS DIVISOR NEGATIVE?
1608 010414 100002          BPL          2$

```

NEQ	(RP)	IF YES, NEGATE QUOTIENT SIGN
NEQ	(S)	AND TAKE ABS. VALUE OF DIVISOR
JSR	PC,UDIV	
TST	(RP)+	NEGATIVE QUOTIENT?
BPL	3*	NO
NEQ	(S)	NEGATE THE QUOTIENT
TST	(RP)+	NEGATIVE DIVIDEND?
BPL	4*	NEGATE THE REMAINDER
NEQ	2(S)	
NEXT		
.ENDC		
1609 010416 005416	201.*	252, STAR, DOCOL
1610 010420 005415	MSTAR, DROP, SEMIS	
1611 010422 004767	204, /MOD, 240, SLMOD, DOCOL	
1612 010426 005726	TOR, STOD, FROMR, MSLAS, SEMIS	
1613 010430 100001		
1614 010432 005415	201, /, 257, SLASH, DOCOL	
1615 010434 005726	SLMOD, SWAP, DROP, SEMIS	
1616 010436 100002	203, MOD, 304, MOD, DOCOL	
1617 010440 005465	SLMOD, DROP, SEMIS	
1618 010444	205, */MOD, 304, SSMOD, DOCOL	
1619	TOR, MSTAR, FROMR, MSLAS, SEMIS	
1620	202, */, 240, SSLA, DOCOL	
1621 010450	SSMOD, SWAP, DROP, SEMIS	
1622 010456	205, M/MOD, 304, MSMOD, DOCOL	
1623	TOR, ZERO, R, USLAS, FROMR	
1624 010464	SWAP, TOR, USLAS, FROMR, SEMIS	
1625 010476		
1626 010504		
1627 010510		
1628 010516		
1629		
1630 010526		
1631 010536		
1632		
1633 010544		
1634 010556		
1635 010564		
1636 010570		
1637 010600		
1638 010606		
1639 010610		
1640 010622		
1641 010630		
1642 010634		
1643		
1644		
1645		
1646		

```

1648                                     ; *****
1649                                     ;
1650                                     ;     DISK I/O (SECTION COMMON TO ALL OPERATING SYSTEMS)
1651                                     ;     NOTE THAT EACH OPERATING SYSTEM DEFINED 'R/W' - READ
1652                                     ;     OR WRITE A 1024-BYTE RANDOM-ACCESS BLOCK.
1653                                     ;
1654                                     ; *****
1655                                     ;
1656                                     ;
1657                                     ; 'USE' AND 'PREV' MOVED TO USER AREA
1658                                     ;
1659 010646                                HEAD    204, +BUF, 240, PBUF, DOCOL          ; ***** +BUF
1660 010660 002600' 000060' 000004      . WORD    BBUF, LIT, 4, PLUS, PLUS, DUP, LIMIT, AT, EQUAL
1661 010666 001634' 001634' 002054'
1662 010674 003026' 002176' 003452'
1663 010702 000144' 000010' 002004'      . WORD    ZBRAN, XXT1-, , DROP, FIRST, AT
1664 010710 003012' 002176'
1665 010714 002054' 003310' 002176' XXT1: . WORD    DUP, PREV, AT, SUB, SEMIS
1666 010722 003436' 001446'
1667
1668 010726                                HEAD    206, UPDATE, 240, UPDAT, DOCOL          ; ***** UPDATE
1669 010742 003310' 002176' 002176'      . WORD    PREV, AT, AT, LIT, 100000, OR, PREV
1670 010750 000060' 100000' 001322'
1671 010756 003310'
1672 010760 002176' 002242' 001446'      . WORD    AT, STORE, SEMIS
1673
1674 010766                                HEAD    215, EMPTY-BUFFERS, 323, MTBUF, DOCOL          ; ***** EMPTY-BUFFERS
1675 011010 003012' 002176' 003026'      . WORD    FIRST, AT, LIMIT, AT, OVER, SUB, ERASE, SEMIS
1676 011016 002176' 001762' 003436'
1677 011024 005632' 001446'
1678
1679 011030                                HEAD    205, FLUSH, 310, FLUSH, DOCOL          ; ***** FLUSH
1680
1681 011042 003026' 002176' 003012'      . WORD    SOME SYSTEMS DEFINE THIS IN THE EDITOR, NOT HERE.
1682 011050 002176' 000334'
1683 011054 000360' 002176' 001606' XXTA: . WORD    LIMIT, AT, FIRST, AT, XDO
1684 011062 000144' 000024' 000360'
1685 011070 003340' 000360' 002176'
1686 011076 000060' 077777' 001302'      . WORD    I, AT, ZLESS, ZBRAN, XXT7-, , I, TWOP, I, AT
1687 011104 002510' 015174'
1688 011110 002600' 000060' 000004 XXT7: . WORD    LIT, 77777, AND, ZERO, RW
1689 011116 001634' 000250' 177732
1690 011124 011006' 001446'
1691
1692 011130                                HEAD    203, DR0, 260, DRZER, DOCOL          ; ***** DR0
1693
1694 011140 002510' 003114' 002242'      . WORD    SELECT DRIVE #0 - NOT USED WITH RT11 OR RSX11
1695 011146 001446'
1696
1697 011150                                HEAD    203, DR1, 261, DRONE, DOCOL          ; ***** DR1
1698
1699 011160 000060' 000360' 003114'      . WORD    SELECT DRIVE #1 - NOT USED IN RSX11 OR RT11
1700 011166 002242' 001446'
1701
1702 011172                                HEAD    206, BUFFER, 240, BUFBE, DOCOL          ; ***** BUFFER
1703 011206 003274' 002176' 002054'      . WORD    LIT, 240, , OFFSET, STORE, SEMIS
1704 011214 001506'

```

```

1688 011216 010656' 000144' 177774 XXT2: . WORD PBUF, ZBRAN, XXT2-. , USE, STORE
      011224 003274' 002242'
1689 011230 001540' 002176' 001606' . WORD R, AT, ZLESS, ZBRAN, XXT3-.
      011236 000144' 000024
1690 011242 001540' 003340' 001540' . WORD R, TWOP, R, AT, LIT, 77777, AND
      011250 002176' 000060' 077777
      011256 001302'
1691 011260 002510' 015174' . WORD ZERO, RW
1692 011264 001540' 002242' 001540' XXT3: . WORD R, STORE, R, PREV, STORE, FROMR, TWOP, SEMIS
      011272 003310' 002242' 001524'
      011300 003340' 001446'
1693
1694 011304 HEAD 205, BLOCK, 313, BLOCK, DOCOL ; ***** BLOCK
1695 ; CHANGED TO MASK OFF THE UPDATE BIT WHEN COMPARING
1696 011316 003114' 002176' 001634' . WORD OFFSET, AT, PLUS, TOR
      011324 001506'
1697 011326 003310' 002176' 002054' . WORD PREV, AT, DUP, AT, LIT, 077777, AND, R, SUB, ZBRAN, XXT4-.
      011334 002176' 000060' 077777
      011342 001302' 001540' 003436'
      011350 000144' 000066
1698 011354 010656' 001556' 000144' XXT5: . WORD PBUF, ZEGU, ZBRAN, XXT6-.
      011362 000024
1699 011364 002004' 001540' 011204' . WORD DROP, R, BUFFE
1700 011372 002054' 001540' 002520' . WORD DUP, R, ONE, RW, TWO, SUB
      011400 015174' 002530' 003436'
1701 011406 002054' 002176' 000060' XXT6: . WORD DUP, AT, LIT, 077777, AND, R, SUB, ZEGU
      011414 077777 001302' 001540'
      011422 003436' 001556'
1702 011426 000144' 177724 . WORD ZBRAN, XXT5-.
1703 011432 002054' 003310' 002242' . WORD DUP, PREV, STORE
1704 011440 001524' 002004' 003340' XXT4: . WORD FROMR, DROP, TWOP, SEMIS
      011446 001446'
1705
1706
1707
1708
1709
1710
1711 011450 HEAD 206, (LINE), 240, PLINE, DOCOL ; ***** (LINE)
1712 011464 001506' 002564' 002600' . WORD TOR, CL, BEUF, BSMOD, FROMR, BSCR
      011472 010554' 001524' 002614'
1713 011500 010454' 001634' 011314' . WORD STAR, PLUS, BLOCK, PLUS, CL, SEMIS
      011506 001634' 002564' 001446'
1714
1715 011514 HEAD 205, . LINE, 305, DLINE, DOCOL ; ***** . LINE
1716 011526 011462' 005020' 004714' . WORD PLINE, DTRAI, TYPE, SEMIS
      011534 001446'
1717
1718 011536 HEAD 207, MESSAGE, 305, MESS, DOCOL ; ***** MESSAGE
1719 011552 002730' 002176' 000144' . WORD WARN, AT, ZBRAN, XXW5-. , DDUP, ZBRAN, XXW3-. , LIT, 4
      011560 000034' 003644' 000144'
      011566 000022' 000060' 000004
1720 011574 003114' 002176' 002614' . WORD OFFSET, AT, BSCR, SLASH, SUB, DLINE
      011602 010514' 003436' 011524'
1721 011610 000122' 000016 XXW3: . WORD BRAN, XXW4-.
1722 011614 005104' XXW5: . WORD PDDTG
1723 011616 006 . BYTE 6

```

```

1724 011617      115      123      107      . ASCII /MSQ # /
      011622      040      043      040
1725                                     . EVEN
1726 011626      013144'      . WORD      DOT
1727 011630      001446'      XXW4: . WORD      SEMIS
1728                                     ,
1729 011632                                     HEAD      204, LOAD, 240, LOAD, DOCOL      ; ***** LOAD
1730 011644      003040' 002176' 001506'      . WORD      BLK, AT, TOR, IN, AT, TOR, ZERO, IN, STORE
      011652      003052' 002176' 001506'
      011660      002510' 003052' 002242'
1731 011666      002614' 010454' 003040'      . WORD      BSCR, STAR, BLK, STORE, INTER, FROMR, IN, STORE
      011674      002242' 007274' 001524'
      011702      003052' 002242'
1732 011706      001524' 003040' 002242'      . WORD      FROMR, BLK, STORE, SEMIS
      011714      001446'
1733                                     ,
1734 011716                                     HEAD      303, -->, 276, ARROW, DOCOL      ; ***** -->
1735 011726      004314' 002510' 003052'      . WORD      QLOAD, ZERO, IN, STORE, BSCR, BLK, AT, OVER
      011734      002242' 002614' 003040'
      011742      002176' 001762'
1736 011746      010534' 003436' 003040'      . WORD      MOD, SUB, BLK, PSTOR, SEMIS
      011754      002072' 001446'
1737                                     ,
1738                                     ,
1739                                     ,
1740                                     ,
1741                                     , NOTE - THE INSTALLATION-DEPENDENT I/O IS AT THE END
1742                                     , OF THE DICTIONARY - JUST BELOW 'TASK'. 'XI/O' IS THE
1743                                     , PRIMITIVE READ OR WRITE OF A 512-BYTE BLOCK.
1744                                     ,
1745                                     ,
1746                                     ,
1747                                     ,
1748                                     ,
1749                                     ,

```

```

1751 *****
1752 MISCELLANEOUS HIGHER LEVEL
1753 *****
1754 *****
1755 *****
1756 *****
1757 *****
1758 HEAD 301, '247, TICK, DOCOL , *****
1759 .WORD DF IND, ZEGU, ZERO, QERR, DROP, LITER, SEMIS
1760
1761 HEAD 206, FORGET, 240, FORCE, DOCOL , ***** FORGET
1762 .WORD CURR, AT, CONT, AT, SUB, LIT, 30, QERR, TICK, DUP
1763
1764 .WORD FENCE, AT, LESS, LIT, 25, QERR
1765
1766 .WORD DUP, NFA, DP, STORE, LFA, AT, CONT, AT
1767
1768 .WORD STORE, SEMIS
1769
1770 HEAD 204, BACK, 240, BACK, DOCOL , ***** BACK
1771 .WORD HERE, SUB, COMMA, SEMIS
1772
1773 HEAD 305, BEGIN, 316, BEGIN, DOCOL , ***** BEGIN
1774 .WORD QCOMP, HERE, ONE, SEMIS
1775
1776 HEAD 305, ENDIF, 306, ENDIF, DOCOL , ***** ENDIF
1777 .WORD QCOMP, TWO, QPAIR, HERE, OVER, SUB, SWAP, STORE, SEMIS
1778
1779 HEAD 304, THEN, 240, THEN, DOCOL , ***** THEN
1780 .WORD ENDIF, SEMIS
1781
1782 HEAD 302, DO, 240, DO, DOCOL , ***** DO
1783 .WORD COMP, XDO, HERE, LIT, 3, SEMIS
1784
1785 HEAD 304, LOOP, 240, LOOP, DOCOL , ***** LOOP
1786 .WORD LIT, 3, QPAIR, COMP, XLOOP, BACK, SEMIS
1787
1788 HEAD 305, +LOOP, 320, PLOOP, DOCOL , ***** +LOOP
1789 .WORD LIT, 3, QPAIR, COMP, XPLOD, BACK, SEMIS
1790
1791

```

1791 012326		HEAD	305, UNTIL, 314, UNTIL, DOCOL	; ***** UNTIL
1792 012340 002520' 004232' 004346'		. WORD	ONE, QPAIR, COMP, ZBRAN, BACK, SEMIS	
012346 000144' 012114' 001446'				
1793				
1794 012354		HEAD	303, END, 304, END, DOCOL	; ***** END
1795 012364 012336' 001446'		. WORD	UNTIL, SEMIS	
1796				
1797 012370		HEAD	305, AGAIN, 316, AGAIN, DOCOL	; ***** AGAIN
1798 012402 002520' 004232' 004346'		. WORD	ONE, QPAIR, COMP, BRAN, BACK, SEMIS	
012410 000122' 012114' 001446'				
1799				
1800 012416		HEAD	306, REPEAT, 240, REPEAT, DOCOL	; ***** REPEAT
1801 012432 001506' 001506' 012400'		. WORD	TOR, TOR, AGAIN, FROMR, FROMR, TWO, SUB, ENDIF, SEMIS	
012440 001524' 001524' 002530'				
012446 003436' 012160' 001446'				
1802				
1803 012454		HEAD	302, IF, 240, IF, DOCOL	; ***** IF
1804 012464 004346' 000144' 003362'		. WORD	COMP, ZBRAN, HERE, ZERO, COMMA, TWO, SEMIS	
012472 002510' 003416' 002530'				
012500 001446'				
1805				
1806 012502		HEAD	304, ELSE, 240, ELSE, DOCOL	; ***** ELSE
1807 012514 002530' 004232' 004346'		. WORD	TWO, QPAIR, COMP, BRAN, HERE, ZERO, COMMA	
012522 000122' 003362' 002510'				
012530 003416'				
1808 012532 002026' 002530' 012160'		. WORD	SWAP, TWO, ENDIF, TWO, SEMIS	
012540 002530' 001446'				
1809				
1810 012544		HEAD	305, WHILE, 305, WHILE, DOCOL	; ***** WHILE
1811 012556 012462' 003340' 001446'		. WORD	IF, TWOP, SEMIS	
1812				
1813				
1814				
1815				
1816 012564		HEAD	206, SPACES, 240, SPACS, DOCOL	; ***** SPACES
1817 012600 002510' 010236' 003644'		. WORD	ZERO, MAX, DDUP, ZBRAN, XXR4-, , ZERO, XDO	
012606 000144' 000014' 002510'				
012614 000334'				
1818 012616 003622' 000202' 177774	XXRA:	. WORD	SPACE, XLOOP, XXRA-	
1819 012624 001446'	XXR4:	. WORD	SEMIS	
1820				
1821 012626		HEAD	202, ^/C#/, 240, BDI08, DOCOL	; ***** <#
1822 012636 005724' 003262' 002242'		. WORD	PAD, HLD, STORE, SEMIS	
012644 001446'				
1823				
1824 012646		HEAD	202, #>, 240, EDI08, DOCOL	; ***** #>
1825 012656 002004' 002004' 003262'		. WORD	DROP, DROP, HLD, AT, PAD, OVER, SUB, SEMIS	
012664 002176' 005724' 001762'				
012672 003436' 001446'				
1826				
1827 012676		HEAD	204, SIGN, 240, SIGN, DOCOL	; ***** SIGN
1828 012710 003564' 001606' 000144'		. WORD	ROT, ZLESS, ZBRAN, XXR1-, , LIT, 55, HOLD	
012716 000010' 000060' 000055				
012724 005674'				
1829 012726 001446'	XXR1:	. WORD	SEMIS	
1830				
1831 012730		HEAD	201, #, 243, DI0, DOCOL	; ***** #

```

1832 012736 003200' 002176' 010620' . WORD BASE, AT, MSMOD, ROT, LIT, 11, OVER, LESS
      012744 003564' 000060' 000011
      012752 001762' 003502'
1833 012756 000144' 000010 000060' . WORD ZBRAN, XXR2-. , LIT, 7, PLUS
      012764 000007 001634'
1834 012770 000060' 000060 001634' XXR2: . WORD LIT, 60, PLUS, HOLD, SEMIS
      012776 005674' 001446'
1835
1836 013002
1837 013012 012734' 001762' 001762' XXR3: . HEAD 202, 08, 240, DI08, DOCOL ; ***** 08
      013020 001322' 001556' 000144' . WORD DIO, OVER, OVER, OR, ZEGU, ZBRAN, XXR3-. , SEMIS
      013026 177764 001446'
1838
1839 013032 . HEAD 203, D, R, 322, DDOTR, DOCOL ; ***** D. R
1840 013042 001506' 002026' 001762' . WORD TOR, SWAP, OVER, DABS, BDIGS, DI08, SION, EDIGS
      013050 010162' 012634' 013010'
      013056 012706' 012654'
1841 013062 001524' 001762' 003436' . WORD FROMR, OVER, SUB, SPACS, TYPE, SEMIS
      013070 012576' 004714' 001446'
1842
1843 013076 . HEAD 202, . R, 240, DOTR, DOCOL ; ***** . R
1844 013106 001506' 010106' 001524' . WORD TOR, STOD, FROMR, DDOTR, SEMIS
      013114 013040' 001446'
1845
1846 013120 . HEAD 202, D, . 240, DDOT, DOCOL ; ***** D.
1847 013130 002510' 013040' 003622' . WORD ZERO, DDOTR, SPACE, SEMIS
      013136 001446'
1848
1849 013140 . HEAD 201, . , 256, DOT, DOCOL ; ***** .
1850 013146 010106' 013126' 001446' . WORD STOD, DDOT, SEMIS
1851
1852 013154 . HEAD 201, ?, 277, QUEST, DOCOL ; ***** ?
1853 013162 002176' 013144' 001446' . WORD AT, DOT, SEMIS
1854
1855 013170 . HEAD 202, U, . 240, UDOT, DOCOL ; ***** U.
1856 013200 002510' 013126' 001446' . WORD ZERO, DDOT, SEMIS
1857
1858
1859
1860
1861 013206 . HEAD 204, LIST, 240, LIST, DOCOL ; ***** LIST
1862 ( N---, LIST GIVEN SCREEN. )
1863 013220 004510' 001056' 002054' . WORD DEC, CR, DUP, SCR, STORE, PDOT0
      013226 003076' 002242' 005104'
1864 013234 006 . BYTE 6
1865 013235 123 103 122 . ASCII /SCR # /
      013240 040 043 040
1866 . EVEN
1867 013244 013144' 000060' 000020 . WORD DOT, LIT, 20, ZERO, XDO
      013252 002510' 000334'
1868 013256 001056' 000360' 002540' XXZ1: . WORD CR, I, THREE, DOTR, SPACE
      013264 013104' 003622'
1869 013270 000360' 003076' 002176' . WORD I, SCR, AT, DLINE, XLOOP, XXZ1-. , CR, SEMIS
      013276 011524' 000202' 177754
      013304 001056' 001446'
1870
1871 013310 . HEAD 205, INDEX, 330, INDEX, DOCOL ; ***** INDEX

```

```

1872      LIST FIRST LINE OF A RANGE OF DISK SCREENS.
1873 013322 001056' 003322' 002026' . WORD CR, ONEP, SWAP, XDO
      013330 000334'
1874 013332 001056' 000360' 002540' XXZ2: . WORD CR, I, THREE, DOTR, SPACE, ZERO, I, DLINE
      013340 013104' 003622' 002510'
      013346 000360' 011524'
1875 013352 001046' 000144' 000004 . WORD QTERM, ZBRAN, XXZ3-. , LEAVE
      013360 001466'
1876 013362 000202' 177746 001446' XXZ3: . WORD XLOOP, XXZ2-. , SEMIS
1877
1878 013370 . HEAD 205, TRIAD, 304, TRIAD, DOCOL ; ***** TRIAD
1879 . LIST DISK SCREENS THREE PER PAGE.
1880 013402 000060' 000014 001020' . WORD LIT, 14, EMIT ; FORM FEED
1881 013410 002540' 010514' 002540' . WORD THREE, SLASH, THREE, STAR, THREE
      013416 010454' 002540'
1882 013422 001762' 001634' 002026' . WORD OVER, PLUS, SWAP, XDO
      013430 000334'
1883 013432 001056' 000360' 013216' XXZ4: . WORD CR, I, LIST, XLOOP, XXZ4-. , CR, LIT, 17, MESS, CR, SEMIS
      013440 000202' 177770 001056'
      013446 000060' 000017 011550'
      013454 001056' 001446'
1884
1885 013460 . HEAD 205, VLIST, 324, VLIST, DOCOL ; ***** VLIST
1886 013472 000060' 000200 003064' . WORD LIT, 200, OUT, STORE, CONT, AT, AT
      013500 002242' 003132' 002176'
      013506 002176'
1887 013510 003064' 002176' 000060' XXZ5: . WORD OUT, AT, LIT, 100, GREAT, ZBRAN, XXZ6-.
      013516 000100 003532' 000144'
      013524 000012
1888 013526 001056' 002510' 003064' . WORD CR, ZERO, OUT, STORE
      013534 002242'
1889 013536 002054' 006610' 003622' XXZ6: . WORD DUP, IDDOT, SPACE, SPACE, PFA, LFA, AT
      013544 003622' 004046' 003762'
      013552 002176'
1890 013554 002054' 001556' 001046' . WORD DUP, ZEGU, QTERM, OR, ZBRAN, XXZ5-. , DROP, SEMIS
      013562 001322' 000144' 177722
      013570 002004' 001446'
1891
1892 . IFDF LINKS
1893 . HEAD 205, VLINK, 313, XVLINK, DOCON ; ***** VLINK
1894 . THIS IS ONLY USED FOR LINKAGE FROM FORTH TO SUBROUTINES
1895 . IN OTHER LANGUAGES. SEE USER'S GUIDE FOR DOCUMENTATION.
1896 . WORD VLINK
1897 . ENDC
1898
1899
1900
1901
1902
1903

```

```

1905 ; *****
1906 ;
1907 ; INSTALLATION-DEPENDENT SECTION (TERMINAL AND DISK I/O, AND TRAPS)
1908 ;
1909 ; *****
1910 ;
1911 ;
1912 ; *****
1913 ;
1914 ; RSX-11M TERMINAL I/O
1915 ;
1916 ; *****
1917 ;
1918 ; . IFDF RSX11
1919 ;
1920 ; . EVEN
1921 ; NOTE - FOR RSX-11 ON HEAVILY LOADED MACHINES, IT IS BETTER
1922 ; FOR 'KEY' TO READ A WHOLE LINE AT A TIME, AND UNPACK IT.
1923 ; ALSO, 'KEY' SHOULD EMIT A LINE FEED WHEN A CARRIAGE RETURN
1924 ; HAS BEEN READ.
1925
1926 PEMIT: JSR R1, ITERM ; INITIALIZE RSX?
1927 ; INCREMENT 'OUT', UNLESS A CONTROL CHARACTER BEING OUTPUT.
1928 CMP (S), #40 ; TEST FOR CONTROL CHARACTER
1929 BLT 1%
1930 INC 42(U) ; INCREMENT 'OUT'
1931 1%:
1932 JSR R1, XCOU ; OUTPUT A CHARACTER
1933 NEXT
1934 ;
1935 PKEY: JSR R1, ITERM ; INITIALIZE RSX?
1936 TST ITERM ; ZERO MEANS READ NEW LINE
1937 BNE XCHAR
1938 XLINE: MOV #XBUF, ITERM ; READ NEW LINE
1939 @IOW#C IO. RVB, 4, 4, , IOSTAT, , <XBUF, 80.>
1940 ADD #XBUF, IOSTAT+2 ; TERMINATE LINE WITH CR
1941 MOVB #15, @IOSTAT+2
1942 XCHAR: TST -(S) ; DECREMENT STACK POINTER
1943 MOVB @INTERM, (S) ; FOR THIS BYTE INSTRUCTION
1944 INC ITERM
1945 BIC #177600, (S)
1946 CMP (S), #15 ; IF CR IS BEING SENT,
1947 BNE XRETRN
1948 CLR ITERM ; THEN READ NEW LINE NEXT TIME.
1949 MOV #12, -(S) ; AND ALSO EMIT A LINE FEED
1950 JSR R1, XCOU
1951 XRETRN: NEXT
1952 ;
1953 PETER: JSR R1, ITERM ; INITIALIZE RSX?
1954 MOV QFLAG, -(S)
1955 CLR QFLAG
1956 NEXT
1957 ; PUT THE AST ROUTINE HERE
1958 AST1: MOV (RP)+, QFLAG ; SET UP FOR NEXT '?TERMINAL'
1959 ; NOTE THAT 'RP' IS SYSTEM STACK.
1960 CMP QFLAG, #3 ; TEST FOR ^C
1961 BNE 1%

```

```

1962                                EXIT$S
1963                                ASTX$S
1964                                ;
1965                                ;
1966                                PCR: JSR      R1, ITERM      ; INITIALIZE RSX?
1967                                MOV      #15, -(S)
1968                                JSR      R1, XCOU
1969                                MOV      #12, -(S)
1970                                JSR      R1, XCOU
1971                                NEXT
1972                                ;
1973                                XCOU: MOV      (S)+, IOCHR
1974                                GIOW$C IO.WVB!TF.WAL, 4, 4,, IOSTAT,, <IOCHR, 1>
1975                                RTS      R1
1976                                ;
1977                                ITERM:                                ; INITIALIZE RSX IF FIRST TIME THROUGH
1978                                CMP      ITERM, #-1      ; FIRST TIME TERMINAL I/O?
1979                                BNE      RSXRTS
1980                                CLR      ITERM      ; YES
1981                                ALUN$C 4, TI, 0      ; ASSIGN LUN
1982                                GIOW$C IO.ATA, 4,, , , <AST1>      ; ATTACH - UNSOLICITED I/O
1983                                SVTK$S #TRAPV, #6      ; SET UP FOR TRAPS
1984                                RSXRTS: RTS      R1
1985                                ITERM: .WORD -1      ; FLAG FOR FIRST TIME TERMINAL I/O.
1986                                ; DO NOT REINITIALIZE 'INTERM' AT COLD START, LEST ASSIGNMENT DONE TWICE.
1987                                IOCHR: .WORD 0      ; TEMPORARY AREA FOR 'KEY' AND 'EMIT'
1988                                QFLAG: .WORD 0      ; FOR '?TERMINAL'
1989                                XBUFF: .BLKB 82.      ; TERMINAL BUFFER FOR RSX LINE I/O
1990                                ;
1991                                ;
1992                                ;
1993                                HEAD      203, BYE, 305, BYE      ; ***** BYE (LOG OFF)
1994                                CLOSE$ #FDBIO ; CLOSE DISK I/O
1995                                EXIT$S
1996                                ;
1997                                ;
1998                                ; *****
1999                                ;
2000                                ; RSX11-M DISK I/O
2001                                ;
2002                                ; *****
2003                                ;
2004                                HEAD      204, XI/O, 240, XI/O      ; ***** XI/O (RSX)
2005                                ; PHYSICAL READ-WRITE
2006                                ; ADDRESS BLOCK# FLAG ==> REPORT. READS OR WRITES A 512-BYTE BLOCK.
2007                                ; FLAG 1=READ, 0=WRITE. REPORT '0'=GOOD I/O, '1'=I/O ERROR.
2008                                ; IF REPORT IS '1', THEN NEXT ON STACK IS '1'=OPEN ERROR,
2009                                ; '2'=READ ERROR, '3'=WRITE ERROR, '4'=WAIT ERROR, '5'=ARGUMENT
2010                                ; ERROR (FLAG NOT '0' OR '1').
2011                                CLR      DSKERR ; FOR I/O ERROR REPORT
2012                                TST      OPENF ; DISK FILE ALREADY OPENED?
2013                                BNE      2$
2014                                OPEN$M #FDBIO
2015                                BCC      2$
2016                                MOV      #1, DSKERR ; ERROR IN OPEN
2017                                BR       ERRR
2018                                2$: MOV      #1, OPENF ; INDICATE FILE IS OPEN

```

```

2019      CLR      VIRBLK
2020      MOV      2(S), VIRBLK+2      ; SET UP VIRTUAL BLOCK NUMBER
2021      MOV      4(S), IOADDR      ; SET UP I/O ADDRESS
2022      TST      (S)      ; WAS TOP OF STACK - READ OR WRITE?
2023      BEQ      WRITE
2024      CMP      (S), #1
2025      BEQ      READ
2026      MOV      #5, DSKERR      ; ERROR, FLAG NOT EITHER '0' OR '1'
2027      BR       ERRR
2028      READ: READ#  #FDBIO, IOADDR, , #VIRBLK, #2
2029      BCC      WAIT
2030      MOV      #2, DSKERR      ; ERROR IN READ
2031      BR       ERRR
2032      WRITE: WRITE# #FDBIO, IOADDR, , #VIRBLK, #2
2033      BCC      WAIT
2034      MOV      #3, DSKERR      ; ERROR IN WRITE
2035      BR       ERRR
2036      WAIT: WAIT#
2037      BCC      DONE
2038      MOV      #4, DSKERR      ; ERROR IN WAIT
2039      BR       ERRR
2040      DONE: ADD      #6, S
2041      CLR      -(S)      ; INDICATE GOOD I/O
2042      BR       DONE2
2043      ERRR: ADD      #6, S
2044      MOV      DSKERR, -(S)      ; RETURN THE ERROR INDICATOR
2045      MOV      #1, -(S)      ; INDICATE ERROR IN I/O
2046      DONE2: NEXT
2047      FSRSZ# 0
2048      FDBIO: FDBDF#
2049      FDRC#A  FD. RWM
2050      FDBK#A  , 512, , 2, IOSTAT
2051      FDOP#A  3, DESCR, , FD. MFY
2052      DESCR: .WORD  0, 0      ; USE DEFAULT DEVICE
2053      .WORD  0, 0      ; AND DIRECTORY.
2054      .WORD  FILSZ, FIL
2055      FIL: .ASCII /FORTH. DAT/
2056      FILSZ# -FIL
2057      .EVEN
2058      ;
2059      OPENF: .WORD  0      ; FLAG FOR FIRST TIME DISK I/O
2060      ; DO NOT INITIALIZE 'OPENF' AT COLD START
2061      DSKERR: .WORD  0      ; SPACE FOR DISK ERROR MESSAGE
2062      IOADDR: .WORD  0      ; ADDRESS FOR DISK READ/WRITE
2063      IOSTAT: .BLKW  2      ; I/O STATUS REPORT
2064      VIRBLK: .BLKW  2      ; VIRTUAL BLOCK NUMBER
2065      ;
2066      HEAD    212, BLOCK-READ, 240, BREAD, DOCOL      ; ***** BLOCK-READ
2067      ; ( ADDRESS BLOCK# ==> REPORT). REPORT: 0=GOOD READ, ELSE ERROR
2068      .WORD  ONE, XIO, SEMIS
2069      ;
2070      HEAD    213, BLOCK-WRITE, 305, BWRT, DOCOL      ; ***** BLOCK-WRITE
2071      ; ( ADDRESS BLOCK# ==> REPORT). REPORT: 0=GOOD WRITE, ELSE ERROR
2072      .WORD  ZERO, XIO, SEMIS
2073      ;
2074      HEAD    203, I/O, 317, IO, DOCOL      ; ***** I/O
2075      ; READ OR WRITE 512-BYTE BLOCK, HANDLE ERRORS.

```

```

2076      ( ADDRESS BLOCK# FLAG(1=READ,0=WRITE) ==> )
2077      .WORD DUP,ONE,EQUAL,ZBRAN,XXS1-,DROP,BREAD
2078      .WORD ZBRAN,XXS2-,CR,PDOTG
2079      .BYTE 22
2080      .ASCII /DISK READ ERROR # /
2081      .EVEN
2082      .WORD DOT,ABORT
2083      .WORD BRAN,XXS3-
2084      .WORD ZEGU,ZBRAN,XXS4-,BWRIT,ZBRAN,XXS5-
2085      .WORD CR,PDOTG
2086      .BYTE 23
2087      .ASCII /DISK WRITE ERROR # /
2088      .EVEN
2089      .WORD DOT,ABORT
2090
2091      XXS5:
2092      XXS4:
2093      XXS3:
2094      ,
2095      HEAD 203,R/W,327,RW,DCOL      ; ***** R/W
2096      ; READ OR WRITE 1024-BYTE SCREEN ( ADDRESS SCREEN# FLAG ==> )
2097      ; NOTE THAT SCREEN N IS BLOCKS 2N-1 AND 2N.
2098      .WORD TOR,TWO,STAR,OVER,ONE,SUB,R,10
2099      .WORD SWAP,LIT,512,PLUS
2100      .WORD SWAP,FROMR,10,SEMIS
2101      ,
2102      .ENDC
2103
2104      ; *****
2105      ;
2106      ; RT-11 TERMINAL I/O
2107      ; *****
2108      ;
2109      .IFDF RT11
2110      .ITEM: CMP INTERM,#-1
2111      .BNE RTRTS
2112      .CLR INTERM
2113      .RCTRLO
2114      .TRPSET #TRAPBL,#TRAPZ      ; RESET CNTL-0
2115      RTRTS: RTS R1
2116      ,
2117      .PEMIT:
2118
2119      JSR R1,INTERM
2120      ; INCREMENT 'OUT', UNLESS A CONTROL CHARACTER BEING OUTPUT.
2121      CMP (S),#40      ; TEST FOR CONTROL CHARACTER
2122      BLT 1$
2123      INC 42(U)      ; INCREMENT 'OUT'
2124
2125      1$:
2126      JSR R1,XCOUT
2127      NEXT
2128
2129      PKEY:
2130      JSR R1,INTERM
2131      .TTYIN
2132      BIC #177600,R0
2133      CMP R0,#12
2134      BEQ PKEY
2135      MOV R0,-(S)
2136      ; IGNORE LINEFEED

```

```

2133             NEXT
2134
2135             ;
2136             ; PGTER: JSR      R1, INTERM
2137                     MOV      44, -(RP)           ; SAVE JSW
2138                     BIS      #10100, 44         ; SET BITS 6 AND 12 OF JSW
2139                     .TTINR
2140                     BCC      1$
2141             ; IF CARRY SET, NO CHARACTER - SUPPLY ZERO
2142             CLR      R0
2143             1$:     BIC      #177600, R0
2144                     CMP      R0, #12           ; IGNORE LINEFEED
2145                     BNE      2$
2146                     CLR      R0
2147             2$:     MOV      R0, -(S)
2148                     MOV      (RP)+, 44         ; RESTORE JSW
2149                     NEXT
2150
2151             ;
2152             ; PCR: JSR      R1, INTERM
2153                     MOV      #15, -(S)
2154                     JSR      R1, XCOU
2155                     MOV      #12, -(S)
2156                     JSR      R1, XCOU
2157                     NEXT
2158
2159             ;
2160             ; XCOU: MOV      (S)+, R0
2161                     .TTYOUT
2162                     RTS      R1
2163
2164             ;
2165             ; INTERM: .WORD  -1                ; FLAG FOR FIRST TIME TERMINAL I/O
2166             ; DO NOT INITIALIZE 'INTERM' AT COLD START
2167             ; IOCHR: .WORD  0                  ; TEMPORARY AREA FOR 'KEY', 'EMIT'
2168             ;
2169             ; HEAD      203, BYE, 305, BYE          ; ***** BYE (RT)
2170             ; .EXIT
2171
2172             ;
2173             ; *****
2174             ; RT-11 DISK I/O
2175             ;
2176             ; *****
2177             ;
2178             ; HEAD      204, XI/O, 240, XI/O      ; ***** XI/O (RT)
2179             ; CLR      DSKERR
2180             ; TST      OPENF
2181             ; BNE      RTOPEN
2182             ; MOV      #1, OPENF          ; INDICATE FILE IS OPEN
2183             ; NOW OPEN THE FILE
2184             ; .SETTOP # -2
2185             ; MOV      #RTSTAT, R1
2186             ; .DSTATUS R1, #RTFILE
2187             ; BCC      1$
2188             ; MOV      #1, DSKERR
2189             ; BR      RTRET
2190             ; 1$:     TST      4(R1)          ; HANDLER IN?
2191                     BNE      2$
2192                     .FETCH HANDLR, #RTFILE

```

```

2190          BCC      2$
2191          MOV      #2, DSKERR
2192          BR       RTRET
2193          2$:      . LOOKUP #LOOK1, #0, #RTFILE
2194          BCC      RTOPEN
2195          MOV      #3, DSKERR
2196          BR       RTRET
2197          RTOPEN:  ; FILE IS OPEN - NOW READ IT
2198          MOV      2(S), R1 ; BLOCK #
2199          DEC      R1          ; BEGINS AT 1 IN FORTH
2200          MOV      4(S), IOADDR ; BUFFER ADDRESS
2201          TST      (S)
2202          BEQ      WRITE
2203          CMP      (S), #1
2204          BEQ      READ
2205          MOV      #5, DSKERR
2206          BR       RTRET
2207          READ:    . READW #RTBLK, #0, IOADDR, , R1
2208          BCC      1$
2209          MOV      #6, DSKERR
2210          1$:      BR       RTRET
2211          WRITE:   . WRITW #RTBLK, #0, IOADDR, , R1
2212          BCC      2$
2213          MOV      #7, DSKERR
2214          2$:      BR       RTRET
2215          RTRET:   ADD      #6, S
2216          MOV      DSKERR, -(S)
2217          TST      DSKERR
2218          BEQ      1$
2219          MOV      #1, -(S)          ; INDICATE ERROR OCCURRED
2220          1$:      NEXT
2221          RTFILE:  . RAD50 /DK FORTH DAT/
2222          RTBLK:   . BYTE  0, 10
2223                  . WORD  0, 0, 256, 0
2224          ;
2225          OPENF:   . WORD  0          ; FLAG FOR FIRST TIME DISK I/O
2226          DSKERR:   . WORD  0          ; SPACE FOR DISK ERROR MESSAGE
2227          IOADDR:   . WORD  0          ; ADDRESS FOR DISK READ/WRITE
2228          RTSTAT:   . BLKW  4          ; DISK I/O STATUS
2229          LOOK1:    . BLKW  3          ; EMT ARGUMENT BLOCK
2230          TRAPBL:   . BLKW  2          ; EMT ARGUMENT BLOCK
2231          ;
2232          HEAD      212, BLOCK-READ, 240, BREAD, DOCOL          ; ***** BLOCK-READ
2233          ; ( ADDRESS BLOCK# ==> REPORT).  REPORT: 0=GOOD READ, ELSE ERROR
2234          . WORD    ONE, X10, SEMIS
2235          ;
2236          HEAD      213, BLOCK-WRITE, 305, BWRIT, DOCOL          ; ***** BLOCK-WRITE
2237          ; ( ADDRESS BLOCK# ==> REPORT).  REPORT: 0=GOOD WRITE, ELSE ERROR
2238          . WORD    ZERO, X10, SEMIS
2239          ;
2240          HEAD      203, I/O, 317, IO, DOCOL          ; ***** I/O
2241          ; READ OR WRITE 512-BYTE BLOCK, HANDLE ERRORS.
2242          ; ( ADDRESS BLOCK# FLAG(1=READ, 0=WRITE) ==> )
2243          . WORD    DUP, ONE, EQUAL, ZBRAN, XXS1-, , DROP, BREAD
2244          . WORD    ZBRAN, XXS2-, , CR, PDOTQ
2245          . BYTE    22
2246          . ASCII   /DISK READ ERROR # /

```

```

2247          . EVEN
2248          . WORD   DOT, ABORT
2249          . WORD   BRAN, XXS3-.
2250          . WORD   ZEGU, ZBRAN, XXS4-. , BWRIT, ZBRAN, XXS5-.
2251          . WORD   CR, PDOTQ
2252          . BYTE   23
2253          . ASCII  /DISK WRITE ERROR # /
2254          . EVEN
2255          . WORD   DOT, ABORT
2256          .
2257          .
2258          .
2259          .
2260          .
2261          .
2262          .
2263          .
2264          .
2265          .
2266          .
2267          .
2268          .
2269          .
2270          .
2271          .
2272          .
2273          .
2274          .
2275          .
2276          .
2277          .
2278          .
2279          .
2280          .
2281          .
2282          .
2283          .
2284          .
2285          .
2286          .
2287          .
2288          .
2289          .
2290          .
2291          .
2292          .
2293          .
2294          .
2295          .
2296          .
2297          .
2298          .
2299          .
2300          .
2301          .
2302          .
2303          .

```

XXS2: . WORD DOT, ABORT
 XXS1: . WORD BRAN, XXS3-.
 . WORD ZEGU, ZBRAN, XXS4-. , BWRIT, ZBRAN, XXS5-.
 . WORD CR, PDOTQ
 . BYTE 23
 . ASCII /DISK WRITE ERROR # /
 . EVEN
 . WORD DOT, ABORT
 XXS5: .
 XXS4: .
 XXS3: . WORD SEMIS
 ,
 HEAD 203, R/W, 327, RW, DDCOL , ***** R/W
 ; READ OR WRITE 1024-BYTE SCREEN. (ADDRESS SCREEN# FLAG ==>)
 ; NOTE THAT SCREEN N IS BLOCKS 2N-1 AND 2N.
 . WORD TOR, TWO, STAR, OVER, OVER, ONE, SUB, R, IO
 . WORD SWAP, LIT, 512, PLUS
 . WORD SWAP, FROMR, IO, SEMIS
 ,
 ,
 . ENDC
 ,
 ; *****
 ;
 ; STAND-ALONE TERMINAL I/O
 ;
 ; *****
 ;
 . IFDF ALONE
 PERMIT:
 ; INCREMENT 'OUT', UNLESS A CONTROL CHARACTER BEING OUTPUT.
 CMP (S), #40 ; TEST FOR CONTROL CHARACTER
 BLT 1\$
 INC 42(U) ; INCREMENT 'OUT'
 1\$: TST @#177564
 BEQ 1\$
 MOV (S)+, @#177566
 NEXT
 ;
 PKEY: ~~TST~~ @#177560
 BEQ PKEY
 CLR @#177560
 MOVB @#177562, R1
 BIC @#177600, R1
 CMP #177, R1
 BNE 1\$
 MOV #10, R1
 1\$: MOV R1, -(S)
 NEXT
 ;
 PQTER: ~~TST~~ @#177560
 BEQ 1\$
 MOV @#177562, -(S)
 BR 2\$
 1\$: CLR -(S)
 2\$: CLR @#177560

```

2304 013710                                NEXT
2305
2306 013714 005737 177564                PCR:  TST      @#177564
2307 013720 001775                        BEQ      PCR
2308 013722 012737 000015 177566        MOV      #15, @#177566
2309 013730 005737 177564                1$:  TST      @#177564
2310 013734 001775                        BEQ      1$
2311 013736 012737 000012 177566        MOV      #12, @#177566
2312 013744                                NEXT
2313
2314 013750                                ;      HEAD      203, BYE, 305, BYE                ; ***** BYE (ALONE)
2315
2316 013760 000000                                HALT
2317
2318                                ; *****
2319                                ;
2320                                ;      STAND-ALONE DISK I/O
2321                                ;
2322                                ; *****
2323                                ;
2324                                177170      RXCS=177170      ; CONTROL AND STATUS REGISTER
2325                                177172      RXDB=177172      ; DATA BUFFER REGISTER
2326                                ;
2327                                ;
2328 013762                                ;      HEAD      204, NRTS, 240, NRTS                ; ***** NRTS
2329                                ;      ADDR N TRN SECN... ADDR1 TR1 SEC1 N -> FLAG
2330                                ;      READ N SECTORS.  USES R0, R1, R2
2331                                ;      THIS OPERATION IS IN CODE TO KEEP UP WITH DISK TIMING FOR
2332                                ;      STANDARD PDP-11 SECTOR SKEWING.
2333 013774 012501                                MOV      (S)+, R1                ; # OF SECTORS TO READ
2334 013776 012702 000010                1$:  MOV      #10, R2                ; RETRY COUNT
2335 014002 012700 000007                21$: MOV      #7, R0                ; 'READ' COMMAND
2336 014006 004767 000172                JSR      PC, DRIV2                ; ADJUST R0 COMMAND
2337                                ; IF SECOND DRIVE
2338 014012 010037 177170                MOV      R0, @#RXCS                ; READ COMMAND
2339 014016 032737 000200 177170        2$: BIT      #200, @#RXCS                ; WAIT FOR TRANSFER FLAG
2340 014024 001774                        BEQ      2$
2341 014026 011537 177172                MOV      (S), @#RXDB                ; SECTOR #
2342 014032 032737 000200 177170        3$: BIT      #200, @#RXCS                ; WAIT FOR TRANSFER FLAG
2343 014040 001774                        BEQ      3$
2344 014042 016537 000002 177172        MOV      2(S), @#RXDB                ; TRACK #
2345 014050 032737 000040 177170        4$: BIT      #40, @#RXCS                ; WAIT FOR DONE FLAG
2346 014056 001774                        BEQ      4$
2347                                ; CHECK FOR ERROR
2348 014060 005737 177170                TST      @#RXCS
2349 014064 002423                        BLT      20$                ; ERROR
2350
2351 014066 012737 000003 177170                MOV      #3, @#RXCS                ; 'EMPTY' COMMAND
2352                                ;      EMPTY THE CONTROLLER'S BUFFER
2353 014074 016500 000004                MOV      4(S), R0                ; ADDRESS TO RECEIVE DATA
2354 014100 012745 000200                MOV      #200, -(S)                ; COUNT OF TIMES TO LOOP
2355 014104 032737 000200 177170        6$: BIT      #200, @#RXCS                ; WAIT FOR TRANSFER FLAG
2356 014112 001774                        BEQ      6$
2357 014114 113720 177172                MOVB     @#RXDB, (R0)+
2358 014120 005315                        DEC      (S)                ; DECREMENT THE COUNT
2359 014122 001370                        BNE      6$
2360 014124 005725                        TST      (S)+                ; POP THE COUNT

```

```

2361      ; CHECK FOR ERROR
2362      TST  @#RXCS
2363      BGE  7$
2364      20$: ; ERROR, SO RE-TRY
2365      MOV  #40000, @#RXCS
2366      BIT  177170
2367      22$: ; CLEAR ERROR STATUS
2368      BEQ  22$
2369      DEC  R2
2370      BGT  21$
2371      MOV  #-1, -(S)
2372      NEXT
2373      7$: ; ERROR EXIT
2374      ADD  #6, S
2375      DEC  R1
2376      BNE  1$
2377      ; LOOP UNLESS ALL SECTORS READ
2378      CLR  -(S)
2379      NEXT
2380      ; SUBROUTINE TO ADJUST COMMAND FOR SECOND DISK DRIVE
2381      ; NOTE - 'NMTS' ALSO USES THIS SUBROUTINE.
2382      ; NOTE USE OF R0, AND OF FORTH STACK.
2383      DRIVE2: 2(S), #114
2384      BLE  10$
2385      SUB  #115, 2(S)
2386      BIS  #20, R0
2387      RTS  PC
2388      10$: ; SUBTRACT 77
2389      ; SET UNIT-SELECT BIT
2390      ; ***** NMTS
2391      HEAD 204, NMTS, 240, NMTS
2392      ADDR TRN SECTN... ADDR1 TR1 SEC1 N -> FLAG
2393      ; WRITE N SECTORS. USES R0, R1, R2.
2394      MOV  (S)+, R1
2395      MOV  #10, R2
2396      1$: ; # OF SECTORS TO BE WRITTEN
2397      MOV  #1, @#RXCS
2398      21$: ; RE-TRY COUNT
2399      BIT  #200, @#RXCS
2400      BEQ  2$
2401      ; 'FILL' COMMAND
2402      ; WAIT FOR TRANSFER FLAG
2403      ; NOW FILL THE BUFFER
2404      MOV  4(S), R0
2405      MOV  #200, -(S)
2406      BIT  #200, @#RXCS
2407      BEQ  3$
2408      MOV  (R0)+, @#RXDB
2409      DEC  (S)
2410      BNE  3$
2411      TST  (S)+
2412      ; CHECK FOR ERROR
2413      TST  @#RXCS
2414      BLT  20$
2415      ; ERROR
2416      ; 'WRITE' COMMAND
2417      ; ADJUST IF SECOND DRIVE
2418      ; 'WRITE' COMMAND
2419      ; WAIT FOR TRANSFER FLAG
2420      ; MOVE SECTOR #
2421      ; WAIT FOR TRANSFER FLAG
2422      ; MOVE TRACK #

```

```

2418 014374 032737 000040 177170 7*: BIT #40,@#RXCS ; WAIT FOR DONE FLAG
2419 014402 001774 BEQ 7*
2420 ; CHECK FOR ERROR
2421 014404 005737 177170 TST @#RXCS
2422 014410 002015 BGE 10*
2423 014412 20*: ; ERROR SO RE-TRY
2424 014412 012737 040000 177170 MOV #40000,@#RXCS ; CLEAR ERROR STATUS
2425 014420 032737 000040 177170 22*: BIT #40,@#RXCS
2426 014426 001774 BEQ 22*
2427 014430 005302 DEC R2 ; RE-TRY COUNT
2428 014432 003306 BGT 21*
2429 014434 012745 177777 MOV #-1,-(S) ; ERROR INDICATOR
2430 014440 NEXT ; EXIT
2431 014444 062705 000006 10*: ADD #6,S ; GOOD WRITE, SO POP THE 3 ARCS
2432 014450 005301 DEC R1
2433 014452 001274 BNE 1* ; LOOP UNLESS ALL SECTORS WRITTEN
2434 ;
2435 014454 005045 CLR -(S) ; GOOD-WRITE INDICATOR
2436 014456 NEXT
2437 ;
2438 014462 HEAD 203,RTS,323,RTS,DOCOL ; ***** RTS
2439 ; ADDR TR SEC ->
2440 ; READ A SINGLE SECTOR.
2441 014472 002520' 013772' 000144' . WORD ONE,NRTS,ZBRAN,1*-.,PDOTQ
014500 000036 005104'
2442 014504 026 . BYTE 26
2443 014505 104 111 123 . ASCII /DISK READ ERROR IN RTS/
014510 113 040 122
014513 105 101 104
014516 040 105 122
014521 122 117 122
014524 040 111 116
014527 040 122 124
014532 123
2444 . EVEN
2445 014534 007574' . WORD QUIT
2446 014536 001446' 1*: . WORD SEMIS
2447 ;
2448 014540 HEAD 203,WTS,323,WTS,DOCOL ; ***** WTS
2449 ; ADDR TR SEC ->
2450 ; WRITE A SINGLE SECTOR.
2451 014550 002520' 014240' 000144' . WORD ONE,NWTS,ZBRAN,1*-.,PDOTQ
014556 000040 005104'
2452 014562 030 . BYTE 30
2453 014563 104 111 123 . ASCII /DISK WRITE ERROR IN WTS /
014566 113 040 127
014571 122 111 124
014574 105 040 105
014577 122 122 117
014602 122 040 111
014605 116 040 127
014610 124 123 040
2454 . EVEN
2455 014614 007574' . WORD QUIT
2456 014616 001446' 1*: . WORD SEMIS
2457 ;
2458 ;

```

```

2459 014620          HEAD      205, SKEW1, 261, SKEW1, DOCOL          ; ***** SKEW1
2460                ; SEQUENCE -> TRACK SECTOR
2461                ; HANDLE THE SECTOR SKEWING.
2462                ; NOTE - 'SEQUENCE #' IS ZERO-ORIGIN INDEX OF SECTOR (SKEWED).
2463                ; NOTE - 'SKEW1' DOES SKEWING OF ONLY ONE DRIVE; 'SKEW' GENERALIZES
2464                ; 'SKEW1' TO BOTH DRIVES.
2465 014632 002054' 000060' 000032          . WORD      DUP, LIT, 32, SLASH, SWAP
          014640 010514' 002026'
2466 014644 001762' 000060' 000006          . WORD      OVER, LIT, 6, STAR, OVER, DUP, PLUS, PLUS, SWAP
          014652 010454' 001762' 002054'
          014660 001634' 001634' 002026'
2467 014666 000060' 000032 010534'          . WORD      LIT, 32, MOD, LIT, 15, SLASH, PLUS
          014674 000060' 000015 010514'
          014702 001634'
2468 014704 000060' 000032 010534'          . WORD      LIT, 32, MOD, ONEP
          014712 003322'
2469 014714 002026' 003322' 002026'          . WORD      SWAP, ONEP, SWAP, SEMIS
          014722 001446'

2470                ;
2471 014724          HEAD      206, S-SKIP, 240, SSKIP, DOVAR          ; ***** S-SKIP
2472                ; VARIABLE - NUMBER OF SECTORS SKIPPED AT BEGINNING OF DISK.
2473                ; DEFAULT IS 56 DECIMAL (SKIP AN RT-11 DIRECTORY). ALSO, TRACK
2474                ; ZERO IS SKIPPED, FOR COMPATIBILITY.
2475 014740 000070          . WORD      56.
2476                ;
2477 014742          HEAD      206, S-USED, 240, SUSED, DOVAR          ; ***** S-USED
2478                ; VARIABLE - NUMBER OF SECTORS USED ON ONE DISK.
2479                ; NORMALLY, S-USED + S-SKIP = 1976. (2002. - 26. OF TR 0).
2480 014756 003600          . WORD      1920.
2481                ;
2482 014760          HEAD      204, SKEW, 240, SKEW, DOCOL          ; ***** SKEW
2483                ; SEQUENCE# -> TRACK SECTOR
2484 014772 002054' 003322' 014754'          . WORD      DUP, ONEP, SUSED, AT, GREAT, ZBRAN, 1*-
          015000 002176' 003532' 000144'
          015006 000036
2485 015010 014754' 002176' 003436'          . WORD      SUSED, AT, SUB, SSKIP, AT, PLUS, SKEW1
          015016 014736' 002176' 001634'
          015024 014630'
2486 015026 002026' 000060' 000115          . WORD      SWAP, LIT, 77, PLUS, SWAP
          015034 001634' 002026'
2487 015040 000122' 000012          . WORD      BRAN, 2*-
2488 015044 014736' 002176' 001634' 1%:          . WORD      SSKIP, AT, PLUS, SKEW1
          015052 014630'
2489 015054 001446'          2%:          . WORD      SEMIS
2490                ;
2491 015056          HEAD      206, NSETUP, 240, NSET, DOCOL          ; ***** NSETUP
2492                ; ADDR SEQUENCE# N -> ADDR N TRN SECN... ADDR1 TR1 SEC1
2493                ; THIS PREPARES A WHOLE SCREEN (IF N=8) FOR 'NRTS' OR 'NWTs'.
2494 015072 003564' 001762' 000060'          . WORD      ROT, OVER, LIT, 128, STAR, PLUS, ROT, ROT
          015100 000200 010454' 001634'
          015106 003564' 003564'
2495 015112 001762' 001634' 002520'          . WORD      OVER, PLUS, ONE, SUB, SWAP, ONE, SUB, SWAP
          015120 003436' 002026' 002520'
          015126 003436' 002026'
2496 015132 000034'          . WORD      XDO
2497 015134 000060' 000200 003436' 2%:          . WORD      LIT, 128, SUB, DUP, I, SKEW, ROT
          015142 002054' 000360' 014770'

```

```

015150 003564'
2498 015152 000060' 177777 000250' . WORD LIT, -1, XPLOO, 2*-
015160 177754
2499 015162 002004' 001446' . WORD DROP, SEMIS
2500
2501 015166 HEAD 203, R/W, 327, RW, DOCOL ; ***** R/W
2502 ; READ OR WRITE 1024-BYTE SCREEN.
2503 ; ADDR BLOCK# FLAG(R=1, W=0) ->
2504 015176 001506' 002520' 003436' . WORD TOR, ONE, SUB, LIT, 8, STAR, FROMR
015204 000060' 000010 010454'
015212 001524'
2505 ; CHANGE THE SCREEN # TO FIRST SEQUENCE #.
2506 ; IF READ, SETUP AND READ 8 SECTORS
2507 015214 000144' 000054 000060' . WORD ZBRAN, 1*- , LIT, 8, NSET, LIT, 8, NRTS
015222 000010 015070' 000060'
015230 000010 013772'
2508 015234 000144' 000030 005104' . WORD ZBRAN, 2*- , PDOTQ
2509 015242 020 . BYTE 20
2510 015243 104 111 123 . ASCII /DISK READ ERROR /
015246 113 040 122
015251 105 101 104
015254 040 105 122
015257 122 117 122
015262 040
2511 . EVEN
2512 015264 007574' . WORD QUIT
2513 015266 000122' 000050 2*: . WORD BRAN, 3*-
2514 ; SETUP AND WRITE 8 SECTORS
2515 015272 000060' 000010 015070' 1*: . WORD LIT, 8, NSET, LIT, 8, NWTs
015300 000060' 000010 014240'
2516 015306 000144' 000030 005104' . WORD ZBRAN, 4*- , PDOTQ
2517 015314 020 . BYTE 20
2518 015315 104 111 123 . ASCII /DISK WRITE ERROR/
015320 113 040 127
015323 122 111 124
015326 105 040 105
015331 122 122 117
015334 122
2519 . EVEN
2520 015336 007574' . WORD QUIT
2521 015340 4*:
2522 015340 001446' 3*: . WORD SEMIS
2523 ;
2524 . ENDC
2525 ;
2526 ; *****
2527 ;
2528 ; TRAP RECOVERY SECTION, RSX-11M
2529 ;
2530 ; *****
2531 ;
2532 . IFNDF ALONE ; STAND-ALONE MUST HANDLE OWN INTERRUPTS.
2533 HEAD 205, TRAPS, 323, TRAPS, DOCOL ; ***** TRAPS
2534 . WORD CR, PDOTQ
2535 . BYTE 14
2536 . ASCII /TRAP-ERROR,
2537 . EVEN

```

```

2538 . WORD DOT, SWAP, UDOT, UDOT, QUIT
2539 . ENDC
2540 ;
2541 . IFDF RSX11
2542 TRAPV: . WORD TRAP0, TRAP1, TRAP2, TRAP3, TRAP4, TRAP5
2543 TRAP0: CLR R1 ; TRAP # 0
2544 BR TRAPZ
2545 TRAP1: MOV #1, R1 ; TRAP # 1
2546 ADD #6, SP ; DROP MMU INFO
2547 BR TRAPZ
2548 TRAP2: MOV #2, R1
2549 BR TRAPZ
2550 TRAP3: MOV #3, R1
2551 BR TRAPZ
2552 TRAP4: MOV #4, R1
2553 BR TRAPZ
2554 TRAP5: MOV #5, R1
2555 BR TRAPZ
2556 ; TO RETURN FROM TRAP HANDLER, SET UP STACK, ETC. FOR FORTH 'TRAPS'
2557 ; DON'T USE RTT OR RTI.
2558 TRAPZ: MOV (SP)+, -(S) ; PC
2559 MOV (SP)+, -(S) ; PS
2560 MOV R1, -(S) ; TRAP #
2561 MOV #TRAPS+2, IP ; EXECUTE 'TRAPS'
2562 NEXT
2563 ;
2564 . ENDC
2565 ; *****
2566 ;
2567 ; TRAP RECOVERY SECTION. RT-11
2568 ;
2569 ; *****
2570 . IFDF RT11
2571 TRAPZ: BCS 1$
2572 ; IF CARRY CLEAR, TRAP 4
2573 MOV #4, R1
2574 BR 2$
2575 1$: MOV #10, R1
2576 2$: MOV (SP)+, -(S) ; PC
2577 MOV (SP), -(S) ; PS
2578 MOV R1, -(S) ; TRAP #
2579 MOV #3$, -(SP) ; SO RTI WILL RESTORE PC TO '3$'
2580 . TRPSET #TRAPBL, #TRAPZ ; RE-SET TRAPS
2581 RTI
2582 3$: MOV #TRAPS+2, IP ; EXECUTE 'TRAPS'
2583 NEXT
2584 . ENDC
2585 ;
2586 ;

```

```

2588 ; NOTE - 'W' ('DW') IS USED ONLY FOR TESTING - TO GET OCTAL OUTPUT
2589 ; WHEN ' ' IS NOT WORKING DURING SYSTEM DEVELOPMENT.
2590 ;
2591 ;
2592 ; HEAD 202, W, 240, DW ; ***** W
2593 ; MOV (S), XOUT
2594 ; ROL XOUT
2595 ; ROL XOUT
2596 ; MOV XOUT, IOCHR
2597 ; ROR XOUT
2598 ; BIC #177776, IOCHR
2599 ; ADD #60, IOCHR
2600 ; MOV IOCHR, -(S)
2601 ; JSR R1, XCOUNT
2602 ; MOV #5, XCOUNT
2603 ; XLP: ROL XOUT
2604 ; ROL XOUT
2605 ; ROL XOUT
2606 ; ROL XOUT
2607 ; MOV XOUT, IOCHR
2608 ; ROR XOUT
2609 ; BIC #177770, IOCHR
2610 ; ADD #60, IOCHR
2611 ; MOV IOCHR, -(S)
2612 ; JSR R1, XCOUNT
2613 ; DEC XCOUNT
2614 ; BNE XLP
2615 ; MOV #40, IOCHR
2616 ; MOV IOCHR, -(S)
2617 ; JSR R1, XCOUNT
2618 ; NEXT
2619 ; XOUT: .WORD 0
2620 ; XCOUNT: .WORD 0
2621 ;
2622 ;
2623 ;
2624 ;
2625 ;
2626 ; *****
2627 ;
2628 ; THE FOLLOWING TWO DEFINITIONS ARE NOT PURE CODE, SO THEY WERE
2629 ; MOVED HERE, NEAR THE END OF THE DICTIONARY.
2630 ;
2631 ; *****
2632 ;
2633 015342 HEAD 305, < CODE>, 305, SEMIC, DOCOL ; ***** CODE
2634 ; CREATE NEW DATA TYPE WITH CODE ROUTINE WRITTEN IN ASSEMBLY.
2635 015354 004256' 004346' 004562' .WORD QCSP, COMP, PSCOD, LBRAC, SMUDG, SEMIS
2636 015362 004374' 004440' 001446'
2637 ; NOTE: LATER, THE ASSEMBLER WILL PATCH THIS DEFINITION.
2638 ;
2639 015370 HEAD 305, FORTH, 310, FORTH, DODOE ; ***** FORTH
2640 015402 007504' .WORD DOVOC
2641 015404 120201 .WORD 120201 ; DUMMY HEADER AT INTERSECTION
2642 015406 015412' .WORD TASK-10
2643 015410 000000 XXVOC: .WORD 0 ; THE VOCABULARY LINK (FOR FUTURE USE)
2644 015412 HEAD 204, TASK, 240, TASK, DOCOL ; ***** TASK

```

2644 015424 001446'

. WORD SEMIS

2645

;

2646

;

2647

;

2648

;

```

2650                                     ; *****
2651                                     ;
2652                                     ;   STACKS AND BUFFERS
2653                                     ;
2654                                     ; *****
2655                                     ;
2656                                     ;   NOTE - 'UP', 'OPENF', 'INTERM', AND DISK BUFFERS ARE
2657                                     ;   INITIALIZED AT COLD START, OR AT FIRST TIME THROUGH.
2658                                     ;   .EVEN
2659 015426 XDP:                                     ; DICTIONARY STARTS HERE
2660 015426         .BLKB   8000.                 ; FOR DICTIONARY AND COMP. STACK
2661         ; INCREASE THIS NUMBER TO USE A LARGER MEMORY SIZE.
2662 035126 XSO:                                     ; START OF COMPUTATION STACK
2663 035126         .BLKW   2                     ; IN CASE OF EMPTY STACK
2664                                     ;
2665                                     ;
2666                                     ;
2667                                     ;
2668                                     ;
2669 035132 DSKBUF:                                ; ROOM FOR 3 1K DISK BUFFERS
2670         ; INITIALIZE BUFFERS' UPDATE BITS, AND TERMINATING NULLS, TO ZERO.
2671         ; NOTE - THESE BUFFERS ARE CLEARED AT COLD START, ANYWAY,
2672         ; BECAUSE A STAND-ALONE BOOT MAY NOT INITIALIZE HIGH MEMORY,
2673         ; AND ALSO SO THAT THE NUMBER OR LOCATION OF BUFFERS CAN BE
2674         ; CHANGED AT RUN TIME.
2675 035132 000000         .WORD   0
2676 035134         .BLKB   1024.
2677 037134         .WORD   0
2678 037136 000000         .WORD   0
2679 037140         .BLKB   1024.
2680 041140 000000         .WORD   0
2681 041142 000000         .WORD   0
2682 041144         .BLKB   1024.
2683 043144 000000         .WORD   0
2684 043146 ENDBUF:                                ; CAUTION - 'ENDBUF' - 'DSKBUF' MUST BE EXACT MULTIPLE
2685         ; OF THE BUFFER LENGTH PLUS 4.
2686                                     ;
2687                                     ;
2688                                     ;
2689                                     ;
2690                                     ;
2691         ; 'XTIB', 'XR0', AND 'XUP' ARE ONLY USED IN BOOT-UP TABLE;
2692         ; THEREFORE THE AREAS DEFINED HERE CAN BE MOVED AT RUN TIME.
2693 043146 XTIB:         .BLKW   42.                 ; TERMINAL INPUT BUFFER
2694 043272         .BLKW   50.                     ; FOR RETURN STACK
2695 043436' XR0=
2696 043436 XUP:         .BLKW   100                 ; ROOM FOR 100 USER VARIABLES
2697                                     ;
2698                                     ;
2699         .IFDF   RT11
2700         ; DISK HANDLER GOES HERE
2701 HANDLR: .WORD   .+2
2702         .ENDC
2703                                     ;
2704                                     ;
2705         ; NOTE - CHANGE THE FOLLOWING LINE TO '.END' IF LINKING TO OTHER LANGUAGES.
2706 000000'         .END   ORIGIN

```

ABORT	007652R	DOES	004632R	INDEX	013320R	PORIO	002632R	THEN	012214R
ABS	010134R	DOT	013144R	INTER	007274R	PQTER	013666R	THREE	002540R
AGAIN	012400R	DOTQ	005140R	IP	=%000004	PREV	003310R	TIB	002676R
ALLOT	003402R	DOTR	013104R	KEY	001030R	PSCOD	004562R	TICK	011764R
ALONE	= 000001	DOUSE	002474R	LATES	003742R	PSTOR	002072R	TOGOL	002124R
AND	001302R	DOVAR	002450R	LBRAC	004374R	QALIO	005226R	TOR	001506R
ARROW	011724R	DOVOC	007504R	LEAVE	001466R	QCOMP	004152R	TRAV	003674R
AT	002176R	DP	002756R	LESS	003502R	QCSP	004256R	TRIAD	013400R
BACK	012114R	DPL	003212R	LFA	003762R	QERR	004120R	TWO	002530R
BASE	003200R	DPLUS	001652R	LIMIT	003026R	GEXEC	004202R	TWOP	003340R
BBUF	002600R	DRIV2	014204R	LINK	= 015412R	GLOAD	004314R	TYPE	004714R
BCOMP	007032R	DRONE	011156R	LINK2	= 015412R	GPAIR	004232R	U	=%000003
BDIOS	012634R	DROP	002004R	LIST	013216R	GSTAC	007214R	UDIV	001226R
BEGIN	012136R	DRZER	011136R	LIT	000060R	QTERM	001046R	UDOT	013176R
BL	002552R	DSKBUF	035132R	LITER	007066R	QUERY	005446R	ULESS	007154R
BLANK	005654R	DTRAI	005020R	LOAD	011642R	QUEST	013160R	UMULT	001144R
BLK	003040R	DUP	002054R	LOOP	012256R	QUIT	007574R	UNTIL	012336R
BLOCK	011314R	ECELL	004770R	MAX	010236R	R	001540R	UPDAT	010740R
BRAN	000122R	EDIOS	012654R	MESS	011550R	RBRAC	004412R	UPPER	006416R
BSCR	002614R	ELSE	012512R	MIN	010206R	REPEAT	012430R	USE	003274R
BUFFE	011204R	EMIT	001020R	MINUS	001712R	RNUM	003250R	USER	002466R
BUILD	004612R	ENCL	000676R	MLOOP	000566R	ROT	003564R	USLAS	001214R
BYE	013756R	ENC1	000712R	MOD	010534R	RP	=%000006	USTAR	001132R
CAT	002216R	ENC2	000726R	MSLAS	010352R	RPSTO	001426R	VAR	002442R
CENT	007730R	ENC3	000744R	MSMOD	010620R	RTS	014470R	VLIST	013470R
CFA	004002R	ENC4	000772R	MSTAR	010266R	RW	015174R	VOCAB	007444R
CL	002564R	END	012362R	MTBUF	011006R	RXC8	= 177170	VOCL	002776R
CMOVE	001070R	ENDBUF	043146R	NFA	004020R	RXDB	= 177172	W	=%000002
COLD	007726R	ENDIF	012160R	NOFAST	000560R	RZERO	002664R	WARN	002730R
COLON	002314R	EQUAL	003452R	NOFST1	000604R	S	=%000005	WENT	010012R
COMMA	003416R	ERASE	005632R	NRTS	013772R	SCR	003076R	WHILE	012554R
COMP	004346R	ERRDR	006530R	NSET	015070R	SCSP	004074R	WIDTH	002712R
CON	002406R	EXEC	000102R	NULL	005476R	SEMI	002354R	WORD	005750R
CONT	003132R	EXPEC	005254R	NUMB	006206R	SEMIC	015352R	WTS	014546R
COUNT	004670R	FAILED	000646R	NWTS	014240R	SEMIS	001446R	W2	010016R
CR	001056R	FAST	000532R	OCTAL	004534R	SIGN	012706R	XDO	000334R
CREAT	006700R	FCOMP	000532R	OFSET	003114R	SKEW	014770R	XDP	015426R
CSP	003236R	FENCE	002744R	ONE	002520R	SKEW1	014630R	XLOOP	000202R
CSTOR	002270R	FILL	005572R	ONEP	003322R	SLASH	010514R	XMATCH	000544R
CURR	003150R	FIRST	003012R	OR	001322R	SLMOD	010474R	XOR	001340R
DABS	010162R	FLD	003224R	ORIGIN	000000R	SMUDQ	004440R	XPLOO	000250R
DDOT	013126R	FLUSH	011040R	OUT	003064R	SPACE	003622R	XR0	= 043436R
DDOTR	013040R	FORQE	012016R	OVER	001762R	SPACS	012576R	XS0	035126R
DDUP	003644R	FORTH	015400R	PABOR	006512R	SPAT	001366R	XTIB	043146R
DEC	004510R	FROMR	001524R	PAD	005724R	SPSTO	001406R	XUP	043436R
DEFIN	007532R	GFORTH	000000R	PAREN	007552R	SSKIP	014736R	XXB1	007606R
DFIND	006336R	GO	010056R	PBUF	010656R	SSLA	010576R	XXB2	007636R
DIQ	012734R	GREAT	003532R	PCR	013714R	BSMOD	010554R	XXD2	006726R
DIQIT	000400R	HERE	003362R	PDOTQ	005104R	STAR	010454R	XXD6	007106R
DIQS	013010R	HEX	004462R	PEMIT	013574R	STATE	003164R	XXA	007340R
DLINE	011524R	HLD	003262R	PFA	004046R	STOD	010106R	XXE1	006472R
DLITE	007124R	HOLD	005674R	PFIND	000506R	STORE	002242R	XXE2	006430R
DMINU	001734R	I	000360R	PKEY	013624R	SUB	003436R	XXE3	006404R
DO	012230R	IDDOT	006610R	PLINE	011462R	SUSED	014754R	XXE4	007276R
DOCOL	002340R	IF	012462R	PLPOP	012306R	SWAP	002026R	XXE5	007326R
DOCON	002420R	IMMED	007412R	PLUS	001634R	SZERO	002652R	XXE6	007332R
DODOE	004646R	IN	003052R	PNUMB	006072R	TASK	015422R	XXE7	007372R

XXFA	006324R	XXK1	005266R	XXN4	006546R	XXT1	010714R	XXW6	005030R
XXF3	006074R	XXK2	005346R	XXN5	007144R	XXT2	011216R	XXW7	005062R
XXF4	007364R	XXK3	005424R	XXRA	012616R	XXT3	011264R	XXZ1	013256R
XXF5	007370R	XXK4	005376R	XXRB	010176R	XXT4	011440R	XXZ2	013332R
XXF6	006244R	XXK5	005400R	XXR1	012726R	XXT5	011354R	XXZ3	013362R
XXF7	006312R	XXL1	004734R	XXR2	012770R	XXT6	011406R	XXZ4	013432R
XXG4	006170R	XXL2	004752R	XXR3	013012R	XXT7	011110R	XXZ5	013510R
XXQ5	006162R	XXL3	004754R	XXR4	012624R	XXVOC	015410R	XXZ6	013536R
XXI1	005774R	XXL6	005202R	XXR5	010190R	XXWA	005066R	ZBRAN	000144R
XXI2	006000R	XXL7	005212R	XXR6	010254R	XXW3	011610R	ZEGU	001556R
XXJ1	005550R	XXN1	003700R	XXR7	010224R	XXW4	011630R	ZERO	002510R
XXJ2	005554R	XXN2	004136R	XXTA	011054R	XXW5	011614R	ZLESS	001606R
XXJ4	005560R	XXN3	004140R						

. ABS. 000000 000
043636 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 2454 WORDS (10 PAGES)
DYNAMIC MEMORY: 18694 WORDS (71 PAGES)
ELAPSED TIME: 00:01:51
FORTH, FORTH=FORTH

Boot Block

```

1      ; BOOT BLOCK.
2      ;
3      ; THIS CODE IS IN TRACK 1 SECTOR 1 OF THE DISK, FOR STAND-ALONE BOOT.
4      ;
5      ; THIS BOOT BLOCK READS A 2-SECTOR LOADER (256 BYTES) INTO THE LAST
6      ; 256 BYTES OF THE FIRST 8K OF MEMORY. THIS LOADER IS ON DISK IN THE
7      ; LAST TWO SECTORS OF SCREENS 40-47 OF THE 'FORTH.DAT' FILE ( THESE
8      ; ARE SECTORS 26 AND 30 - OCTAL - OF TRACK 21). WHEN THE LOADER
9      ; EXECUTES IT READS IN THE REST OF THE BINARY IMAGE (STARTING AT
10     ; ZERO) AND BRANCHES TO ADDRESS 1000 TO START EXECUTION OF THE
11     ; LOADED PROGRAM. THE LOADER, WHICH IS TABLE-DRIVEN, IS CREATED
12     ; AND WRITTEN TO DISK BY FORTH SCREENS 36 AND 37.
13     ;
14     000000 000240
15     000002 000415
16     000020'
17     000020 000064
18     000022 000340
19     000034'
20     000034 000000
21     000036 012700 017400
22     000042 012706 001000
23     000046 012704 177170
24     000052 000004
25     000054 000413
26     000060'
27     000060 000034'
28     000062 000341
29     000064 005714
30     000066 001776
31     000070 100743
32     000072 000002
33     000100'
34     000100 000072'
35     000102 000340
36     000104 010405
37     000106 005725
38     000110 012701 000026
39     000114 012714 000007
40     000120 000004
41     000122 010115
42     000124 000004
43     000126 012715 000021
44     000132 000004
45     000134 012714 000003
46     000140 000004
47     000142 111520
48     000144 000004
49     000146 105714
50     000150 100774
51
52     000152 022701 000026
53     000156 001402
54     000160 000167 017214
55     000164 012701 000030
56     000170 000751
57     000001

; ORIGIN: NOP
; BR SETUP
; =ORIGIN+20
; .WORD 64 ; IOT TRAP - USED FOR WAIT SUBROUTINE
; .WORD 340 ; SET PRIORITY 7
; =ORIGIN+34
; ERROR: HALT
; SETUP: MOV #17400,R0 ; 2-SECTOR LOADER WILL START AT 17400
; MOV #1000,SP
; MOV #177170,R4
; IOT ; WAIT FOR DISK TO FINISH
; BR CONTIN ; SKIP OVER INTERRUPT VECTOR
; =ORIGIN+60
; .WORD ERROR ; HALT IF TERMINAL INTERRUPT
; .WORD 341 ; FLAG (WHICH HALT)
; WAIT: TST (R4) ; THIS SUBROUTINE IS CALLED BY IOT
; BEQ WAIT ; WAIT FOR SOMETHING
; BMI ORIGIN ; IF ERROR, START OVER (BRANCH TO ZERO)
; RETURN: RTI
; =ORIGIN+100
; .WORD RETURN ; IF CLOCK INTERRUPT, JUST RTI
; .WORD 340
; CONTIN: MOV R4,R5 ; PUT #177172
; TST (R5)+ ; IN R5
; MOV #26,R1 ; READ SECTOR 26 FIRST
; RDLOOP: MOV #7,(R4) ; START THE READ OPERATION
; IOT ; WAIT (FOR TRANSFER FLAG)
; MOV R1,(R5) ; MOVE SECTOR # TO CONTROL REGISTER
; IOT
; MOV #21,(R5) ; MOVE TRACK #
; IOT
; MOV #3,(R4) ; EMPTY BUFFER COMMAND
; IOT
; CHLOOP: MOVB (R5),(R0)+ ; THIS LOOP MOVES 200 CHARACTERS
; IOT
; TSTB (R4) ; TEST FOR TRANSFER FLAG
; BMI CHLOOP
; WHEN GET HERE, DONE TRANSMITTING ONE SECTOR'S CHARACTERS
; CMP #26,R1 ; GO AROUND AGAIN IF TRACK WAS 26
; BEQ 1$
; JMP ORIGIN+17400 ; ELSE JUMP TO THE LOADER AT 17400
1$: MOV #30,R1 ; SECOND (AND LAST) TIME, READ SECTOR 30
; BR RDLOOP
; END

```

.MAIN. MACRO M1113 20-JAN-80 18:10 PAGE 1-1
SYMBOL TABLE

CHLOOP	000142R	ERROR	000034R	RDLOOP	000114R	SETUP	000036R
CONTIN	000104R	ORIGIN	000000R	RETURN	000072R		

.ABS. 000000 000
000172 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 89 WORDS (1 PAGES)
DYNAMIC MEMORY: 18694 WORDS (71 PAGES)
ELAPSED TIME: 00:00:04
BOOT, BOOT=BOOT

WAIT 000064R

Boot Block

