BRS REFERENCE DOCUMENT

Issued March 3, 1967

BY

A. Hardy
D.L. Schmidt
A.P. Scott
V.E. Van Vlear

TABLE OF CONTENTS

1.0 SCHEDULING, FORKS AND PROGRAM INTERACTION

NUMBER: 78

NAME: SAIR

FUNCTION: Arm/Disarm Software Interrupts

STATUS: User

CALLING SEQUENCE:
```
LDA   M
BRS   78
```

M is the complete new interrupt mask.

DESCRIPTION: The new interrupt mask is substituted for the old one. A user may arm interrupts 1-10. An exec fork may arm interrupt 11 also. Interrupt 1 is in bit 4 of the mask word. The interrupts are as follows:

1 Interrupt if Program Panic (BRS 10 or Escape)

2 Interrupt if Memory Panic

3 Interrupt if Lower Fork terminates

4 Interrupt if any I/O condition occurs which sets a flag bit (0,7 or 8 in file number word)

11 Interrupt if DSU error

5 through 10 interrupts on condition set by user

Location 200 octal plus the interrupt number must be set to point to a routine to process the interrupt. When the interrupt occurs an SBRM* is executed to the location pointed to. If it is desired to return to the point in the program interrupted, the user must BRR to the location where the return was saved.

Example:

```
   SET          INTERRUPT ROUTINE       RETURN
LDA =ESCAPE     ESCAPE ZRO ESCRTN    BRR ESC RTN
STA 201B                 .
 .                       .
 .                       .
```

REGISTERS AFFECTED: None

NUMBER: 79

NAME: SIIR

FUNCTION: Cause Interrupt

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   79
```

N = Interrupt number. N has the range of 5 to 10.

DESCRIPTION: Parallel forks in the structure are searched first and then higher forks. The interrupt will be caused in the first fork found which has the interrupt armed. If no fork has the interrupt armed, it is treated like a NOP. This would normally be used to cause interrupts 5 through 10 to interrupt.

REGISTERS AFFECTED: None

NUMBER: 49

NAME: SRIR

FUNCTION: Read Interrupts Armed

STATUS: User

CALLING SEQUENCE: BRS 49

DESCRIPTION: Reads the interrupt mask into the A register. Bit 4 corresponds to interrupt number 1, 5 to number 2 and etc. There are 11 programmable interrupts. See also BRS 78 (1.1.1).

REGISTERS AFFECTED: A

NUMBER: BE+12

NAME: TIMINT

FUNCTION: Interrupts a fork after a specified period of time.

STATUS: User

CALLING SEQUENCE:
```
LDA   M
LDB   T
LDX   N
BRS   BE+12
NORMAL RETURN
```

M is the new interrupt mask.
T is the time in milliseconds after which the fork will be interrupted.
N is the interrupt number.

DESCRIPTION: The fork issuing this BRS will be interrupted after the delay if the interrupt specified by N is armed at that time. (Exception: The interrupt will be ignored if the fork is dismissed on a BRS 9 (1.3.1) at the time of the interrupt.) If a fork gives this BRS again with the same N before the time has passed, the new time will be set. A fork may have a maximum of three timing interrupts pending simultaneously. See also, BRS 81 (1.5.3).

REGISTERS AFFECTED: None

NUMBER: 90

NAME: DFR

FUNCTION: Declare a Fork for "Escape"

STATUS: User

CALLING SEQUENCE: BRS 90

DESCRIPTION: In case the user types "Escape" or a fork panics, this fork will be activated. A fork panic is a fork status of 0, 1, or 2. See also, BRS 10 (1.6.1).

REGISTERS AFFECTED: None

NUMBER: 46

NAME: NROUT

FUNCTION: Turn Escape Off

STATUS: System

CALLING SEQUENCE: BRS 46

DESCRIPTION: This BRS will set up to remember an escape interrupt, but not allow the program to be interrupted. It will stack the first escape occurring and ignore any subsequent ones.

A program running with escape turned off cannot be terminated by a higher fork.

See also, BRS 26 (1.2.4) and 47 (1.2.3).

REGISTERS AFFECTED: None

NUMBER: 47

NAME: SROUT

FUNCTION: Turn Escape On

STATUS: System

CALLING SEQUENCE: BRS 47

DESCRIPTION: This BRS reverses BRS 46 (1.2.2); that is, reactivates the escape interrupt. If an escape interrupt was stacked (remembered) while in an Off condition, the interrupt will occur.

REGISEERS AFFECTED: None

NUMBER: 26

NAME: SKROUT

FUNCTION: Skip if Escape Waiting

STATUS: System

CALLING SEQUENCE: BRS 26
EXCEPTION RETURN
NORMAL RETURN

DESCRIPTION: Checks for a stacked escape for this program and if there is one, transfers control to the "normal return" or to the "exception return" if there is not an escape stacked. Significant only after BRS 46 (1.2.2).

REGISTERS AFFECTED: None

NUMBER: 9

NAME: FKST

FUNCTION: Open Fork

STATUS: User

CALLING SEQUENCE:
```
LDA   T
BRS   9
```

T = Address of a "Panic Table"
Bits 0 through 4 of register A have the following significance:

0 = Make fork system if current fork is system.
1 = Set fork relabeling from panic table. Otherwise use current relabeling.
2 = Propogate escape assignment to fork (See BRS 90, 1.2.1).
3 = Make fork fixed memory. It is not allowed any more memory than it started with.
4 = Make fork local memory. New memory will be assigned to it independent of the controlling fork.
5 = Make fork privileged if current fork is privileged.

DESCRIPTION: BRS 9 is used to create dependent entries in the PAC table. The panic table indicated by register A must not be the same for two forks of the same fork or overlap a page boundary; if it is, BRS 9 is illegal. BRS 9 creates a new fork as a fork of the fork creating it, which is called the controlling fork. The fork is lower in the hierarchy of forks than the controlling fork. The controlling fork may itself be a fork of some still higher fork. For more detailed information see Document No. 30.10.31.

When BRS 9 is executed by a user fork, the user fork is dismissed until the lower fork terminates. This has the same effect as issuing a BRS 31 (1.5.4) immediately after a BRS 9. A user may not have more than eight forks in his fork structure. This includes the system fork and one fork for each system BRS that is active. Only one system BRS can be active.

REGISTERS AFFECTED: None.

NUMBER: 57

NAME: CQO

FUNCTION: Guarantee 16ms Computing

STATUS: User

CALLING SEQUENCE: BRS 57

DESCRIPTION: This BRS guarantees to the user upon return at least 16 msec. of uninterrupted computation. This is done by dismissing the user if less than 16 msec. remain in his time quantum.

This time will include some system overhead. Thus, if the time required is very close to 16 msec., a BRS 45 (1.5.1) should be used. BRS 45 guarantees several times this amount.

REGISTERS AFFECTED: None

NUMBER: 30

NAME: FKRD

FUNCTION: Read Fork

STATUS: User

CALLING SEQUENCE:
```
LDA   P
BRS   30
```

P = Panic Table Address

DESCRIPTION: Reads the current status of a lower fork into the panic table indicated by the A register. It does not influence the operation of the fork in any way.

REGISTERS AFFECTED: None

NUMBER: 107

NAME: FKRA

FUNCTION: Read All Fork Statuses

STATUS: User

CALLING SEQUENCE: BRS 107

DESCRIPTION: The status of all lower forks is recorded in the appropriate panic tables.

REGISTERS AFFECTED: None

NUMBER: 45

NAME: SQO

FUNCTION: Dismiss on Quantum Overflow

STATUS: User

CALLING SEQUENCE: BRS 45

DESCRIPTION: This BRS causes the user to be dismissed as though he had overflowed his quantum. It guarantees that the next time he is started he will have a complete short time quantum. See BRS 57 (1.3.2) to guarantee 16 msec.

REGISTERS AFFECTED: None

NUMBER: 72

NAME: EXDMS

FUNCTION: System Fork Dismissal

CALLING SEQUENCE:
```
LDX   N
BRS   72
```

N = The number of the queue that the fork is to be put on.

DESCRIPTION: Dismisses a system fork and puts it on the specified queue. Returns to call +1 when recalled.

0 = Teletype queue
1 = I/O queue
2 = Short time quantum queue
3 = Long time quantum queue

REGISTERS AFFECTED: None

NUMBER: 81

NAME: WREAL

FUNCTION: Dismiss for Specified Amount of Time

STATUS: User

CALLING SEQUENCE:
```
LDA   T
BRS   81
```

T = Dismissal time in milliseconds.

DESCRIPTION: The fork is dismissed for the number of milliseconds specified in A. See also, BE+12 (1.1.4).

REGISTERS AFFECTED: A

NUMBER: 31

NAME: FKWT

FUNCTION: Wait for Fork to Cause a Panic

STATUS: User

CALLING SEQUENCE:
```
LDA   P
BRS   31
```

P = Panic Table Address

DESCRIPTION: Causes the controlling fork to be dismissed until the lower fork, or forks, causes a panic. When it does, the controlling fork is reactivated at the instruction following this BRS, and the panic table contains the status of the fork on its dismissal. The status is also put into the X register. The panic table address is put into the A register.

The controlling fork must have armed an interrupt or a lower fork must execute a BRS 10 (1.6.1).

REGISTERS AFFECTED: X,A

NUMBER: 106

NAME: FKWA

FUNCTION: Wait for Any Fork to Terminate

STATUS: User

CALLING SEQUENCE: BRS 106

DESCRIPTION: Fork is dismissed until some lower fork terminates. Whan a lower fork terminates, the panic table address will be left in A.

REGISTERS AFFECTED: None

NUMBER: 109

NAME: DMS

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

NUMBER: 10

NAME: PPAN

FUNCTION: Programmed Panic. Terminates a Fork.

STATUS: User

CALLING SEQUENCE: BRS 10

> BRS 10 terminates the fork that issues it and returns control to the higher fork. It is just like typing "escape" on the teletype.

DESCRIPTION: Terminates a lower fork. This condition can be distinguished from a panic caused by the escape key only by the fact that in the former case the program counter in the panic table points to a word containing BRS 10. This BRS would normally be used to terminate a fork when it is finished. The information in the panic table would, therefore, only be useful to a higher fork or to this fork only if interrupt 4 has been armed by this fork.

REGISTERS AFFECTED: None

NUMBER: 32

NAME: FKTM

FUNCTION: Terminate a Fork

STATUS: User

CALLING SEQUENCE:
```
LDA   P
BRS   32
```

P = Panic Table

DESCRIPTION: Causes a lower fork to be unconditionally terminated and its status to be stored into the panic table. The X register contains the status word upon return.

REGISTERS AFFECTED: X

NUMBER: 73

NAME: EPPAN

FUNCTION: Economy Panic

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   73
```

N = Number of forks to terminate.

DESCRIPTION: This is like doing a BRS 10 for each of the forks specified. Forks are terminated going up until the system fork is reached or until N forks have benn terminated.

REGISTERS AFFECTED: None

NUMBER: 108

NAME: FKTA

FUNCTION: Terminates All Forks

STATUS: User

CALLING SEQUENCE: BRS 108

DESCRIPTION: All lower forks are terminated and their status read into the corresponding panic tables.

REGISTERS AFFECTED: None

NUMBER: 1

NAME: MONOPN

FUNCTION: Open a File of a Specific Device

STATUS: System

CALLING SEQUENCE:
```
LDA   ±I
LDX   D
BRS   1
EXCEPTION RETURN
NORMAL RETURN
```

File number will be in register A on Normal Return.
I = The relative address (DSU Address MOD 4) of the file's Index Block for DSU files, or unit number for magnetic tape, otherwise anything.
- = Make the file read only.
+ = Make the file read/write.
D = Device number.

Available device numbers are as follows:

1. Paper tape input.
2. Paper tape output
3. Card input.
4. Magnetic tape input.
5. Magnetic tape output.
7. Printer.
8. Sequential DSU input
9. Sequential DSU output.
10. Random DSU

DESCRIPTION: The "open file" BRS is used to condition a file for input or output processing. If the file is successfully opened, control is transferred to the normal return; otherwise control is transferred to the exception return. Exception conditions are as follows:

1. Device in use or not available.
2. File in use.

A file may be opened for input any number of times for the purpose of multiple user access or multiple processing by a single user. A file that is opened for output cannot be opened again until it is closed. See also, BRS's: 2, 3, 20, 82.

REGISTERS AFFECTED: A,X

NUMBER: 110

NAME: RDU

FUNCTION: Read Device and Unit

STATUS: User

CALLING SEQUENCE:
```
LDA   =FILE No.
BRS   110
NORMAL RETURN
```

DESCRIPTION: Output X = device number.
A = Unit number.
See BRS 1 (2.1.1) for device number description.

REGISTERS AFFECTED: A,X

NUMBER: 2

NAME: MONCLS

FUNCTION: Close a File

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   2
NORMAL RETURN
```

N = File number (obtained when file was opened).

DESCRIPTION: The "close file" BRS is used to indicate to the system all processing is completed on this file. All necessary termination processing will be completed and control will be transferred to the normal return. See also, BRS's 1, 8, and 82.

REGISTERS AFFECTED: None

NUMBER: 20

NAME: CFILE

FUNCTION: Close a File

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   20
```

N = File Number

DESCRIPTION: The "close file" BRS is used to indicate to the system all processing is completed on this file. If the file number indicates Mag Tape, the file will be terminated and if output, the End of File will be written; but in either case, the tape will be positioned at the start of the next file and the tape is de-allocated. All registers are clobbered.

REGISTERS AFFECTED: All

NUMBER: 8

NAME: IOH

FUNCTION: Close all Files

STATUS: User

CALLING SEQUENCE: BRS 8
NORMAL RETURN

DESCRIPTION: The "close all files" BRS is used to indicate to the system all processing is completed on all files. The system will complete all necessary termination processing on all files and transfer control to the normal return. BRS 8 is always executed when control returns to the system. This BRS will not close magnetic tape files correctly. See also, BRS 1 (2.1.1), 2 (2.1.3), 82 (2.1.8), and 17 (2.2.3).

REGISTERS AFFECTED: None.

NUMBER: 66

NAME: DFDL

FUNCTION: Delete DSU File Data

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   66
NORMAL RETURN
```

N = File Number

DESCRIPTION: This BRS will return to available storage all DSU blocks which are assigned to the indicated file and clear the index block of DSU addresses.

REGISTERS AFFECTED: None

NUMBER: 67

NAME: DFER

FUNCTION: Delete a Specified Block of the DSU

STATUS: System

CALLING SEQUENCE:
```
LDA   D
BRS   67
NORMAL RETURN
```

D = Address of the DSU block.

DESCRIPTION: This BRS will return the DSU block indicated by the address in register A to available storage and transfers control to the normal return. This BRS should be used to delete Index Blocks. The BRS does not clear the Index Block address from the Customer File Directory.

REGISTERS AFFECTED: None

NUMBER: 82

NAME: SWSF

FUNCTION: Switch Sequential File Type

STATUS: User

CALLING SEQUENCE:
```
LDA   N
LDB   C
BRS   82
```

N = File number
C = 0 will make the file an input file.
C = 1 will make the file an output file.

DESCRIPTION: This BRS sets the file type to input or output depending on the contents of register B regardless of its current file type and transfers control to the normal return. RESTRICTION: If the sign bit of register A was set when the BRS 1 (2.1.1) was executed to open the file, it cannot be switched from input to output. A violation results in an instruction trap.

REGISTERS AFFECTED: None

NUMBER: 87

NAME: DFRX

FUNCTION: Read DSU File Index Block

STATUS: System

CALLING SEQUENCE:
```
LDA   D
LDX   W
BRS   87
NORMAL RETURN
```

D = DSU address of the index block (MOD 4)
W = Core address into which the block is to be read.

DESCRIPTION: Reads the specified block into the given core location and transfers control to the normal return. The block read is the size of the currently defined index block. The size of an index block varies with the assembly.

REGISTERS AFFECTED: None

NUMBER: 104

NAME: RSYB

FUNCTION: Read a Page from the RAD

STATUS: System

CALLING SEQUENCE:
```
LDA   C
LDB   R
BRS   104
```

C = Core Address
R = RAD Address

DESCRIPTION: Reads one page from the RAD starting at the address R into a page in core. C may be any location in that page. The data will start in the first word of the page.

Uncorrectable RAD errors result in an instruction trap or interrupt 11 if it is armed. Try command again.

REGISTERS AFFECTED: None

NUMBER: 105

NAME: WSYB

FUNCTION: Write a Page on the RAD

STATUS: System

CALLING SEQUENCE:
```
LDA   C
LDB   R
BRS   105
NORMAL RETURN
```

DESCRIPTION: Writes one page on the RAD starting at the address R from a page in core. C may be any location in that page. The data will start in the first word of the page.

Uncorrectable RAD errors result in an instruction trap or interrupt 11 if it is armed. Try command again.

REGISTERS AFFECTED: None

NUMBER: 113

NAME: DFCD

FUNCTION: Compute File Size of a DSU File

STATUS: User

CALLING SEQUENCE:
```
LDA   =File Number
BRS   DFCD
NORMAL RETURN
```

DESCRIPTION: Adds the number of data words (in multiples of 255) in the file to the number in the X register. Returns the result in X.

REGISTERS AFFECTED: X

NUMBER: 114

NAME: MTDI

FUNCTION: Turn Off Run-away Magnetic Tape

STATUS: System

CALLING SEQUENCE: BRS MIDI
NORMAL RETURN

DESCRIPTION: Issues commands to try to stop the tape.

REGISTERS AFFECTED: None

NUMBER: 118

NAME: TGET

FUNCTION: Allocate Magnetic Tape Unit

STATUS: System

CALLING SEQUENCE:
```
LDA    =Tape Number
BRS    118
EXCEPTION RETURN
NORMAL RETURN
```

DESCRIPTION: Assigns tape requested to the user. If tape is already busy with someone else the execption return is executed.

REGISTERS AFFECTED: None

NUMBER: 119

NAME: TREL

FUNCTION: De-Allocate Magnetic Tape Unit

STATUS: System

CALLING SEQUENCE:
```
LDA    =Tape Number
BRS    119
NORMAL RETURN
```

DESCRIPTION: Releases the tape specified. Releases regardless of who had it.

REGISTERS AFFECTED: None

NUMBER: BE+9

NAME: RDSYB

FUNCTION: Read DSU Page

STATUS: System

CALLING SEQUENCE:
```
LDA   C
LDB   R
BRS   BE+9
```

C = Core Address
R = RAD Address

DESCRIPTION: Use like 104. Can only be called by the system exec. BE+1 (2.1.19) can be used to perform this function.

REGISTERS AFFECTED: None

NUMBER: BE+10

NAME: WDSYB

FUNCTION: Write DSU Page

STATUS: System

CALLING SEQUENCE:
```
LDA   C
LDB   R
BRS   BE+10
```

C = Core Address
R = RAD Address

DESCRIPTION: Use like 105 (2.1.11). Can only be called by the system exec. BE+2 (2.1.20) should be used to perform this function.

REGISTERS AFFECTED: None

NUMBER: BE+7

NAME: BPTEST

FUNCTION: Test a Breakpoint Switch

STATUS: System

CALLING SEQUENCE:
```
LDX    =Switch Number
BRS    BE+7
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

NUMBER: BE+1 1240

NAME: ARD

FUNCTION: Read DSU

CALLING SEQUENCE:
```
LDA   =Core Address  1000
LDB   =Disc Address  21000
LDX   =Number of Words 200
BRS   BE+1
NORMAL RETURN
```

DESCRIPTION: Reads from the disc as specified. Errors result in an instruction trap, or programmed interrupt 11 if it is armed. No two forks that are to run simultaneously should both use this BRS.

REGISTERS AFFECTED: None

NUMBER: BE+2

NAME: AWD

FUNCTION: Write DSU

STATUS: System

CALLING SEQUENCE:
```
LDA   =Core Address
LDB   =Disc Address
LDX   =Number of Words
BRS   BE+2
```

DESCRIPTION: Like BE+1 (2.1.19). The number of words must be a multiple of 64 and greater than 0.

REGISTERS AFFECTED: None

NUMBER: 15

NAME: GFN

FUNCTION: Reads Input File Name from a Command File and Looks up the File Name in the User's File Directory.

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   15
EXCEPTION RETURN
NORMAL RETURN
```

N = Command File Number

DESCRIPTION: The routine ignores leading spaces, leading multi-blanks, and leading carriage return characters. It then uses the BRS 37 (5.2.2) to look up the file name in the user's file directory hash table.* It returns in the registers for both returns exactly what the BRS 37 puts there, which is:

Exception Return:	X:	Pointer to the input file name string pointers.
	A & B:	Input file name string pointers.
Normal Return:	A:	Pointer to the string pointers of the desired file in the file directory hash table.
	B:	The value word of the hash table entry.
	X:	Clobbered.

*The exception return is taken if the input file name string cannot be located in the file directory.

NOTE: The information contained in the registers cannot be used directly by the user since the addresses are in the T.S. Block; this BRS is normally followed by the BRS 16.

If the input file name string begins with a left paren, or with the full quote, the file name will be located in another user's file directory or in the public file directory, respectively; in these cases, the input command file must be the teletype. Since the BRS 37 is not used in this case, the information in the registers is of no practical use to the user, and the BRS MUST be followed by the BRS 16 (2.2.2) as indicated under the BRS 16.

REGISTERS AFFECTED: None

NUMBER: 16

NAME: GIFNB

FUNCTION: Open Input File in File Directory.

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   15
BRU   (Error)
BRS   16
EXCEPTION RETURN
NORMAL RETURN
```

N = File Directory Pointer Address

DESCRIPTION: Opens an input file located in the user's file directory. The BRS requires in A, the location of the first word of the entry in the file directory hash table. The exception return is taken if the pointer in A is not pointing to a proper location in the hash table, or if the file cannot be opened for any reason, such as a physical device that cannot be an input file. The file directory pointer may be obtained from a BRS 15 or a BRS 18.

Exception Return:		All registers clobbered.
Normal Return:	A:	File Number
	B:	File Type (0-4)
	X:	File Size

REGISTERS AFFECTED: All

NUMBER: 17

NAME: UABORT

FUNCTION: Close all Files (Including Mag Tape)

STATUS: User

CALLING SEQUENCE: BRS 17

DESCRIPTION: If mag tape has been used, the last record will be terminated and if output, the End of File will be written; in either case the tape will be positioned at the start of the next file. The tape is then closed and the unit is de-allocated. See also, BRS 8 (2.1.5). All registers are clobbered.

REGISTERS AFFECTED: All

NUMBER: 18

NAME: GOFNA

FUNCTION: Reads File Name from a Command File and Looks Up the File Name in the User's File Directory. The Command File Must Be an Input File.

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   18
EXCEPTION RETURN
NORMAL RETURN
```

N = Command File Number
Bit 1 = 1 of A Register = Assume a file name is correct and does not type "OLD FILE" or "NEW FILE".

DESCRIPTION: The routine ignores leading spaces, leading multi-blanks, and leading carriage return characters. If the string begins and ends with a single quote or a slash, the string is terminated for look-up with this character and the string is looked up in the user's file directory using the BRS 5 (5.2.1). A confirming carriage return must follow the quote or slash before the string is looked up. The exception exit is taken if the character is not a carriage return. If the string is found in the file directory hash table, the message "OLD FILE" is typed, otherwise the message "NEW FILE" is typed. If a confirming line feed, carriage return, or period is then next in the input string, the normal return will be taken, otherwise the exception return. In the case of a new file, the file name is inserted conditionally into the file directory.

If the string begins with a character other than a single quote or a slash, the string is looked up in the user's file directory using the BRS 37 (5.2.2). If the string is not located, the error exit is immediately taken caus- the exception return. The exception return will also be caused if the file is read only as indicated by the flag in the file directory.

Exception Return: All clobbered.
Normal Return:
- A - Location of the file in the directory hash table.
- B - Confirming character in case of a quote or slash file; otherwise, the file directory hash table value word.
- X - Clobbered.

REGISTERS AFFECTED: All

NUMBER: 19

NAME: GOFNB

FUNCTION: Open Output File Located in File Directory

STATUS: User

CALLING SEQUENCE:
```
LDA   N1
LDB   N2 (For Tape Files Only)
LDX   N3
BRS   19
EXCEPTION RETURN
NORMAL RETURN
```

N1 - Information supplied in A by BRS 18 (2.2.4) (location in the file directory).
N2 - File Size (as supplied in X by BRS 16) for tape files only.
N3 - File Type (as supplied in B by BRS 16).

DESCRIPTION: Opens an output file located in the user's file directory. The information required in the registers is indicated above. The word in A is checked for legality. If it is now a valid pointer, the exception return is taken. The exception return is also taken if the file cannot be opened for any reason, such as a physical device that cannot be used for output. In the case of a new file, the file directory entry is completed. If the new file is a DSU file and it cannot be opened, the message "NO ROOM" is typed. The message "FILE TYPE WRONG" is typed as appropriate.

```
Exception Return:      All clobbered.
Normal Return:      A - File Number.
                B & X - Clobbered.
```

REGISTERS AFFECTED: All

NUMBER: 48

NAME: GSFN

FUNCTION: Look up Input/Output File Name

STATUS: User

CALLING SEQUENCE:
```
LDP   N
BRS   48
EXCEPTION RETURN
NORMAL RETURN
```

N = String pointers for the file name.

DESCRIPTION: The file name is looked up in the file directory hash table using the BRS 5 (5.2.1). If it is not there, the exception return is taken.

Exception Return: A & B - No change.
X - Clobbered.

Normal Return: A & B - Location in file directory hash table. Can be used by BRS 16 (2.2.2) or BRS 19 (2.2.5).
X - Clobbered.

REGISTERS AFFECTED: All

NUMBER: 60

NAME: GSFI

FUNCTION: Look Up Input/Output File Name and Insert if New.

STATUS: User

CALLING SEQUENCE:
```
LDP    N
BRS    60
EXCEPTION RETURN
NORMAL RETURN
```

N = String pointers for the file name.

DESCRIPTION: The file name is looked up in the file directory hash table using the BRS 5 (5.2.1). If it is not there, it is inserted in the hash table. The exception return is taken if it cannot be inserted in the case of a full directory.

Exception Return:	A & B	- No change.
	X	- Clobbered.
Normal Return:	A & B	- String pointer to location in file directory hash table.
	X	- Clobbered.

NAME: CIO

FUNCTION: Character Input/Output

STATUS: User

CALLING SEQUENCE:
```
LDA   C (Output Only)
CIO   N
```

C = 8 bit data character right justified.
N = Address of word containing a file number.

DESCRIPTION: CIO is used to input or output a single character from, or to, a file from the A register. On input an End of Record or End of File condition will set bits 0 and 8 or bits 0 and 7 in the file number and return a 134_8 or 137_8 character, respectively. If interrupt 4 is armed (see BRS 78, "1.1.1"), it will occur. The End of Record occurs on the next input operation after the last character of the record has been input and the End of File condition occurs on the next input operation after the End of Record which signals the last record of the file. If an error occurs, bits 0 and 6 will be set in N and interrupt 4 will occur if it is armed.

WIO and BIO should not be mixed with CIO to read or write a given file.

REGISTERS AFFECTED: A

NAME: WIO

FUNCTION: Word Input/Output

STATUS: User

CALLING SEQUENCE:
```
LDA   D (Output Only)
WIO   N
```

D = Data word to be written
N = Address of word containing a file number.

DESCRIPTION: WIO is used to input or output a word of data to or from the A register. On input an End of Record condition returns a word of three 134_8 characters and sets bits 0 and 8 in the file number word. If interrupt 4 is armed (see BRS 78, "1.1.1"), it will occur. An End of File condition returns a word of three 137_8 characters and sets bits 0 and 7 in the file number word. If interrupt 4 is armed, it will occur. If an End of Record or File condition occurs with a partially filled out word, the word is completed with 134_8 or 137_8 characters. If an error occurs, bits 0 and 6 are set in N. If interrupt 4 is armed it will occur.

CIO and WIO should not be mixed to read or write a given file.

REGISTERS AFFECTED: A.

NAME: BIO

FUNCTION: Blocked Input/Output

STATUS: User

CALLING SEQUENCE:
```
LDA   W
LDX   I
BIO   N
EXCEPTION RETURN
NORMAL RETURN
```

I = Starting memory address.
W = Number of words to be read or written.
N = Address of word containing a file number.

DESCRIPTION: BIO is used to input a block of words to memory or output a block of words from memory. The A register will contain the first memory location not read into or out of at the end of the operation. If the operation is completed successfully, control will be transferred to the normal return, otherwise control will be transferred to the exception return.

On input an End of Record or End of File condition will set bits 0 and 8 or 0 and 7 in the file number. An error will set bits 0 and 6. Interrupt 4 will occur if armed when any of these bits are set.

Exception conditions are:

1. End of Record
2. End of File
3. Bad Record

If bit 1 is on in the Data Block disc address in the Index Block of a DSU file, it indicates the end of the data blocks and is the end of a logical record.

REGISTERS AFFECTED: A,X

NAME: CTRL

FUNCTION: Input/Output Control (only tape is implemented)

STATUS: System

CALLING SEQUENCE:
```
LDA   C
CTRL  N
```

C = Control number
N = File number

DESCRIPTION: CTRL provides the following control functions for tape files:

Control #	Description
1.	Write end of record on output. Record count not used.
2.	Backspace physical block.
3.	Forward space physical block.
4.	Backspace file.
5.	Erase tape (output only) (3 inches).
6.	Rewind.
7.	Write EOF. Output only.
8.	Long erase. Output only.

REGISTERS AFFECTED: None

NUMBER: 23

NAME: LNKS

FUNCTION: Link/Unlink TTY - Not implemented

STATUS: User

CALLING SEQUENCE:
```
LDX   T
LDA   A
LDB   C
BRS   23
```

T = Teletype number
A = Address of a list of teletype numbers terminated with -2.
C = Control word. The bits of this word are as follows:

Bit 0 = 0 = Output LCW, 1 = Input LCW.
Bit 1 = 0 = Clear all links first, 1 = Do not clear links first.
Bit 2 = 0 = Set link bits for TTY whose numbers are in the table.
Bit 2 = 1 = Clear link bits for TTY whose numbers are in the table.

DESCRIPTION: This BRS is used to set the link bit for TTY T in the LCW. Associated with each TTY are two words called the absolute input and absolute output link control words (LCW's). Each of these words contain one bit for each TTY in the system (maximum of 24). Also associated with each TTY are relative LCW's for input and output. The bits in these LCW's are set by this BRS. From the old relative LCW and the information supplied in the calling sequence a new relative LCW is created. Each time any relative LCW is changed, the absolute LCW's are all recomputed.

Link bits set in the input LCW cause input characters to be stored in the buffer of all TTY's linked to the controlling TTY. Link bits set in the output LCW cause output characters, including echoes, to be output to all TTY's linked to the controlling TTY.

REGISTERS AFFECTED: None

NUMBER: 24

NAME: LNKC

FUNCTION: Unlink - not implemented

STATUS: System

CALLING SEQUENCE:
```
LDX   T
BRS   24
```

T = Teletype Number

DESCRIPTION: This BRS is used to clear all links, input and output, to or from TTY T.

REGISTERS AFFECTED: None

NUMBER: 25

NAME: MSGS - Not implemented

FUNCTION: Set Accept Messages and Set Accept Input Indicators.

STATUS: System

CALLING SEQUENCE:
```
LDX   T
LDA   I
BRS   25
```

T = Teletype number (must be controlling TTY or an attached TTY).
I = Bit 23 on to set "Accept Messages" Indicator.
I = Bit 24 on to set "Accept Input" Indicator.

DESCRIPTION: This BRS allows the user to specify whether or not messages from outside will be accepted, and whether or not input from outside will be accepted for his controlling teletype or for one which he has attached. The accept message indicator controls execution of OST instructions and the setting of teletype output links. The accept input indicator controls execution of STI instructions and the setting of teletye input links. Setting or clearing of these indicators will not affect any TTY links currently set.

REGISTERS AFFECTED: None

NUMBER: 27

NAME: ASTT - not implemented

FUNCTION: Attach TTY to this program

STATUS: System

CALLING SEQUENCE:
```
LDA   T
BRS   27
EXCEPTION RETURN
NORMAL RETURN
```

T = Teletype Number

DESCRIPTION: To give total control over a TTY to the requesting program. If the indicated TTY is free, it is attached to the requesting program and transfers control to the "normal return". If it is not free, control is transferred to the "exception return".

REGISTERS AFFECTED: None

NUMBER: 28

NAME: RSTT - not implemented

FUNCTION: Release TTY

STATUS: System

CALLING SEQUENCE:
```
LDA   T
BRS   28
```

T = Teletype Number

DESCRIPTION: Returns to a free status the TTY indicated by the A register. If the TTY was not attached to the requesting program a "panic" will be executed.

NOTE: All attached teletypes are released when the user logs out.

REGISTERS AFFECTED: None

NUMBER: BE+3

NAME: CARRY

FUNCTION: Test for Carrier Presence

STATUS: System

CALLING SEQUENCE:
```
LDA    =LINE #
BRS    BE+3
EXCEPTION RETURN - No Carrier
NORMAL RETURN - Carrier Present
```

DESCRIPTION: This BRS gives a skip return, if the carrier signal is present on the line identical in A. No carrier signal - no skip.

REGISTERS AFFECTED: None

NUMBER: BE+6

NAME: TTYON

FUNCTION: Turns a Teletype Line On or Off.

STATUS: System

CALLING SEQUENCE:
```
LDA   =TTY #
LDB   =0 (off) or -1 (on)
BRS   BE+6
NORMAL RETURN
```

DESCRIPTION: Issues the EOM and POT commands which cause the line to be turned off (hung up) or made ready to accept an incoming call.

REGISTERS AFFECTED: None

NUMBER: 11

NAME: CIB

FUNCTION: Clear the Teletype Input Buffer

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   11
```

T = Teletype number (-1 is used to indicate the controlling teletype).

DESCRIPTION: Sets the buffer pointers to indicate there are no characters in the TTY input buffer.

REGISTERS AFFECTED: None

NUMBER: 29

NAME: COB

FUNCTION: Clear the Output Buffer

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   29
```

T = Teletype Number (-1 indicates the controlling TTY)

DESCRIPTION: Sets the buffer pointers to indicate there are no characters in the TTY output buffer.

REGISTERS AFFECTED: None

NUMBER: 12

NAME: CET

FUNCTION: Declare Echo Table

STATUS: User

CALLING SEQUENCE:
```
LDX   T
LDA   R
BRS   12
```

T = Teletype number (-1 is used to indicate the controlling TTY).
R = ± 1, 2, or 3 to indicate the proper echo table If the sign bit of R is set, each 8 bit character read from the teletype is transmitted unchanged to the user's program. No echoes are generated while in this special 8-level mode. Teletype output is not affected.

DESCRIPTION: BRS 12 sets the echo table for the TTY indicated by Register X. Echo tables are as follows:

0 = Echo each character just as it was received and break on all characters.
1 = Same echo as 0 but all characters except letters, digits and spaces are break characters.
2 = Same echo as 0, but the only break characters are control characters (including carriage return and line feed).
3 = No echo for any character and break on all characters.

REGISTERS AFFECTED: None

NUMBER: 40

NAME: RDET

FUNCTION: Read Echo Table

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   40
```

T = Teletype number

DESCRIPTION: Reads the echo table number (0,1,2,3) into the A register.

If the teletype is not in eight-level input mode, reads the echo table number (0,1,2,3) into the A register. If the teletype is in eight-level input mode, the sign bit of A is set, the address field contains the terminal character.

REGISTERS AFFECTED: A

NUMBER: 13

NAME: SKI

FUNCTION: Test Input Buffer for Empty

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   13
EXCEPTION RETURN
NORMAL RETURN
```

T = Teletype number (-1 is used to indicate the controlling TTY)

DESCRIPTION: This BRS tests for the presence of input characters in the buffer. If the buffer is empty, control is transferred to the "normal return". If there are any characters in the input buffer, control is transferred to the "exception return".

REGISTERS AFFECTED: None

NUMBER: 14

NAME: DOB

FUNCTION: Dismiss Until the Teletype Output Buffer is Empty.

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   14
```

T = Teletype number (-1 is used to indicate the controlling TTY).

DESCRIPTION: Dismiss this fork until the teletype output buffer indicated is empty. It is dismissed only until the last character is transmitted. This fork might be restarted before the last character interrupt has occurred, therefore, caution should be exercised.

REGISTERS AFFECTED: None

NUMBER: 85

NAME: SET8P

FUNCTION: Set Special Teletype Output

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   85
```

T = Teletype number (-1 is used to indicate controlling TTY).

DESCRIPTION: Sets teletype to 8-level output mode. The teletype specified must either be the controlling teletype or an attached teletype. Eight-level is transmitted to the teletype exactly as it is received from the user program.

REGISTERS AFFECTED: None

NUMBER: 86

NAME: CLR8P

FUNCTION: Clear Special Teletype Output

STATUS: User

CALLING SEQUENCE:
```
LDX   T
BRS   86
```

T = Teletype number (-1 is used to indicate controlling TTY).

DESCRIPTION: Puts the teletype output back into normal mode. The teletype specified must either be the controlling teletype or attached.

REGISTERS AFFECTED: None

NUMBER: BE+11

NAME: CRSW

FUNCTION: To Allow the User to Ignore Line Feed or Carriage Return when it Follows a Carriage Return or Line Feed.

STATUS: User

CALLING SEQUENCE:
```
LDA   =0 (ignore) =-1 (do not ignore)
BRS   BE+11
NORMAL RETURN
```

DESCRIPTION: The contents of the A register will give the following results. If A is negative, all line feeds and carriage returns received from the TTY will be sent to the program and echoed. If A is positive, a line feed after a carriage return received from the TTY will be ignored (not sent to the program and not echoed) and a carriage return after a line feed will be ignored (not sent to the program and not echoed). In all cases the first line feed or carriage return received will be sent to the program and echoed plus echo it's compliment.

REGISTERS AFFECTED: None

NAME: TCI

FUNCTION: Teletype Character Input

STATUS: User

CALLING SEQUENCE: TCI M

M = Memory location.

DESCRIPTION: This SYSPOP reads the character from the teletype input buffer and places it into the location M right justified. The remainder of location M is cleared. The character is also placed in the A register right justified.

REGISTERS AFFECTED: A

NAME: TCO

FUNCTION: Teletype Character Output

STATUS: User

CALLING SEQUENCE: TCO M

M = Memory address.

DESCRIPTION: This SYSPOP outputs the character from the right-most eight bits of location M to the controlling teletype. In addition to the ordinary ASCII characters, all teletype output operations will accept 135_8 as a multiple blank character. The next character will be taken as a blank count, and the indicated number of blanks will be typed.

REGISTERS AFFECTED: None

NAME: IST (Not implemented)

FUNCTION: Input From Specified Teletype

STATUS: User

CALLING SEQUENCE: IST T

T = Teletype number

DESCRIPTION: IST is used to input a character from an attached teletype. The character will be right justified in the A register upon return.

REGISTERS AFFECTED: None

NAME: OST (Not implemented)

FUNCTION: Output to Specified Teletype

STATUS: User

CALLING SEQUENCE: OST T

T = Teletype number

DESCRIPTION: OST is used to output a character in the A register to a specified teletype. This instruction is used for output to an attached teletype. It's accept message bit must be set or an illegal instruction panic will be generated.

REGISTERS AFFECTED: None

NAME: STI

FUNCTION: Simulate Teletype Input

STATUS: User

CALLING SEQUENCE: STI T

T = Teletype number

DESCRIPTION: This BRS is used to simulate teletype input. It puts the character in the A register into the input buffer of the specified teletype. It is legal for a user fork only if T equals the controlling TTY or -1.

REGISTERS AFFECTED: None

NUMBER: 4

NAME: MPT

FUNCTION: Release a Page of Memory

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   4
```

N = Contains any address in the page to be released.

DESCRIPTION: The PMT entry for the block is removed and in any other fork which has this PMT byte in its relabeling, the byte is cleared to 0.

REGISTERS AFFECTED: None

NUMBER: 121

NAME: DPMTE

FUNCTION: Release Specified PMT Entry

STATUS: User

CALLING SEQUENCE:
```
LDA   R
BRS   121
```

R = Relabeling byte

DESCRIPTION: Releases the specified page from the PMT. It is exactly like a BRS 4 (4.1.1) except that it takes a byte number instead of an address.

Instruction Trap:

1) Byte not in PMT.
2) A user fork tried to release a system page.

REGISTERS AFFECTED: None

NUMBER: 120

NAME: APMTE

FUNCTION: Assign PMT Entry

STATUS: System

CALLING SEQUENCE:
```
LDA   R
BRS   120
```

R = Relabeling byte

DESCRIPTION: Obtains a new page for the relabeling byte specified. This BRS is used only in the recover routine in the exec.

Instruction Trap:

1) PMT entry is already assigned.
2) The relabeling byte number was not in the PMT.

REGISTERS AFFECTED: None

NUMBER: 43

NAME: RDRL

FUNCTION: Read Pseudo-Relabeling

STATUS: User

CALLING SEQUENCE: BRS 43

DESCRIPTION: Reads the current pseudo-relabeling registers into registers A and B.

REGISTERS AFFECTED: A,B

NUMBER: 44

NAME: STRL

FUNCTION: Set Pseudo-Relabeling

STATUS: User

CALLING SEQUENCE:
```
LDA   R1
LDB   R2
BRS   44
```

R1 & R2 = Relabeling factors

DESCRIPTION: This BRS takes the contents of registers A and B and stores them into the current pseudo-relabeling registers. It also causes the real relabeling to be reset to correspond to the new pseudo-relabeling.

This BRS will result in an instruction trap for any of the following reasons:

1) Swapping in the new pages was not completed. (Usually because of a RAD failure.)
2) The user tried to relabel over a system page.
3) The user tried to relabel over a page he did not have. (This is not the way to obtain more memory.)

REGISTERS AFFECTED: None

NUMBER: 116

NAME: RURL

FUNCTION: Read User Relabeling

STATUS: System

CALLING SEQUENCE: BRS 116

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

NUMBER: 117

NAME: SURL

FUNCTION: Set User Relabeling

STATUS: System

CALLING SEQUENCE:
```
LDA   RL1
LDB   RL2
BRS   117
```

RL1 and RL2 are the new values for the program relabeling.

DESCRIPTION: Sets the program relabeling as specified. This BRS is used by the system. 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.

REGISTERS AFFECTED: None

NUMBER: 122

NAME: MPAN

FUNCTION: Simulate Memory Panic

STATUS: System

CALLING SEQUENCE:
```
LDA   A
BRS   122
```

A = Core address

DESCRIPTION: This BRS gets new memory for a class 3 BRS. If it succeeds the new memory is put into the relabeling of the calling program. Can be issued from a class 3 BRS only.

If a memory trap occurs, it looks to the calling program like it came from the BRS instruction.

REGISTERS AFFECTED: None

NUMBER: 56

NAME: MBEX

FUNCTION: Make Page System

STATUS: System

CALLING SEQUENCE:
```
LDA   P
BRS   56
```

P = Pseudo-Relabeling byte for page.
If bit 0 of A = 1, page will be made system.
If bit 0 of A = 0, page will be made not system.

DESCRIPTION: Sets the use mode of a page depending on the value of bit 0 in the A register.

Bit 0 of A is set to 1 if page was formerly system or 0 if it was not.

REGISTERS AFFECTED: A

NUMBER: 80

NAME: MBRO

FUNCTION: Make Page Read Only

STATUS: User

CALLING SEQUENCE:
```
LDA   P
BRS   80
```

P = PMT/SMT number
If bit 0 of A = 1, make page read only.
If bit 0 of A = 0, make page read-write.

DESCRIPTION: Sets the read-write status of the entry according to the value of A. An SMT entry can only be changed by a system fork. The former status of the entry is returned in A.

Instruction trap:

1) Specified entry is not in use.
2) The swap failed.

REGISTERS AFFECTED: A

NUMBER: BE+5

NAME: PEBRS

FUNCTION: Reads or Sets One Word in the Monitor

STATUS: System

CALLING SEQUENCE:
```
LDA   V
LDB   0 or -1
LDX   =Location in Monitor Relabeling
BRS   BE+4
RETURN
```

V = New value for word if it is to be set.
The contents of the location are returned in the A register.
If B is positive, the word is read.
If B is negative, the word is changed and the old value returned in A.

DESCRIPTION: Allows a system program to read or set the contents of any location in the monitor relabeling.

The original contents of the location are always returned in the A register.

REGISTERS AFFECTED: A

NUMBER: 68

NAME: EBSM

FUNCTION: Enter Block in SMT - Not implemented.

STATUS: System

CALLING SEQUENCE:
```
LDA   B
BRS   68
```

B = Byte number in users pseudo-relabeling

DESCRIPTION: A free SMT entry is found and the PMT entry put into it. The SMT number is returned in A.

REGISTERS AFFECTED: A

NUMBER: 69

NAME: GBSM

FUNCTION: Get SMT Block to PMT

STATUS: Subsystem

CALLING SEQUENCE:
```
LDA   S
BRS   69
```

S = SMT number

DESCRIPTION: Puts the SMT entry into the first free PMT entry. The PMT entry number is returned in A.

Instruction trap:

1) A user program tries to relabel a system SMT entry.
2) The SMT number is not valid.

Memory trap:

There were no free PMT entries.

REGISTERS AFFECTED: A

NUMBER: 33

NAME: GETSTR

FUNCTION: Read String

STATUS: User

CALLING SEQUENCE:
```
LDA   A
LDB   T
LDX   T
BRS   33
```

A = Address of string pointer
T = Terminal character
F = File number
Bit 0 of A on = The string is taken as null with the second pointer equal to the first.

DESCRIPTION: This BRS reads characters from the file and appends them to the string until the terminal character is reached. The terminal character is not appended to the string. It returns the updated string pointers in the A and B registers and updates the end string pointer in memory.

REGISTERS AFFECTED: A,B

NUMBER: 34

NAME: OUTMSG

FUNCTION: Output Message

STATUS: User

CALLING SEQUENCE:
```
LDX   F
LDA   W
LDB   C
BRS   34
```

F = File number
W = Beginning word address
C = Character count or -1

DESCRIPTION: This BRS outputs C consecutive characters starting with the first character of the specified word. If B = -1, characters are output until a / is encountered; the character $ is interpreted as a carriage return and line feed.

REGISTERS AFFECTED: None

NUMBER: 35

NAME: OUTSTR

FUNCTION: Output String

STATUS: User

CALLING SEQUENCE:
```
LDX   F
LDA   P
LDB   P+1
BRS   35
```

F = File number
P, P+1 = A string pointer pair

DESCRIPTION: Outputs the string indicated by the string pointers in registers A and B to the specified file.

REGISTERS AFFECTED: None

NUMBER: BE+14

NAME:

FUNCTION: Input String with Edit

STATUS: User

CALLING SEQUENCE:

NAME: CIT

FUNCTION: Character Input and Test

STATUS: User

CALLING SEQUENCE:
```
LDA   N
CIT   F
EXCEPTION RETURN
NORMAL RETURN
```

N = Character to be tested.
F = File Number (see CIO) (Input Only)

DESCRIPTION: The character in the A register is compared against the next character in the input file. If it compares, the normal return is taken and the character is removed from the input buffer. If it does not compare, the character is left in the input buffer and is returned in A.

Exception Return:	A -	The next character in the input buffer.
	B & X -	No change
Normal Return:	A -	The character supplied remains in A (the character is removed from the input buffer).

REGISTERS AFFECTED: A

NUMBER: 5

NAME: SSCH

FUNCTION: Look Up String in Hash Table

STATUS: User

CALLING SEQUENCE:
```
LDA   P
LDB   P+1
LDX   T
BRS   5
EXCEPTION RETURN
NORMAL RETURN
```

P and P+1 = String pointers for a string to be looked up.
T = Address of a three word table of the form:

```
ZRO Hash Table Beginning Address
ZRO Hash Table End Address
ZRO 0
```

DESCRIPTION: BRS 5 (5.2.1) searches the hash table for a string to match the string indicated by A and B registers. If successful it returns in register B the address of the hash table string pointers, and in register A, the string "value" and executes the "normal" return. Otherwise, it executes the "exception" return with registers A, B and X unchanged and the address of the next free hash table entry in word 3 of the table is pointed to by register X. (Word 3 will be -1 if the table is full.) The "value" is the hash image for this string.

See BRS 6 (5.2.3)

REGISTERS AFFECTED: A,B

NUMBER: 37

NAME: GSLOOK

FUNCTION: General String Lookup

STATUS: User

CALLING SEQUENCE:
```
LDA   F
LDB   S
LDX   T
BRS   37
EXCEPTION RETURN
NORMAL RETURN
```

F = Input file number
S = Address of string pointer pair.
T = Address of the Hash Table Control Table.

DESCRIPTION: The hash table is scanned for a string to match the given one. If none is found but the given string matches the initial part of some hash table string characters from the input file are appended until the string is long enough either to determine a unique hash table string, with a matching initial part, or for no match to be possible. In the former case, more characters are taken from input until an exact match is obtained or no match is possible; in this latter case, the match is still valid, and the last character (which caused the mis-match) is left in the input file.

Exits are as follows: (1) The exception return is taken on the no-match condition with a string pointer in A, B to the string so far collected. X is undisturbed. (2) The normal return is taken on a match with the address of a hash table string pointer in A and the string "value" in B. X is undisturbed.

The "value" is the hash image for the string.

See also, Berkeley Document No. 30.10.20 for more details.

REGISTERS AFFECTED: None

NUMBER: 6

NAME: SSIN

FUNCTION: Insert String in Hash Table

STATUS: User

CALLING SEQUENCE: A,B, & X must have the output from BRS 5
BRS 6

DESCRIPTION: BRS 6 inserts the string pointer into the hash table at the point determined by the last BRS 5 (5.2.1) which did not find a match. If the hash table is full (word 3 of the table pointed to by X is -1) an "Illegal Instruction" trap results. BRS 6 is intended for use in conjunction with BRS 5. It should be used only after BRS 5 has failed to find a match. Furthermore, string pointers should not be placed in the hash table in any manner other than with BRS 6 (otherwise the scanning algorithm used in BRS 5 may cause undesired results).

BRS 6 does not physically move the string to which registers A and B point. On return, register B contains the address of the first word of the new hash table entry and register A containing the "value" word of the entry.

REGISTERS AFFECTED: A,B

NAME: STP

FUNCTION: Store Pointers

STATUS: User

CALLING SEQUENCE: STP A

A = Address of a string pointer pair.

DESCRIPTION: This SYSPOP is generally used in conjunction with LDP. It stores the contents of the A and B registers into the string pointers indicated in the calling sequence.

REGISTERS AFFECTED: None

NAME: LDP

FUNCTION: Load Pointers

STATUS: User

CALLING SEQUENCE: LDP A

A = Address of a string pointer pair.

DESCRIPTION: This SYSPOP loads the string pointers indicated in the calling sequence into the A & B registers.

REGISTERS AFFECTED: None

NAME: SKSE

FUNCTION: Skip on String Equal

STATUS: User

CALLING SEQUENCE:
```
LDA   B
LDB   E
SKSE   A
EXCEPTION RETURN
NORMAL RETURN
```

A = Address of a string pointer pair.
B = Beginning string pointer.
E = End string pointer.

DESCRIPTION: If the string addressed by the pointers in the A and B registers is identical with the string addressed by A of the calling sequence, control will be transferred to the normal return. Otherwise, control will be transferred to the exception return. If the strings are of different lengths or have different contents, control will be transferred to the exception return.

REGISTERS AFFECTED: None

NAME: SKSG

FUNCTION: Skip on String Greater

STATUS: User

CALLING SEQUENCE:
```
LDA   B
LDB   E
SKSG   A
EXCEPTION RETURN
NORMAL RETURN
```

B = Beginning string pointer
E = End string pointer
A = Address of a string pointer pair.

DESCRIPTION: The SYSPOP compares the string indicated by A and B registers with the string indicated by A of the calling sequence, character by character and terminates with the first unequal character. The numerical internal code representation of characters is used to determine inequality. If the strings are unequal for the entire length of the shorter one, the longer one is indicated as greater. If the contents of the string addressed by the A and B registers is greater than the contents of the string addressed by A, control will be transferred to the normal return. Otherwise, control is transferred to the exception return.

REGISTERS AFFECTED: None

NAME: GCI

FUNCTION: Get Character and Increment

STATUS: User

CALLING SEQUENCE: GCI A
EXCEPTION RETURN
NORMAL RETURN

A = Address of a string pointer pair.

DESCRIPTION: This SYSPOP reads into the A register, the first character from the string indicated by the beginning string pointer given in the calling sequence. If the string is null or empty, nothing is done and control is transferred to the exception return. If the string is not null it's first character is loaded into the A register right-justified, and the beginning string pointer is incremented by one such that the beginning string pointer now points to the string with the first character deleted. Control is transferred to the normal return. Unless a copy of the original pointer is saved, the contents of the string are effectively destroyed.

REGISTERS AFFECTED: A

NAME: WCI

FUNCTION: Write Character and Increment

STATUS: User

CALLING SEQUENCE: WCI P

P = Address of string pointer pair.

DESCRIPTION: WCI writes the character in the A register on the end of the string addressed by the end string pointer. The end string pointer is incremented by 1.

REGISTERS AFFECTED: B

NAME: GCD

FUNCTION: Get Character & Decrement

STATUS: User

CALLING SEQUENCE: GCD P
EXCEPTION RETURN
NORMAL RETURN

P = Address of a string pointer pair.

DESCRIPTION: A GCD is, in every way, similar to GCI except that the character is taken from the end of the specified string.

The last character on the string is loaded in the A register, and end string pointer is decremented so that it points to the previous character in the string. Control is transferred to the exception return if the end pointer is not greater than the beginning pointer before it is decremented.

REGISTERS AFFECTED: N

NAME: SCH

FUNCTION: Write Character

STATUS: User

CALLING SEQUENCE:
```
LDA   C
WCH   T
```

C = A character right-justified in the A register.
T = The address of a three word table. The table is as follows:

Word 0 = A character address
Word 1 = A character address
Word 2 = A transfer address

DESCRIPTION: This SYSPOP tries to write a character into the area defined by the character addresses in the table. Provided that the second address in the table is greater than the first address, WCH will write the character in A register into the character position indicated by the first character address plus one and will increment the first character address in the table. If the first character address is equal to or greater than the second character in the table the character is not written and control is transferred to the third word of the table with A and X registers undisturbed and the address of the WCH in the B register. The address in the third word of the table can be an exit to a routine which allocates more memory or garbage collects the remaining characters.

REGISTERS AFFECTED: None

NAME: WCD

FUNCTION: Writes Character and Decrement

STATUS: User

CALLING SEQUENCE: WCD P

P = Address of a string pointer pair.

DESCRIPTION: This SYSPOP writes the character in the A register on the beginning of the string and decrements the beginning string pointer.

REGISTERS AFFECTED:

NUMBER: 36

NAME: OUTNUM

FUNCTION: Output number

STATUS: User

CALLING SEQUENCE:
```
LDX   F
LDA   N
LDB   R
BRS   36
```

F = File number
N = Number to be output
R = Radix

DESCRIPTION: Outputs a number in the radix R. The number will be output as an unsigned 24 bit integer. If the radix is less than 2, an instruction trap will be given.

REGISTERS AFFECTED: None

NUMBER: 38

NAME: GETNUM

FUNCTION: Read Number

STATUS: User

CALLING SEQUENCE:
```
LDX   F
LDB   R
BRS   38
```

F = File number
R = Radix

DESCRIPTION: Inputs an integer to any radix. The number may be preceded by a plus or minus sign. On exit the number will be in the A register. The conversion is terminated by any non-numeric character which will be in the B register on exit. The number is computed by multiplying the number obtained at each stage by the radix and adding the new digit.

REGISEERS AFFECTED: A,B

NUMBER: 52

NAME: FFI

FUNCTION: Formatted Input

STATUS: User

CALLING SEQUENCE:
```
LDX   FORMAT
BRS   52
BRU   X
```

DESCRIPTION: This routine reads characters from a file specified in the format work, FORMAT. FORMAT also specifies the format of the input. Free form input from the teletype results when FORMAT = 0. A skip return is generated if and only if (1) the input is free form, and (2) the input is floating point. The internal translation of the input file is stored in A,B.

REGISTERS AFFECTED: A,B,X

NUMBER: 53

NAME: FFO

FUNCTION: Formatted Output

STATUS: User

CALLING SEQUENCE:
```
LDX   FORMAT
BRS   53
```

DESCRIPTION: The integer in A or the double word floating point value in A,B is output to the file according to the file number and format specified in FORMAT.

REGISTERS AFFECTED: None

NAME: SIC

FUNCTION: String to Internal Conversion

STATUS: User

CALLING SEQUENCE:

```
LDX    FORMAT
SIC    POINTER
BRU    INTEGER
BRU    FLOATING
```

DESCRIPTION: See String Processing System documents. FORMAT describes the type of conversion to be done.

The contents of POINTER point to the character immediately preceding the character string. POINTER+1 contains the character address of the last character of the string.

INTEGER and FLOATING are routines that handle the converted input. Error flags, if applicable, are in the index register A, double word value corresponding to the string is in A,B upon return.

REGISTERS AFFECTED: A,B,X

NAME: ISC

FUNCTION: Converts Internal Numbers to Formatted Output Strings

STATUS: User

CALLING SEQUENCE:
```
LDP   M
LDX   FORMAT
ISC   POINTER
```

DESCRIPTION: See String Processing Documents. FORMAT describes the type of conversion to be done. The contents of POINTER point to the character immediately preceding the character string. POINTER+1 contains the character address of the character immediately preceding the position where the first character of output is to go. M,M+1 contain the floating point word to be converted. Pointer+1 is incremented once for each character added to the string.

REGISTERS AFFECTED: A,B,X

NUMBER: 50

NAME: FFIX

FUNCTION: Conversion from Floating Point to Fixed Point.

STATUS: User

CALLING SEQUENCE: BRS 50

DESCRIPTION: Fixes the double word floating point value in (A,B). The integer part is left in A. The fractional part is left adjusted in B.

REGISTERS AFFECTED: A,B

NUMBER: 51

NAME: FFLT

FUNCTION: Conversion from Fixed Point to Floating Point

STATUS: User

CALLING SEQUENCE: BRS 51

DESCRIPTION: The integer in A is converted to a normalized floating point value in A,B.

REGISTERS AFFECTED: A,B

NUMBER: 21

NAME: FNA

FUNCTION: Floating Negate

STATUS: User

CALLING SEQUENCE: BRS 21

DESCRIPTION: The double word floating point value in the A and B registers is negated.

REGISTERS AFFECTED: A,B

NAME: FAD

FUNCTION: Floating Point Addition

STATUS: User

CALLING SEQUENCE: FAD N

DESCRIPTION: SYSPOP FAD (A,B)+(M,M+1)
A floating addition is performed to the contents of memory location M and M+1 and the A and B registers. The results are left in the A and B registers.

REGISTERS AFFECTED: A,B

NAME: FSB

FUNCTION: Floating Point Subtraction

STATUS: User

CALLING SEQUENCE: FSB N

DESCRIPTION: (A,B) - (M,M+1)
The contents of memory locations M and M+1 are subtracted (floating subtraction) from the contents of the A and B registers. The results are left in the A and B registers.

REGISTERS AFFECTED: A,B

NAME: FMP

FUNCTION: Floating Point Multiplication

STATUS: User

CALLING SEQUENCE: FMP M

DESCRIPTION: (A,B)*(M,M+1)
The contents of memory locations M and M+1 are multiplied (floating multiplication) by the A and B registers and the results left in the A and B registers.

REGISTERS AFFECTED: A,B

NAME: FDV

FUNCTION: Floating Point Divide

STATUS: User

CALLING SEQUENCE: FDV M

DESCRIPTION: (A,B)/(M,M+1)

The contents of the A and B registers are divided (floating divide) by the contents of memory locations M and M+1 with the quotient left in the A and B registers.

REGISTERS AFFECTED: A,B

NUMBER: 95

NAME: ECDUMP

FUNCTION: Dump

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   95
```

N = File number

DESCRIPTION: This BRS writes the entire current state of the machine (user's program only) on the specified file, which is made type 4. The status of the pseudo-relabeling registers and all information necessary to restart the user from his current situation are written on thd dump file so it can be restored by a recovery procedure. The only information not preserved are any shared memory entries which may be in the pseudo-relabeling.

NOTE: Dumps created by one system cannot be recovered by another, i.e., 1.81 to 1.85.

REGISTERS AFFECTED: All

NUMBER: 96

NAME: ECRECV

FUNCTION: Recover

STATUS: User

CALLING SEQUENCE:
```
LDA   N
BRS   96
```

N = File number

DESCRIPTION: This BRS reads the dump file written by a BRS 95 and recovers the machine status as it appeared at the time the dump was taken.

REGISTERS AFFECTED: All

NUMBER: 42

NAME: RREAL

FUNCTION: Read Real-Time Clock

STATUS: User

CALLING SEQUENCE: BRS 42

DESCRIPTION: Read the real-time clock in the A register. Time is given as a 24 bit binary number representing 60ths of a second. The clock is set to zero when the system is started and it is incremented by one at every 1/60th second. A binary form of the month, date and start-up time is put in B. From A and B the user can calculate the month, date and time.

REGISTERS AFFECTED: A,B

NUMBER: 91

NAME: EXRTIM

FUNCTION: Read Date and Time into a String

STATUS: User

CALLING SEQUENCE:
```
LDA   S
LDB   S+1
BRS   91
```

S = Beginning string pointer.
S+1 = 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 hh:mm

MM=Month
dd=Day
hh=Hours counted from 0 to 24
mm=Minutes

REGISTERS AFFECTED: None

NUMBER: 88

NAME: RTEX

FUNCTION: Read Execution Time

STATUS: System

CALLING SEQUENCE: BRS 88

DESCRIPTION: Returns the execution time for the job in A.

REGISTERS AFFECTED: A

NUMBER: 41

NAME: IORET

FUNCTION: Return from I/O Subroutine

STATUS: User

CALLING SEQUENCE: BRS 41

DESCRIPTION: This is used by the author of an I/O subroutine to return to the calling program.

REGISTERS AFFECTED: A

NUMBER: 111

NAME: BRSRET

FUNCTION: Return from Class 3 BRS

STATUS: System

CALLING SEQUENCE: BRS 111

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

NUMBER: 112

NAME: TSOFF

FUNCTION: Turn Off Teletype Station

STATUS: System

CALLING SEQUENCE:
```
LDA    Job Number
BRS    112
```

DESCRIPTION: This BRS is known as suicide. The job disappears completely from the system.

The teletype line associated with the job will be set ready for another job if he merely logged out.

REGISTERS AFFECTED: All

NUMBER: 71

NAME: SKXEC

FUNCTION: Skip if System

STATUS: User

CALLING SEQUENCE: BRS 71

DESCRIPTION: The B register is set to the value of the use code which the user has set for the job. These values are:

Value of B	Use Code
1	Subsystem User
0	User
-1	Both
-2	System

The BRS skips if the B register is negative.

REGISTERS AFFECTED: B

NUMBER: BE+5

NAME: SDBM

FUNCTION: Set Disc Bit Map

STATUS: System

CALLING SEQUENCE:
```
LDA    =Address of X Block Mod 4
BRS    BE+5
EXCEPTION RETURN
NORMAL RETURN
```

Exception Return - A contains address that was in conflict.

DESCRIPTION: Turns off bits in the disc bit map for the X block and each data block referenced by the index block. If any conflicts occur (the bit is already off), the address is left in the A register and the exception return is taken. A conflict also increments one of two counters, XBERR or FDERR, for errors in the X block or the file directory respectively.

When the bit map has been set, one more call is made to this BRS with A negative. At that time a switch is set allowing output files to be opened; the new overflow pointer is set from B and the accounting area pointer is set from X.

REGISTERS AFFECTED: A

NUMBER: BE+8

NAME: CRASH

FUNCTION: To Crash the System

STATUS: System

CALLING SEQUENCE: BRS BE+8
NO RETURN

DESCRIPTION: Saves the registers in 5501, 5502, 5503. Saves 0 in MCRO. Turns off the clock and disables the interrupts. Moves the TS block into real page 7 and the current relabeled page into real page 6.

REGISTERS AFFECTED: None

NUMBER: BE+13

NAME: SETSW

FUNCTION: Sets System Exec Switches in SYMS

STATUS: System

CALLING SEQUENCE:
```
LDA   V
LDX   N
BRS   BE+13
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.

REGISTERS AFFECTED: A

NUMBER: 152

NAME: EXS

FUNCTION: Execute Instruction in System Mode

STATUS: System

CALLING SEQUENCE: EXS I

I = Address of the instruction to be executed.

DESCRIPTION: This SYSPOP will cause the instruction pointed to by I to be executed in the system mode.

REGISTERS AFFECTED: Depends on instruction.

9.0 INDEX OF BRS'S AND SYSTEM OPERATORS

9.1 BRS's

24	3.1.2	Unlink All TTY's
25	3.1.3	Set Teletype to Accept/Refuse Links
26	1.2.4	Skip if Escape Waiting
27	3.1.4	Attach TTY to Calling Program
28	3.1.5	Release Attached TTY
29	3.2.2	Clear the Output Buffer
30	1.4.1	Read Status of a Lower Fork
31	1.5.4	Wait for Specific Fork to Cause a Panic
32	1.6.3	Terminates a Specified Lower Fork
33	5.1.1	Read String
34	5.1.2	Output Message
35	5.1.3	Output String
36	6.1.1	Output Number to Specified Radix
37	5.2.2	General String Look Up
38	6.1.2	Input Number to Specified Radix
40	3.2.4	Read Echo Table
41	8.4	Return from I/O Subroutine
42	8.1	Read Real-Time Clock
43	4.1.4	Read Pseudo Relabeling
44	4.1.5	Set Pseudo Relabeling
45	1.5.1	Dismiss on Quantum Overflow
46	1.2.2	Turn Escape Off
47	1.2.3	Turn Escape On
*48	2.2.6	Look Up Input/Output File Name

49	1.1.3	Read Interrupts Armed
50	6.2.1	Conversion from Floating Point to Fixed Point
51	6.2.2	Conversion from Fixed Point to Floating Point
52	6.1.3	Formatted Floating Point Input
53	6.1.4	Formatted Floating Point Output
56	4.1.9	Make Page System
57	1.3.2	Guarantee 16ms Computing
*60	2.2.7	Look up I/O File Name and Insert in File Directory if not Found
66	2.1.6	Delete DSU File Data
67	2.1.7	Delete DSU File Index Block
68	4.2.1	Make Pseudo-Page Shareable
69	4.2.2	Get SMT Block to PMT
72	1.5.2	System Dismissal
73	1.6.4	Terminates a Specified Number of Lower Forks
78	1.1.1	Arm/Disarm Software Interrupts
79	1.1.2	Cause Specified Software Interrupts
80	4.1.10	Make Page Read Only
81	1.5.3	Dismiss for Specified Amount of Time
82	2.1.8	Switch Sequential File Type
85	3.2.7	Set Special TTY Output
86	3.2.8	Clear Special TTY Output
87	2.1.9	Read DSU File Index Block

88	8.3	Read Execution Time
90	1.2.1	Declare a Fork for Escape
91	8.2	Read Date and Time into a String
*95	7.1	Dump Program and Status on File
*96	7.2	Recover Program and Status from File
104	2.1.10	Read a Page (2048 words) from RAD
105	2.1.11	Write a Page (2048 words) to RAD
106	1.5.5	Wait for Any Fork to Terminate
107	1.4.2	Read Status of all Lower Forks
108	1.6.5	Terminate All Lower Forks
109	1.5.6	Dismiss Calling Fork
110	2.1.2	Read Device and Unit
111	8.5	Return from Exec BRS (Exec Only)
112	8.6	Turn Off Teletype Station (Exec Only)
113	2.1.12	Compute File Size of a Disc File
114	2.1.13	Turn Off Run-Away Magnetic Tape
116	4.1.6	Read User Relabeling
117	4.1.7	Set User Relabeling
118	2.1.14	Allocate Magnetic Tape Unit
119	2.1.15	De-Allocate Magnetic Tape Unit
120	4.1.3	Acquire a New Page
121	4.1.2	Release Specified Page from PMT
122	4.1.8	Simulate Memory Panic
(123 BE+1	2.1.19	Read DSU
BE+2	2.1.20	Write DSU

BE+3	3.1.6	Test for Carrier Present
BE+4	4.1.11	Read/Write One Word in the Monitor
BE+5	8.8	Set Disc Bit Map
BE+6	3.1.7	Turn a Teletype Line On or Off
BE+7	2.1.18	Test a Breakpoint Switch
BE+8	8.9	To Crash the System for Error Diagnostic
BE+9	2.1.16	Read DSU Page
BE+10	2.1.17	Write DSU Page
BE+11	3.2.9	Ignore Line Feed or Carriage Return When Followed by Carriage Return or Line Feed Respectively
BE+12	1.1.4	Arm Timing Interrupt
BE+13	8.10	Sets System Exec Switches in SYMS
BE+14	5.1.4	Input String with Edit

9.2 SYSTEM OPERATORS

BIO	2.3.2	Block Input/Output
CIO	2.3.1	Character Input/Output
CIT	5.1.5	Character Input and Test
CTRL	2.3.4	Input/Output Control
EXS	8.11	Execute Instruction in System Mode
FAD	6.2.4	Floating Point Addition
FDV	6.2.7	Floating Point Division
FMP	6.2.6	Floating Point Multiplication
FSB	6.2.5	Floating Point Subtract
GCD	5.4.3	Get Character from End of String and Decrement End Pointer

GCI	5.4.1	Get Character from Beginning of String and Increment Beginning Pointer
ISC	6.1.6	Internal to String Conversion
IST	3.2.12	Input from Specific TTY
LDP	5.3.2	Load String Pointer
OST	3.2.13	Output to Specific TTY
SKSE	6.3.3	Skip if String Equal
SKSG	5.3.4	Skip if String Greater
SIC	6.1.5	String to Internal Conversion
STI	3.2.14	Simulate TTY Input
STP	5.3.1	Store String Pointer
TCI	3.2.10	Teletype Character Input
TCO	3.2.11	Teletype Character Output
WCD	5.4.4	Put Character on Beginning of String and Decrement Beginning Pointer
WCH	5.4.5	Write Character to Memory by Table
WCI	5.4.2	Put Character on End of String and Increment End Pointer
WIO	2.2.3	Word Input/Output

Those BRS's marked with an asterisk are executive BRS's and all others are monitor BRS's.

10.0 APPENDIX A, GLOSSARY OF TERMS

BREAKPOINT SWITCH

Refers to the four toggle switches physically located on the computer console.

COMMAND FILE

The particular file from which the commands to the System Executive and Subsystems are input. For teletype input the command file number is zero.

CUSTOMER FILE DIRECTORY

The names of all files for a particular user are recorded in this directory.

DEVICE TABLE

Device	Number
Paper Tape Input	1
Paper Tape Output	2
Magnetic Tape Input	4
Magnetic Tape Output	5
Hollerith Card Output	6
Binary Card Output	7
High Speed Printer Output	11
Hollerith Card Input	12
Binary Card Input	13

DSU BLOCK

Four consecutive sectors on the disc whose beginning addresses are MOD 4. A block consists of 256 words.

DSU DATA BLOCK

A DSU block with pointers in the first and second words. The first word points to the first relevant data word. The second word points to the last relevant data word.

DSU FILE

A file stored on the Disc Storage Unit. Each file consists of at least an Index Block and if the file contains data, then a sufficient number of DSU blocks to record the data.

FILE NUMBER

A file number is assigned by the system to files as they are opened. Also, there are fixed file numbers for certain devices. These are as follows:

0	Teletype Input
1	Teletype Output
2	Nothing

FILE TYPE

There are four standard file types. They are as follows:

1 File written by the System Executive as commanded by the "SAVE" command.

2 General Binary File created by a subsystem, i.e., a FORTRAN Object Program.

3 Symbolic File

4 Dump File

FORK

A fork is all or part of a program. A program may consist of many forks and these forks may be in a heirarchy one to another. Forks are different from subroutines in that all forks making up a program could be theoretically executing simultaneously. At least one fork is associated with each active user in the system.

FORK STATES

-2	Dismissed for I/O.
-1	Running.
0	Dismissed on escape key or programmed panic.
1	Dismissed on illegal instruction panic.
2	Dismissed on memory panic.

INDEX BLOCK

A DSU Block (256 words) which contains the DSU addresses for all data blocks of a file. Words 0 through 120 contain a DSU address which is MOD 4 in bits 6 to 23. Bits 0 and 5 of these words are unused. Bit 2 indicates an End of Record data block.

INDEX BLOCK (cont.)

Words 121 and 122 are link pointers and 123 is a hash total. Words 124 through 130 contain the file name and word 131 contains user numbers.

PAC TABLE

Each fork is defined by a program active table. This table contains most of the information required to control selection, execution and interruption of the fork (additional information is stored in the users TS page).

PAGE

A page can exist on RAD, DSU or in-core memory but in all cases refers to 2048 words.

PANIC

A panic is a signal to the system to break execution of a fork.

PANIC, INSTRUCTION

A panic caused by attempting to execute an instruction which cannot be executed in the user mode, such as a halt or device I/O instruction or a BRS which is not available to the user.

PANIC, MEMORY

A panic caused by a fork attempting to address memory outside its range or write on memory which is set to read only.

PANIC TABLE

Word
0 = Program Counter
1 = A Register
2 = B Register
3 = X Register
4 = First Relabeling Register
5 = Second Relabeling Register
6 = Status

PANIC TABLE (cont.)

The status word may be:

-2 Dismissed for Input/Output
-1 Running
0 Dismissed on Escape or BRS 10
1 Dismissed on Illegal Instruction Panic
2 Dismissed on Memory Panic

A Panic Table must not overlap a page boundary.

QUANTUM, LONG TIME

The maximum length of time a fork can run before the schedule checks for other forks to be run.

QUANTUM, SHORT TIME

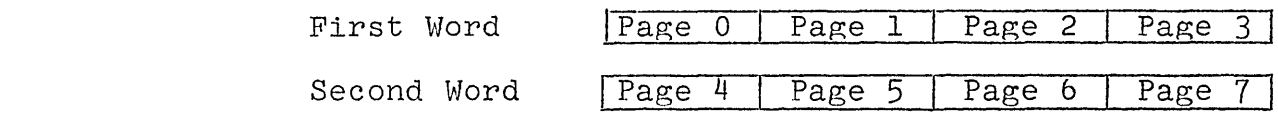
The minimum length of time a fork will run before the scheduler checks for other forks to be run which were dismissed for I/O.

RELABELING, PSEUDO

See Relabeling Registers

RELABELING REGISTERS

The relabeling registers are used to indicate a page number which has been assigned to a user for a particular logical page. They are of the form:



STRING POINTERS

A pair of pointers which contain a character address of the character before the first character of a string and a character address of the last character of the string.

STRING, NULL

A pair of string pointers whose character addresses are the same.