



Handwritten text on a yellowed, torn piece of paper. The text is written in red ink and appears to be a list or a set of instructions. The characters are stylized and somewhat difficult to decipher, but they seem to be arranged in a vertical column. The paper is heavily aged and has irregular edges.



1) Command processor, but BEAD GHOST

ai 2) an example with the ^{BASIC} BGR

3) temporary disk limits
fixed ECS limits

4) CAL TSS has a bug ??

~~5) ToB
see Scope?~~

~~6) Return Key ??~~

7) What namespace 'get' in SCOPE?

8) Good terminology for cell /
SCOPE calls

~~9) Shazam~~

~~10) "prompt"~~

C,DISK,S
???
Q
BEAD HERE
C,DISK,S
DISK HERE
GET,ECSINIT
OBTAINED
FIN
BEAD HERE
C,PRINTER,S,ECSINIT↑
SPACE,100000
OK

TOO MUCH
C,DISK,S
DISK HERE
GET,XTEXT
OBTAINED
<L
OBTAINED
OBTAINED
OBTAINED
OBTAINED
OBTAINED
OBTAINED
OBTAINED
OBTAINED
OBTAINED
OBTAINED
OBTAINED
LIST EXHAUSTED
FIN
BEAD HERE
LOGOUT
VV
LOGGED OUT
TRIM
OK
SPA ←CE,30000
OK
C,DI↑
USER,VV
REMEMBER TO LOGOUT
C,DISK.S
BAD SUB
C,DISK,S
DISK HERE

*List of VV & GENE files
on old DISK,S - check against
my stuff now*

LIST, VV		
XTAXI	400	311
TTEST	400	315
W	400	325
BINARY	33400	401
TEVCH1	3400	721
LAST	400	745
CLIST	400	755
SUBPROC	13400	1001
TALOC1	4000	1271
ISUBPS	2400	1311
TEVCH	25400	1325
TCLST1	2000	1341
TCLST	15000	1345
TCOMP	4000	1351
TMISC1	1000	1375
TMISC	13000	1401
ALLOC	20400	1461 ✓
IFILE	13000	1771
TSUBP1	1000	2001
TSUBP	20000	2005
IFILE1	1400	2325
TMACROS	2400	2365
ECSINIT	2400	2561 ✓
TALOC	12400	3031
ECSMAC?	400	3211 -
HOROR	1000	4065
TTIME1	1000	4355
TTIME	13000	4361
TCOMP1	1400	4365
MAPS	12000	4505 -
SWAP	4400	4545 -
SYSENT	15000	4561 -
PROCESS	6400	4635 ✓
SNO	1000	1000 4641 ✓
ECSACT	1400	4661 ✓
INITL	7400	4775 ✓
INITL	1400	5411
RESULT	2400	2400 5451

LIST COMPLETE

ENTER TITLE LINE

STURGIS

SCOPE FORMAT CONVENTIONS ?

YES

EVCH CLEARED

PRINTING ...

COMMAND PROCESSOR HERE

!LOGOFF

GOOD DAY

CAL TSS VERSION 1.2

~~TEMPORARY~~ DIRECTORY?

.KEITH

COMMAND PROCESSOR HERE

!SERV

SERVICES HERE

*FIN

COMMAND PROCESSOR HERE

!DISKSLD

MOUNT DISK,S DUMP TAPE

ENTER LOAD, LIST, OR FIN

↑LIST

ENTER USER NAMES TO DLIST OR ↑.↑ TO QUIT (\$\$ LISTS USERS)

+VV

+. .

BEGIN SCANNING TAPE

XTEXT VV 400

TTEST VV 400

W VV 400

BINARY VV 34000

TEVCH1 VV 3400

LAST VV 400

CLIST VV 400

SUBPROC VV 13400

TALOC1 VV 4000

ISUBPS VV 2400

TEVCH VV 25400

TCLST1 VV 2000

TCLST VV 15000

TCOMP VV 4000

TMISC1 VV 1000

TMISC VV 13000

ALLOC VV 24000

TFILE VV 13000

TSUBP1 VV 1000

TSUBP VV 20000

TFILE1 VV 1400

TMACROS VV 2400

ECSINIT VV 2400

TALOC VV 12400

ECSMAC VV 400

HOROR VV 1000

TTIME1 VV 1000

TTIME VV 13000

TCOMP1 VV 1400

MAPS VV 12000

SWAP VV 4400

SYSENT VV 15000

PROCESS VV 12400

SNO VV 1000

ECSACT VV 1400

INITL VV 7000

VTEXT VV 400

PROCSYM VV 1400

END OF DISK,S TAPE

ENTER LOAD, LIST, OR FIN

↑

- ✓ 1 Dec '70 TCLST bug 14 Dec '70
- ✓ " remove stack error from jump call 14 Dec '70
- ✓ " put option bits on TRDB, SETLIN 14 Dec '70
- 5 " more stringent stop limit on blocks
- 5 Dec '70 compactor timing
- " compactor test
- ✓ " design the "AB Cruiser" & finish 4 Jan 71
- designing the AB's
- ✓ 7 Dec use C or change unique name 4 Jan 71
- 16 Dec repair systems vis-a-vis CL140 failure
- ~~also, 4560-90?~~
- 17 Dec recover TSUBP stuff
- 28 Dec get Compactor out of central
- 28 Dec fix up " timing & display compent,
- ✓ 28 Dec MOVE BLOCK ACCOUNTING TOLC. Jan '71
- 28 Dec Block-in-map / Change U.N. handle
- 4 Jan Change UN count on maps
- 1 Feb Chgwd, letdce, Cbrdae
- 1 Feb Det TYPECODE Table out of CM
- 1 Feb Redo ECSSUB
- 9 June '71 Fix MOVMT to conform to DONATE
- " ~~opt~~ steal NOT slots
- " option bits for new style AB's

FOR M4B

✓ ~~fake return if m4ref is 1??~~ ~~bug Paul no~~

✓ ~~Whether interrupt status after sub call?~~
~~cell 2~~

? Page 37???

Page 19 - 3,2 error on more block
Write up jump call

Errors on return with parameters

Where are capability option bit these
days (vis-a-vis copy cop) - low bits

What gives display process?

Option bits on none $\rightarrow$ any cop?

Display system clock times

Return & redo op

✓ ~~Revised m4 operations (page follow)~~

Find with son

Parameters to Change unique row

✓ ~~New AB operations~~

Change to OPER macro in INITL, to facilitate putting names in the directory

1) ~~not~~ remake OPER! "~~OPER MACRO NAME, ACTION, ...~~"
"MACRO OPER, NAME, ACTION, ..."

2) remove all the actual oper macros, starting with read file, to an XTEXT file called OPER, XTEXT; call in the file appropriately

3) add the C. names to the location field of each occurrence of the oper macro (ignored by initl.)

Talk to Howard

LITTLE THINGS

31 May 1)
700

fix S.ECSRA so that they won't assemble
S.ECSFL
unless the low 6 bits are zero; then fix the
allocation writing.

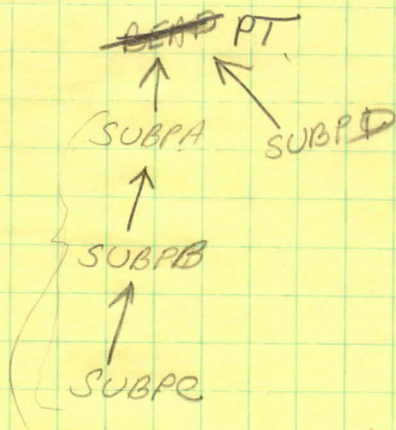
checks for subprocess call

1. not room in stack
2. not room is subp FL for params
3. not room in subp local C-list for params
4. call non-existent subp

checks for block params

- Block data
1. negative user pointer
 2. block extends beyond FL (check boundary)
 3. user count >T. 'max count' in op
 4. in called subp - block is properly placed in param area - 1st word is actual size of block - next param is 'max count' + 1
 5. negative index
 6. block spans local to next C-list.
 7. block not start in local C-list
 8. block spans 2 non local C-list
 9. block extends too far
 10. C-list gone
 11. param is passed as $\frac{1}{\text{count}} \frac{\text{actual index}}{\text{count}}$
next cap is at index + max count
 12. user supplied count is >T. 'max count'

availability
blocks



0

- 1) enter subp C.
- 2) call subp A
- 3) do return w/params

check →

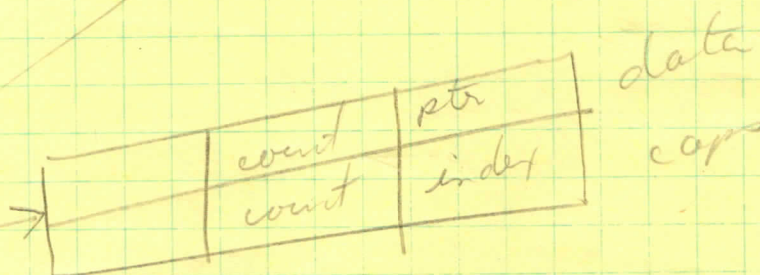
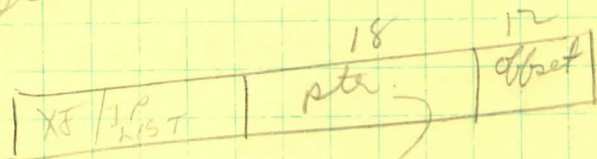
- 1) negative counts and starting ptrs ✓
- 2) data return block not all within full FL ✓
- 3) capability block not all within full C-list ✓
- 4) capability block completely in local C-list of subp A ✓
- 5) capability block in local C-list of A and also partly in C-list of B ✓
- 6) capability block partly in C-list of B and partly in C-list of C ✓
- 7) return control - neg counts ✓
- neg ptr to return control ✓
- zero ptr to return control ✓
- neg ptr or index in return control ✓
- proper count used on return ✓
- return block in range (C-list, FL) ✓
- 8) #params > ECS buffer

- 1) enter subp C
- 2) call subp A
- 3) call subp D
- 4) do return w/params - check 1)

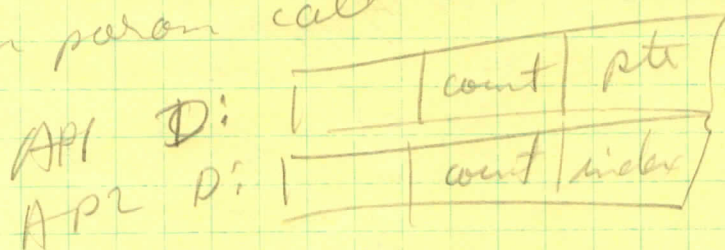
- a) cap blk falls in local C-list of A
- b) cap blk spans local C-list and C-list of B
- c) cap blk spans C-list of B and C-list of C

make sure gets in ECS copy of local C-list

return authority

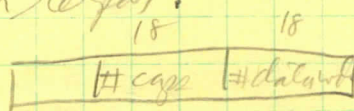


return param call



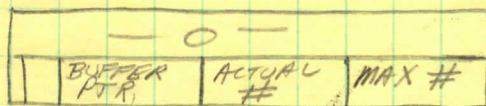
ECS buffer control on regar:

atp. PAPAUC



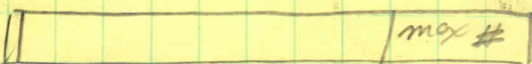
in AP area

block param



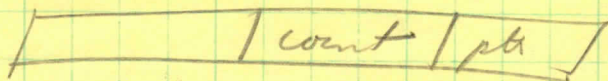
↑ abs ccs addn
0 = data
1 = capability

P.S. data - block param



↑
0 = data
1 = capability

IP/LST



TITLE Test capability across Clist boundary structure
EJECT

SP1060 TOPER 1,,,K.CALLC,(42/10,18/CSTORE+1),(42/10,18/CCAP+1), 8,
K.CALLA, 0, 0, 0,
C.RETPAR, 0, (42/4, 18/ACAP+1), 0,
C.Return, 0, 0, 0,
0, 0, 0, 0.

SP1061 Taper  BCAP_F1

SP1062 Tgor  CCAP+1
F-

SP1063 $T_{\text{for}} =$ ACAP_{±28}

SP 1064 T_{for} = BCAP+28

SP1065 $T_{\text{por}} =$ _____ 90-5

SP1066 Taper 3, 3, 3, —
90-4

TITLE various errors in return control parameters EJECT

* normal, base-line, really works parameters

SP1070 Tqor $\overset{K.CALLC}{1, 1, 1} (42/2, 18/SPFLC-2), (42/2, 18/28), 8,$
 $K.CALLA, 0, 0, 0,$
 $\Delta.RetPar, (42/2, 18/ASTORE+1), (42/2, 18/ACAP+1), 0,$
 $C.Return, 0, 0, 0,$
 $0, 0, 0, 0$

* neg par ret count

SP1071 $3, ?, ?, K.CALLC, (42/-1, \text{---})$

* neg cap ret count

SP1072 $(42/2), (42/-1)$

* par ret count + ptr too large

SP1073 Tqor $3, ?, ?, K.CALLC, (42/2, 18/SPFLC-1), (\text{---})$

* ~~par~~ ret count + index too large

SP1074 Tqor $3, ?, ?, K.CALLC, (\quad), (42/2, 18/29)$

* neg par ret ptr

SP1075 Tqor $3, ?, ?, K.CALLC, (42/2, 18/-1) \text{---}$

* neg cap ret index

SP1076 $(\quad 18/SPFLC, (42/2, 18/-1)$

* no return authorized

SP1077 Taper 3, ?, ?, K.CALLA, 0, 0, 0,
C. Retgar, (42/2, 18/astr+1), (42/2, 18/aup+1), 0,
0, 0, 0, 0,
0, 0, 0, 0,

* ret auth'd, but ^{0, 0, 0, 0} nothing ret'd
SP1078
* neg ptr to ret auth

~~SP1078 Taper 3, ?, ?~~

DOPER C.WFILE, K.FILE, BEG SPC+LOOPS, INST 1, 1
USE CODE
INST 1 VFD 12/0130B, 18/-6, 18/-1, 1000
USE *

SP1079 Taper 3, ?, ?, K.CALLA, (42/2, 18/astr+1), (42/2, 18/aup+1), 0,
K.CALLA, 0, 0, 0,
C.Retgar, 0, 0, 0,
C.Return, 0, 0, 0,
0000

SP1079 same but return $\leftrightarrow$ Retgar

* ptr to ret auth $\geq$ FL

Doper INST 2
USE
INST 2 VFD 12/0130B, 18/-6, 18/SPFLC, 12/1000
USE

SP1080 Taper |

* Too few returns, too many dots

SP1081 T ofor 3, ?, ?, K. Calla, (42/2, 18/CSTORE+1), (42/2, CCAH), 8,
K. Calla, 0, 0, 0,
C. Retfor, (42/3, 18/ASTORE+1), (—), 0
C. Return

* Too many caps

SP1082 T ofor $\leq$
C. Retfor, (42/2, 18/ASTORE+1), (42/3, 18/ASTORE+1), 0,
—

372 * return more stuff than the buffer will hold

SP1083 T ofor 3, ?, ?, K. Calla, 0, 0, 0,

K. Calla, 0, 0, 0,

C. Retfor, (42/251, 18/0),

0

0

192

64

256

280

①

* PAGE

*

* TEST FILE CREATION OPERATION.

*

* CREATE.FILE(^{C:}AB, ^{D:}WC.INDEX, ^{D:}NSN, ^{D:}SNL)

*

* AB AND WC.INDEX ARE RESPECTIVELY THE ALLOCATION BLOCK
* TO BE CHARGED AND THE INDEX IN THE WORKING C-LIST TO
* RETURN THE NEW FILE CAPABILITY NO TESTS WILL BE RUN CONCERNING THIS

*

* NSN IS THE NUMBER OF SHAPE NUMBERS THE FILE IS
* TO HAVE. THIS NUMBER MUST BE POSITIVE (>0) AND
* ^{NO GREATER} ~~SMALLER~~ THAN P.SCRL-3 SINCE THE FILE DESCRIPTION
* MUST FIT IN P.SCR, AND ITS LENGTH IS $3+NSN$.

*

* CREATE 1 LEVEL FILE.C.FILE,

F1 TOPER N,,, ~~TEMP~~ AB, C.TEMP, 1, SHAPES

*

* CAUSE Z, 0, 3 ERROR (NON POSITIVE NSN)
*, C.FILE,

F2 TOPER E, 60/2, (42/3, 18/0), AD, C.TEMP, 0, SHAPES

*

* CAUSE Z, 1, 3 ERROR (NSN TOO LARGE)
*, C.FILE,

F3 TOPER E 60/2, (42/3, 18/1), AB, C.TEMP, P.SCRL-2, SHAPES

*

* CREATE (P.SCRL-3) LEVEL FILE.

*

, C.FILE,
F4 TOPER N,,, AB, C.TEMP, P.SCRL-3, SHAPES

(2)

* SNL IS THE ADDRESS OF A LIST OF SHAPE NUMBERS.
* IT MUST BE NON NEGATIVE, AND NOT GREATER THAN
* FL-NSN.

*
* ~~CREATE FL~~ ^{CAUSE} ~~2,2,4~~ ERROR (SNL IS NEGATIVE)
*

F5 TOPER E,60/2,(42/4,19/2),C.CFILE,AB,C.TEMP,1,-1

*
* CAUSE 2,1,4 ERROR (SNL+NSN >= FL)
*

F6 TOPER E,60/2,(42/4,18/1),C.CFILE,AB,C.TEMP,1,FL-1

* THE SHAPE NUMBER LIST CONSISTS OF A SERIES OF 60-BIT
* ITEMS, ALL AFTER THE FIRST OF WHICH SHOULD BE A
* NON-NEGATIVE POWER OF TWO ^{AND ALL MUST BE <} ~~LESS THAN~~ ^{2¹⁷} THE ^{PRODUCT} ~~SUM~~
* OF THE SHAPE NUMBERS MUST BE < 2⁵⁹.

*
* ^{CAUSE} ~~CREATE~~ 3,7 ERROR ^{NON POSITIVE} ~~(NEGATIVE~~ SHAPE NUMBER)
*

F7 TOPER E,60/3,~~60/7~~ 60/7,C.CFILE,AB,C.TEMP,1,SNL.^{ZERO}~~ONE~~

*
* CAUSE 3,8 ERROR (SHAPE NUMBER >= 2¹⁷)
*

F8 TOPER E,60/3,60/8,C.CFILE,AB,C.TEMP,1,SNL.BIG

*
* CAUSE 3,9 ERROR (SHAPE NUMBER OTHER THAN S1 NOT POWER)
*

F9 TOPER E,60/3,60/9,C.CFILE,AB,C.TEMP,2,^{NOT POW} ~~SHPS~~

③

* CAUSE 3, 10 ERROR (TOTAL SIZE OF FILE $\geq 2^{159}$)

*

F10 TOPER E, 60/3, 60/10, C.CFILE, AB, C.TEMP, 4, BIGFILE

*

* CREATE VERY LARGE FILE (LENGTH = 2^{158})

*

F11 TOPER BN,,, C.CFILE, AB, C.TEMP, 4, NOTBIG

*

SHAPES BSS 0

DUP P.SGRL-2,1

DATA 1 .. 1=210 IS A GOOD SHAPE NUMBER

*

SNL. ^{ZERO}~~ONE~~

DATA 0 .. ILLEGAL SHAPE NUMBER

SNL.BIG

DATA 131072 .. =217, TOO LARGE FOR A SHAPE #

*

^{NOTPOW}
~~BADSHPS~~

DATA 100,3 .. SECOND SHAPE NUMBER NOT A POW of 2

*

BIGFILE

DATA ~~2048~~ 4097,
~~65536~~ 65536, 65536, 65536 .. $1^2 + 3 \times 16 = \text{~~59~~} > 59$

*

NOT2BIG

DATA 2048, 65536, 65536, 65536 .. $11 + 3 \times 16 = 59$

*

*

THREEEN

DATA 2,2,2

MACRO ECSSUB, NAME, BUF

NAME {

SAO BUF

SXO 0000

RECS ~~LENGTH~~ ~~QRBL~~ ~~QRBL~~ 0000 B|ECSS

EQ BUF RECS

USE 'CBLOCK' 2 1/1,

VFD 30/ ~~59~~ / NAME * NAME

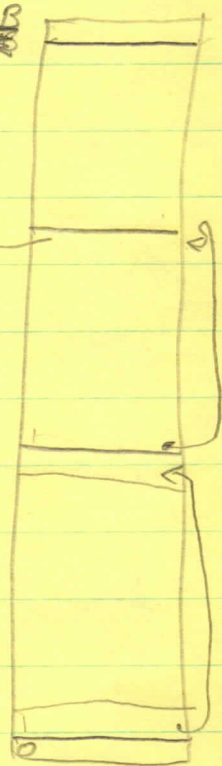
LOC BUF

~~BSS~~ ~~SET~~ ~~0~~

ENOM

~~QRBL~~
QRBL

B
EECSS



MACRO ENDSUB, NAME, BUF

~~QRBL~~ BSSO

~~SET~~ ~~USE~~ * ~~QRBL~~ VFD 60/ * - QRBL

00 - alternately

00

11

ENOM

put new name where name was 0 :

MACRO ALTENT, NAME, BUF, ~~NEWNAM~~ ~~SUCHNAM~~

NAME

SAO

SXO

RECS

EQ NEWNAM - BUF

USE 'CLOCK'

BSS 0

VFD 2/3, 58 / NAME

USE *

ENOM

(X0 = first available ECS)

~~SX2~~
SA1 = X0 + ECS SB3-4

~~SAT~~
~~TX1~~ X1-X2
SB4 = XB ECSSUB

SCAN ~~NO WORK~~ SAI PL A1-1, WORKTODD SB3 A1
EQ B3, B4, ~~WORKTODD~~

SA1 X1
EQ ~~SCAN~~ ~~NO WORK~~

LX1 1
NG X1, ALTENT

WORKTODD SA0 SB2 X1+1 . addr of code to move

SB2 A0

SB3 A1 . beyond end

SB2 B3-B2 . count

WECS B2 . move code

~~SX0~~ X0+B2 . advance ECS addr

~~SX1~~ A0

SA1 A0 . get main entry PTR

~~SX2~~ A0

SA2 X1 . pickup SA0, SX0 word

BX6 X2+X0 . stick in ECS addr

SA6 ~~TX1~~ .

SX7 B2 . word count

AX7 30 . stick in in an KECS dummy

SA7 dummy

BX7 XB+XB

~~SA1 A1-1~~

ALTENT

SA7 A6+1
JP SCAN