

TYMSHARE REFERENCE MANUAL

BRS

OCTOBER 1971

TYMSHARE, INC.
525 UNIVERSITY AVENUE, SUITE 220
PALO ALTO, CALIFORNIA 94301

CONTENTS

DATE: 71/10/01

BRS REFERENCE MANUAL OCTOBER 1, 1971

WRITTEN BY ANN HARDY
 NORMAN HARDY
 MARK KAHR
 DAVID GARDNER
 VERNE VAN VLEAR

TABLE OF CONTENTS

SECTION	SUBJECT
0.0	SYSPOP INDEX
0.1	INDEX BY TYPE
0.2	SEQUENTIAL INDEX
1.0	SEQUENTIAL BRS INDEX
2.0	BRS AND SYSPOP INDEX BY TYPE
3.0	BRS'S IN NUMERICAL SEQUENCE
4.0	SYSPOP'S IN NUMERICAL SEQUENCE
5.0	GLOSSARY
6.0	APPENDIX A BRS 7 TABLES
7.0	APPENDIX B RANDOM FILE SUMMARY
8.0	APPENDIX C OP CODE FLAGS FOR TAPE BRS'S
9.0	APPENDIX D EXECUTIVE ERROR CODES
10.0	APPENDIX E FLOATING ARITHMETIC SUMMARY
11.0	APPENDIX F MACHINE LANGUAGE SUBROUTINES
12.0	APPENDIX G ALL ABOUT PMTS AND SMTS
13.0	APPENDIX H TYMNET INFORMATION
14.0	APPENDIX I FORKING INFORMATION
15.0	APPENDIX J INDEX AND REFERENCES

SECTION 0.0 SYSDOP INDEX

SECTION 0.1 INDEX OF SYSDOPS BY TYPE

FILE INPUT/OUTPUT

CIT(134)	CHARACTER INPUT AND TEST
CIO(161)	CHARACTER INPUT/OUTPUT
WIO(160)	WORD INPUT/OUTPUT
BIO(176)	BLOCK INPUT/OUTPUT

RANDOM FILE OPERATORS

RSP(125)	READ SIZE PARAMETERS
SSP(126)	SET SIZE PARAMETERS
RCP(131)	READ CURSOR POSITION
SCP(132)	SET CURSOR POSITION
PCE(133)	POSITION CURSOR AND ERASE

TELETYPE INPUT/OUTPUT

TCI(174)	TELETYPE CHARACTER INPUT
TCO(175)	TELETYPE CHARACTER OUTPUT

CHARACTER STRING MANIPULATION

LDP(166)	LOAD STRING POINTER
STP(167)	STORE STRING POINTER
ISC(140)	INTERNAL TO STRING CONVERSION
SIC(141)	STRING TO INTERNAL CONVERSION
WCD(135)	WRITE CHARACTER AND DECREMENT
GCD(137)	GET CHARACTER AND DECREMENT
WCI(157)	WRITE CHARACTER AND INCREMENT
GCI(165)	GET CHARACTER AND INCREMENT
SKSG(162)	SKIP IF STRING GREATER
SKSE(163)	SKIP IF STRING EQUAL
WCH(164)	WRITE CHARACTER TO MEMORY BY TABLE

FLOATING POINT OPERATIONS

FAD(156)	FLOATING ADD
FSB(155)	FLOATING SUBTRACT
FMP(154)	FLOATING MULTIPLY
FDV(153)	FLOATING DIVIDE
FFMP(172)	FLOATING MULTIPLY
FFAD(152)	FLOATING ADD
FFDI(147)	FLOATING DIVIDE INVERTED
FFDV(146)	FLOATING DIVIDE
FFSI(143)	FLOATING SUBTRACT INVERTED
FFSB(142)	FLOATING SUBTRACT
STFM(130)	STORE FLOATING ACCUMULATOR
LDFM(127)	LOAD FLOATING ACCUMULATOR
FFSID(124)	FORTRAN II FLOATING SUBTRACT INVERTED
FFSBD(123)	FORTRAN II FLOATING SUBTRACT

FFDVD(122)	FORTTRAN II FLOATING DIVIDE
FFMPD(121)	FORTTRAN II FLOATING MULTIPLY
FFADD(120)	FORTTRAN II FLOATING ADD
STFMD(117)	FORTTRAN II STORE FLOATING ACCUMULATOR
LDFMD(116)	FORTTRAN II LOAD FLOATING ACCUMULATOR
FFDID(115)	FORTTRAN II FLOATING DIVIDE INVERTED

MISCELLANEOUS

SNE(136)	SKIP IF NOT EQUAL
SKP(144)	SKIP IF POSITIVE
SKL(145)	SKIP IF LESS THAN
SBRM(170)	SYSTEM BRM FOR INDIRECT LINKAGE

SECTION 0.2 SEQUENTIAL INDEX OF SYSTEM OPERATORS - SYSPOPS

115	FFDID	FORTTRAN II FLOATING DIVIDE INVERTED
116	LDFMD	FORTTRAN II LOAD FLOATING ACCUMULATOR
117	STFMD	FORTTRAN II STORE FLOATING ACCUMULATOR
120	FFADD	FORTTRAN II FLOATING ADD
121	FFMPD	FORTTRAN II FLOATING MULTIPLY
122	FFDVD	FORTTRAN II FLOATING DIVIDE
123	FFSBD	FORTTRAN II FLOATING SUBTRACT
124	FFSID	FORTTRAN II FLOATING SUBTRACT INVERTED
125	RSP	READ FILE SIZE PARAMETERS
126	SSP	SET FILE SIZE PARAMETERS
127	LDFM	LOAD FLOATING ACCUMULATOR
130	STFM	STORE FLOATING ACCUMULATOR
131	RCP	READ CURSOR POSITION
132	SCP	SET CURSOR POSITION
133	PCE	POSITION CURSOR AND ERASE
134	CIT	CHARACTER INPUT AND TEST
135	WCD	WRITE CHARACTER AND DECREMENT
137	GCD	GET CHARACTER AND DECREMENT
140	ISC	INTERNAL TO STRING CONVERSION
141	SIC	STRING TO INTERNAL CONVERSION
142	FFSB	FLOATING SUBTRACT
143	FFSI	FLOATING SUBTRACT INVERTED
146	FFDV	FLOATING DIVIDE
147	FFDI	FLOATING DIVIDE INVERTED
152	FFAD	FLOATING ADD
153	FDV	FLOATING POINT DIVIDE
154	FMP	FLOATING POINT MULTIPLY
155	FSB	FLOATING POINT SUBTRACT
156	FAD	FLOATING POINT ADD
157	WCI	WRITE CHARACTER AND INCREMENT
160	WIO	WORD INPUT/OUTPUT
161	CIO	CHARACTER INPUT/OUTPUT
162	SKSC	SKIP IF STRING GREATER
163	SKSE	SKIP IF STRING EQUAL
164	WCH	WRITE CHARACTER TO MEMORY BY TABLE
165	GCI	GET CHARACTER AND INCREMENT
166	LDP	LOAD STRING POINTER

167 STP	STORE STRING POINTER
170 SBRM	SYSTEM BRM FOR INDIRECT LINKAGE
172 FFMP	FLOATING MULTIPLY
173 BRS	BRANCH TO SYSTEM
174 TCI	TELETYPE CHARACTER INPUT
175 TCO	TELETYPE CHARACTER OUTPUT
176 BIO	BLOCK INPUT/OUTPUT

SECTION 1.0 SEQUENTIAL BRS INDEX

NOTE: BRS'S MARKED WITH AN ASTERISK ARE EXECUTIVE BRS'S. ALL OTHERS ARE MONITOR BRS'S. BRS'S WITH A + INDICATES THE BRS IS RESTRICTED TO USERS WITH STATUS.

BRS. PAGE STATUS DESCRIPTION

1	15	+	OPEN FILE
2	15	+	CLOSE FILE
3	15	-	MAKE INDIRECT POINTER
4	17	-	KILL PAGE FROM ADDRESS
5	17	-	RETURN STATUS OF CALLING FORK
6*	17	-	SET FILE PARAMETERS (DECLARE FILE)
7	18	+	READ MONITOR TABLES
8	19	+	CLOSE ALL FILES
9	19	-	START FORK
10	20	-	PROGRAMMED PANIC TERMINATES A FORK
11	20	-	CLEAR INPUT BUFFER
12	21	-	SET ECHO TABLE
13	21	-	SKIP IF CHARACTERS IN INPUT BUFFER
14	22	-	DISMISS UNTIL OUTPUT BUFFER EMPTY
15*	22	-	LOOKUP FILE IN FILE DIRECTORY
16*	22	-	OPEN FILE FROM FILE DIRECTORY POINTER ADDRESS
17	22	-	CLOSE ALL FILES
18*	23	-	LOOKUP FILE FOR OUTPUT
19*	23	-	OPEN FILE FOR OUTPUT FROM FILE DIRECTORY POINTER ADDRESS
20	23	-	CLOSE A FILE
21	24	-	FLOATING POINT NEGATE
22	24	+	PREVENT FORK FROM BEING TERMINATED
23	24	+	ALLOW FORK TO BE TERMINATED
25	24	+	GRAB BIT FROM MAP
26	25	-	SKIP IF PANIC OR OFF INTERRUPT PENDING
27	25	+	START STATISTICS
28	26	+	STOP STATISTICS
29	26	-	CLEAR OUTPUT BUFFER
30	26	+	GIVE BIT BACK TO BIT MAP
31	26	-	WAIT FOR SPECIFIED FORK TO TERMINATE
32	27	+	CLEAN UP -2 OR -3
33	27	-	READ STRING
34	27	-	WRITE MESSAGE
35	28	-	WRITE STRING
36	28	-	WRITE NUMBER
37*	28	-	RENAME A FILE TO SAME LENGTH
38	29	-	READ NUMBER
39	29	-	READ CONTROL PARAMETER WORD
40	30	-	READS ECHO TABLE
41	30	+	READ DISC ADDRESS OF CURRENT DATA BLOCK
42	30	-	READ CLOCK
43	31	-	READ PSEUDO-RELABELING
44	31	-	SET PSEUDO-RELABELING
45	31	-	DISMISS FOR 4 SECONDS
46	31	+	SET NON-TERMINABILITY
47	32	+	CLEAR NON-TERMINABILITY

48*	32	-	LOOKUP FILE IN FILE DIRECTORY USING STRING POINTERS
49	32	-	READ INTERRUPTS ARMED
50	32	-	CONVERSION FROM FLOATING POINT TO FIXED POINT
51	33	-	CONVERSION FROM FIXED POINT TO FLOATING POINT
52*	33	-	FORMATTED INPUT
53*	33	-	FORMATTED OUTPUT
54	34	+	GRAB BIT FROM BIT MAP
55	34	+	DISMISS UNTIL DISC WORK DONE
56	34	+	MAKE PMT POINTER INDIRECT FOR RECOVER
57	34	+	MAKE EXPANDED SMT BYTE ACCESSABLE TO USERS
58*	35	-	RENAME A FILE
59*	35	-	CHANGE EFFECTIVE USER NAME FOR PUBLIC FILES
60*	36	-	LOOKUP FILE NAME IN FILE DIRECTORY FOR OUTPUT
62*	36	-	OPEN FILE FOR INPUT
63*	36	-	OPEN FILE FOR OUTPUT
64*	37	-	OPEN FILE FOR INPUT USING STRING POINTERS
65*	37	-	OPEN FILE FOR OUTPUT USING STRING POINTERS
66	38	+	ERASE DISC FILE
67*	38	-	READ TS PAGE
68*	39	-	READS FILE NAME FROM FILE DIRECTORY INTO MEMORY
69*	40	-	DELETE A FILE
70	40	-	COUNT FREE PAGES
71	41	-	SKIP IF SYSTEM OR EXEC STATUS SET
72	41	+	EXEC DISMISS
73	41	-	READ ERCODE
77	42	-	CHANGE TOUT FILE
78	42	-	ARM/DISARM SOFTWARE INTERRUPT
80	43	-	SET PAGE READ ONLY
81	44	-	DISMISS FOR A SPECIFIED PERIOD OF TIME
82	44	-	SKIP IF FLOATING ACCUMULATOR NEGATIVE
83	44	-	SKIP IF FLOATING ACCUMULATOR ZERO
84	44	-	SKIP IF FLOATING ACCUMULATOR NON-ZERO
85	45	-	SET 8-LEVEL OUTPUT MODE
86	45	-	RESET 8-LEVEL OUTPUT MODE
87	45	+	CONVERT PORT NUMBER TO JOB NUMBER
88	46	-	READ CPU TIME
89	46	-	READ RESOURCE METERING
90	46	-	DECLARE FORK FOR PANIC
91*	47	-	STORE DATE AND TIME INTO A STRING
92	47	-	SET VALUE OF FILE SIZE QUANTUM FOR ALL FILES TO BE OPENED
93	47	+	RESET RESOURCE METERING
94	48	+	CLEAR FORK STRUCTURE
95*	48	+	ACQUIRE AND RELEASE OVER FLOW GROUPS
96*	48	-	READ FILE ATTRIBUTES AND FILE DIRECTORY DATA
97	49	+	RESET SUBSYSTEM COUNTER
98	50	+	INCREMENT SUBSYSTEM COUNTER
99	50	+	READS SUBSYSTEM COUNTER
100	50	+	ASSIGNS A DEVICE TO A USER
101	50	+	UNASSIGN DEVICE
102	50	-	READ MAG TAPE
103	51	-	WRITE MAG TAPE
104	51	-	REPORTS WHO HAS DEVICE
105	51	-	MAG TAPE CONTROLS
106	52	-	PRINT ON PRINTER

107	52	-	SET MAG TAPE PARITY
108	52	-	TEST MAG TAPE DENSITY
109	52	+	DISMISS
110	53	-	TEST MAG TAPE READY
111	53	+	RETURN FROM CLASS 3 BRS
112	53	+	REMOVE A JOB FROM THE SYSTEM
113	53	-	DISABLE ESCAPES
115	53	+	TERMINATE CLASS 3 BRS FORK WITH PANIC
116	54	-	READ PROGRAM RELABELING
117	54	-	SET PROGRAM RELABELING
118	54	-	LOAD FLOATING ACCUMULATOR FROM A & B
119	54	-	STORE FLOATING ACCUMULATOR INTO A & B
130	54	+	TEST A BREAKPOINT SWITCH
131	55	+	TO CRASH THE SYSTEM
132	55	-	READ TERMINAL CHARACTERISTICS
134	57	-	READ IN CHARACTERS IN "PAPER TAPE MODE"
135	58	-	INTERRUPTS A FORK AFTER A SPECIFIED PERIOD OF TIME
136	58	+	SETS SYSTEM EXEC SWITCHES IN SYMS
137	58	-	LIST PORT NUMBERS IN MEMORY
138	59	+	SUPERVISOR REQUEST
139	59	-	ZAP AN AUXILARY CIRCUIT
140	60	-	READS THE DATE INCLUDING THE YEAR INTO THE USERS MEMORY
141*	60	+	GETS EXEC SUBROUTINES
142	60	-	ALLOWS USER EXECUTION OF EXEC COMMAND
143	61	+	SKIP IF BIT MAP SET
144	61	+	GETS A BUFFER
145	61	+	RETURNS A BUFFER
146	61	+	READS NUMBER OF QUEUE ENTRIES
147	62	-	CLOSES ALL EXCEPT COMMANDS-FROM FILE
148	62	-	NEGATE FLOATING ACCUMULATOR
149	62	-	FIX FLOATING ACCUMULATOR AND STORE IN A & B
150	62	-	FLOAT A & B AND PLACE IN THE FLOATING ACCUMULATOR
151*	62	-	CHANGE COMMANDS-FROM FILE
153	63	+	SET CELL 211
154	63	+	RESET CELL 211
155	63	-	TEST PORT FOR CHARACTERS
156	63	-	ZAP ALL AUXILARY CIRCUITS
157	64	+	COPY LEVEL 3 INDEX BLOCK TO MEMORY
158	64	-	CHARGE ROYALTY UNITS
159	64	+	ANSWER LINES
160	64	-	CHANGE TERMINAL CHARACTERISTICS
161	65	-	IGNORE PANICS
162	65	-	ENABLE PANICS
163	65	+	COPY NET TO USER
164	65	-	DISMISS UNTIL OUTPUT BUFFER EMPTY AND/OR ACTIVATE USER AFTE
166	66	+	HANG PORT 0
167	66	-	READ AND RESET DISK ERROR FLAG
168	67	-	DISABLE SHIFT CONTROL 0
169	67	-	RE-ARM SHIFT CONTROL 0
170	67	-	SET LIMIT FOR PREMIUM CHARGE PROGRAM CHARGES
171	68	-	IDENTIFY PREMIUM CHARGE PROGRAM
172	68	-	SEND CONTROL INFORMATION TO REMOTE

SECTION 2.0 BRS AND SYSPOP INDEX BY TYPE

FILE INPUT/OUTPUT OPERATIONS

OPEN FILES

62 OPEN FILE FOR INPUT
63 OPEN FILE FOR OUTPUT
64 OPEN FILE FOR INPUT WITH STRING POINTERS
65 OPEN FILE FOR OUTPUT WITH STRING POINTERS
151 CHANGE A "COMMANDS FROM" FILE

CLOSE FILES

20 CLOSE A FILE
17 CLOSE ALL FILES
147 CLOSE ALL EXCEPT COMMANDS FROM FILE

FILE INPUT/OUTPUT OPERATIONS

CIO (161) CHARACTER INPUT/OUTPUT
WIO (160) WORK INPUT/OUTPUT
BIO (176) BLOCK INPUT/OUTPUT
CIT (134) CHARACTER INPUT AND TEST

FILE NAME MANIPULATION

37 RENAME FILE
58 RENAME FILE
59 CHANGE CHARACTERS IF FILE NAME
68 READ FILE NAME FROM FILE DIRECTORY
48 LOOK UP FILE NAME
92 SET DEFAULT FILE QUOTA

FORK MANAGEMENT

5 READ CALLING FORK STATUS
9 START FORK
10 TERMINATE FORK
90 DECLARE FORK FOR TERMINATION ON "PANIC"

DEVICE INPUT/OUTPUT

102 READ TAPE
103 WRITE TAPE
104 WHO HAS DEVICE
105 CONTROLS FOR TAPE
106 PRINT
107 SET TAPE PARITY
108 TEST TAPE DENSITY
110 TEST TAPE READY

RANDOM FILE OPERATIONS

RSP (125) READ FILE SIZE PARAMETERS

SSP (126)	SET FILE SIZE PARAMETERS
RCP (131)	READ CURSOR POSITION
SCP (132)	SET CURSOR POSITION
PCE (133)	POSITION CURSOR AND ERASE

MISCELLANEOUS FILE OPERATIONS

69	DELETE A FILE
96	REPORT FILE DIRECTORY DATA
6	SET FILE PARAMETERS

TELETYPE INPUT/OUTPUT OPERATIONS

TCI	TELETYPE CHARACTER INPUT
TCO	TELETYPE CHARACTER OUTPUT
13	TEST INPUT BUFFER FOR EMPTY
11	CLEAR INPUT BUFFER
14	DISMISS UNTIL OUTPUT BUFFER EMPTY
29	CLEAR OUTPUT BUFFER
12	DECLARE ECHO TABLE OR SET 8-LEVEL INPUT
40	READ ECHO TABLE
85	SET 8-LEVEL OUTPUT
86	CLEAR 8-LEVEL OUTPUT
134	SET LINE FEED OR CARRIAGE RETURN IGNORE
24	CHANGE TERMINAL CHARACTERISTICS
132	READ TYMNET TERMINAL CHARACTERISTICS
155	DISMISS IF TCO WOULD CAUSE A DISMISS OR CHARACTERS IN INPUT BUFFER
160	SET TYMNET TERMINAL CHARACTERISTICS
164	DISMISS UNTIL OUTPUT BUFFER EMPTY OR UNTIL A SPECIFIED PERIOD OF TIME
172	SEND CONTROL INFORMATION TO TYMNET

"PANIC" AND SHIFT CONTROL O OPERATIONS

113	TURN PANIC OFF
114	TURN PANIC ON
161	DISARM "PANICS" AND SHIFT CONTROL O
162	REARM "PANICS" AND SHIFT CONTROL O
168	DISARM SHIFT CONTROL O
169	REARM SHIFT CONTROL O

AUXILARY PORT OPERATIONS

137	LIST PORTS IN USE
139	ZAP AUXILARY PORT
156	ZAP ALL AUXILARY PORTS

MEMORY OPERATIONS

3	MAKE PMT POINTER INDIRECT
4	RELEASE A PAGE OF MEMORY
121	RELEASE A PAGE OF MEMORY
43	READ PSEUDO-RELABELING

44	SET PSEUDO-RELABELING
116	READ PROGRAM RELABELING
117	SET PROGRAM RELABELING
80	MAKE PAGE READ ONLY
70	COUNT FREE USER PAGES

STRING PROCESSING OPERATIONS

STRING INPUT/OUTPUT

33	READ STRING
34	OUTPUT MESSAGE
35	OUTPUT STRING

STRING MANIPULATION

STP	STORE STRING POINTER
LDP	LOAD STRING POINTER
SKSE	SKIP IF STRING EQUAL
SKSG	SKIP IF STRING GREATER

CHARACTER MANIPULATION

GCI	GET CHARACTER FROM BEGINNING OF STRING AND INCREMENT BEGINNING POINTER.
WCI	PUT CHARACTER ON END OF STRING AND INCREMENT END POINTER.
GCD	GET CHARACTER FROM END OF STRING AND DECREMENT END POINTER
WCD	PUT CHARACTER ON BEGINNING OF STRING AND DECREMENT BEGINNING POINTER

NUMBER OPERATION

NUMBER INPUT/OUTPUT

36	OUTPUT NUMBER TO SPECIFIED RADIX
38	INPUT NUMBER TO SPECIFIED RADIX
52	FORMATTED FLOATING POINT INPUT
53	FORMATTED FLOATING POINT OUTPUT
SIC	STRING TO INTERNAL CONVERSION
ISC	INTERNAL TO STRING CONVERSION

ARITHMETIC OPERATIONS

21	FLOATING POINT NEGATE
50	CONVERSION FROM FLOATING POINT TO FIXED POINT
51	CONVERSION FROM FIXED POINT TO FLOATING POINT
SNE	SKIP IF NOT EQUAL
SKP	SKIP IF POSITIVE
SKL	SKIP IF LESS THAN
FAD	FLOATING POINT ADDITION
FSB	FLOATING POINT SUBTRACT

FMP FLOATING POINT MULTIPLICATION
 FDV FLOATING POINT DIVISION

HARDWARE FLOATING ARITHMETIC

FFMP FLOATING MULTIPLY
 FFAD FLOATING ADD
 FFDI FLOATING DIVIDE INVERTED
 FFDV FLOATING DIVIDE
 FFSI FLOATING SUBTRACT INVERTED
 FFSB FLOATING SUBTRACT
 STFM STORE FLOATING ACCUMULATOR
 LDFM LOAD FLOATING ACCUMULATOR

FFSID FLOATING SUBTRACT INVERTED
 FFSBD FLOATING SUBTRACT
 FFDVD FLOATING DIVIDE
 FFMPD FLOATING MULTIPLY
 FFADD FLOATING ADD
 STFMD STORE FLOATING ACCUMULATOR
 LDFMD LOAD FLOATING ACCUMULATOR
 FFDID FLOATING DIVIDE INVERTED

82 SKIP IF FLOATING ACCUMULATOR NEGATIVE
 83 SKIP IF FLOATING ACCUMULATOR ZERO
 84 SKIP IF FLOATING ACCUMULATOR NON-ZERO
 118 LOAD FLOATING ACCUMULATOR
 119 STORE FLOATING ACCUMULATOR
 148 NEGATE FLOATING ACCUMULATOR
 149 FIX FLOATING ACCUMULATOR TO A
 150 FLOAT A TO FLOATING ACCUMULATOR

PROPREITARY PROGRAMS

158 CHARGE ROYALTY UNITS
 170 SET LIMIT FOR PREMIUM CHRAGE PROGRAM
 171 IDENTIFY PREMIUM CHARGE PROGRAM

MISCELLANEOUS

39 READ CPARW AND AUNN
 42 READ REAL TIME CLOCK
 45 DISMISS FOR 3 SECONDS
 67 READ USERS TS PAGE
 71 SKIP IF SYSTEM STATUS SET
 73 READ ERCODE
 78 ARM/DISARM SOFTWARE INTERRUPTS
 81 DISMISS FOR SPECIFIED AMOUNT OF TIME
 88 READ EXECUTION TIME
 89 READ RESOURCE METERING
 91 READ DATE AND TIME INTO STRING
 140 READ TIME AND DATE INTO A STRING INCLUDING YEAR
 142 EXECUTE AN EXEC COMMAND
 SBPM INDIRECT SUBROUTINE LINKAGE

SYSTEM AND SUBSYSTEM RESTRICTED BRS'S

PANIC CONTROL

46 TURN "PANICS" OFF AND DISABLE OFF INTERRUPTS
47 TURN "PANICS" ON AND ENABLE OFF INTERRUPTS
26 SKIP IS PANIC WAITING

FORK MANAGEMENT

22 PREVENT TERMINATION
23 ALLOW TERMINATION
72 EXEC DISMISS
94 CLEAR FORKING STRUCTURE
109 DISMISS
22 PREVENT TERMINATION
111 TERMINATE CLASS 3 BRS FORK
115 TERMINATE CLASS 3 BRS FORK WITH RUBOUT

DEVICE INPUT/OUTPUT

100 ASSIGN DEVICE
101 DEASSIGN DEVICE

INPUT/OUTPUT

1 OPEN A FILE
2 CLOSE A FILE
8 CLOSE ALL FILES
41 RETURN DISC ADDRESS OF CURRENT DATA BLOCK
66 DELETE DISC FILE
122 READ DISC WITHOUT DISMISS
123 WRITE DISC WITHOUT DISMISS
124 READ DISC
125 WRITE DISC
130 TEST A BREAK POINT SWITCH
157 COPY LEVEL 3 INDEX BLOCK POINTER TO MEMORY
167 TEST FOR DISC ERROR

TELETYPE CONTROL

32 CLEAR -2 OR -3 STATE
87 GET JOB NUMBER FROM PORT NUMBER
126 TEST FOR CARRIER PRESENCE
129 TURN CHANNEL ON OR OFF
159 SEND UP/SHUT
163 COPY NET TO USER
166 HANG PORT 0

MEMORY CONTROL

56 RECOVER PMT BYTE

120 ASSIGN PMT ENTRY
57 MAKE INTO USER PAGE

EXEC COMMAND BRS'S

127 LOOK AT MEMORY

BIT MAP CONTROL

25 GRAB BIT MAP BIT
30 GIVE BIT TO BIT MAP
54 GRAB BIT FROM MAP AND RETURN DISC ADDRESS
128 SET BIT MAP
143 SKIP IF BIT MAP SET

STATISTICS

27 START STATISTICS
28 STOP STATISTICS

FILE DIRECTORY CONTROL

95 ACQUIRE AND RELEASE OVERFLOW GROUPS

SUPERVISOR OPERATIONS

138 SEND SUPERVISOR REQUEST
153 SET CELL 211B
153 RESET CELL 211B

MISCELLANEOUS

7 READ TABLES
131 CRASH SYSTEM
136 SET EXEC SWITCHES
141 GET EXEC SUBROUTINES
144 GET A BUFFER
145 RETURN A BUFFER
97 RESET SUBSYSTEM COUNTER
98 INCREMENT SUBSYSTEM COUNTER
99 READ SUBSYSTEM COUNTER
55 DISMISS IF JOB USING DISC
61 CHANGE TELETYPE WORKING SET
93 RESET RESOURCE METERING
146 READ QUEUE COUNTERS

SECTION 3.0 BRS'S IN NUMERICAL SEQUENCE

BRS 1

Date: 71/09/03
 Name: [open] [file]
 Status: Exec
 Input: A= [File Directory Pointer Address]
 B= Privilege bits for random files
 Output:
 Skip Return: A= File number
 (success) B= Same as input
 X= Index block pointer

 Return: A= Disc address
 (failure) B= Same as input
 X= Error number

Description: Opens a file with the index block pointer in A. If the file cannot be opened for any one of the reasons listed below, a non-skip return is taken and the error number returned in the X register. If a file is opened for output any other attempts to open that file will result in "FILE BUSY".

Privilege bits in X:
 Bit 18 (40B) = User can do SCP syspop
 19 (20B) = 0
 20 (10B) = 0
 21 (4B) = User may do PCE syspop
 22 (2E) = Set to write mode
 23 (1B) = Set to read mode

Error numbers in X:
 1 = File busy
 2 = All buffers busy
 3 = Bitmap not set
 4 = [Fatal exception] for disc error
 5 = Disc full or out of file numbers

Instruction traps: insufficient status

BRS 2

Date: 71/09/03
 Name: [close] [file]
 Status: Exec
 Input: A= File number returned by BRS 1
 Output: A= Same as input
 B= Same as input
 X= Same as input
 Description: Closes file opened by BRS 1.
 Instruction traps: Insufficient status, bad file number.

BRS 3

Date: 71/09/03
 Name: Make [indirect pointer]
 Status: User, subsystem, system or Exec (depending on operation)
 Input: A= [relabeling] byte or [subsystem byte number] (expanded SMT)
 X= Port number (only used if 43B A 100B)
 Output: A= Relabeling byte
 B= Same as input
 X= Same as input
 Description: This BRS has several functions. They are:

1. To set up a pointer to an SMT (byte 43B)
2. To set up a pointer to a PMT in channel number in X (44B byte 100B)
3. To set up a pointer to an [expanded SMT] (100B byte 215B)

If the sign bit is on, the byte will be made read only.
 The BRS 3 will return a 6 bit relabeling byte number in the A register, this places the byte in the user's PMT's but not (!!!) in his relabeling. In order to gain access to the byte, the program must do a BRS 44 on the byte number returned in A. If the sign bit is 0, then Exec status is required to do a BRS 3. If the sign bit is 1, then the byte 100B system is required, otherwise subsystem status is tested for. If the original owner of that pages kills that page, the pointer to that page will be removed from the current forks relabeling. If the byte is 150B, 151B or bytes 165B - 167B, then user status is allowed.

Subsystem Relabeling Byte Table

Subsystem	Relabeling or SMT byte(s) numbers											
-----	-----											
BFortran(C)	201	202	203	204	205	206	207	210	211	212	213	214
BFortran(R)	31	32	33	34	35							
Cal	112	113	114	115	116							
Common	132	133	134									
DDT	41*	42*										
Editor	26	27	30									
f2C	163	164										
F2OS	165*	166*	167*									
FOS	150*	151*	152*									
FTC	130	131										
MUSE	147	153	154	155	170	175	176	177				
NBASIC	117	120	121	122	123	124	125	156	157	160	161	
SBASIC	101	102	103	104	105	106	107					
Sfortran	135	136	137	140	141	142	143	144				
XDDT	36*	37*										

An asterisk indicates that the byte is available to user status programs.

BRS 3 Status Table

Byte no. -----	Read Only -----	Read/Write -----
Byte 44B	System	Exec
43B Byte 77B	System	Exec
Byte 77B	Subsystem	Exec

Instruction Traps: Insufficient status, bad byte number.

BRS 4

Date: 71/09/03

Name: [kill] page from address

Status: User

Input: A= Address in page wanted to kill

Output: A= Same as input

B= Same as input

X= Same as input

Description: Releases page that the address in A points to.

If any fork has this page in it's relabeling, it is cleared from it.

Instruction traps: Attempt to release SMT. Insufficient status when trying to release the TS page.

BRS 5

Date: 71/09/03

Name: Return status of calling [fork]

Status: User

Input: None

Output: A= Status of calling fork

B= Same as input

X= Same as input

Description: Reads status of calling fork and returns number

in A: 0= No status

1= Subsystem

3= System

7= Exec

Instruction traps: None.

BRS 6

Date: 71/09/03

Name: Set [file parameters] (declare file)

Status: User or system

Input: A= [file directory pointer address] (from BRS 15 or 48)

X= File parameter bits (see below)

Output: A= destroyed

Skip Return: B= Destroyed

(success) X= Destroyed

Return: A= Destroyed

(failure) B= Destroyed

X= Destroyed

Description: Sets the file directory bits as the declare command

would. Bits 0-11 of X contain a mask of the bits that are to be changed. Bits 12-23 of X contain the new values the program wishes them to be. If X= -1, then the premium charge is turned on. This requires system status. The following is a list of valid bits for the X register:

Bits 12-23 of X = XWV USR QNM LKJ

X= Public remote (4B3). Allows program to access files from the directory from which the program was called.

W= Public write (2B3). Allows other users out the account to write on the file.

V= Reserved. (1B3) not currently used

U= Not private read (4B2). Prevents user from reading file in his own directory.

S= Not private write (2B2). Makes the file "read only".

R= Append only. (1B2). Any output to this file is append to the file and the file contents not cleared.

Q= Account public. (40B) available to the entire account.

N= Proprietary. (20B). The memory is cleared after the program terminates. Also not private readable.

M= Subsystem status. (10B). Starts fork up with subsystem status.

L= System status. (4B). Starts fork up with system status.

K= Exec status. (2B). Starts fork up with Exec status.

J= Init, flag. (1B). Runs program when user logs in.

Instruction traps: None.

BRS 7

Date: 71/09/03

Name: Read [monitor tables]

Status: Dependent on table (see below)

Input: A= Address in program memory where table is to be written.

X= Table number

Output: A= Address of first word beyond table in user's buffer

B= Same as input

X= Same as input

Description: Reads a monitor table and places it in the user's memory. If the sign bit of A is on, the BRS 7 will copy the user's buffer into the monitor. This always requires EXEC status.

Table no.	Read Status	Write Status	Table Size	Description
0	System	Exec	45B	Miscellaneous Monitor Counters
1	System	Exec	20D	RAD Error List
2	System	Exec	10D	Disc Error List
3	System	Exec	32D	Activation Swap Counters
4	System	Exec	6D	Multiplexer Communication
5	System	Exec	32D	Real Memory Table
6	System	Exec	40D	Global File Flags
7	System	Exec	42d	Job to Teletype Conversion

8	System	Exec	1D	ALARM.
9	System	Exec	40D	Global Index Block Pointers
10	System	Exec	1D	Highest Disk Address
11	User	Exec	1D	Machine Number in Ascii
12	User	Exec	1D	Eqpt. Equipment
13	User	Exec	2D	Accounting Cursor Positions
14	System	Exec	NPORT	WERIS Table
15	System	Exec	NJOB	PMTTP Table
16	System	Exec	NJOB	CPARW Table
17	System	Exec	NJOB	AUNN Table
18	System	Exec	NJOB	FGLIST Table
18	System	Exec	19D	Monitor Table
19	System	Exec	32D	RAD Bit Map
20	System	Exec	NJOB	ETTB Table
21	System	Exec	NJOB	PTCKS Table
22	System	Exec	NJOB	CCT Table
23	System	Exec	NPORT	TTYASC Table
24	System	Exec	20D	Queue Tables
25	System	Exec	2D	Montab Table

For further information, see Appendix A.

Instruction traps: Illegal table number, insufficient status for read or write.

BRS 8

Date: 71/09/03

Name: [close] all [file]s

Status: Exec

Input: None

Output: A= Same as input

B= Same as input

X= Same as input

Description: Closes all open files, but doesn't reset CIN or TOUT.

Used by the Exec before logging the user out.

Instruction traps: Insufficient status.

BRS 9

Date: 71/09/03

Name: Start [fork]

Status: User

Input: A= Bits 0-7= Fork bits (see below)

3-23= Address of [panic table]

Output: A= Same as input

B= Same as input

X= Same as input

Description: Creates dependent entries in the [PAC] table entries for each user. A fork which calls a BRS 9 is called the [controlling fork]. The fork produced by the BRS 9 is called the [lower fork].

Each controlling fork may be a lower fork for some other controlling fork.

The controlling fork is dismissed until the lower fork terminates.

A user cannot have over 8 forks (this includes the Exec fork and each

fork for an Exec BRS in progress). Only one Exec BRS can be active at

a time. If a fork attempts to start a lower fork with status higher than itself, the BRS 9 will ignore it.

Fork bits:

- Bit 0= Give fork [Exec status] if controlling fork has Exec status
 1= Set fork relabeling from panic table. Otherwise, use the current relabeling of the controlling fork
 2= Propagate [panic assignment] to fork if controlling fork has it. (See BRS 90)
 3= Make fork [fixed memory]. It is not allowed to gain more memory. Any attempts will result in a memory trap.
 4= Make fork [local memory]. New memory is
 5= Give fork [subsystem status] if issuing fork has subsystem status.
 6= Give fork [system status] if issuing fork has system status.

Panic table format:

Word	Description
0	Program counter
1	A register
2	B register
3	X register
4	First relabeling register
5	Second relabeling register
6	Status word. -2 = Dismissed for input/output -1 = Running 0 = Dismissed on [panic] or BRS 10 1 = Dismissed on instruction trap 2 = Dismissed on memory [trap]

Instruction traps: Issuing fork was under local memory bounds, panic table overlaps page boundry, attempting to restart same fork, attempt to create 9th fork.

BRS 10

Date: 71/09/08

Name: Programmed Panic [terminate]s a [fork]

Status: User

Input: None

Output: A= Same as input

B= Same as input

X= Same as input

Description: Terminates a fork in much the same manner as a panic would do. The only difference between a BRS 10 and a panic is that the program counter points to the BRS 10. This BRS is normally used to terminate a running fork when it is finished.

Instruction traps: None.

BRS 11

DATE: 71/09/08

Name: Clear [input buffer]

Input: X= Port number (-1, 0 or 1 for command port)

Output: A= Same as input

B= Same as input

X= Same as input

Description: Clears the input buffer.

Instruction traps: Bad port number in X (takes interrupt 10 if armed).

BRS 12

DATE: 71/09/09

Name: Set [echo table]

Status: User

Input: A= Echo table number or terminating character for 8 level input.

X= Port number (0 or -1 for [command port])

Output: A= Same as input

B= Same as input

X= Same as input

Description: If the sign bit of A is on, then the low order 8 bits of A contain the character to terminate on. In 8 level mode, characters are read from the [terminal] and transmitted unchanged to the user program. If bit 15 (400B) is on, there can be no terminating character.

The program must terminate the 8 level mode by itself, or the user will be hung and must hang up his phone. If bit 4 (2B6) is on, 8 level output mode is set.

Echo tables: (In X):

0= [echo] each character as it was received and break on all characters.

1= Echo each character and break on all letters, digits and spaces.

2= Echo each character and break on control characters (including carriage return and line feed). This is the normal mode set by the Exec.

3= No echo for each character and break on all characters.

Instruction traps: Bad echo table in A.

BRS 13

DATE: 71/09/20

Name: Skip if characters in [input buffer]

Status: User

Input: X= Port number (-1, 0 or 1 for command port)

Output:

Skip return: A= Preserved

(success) B= Preserved

X= Preserved

return: A= Preserved

(failure) B= Preserved

X= Preserved

Description: Skip if there are any characters in the input buffer.

Instruction traps: Bad port number in X (takes interrupt 10 if armed).

BRS 14

Date: 71/08/20
 Name: [Dismiss] until [output buffer] empty
 Status: User
 Description: Dismisses the user until [output backpressure] has ceased.
 Note: This BRS is obsolete. The recommended new BRS is BRS 164.
 Instruction traps: Bad port number (if int. 10 is not armed).

BRS 15

DATE: 71/09/21
 Name: [Lookup] [file] in [file directory]
 Status: User
 Input: A= Command file
 Output:
 Skip return: A= File directory pointer address
 (success) B= Destroyed
 X= File directory pointer address
 return: A= Destroyed
 (failure) B= Destroyed
 X= Exec ercode
 Description: Inputs a file name (see Glossary) from the command file in A and looks it up in the appropriate file directory. If the file does not exist, or the file name is in bad form, the non-skip return is taken.
 Instruction traps: Bad file number in A.

BRS 16

DATE: 71/09/21
 Name: [Open] [file] from file directory pointer address
 Status: User
 Input: A= File directory pointer address
 Output:
 Skip return: A= File number
 (success) B= File type (see Glossary)
 X= File size
 return: A= Destroyed
 (failure) B= Destroyed
 X= Exec ercode (see Appendix D)
 Description: Opens a file looked up by a BRS 15, 48 or 68. The BRS fails if the file directory pointer address in A is not pointing to a proper location in the TS block or if the file cannot be opened. (See BRS 62 for more information on an opening).
 NOTE: The BRS 16 does not (!!!) allow the user to set extra-ordinary file privilege bits. The only ones set are ability to do an SCP and read mode.
 Instruction traps: None

BRS 17

Date: 71/08/20
 Name: [Close] all [files]
 Status: User
 Input: None
 Output: A= Same as Input
 B= Same as Input
 C= Same as Input
 Description: Closes all open files, sets [command file] to teletype and [TOUT] file to teletype.
 Instruction traps: None

BRS 18

DATE: 71/09/21
 Name: [Lookup] [file] for output
 Status: User
 Input: A= Command file
 Output:
 Skip return: A= File directory pointer address
 (success) B= Confirming character
 X= Destroyed
 return: A= Destroyed
 (failure) B= Destroyed
 X= Exec ercode (see Appendix D)
 Description: The BRS reads the file name (see Glossary) from the command file and tests it for validity. If the file name is improperly constructed, the non-skip return will be taken. If the "New File"/"Old File" message is to be typed, bit 1 of A must be on.
 Instruction traps: Bad file number in A.

BRS 19

DATE: 71/09/21
 Name: [Open] [file] for output from file directory pointer address
 Status: User
 Input: A= File directory pointer address (from BRS 18 only!!!)
 X= File type (see Glossary)
 Output:
 Skip return: A= File number
 (success) B= Old file type
 X= Destroyed
 return: A= Destroyed
 (failure) B= Destroyed
 X= Destroyed
 Description: Opens the file input to the BRS 18 for output. The skip output contains the same information as a BRS 63. If the file cannot be opened the non-skip return will be taken.
 NOTE: This BRS cannot set any random file bits. It is opened as a BRS 63 would (e.g., erase the file).
 Instruction traps: Bad file type in X.

BRS 20

Date: 71/08/20

Name: [Close] a [file]
Status: User
Input: A= Open file number
Output: A= Same as Input
B= Same as Input
X= Same as Input

Description: Closes the file with the file number in A. If any disc use is pending on this file, the monitor will dismiss until it is completed. If the file is open for output or update and currently in write mode, the file is erased from the cursor to the end of file.

Instruction traps: Bad file number, file not open

BRS 21

Date: 71/08/20

Name: FNA

Status: User

Input: A= Floating Point number
B= Floating point number

Output: A= Negated point number
B= Negated point number

Description: The double work [floating point] number in A and B is negated and returned to A and B.

Instruction traps: None

BRS 22

Date: 71/08/20

Name: Prevent fork from being terminated

Status: Exec

Input: None

Output: A= Same as input
B= Same as input
X= Same as input

Description: Prevents fork from being terminated on quantum overflow or I/O dismissal.

Instruction traps: Insufficient status

BRS 23

Date: 71/08/20

Name: Allow fork to be terminated

Status: Exec

Input: None

Output: A= Same as input
B= Same as input
X= Same as input

Description: Allows fork to be swapped immediately. The pages are unlocked and released (not killed!)

Instruction traps: Insufficient status

BRS 25

Date: 71/08/20

Name: Grab bit from map

Status: Exec

Input: A= Disk address to grab from disc [bit map]

Output:

Skip Return: A= Same as input

B= Same as input

X= Same as input

Return: A= Same as input

B= Same as input

X= Same as input

Description: Grabs bit from bit map and skips if none of the following conditions exist:

1) Bit map not set

2) Bit already mapped

3) Garbage in A (improper address in A)

This BRS is used by the MAP routine to reserve a disk page on each disc pack on 2314 systems and to reserve the statistics blocks, if any exist.

Instruction traps: Insufficient status.

BRS 26

Date: 71/08/20

Name: Skip if [Panic] or [Off Interrupt] Pending

Status: User

Input: None

Output:

Skip Return: A= Same as input

B= Same as input

X= Same as input

Return: A= Same as input

B= Same as input

X= Same as input

Description: Skips if an interrupt is pending on the [phantom user] queue (for BRS 46 and 47) or if an interrupt is stack while a BRS 113 or 114 is set.

Note: This BRS will not work for BRS 161 or 162.

References: BRS 46, 47, 161, 162

Instruction traps: None

BRS 27

Date: 71/08/20

Name: Start [Statistics]

Status: Exec

Input: None

Output: A= current buffer number

B= Same as input

X= Same as input

Description: The first time it is called, the BRS 27 sets STSW to -1 and returns A= 0 (initial buffer number). If the BRS 27 is called after statistics are turned on, A returns the current buffer number.

Instruction traps: Bit map not set, insufficient status

BRS 28

Date: 71/08/20

Name: Stop [Statistics]

Status: Exec

Input: None

Output: A= Same as input

B= Same as input

X= Same as input

Description: Stops statistics by placing 3777777B in STSW, also writes out last buffer and appends end of statistics record type.

Instruction traps: Insufficient status

BRS 29

Date: 71/08/20

Name: Clear [output buffer]

Status: User

Input: X= -1 or 0 for user's [command port]

Output: A= Same as input

B= Same as input

X= Same as input

Description: Sends a [character gobble] in the downstream direction, clearing characters out.

Instruction traps: Improper port number in X (if interrupt 8 is armed, it will occur instead), port not output port.

BRS 30

Date: 71/08/20

Name: Give bit back to [bit map]

Status: Exec

Input: A= Disc address to release from map

Output:

Skip Return: A= Same as input

(success) B= Same as input

X= Same as input

Return: A= Same as input

(failure) B= Same as input

X= Same as input

Description: Returns the disc address in A to the bit map, if the following conditions do not apply:

- 1) Bit map not set
- 2) Garbage in A improper address in A)
- 3) Bit already mapped

Instruction traps: Insufficient status

BRS 31

Date: 71/08/20

Name: Wait for specified [fork] to [terminate]

Status: User

Input: A= [Panic table] address

Output: A= Same as input

B= Same as input

X= Status of fork (reason for termination)

Description: Dismisses until the specified fork panics in some way

X contains the status word as returned by the BRS g.

Instruction traps: None

BRS 32

DATE: 71/07/22

Name: Clean up -2 or -3

Status: System

Input: X = Port number

Output: A = Same as Input

B = Same as Input

X = Same as Input

Description: Places a "never run" activation condition for the channel in X.

Instruction Traps: Insufficient status

BRS 33

DATE: 71/09/21

Name: Read [string]

Status: User

Input: A= Address of a [string pointer]

B= Terminating character

X= Input [file number]

Output: A= First word of string pointer

B= Second word of string pointer

X= Same as input

Description: Reads a string from the input file until the character in B is reached. When the BRS returns, it modifies the string pointers in memory and places them in the A and B registers.

The terminating character is not appended to the string.

Instruction traps: Illegal file number in X.

BRS 34

Date: 71/08/20

Name: Write [message]

Status: User

Input: A= Address of message in memory

B= -1 to terminate on /, \$= carriage return or the number of characters to print.

X= Output [file number]

Output: A= Last Character in string

B= Destroyed

X= Preserved

Description: If B is equal to -1, then the BRS will print out the message whose address was contained in the A register. In this mode, all dollar signs (\$) are printed on carriage returns and the end of a message is signified by a slash (/). If B is positive, it will be

the number of characters to be printed starting at the location in A.

Example: To print the message "Tymshare BRS manual" followed by a carriage return:

```
PRINT LDA =MSG; LDB =-1; LDX =1 (To terminal); BRS 34; ...
MSG      ASC 'Tymshare BRS manual $/'
```

Note: The message must be in 7 bit ASCII, not 6 bit text.
Instruction Traps: Bad File number in X

BRS 35

Date: 71/08/20
Name: Write [string]
Status: User
Input: A= First word of string pointer
B= Second word of string pointer
X= Output [file number]
Output: A= Destroyed
B= Destroyed
X= Preserved

Description: Writes the string whose string pointers are contained in A and B to the file number in X. If system status is set, control characters will be written as & character (as in EDITOR).
Instruction traps: Illegal file number in X.

BRS 36

Date: 71/08/20
Name: Write [number]
Status: User
Input: A= Number to be written
B= Radix (1)
X= Output [file number]
Output: A= Same as input
B= Same as input
X= Same as input

Description: Writes the number in A to the file specified in X. Radixes up to 16 (hexidecimal) will work.
Instruction traps: Illegal file number in X, bad radix in B.

BRS 37

DATE: 71/09/22
Name: [Rename] a [file] to same length
Status: User
Input: A- First word of string pointer
B= Second word of string pointer
X= [File directory pointer address]
Output:
Skip Return: A= Destroyed
(success) B= Destroyed

X= Destroyed
 return: A= Destroyed
 (failure) B= Destroyed
 X= Exec ercode

Description: Renames the file whose file directory pointer address is in X to the file name (see Glossary) pointed to by A and B. The length of the new string (including comments) must be the same as the old string.

Example: LDP OLDPTR; BRS 48; BRS 10; LDP NEWPTR
 BRS 37; BRS 10; ...

OLDPTR DATA 3*OLDSTR-1,3*OLDSTR+7-1
 NEWPTR DATA 3*NEWSTR-1,3*NEWSTR+7-1
 OLDSTR ASC 'OLDNAME'
 NEWSTR ASC 'NEWNAME'

Instruction traps: None.

BRS 38

Date: 71/08/23

Name: Read [number]

Status: User

Input: B= Radix (1)
 X= Input [file number]

Output: A= Number
 B= Terminating character
 X= Same as input

Description: Reads a number from the file specified by the file number in X until a non-numerical (this includes .) character is found. A dash (-) as the first character will negate the number.

Instruction traps: Illegal file number in X, bad radix in B.

BRS 39

Date: 71/08/23

Name: Read [control parameter word]

Status: User

Input: None

Output: A= Control parameter word (see table below)
 B= [Universal user number]
 X= Same as input

Description: Reads the control parameter word into the A register and the universal user number into B. The A register has the following significance:

4B7 = Tymshare (Proprietary) Status
 2B7 = Not Used.
 1B7 = Operator Status.
 4B6 = Oper Login
 2B6 = Account Supervisor
 1B6 = Not Used.
 4B5 = Project Code Validation Required.
 2B5 = Validated on 2 or more Computers
 1B5 = Not Used

4B4 = Not allowed to change own password
 2B4 = Not Used.
 1B4 = Reserved.
 2B = "MAIL WAITING" message waiting

Instruction traps: None

BRS 40

DATE: 71/09/21

Name: Reads [echo table]

Status: User

Input: X= Port number (-1, 0 or 1 for command port)

Output: A= Echo table number + terminal characteristics

B= Preserved

X= Preserved

Description: Reads echo table. (See BRS 12). The BRS also returns the following bits in A:

Bit 0 (4B7) = Eight Level Input
 Bit 2 (1B7) = Lower Case Input
 Bit 4 (2B6) = Eight Level Output
 Bit 5 (1B6) = Echo Control I
 Bit 6 (4B5) = Echo LF as CR
 Bit 7 (2B5) = Echo CR as LF
 Bit 15 (2B4) = Lower Case Output

Instruction traps: Bad port number in X (unless interrupt 10 is armed).

BRS 41

Date: 71/08/23

Name: Read disc address of current [data block]

Status: System

Input: None

Output: A= Disc address of current data block

B= Preserved

X= Preserved

Description: Reads the disc address of the current data block for the last file accessed. Used in MAP for the raw accounting file.

Instruction traps: Insufficient status

BRS 42

Date: 71/08/23

Name: Read clock

Status: User

Input: None

Output: A= Number of clock ticks since startup

B= Last time of system start up

X= Year - 1964

Description: Returns the number of clock ticks (1/60 sec) since startup

in A, the last system start up time in B, and the year - 1964 in X.
Instruction traps: None

BRS 43

DATE: 71/09/21

Name: Read pseudo-[relabeling]

Status: User

Input: None

Output: A= First relabeling register (See Glossary)

B= Second relabeling register

X= Preserved

Description: Reads the pseudo-relabeling registers (See Appendix G)
for the currently active fork into A and B.

Instruction traps: None

BRS 44

DATE: 71/09/21

Name: Set pseudo-[relabeling]

Status: User, system or Exec

Input: A= First relabeling register (See Glossary)

B= Second relabeling register

B= Preserved

X= Preserved

Description: Sets the pseudo-relabeling for the current running fork.
If one of the bytes is 44B and is read-only, it will require at
least system status to relabel over; if it is read write, it will require
Exec status to relabel over. See Appendix G for further information
on relabeling.

Instruction traps: Swap error, insufficient status to relabel page,
attempt to relabel over a non-existent page (not the way to obtain more
memory!).

BRS 45

Date: 71/08/23

Name: [Dismiss] for 4 seconds

Status: User

Input: None

Output: A= Preserved

B= Preserved

X= Preserved

Description: Dismisses for 3 seconds

Instruction traps: None

BRS 46

Date: 71/08/23

Name: Set [non-terminability]

Status: Exec

Input: None

Output: A= Preserved

B= Preserved

X= Preserved

Description: After setting non-terminability, the first panic typed is stacked and further panics forgotten. A waiting panic can be tested for with a BRS 26.

Instruction traps: Insufficient status

BRS 47

Date: 71/08/23

Name: Clear [non-terminability]

Status: Exec

Input: None

Output: A= Preserved

B= Preserved

X= Preserved

Description: Clears non-terminability and remembers the stacked panic, if any.

Instruction traps: Insufficient status

BRS 48

DATE: 71/09/21

Name: [Lookup] [file] in [file directory] using string pointers

Status: User

Input: A= First word of string pointers

B= Second word of string pointers

Output:

Skip return: A= File directory pointer address

(success) B= Destroyed

X= File directory pointer address

return: A= Preserved

(failure) B= Preserved

X= File directory group number

Description: Takes the string pointer in A and B and looks it up in the appropriate file directory. If the file name (see Glossary) is badly constructed, or cannot be found, the non-skip return is taken. The error return can be used to find the file directory group number.

Instruction traps: None

BRS 49

Date: 71/08/23

Name: Read [interrupts] armed

Status: User

Input: None

Output: A= Interrupts armed (see BRS 78)

B= Preserved

X= Preserved

Description: Reads the interrupt mask into the A register.

Instruction traps: None

BRS 50

DATE: 71/09/21
Name: [Conversion] from [floating point] to fixed point
Status: User
Input: A= First word of floating point number
B= Second word of floating point number
Output: A= Integer part
B= Fractional part
X= Preserved
Description: Fixes the floating point number in A and B to A. The fraction part is placed in B left justified.
Instruction traps: None

BRS 51

DATE: 71/09/21
Name: [Conversion] from fixed point to [floating point]
Status: User
Input: A= Integer value
Output: A= First word of floating point number
B= Second word of floating point number
X= Preserved
Description: Converts the integer in A to a normalized floating point number in A and B.
Instruction traps: None

BRS 52

DATE: 71/09/21
Name: [Format]ted input
Status: User
Input: X= Format (See Appendix E)
Output:
Skip return: A= First word of floating point number
(success) B= Second word of floating point number
X= Destroyed
return: A= Destroyed
(errors) B= Destroyed
X= Error code (See Appendix E)
Description: Reads characters from the file specified through the format word in the X register. If any errors exist, the non-skip return will be taken.
Instruction traps: Bad file number in X.

BRS 53

DATE: 71/09/21
Name: [Format]ted output
Status: User
Input: A= First word of floating point number
B= Second word of floating point number
X= Format (See Appendix E)
Output: A= Preserved

B= Preserved

X= Preserved

Description: The floating point number (or integer) is output to the file number in the X register according to the formats listed in Appendix E.

Instruction traps: Bad file number in X.

BRS 54

Date: 71/08/23

Name: Grab bit from [bit map]

Status: Exec

Input: None

Output:

Skip Return: A= Disc address of bit grabbed

B= Preserved

X= Preserved

Return: A= Preserved

B= Preserved

X= Preserved

Description: Grabs a bit from the bit map and returns the disc address of that bit. If the map is not set, the BRS will not skip.

Instruction traps: Insufficient status

BRS 55

Date: 71/08/23

Name: Dismiss until disc work done

Status: User

Input: None

Output: A= Preserved

B= Preserved

X= Preserved

Description: Dismisses the job until all of the pending disc work is done.

Instruction traps: None

BRS 56

DATE: 71/09/20

Name: Make [PMT] pointer indirect for [Recover]

Status: Exec

Input: A= PMT byte number or SMT byte number to be pointed at.

B= PMT number

X= Channel number for new byte

Output: A= New byte number

B= Preserved

X= Preserved

Description: Recovers the page in A and places it in the byte number specified in B. Used by the Exec in the "recover" command.

Instruction traps: Bad byte number in B.

BRS 57

Date: 71/08/23

Name: Make [expanded SMT] byte accessable to users

Status: Exec

Input: A= Expanded SMT number

Output: A= Preserved

B= Preserved

X= Preserved

Description: Makes the byte in A accessable to users without status.

Used in WSD for FOS, F20S

Instruction traps: Insufficient status, bad byte number

BRS 58

DATE: 71/09/16

Name: [Rename] a [file]

Status: User

Input: A= First word of string pointer for old name

B= Second word of string pointer of old name

(Also first word of string pointer of new name)

X= Second word of string pointer of new name

Output:

Skip return: A= Destroyed

(success) B= Destroyed

X= Destroyed

return:

(failure) A= Destroyed

B= Destroyed

X= Ercode (see Appendix D)

Description: Renames the old file pointed to by A and B to the file name string concatenated to the old string as pointed to by B and X. If any errors occur, the non-skip return will be taken and the ercode put in X. Example:

```
RENAME LDP OLDPTR; LDX NEWPRT+1; BRS 58
      BRU ERROR (Error Routine); ...
```

```
OLDPTR DATA 3*OLDSTR-1
```

```
NEWPTR DATA 3*OLDSTR+6
```

```
      DATA 3*OLDSTR+17
```

```
OLDSTR ASC 'OLDFILENEWFILENAME'
```

The BRS 58 differs from the BRS 37, in that it can rename files to file lengths different than the old length.

Instruction traps: None

BRS 59

DATE: 71/09/16

Name: Change effective [User Name] for public files

Status: User

Input: A= Desired character to be used in user name in place of #.

Output: A= Destroyed

B= Destroyed

X= Destroyed

Description: The Exec will substitute the selected character in place of every occurrence of a "#" discovered in a user name. If the mode is to be reset, then A is to be set to -1 and the BRS called again.
Instruction traps: None

BRS 60

DATE: 71/09/16

Name: Lookup [file name] in [file directory] for output

Status: User

Input: A= First word of string pointer

B= Second word of string pointer

Output:

Skip return: A= File directory pointer address

(success) B= Destroyed

X= File directory pointer address

return: A= Destroyed

(failure) B= Destroyed

X= Ercode (see appendix D)

Description: The file name is looked up in the file directory and checked for legality (see Glossary). The BRS 60 is usually followed by a BRS 19. The BRS 60 is to the BRS 19 as the BRS 48 is to the BRS 16.

Instruction traps: None

BRS 62

DATE: 71/09/16

Name: [Open] [file] for [input]

Status: User

Input: A= Bits 0-7 = random file options (see Appendix B)

8-23= command file number

Output:

Skip return: A= File number

(success) B= File type (see Glossary)

X= File size in characters

return: A= Destroyed

(failure) B= Destroyed

X= Exec ercode (see Appendix D)

Description: Reads the input file name from the command file specified in bits 8-23 of A. (See Glossary on valid file names). If bits 0-7 are 0, the default will be: 1). The ability to do an SCP and 2). The file is set to read mode. Any attempt to open an input file for output will result in a trap.

Instruction traps: Attempt to set write mode, bad command file number in A.

BRS 63

DATE: 71/09/16

Name: [Open] [file] for [output]

Status: User

Input: A= Bits 0-7 = random file options (see Appendix B)

8-23= command file number
X= File type

Output:

Skip return: A= File number
(success) B= Old file type (if any)
X= Destroyed
return: A= Destroyed
(failure) B= Destroyed
X= Exec ercode (see Appendix D)

Description: Reads the output file name from the command file specified in bits 8-23 of A. (See Glossary on valid file names). If bits 0-7 are 0, the default is taken to be: 1). The ability to do an SCP, 2). The ability to erase information (PCE) and 3). The file is set to write mode. If the read mode bit is on, the file is opened as [random I/O] and the original contents of the file are not destroyed. If the file being opened for random I/O is an append only file, any attempts to do a PCE will fail. If bits 0-7 of A are 0, the file is erased and all data blocks released.

Instruction traps: Bad command file number in A.

BRS 64

DATE: 71/09/16

Name: [Open] [file] for [input] using string pointers

Status: User

Input: A= bits 0-7 = random file options (see Appendix B)
bits 8-23= first word of string pointer
B= second word of string pointer

Output:

Skip return: A= File number
(success) B= File type (see Glossary)
X= File size in characters
return: A= Destroyed
(failure) B= Destroyed
X= Exec ercode (see Appendix D)

Description: Opens the file whose string pointers are in A and B for input. The BRS 64 obeys all the rules regarding random files as the tBRS 62. See BRS 62 for further information.

Instruction traps: None

BRS 65

DATE: 71/09/16

Name: [Open] [file] for [output] using string pointers

Status: User

Input: A= bits 0-7 = random file options (see Appendix B)
bits 8-23= first word of string pointers
B= Second word of string pointers
If 'OLD FILE'/'NEW FILE' message is to be typed:
X= bits 0-11 = file type (see Glossary)
bits 12-23= command file number
If 'OLD FILE'/'NEW FILE' message is not to be typed:
X= bits 0-11= 0
bits 12-23=file type (see Glossary)

Output:

Skip return: A= file number
 (success) B= Old file type (if any)
 X= Destroyed
 return: A= Destroyed
 (failure) B= Destroyed
 X= Exec ercode (see Appendix D)

Description: Opens the file whose string pointers are in A and B for output. The BRS 65 obeys the same rules for random files as the BRS 63. See the BRS 63 for further information. The main difference is the 'OLD FILE'/'NEW FILE' question, which can be controlled by the setting of bits 0-11 of X.

Instruction traps: Bad command file number in X.

BRS 66

DATE: 71/09/16

Name: Erase disc [file]

Status: Exec

Input: A= File number

Output:

Skip return: A= Preserved
 (success) B= Preserved
 X= Preserved
 return: A= Preserved
 (failure) B= Preserved
 X= Preserved

Description: Erases the file whose file number is in A. The non-skip return is taken if the map is not set of on I/O error.

Instruction traps: Insufficient status, bad file number or file not opened.

BRS 67

DATE: 71/09/17

Name: Read [TS page]

Status: User

Input: A= Index into table
 B= Location in user's memory
 X= Number of words to be copied into memory

Output: A= Destroyed
 B= Destroyed
 X= Destroyed

Description: Reads into memory a section of the user's TS page. The index will allow the program to get only a selected portion. Example: To print out the project code:

```
PROJ  LDA =59; LDB =BFR; LDX =4; BRS 67
      LDA =BFR; LDB =12; LDX =1; BRS 34
      TCO =155B; BRS 10
BFR   BSS 4
```

List of indices:

List of Indices for the BRS 67:

Index	Description
-----	-----
0-5	Words 2-6 of the Exec Panic Table
6	Command Input File
7	Command Output File (TOUT File)
8	User Number
9	Status Flag Word
10-14	Exec Pop Vectors
15-16	Exec Relabeling
17-18	Program Relabeling
19-20	Subsystem Relabeling
21	Number of clock ticks at Login
22-28	Exec Panic Table
29	Ercode (See Appendix D)
30	REMAC
31	Account Number
32	File Directory LUD locator
33	LUD locator for this user (Same as WERIS)
34	File Directory User Number
35	Current Group Number
36	Number of Groups
37	Last Exec ID number
38-39	Temporary Exec Storage
40	Used as switch for type of Logon for Accounting
41	Disc Buffer Location in Exec
42	BRS 48/60 switch, negative if BRS 48/60
43	Proprietary Program Switch
44	Break Table for Exec BRS's
45	File Directory Account Number (Set by GFD)
46	Init Switch
47	Next Account Storage (For Disc Error)
48-54	Temporary Exec Storage
55	Name Change Character (-1 for Non Active) See BRS 59
56	Error Switch (-1 to type all Error Messages)
57	SWOFF
58	SWTM
59-62	Project Code
63	File Position in directory
64	File Directory Pointer Address for last File
65	Current File Directory Group Number

Warning: If the buffer specified does not point to page somewhere in the relabeling, the Exec will place the data in a new page if possible, but will not (!!!) affect the user relabeling.

Instruction traps: Bad index in A.

BRS 68

DATE: 71/09/17

Name: Reads [file] name from [file directory] into memory

Status: User

Input: A= Null string pointer
 B= Null string pointer
 X= Index to file into file directory

Output:

Skip return: A= File directory pointer address
 (success) B= Ending string pointer
 X= Destroyed
 return: A= 0 if index is invalid
 (failure) B= Destroyed
 X= Exec ercode

Description: Reads the file name for the index in X. If X is greater than the last file position, the BRS will skip. Example: Types all file names in a directory.

TYPE CLX

LOOP EAX 1,2; STX COUNT; LDP PTR; BRS 68; BRS 10 (ALL DONE)
 LDA PTR; LDX =1; BRS 35; TCO =155B; LDX COUNT; BRU LOOP
 PTR DATA 3*STRING-1, 3*STRING-1
 COUNT ZRO
 STRING BSS 3*70

Warning: If the buffer specified does not point to page somewhere in the relabeling, the Exec will place the data in a new page if possible, but will not (!!!) affect the user relabeling.
 Instruction traps: Directory declared "Not Listable"

BRS 69

DATE: 71/09/20
 Name: [Delete] a [file]
 Status: User
 Input: A= File directory pointer address
 Output:
 Skip return: A= Destroyed
 (success) B= Destroyed
 X= Destroyed
 return: A= Destroyed
 (failure) B= Destroyed
 X= Exec ercode

Description: Deletes the file whose file directory pointer address is in A. If the file is write protected, the message "WRITE PROTECTED" is printed and the error return taken.

Instruction Traps: None.

BRS 70

Date: 71/08/23
 Name: Count free [pages]
 Status: User
 Input: None
 Output: A= Number of free pages
 B= Preserved

X= Preserved

Description: Returns number of pages not in use. The amount of unused memory can be calculated by doing a LSH 2 with B cleared.
 Instruction traps: None

BRS 71

Date: 71/08/23
 Name: Skip if System or Exec status set
 Status: User
 Input: None
 Output:
 Skip Return: A= Preserved
 B= 0
 = 4B7
 = 6B7
 X= Preserved
 Return: A= Preserved
 B= 0
 X= Preserved

Description: Skips if the user has set System or Exec.
 Instruction traps: None

BRS 72

Date: 71/08/23
 Name: Exec [dismiss]
 Status: Exec
 Input: B= PTEST (see BRS 7)
 X= Queue number
 Output:
 Skip Return: A= Queue number
 B= PTEST
 X= Queue number
 Return: None, always skips
 Description: Dismisses the job on the queue in X with the PTEST in B. The queue numbers are:

X	Queue
----	-----
0	QTI
1	QIO
2	QSQ
3	QQC
4	QQE

Instruction traps: Insufficient tables

BRS 73

Date: 71/08/23
 Name: Read [ERCODE]
 Status: User
 Input: None

Output: A= ERCODE
B= Preserved
X= Preserved

Description: Returns the ERCODE word in A. If the ERCODE is 99, then the address is the monitor address that detected the error.
Instruction traps: None

BRS 74

Date: 71/08/23

Note: This BRS no longer exists, use the following code:
CLX; BRS 132; MRG =400B; BRS 160

BRS 75

Date: 71/08/23

Note: This BRS no longer exists, use the following code:
CLX; BRS 132; ETR =2400B; BRS 160

BRS 76

Date: 71/08/23

Note: This BRS no longer exists, use the following code:
CLX; BRS 132; SKA =400B; SKIP; NON SKIP

BRS 77

DATE: 71/09/20

Name: Change [TOUT] file

Status: User

Input: A= File number

Output: A= Old tout file number

B= Preserved

X= Preserved

Description: Closes the present teletype output file if one is open and uses the output file specified in A. This BRS works in conjunction with the Exec "TOUT" command, which causes all Exec output to go to a file.

Instruction traps: Bad file number in A.

BRS 78

DATE: 71/09/20

Name: Arm/Disarm [software interrupt]

Status: User

Input: A= New interrupt mask

Output: A= Preserved

B= Preserved

X= Preserved

Description: The new interrupt mask in A is substituted for the old one. A user status program may arm interrupts 1-10; a system status program may arm interrupt 11 (disc errors) also. Location 200B +

interrupt number (1-13B) must have the address of the routine to process the interrupt. When the interrupt occurs, the program counter is stored at the address of the routine and execution of the interrupt routine begins at the address specified in location 200B+ interrupt number + 1. The interrupts currently implemented are:

No.	Description of Interrupt
1	Interrupt if Program Panic (Includes Terminal "Panics" and BRS 10s)
2	Interrupt if Memory Panic (Read Only Violation, etc.)
3	Interrupt if Lower Fork terminates
4	Interrupt on unusual I/O condition: A register contains: Bit 0 = 1 Bit 5 = 1 if File Size Overflow Bit 6 = 1 if I/O error Bit 7 = 1 if End of File Bits 18-23 = File number
5	Interrupt on timeout. The BRS 78 can only disarm, not arm. See BRS 135.
6	Interrupt on Floating Point Overflow
7	Interrupt on Floating Point Underflow (Not Implemented)
8	Interrupt on Buffer Cleared, but circuit still valid.
9	Interrupt on dismiss will occur on CIO, in or out (See BRS 155)
10	Interrupt on Bad file number for CIO. Could be a zapped circuit.
11	Interrupt on Disk Error. Requires System Status to set.

Example:

```
ARM   LDA =PANIC; STA 201B (Interrupt Location); LDA =2000000B
      BRS 78 (ARM FOR PANICS)
```

```
PANIC ZRO PC   Save program counter in PC
```

```
.
```

```
Do interrupt routine
```

```
.
```

```
LDA =Z000000B; BRS 78 (RE-AMR);BRU* PC
```

```
PC   ZRO   0   PC SAVE
```

Each time a panic is hit, the code at "PANIC" is executed.
Instruction traps: None.

BRS 80

DATE: 71/09/20

Name: Set page [read only]

Status: User or Exec

Input: A= PMT/SMT number

Output: A= Old status of byte

B= Preserved

X= Preserved

Description: If the sign bit of A is on, the byte will be made read only. If the sign bit is off, the byte will be made read-write.

Any attempt to store into a read-only page will result in a memory trap or take the interrupt if armed. Only Exec status forks can set pages read-write. Expanded SMT's cannot be made read-write via the BRS 80.

Instruction traps: Insufficient status, bad byte number, byte number not in use.

BRS 81

DATE: 71/09/20

Name: Dismiss for a specified period of time

Status: User

Input: A= Number of milliseconds to dismiss for

Output: A= Preserved

B= Preserved

X= Preserved

Description: The fork is dismissed for the number of milliseconds in A. The minimum dismissal time is 2 seconds.

BRS 82

DATE: 71/09/21

Name: Skip if [floating accumulator] negative

Status: User

Input: None

Output:

Skip return: A= Preserved

(success) B= Preserved

X= Preserved

return: A= Preserved

(failure) B= Preserved

X= Preserved

Description: Skips if the floating accumulators are negative. If floating point hardware does not exist, the sign bit of the simulated accumulator will be tested.

Instruction traps: None

BRS 83

DATE: 71/09/21

Name: Skip if [floating accumulator] zero

Status: User

Input: None

Output:

Skip return: A= Contents of floating accumulator

(success) B= Contents of floating accumulator

X= Preserved

return: A= Contents of floating accumulator

(failure) B= Contents of floating accumulator

X= Preserved

Description: Skips if the floating accumulators are zero. Also places the contents of the floating accumulator in A and B. If floating point hardware does not exist, the simulated accumulator will be tested.

Instruction traps: None

BRS 84

DATE: 71/09/21

Name: Skip if [floating accumulator] non-zero

Status: User

Input: None

Output:

Skip return: A= Contents of floating accumulator
(success) B= Contents of floating accumulator
X= Preservedreturn: A= Contents of floating accumulator
(failure) B= Contents of floating accumulator
X= Preserved

Description: Skip if the floating accumulators are non-zero. Also places the contents of the floating accumulators in A and B. If floating point hardware does not exist, the simulated accumulator will be tested.

Instruction traps: None

BRS 85

DATE: 71/09/21

Name: Set 3-level output mode

Status: User

Input: X= Port number (-1, 0 or 1 for command port)

Output: A= Preserved
B= Preserved
X= Preserved

Description: Sets terminal to [eight-level] output mode, eight bit characters to transmitted to the terminal exactly as they are from the program.

Instruction traps: Bad port number in X (unless interrupt 10 is armed).

BRS 86

DATE: 71/09/21

Name: Reset 3-level output mode

Status: User

Input: A= Port number (-1, 0 or 1 for command port)

Output: A= Preserved
B= Preserved
X= Preserved

Description: Resets [eight-level] mode. Characters are hence forth translated by the monitor.

Instruction traps: Bad port number in X (unless interrupt 10 is armed).

BRS 87

DATE: 71/07/22

Name: Convert [port number] to job number

Status: Subsystem

Input: X = Port number

Output: A = Job number (If any)

B = Preserved

X = Preserved

Description: Converts the port number in X to a job number returned in A.

Instruction traps: Insufficient status

BRS 88

DATE: 71/09/10

Name: Read [CPU] time

Status: User

Input: None

Output: A= CPU time in clock ticks (1/60 second)

B= Preserved

X= Preserved

Description: Reads the CPU time for the job and returns it in A.

Instruction traps: None

BRS 89

DATE: 71/09/10

Name: Read [resource metering]

Status: User

Input: A= Address of 6 word buffer to put data

Output: A= Preserved

B= Preserved

X= Preserved

Description: Reads into the buffer 6 words of resource metering described below.

Word	Description
----	-----
0	Disc Use. Number of Disc accesses
1	Swap Count. Number of times job was swapped
2	Total number of characters input and output
3	Page Ticks. 1 Page tick = 1 page/clock tick
4	CPU time in clock ticks
5	Number of clock ticks since login

Instruction traps: None

BRS 90

DATE: 71/09/10

Name: Declare fork for [panic]

Status: User

Input: A= Preserved

B= Preserved

X= Preserved

Description: Indicates that the current fork is the highest to [terminate] if the user types a panic. If the fork which did the BRS 90 has armed interrupt 1, the interrupt will be take instead of terminating

the fork.

Instruction traps: None

BRS 91

DATE: 71/09/10

Name: Store [date] and [time] into a string

Status: User

Input: A= First word of string pointer

Output: A= Preserved

B= String point + length of date string

X= Preserved

Description: Reads the current date and time into a null string address pointed to by A. The resulting string has the form MM/DDSHH:NNSSSS

where: M= Month

D= Date

H= Hours (from 0-23)

N= Minutes

S= Space (note: 4 spaces at end)

Instruction traps: Invalid pointer in A.

BRS 92

DATE: 71/09/10

Name: Set value of [file size quantum] for all files to be opened until logout time

Status: User

Input: A= Maximum character count

Output: A= Preserved

B= Preserved

X= Preserved

Description: Sets the physical file size quantum allotted to open output files. A must be between 7377B and 20000000B. If A is between 3248640D and 20000000B, the maximum quantum is set to 3248640D, which is the absolute maximum for all files. The maximum set at logon time is 1364000B characters. If the file quantum is dep;eted, an instruction trap will occur at the syspop which caused it unless interrupt 4 is armed. The SSP syspop overrides the effect of a BRS 92 for the file the referenced by the SSP.

Note: Because of overhead in the file system, physical size is usually greater than data size. A one character data file requires 3000B character of physical space on the disc. Overhead for large files is about 3 percent.

Instruction traps: Bad argument in A.

BRS 93

DATE: 71/09/10

Name: Reset resource metering

Status: Exec

Input: None

Output: A= Preserved

B= Preserved

X= Preserved

Description: Resets the resource metering counters.
(See BRS 89).

Instruction traps: Insufficient status

BRS 94

DATE: 71/09/10

Name: Clear fork structure

Status: System

Input: None

Output: A= Destroyed

B= Destroyed

X= Destroyed

Description: Returns job to the Exec at the instruction following the BRS 9. No panic tables are saved.

Instruction Traps: Insufficient Status

BRS 95

DATE: 71/09/10

Name: Acquire and release [over flow groups]

Status: Exec

Input: A= 0 to return next available overflow pointer
= Overflow pointer to turn on/off bit in bit map

Output: A= Destroyed or contains next available overflow pointer

B= Destroyed.

X= Destroyed.

Description: If A is 0, then the next available overflow group is returned in A. If A is an overflow group (file directories 4000B-4377B), and the sign bit is off, the overflow group is released and it's corresponding bit turned off in the bit map. If A is an overflow group and the sign bit is on, the overflow group is taken and the bit turned on in the bit map.

Instruction traps: Insufficient status.

BRS 96

DATE: 71/09/16

Name: Read [file attributes] and [file directory] data

Status: User

Input: B= Buffer address

X= Word count

Output: A= Contents of last word in buffer

B= Destroyed

X= Destroyed

Description: Reads the file attributes and file directory data for the last file accessed by an Exec BRS (for example, a BRS 48 or 62) into the buffer specified by B. Data in buffer:

Word 0: Group number in bits 0-11

Prime directory in bits 12-23

Words 1-4: File directory data words (see below)

Words 5-N: File name with sign bit on in last word

File directory data words:

```

Word 1:  OAA AAA AAA AAY YYY EEE EDD DDD
Word 2:  BCO HHH HHH HHH FFF FFF FFF FFF
Word 3:  GGG TTT 000 0IZ XWV USR QNM LKJ
Word 4:  PPP PPP PPP PPP PPP PPP PPP PPP

```

Where:

```

A = Access Count
B = Changed File
C = Extra Changed File
D = Creation Day-1
E = Creation Month-1
F = File Size (1 = 256 words)
G = Good Disc File if 010, BAD if 100 and Init Dummy if 000
H = Mapping Control
I = Init lock Flag           2B4
J = Init Flag                1
K = Exec status              2
L = System Status            4
M = Subsystem Status         10B
N = Proprietary              20B
P = Index Block Pointer
Q = Account Public           40B
R = Append Only              1B2
S = Not Private Writable     2B2
T = File Type (From 1 - 4)
U = Not Private Read         4B2
V = Reserved.
W = Public Write             2B3
X = Public Remote           4B3
Y = Creation Year - 1964
Z = Premium Charge           1B4

```

See the BRS 6 for a fuller explanation of these fields.

Warning: If the buffer specified does not point to a page somewhere in the relabeling, the Exec will place the data in a new page if possible, but will not (!!!) affect the user relabeling.

Instruction traps: None

BRS 97

```

DATE: 71/09/16
Name: Reset subsystem counter
Status: Subsystem
Input:  A= Counter number (0-31)
        B= Value to be set
Output: A= Preserved
        B= Preserved

```

X= Preserved

Description: Sets the counter in A to value to B.

Instruction traps: Insufficient status, bad counter number in A.

BRS 98

DATE: 71/09/16

Name: Increment subsystem counter

Status: Subsystem

Input: A= Counter number (0-31)

B= Number to be added to counter in A

Output: A= Preserved

B= Preserved

X= Preserved

Description: Adds contents of B to the counter in A.

Instruction traps: Insufficient status, bad counter number in A.

BRS 99

DATE: 71/09/16

Name: Reads subsystem counter

Status: Subsystem

Input: A= Counter number (0-31)

Output: A= Contents of counter

B= Preserved

X= Preserved

Description: Reads the subsystem counter specified by A.

Instruction traps: Insufficient status, bad counter in A.

BRS 100

DATE: 69/05/13

FUNCTION: ASSIGNS A DEVICE TO A USER

STATUS: OPERATOR

INPUT: A= DEVICE NUMBER

X= CHANNEL NUMBER OR -1

DEVICE NUMBERS: 0=TAPE0, 1=TAPE1, 2=PRINTER

RETURNS: NO SKIP=ERROR, DEVICE ALREADY ASSIGNED.

SKIP=NORMAL RETURN

DESCRIPTION: THE DEVICE IS ASSIGNED TO THE USER. AFTER THAT THE USER CAN DRIVE THE DEVICE DIRECTLY USING THE OTHER BRS'S. MAG TAPE IS SET TO ODD PARITY.

ONLY ONE DEVICE CAN BE ASSIGNED AT A TIME.

BRS 101

DATE: 69/05/13

FUNCTION: UNASSIGN DEVICE

STATUS: OPERATOR

DESCRIPTIONS: RESETS DEVICE ASSIGNMENT. USER CAN NO LONGER ACCESS DEVICE WITH BRS'S.

REGISTERS AFFECTED: NONE

BRS 102

DATE: 70/09/20

FUNCTION: READ [MAG TAPE]

STATUS: USER

INPUT: A= CORE BUFFER ADDRESS, B= CORE BUFFER SIZE
X= NUMBER OF RECORDS (MAX 64)

THIS BRS READS MANY RECORDS INTO CORE. IT ASSUMES THAT THE RECORD SIZE IS UNKNOWN. IT READS SUCESSIVE RECORDS INTO THE BUFFER AND PRECEEDS EACH RECORD WITH A CODE WORD THAT EXPLAINS ABOUT THAT RECORD. IN THE NORMAL CASE THE ADDRESS FIELD OF THIS WORD HOLDS THE NUMBER OF 940 WORDS IN THE RECORD. FOR INTERPRETATION OF THE OPCODE FIELDS SEE APPENDIX C. THE BUFFER MUST BE ENTIRELY IN ONE PAGE.

BRS 103

DATE: 71/06/30

FUNCTION: WRITE [MAG TAPE]

STATUS: USER

INPUT: A= CORE ADDRESS OF DATA IN THE FOLLOWING FORMAT.

(A) COMMUNICATION FROM SYSTEM TO USER.

(A)+1 1ST RECORD WORD COUNT

(A)+2 THROUGH (A)+N+1 RECORD DATA.

(A)+N+2 NEXT RECORD WORD COUNT.

ETC. COUNT =0 AFTER LAST RECORD.

OUTPUT: (A) CONTAINS ADDRESS OF "WORD COUNT" WORD FOR LAST RECORD WRITTEN. THE OP CODE OF (A) CONTAINS ONE OF THE FLAGS LISTED UNDER APPENDIX C.

DESCRIPTION: ALL THE DATA AND COMMUNICATIONS WORDS MUST BE IN ONE PAGE. USER MUST ERASE TAPE AT LOAD POINT AND MAY NOT WRITE BEYOND THE REFLECTIVE SPOT EXCEPT FOR END OF FILE MARKS.

REGISTERS AFFECTED: NONE

BRS 104

DATE: 69/05/13

FUNCTION: REPORTS WHO HAS DEVICE

STATUS: USER

OUTPUT: A= DEVICE
X= CHANNEL

DESCRIPTION: CAN BE USED TO DETERMINE WHAT W-BUFFER DEVICE IS ASSIGNED AND WHAT CHANNEL THE USER IS ON WHO IS USING THE DEVICE.

DEVIOL NUMBERS ARE 0=TAPE 0, 1=TAPE 1, 2=PRINTER.

A -1 IN THE A REGISTER MEANS NO DEVICE IS ASSIGNED.

REGISTERS AFFECTED: A,X.

BRS 105

DATE: 69/05/31

FUNCTION: [MAG TAPE] CONTROLS

STATUS: USER

INPUT: A= CONTROL NUMBER.
1= WAIT UNTIL TAPE IS READY
2= BACKSPACE RECORD
3= FORWARD SPACE FILE

4= BACKSPACE FILE
5= WRITE 3 INCHES OF BLANK TAPE
6= REWIND
7= WRITE END OF FILE.

OUTPUT: THE OP CODE OF A HAS ONE OF THE FLAGS LISTED UNDER
APPENDIX C. THE ADDRESS OF A IS DESTROYED.
REGISTERS AFFECTED: A

BRS 106

DATE: 71/02/03

FUNCTION: PRINT ON [PRINTER]

STATUS: USER

INPUT: A= CORE ADDRESS OF DATA IN THE FOLLOWING FORMAT, X= WORD COUNT

(A) COMMUNICATION WORD FROM SYSTEM TO USER. SEE APPENDIX C

(A)+1 PAPER CONTROL. NEG FOR SKIP, POSITIVE FOR UPSPACE.

(A)+2 - (A)+34 ONE LINE OF DATA IN 6 BIT BCD.

(A)+35 PAPER CONTROL

ETC.

OUTPUT: (A) CONTAINS ONE OF THE OP CODE FLAGS LISTED UNDER BRS 102.

DESCRIPTION: ALL DATA AND COMMUNICATIONS WORDS MUST BE IN ONE PAGE.

ALL LINES MUST BE FULL LENGTH.

IT IS REQUIRED THAT THE PRINTER BE ASSIGNED TO THE CURRENT JOB.

IT IS NOT NORMAL OPERATING PROCEDURE TO ASSIGN THE PRINTER TO

JOBS NOT RUN BY THE OPERATOR.

REGISTERS AFFECTED: NONE

BRS 107

DATE: 69/05/13

FUNCTION: SET [MAG TAPE] [PARITY]

STATUS: USER

INPUT: A IS NEG. FOR [BCD] (EVEN PARITY).

A IS POSITIVE FOR BINARY (ODD PARITY).

DESCRIPTION: IF THIS BRS IS NOT USED, TAPE WILL BE READ IN BINARY.

REGISTERS AFFECTED: NONE

BRS 108

DATE: 69/05/13

FUNCTION: TEST [MAG TAPE] [DENSITY]

STATUS: USER

OUTPUT: A=0 FOR 200

A=1 FOR 556

A=2 FOR 800

REGISTERS AFFECTED: A

BRS 109

DATE: 71/02/06

FUNCTION: DISMISS

STATUS: USER

CALLING SEQUENCE: BRS 109

DESCRIPTION: THE FORK IS DISMISSED. IT CAN ONLY BE ACTIVATED

AGAIN BY A PROGRAM INTERRUPT WHICH HAS BEEN ARMED BY THIS FORK
OR THE TERMINATION OF A LOWER FORK.
REGISTERS AFFECTED: NONE

BRS 110

DATE: 71/02/03
FUNCTION: TEST [MAG TAPE] READY
STATUS: USER
RETURNS: NO SKIP = NOT READY
 SKIP = READY

BRS 111

DATE: 69/05/13
FUNCTION: RETURN FROM CLASS 3 BRS
STATUS: EXEC
DESCRIPTION: THIS BRS IS USED ONLY BY THE AUTHOR OF CLASS 3
BRS'S. IT IS THE ONLY NORMAL TERMINATION OF A CLASS 3 BRS. IT
CORRESPONDS TO A BRS 10 FOR OTHER FORKS.
INSTRUCTION TRAP:
BRS ISSUED BY A FORK WHICH WAS NOT A CLASS 3 BRS.
REGISTERS AFFECTED: NONE

BRS 112

DATE: 69/05/13
FUNCTION: REMOVE A JOB FROM THE SYSTEM
STATUS: EXEC
INPUT: A= JOB NUMBER
NO RETURNS
REGISTERS AFFECTED: ALL

BRS 113

DATE: 70/01/30
FUNCTION: DISABLE ESCAPES
STATUS: USER
DESCRIPTION: ALLOWS USER TO PREVENT TERMINATION FROM AN ALTMODE
OR A SHIFT-CONTROL O. IT IS RESET BY A BRS 114
REGISTERS AFFECTED: NONE

BRS 114

DATE: 70/01/30
STATUS: USER
DESCRIPTION: RESETS THE CONDITION SET BY A BRS 113.
REGISTERS AFFECTED: NONE

BRS 115

DATE: 69/05/13
FUNCTION: TERMINATE CLASS 3 BRS FORK WITH PANIC
STATUS: EXEC BRS
DESCRIPTION: TERMINATES THE EXEC BRS AND CAUSES A RUBOUT FOR THE JOB.

REGISTERS AFFECTED: NONE

BRS 116

DATE: 69/05/13

FUNCTION: READ PROGRAM RELABELING

STATUS: USER

OUTPUT: A,B = PROGRAM PSEUDO-RELABELING.

DESCRIPTION: PUTS THE PROGRAM RELABELING INTO A AND B. THIS IS WHAT THE SYSTEM EXECUTIVE USES AS PROGRAM RELABELING. IT IS KEPT IN THE TS BLOCK.

REGISTERS AFFECTED: A, B

BRS 117

DATE: 69/05/13

FUNCTION: SET PROGRAM RELABELING

STATUS: USER

INPUT: A,B= THE NEW VALUES FOR THE PROGRAM RELABELING

DESCRIPTION: SETS THE PROGRAM RELABELING IN THE TS BLOCK AS SPECIFIED. USER PROGRAMS SHOULD USE BRS 44 TO SET RELABELING FOR A FORK.

INSTRUCTION TRAP:

1) A SPECIFIED RELABELING BYTE WAS NOT ASSIGNED.

2) A USER FORK TRIED TO RELABEL A SYSTEM BYTE.

THIS IS THE PROGRAM RELABELING TYPED BY THE STATUS COMMAND. IT SHOULD CORRESPOND TO THE RELABELING OF THE FIRST NON-SUBSYSTEM FORK BELOW THE EXECUTIVE. IF THE FORK IS RUNNING UNDER DDT, DDT WILL UPDATE THIS RELABELING. OTHERWISE, IT IS THE RESPONSIBILITY OF THE USER. IT IS PARTICULARLY IMPORTANT THAT THIS RELABELING BE SET CORRECTLY BEFORE ISSUING A DUMP COMMAND.

REGISTERS AFFECTED: NONE

BRS 118

DATE: 70/07/05

FUNCTION: LOAD FLOATING ACCUMULATOR FROM A & B

STATUS: USER

DESCRIPTION: LOADS THE FLOATING ACCUMULATOR FROM THE A AND B REGISTERS. IF FLOATING POINT HARDWARE IS NOT IMPLEMENTED, THE SIMULATED ACCUMULATORS WILL BE SET.

REGISTERS AFFECTED: NONE

BRS 119

DATE: 70/07/05

FUNCTION: STORE FLOATING ACCUMULATOR INTO A & B

STATUS: USER

DESCRIPTION: STORES THE CONTENTS OF THE FLOATING ACCUMALATORS INTO REGISTERS A AND B. IF FLOATING POINT HARDWARE IS NOT IMPLEMENTED, THE SIMULATED ACCUMULATORS WILL BE READ INTO A AND B.

REGISTERS AFFECTED: A,B

BRS 130

DATE: 69/05/13

FUNCTION: TEST A BREAKPOINT SWITCH

STATUS: SUBSYSTEM

CALLING SEQUENCE: LDX =SWITCH NUMBER

BRS 130

SWITCH UP RETURN

SWITCH DOWN RETURN

DESCRIPTION: TESTS THE BREAKPOINT SWITCH (1,2,3,4) INDICATED
IN X. IF THE SWITCH IS DOWN, THE BRS SKIPS ON RETURN.

REGISTERS AFFECTED: NONE

BRS 131

DATE: 69/05/13

FUNCTION: TO CRASH THE SYSTEM

STATUS: EXEC

NO RETURN

DESCRIPTION: SAVES THE REGISTERS IN SS01, SS02, SS03. SAVES
0 IN MCR0. TURNS OFF THE CLOCK AND DISABLES THE INTERRUPTS.

MOVES THE TS BLOCK INTO REAL PAGE 14.

REGISTERS AFFECTED: NONE

BRS 132

DATE: 71/07/23

Name: Read Terminal Characteristics

Status: User

Input: X = 0 or - 1 for user's [Command Port]
or (Port/2-10) for port number

Output: A = [Terminal Characteristics]

B = Preserved

X = Preserved

Description: Reads terminal characteristics and returns them in A. If
X contains an invalid port number, and interrupt 10 is armed, the interrupt
will be taken. The terminal characteristics are derived from the
table below:

Byte 0 (Bits 0-3)

Bit 0 = Can Echo

Bit 1 = Echo Control I

Bit 2 = Echo CR Rubout to LF

Bit 3 = Echo LF to CR

Byte 1 (Bits 4-7)

Bit 0 = CR Delay

Bit 1 =]

Bit 2 =] Input baud Rate

Bit 3 =]

Byte 2 (Bits 8-11)

Bit 0 =]
 Bit 1 =] Output baud Rate
 Bit 2 =]
 Bit 3 = Requires Parity (1 for on)

Byte 3 (Bits 12-15)

Bit 0 =]
 Bit 1 =] Parameter "A" (See Below)
 Bit 2 =]
 Bit 3 = Half Duplex (1 for on)

Byte 4 (Bits 16-19)

Bit 0 =]
 Bit 1 =] Parameter "B" (See Below)
 Bit 2 =]
 Bit 3 = Not Used

Byte 5 (Bits 20-23)

Bit 0 =]
 Bit 1 =] Parameter
 Bit 2 =] "C"
 Bit 3 =]

Both the Input and Output baud rates are calculated by an index into a table of baud rates. To arrive at this index, take the desired baud rate and divide it into 4800, that is: $(4800/\text{Baud Rate}) = N$.

N is then compared for the closest match in the following table:

Index	"N"
0	54B [110 Baud]
1	40B [150 Baud]
2	20B [300 Baud]
3	14B
4	10B
5	4B
6	4B
7	4B

The "A", "B" and "C" parameters are what determine both the speed of the terminal and the carriage return delay. If the terminal is a Line fee delay terminal, then the delay is calculated by the following algorithm: If $n=1$, then $F(n) = "A"$ else if "C" n , then $F(n) = C - n + B$ else $F(n) = B$.

where n is the number of characters in a line to be output.
 If the terminal is a Carriage Return Delay device (Bit 0 of Byte 1),
 then the delay is calculated as thus: $F(n)=\text{Min}[n/(2^{**}A)]+B,C$
 The "C" parameter is calculated by an index into the following table:

Index	CPARAM
-----	-----
0	10D
1	14D
2	19D
3	26D
4	36D
5	50D
6	69D
7	95D
10	131D

Each terminal when it is logged in has it's terminal characteristics set by the Exec. The following is a list of the Identifying Characters and their parameters:

ID	Terminal Characteristics
--	-----
A	56440000B
C	56444307B
E	56444146B
G	54442013B
B	56220000B
F	56244307B
J	56230000B
N	56244146B
CR	56000000B
D	56000000B

The only difference between the CR and D identifier is that the remote knows the differences. Therefore, a terminal cannot change into a 2741 and back again. Only a 2741 can change and return to it's original state.

Instruction Traps: Invalid port number (If interrupt 10 is not armed).

BRS 134

DATE: 71/08/24

Name: Read in characters in "[Paper Tape mode]"

Status: User

Input: None

Output: A = Preserved

B = Preserved

X = Preserved

Description: This BRS is no longer functional. It presently is a NOP. Programs using this BRS should do their own routines

to read in characters and test for linefeed before carriage return and vice versa.
Instruction Traps: None

BRS 135

DATE: 71/09/22

FUNCTION: INTERRUPTS A FORK AFTER A SPECIFIED PERIOD OF [TIME]

STATUS: USER

A= THE NEW [INTERRUPT] MASK.

DESCRIPTION: THE FORK ISSUING THIS BRS WILL BE INTERRUPTED AFTER THE DELAY IF INTERRUPT NUMBER 5 IS ARMED AT THAT TIME. INTERRUPT WILL BE TERMINATED. THE BRS 78 DISARMS THE BRS 135. ALSO SEE THE BRS 81.

REGISTERS AFFECTED: NONE

BRS 136

DATE: 71/09/22

FUNCTION: SETS SYSTEM EXEC SWITCHES IN SYMS

STATUS: EXEC

CALLING SEQUENCE: LDA V
LDX N
BRS 136
NORMAL RETURN

V = NEW SWITCH VALUE

N = SWITCH NUMBER

DESCRIPTION: THE SWITCH IS SET TO THE NEW VALUE AND THE OLD VALUE IS RETURNED IN A. TABLE OF SWITCHES:

X	Description of Switch
0	SWEX. Exec Switch
1	SWLET. Letter Routine Switch
2	PACST.
3	Zero Cell. Not Used.
4	ACA. Accounting Area
5	DOWN. No longer referenced by Monitor or Exec.
6	FATIG. Minned by Monitor if REAL is about to overflow
7	SWMBL. Mail Box List Busy
8	SWMBM. Mail Box Messages Busy
9	ENMSK. Subsystem Enable Mask. No longer referenced by Monitor or Exec
10	SWFD. File Directory Busy for user specified.

REGISTERS AFFECTED: A

BRS 137

DATE: 71/07/23

Name: List [port numbers] in memory

Status: User

Input: X = Address in user's page where table is to be put.

Output: A = Length of list

B = Preserved

X = Preserved

Description: Lists his command port (0) in the head of the list. Auxiliary ports are then appended to the list.

Instruction Traps: None

BRS 138

DATE: 71/07/23

Name: [Supervisor] Request

Status: Exec

Input: None

Output: A = Error Number (If any)

B = Preserved

X = Auxiliary [Port] number

Description: Places a request on the Supervisor (See Appendix H) to get an [auxiliary circuit]. The monitor dismisses the user until the Supervisor sets up a circuit to the user's program. The user cannot shift control 0 out of this condition until the BRS 138 returns. The user then outputs to the teletype (File number 1), a login string. This string must be terminated by a dash.

Supervisor Error Table

C(A) = 1 = Format Error

2 = Bad User Name

3 = Bad MUD

4 = System Unavailable

Example: To log in under the user name "FALSENAME" on computer 25:

```
LOGIN BRS 138; SKE =0; HLT (Impossible); LDP LOGPTR
```

```
LDX =1; BRS 35; BRS 138; SKE =0; BRU ERROR
```

```
STX IF; EAX 1,2; STX OF; ...
```

```
LOGPTR DATA 3*LOGSTR-1,3*LOGSTR+12
```

```
LOGSTR ASC 'FALSENAME:25-'
```

This string is sent to the Supervisor which attempts to find this user in the [MUD]. Another BRS 138 is then done and will return an error code in A. If A is zero, then no errors have occurred and the auxiliary port number can be found in X. Notice: The port number in X is the input port. The output port is found by adding 1 to X. The port number in X can now be used by the character SYSPOPS and BRS's (CIO, CIT, BRS 35, 33 etc.)

Instruction Traps: Insufficient status

BRS 139

DATE: 71/07/27

Name: [Zap] an [auxiliary circuit]

Status: User

Input: X = [Input port] number (As returned by BRS 138)

Output: A = Preserved

B = Preserved

X = Preserved

Description: Zaps the auxiliary circuit (Removes it from Tymnet).
Instruction Trap: Invalid port number in X.

BRS 140

DATE: 71/03/24

FUNCTION: READS THE DATE INCLUDING THE YEAR INTO THE USERS MEMORY

STATUS: USER

INPUT: A=BEGINNING STRING POINTER

B=ENDING STRING POINTER

DESCRIPTION: THE CURRENT DATE AND TIME ARE APPENDED TO THE STRING PROVIDED IN A AND B REGISTERS AND THE RESULTING STRING POINTERS ARE RETURNED IN THE A AND B REGISTERS. THE CHARACTERS APPENDED TO THE STRING HAVE THE FORM:

MM/DD/YY HH:MM

MM=MONTH

DD=DAY

YY=YEAR

HH=HOURS COUNTED FROM 0 TO 24

MM=MINUTES.

SEE BRS 91

REGISTERS AFFECTED: B

BRS 141

DATE: 71/09/22

FUNCTION: GETS EXEC SUBROUTINES

STATUS: EXEC

DESCRIPTION: CHECKS THAT THE ISSUING FORK HAS EXEC STATUS. IF IT DOES, AN EXEC BRS IS ISSUED. THIS BRS IS USED TO ALLOW EXEC STATUS FORKS TO ACCESS SUBROUTINES IN THE EXEC. TABLE:

X	Description of Routine
0	Turns Letter on for All current users.
1	Super GFD. A = File Directory Number
2	Set switch to force Exit if negative. A = Switch Value
3	Counts number of users on system. Returns A = NU
4	Not Used.
5	Checks for User on system. A = User number
6	Checks for Operator Timeout. Skips if Still Pending
7	Set File Group Busy. A = File Directory Group Number
8	Reads last Ercode.
9	Changes Index Block Pointer with new pointer in A. Deletes old.
10	Writes Accounting
11	Change EXFLAG.
12	Set File Parameters from A and Date in B.

REGISTERS AFFECTED: NONE

BRS 142

DATE: 71/03/25

FUNCTION: ALLOWS USER EXECUTION OF EXEC COMMAND

STATUS: USER

INPUT: A = FIRST 3 CHARACTERS OF EXEC COMMAND OR SUB-SYS. NAME (OR)

A = SAME AS ABOVE WITH SIGN BIT (RETURN TO BRS)

NO OUTPUT

RETURN: NO RETURN WITH FIRST OPTION (NO SIGN BIT)

RETURNS: NO SKIP: ILLEGAL COMMAND TO RETURN

SKIP: COMMAND EXECUTED

DESCRIPTION: GENERAL

THE FIRST 3 CHARACTERS OF THE COMMAND OR SUB-SYSTEM NAME ARE PLACED IN THE "A" REGISTER. IT ALWAYS APPEARS TO THE EXEC THAT THE COMMAND WAS TERMINATED BY A CARRIAGE-RETURN. COMMAND ARGUMENTS ARE TAKEN FROM THE TELETYPE OR THE COMMANDS-FROM FILE. THE USUAL PROMPTING MESSAGES ARE TYPED.

DESCRIPTION: NO SIGN BIT IN "A"

ALL COMMANDS ARE LEGAL. THE EXEC DOES A RESET BEFORE EXECUTING THE COMMAND. THE COMMAND IS EXECUTED AS IF IT WAS TYPED IN THE EXEC TERMINATED BY A CARRIAGE RETURN.

DESCRIPTION: SIGN BIT IN "A"

ONLY COMMANDS THAT DO NOT NORMALLY DO A RESET ARE LEGAL. THE ILLEGAL COMMANDS (SUCH AS LOGOUT, RESET, RUN, CALL SUB-SYSTEM, ETC.) WILL CAUSE THE ERROR RETURN (NO SKIP) TO BE TAKEN. "ESCAPES" DURING THE EXECUTION OF THE PROGRAM WILL USUALLY CAUSE A RETURN TO THE BRS WITH A SKIP.

BRS 143

DATE: 70/07/06

FUNCTION: SKIP IF BIT MAP SET

STATUS: SYSTEM

DESCRIPTION: SKIPS IF SDBM8 IS ZERO (BIT MAP SET).

RETURNS: SKIP RETURN = BIT MAP SET

NO SKIP = BIT MAP NOT SET

REGISTERS AFFECTED: NONE

BRS 144

DATE: 69/05/13

FUNCTION: GETS A BUFFER

STATUS: EXEC

OUTPUT: A = ADDRESS OF DATA AREA IN BUFFER.

RETURNS: NO SKIP: NO FREE BUFFERS

SKIP: NORMAL RETURN

BRS 145

DATE: 69/05/13

FUNCTION: RETURNS A BUFFER

STATUS: EXEC

INPUT: A = ADDRESS OF DATA AREA IN BUFFER TO BE RETURNED

DESCRIPTION: RETURNS THE BUFFER TO THE MONITOR.

REGISTERS AFFECTED: NONE

BRS 146

DATE: 71/03/11

FUNCTION: READS NUMBER OF QUEUE ENTRIES

STATUS: SYSTEM

OUTPUT: A=NUMBER OF JOBS ON QTI, B=NUMBER OF JOBS ON QIO,
X=NUMBER OF JOBS ON QQE

DESCRIPTION: THIS BRS WILL BE CHANGED SOON SO THAT THE SIGN
BIT OF A WILL BE ONE IF THERE IS A JOB ON QQC.

REGISTERS AFFECTED: ALL

BRS 147

DATE: 69/05/13

FUNCTION: [CLOSE]S ALL EXCEPT COMMANDS-FROM [FILE]

STATUS: USER

REGISTERS AFFECTED: NONE

BRS 148

DATE: 70/07/06

FUNCTION: NEGATE FLOATING ACCUMULATOR

STATUS: USER

DESCRIPTION: THE CURRENT CONTENTS OF THE FLOATING ACCUMULATOR ARE
NEGATED. OVERFLOW WILL CAUSE THE OVERFLOW INDICATOR TO BE SET.
IF FLOATING POINT HARDWARE IS NOT IMPLEMENTED, THE SIMULATED ACC-
UMULATORS WILL BE NEGATED.

REGISTERS AFFECTED: NONE

BRS 149

DATE: 70/07/06

FUNCTION: FIX FLOATING ACCUMULATOR AND STORE IN A & B

STATUS: USER

DESCRIPTION: FIX THE FLOATING ACCUMULATORS AND PLACE IN A AND B.
THE NUMBER IN THE FLOATING ACCUMULATOR IS CONVERTED TO A INTEGER
AND STORED IN THE A REGISTER.

REGISTERS AFFECTED: A

BRS 150

DATE: 70/07/06

FUNCTION: FLOAT A & B AND PLACE IN THE FLOATING ACCUMULATOR

STATUS: USER

DESCRIPTION: THE NUMBER IN A AND B IS CONVERTED TO A FLOATING POINT
NUMBER AND STORED IN THE FLOATING ACCUMULATOR.

REGISTERS AFFECTED: B

BRS 151

DATE: 71/03/25

FUNCTION: CHANGE COMMANDS-FROM FILE

STATUS: USER

INPUT: A = FILE NUMBER OF COMMANDS-FROM FILE.

DESCRIPTION: CLOSSES THE COMMANDS-FROM FILE IF ONE IS OPEN AND

SETS THE COMMANDS-FROM FILE TO THE FILE SPECIFIED IN A.
THIS BRS WORKS IN CONJUNCTION WITH THE EXEC "COMMANDS" COMMAND WHICH
OPENS A FILE FOR INPUT AND CAUSES ALL EXEC INPUT TO COME FROM THIS
FILE. THE FILE NUMBER IS PASSED TO SUB-SYSTEMS AND GOTO PROGRAMS
IN THE "A" REGISTER. SEE ALL BRS 77 FOR TELETYPE-OUTPUT CONTROL.
REGISTERS AFFECTED: NONE SAVED

BRS 153

DATE: 70/07/27
Name: Set cell 211
Status: User
Input: A = Page 0 of [Supervisor]
Output: A = Preserved
B = Preserved
X = Preserved
Description: Sets cell [211B] to page zero of the Supervisor. This can
only be done by the Supervisor.
Instruction Traps: User not Supervisor, 211 not zero.

BRS 154

DATE: 70/07/27
Name: Reset cell 211
Status: User
Input: None
Output: A = Preserved
B = Preserved
X = Preserved
Description: Sets [211B] to 0 and unlocks the page.
Instruction Traps: User not [Supervisor] or 211 already zero.

BRS 155

DATE: 70/07/27
Name: Test [port] for characters
Status: User
Input: X = port number
Output:
Skip return: A = Preserved
B = Preserved
X = Preserved
Return: A = Preserved
B = Preserved
X = Preserved
Description: The BRS will skip:
1) If the port is an [input port], if there are any characters in
the input buffer.
2) If the port is an [output port], if a TCO would cause a dismiss.
Instruction Trap: Illegal port number.

BRS 156

DATE: 70/07/27

Name: [Zap] all [Auxiliary circuitS]

Status: User

Input: None

Output: A = Preserved

B = Preserved

X = Preserved

Description: Zaps (See Appendix H) all auxiliary circuits this job.

Instruction Traps: None

BRS 157

DATE: 70/07/27

Name: Copy [Level 3] [Index Block] to memory

Status: System

Input: A = File number

X = Address in user's memory

Output: A = Preserved

B = Preserved

X = Preserved

Description: Copies the current level 3 index block from the file whose file number in in A to user's memory.

Instruction Traps: Insufficient status, file not opened, unrecoverable error.

BRS 158

DATE: 71/07/27

Name: Charge [royalty units]

Status: User

Input: A = number of units to charge

Output: Skip return: A = Preserved

B = Preserved

X = Preserved

Return: A = Preserved

B = Preserved

X = Preserved

Description: Adds the number of units in A to the current royalty charge. If an overflow occurs or a negative number is given, then non-skip return will be taken. 1 Royalty unit = \$.001

Instruction Traps: None

BRS 159

DATE: 71/07/27

Name: [Answer] lines

Status: Exec

Input: A = 0 to "shut" the system (close all lines)

= 1 to "up" the system (answer all lines)

Output: A = Preserved

B = Preserved

X = Preserved

Description: Answers or shut lines to system.

Instruction Traps: Insufficient status.

BRS 160

DATE: 71/07/27

Name: Change [terminal characteristics]

Status: User

Input: A = New terminal characteristics (see BRS 132 for list of characteristics)

X = Port number (-1 for command port)

Output: A = Preserved

B = Preserved

X = Preserved

Description: Changes the characteristics of the terminal. A one in a bit position is ON, a zero is OFF.

Instruction Traps: Bad port number (if interrupt 10 is not armed).

BRS 161

DATE: 71/07/27

Name: Ignore Panics

Status: User

Input: X = Port number (-1 for command port)

Output: A = Preserved

B = Preserved

X = Preserved

Description: Disables escapes for the port in X. Panics are not stacked.

Instruction Traps: Bad port number (if interrupt 10 is not armed).

BRS 162

DATE: 71/07/27

Name: Enable Panics

Status: User

Input: X = Port number (-1 for command port)

Output: A = Preserved

B = Preserved

X = Preserved

Description: Reset escape control set by BRS 161.

Instruction Traps: Bad port number (if interrupt 10 is not armed).

BRS 163

DATE: 71/07/27

Name: Copy Net to User

Status: System

Input: None

Output: A = Preserved

B = Preserved

X = Preserved

Description: Copies TYMNET (page 12) to users memory starting at 34000B.

Instruction Traps: Insufficient status

BRS 164

DATE: 71/07/27

Name: Dismiss until [output buffer] empty and/or activate user after a period of time.

Status: User

Input: A= Number of millisecond to activate at if [Yellow Ball] doesn't return.

X= Port number (-1 for command port)

Output: A= Preserved

B= Preserved

X= Preserved

Description: Sends a yellow ball and waits for it to return, signifying that the output buffer is empty. If the yellow ball does not return after A milliseconds, then the user is again activated. This BRS is recommended over the BRS 14.

Instruction traps: Bad port number (if, interrupt 10 is not armed).

BRS 166

DATE: 71/09/10

Name: Hang port 0

Status: Exec

Input: A= Same as input

B= Same as input

X= Same as input

Description: Hangs port 0 (zero). The BRS 129 will not allow port 0 to be hung.

Instruction traps: Insufficient status.

BRS 167

DATE: 71.09.21

Name: Read and reset disk error flag

Status: User

Input: None

Skip return: A= Preserved

(success) B= Preserved

X= Preserved

return: A= Preserved
(failure) B= Preserved
X= Preserved

Description: Tests for a disc error caused by a BRS 122, 123, 124, or 125. The BRS will skip if there were errors since the last time the BRS was issued.

Instruction traps: None

BRS 168

DATE: 71/09/16

Name: Disable [shift control O]

Status: User

Input: X= Port number (-1, 0 or 1 for command port)

Output: A= Preserved

B= Preserved

X= Preserved

Description: Disables shift control O's (oh), but allows panics to still be active.

Instruction traps: Illegal port number in X (takes interrupt 10 if armed).

BRS 169

DATE: 71/09/16

Name: Re-arm [shift control O]

Status: User

Input: X= Port number (-1, 0 or 1 for command port)

Output: A= Preserved

B= Preserved

X= Preserved

Description: Re-arms shift control O's. They will now clear forks to the Exec.

Instruction traps: None

BRS 170

DATE: 71/09/21

Name: Set limit for [premium charge] program charges

Status: User

Input: None

Output: A= Preserved

B= Preserved

X= Preserved

Description: Calls on the Exec to set the maximum premium charge. When the BRS is done, the Exec will type "Please Type Maximum Royalty Program Charges You Will Permit:". At that time, the user must input the maximum number of charge units. One royalty unit equals \$0.001. There is a maximum limit of \$16,777 of Royalty charges per logon. This is in addition to Tymshare's resource charging (CRU and Connect).

NOTE: A premium charge program cannot charge until it does BRS 170.

Instruction traps: None

BRS 171

DATE: 71/09/21

Name: Identify [premium charge] program

Status: User

Input: A= Premium charge program number

Output: A= Preserved

B= Preserved

X= Preserved

Description: Each premium charge program is assigned an identity number. When the Exec starts the program, the ident is written to the accounting. Tymshare will pay the author of the program when the bills sent by Tymshare have been paid.

Instruction traps: None

BRS 172

DATE: 71/09/21

Name: Send control information to remote

Status: User

Input: A= Special character

X= Port number (-1, 0 or 1 for command port)

Output: A= Preserved

B= Preserved

X= Preserved

Description: Sends type 1 record through [Tymnet] to remote. (See Appendix H). The A register contains a special character which can set terminal characteristics etc.

Instruction traps: Bad port number in X (takes interrupt 10 if armed).

SECTION 6.0 APPENDIX A

BRS 7 TABLES

DATE: 71/09/17

Description of the Miscellaneous table as typed by 1COUNT:

Pwfl Power Failures
 Mapnum Map Number
 Dbits Number of Disc blocks remaining
 RS Number of RAD channel errors
 Tnictr Number of teletype "on" interrupts
 Tficttr Number of teletype "off" interrupts
 Spare1 First Monitor Spare
 Spare2 Second Monitor Spare
 Tiictr Number of teletype input interrupts
 Tiibad Bad channel on input interrupt
 Toictr Number of teletype output interrupts
 Toibad Bad channel output interrupt
 Nu Number of teletypes on
 Utime Number of user on seconds
 Real Number of clock ticks
 Wettb Time not used to speed up disk I/O
 Settb Unsold time (Time used by system)
 Stime Time in system mode
 Spt Page tick for all users
 Pages Pages swapped
 Sswc Swap count for all users
 Rpage Pages read
 Pagew Pages written
 Sdskmt Disk access units for all users
 Arnot Arm Motion
 XBerr Index block errors while setting map
 FDerr File Directory errors while setting map
 Extpu Extra Phantom User entries at BRS 112
 NB112 Number of logouts
 Rber Rubout error on Phantom User
 UpShut Contains 1 if "shut", 2 if "up".
 Chrdy Cte channel not ready
 Lcdfib Lost characters due to full input buffer
 Eqpt Equipment cell

The symbols D0 through D5 are counters used for experimentation only. Eqpt has the following significance:

8 = Dual Data Products Disc
 4 = 2314
 2 = Phase II (Tymnet)
 1 = Floating Point Hardware

Description of the RAD error table:

If a write error has occurred, the sign bit will be set. The monitor

error table RAERL has space for recording 20D RAD errors. A RAD error is one of two types. If the low order 5 bits are equal to 35B, the error is a channel error (RC), otherwise the error word is the first half of an addressing error (RA). If the error is an addressing error, the first word is the address the monitor attempted. The second word is the address the monitor got. If the error is a channel error, the error word will be the POT word, and hence no conversion is needed.

Format of the RAD addressing error word:

KNN NNN NNN NRR RRR UUW WWW WWW

Where:

K = Read/Write key. 1 for Write, 0 for Read
 N = Not used
 R = High order 5 bits of POT word
 U = User Data
 W = Write Count

Description of the Disc Error table

If the Disc is a IBM 2314:

SSS PPP CCC CCC CCH HHH HRR RDD

Where:

S = Checksum (First 3 bits only)
 P = Pack Number
 C = Cylinder
 H = Head
 R = Record
 D = Bits used for data chaining

If the disc is a Data Products:

NNN NND DDD DDT TTT TTT TSS SSS

Where:

N = Not used
 D = Physical disc number
 T = Logical track pair
 S = Sector

The monitor has space for 10D Disc errors.

Description of QTIGO:

Each counter counts the number of swaps per activation condition as follows:

1 Dismiss until more room on the ring (TROSW)
 2 Not Used.
 3 Teletype early warning (Output buffer)
 4 Special test (Wait for tape clear)
 5 Not used.
 6 Input buffer, BRS 81, BRS 45 or BRS 164 Dismissal
 7 Special fork not on queues.
 Address:
 0 Dead
 1 Running
 2 BRS 31. Dismiss until lower fork panics
 3 Uncorrectable RAD errors
 4 Exec BRS
 5 BRS 109. Dismiss until off interrupt
 6 BRS 9
 10 Disc use/W Buffer
 11 Not used.
 12 Wait for unit to clear
 13 Not used.
 14 Exec BRS
 15 BRS 9
 16 Quantum Overflow
 17 Not Used.
 20 Exec level panic
 21 Ordinary panic
 22 3 sec w Buffer time out/Disc errors
 23 BRS 111
 24 BRS 44
 25 Count for BRS 44
 26 Count for RFK
 27 Compute Job count
 30 PU Count
 31 Not Used.
 32 Supervisor Request (Type 5 Out)
 33 Tape read and Printer
 34 Not Used.
 35 TRAPT

Description of Multiplexer Communication Words:

The base communicates with the 940 through 6 cells following 210B. These cells contain the following:

210B ALARM (See description below)
 211B Supervisor's page zero. When the Supervisor is running, the Supervisor does a BRS 153 to lock his page zero in core. The monitor then places the pages's RMT number in 211B for the base to put data in. Cell 211B is non zero both when the Supervisor is running and when the Supervisor is awake and trying to take over the net. The Supervisor does a BRS 154 to release his page zero when he goes to sleep.
 212B+ UP flags. The base clobbers these flags every once in a while
 213B and checks them a few clock ticks later. If the 940 has not

reset the flags, the base presumes the 940 is down and sends a message to the Supervisor telling it so. The keys is 40516337B and 51506262B.

- 214B This cell contains the BRM to the clock interrupt routine in the base.
 215B Cell 215B contains a BRM MONCR.

Description of the Global File Flags:

Both the Global flag and the Index Block Pointer array are indexed by a global file number. This global file number is not to be confused with the local file number associated with each user. Each file open on the system is assigned a global file number and a global index block pointer. The index into one table is the same as the other.

DDD DDD DFF IJJ JJJ JPP PPP PPP

Where:

- D = Initalized to -1. Incremented for each Disc job started, Decremented by the interrupt routine
 F = Local File number - 3
 I = Error detected in IDM
 J = Job number
 P = Privelege Flags for file. (High Order 8 bits only). These are the random file flags.

Description of the Job to Teletype Conversion table:

This table is referenced by a job number which will return the following parameters:

DDD DDD DNN NNN TRP EEE CCC CCC

Where:

- D = Disc or W buffer busy
 N = Not Used.
 T = Termination pending
 R = User Rubout
 P = PUTIM busy. (Time out interrupt going)
 E = RAD errors
 C = Teletype number

Description of ALARM:

ALARM is location 210B in the 940 monitor. Whenever a bit is set, the Varian 620/I will howl until that bit is reset. The following is a list of bits currently in use:

- Bit 23: Real about to overflow
 Bit 22: Disc map almost/completely filled.

Description of the Global Index Block Pointers:

Each file has a index block associated with it. When that file is opened, it's index block pointer is placed in this array. This index block pointer is not the same as the one in the file directory however. The index block pointer in the file directory is shifted to the right two (2) places. Therefore, to convert from a file index block pointer, the program must do a LSH 2 on the file directory index block pointer.

Description of DSCTOP:

DSCTOP is the highest disc address that any BRS 122, 123, 124 or 125 can access.

Description of MACH:

MACH is the machine number. Machine numbers have the following format: (Location) (Machine number) . The locations can have the following letters:

C	Cupertino, California
N	Englewood, New Jersey
P	Paris, France (CEGOS)

Description of STACT:

STACT is a two word array that contains the beginning and current cursor of the Raw Accounting file.

Description of the WERIS table: (Port)

With the exception of auxiliary ports, the WERIS table contains the LUD locators for each port. These locators are of use when looking up information such as User Names and Account numbers.

Description of the PMTP table: (Job)

The PMTP table is a list of the locations in the monitor where each job has it's "Pseudo Memory Table". At the present time, each job has 35 octal entries in the table. The first entry in the table is the TS page, followed by page 44B and so forth. The PMTP table points to the PMT table for each job. Each PMT entry has the following format:

SER PPP PPP AAA AAA AAA ACC CCC

Where:

S	= The page is or has been shared (Via BRS 3)
E	= An exec page. This requires Exec status to relabel over.
R	= Read Only. Set to 1 if Read Only
A	= RAD Address
C	= Code Page (See RMT)

Description of the CPARW table: (Job)

This table contains the Control Parameters for each job. for a complete description of the bit fields, see the BRS 39.

Description of the AUNN table: (Job)

This table contains the universal user number for each job. This is an antiquated table, with the exception of looking up information on Auxillary Ports.

Description of the FGLIST table: (Job)

This table contains the last file directory accessed for each job. This is sometimes reset to zero by the Exec.

Description of the Miscellaneous Monitor Table (Table no.)

This table contains many miscellaneous monitor switches and other interesting data. The following is a list of the counters in sequential order:

ADRSMT Address of SMT in the monitor
 AEXSMT Address of EXSMT in the monitor
 M48K -1 for 48K, 0 for 64K of memory
 MaxNU Maximum Job Limit
 SDBM8 -1 if Bit Map not set, 0 if set
 MapBit A bit which is redundant to MapNum
 DRC Constant used by monitor to locate Exec on the disc
 SW16 -1 if 16 Disc system, 0 if 32 or greater disc system
 RWD 0 if Disc Write Protected
 RWR 0 if RAD Write Protected
 SW205 0 if 205 interrupt not high
 SWXMA Used by 205 interrupt
 Mach Machine Number (See Mach above)
 DiscN Disc number of currently running system
 Vers Versiion of the monitor (2 Words)
 FG940A First Word of Up flags for Base
 FG940B Second Word of Up Flags for Base

Description of RAFR:

RAFR is a 32 word array that contains the RAD bit map. If a bit is on, the page is available for use.

Description of the ETTB table: (Job)

ETTB contains the pure compute time for each job in clock ticks.

Description of the PTCKS table: (Job)

PTCKS contains the amount of pages in memory per clock tick accumulated in the current logon.

Description of CCT (Job)

CCT contains the number of characters input and output for each job.

Description of the TTYASG table: (Port)

TTYASG contains the PACPTR for each job. From the PACPTR, the relabeling and other information can be found.

Description of Q0:

Q0 is the beginning of the monitor queue tables. Each of the 5 queues (QTI, QIO, QSQ, QQC and QQE) have a 4 word array associated with it. The four word arrays have the following format:

First Word: If a job is on the queue, the first word contains the pacptr of the next job to run, otherwise, it contains the address of the table.

Second Word: Contains the PACPTR of the first job on the Queue

Third Word: Contains the address of the Queue Table

Fourth Word: Contains the number of user on the Queue

Description of the MONTAB table:

Montab contains addresses the are only used for peeking. (Via BRS 127). The following is a list of current addresses:

PNEXT Added to PACPTR to get next queue number

PPTR Added to PACPTR to get previous PACPTR if in a chain.

SECTION 7.0 APPENDIX B

RANDOM FILES

DATE: 71/05/07

DEFINITIONS:

CP=CURSOR POSITION. THE ADDRESS OF THE NEXT CHARACTER, WORD OR BLOCK TO BE READ OR WRITTEN. THE LOWEST POSSIBLE CHARACTER ADDRESS IS 0. THE HIGHEST IS 14B6-1.

CPTOP=ADDRESS OF HIGHEST LOCATION WRITTEN.

OPENING RANDOM FILES:

USE NORMAL OPEN FILE BRS'S. THE HIGH ORDER BITS OF THE A REGISTER IN BRS 63 & 65 CONTAIN THE RANDOM FILE PRIVILEGE BITS. PRIVILEGE

NUMBER	BIT	OPTION
P0	0	0
P1	1	0
P2	2	USER MAY EFFECTIVELY EXECUTE THE SCP SYSPOP.
P3	3	0
P4	4	0
P5	5	USER MAY ERASE INFORMATION FROM A FILE. (OUTPUT ONLY)
P6	6	USER MAY SET A FILE TO WRITE MODE. (OUTPUT ONLY)
P7	7	USER MAY SET A FILE TO READ MODE.

IF ALL BITS ARE 0, THE SYSTEM WILL ALLOW ALL FOUR PRIVILEGES FOR NORMAL FILES.

THIS IS TO ALLOW COMPATABILITY WITH PROGRAMS WRITTEN BEFORE RANDOM FILES. FOR READ ONLY FILES THE BRS WILL GIVE AN ERROR RETURN. FOR APPEND ONLY FILES ALL FOUR PRIVILEGES ARE DENIED AND THE CURSOR IS POSITIONED AT THE END OF THE FILE (CPTOP.)

OPENING FILES FOR INPUT WITH BRS 62 & 64 ALLOW READING AND SETTING OF CURSOR POSITIONS BUT PROHIBIT WRITING OR ERASING.

CLOSING RANDOM FILES:

USE NORMAL CLOSE FILE BRS'S. IF THE FILE IS CLOSED WHEN IN READ MODE, ALL THE FILE INFORMATION, INCLUDING UPDATES AND ERASURES, IS SAVED.

FILE SYSPOPS:

WIO - READ OR WRITE ONE WORD ON A FILE.
 CIO - READ OR WRITE ONE CHARACTER ON A FILE.
 BIO - READ OR WRITE A BLOCK OF WORDS ON A FILE.
 RCP - READ CURSOR POSITION
 SCP - SET CURSOR POSITION.
 PCE - POSITION CURSOR AND ERASE.
 SSP - SET PHYSICAL SIZE LIMIT.
 RSP - READ PHYSICAL SIZE, SIZE LIMIT AND ADDRESS OF HIGHEST LOCATION WRITTEN.

FILE SIZE RESTRICTIONS:

FILE DATA IS STORED ON THE DISC IN 1400B CHARACTER BLOCKS AND THE CURSOR POSITION ASSOCIATED WITH THE BEGINNING OF ANY DATA BLOCK

IS A MULTIPLE OF 1400B. THE MONITOR DOES NOT WRITE DATA BLOCKS THAT ARE ALL ZERO; HENCE, A FILE THAT COVERS A LARGE RANGE OF CURSOR POSITIONS MAY GRAB LESS DISC SPACE THAN A SMALL FILE THAT IS DENSELY PACKED WITH INFORMATION. EACH FILE HAS A MINIMUM OVERHEAD OF 1400B CHARACTERS TO MAP EACH FOUR PAGES OF NON-ZERO FILE DATA. THE DETAILS OF THE MAPPING OVERHEAD ARE COMPLICATED AND ARE A FUNCTION OF THE DATA.

EACH FILE THAT IS OPENED FOR OUTPUT IS ASSIGNED A PHYSICAL FILE SIZE QUANTUM TO LIMIT THE AMOUNT OF ADDITIONAL DISC SPACE THAT THE USER MAY GRAB BEFORE CLOSING THE FILE. THIS QUANTUM, 1,364,000B CHARACTERS, IS REFRESHED EVERY TIME THE FILE IS OPENED UNLESS THE FILE SIZE INCLUDING OVERHEAD IS 3,145,727 CHAR. 1,364,000B CHARACTERS ARE ABOUT EQUAL TO ABOUT 60D PAGES OF DATA. IF A FILE QUANTUM IS EXCEEDED, AN INSTRUCTION TRAP IS CAUSED.

FILE FLAGS:

WHEN ONE OF THE FOLLOWING CONDITIONS OCCUR, THE HIGH ORDER BITS OF THE FILE NUMBER ARE SET TO THE FOLLOWING:

CONDITION	FILE FLAG
EOF	402B5
ERROR	404B5
QUANTUM OVERFLOW	410B5

SECTION 8.0 APPENDIX C

OP CODE FLAGS FOR TAPE BRS'S

DATE: 71/09/16

OP CODE FLAGS FOR BRS'S 102, 103, 105 AND 106:

- 00= THE FOLLOWING RECORD WAS READ OR WRITTEN CORRECTLY.
- 01= FOLLOWING RECORD WAS IN ERROR. NO FURTHER TAPE COMMANDS WERE ISSUED.
- 02= END OF FILE. NO MORE READING. THE PREVIOUS RECORD WAS THE LAST.
- 03= LAST TAPE RECORD DUE TO LACK OF CORE SPACE.
NO MORE RECORDS. FOLLOWING RECORD MAY HAVE BEEN TRUNCATED.
- 04= FOLLOWING RECORD WAS TRUNCATED DUE TO OVER RUN OF 3 SEC. TIME LIMIT.
- 05= NO MORE RECORDS DUE TO REACHING SPECIFIED RECORD COUNT.
THE FOLLOWING RECORD IS THE LAST.
- 06= END OF TAPE. NO MORE READING. THE PREVIOUS RECORD WAS THE LAST.
- 07= PRINTER NOT READY OR TAPE NOT READY (NO MORE READING)
- 10= PAGE BOUNDARY ERROR ON WRITE.
- 11= TAPE WRITE PROTECTED
- 12= BEGINNING OF TAPE. NO WRITE.
- 13= DEVICE NO LONGER ASSIGNED.
- 14= "PANIC"

SECTION 9.0 APPENDIX D

EXECUTIVE ERROR CODES

DATE: 71/10/01

ERROR DESCRIPTION OF ERROR
NO.

- 0 NO PREVIOUS ERRORS REPORTED
- 1 FILE BUSY - SOMEONE ELSE PROBABLY HAS THE FILE OPEN
- 2 TOO MANY FILES OPEN
- 3 FILE CANNOT BE WRITTEN NOW. DISK SPACE BEING RECLAIMED.
- 4 INPUT-OUTPUT ERROR RECEIVED FROM DISC STORAGE UNIT
- 5 FILE CANNOT BE WRITTEN NOW. DISK SPACE BEING RECLAIMED.
- 9 SET CURSER POSITION ERROR ON APPEND ONLY FILE
- 10 FILE NAME NOT IN FILE DIRECTORY
- 11 ILLEGAL POSITION IN FILE NAME OF A FILE NAME TERMINATING CHARACTER (GENERALLY BLANK, COMMA, OR SEMI-COLON) AS FIRST CHARACTER.
- 12 UNEXPECTED END OF INPUT FILE NAME STRING.
- 13 FILE NAME STRING POINTERS INVALID OR INDICATE MORE THAN 75 CHARACTERS.
- 14 NO FILE NAME PRECEDING COMMENT PORTION (DASH)
- 15 END OF FILE NAME STRING DISCOVERED IN PROTECTED PORTION OF STRING SUCH AS BETWEEN SLASHES
- 16 CARRIAGE-RETURN FOUND IN PROTECTED PORTION OF FILE NAME STRING
- 17 LINE-FEED FOUND IN PROTECTED PORTION OF FILE NAME STRING
- 18 ADDITIONAL CHARACTER FOUND AFTER SECOND FULL QUOTE OF LIBRARY FILE NAME
- 19 ILLEGAL CHARACTER IN UNPROTECTED PORTION OF FILE NAME USUALLY A PUNCTUATION MARK
- 20 ILLEGAL CHARACTER FOUND IMMEDIATELY AFTER RIGHT-PARENTHESIS ON PUBLIC FILE.
- 21 STRING ENDS ON PUBLIC FILE RIGHT PARENTHESIS
- 22 FILE NAME AND COMMENT TOTAL MORE THAN 75 CHARACTERS.
- 23 INVALID USER NAME USED WHEN ADDRESSING A PUBLIC FILE
- 24 PROTECTED FILE (DECLARED NOT READABLE OR WRITABLE)
- 25 OLD FILE OR NEW FILE MESSAGE NOT CONFIRMED WITH CARRIAGE-RETURN OR LINE-FEED
- 26 ATTEMPT TO ADDRESS LIBRARY FILE FOR OUTPUT
- 27 INVALID PARAMETER FOR BRS 16,19, ETC. (A REGISTER)
- 28 FILE TYPE WRONG (USUALLY FOR TELETYPE)
- 29 INVALID OR BAD FILE FOR OUTPUT (CHECK WITH DIRECTORY COMMAND)
- 30 FILE DIRECTORY GROUP BECAME BUSY DURING OPERATION, RETRY
- 31 GARBAGE GROUP POINTER IN FILE DIRECTORY.
- 32 NUMBER OF FILES QUOTA EXCEEDED
BY LOGGING OUT AND BACK IN, YOU WILL RECEIVE A NEW QUOTA.
- 33 ATTEMPT TO OPEN "INIT" FILE FOR OUTPUT.
- 34 FILE DIRECTORY GROUP CONFLICT BETWEEN TWO USERS, RETRY.
- 35 INVALID PARAMETER FOR BRS 16,19, ETC. (A REGISTER)
- 36 ATTEMPT TO OPEN PROPRIETARY FILE FOR INPUT.
- 37 FILE NOT PRIVATE WRITABLE

38 FILE NOT PUBLIC WRITABLE
39 FILE NOT PRIVATE READABLE
40 NO INDEX BLOCK POINTER (NO DATA) IN FILE DIRECTORY
41 INVALID OR BAD FILE FOR INPUT (CHECK WITH DIRECTORY
COMMAND)
42 LIBRARY FILE NAME USED ILLEGALLY
43 YOU USED THE PUBLIC FILE OPTION ON A FILE THAT IS NOT PUBLIC
44 UNEXPECTED ERROR FROM MONITOR ON FILE SYSPOPS
45 FILE DECLARED READ ONLY
46 GARBAGE IN FILE DIRECTORY
47 ATTEMPT TO ADDRESS A FILE ACROSS ACCOUNT BOUNDARIES ILLEGALLY
48 FILE DIRECTORY NOT ACCOUNT SHARABLE
49 INVALID PARAMETER FOR BRS 16,19, ETC. (A REGISTER)
50 FILE DIRECTORY UNASSIGNED. THE F.D. GROUP IS ALL ZEROS.
51 END OF F.D. CHAIN WITHOUT FINDING FILE.
52 FILE DIRECTORY GROUP BEING CHANGED BY ANOTHER USER.
53 DUMP FILE MAY NOT BE CREATED BY USER PROGRAM.
54 NO FILE TYPE SPECIFIED FOR OUTPUT FILE.
MUST BE 1 TO 3
61 FILE TYPE NOT "GO"
62 ATTEMPT TO CALL PREMIUM CHARGE PROGRAM ILLEGALLY
63 INVALID CONFIRMING CHARACTER
64 EXEC COMMAND USED WITHOUT PROPER STATUS
65 INIT FLAG ATTACHED TO FILE DIR. BUT NOT TO ANY FILE.
RELEASE FLAG BY USING DEINIT COMMAND.
67 SUB-SYSTEM NOT CURRENTLY AVAILABLE.
68 NO SUB-SYSTEM OR GOTO PROGRAM IS CURRENTLY ASSIGNED TO YOU
69 TROUBLE WITH FILE SUCH AS: FILE SIZE DOES NOT AGREE WITH
THE START AND END LOCATIONS OF THE GO FILE. (FILE EMPTY?)
70 NO STARTING ADDRESS FOR GOTO PROGRAM!
71 DUMP FILE NOT COMPATIBLE WITH CURRENT SYSTEM
72 APPARENT DISC ERROR
73 INVALID SEPARATER BETWEEN PARAMETERS FOR AN EXEC COMMAND
74 INVALID PARAMETERS FOR AN IEXEC COMMAND
75 MUST BE LOGGED INTO DIRECTORY TO USE THIS COMMAND.
76 UNABLE TO LOCATE A FILE AT THE SPECIFIED POSITION IN DIRECTORY
77 IMPROPER RESPONSE. PROPER RESPONSES ARE:
Y FOR YES, N FOR NO, CARRIAGE-RETURN OR LINE-FEED
78 INVALID USER NAME USED WHEN ADDRESSING A PUBLIC FILE
79 FILE DIRECTORY CONTROLS SET WRONG
80 INSUFFICIENT USER MEMORY FOR MAIL
USE RESET COMMAND AND RETRY
81 FILE DIRECTORY CONTROLS SET WRONG
82 INVALID SUB-SYSTEM NAME
83 FILE TYPE WRONG - NOT DUMP FILE
84 FILE BUSY - SOMEONE ELSE PROBABLY HAS THE FILE OPEN
85 FILE CANNOT BE WRITTEN NOW. DISK SPACE BEING RECLAIMED.
89 MONITOR ERROR

SECTION 10.0 APPENDIX E

FLOATING POINT

DATE: 71/09/16

FLOATING POINT ARITHMETIC IS INCORPORATED INTO THE 940 SYSTEM THROUGH THE USE OF SYSPOPS. FLOATING POINT HARDWARE ('FPAU' = FLOATING POINT ARITHMETIC UNIT) IS AVAILABLE AS AN OPTION. CONDITIONAL ASSEMBLY OF THE MONITOR ALLOWS THE FLOATING POINT SYSPOPS TO UTILIZE THE FPAU OR TO USE SOFTWARE SIMULATION. THE COMPLETE DESCRIPTION OF EACH SYSPOP SHOULD BE CONSULTED TO DETERMINE SUITABILITY OF USE FOR A GIVEN APPLICATION. THREE SETS OF FLOATING POINT SYSPOPS ARE AVAILABLE:

I. STANDARD SYSPOPS

FLOATING POINT NUMBERS ARE NORMALIZED DOUBLEWORD VALUES. THE FIRST WORD IS A SIGN BIT FOLLOWED BY THE 23-BIT HIGH-ORDER PART OF THE MANTISSA; THE SECOND WORD CONSISTS OF THE 15 LOW-ORDER BITS OF THE MANTISSA FOLLOWED BY A 9-BIT EXPONENT. BOTH WORDS ARE TWO'S COMPLEMENT. THESE SYSPOPS REQUIRE THAT THE RESULT OF ANY OPERATION BE RETURNED IN THE A AND B REGISTERS. IF THE FPAU IS IMPLEMENTED, THE CURRENT CONTENTS OF THE A AND B REGISTERS ARE OUTPUT TO THE FPAU, THE OPERATION PERFORMED, AND THE RESULT COPIED TO THE A AND B REGISTERS. THE FOLLOWING SYSPOPS ARE IMPLEMENTED:

FAD	FLOATING ADD
FSB	FLOATING SUBTRACT
FMP	FLOATING MULTIPLY
FDV	FLOATING DIVIDE
BRS 21	FLOATING NEGATE
BRS 50	FIX A & B TO A-REGISTER
BRS 51	FLOAT A-REGISTER TO A & B
LDP	LOAD DOUBLEWORD (FPAU NOT AFFECTED)
STP	STORE DOUBLEWORD (FPAU NOT AFFECTED)

II. FPAU SYSPOPS

THESE SYSPOPS ARE INCLUDED FOR USE WITH THE FPAU IF IT IS NOT NECESSARY TO HAVE THE RESULT OF EACH OPERATION RETURNED IN THE A & B REGISTERS THEREBY PROVIDING A TIME SAVING DURING SUCCESSIVE FLOATING POINT OPERATIONS. ALL OF THESE SYSPOPS PERFORM THE OPERATION WITH THE CURRENT CONTENTS OF THE FPAU AND LEAVE THE RESULT IN THE FPAU. FOR COMPATIBILITY, THESE SYSPOPS ARE ALSO OPERATIVE IF THE FPAU IS NOT IMPLEMENTED, IN WHICH CASE THEY BEHAVE IDENTICALLY TO THE STANDARD SYSPOPS WITH THE EXCEPTION THAT THE FLOATING POINT ACCUMULATOR IS SIMULATED BY THE MONITOR. THE FORMAT OF THE FLOATING POINT WORD IS THE SAME AS FOR THE STANDARD SYSPOPS. THE FPAU SYSPOPS INCLUDE:

FFAD	FLOATING ADD
FFSB	FLOATING SUBTRACT
FFSI	FLOATING SUBTRACT INVERSE

FFMP	FLOATING MULTIPLY
FFDV	FLOATING DIVIDE
FFDI	FLOATING DIVIDE INVERSE
LDfM	LOAD FLOATING ACCUMULATOR
STfM	STORE FLOATING ACCUMULATOR
BRS 82	SKIP IF FLOATING ACCUMULATOR NEGATIVE
BRS 83	SKIP IF FLOATING ACCUMULATOR = 0
BRS 84	SKIP IF FLOATING ACCUMULATOR # 0
BRS 118	COPY A & B TO FLOATING ACCUMULATOR
BRS 119	COPY FLOATING ACCUMULATOR TO A & B
BRS 148	NEGATE FLOATING ACCUMULATOR
BRS 149	FIX FLOATING ACCUMULATOR TO A-REGISTER
BRS 150	FLOAT A-REGISTER TO FLOATING ACCUMULATOR

III. FORTRAN II SYSPOPS

THESE SYSPOPS ARE FOR USE WITH FORTRAN II COMPATIBLE PROGRAMS. THESE SYSPOPS REQUIRE THAT THE FIRST FLOATING POINT WORD CONSIST OF THE 15 LOW-ORDER BITS OF THE MANTISSA FOLLOWED BY A 9-BIT EXPONENT WHILE THE SECOND WORD CONTAINS A SIGN BIT FOLLOWED BY THE 23 HIGH-ORDER BITS OF THE MANTISSA. FOR EASE IN HANDLING ARRAY VARIABLES, THESE POPS DOUBLE THE CONTENTS OF THE X-REGISTER BEFORE PERFORMING THE EFFECTIVE ADDRESS CALCULATION, AND THEN RESTORE THE INITIAL VALUE OF THE X-REGISTER. THE FOLLOWING SYSPOPS UTILIZE THE FPAU AND THE A & B REGISTERS:

FFADD	FORTTRAN FLOATING ADD
FFSBD	FORTTRAN FLOATING SUBTRACT
FFSID	FORTTRAN FLOATING SUBTRACT INVERSE
FFMPD	FORTTRAN FLOATING MULTIPLY
FFDVD	FORTTRAN FLOATING DIVIDE
FFDID	FORTTRAN FLOATING DIVIDE INVERSE
LDfMD	FORTTRAN LOAD FLOATING
STfMD	FORTTRAN STORE FLOATING

IV. FOS POPS

FOS FLOATING POINT NUMBERS HAVE THE SAME FORMAT AS FOR THE FORTRAN II SYSPOPS. THE FOLLOWING POPS ARE DEFINED IN AND USED BY FOS, LIB, PLIB, FSUBR, ETC. AND ARE INCLUDED HERE FOR REFERENCE BY USERS WHO WRITE ASSEMBLY LANGUAGE ROUTINES FOR FOS. THE POP DEFINITIONS GIVEN ARE FOR ARPAS; FOR NARP, SUBSTITUTE ",2" FOR ",1,1" AND ",1,1" FOR ",2". POPS MARKED BY A "*" DOUBLE THE CONTENTS OF THE X-REGISTER BEFORE COMPUTING THE EFFECTIVE ADDRESS:

*	FST	105B5,1,1	FLOATS A, STORES IN MEMORY
*	STD	107B5,1,1	STORES A&B IN MEMORY
*	LDP	125B5,1,1	LOADS A&B FROM MEMORY
	FTA	126B5,1,1	FLOATS INTEGER IN MEMORY, ADDS TO A&B
	FTS	132B5,1,1	FLOATS INTEGER IN MEMORY, SUBS FROM A&B
	FTM	136B5,1,1	FLOATS INTEGER IN MEMORY, MUL BY A&B
	FTD	142B5,1,1	FLOATS INTEGER IN MEMORY, DIV A&B BY IT
	XST	104B5,1,1	FIXES A&B AND STORES IN MEMORY
	LTF	122B5,1,1	FLOATS INTFGER IN MEMORY TO A&B

*	LTX	123B5,1,1	FIXES A&B FROM MEMORY TO A-REGISTER
*	FLA	131B5,1,1	FLOATING ADD
*	FLS	135B5,1,1	FLOATING SUBTRACT
*	FLM	141B5,1,1	FLOATING MULTIPLY
*	FLD	145B5,1,1	FLOATING DIVIDE
	FLN	147B5,2	FLOATING NEGATE OF A&B

V. FLOATING POINT OVERFLOW

IF THE FLOATING POINT OPERATIONS ARE SOFTWARE SIMULATED, EXPONENT OVERFLOW WILL CAUSE THE CPU OVERFLOW INDICATOR TO BE SET. IF THE INDICATOR IS SET BY A FLOATING POINT OPERATION, IT WILL REMAIN SET UNTIL THE USER EXECUTES THE 'OVT' INSTRUCTION.

IF THE FLOATING POINT ARITHMETIC UNIT IS IMPLEMENTED, EXPONENT OVERFLOW WILL CAUSE THE OVERFLOW INDICATOR IN THE FPAU TO BE SET. THE SETTING OF THE INDICATOR FORCES AN INTERRUPT TO OCCUR. THE INTERRUPT ROUTINE WILL RESET THE FPAU OVERFLOW INDICATOR, SET A FLAG THAT INDICATES THAT OVERFLOW OCCURRED, AND ARM THE MONITOR-TO-USER TRANSITION TRAP. WHEN THE TRANSITION TRAP IS ENTERED, THE OVERFLOW FLAG IS EXAMINED. IF THE FLAG IS SET, THE CPU OVERFLOW INDICATOR IS SET AND SOFTWARE INTERRUPT 6, IF ARMED, WILL BE TRIGGERED. IF THE SOFTWARE INTERRUPT IS NOT ARMED, A RETURN TO THE USER PROGRAM WILL OCCUR. THE USER CAN ISSUE THE 'OVT' INSTRUCTION TO TEST FOR OVERFLOW AND RESET THE CPU OVERFLOW INDICATOR.

VI. INPUT/OUTPUT AND CONVERSION OF FLOATING POINT NUMBERS

BRS 52 AND BRS 53 CAN BE USED FOR THE INPUT/OUTPUT OF FLOATING POINT NUMBERS TO A FILE. BOTH THESE BRS'S REQUIRE A FORMAT WORD IN THE X-REGISTER. THE SIC AND ISC SYSPOPS PERFORM STRING TO FLOATING POINT AND FLOATING POINT TO STRING CONVERSIONS, RESPECTIVELY; THEY ALSO REQUIRE A FORMAT WORD IN THE X-REGISTER. THE DESCRIPTION OF THE FORMAT WORD FOLLOWS THE DESCRIPTION OF THE ERROR CODES FOR THESE ROUTINES. THE FLOATING POINT NUMBER USED BY THE ROUTINES IS THE SAME AS FOR THE STANDARD FLOATING POINT ARITHMETIC SYSPOPS.

"STRING" TO FLOATING POINT CONVERSION:

THE D FIELD IS OVERRIDDEN BY THE PRESENCE OF A DECIMAL POINT. IF A DECIMAL POINT AND/OR E ARE PRESENT, ANY FORM OF A NUMBER IS ACCEPTABLE TO ANY INPUT FORMAT. ILLEGAL CHARACTERS APPEARING ANYWHERE IN THE FIELD MAY BE IGNORED BY SETTING BIT 19 IN THE FORMAT WORD. ALL BLANKS WILL BE CONVERTED TO ZERO. FREE FORM INPUT WILL ACCEPT CHARACTERS UNTIL A TERMINATING CHARACTER (ANY CHARACTER EXCEPT + - . E OR DIGIT) IS INPUT. THE MAXIMUM ALLOWABLE INPUT DIGITS = 12. IF MORE THAN 12 ARE USED, THE MOST SIGNIFICANT 12 WILL BE USED. INSIGNIFICANT LEADING OR TRAILING ZEROS WILL BE IGNORED.

FLOATING POINT TO "STRING" CONVERSION:

CONVERSION REQUIRES A FLOATING POINT NUMBER IN A&B (THE MOST SIGNIFICANT FRACTIONAL PART IN A AND THE LEAST SIGNIFICANT FRACTIONAL PART AND EXPONENT IN B). IF FREE FORM OUTPUT IS USED, THE NUMBER WILL BE OUTPUT IN F16 FORMAT WITH 5 LEADING SPACES IF THE EXPONENT IS LESS THAN 37B. FOR LARGER EXPONENT VALUES, E16.9 FORMAT IS USED.

THE ISC AND SIC SYSPOPS FUNCTION SIMILARLY TO THE BRS 52 AND BRS 53 EXCEPT THE FILE NUMBER BITS MUST BE SET TO ZERO.

ERROR CONDITIONS

- X = 0 NO ERROR WAS DETECTED
- X = 1 NUMBER OF DECIMAL DIGITS AFTER THE DECIMAL POINT EXCEEDS 12 ON FORMATTED INPUT
- X = 2 FIELD TOO SHORT FOR E FORMAT ON OUTPUT. OVERFLOW ACTION WILL BE TAKEN DEPENDING ON THE VALUE OF BIT 15 IN THE FORMAT WORD
- X = 3 INPUT NUMBER EXCEEDS THE MAXIMUM ALLOWABLE BOUNDS
- X = 4 FIELD TOO SHORT FOR F OR I FORMAT ON OUTPUT. OVERFLOW ACTION WILL BE TAKEN DEPENDING ON THE VALUE OF BIT 15 OF THE FORMAT WORD
- X = 5 AN E FORMAT WAS SPECIFIED FOR INPUT BUT THE INPUT STRING DOES NOT CONTAIN AN 'E' OR '.'. THE NUMBER WILL BE CONVERTED USING AN EQUIVALENT F FORMAT
- X = 6 AN ILLEGAL CHARACTER WAS ENCOUNTERED IN THE INPUT SCAN. CHARACTER IS IGNORED.

FORMAT WORD DESCRIPTION

BITS 0- 2 FORMAT TYPE

- 0 (ILLEGAL)
- 1 INTEGER (I FORMAT)
- 2 E FORMAT WITH NUMBER RIGHT-JUSTIFIED IN FIELD
- 3 F FORMAT WITH NUMBER RIGHT-JUSTIFIED IN FIELD
- 4 E FORMAT WITH NUMBER LEFT-JUSTIFIED IN FIELD
- 5 F FORMAT WITH NUMBER LEFT-JUSTIFIED IN FIELD
- 6 (ILLEGAL)
- 7 (ILLEGAL)

BITS 3- 8

NUMBER OF DIGITS FOLLOWING THE DECIMAL POINT

BITS 9-14

TOTAL FIELD WIDTH. IF = 0, FREE-FORM INPUT/OUTPUT IN A FIELD WIDTH OF 16 WILL BE PERFORMED

BIT 15

OVERFLOW ACTION. IF THE FIELD WIDTH IS TOO SMALL ON OUTPUT AND THIS BIT IS SET, THE FIRST CHARACTER OF THE OUTPUT FIELD WILL BE AN ASTERISK. IF THIS BIT IS RESET AND FIELD WIDTH OVERFLOW OCCURS, CHARACTERS ON THE RIGHT WILL BE LOST.

BITS 16-23

FILE NUMBER TO BE USED WITH BRS 52 AND BRS 53

EXAMPLES OF THE USE OF THE FORMAT WORD:

ASSUME THAT THE A&B REGISTERS CONTAIN THE VALUE FOR
FLOATING POINT 123.5 USING BRS 53:

X-REGISTER	OUTPUT
30105---	'123.5'
30206---	'123.50'
00000---	' 123.5000000'
30210---	' 123.50'
30207---	' 123.50'
20514---	' 1.23500E+02'
40514---	' 1.23500E+02 '

SECTION 11.0 APPENDIX F

MACHINE LANGUAGE SUBROUTINES

DATE: 71/09/16

SOME USER PROGRAMS MAY REQUIRE THE COPYING OF A DISC FILE. THE FOLLOWING RECOMMENDED PROCEDURE IS FASTER THAN A CIO LOOP. IF THE (FILE SIZE) MOD 3 IS NON ZERO, A BIO WILL INTRODUCE EXTRANEOUS BLANKS IN THE LAST WORD OF THE FILE. THE FOLLOWING ROUTINE ELIMINATES THAT POSSIBILITY:

```

LENGTH EQU 10000B      LENGTH OF FILE BUFFER (CAN BE CHANGED)
COPY   RCP OUTFIL; MRG =4B7; SCP OUTFIL; HLT (ERROR, TRAP)
      RSP INFIL; STA SIZE
COPYA  LDA =LENGTH; LDX =BUFFER; BIO INFIL; BRU EOF
      LDA =LENGTH; LDX =BUFFER; BIO OUTFIL; NOP; BRU COPYA
EOF    SUB =BUFFER; LDX =BUFFER; SKE =0; BIO OUTFIL; NOP
      RSP OUTFIL; XMA SIZE; MRG =4B7; SCP OUTFIL; HLT; LDA SIZE
      PCE OUTFIL; NOP; BRS 147

```

```

SIZE   ZRO 0      SIZE OF FILE
INFIL  ZRO 0      INPUT FILE NUMBER
OUTFIL ZRO 0      OUTPUT FILE NUMBER
BUFFER BSS LENGTH TRANSFER BUFFER

```

WHEN OPENING A FILE, AN ERROR MAY OCCUR AND THE EXEC MAY NOT TYPE AN ERROR MESSAGE. THE FOLLOWING ROUTINE WILL TAKE THE ERROR NUMBER AND PRINT THE ERROR MESSAGE. (SEE APPENDIX D)

```

WHY    LDP WHYPTR; BRS 64; BPS 10 (NOT THERE); STA WHYFIL
      BRS 73 (PEAD ERCODE); SKA =32B5 (MONITOR ERROR?); BRS 10
      MUL =3; LSH 23; ADD =21; SCP WHYFIL; HLT; WIO WHYFIL; SUB =235B
      MUL =3; LSH 23; SCP WHYFIL; HLT
LOOP   CIO WHYFIL; SKE = ' /'; BRU TYPE; BRS 10
TYPE   SKE = ' $'; BRU *+2; LDA =155B; CIO =1; BRU LOOP

```

```

WHYFIL ZRO 0      WHY FILE NUMBER
WHYPTR DATA (R)WHYSTR-1,(R)WHYSTR+3      USE THIS IN ARPAS AND
WHYPTR DATA 3*WHYSTR-1,3*WHYSTR+3      THIS IN NARP.
WHYSTR ASC '*WHY'

```

ANOTHER APPROACH TO THE SAME SITUATION IS TO USE THE EXECUTE EXEC COMMAND BRS AS THE FOLLOWING:

```

WHY    LDA = 'WHY'; MRG =4B7; BRS 142; ...

```

WHEN TRANSMITTING FILES, IT IS USEFUL TO HAVE A CHECKSUM TO COMPARE WITH THE ORIGINAL. THE FOLLOWING ROUTINE WILL PRINT OUT THE CHECKSUM FOR THE INPUT FILE:

```

CHECK  WIO INFIL; SKN INFIL; BRU *+2; BRU END; STA WIOSUM
      LDA CHKSUM; CLB; LCY 7; ADD WIOSUM; STB CHKSUM
      ADM CHKSUM; BRU CHECK

```

END LDA CHKSUM; LDB =8; LDX =1; BRS 36; BRS 10

INFIL ZRO 0 FILE TO BE CHECKSUMED FILE NUMBER

WIOSUM ZRO 0 TEMPORARY FOR INPUT FROM WIO

CHKSUM ZRO 0 FILE CHECKSUM

SECTION 12.0 APPENDIX G

ALL ABOUT PMT'S AND SMT'S

DATE: 71/03/29

PSEUDO RELABELING IS A MECHANISM WHEREBY A MACHINE LANGUAGE PROGRAM CAN ACCESS MORE STORAGE THAN 16K WORDS WITHOUT USING THE DISK.

A PAGE IS A UNIT OF INFORMATION COMPRISED OF 2048 WORDS. A PAGE IS SOMETIMES IN CORE, SOMETIMES ON THE DRUM(RAD) AND SOMETIMES ON BOTH AT ONCE. WHEN A USER PROGRAM IS RUNNING, IT HAS ACCESS TO EIGHT PAGES OR LESS. A BRS 43 TELLS A USER WHICH OF THE USER'S POSSIBLE EIGHT PAGES ARE IN USE, AND FOR THOSE IN USE IT GIVES A PAGE NAME WHICH CAN BE USED WITH OTHER BRS'S. WHEN A PROGRAM LOADS OR STORES AT AN ADDRESS WHICH DOESN'T CORRESPOND TO A PAGE, THE PROGRAM IS TRAPPED AND RESTARTED LATER WITH A NEW REAL PAGE AT THAT ADDRESS. IF ALL 33 OCTAL PAGES ARE IN USE, A MEMORY TRAP RESULTS. A BRS 44 DIRECTS THE MONITOR TO PLACE PAGES AT THE SPECIFIED ADDRESSES IN THE USER'S PROGRAM. THESE PAGES ARE NAMED WITH THE NAMES GIVEN BY THE MONITOR IN PREVIOUS BRS 43'S OR BRS 3'S. THE BRS 3 IS USED TO POINT TO A SUBSYSTEM BYTE. UNLESS THE FORK HAS EXEC STATUS THE BYTE MUST BE READ ONLY. IF THE BYTE HAS BEEN MADE VISIBLE TO USERS, IT MAY BE RELABELED IN WITHOUT STATUS. THE FOLLOWING IS LIST OF HOW NOT TO OBTAIN MORE MEMORY:

- 1) BY FALLING THROUGH A PAGE. (GOING OVER A PAGE BOUNDRY)
- 2) BY LOADING/STORING INDIRECT THROUGH AN OUT OF BOUNDS ADDRESS
- 3) BY DOING A FXU TO AN OUT OF BOUNDS ADDRESS
- 4) BY EXECUTING A POP IF PAGE 0 IS NOT IN
- 5) BY DOING A BRS 44 OR 117 REQUESTING A NON ACQUIRED BYTE NUMBER.
- 6) BY EXECUTING A BRS 9 AND USING ILLEGAL RELABELING BYTES

THERE ARE SOME PAGES THAT ARE IN EXISTENCE BEFORE THE JOB COMES INTO EXISTENCE. THEY CONTAIN INFORMATION THAT IS COMMONLY USED BY MANY JOBS. MOST OF THESE PAGES HOLD PROPRIETARY INFORMATION. THEY HAVE NUMBERS LESS THAN 43 AND ARE KNOWN AS SMT'S, FOR "SHARED MEMORY" INSTEAD OF "PRIVATE MEMORY".

SECTION 13.0 APPENDIX H

TYMNET

DATE: 71/09/22

THE TYMNET APPENDIX HAS NOT BEEN WRITTEN YET. PLEASE WAIT PATIENTLY.

SECTION 14.0 APPENDIX I

FORKING

DATE: 71/09/22

Sometimes programs need to control other programs. The primary mechanism provided by the operating system for this purpose is the fork. In the Tymshare system a fork is very much like a subroutine. The initiating fork is called the upper fork. It plays the role of the caller of a subroutine. The initiated fork is called the lower fork. It is analogous to the subroutine. The memory of different forks may be comprised of different pages. The upper fork controls the pages that the lower fork starts with. The upper fork may limit the degree of control that a lower fork has of the escape (alt-mode) mechanism. With the absolute memory bit the upper fork can prevent the lower fork from getting access to any other pages than it was stated with. With the local memory bit the upper fork causes references by the lower fork to absent pages to get new pages. Otherwise the page with the corresponding address in the upper fork would be used.

At one time one fork has control over the escape key. The upper fork may delegate this control to a lower fork when it starts it by bit 2 in the accumulator argument to the BRS 9. This is called "propogating the escape".

SECTION 15.0 APPENDIX J

INDEX BY BRS FUNCTION

DATE: 71/10/01

INDEX	BRS NO.S
110 BAUD.....	BRS 132
150 BAUD.....	BRS 132
211B.....	BRS 153, 154
300 BAUD.....	BRS 132
ANSWER.....	BRS 159
AUXILARY CIRCUIT.....	BRS 138, 139
AUXILARY CIRCUITS.....	BRS 156
BCD.....	BRS 107
BIT MAP.....	BRS 25, 30, 54
CHARACTER GOBBLER.....	BRS 11, 29
CLOSE.....	BRS 2, 8, 17, 20, 147
COMMAND FILE.....	BRS 17
COMMAND PORT.....	BRS 12, 29, 132,
CONTROL PARAMTER WORD.....	BRS 39
CONVERSION.....	BRS 50, 51
CPU.....	BRS 88
DATA BLOCK.....	BRS 41
DATE.....	BRS 91
DELETE.....	BRS 69
DENSITY.....	BRS 108
DISMISS.....	BRS 14, 45, 72
ECHO.....	BRS 12
ECHO TABLE.....	BRS 12, 40
EIGHT-LEVEL.....	BRS 85, 86
ERCODE.....	BRS 73
EXEC STATUS.....	BRS 9
EXPANDED SMT.....	BRS 3, 57
FATAL EXCEPTION.....	BRS 1
FILE.....	BRS 1, 2, 8, 15, 16, 18, 19, 20, 37, 48 58, 62, 63, 64, 65, 66, 68, 69, 147
FILE ATTRIBUTES.....	BRS 96
FILE DIPIRECTORY.....	BRS 15, 48, 60, 68, 96
FILE DIRECTORY POINTER ADDRESS..	BRS 1, 6, 37
FILE NAME.....	BRS 60
FILE NUMBER.....	BRS 33, 34, 35, 36, 38
FILE PARAMETERS.....	BRS 6
FILE SIZE QUANTUM.....	BRS 92
FILES.....	BRS 17
FIXED MEMORY.....	BRS 9
FLOATING ACCUMULATOR.....	BRS 82, 83, 84
FLOATING POINT.....	BRS 21, 50, 51
FORK.....	BRS 5, 9, 10, 31
FORMAT.....	BRS 52, 53
INDEX BLOCK.....	BRS 157
INDIRECT POINTER.....	BRS 3
INPUT.....	BRS 62, 64

INPUT BUFFER.....	BRS 11, 13
INPUT PORT.....	BRS 139, 155
INTERRUPT.....	BRS 135
INTERRUPTS.....	BRS 49
KILL.....	BRS 4
LEVEL 3.....	BRS 157
LOCAL MEMORY.....	BRS 9
LOOKUP.....	BRS 15, 18, 48
LOWER FORK.....	BRS 9
MAG TAPE.....	BRS 102, 103, 105, 107, 108, 110
MESSAGE.....	BRS 34
MONITOR TABLES.....	BRS 7
MUD.....	BRS 138
N/(2**A).....	BRS 132
NON-TERMINABILITY.....	BRS 46, 47
NUMBER.....	BRS 36, 38
OFF INTERRUPT.....	BRS 26
OPEN.....	BRS 1, 16, 19, 62, 63, 64, 65
OUTPUT.....	BRS 63, 65
OUTPUT BACKPRESSURE.....	BRS 14
OUTPUT BUFFER.....	BRS 14, 29, 164
OUTPUT PORT.....	BRS 155
OVER FLOW GROUPS.....	BRS 95
PAC.....	BRS 9
PAGES.....	BRS 70
PANIC.....	BRS 9, 26, 90
PANIC ASSIGNMENT.....	BRS 9
PANIC TABLE.....	BRS 9, 31
PAPER TAPE MODE.....	BRS 134
PARITY.....	BRS 107
PHANTOM USER.....	BRS 26
PMT.....	BRS 56
PORT.....	BRS 138, 155
PORT NUMBER.....	BRS 87
PORT NUMBERS.....	BRS 137
PREMIUM CHARGE.....	BRS 170, 171
PRINTER.....	BRS 106
RANDOM I/O.....	BRS 63
READ ONLY.....	BRS 80
RECOVER.....	BRS 56
RELABELING.....	BRS 3, 43, 44
RENAME.....	BRS 37, 58
RESOURCE METERING.....	BRS 89
ROYALTY UNITS.....	BRS 158
SHIFT CONTROL O.....	BRS 168, 169
SOFTWARE INTERRUPT.....	BRS 78
STATISTICS.....	BRS 27, 28
STRING.....	BRS 33, 35
STRING POINTER.....	BRS 33
SUBSYSTEM BYTE NUMBER.....	BRS 3
SUBSYSTEM STATUS.....	BRS 9
SUPERVISOR.....	BRS 138, 153, 154,
SYSTEM STATUS.....	BRS 9
TERMINAL.....	BRS 12

TERMINAL CHARACTERISTICS	BRS 132, 160
TERMINATE	BRS 10, 31, 90
TIME	BRS 91, 135
TOUT	BRS 17, 77
TRAP	BRS 9
TS PAGE	BRS 67
TYMNET	BRS 172
UNIVERSAL USER NUMBER	BRS 39
USER NAME	BRS 59
YELLOW BALL	BRS 164
ZAP	BRS 139, 156