

IDENTIFICATION

Product Code:	DEC-12-ZR8A-D
Product Name:	DIAL-MS File Commands Program Description
Date Created:	July 1, 1970
Maintainer:	Software Services

LAP6-DIAL is an editor, filing system and assembler for use with the PDP-12 computer. The editor and filing portions are derived from the basic LINC program LAP6¹ by Mary Allen Wilkes of Washington University. The assembly portion is derived from several programs used for the PDP-8 computer including PAL-D².

The Digital Equipment Corporation wishes to express to the author, Mary Allen Wilkes (Clark), and the Computer Research Laboratory of Washington University, St. Louis, Missouri, its appreciation for the development set forth in LAP6 as well as its thanks for permission to use parts of the LAP6 program.

¹M. A. Wilkes, LAP6 Handbook, Computer Research Laboratory Tech. Rep. No. 2, Washington University, St. Louis, May 1, 1967.

²PAL-D Assembler Programmer's Reference Manual DEC-D8-ASAA-D.

1.0 PROGRAM OVERVIEW

File Commands, referred to as FILECOMS, is comprised of a series of file handling routines, along with Save Program, Add Binary, and Save Binary in its entirety. FILECOMS is called by the Editor whenever a Print Source, Add Binary, Save Program, or Save Binary is requested, or whenever an assembly is requested of a source file. For any of the above cases, FILECOMS uses the MC parameter table and the index to set up the required arguments. When handling files and setting up arguments, FILECOMS checks to see if the request is possible and/or legitimate. If a command cannot or will not be executed, FILECOMS displays a "NO". Once FILECOMS sets up arguments, control is turned over to the appropriate program.

File Commands is on TBLKs 350, 351, 352, and 353 and in MBLKs 0, 1, 2, and 3 of segment 2 when in core. The Editor communicates with FILECOMS relative to the tag FCSA (FILECOMS Starting Address). To execute a Save Program, for example, the Editor reads in the first block of FILECOMS and executes a JMP FCSA+3. The contents of FCSA+3 are JMP SAVPRG. The FCSA table, which must not be moved without the Editor's knowledge, has the following form:

(4020)	FCSA, JMP SAVBIN	/SAVE BINARY
	JMP ADDBIN	
	JMP .	
	JMP SAVPRG	/SAVE PROGRAM
	JMP LIORAS	/ASSEMBLER
	JMP PRNPRG	/PRINT SOURCE

At location EPLSTP is the "REPLACE?" table which contains the DIAL codes for the phrase "REPLACE?". The "REPLACE?" table is used in conjunction with the Editor's grid pattern display table (A6) to display the phrase "REPLACE?". Because FILECOMS uses the file area extensively, an explanation of index structure is crucial to understanding the secrets of FILECOMS.

The index is located at tape blocks 346 and 347; the LAP6-DIAL system expects it to be two blocks long. The index must be comprised of contiguous tape blocks and the first tape block must be of the form XX6. An index tape block of 352, for example, would not be compatible with a number of LINCTape group instructions that are scattered throughout the system. The left half of the first word on tape block 346 must be the DIAL code for a slash (57), in order to identify the block as an index. The first 10 words of tape block 346 contain 5757 (//) and are never used for file name storage by the DIAL system; the remaining words on both tape blocks are used to store index information. Each file in the index takes up 10 words, so that a DIAL tape can hold a maximum of 77 octal file names. Unfilled portions of the index contain 5757 in all 10 words. For any given 10 word portion, the first four words are reserved for the file name. The DIAL code for the first character of the name is stored in the

left half of the first word; the second character of the name goes in the right half of the first word, etc. The maximum number of characters in a name is eight. Unused half-words in a name = 77 (?). The name FILENAME would be stored in the following manner:

0611	FI
1405	LE
1601	NA
1505	ME

The name DIAL would be saved as:

0411	DI
0114	AL
7777	(Not
7777	Used)

Question marks within a name are treated as ordinary characters. Leading question marks are illegal and may cause unpredictable results. Trailing question marks are ignored.

If a file name refers to a source, the fifth word of the ten word sector holds the first TBLK of the source and the sixth word holds the length. If there is no source, then the fifth and sixth words are both equal to 5757. If a name refers to a binary file, then the seventh word contains the first TBLK of the binary and the eighth word contains its length. If there is no binary, then the seventh and eighth words are both equal to 5757. In words 5, 6, 7, and 10, the low order 9 bits are reserved for tape block and length; bit zero is used to indicate whether source or binary information is present; bits 1 and 2 are not used.

2.0 SAVE PROGRAM AND FILECOMS

When a user requests a Save Program, the Editor exits, reads four tape blocks of FILECOMS (350) into locations 4000-5777, loads the last tape block used in the Working Area into the AC, and jumps to FCSA+3. Save Program first stores away the complement of the source length at location F1, then checks for line number arguments, displaying "NO" if they have been requested. Next, Save Program reads the index (TBLKS 346, 347) into locations 3000-3777. At this point, Save Program checks the left half of the first word in the index for a 57; if there is no 57, Save Program fills locations 3000-3777 with 5757. This move is called making an index. Save Program then calls a subroutine called LOOKUP to check for a name match between the requested name - to be found in the MC parameter table at E6 + 2 to E6 + 5 (2372-2376) - and the index. If there is no name match, Save Program searches the index for a blank entry (a 57 in the left half of the first word). If no blank entry is found, the index is full and Save

INDEX STRUCTURE

TBLK 346

57	57
57	57
57	57
57	57
57	57
57	57
57	57
57	57

INDEX IDENTIFIER (10 WDS)

X ₁	X ₂
X ₃	X ₄
X ₅	X ₆
X ₇	X ₈

NAME (4 WDS)

Y ₁	Y ₂
Y ₃	Y ₄

SOURCE TBLK (1 WD)

SOURCE LENGTH (1 WD)

Z ₁	Z ₂
Z ₃	Z ₄

BINARY TBLK (1 WD)

BINARY LENGTH (1 WD)

TBLK 347

X ₁	X ₂
X ₃	X ₄
X ₅	X ₆
X ₇	X ₈

UNUSED HALF WORDS IN A NAME = 77 (?)

Y ₁	Y ₂
Y ₃	Y ₄

FOR NO SOURCE Y₁=Y₂=Y₃=Y₄=57 (/)

Z ₁	Z ₂
Z ₃	Z ₄

FOR NO BINARY Z₁=Z₂=Z₃=Z₄=57 (/)

UNFILLED PORTION OF THE INDEX
5757 IN ALL WORDS

Program displays "NO". If a blank entry is found, locations E6 + 2 to E6 + 5 are placed into its first four words and the gap search commences. If a name match is found, Save Program must determine whether or not the matched entry has an accompanying source (the match could be binary only). To determine source or binary match, Save Program checks bit 0 of the fifth word in the matched entry. If the match is binary only, the gap search commences. If a requested name matches a source, Save Program displays the phrase "REPLACE?" until a key is struck on the Teletype. If the key struck on the Teletype is not an "R", Save Program returns to the Editor; if an R is typed, 5757 is put into the fifth and sixth words of the matched entry to erase the starting TBLK and length information of the old source. Save Program then calls the gap search subroutines.

Once the index has been properly tended to, Save Program must find room in the file area to store the source. There are 600 TBLKs on a DIAL tape that are available for storage. These 600 TBLKs are divided into two sections:

1. The upper file area (TBLKs 470-777)
2. The lower file area (TBLKs 0-267)

Sources are saved in the file area as close to the index as possible.

The gap search subroutines first scan all upper file entries in the index to determine where the gaps are in the upper file. A gap is any unused portion of the file area. If a file named ONE runs from TBLKs 470-517 and another file named TWO runs from TBLKs 550-600, then a 30 TBLK gap exists from TBLKs 520-547. If there are no files stored in the upper file area, then there exists a 310 TBLK gap from TBLKs 470-777. The gap search subroutines check words 5-10 of all index entries to compute the file gaps. The best TBLK in the upper file area is 470 because it is closest to the index. The gap search routines initially set the best TBLK at 470; then they start searching for gaps. If a gap is found that is smaller than the length of the requested entry, the best TBLK is reset to one greater than the last TBLK of the current entry. For example, if the best TBLK is set to 470 and the length of the requested entry is 20 TBLKs and the gap search routines find a source entry (or binary, for that matter) starting in TBLK 500 (word 5) that is fifty TBLKs long (word 6); i.e., the source in question resides in TBLKs 500-547 inclusive, then the gap is from TBLKs 470-477 (10 TBLKs). This is too small to accommodate an entry of 20 TBLKs, so the best TBLK is reset to 550, which is one more than the last TBLK (547) of the current entry. The gap search subroutines scan the index, resetting the best TBLK every time a gap that is too small is encountered.

Once the upper file has been searched, the entire sequence is repeated for the lower file. When searching the lower file, the best TBLK is set to one less than the start of the current entry when an insufficiently long gap is

encountered. Searching the upper and lower files in the manner described above really finds all those TBLKs that will not work. For example, suppose that the best TBLK in the lower file turned out to be 40; this would mean that there are no gaps between TBLKs 41-267 that are large enough to file the requested source.

Once the best TBLKs for upper and lower file have been established by the gap search routines, Save Program must determine whether or not the requested source will fit in the file area. If the length of the requested source is greater than the difference between the lower bound of the lower file (TBLK 0) and the best TBLK of the lower file, and if it is also greater than the difference between the upper bound of the upper file (TBLK 777) and the best TBLK of the upper file, then the requested source is bigger than all gaps in the file area and it will not fit. For this case, Save Program displays "NO". If the requested source will fit in either the upper or lower file area, Save Program determines which of the two best TBLKs is closest to the index before initiating the actual save.

After file area has been allocated by Save Program, the starting tape block of the requested source is stored in the fifth word of the entry and the source length is stored in the sixth word. Save Program then checks the keyboard. If a key has been struck, the request is inhibited and control is turned over to the Editor; if no key has been struck, the Save Program part of Save Program commences.

Save Program transfers 16 (octal) TBLKs at a time from the working area to the file area until the entire source has been saved. The last transfer may be from 1 to 16 TBLKs, depending on the length of the source being saved.

3.0 PRINT SOURCE AND FILECOMS

Whenever a Print Source is requested, FILECOMS must be called ahead of Print Source to set up arguments for Print Source. The Editor executes this call by reading TBLKs 350-353 of FILECOMS into locations 4000-5777 and doing a JMP FCSA+5. FILECOMS first checks the left half of E6 + 2 (2373) in the MC parameter table for a 77 (?). A 77 indicates that the Working Area is to be printed; for this case FILECOMS reads the TBLKs containing Print Source (363) and TTY (364) into locations 5000-5777, moves the source working area unit to E6 + 6, and executes a JMP 1000. If the Print Source is by file name, FILECOMS checks to see if the requested unit has an index, displaying "NO" if it does not. If the requested unit has an index, FILECOMS calls the routine LOOKUP to check for a name match. If there is no name match, FILECOMS displays "NO". If there is a name match, FILECOMS further checks to see if the name refers to binary only, and if so displays a "NO". If the name match is legal, FILECOMS reads Print Source and TTY into locations 5000-5777, retrieves the first TBLK of the source

from the index and puts it into AC. FILECOMS then calls Print Source by executing a JMP 10000.

4.0 THE ASSEMBLER AND FILECOMS

Whenever an assembly is requested of a source in the file area, FILECOMS must be called to set up arguments for the Assembler. The Editor executes this call by reading TBLKs 350-353 into locations 40000-5777 and doing a JMP FCSA + 4. The index routines are next called by FILECOMS and executed in exactly the same manner as if a Print Source had been requested. FILECOMS places the unit number at location UNITNO (4777), places the QL or LI word at location UNITNO-3 (4774), and puts the starting TBLK of the requested source in the AC. FILECOMS then recalls the Editor to set up further arguments for the Assembler, read it in and start it up.

5.0 SAVE BINARY AND FILECOMS

When a Save Binary is requested, the Editor exits, loads four blocks of FILECOMS into 40000-5777, and jumps to FCSA, File Commands starting address. Save Binary reads the binary header block (57 in the binary working area) into 54000-5777.

E6, the MC parameter table, is examined to determine the starting mode and address. If none were specified (+SB FILENAME,U), E6 contains zero. A LINC-mode halt is stored in the header, and Save Binary jumps to SB020 to find space for the file. If 8-mode was specified, E6 contains 44000 plus the PDP-8 field in which the program is to start. E6+1 contains the 12-bit address within that field. The following instruction sequence is stored in the header:

PDP		
CIF	X	/WHERE X IS 0 OR 10, DEPENDING ON
		/THE FIELD SPECIFIED
JMP	1 377	/START THE PROGRAM
START		/STARTING ADDRESS FROM E6+1

If LINCmode was specified, E6 contains 04000 plus the starting PDP-8 field, and E6+1 contains the 12-bit address within the field. The following instruction sequence is built in the header:

LIF	X	/X IS 0-7, DEPENDING ON THE SEGMENT
		/SPECIFIED.
JMP	START	/START IS THE 10 LOW-ORDER BITS
		/FROM E6+1.

The default starting addresses are field 0, location 2000 for PDP-8 mode and segment 2, location 20 for LINCmode. These are set up by the Editor before

calling FILECOMS, if a starting address was not explicitly requested.

After building these instructions in the header, the starting address is checked for validity. It must be less than 200000 (8K), and in a block which contains assembled code.

Save Binary next obtains the number of binary blocks (location 337 within the header) from the header. One is added to this number, to allow for the header, and the result complemented and stored at Fl. This is used as the length of the desired file during the search for file space. The two's complement of the number of binary blocks is calculated and stored at MBCNT, as a loop control during copying. Save Binary then follows the same sequence as Save Program to read the index, find space for the file, and write the updated index. Next, the I/O control blocks are initialized, an auto-index register is set up for the block map in the header, and copying begins.

Each word of the block map is tested for zero. If it is zero, the corresponding block in the working area is not used, so the input block number is incremented and the loop restarted. If the word is not zero, the corresponding block is read, and MBCNT is tested to see whether it is the last binary block. If it is the last, Save Binary computes the number of blocks currently in the buffer, and goes to SB130 to write them. If not the last block, Save Binary increments the input buffer address, and checks to see whether the buffers are full. If not, the input block number is incremented and the loop re-entered. If the buffers are full, or the last block has been read, HDRSW is tested to determine whether the header block has been written. If not, it is written at this time, and the switch is set. Then the buffers are written, and MBCNT is tested to check for completion. If there is more to go, the I/O control blocks are reinitialized and the loop is re-entered. Otherwise, the Editor is called by jumping to 7777 in field 1.

6.0 ADD BINARY AND FILECOMS

When Add Binary is requested, the Editor exits and loads blocks 350-353 into locations 4000-5777. It then jumps to FCSA+1. Add Binary reads the index of the requested file, and finds the binary requested, displaying "NO" if unsuccessful.

The header and up to 15 blocks of the binary file are read, as well as the header of the binary working area.

The header from the file is scanned by subroutine SCANHD to determine the memory address for which the first block was assembled. If a relocation (other than zero) was specified, it is stored as the new memory address for the program.

The binary data from the file is scanned, and non-zero words moved to the buffer for the binary working area.

When the end of an input block is reached, a running block count (MBLKS) is incremented and checked for end of file. If the end has been reached, the final binary working area block is written, the binary header is written, and control returns to the Editor.

If there remains more binary to add, the header is scanned so that the relocation can be adjusted for skipped blocks and, if the end of the buffer has been reached, the buffer is refilled with up to 15 blocks from the file.

When a non-zero word is found which goes into another block of the binary working area (other than the block currently in core), the old block is written, the header updates, and, if the header indicates that the new block is used, the new block is read in. If there is nothing in the new block, the buffer is cleared.

7.0 FLOW DIAGRAMS (Attached)

8.0 PROGRAM LISTING (Attached)

9.0 MEMORY MAP

Used for LINC tape trans- fers	NOT USED	7777
	Filecoms	60000
	Not Used	54000
	Filecoms	50000
	(LINC Tape S.R.)	44000
		40000
Used for LINC tape trans- fers	NOT USED	
	MC Parameter Table DSC Grid Table	24000
	NOT USED	20000
		0

FILECOMS SYMBOL TABLE

ADVIND	-	The first core location of the subroutine that advances the index pointer (BETA 2) by 10 words at a time and checks for end of index.
AUTO	-	8-mode auto-index register (10) used by SB.
A6	-	The first core location of the Editor's DSC grid table.
BFAD	-	The location containing buffer address, shifted right 8.
BFLN	-	The location containing the length of buffer, in blocks.
BUFFER	-	The buffer address, shifted right 8.
BUFLN	-	The length of the buffer, in blocks.
BUMP	-	The location containing the increment for index pointer to address desired (source or binary) pointers.
BWA	-	The pseudo-unit for binary working area.
CHKIND	-	The first core location of the subroutine that reads in and checks for the validity of an index.
COMPBN	-	The first core location of the subroutine that compares TBLKs in the index with the prospective best TBLK and sets the best TBLK during the gap search for Save Program.
CR1	-	The symbolic code for carriage return; it is equal to 4300 and is used in conjunction with the SHD I instruction.
DATSEG	-	The data segment used throughout FILECOMS. It contains A6, E6, and Index.
DIALU	-	The pseudo-unit that contains the resident DIAL system (100).
DRSTRT	-	The address of DIAL bootstrap routine.
EDRTN	-	The start of routine to call DIAL bootstrap.
ENTER	-	The first core location of the subroutine that searches for a blank slot in the index during a Save Program.
E6	-	The first core location of the MC parameter table.
FCSA	-	The first core location of the FILECOMS starting address table.

FILE - The first TBLK in the upper file area above the index that is available for file storage.

FINDSP - The first core location of the subroutine that sets up arguments for upper and lower file during a gap search.

FREE - The first TBLK+1 in the lower file area below the index that is available for file storage (also the first TBLK of the free area on the DIAL tape).

F1 - The core location that holds the one's complement of the length of a source during a Save Program, or one's complement of the length of the binary (including header) during a Save Binary.

GAPSR - The first core location of the main gap search subroutine which is called during a Save Program and Save Binary.

GETBN - The first core location in the subroutine that sets up the starting TBLK and source length during a gap search.

GETPS - The starting point of code which reads in and calls Print Source.

G1 - The first core location of the subroutine that determines the nearest of the two best TBLKs in the upper and lower file area; it is called during a Save Program.

HDRBLK - The TBLK of binary header, relative to binary working area.

HDRIO - The control block for reading and writing header.

HDRSW - The switch is 7777 if the header has not been written; 0 after writing.

HEADER - The core location in the current segment which contains the header.

HI8FLD - The field containing I/O routines and bootstrap.

H1 - The first core location of the subroutine that is called during a Save Program to determine if the best TBLKs will accommodate the length of the requested source.

INDEX - The core location of the index (3000).

INFIL - The control block used in reading files to be copied.

KBDOPR - The core location in the Editor relative to which FILECOMS effects its return to the Editor after an assembly by file name has been requested.

KBDSEG - The segment containing KBDOPR.

K2 - A tag in the subroutine REPLAC; tests for response from user.

LIORAS - The first core location of the main subroutine that is called by FILECOMS during an assembly by file name.

LOOKUP - The first core location of the subroutine that searches the index for a name match.

LO8FLD - The field in which FILECOMS resides.

MAKIND - The first core location of the subroutine that makes an index.

MAP - The core location in the current segment containing the block map from header.

MBCNT - The location containing minus block count: two's complement of number of blocks to read from working area.

MBFAD - The location containing the minus buffer address, shifted right 8.

MBFEND - The location containing minus (buffer end + 1), shifted right 8.

OUTFIL - The control block for output file during copies.

PDPMOD - The location of the routine for setting up PDP-8 mode start in Save Binary.

PRNPRG - The first core location of the main subroutine that is called by FILECOMS during a Print Source.

PSBLK - The first TBLK of Print Source (relative to DIAL).

PSENT - The entry point of Print Source main code.

PSIN - The control block for reading Print Source.

PSWA - The routine which sets up for printing the source working area.

P1 - Constant 0001.

P2 - Constant 0002.

READ - The page 0 location pointing to Read routine.

REPLAC - The first core location of the subroutine that displays the phrase "REPLACE?" on the scope.

RPLSTR - The first core location of the replace table.

RSTRX - The first core location of the subroutine that writes out the index.

SAVBIN - The first location of Save Binary routine.

SAVPRG - The first location of Save Program routine.

SAYNO - The first core location of the subroutine that displays the word "NO" on the scope.

SB010	-	
SB020	-	
SB030	-	
SB040	-	
SB100	-	
SB110	-	Tags in Save Binary
SB120	-	
SB130	-	
SB140	-	
SB160	-	

SETFLD - The location of LIF instruction inserted into the header.

SP020	-	
SP030	-	
SP100	-	Tags in Save Program
SP110	-	
SP1 0	-	

SRORBN - The first core location of the subroutine that determines whether a name match is source or binary.

SWA - The pseudo-unit that holds the source working area.

SWITCH - A core location that is used by the gap search subroutines in conjunction with the upper and lower file area.

UNITNO - The core location that holds the unit number for assembly by name.

USENO - The core location containing the count of binary blocks used.

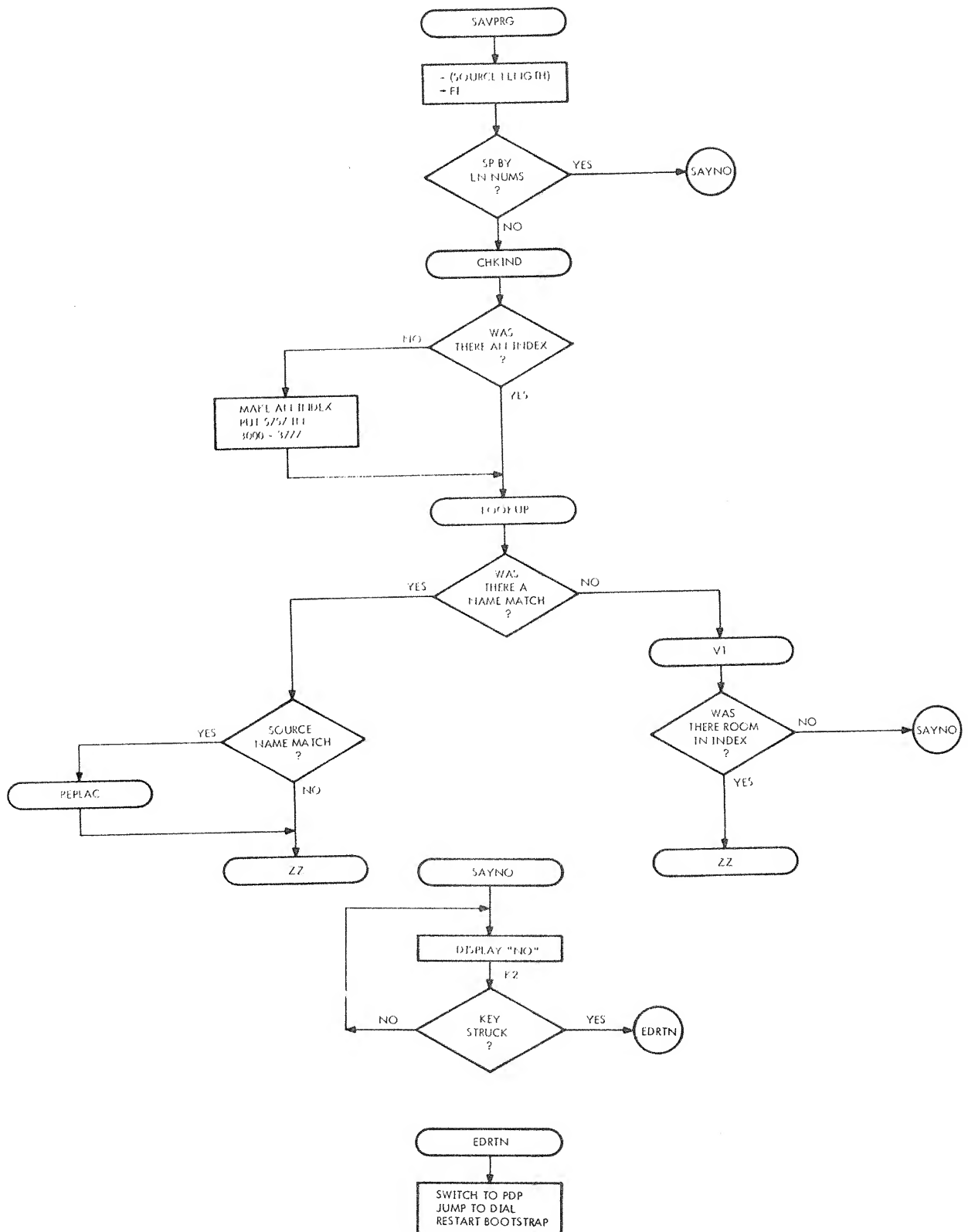
WA - The first TBLK of the Working Area.

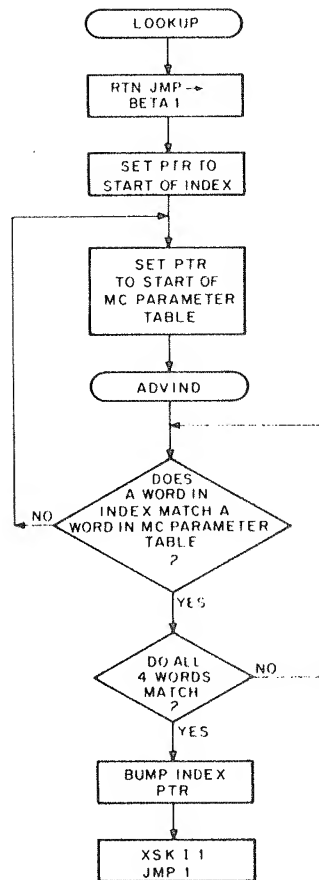
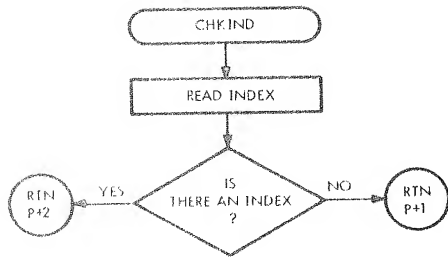
WRITE - The page 0 location pointing to Write routine.

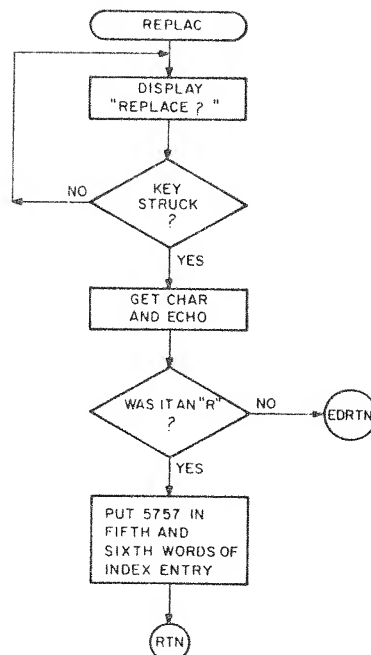
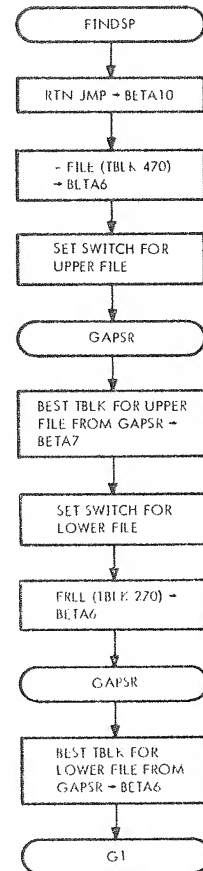
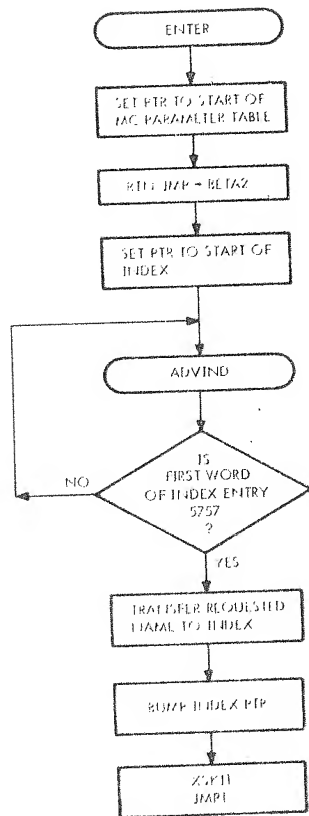
XBLK - The TBLK containing the index.

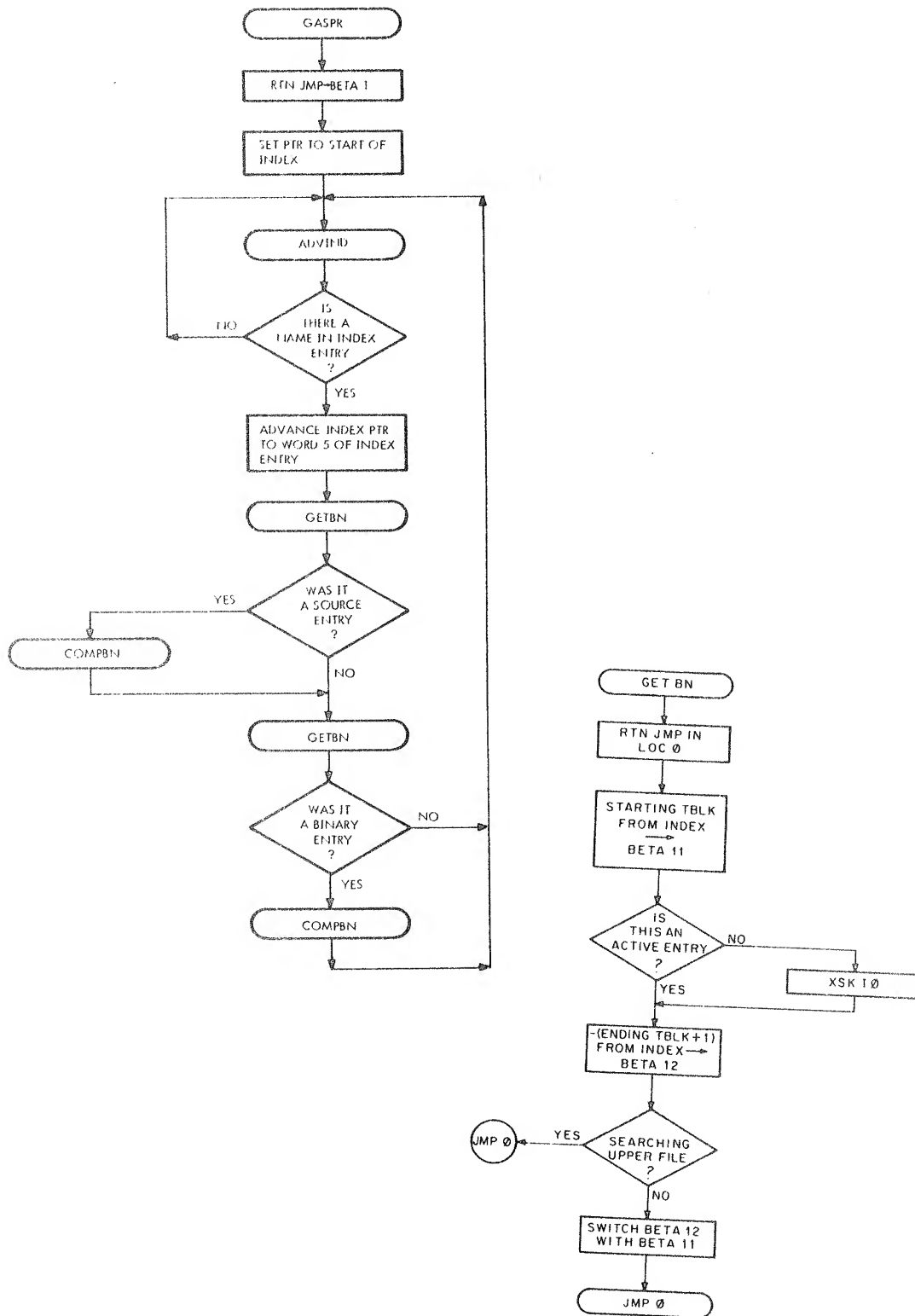
XIO - The control block for reading and writing index.

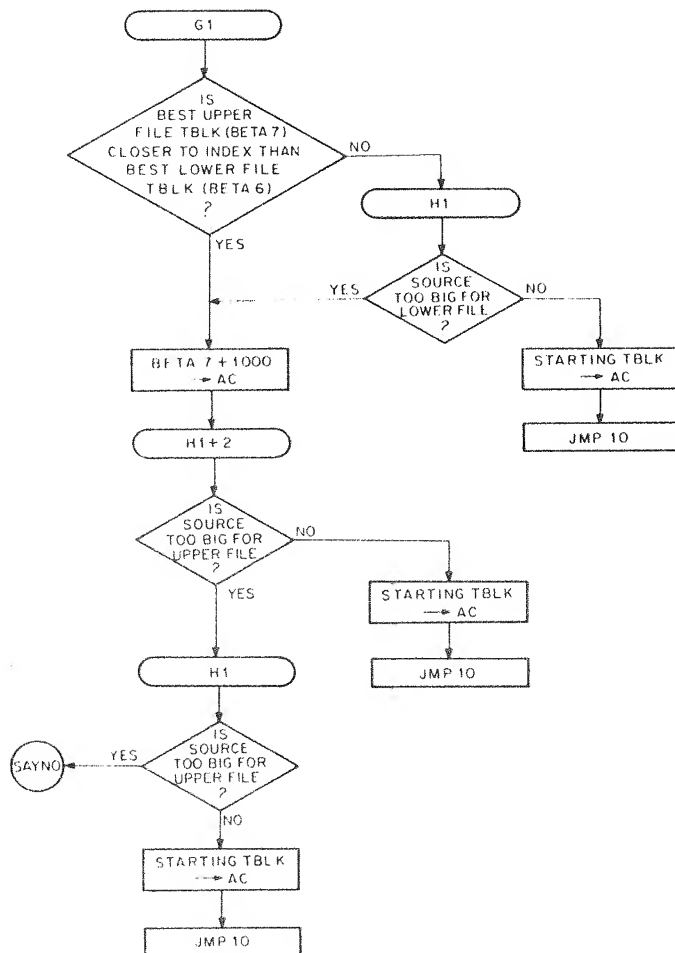
ZERO - Constant zero.

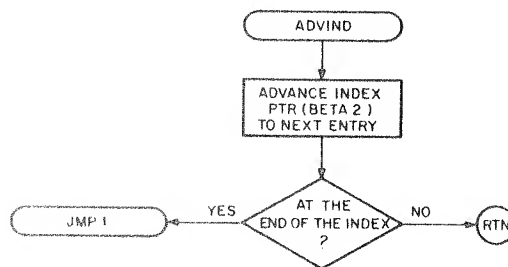
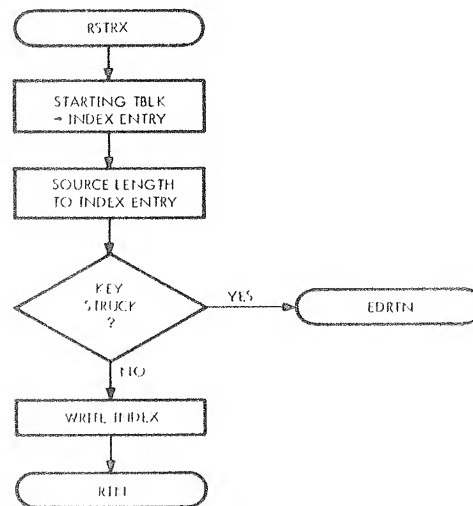
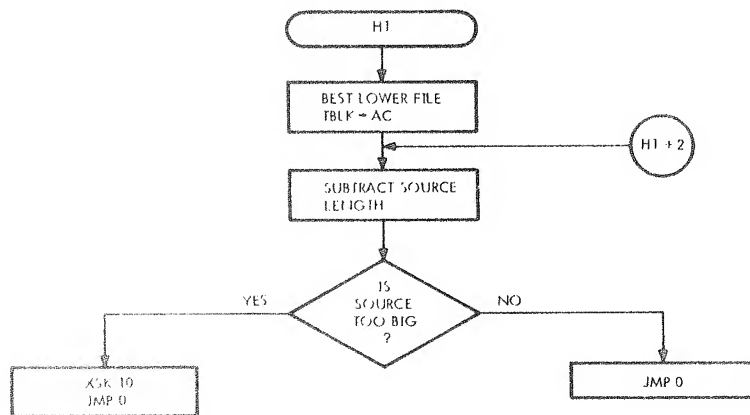


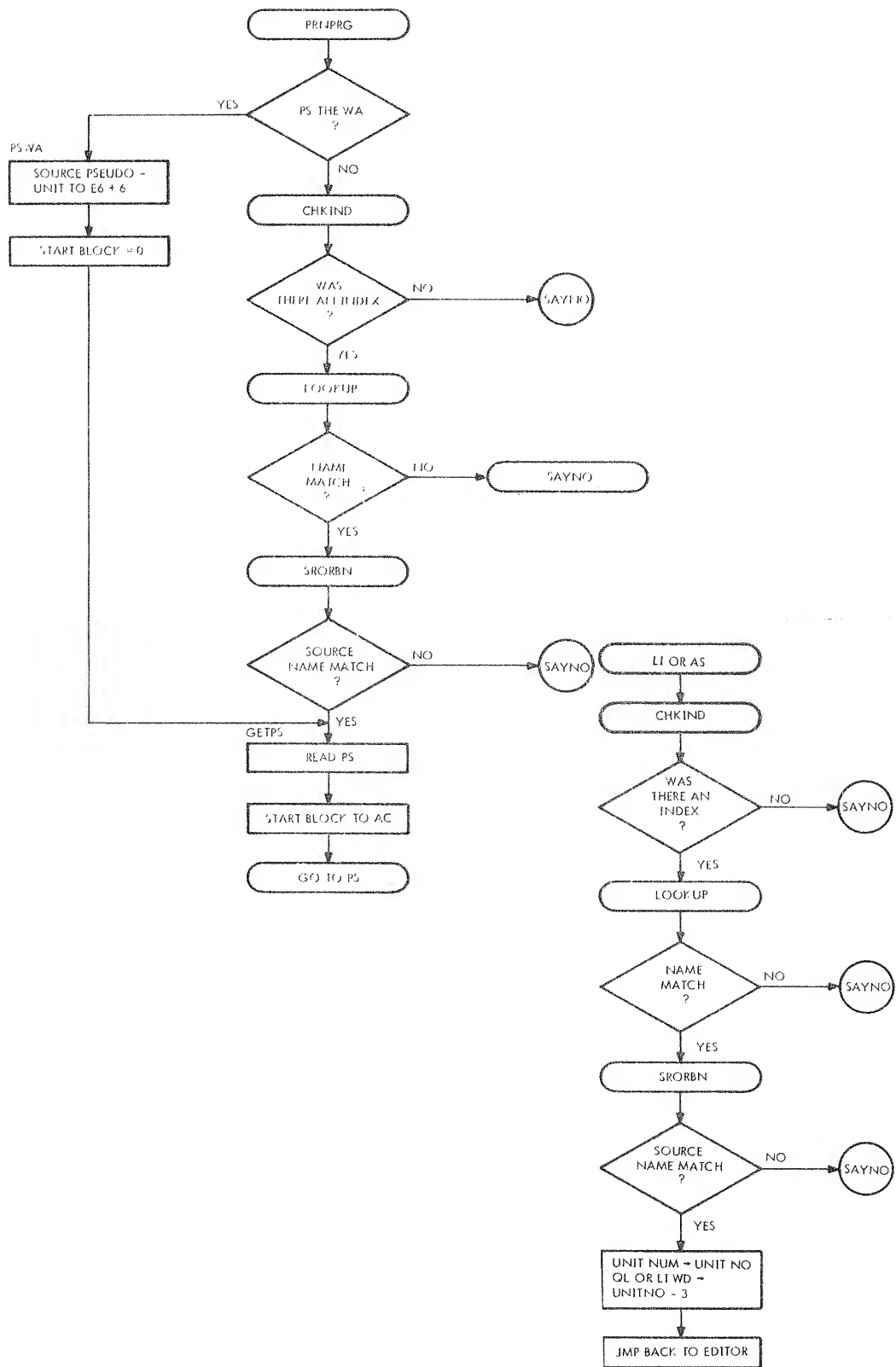


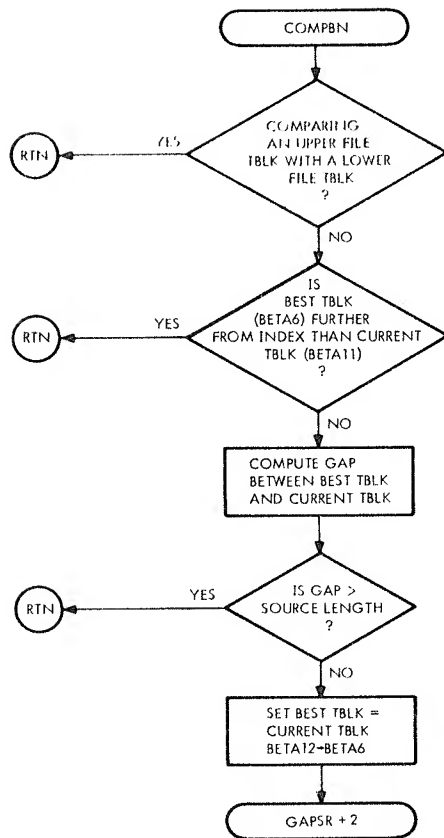


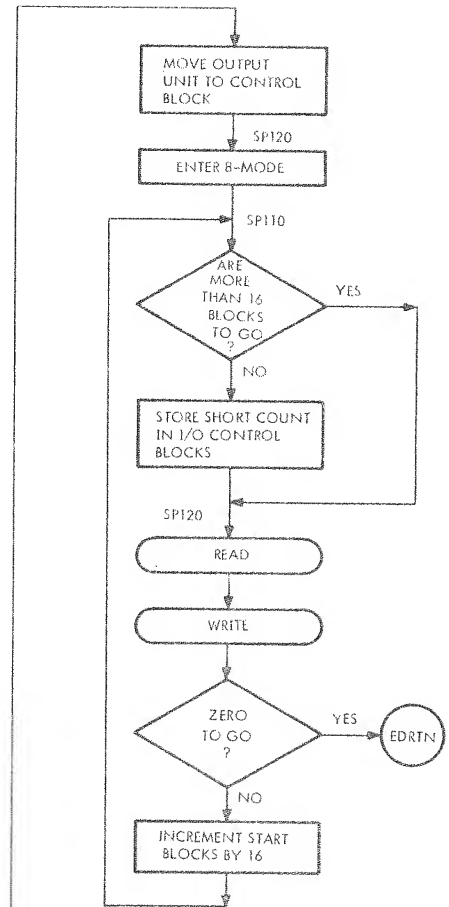
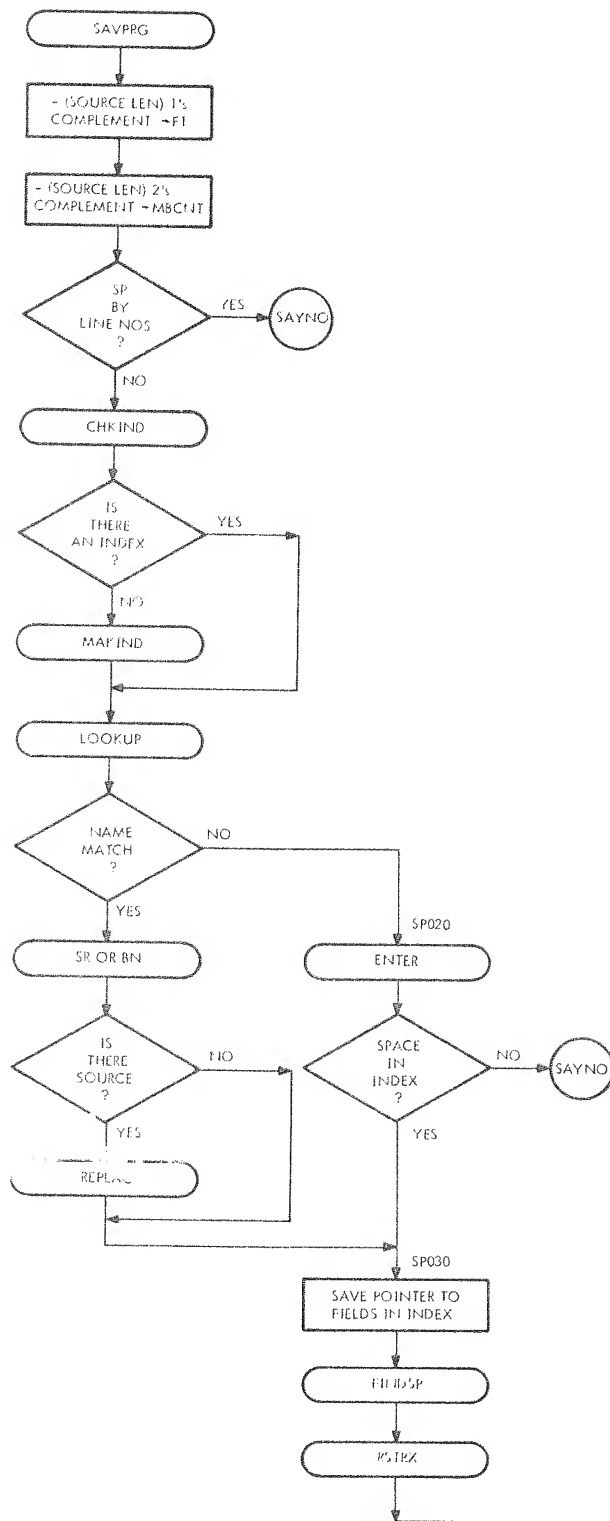


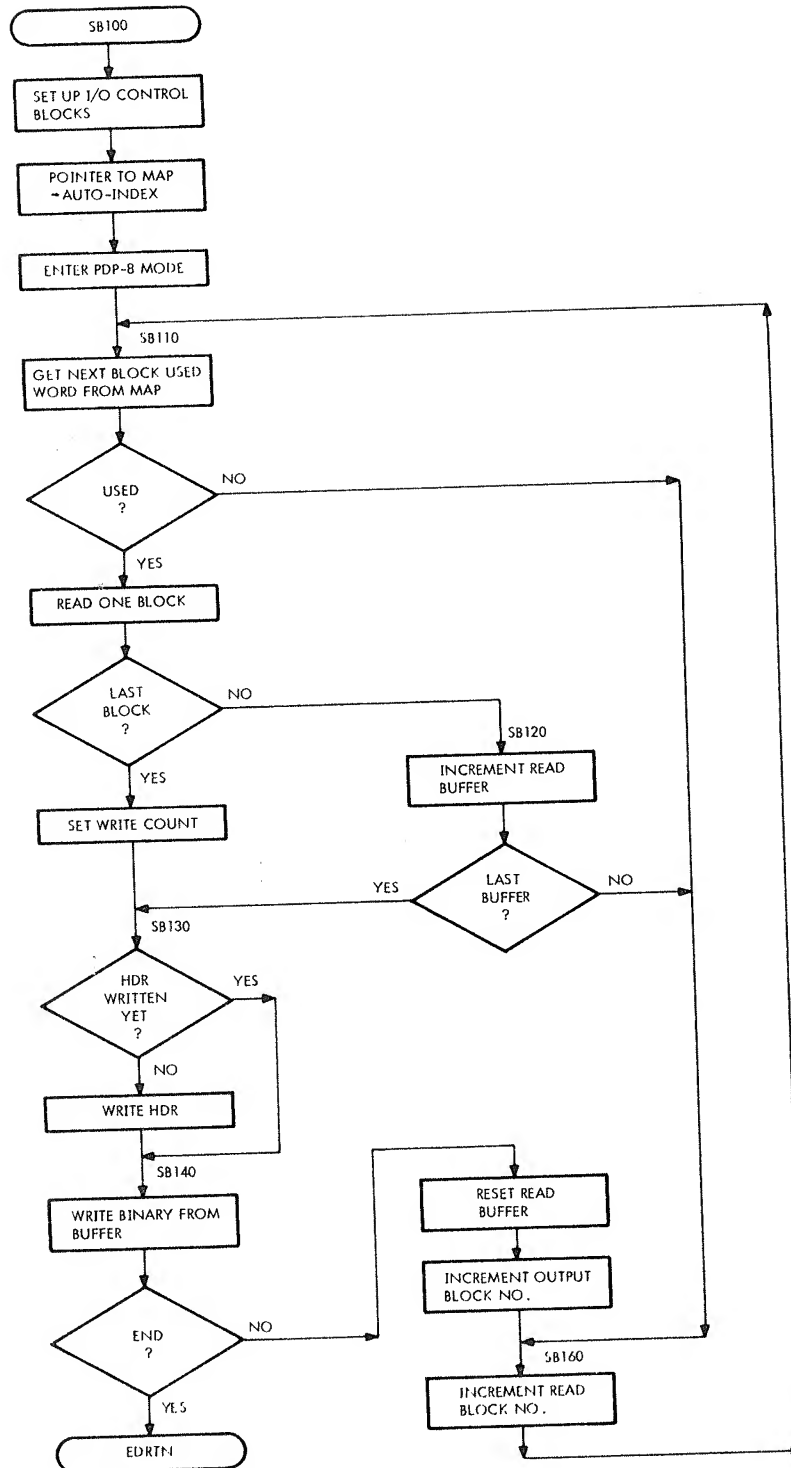


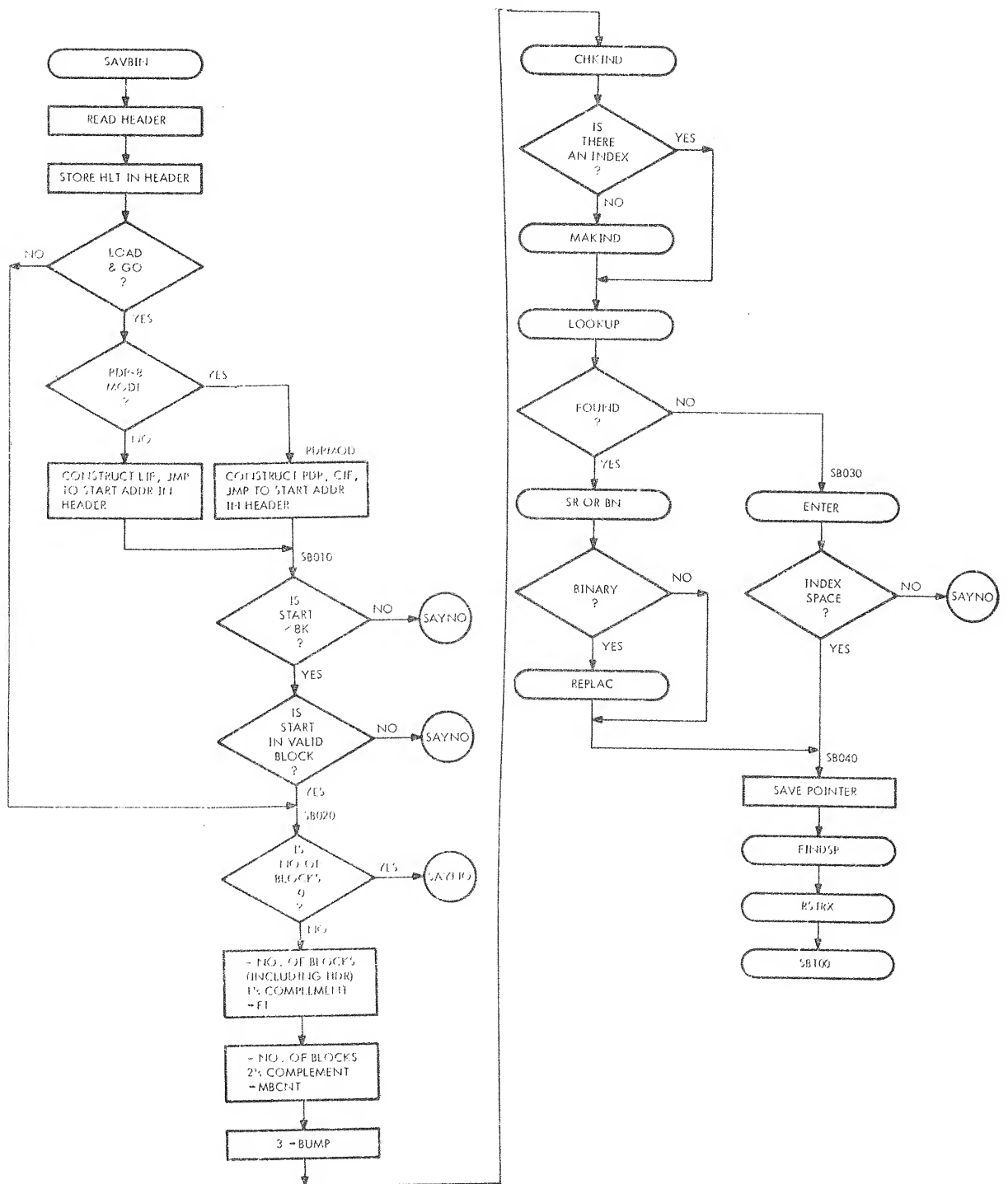













```

0000      *20
0001      / FCMSV4,5      FILE COMMANDS, V4,5
0002      /
0003      /
0004      / LAP6-DIAL FILE COMMANDS ..., SB,SP,AB,LI,QL,AS,PS
0005      /
0006      / MAR 31, 1978
0007      /
0008      / THIS ROUTINE IS A MODIFICATION OF FUNCTIONALLY EQUIVALENT
0009      / CODE FROM LAP6. COMMENTS HAVE BEEN ADDED, TAGS CHANGED FROM
0010      / LETTER-DIGIT TO MNEMONIC NAMES, ABSOLUTE MEMORY ADDRESS REFERENCES
0011      / REMOVED WHERE POSSIBLE, I/O DISCIPLINE AND BUFFERING IMPOSED,
0012      / AND THE WHOLE CONDENSED FROM 7 BLOCKS TO 4, WE ARE DEEPLY INDEBTED,
0013      /
0014      /
0015      / RESIDES IN DIAL BLOCKS 50-53
0016      / OCCUPIES MEMORY LOCATIONS 4000-5777
0017      /
0018      / 16 MBKLS (OCTAL) IN UPPER FIELD ARE USED FOR BUFFERS: 1 2000 - 1 6777
0019      / LOCATIONS 1 7000 TO 1 7777 MUST REMAIN INTACT.
0020      /
0021      / THESE LOCATIONS CONTAIN DIAL I/O ROUTINES AND BOOTSTRAP,
0022      /
0023      /
0024      / THE FOLLOWING AREAS IN THE LOWER PDP-8 FIELD ARE USED:
0025      /
0026      / AUTO-INDEX REGISTER 10 IS USED BY SB AND AB,
0027      / PAGE ZERO LOCATIONS 21 AND 22 ARE ASSUMED TO CONTAIN POINTERS
0028      / TO THE DIAL I/O ROUTINES IN THE UPPER FIELD,
0029      / 2000 - 2200 ARE ASSUMED TO CONTAIN THE DIAL CHARACTER DISPLAY TABLE,
0030      / 2371 - 2377 CONTAIN THE MC PARAMETER TABLE, E6,
0031      / 3000 - 3777 ARE USED TO READ AND WRITE THE DIAL INDEX,
0032      / AND FOR HEADERS DURING AB.
0033      /
0034      / 5400 - 5777 ARE USED FOR THE BINARY HEADER BLOCK DURING SB OPERATIONS.
0035      / 5000 - 5777 ARE USED BY THE MAIN CODE OF PS (PRINT SOURCE).
0036      /
0037      / CODE IS NOT REUSABLE
0038      /
0039      /
0040      /
0041      /
0042      /
0043      /
0044      /
0045      /
0046      /
0047      /
0048      /
0049      /
0050      /
0051      /

```

```

0020      6026      JMP SAVBIN      /EDITOR
0021      7120      JMP AADBIN      /CALLS
0022      6022      JMP          /ALL
0023      6341      JMP SAVPRG      /FCOMS
0024      7023      JMP LIORAS      /HERE
0025      6376      JMP PRNPRG
0026      EJECT

```

SAVE BINARY ENTERS HERE

[illegible]

0205	/	HEADER IS NOW PROPERLY SET UP	
0206	/	LOOK FOR SPACE FOR THE FILE	
0207	/		
0210	/		
0211	SB020,	ADD USENO	/NO OF BLKS USED (FROM HEADER)
0212		AZE I	/BOMB IF ZERO
0213		JMP SAYNO	
0214		ADD P1	/PLUS ONE FOR HEADER
0215		COM	/FINDSP WANTS IT NEGATIVE
0216		STA	
0217		F1	/HOLD FOR SPACE SCAN
0220		ADD P2	/TWO'S COMP OF NO OF BLOCKS (WITHOUT HEADER)
0221		STC MBCNT	/HOLD FOR COPY
0222		ADD P3	
0223		STC BUMP	/REPLACE BUMP CONSTANT IN LOOKUP
0224		JMP CHKIND	/READ IN THE INDEX BLK
0225		JMP MAKIND	/RTN FOR NO INDEX-MAKE ONE
0226		JMP LOOKUP	/SEARCH INDEX FOR NAME IN E6+2
0227		JMP SB030	/NOT FOUND
0230		JMP SRORBN	/CHK IF SYMBOLIC ONLY
0231		JMP REPLAC	/NO-DISPLAY REPLACE FOR VERIFY
0232		JMP SB040	/BIN ENTRY EMPTY, OR REPLACE REQUESTED
0233		JMP ENTER	/PUT NAME IN INDEX
0234	SB030,	JMP SAYNO	/NO SPACE IN THE INDEX
0235	SB040,	SET 4	/SAVE XR PTR
0236	P2,	2	
0237		JMP FINDSP	/FIND BEST SPACE, RETURN TBLK IN AC
0240		JMP RSTRX	/UPDATE & WRT XR
0241	SB100,	LDA I	/BINARY WORK UNIT,..
0242		BWA	
0243		STC INFIL	/...TO CONTROL INPUT
0244		ADD OUTFIL+2&1777	/STARTING BLOCK
0245		STA	/...IS HEADER TBLK...
0246		HDRI0+2	
0247		ADD P1	/...PLUS 1
0250		STC OUTFIL+2	/...IS START OF ACTUAL BINARY
0251		ADD P1	/CONSTANT 1...
0252		STC INFIL+3	/FOR SINGLE-BLOCK INPUT
0253		ADD X10	/OUTPUT UNIT
0254		STA	/...FOR WRITING HEADER...
0255		HDRI0	
0256		STC OUTFIL	/...AND BINARY
0257		LDA I	/GET ADDRESS OF BLOCK MAP
0260		MAP=1:4000	/MAP IN LINC FIELD 2
0261		PDP	/ENTER PMODE FOR COPY
0262		PMODE	
0263		DCA AUTO	/SETUP AUTO-INDEX REG
0264		EJECT	

```

0265 / THIS IS THE COPY LOOP FOR SA...BINARY
0266 / N.B: CODE BETWEEN THIS POINT AND EDRTN MUST BE IN ONE PDF-8 PAGE,
0267 /
0270 /
0271 / SB110,
0272 /
0273 /
0274 /
0275 /
0276 /
0277 /
0300 /
0301 /
0302 /
0303 /
0304 /
0305 /
0306 /
0307 /
0310 /
0311 /
0312 /
0313 /
0314 /
0315 /
0316 /
0317 /
0320 /
0321 /
0322 /
0323 /
0324 /
0325 /
0326 /
0327 /
0330 /
0331 /
0332 /
0333 /
0334 /
0335 /
0336 /
0337 /
0340 /
0341 /
0342 /
0343 /
0344 /
0345 /
0346 /
0347 /
0350 /
0351 /
0352 /
0353 /
0354 /
0355 /
0356 /
0357 /

4206 /GET A WORD FROM MAP
4207 /SKIP IF BLOCK USED
4210 /NOT USED -- BUMP BLOCK NO AND LOOP
4211 /SET INST FIELD FOR READ ROUTINE
4212 /READ ONE BINARY BLOCK
4213 /...FROM BWA
4214 /CONSTANT ONE
4215 /TEST FOR LAST BINARY BLOCK
4216 /DONT STOP YET

/
/
/ WE HAVE JUST READ THE LAST BINARY BLOCK
/
/
/ TAD MBFAD /SUBTRACT (FIRST BUFF - 1)...
4217 TAD INFIL+1 /...FROM CURRENT BUFFER...
4220 DCA OUTFIL+3 /...FOR OUTPUT BLOCK COUNT
4221 JMP SB130
4222

/
/
/ COME HERE AFTER READING ANY BINARY BLOCK EXCEPT THE LAST
/
/
/ SB120,
/
/
/
4223 TAD INFIL+1 /INCREMENT BUFFER ADDRESS
4224 DCA INFIL+1
4225 TAD MBFEND /BUFFER LIMIT...
4226 TAD INFIL+1 /...COMPARED TO BUFFER ADDRESS
4227 SPA CLA /DUMP BUFFERS NOW IF FULL
4230 JMP SB160 /OTHERWISE BUMP INPUT BLOCK

/
/
/ COME HERE TO WRITE THE BUFFERS:
/
/
/ EITHER THEY ARE FULL, OR THERE IS NO MORE TO READ.
/
/
/ SB130,
/
/
/
4231 TAD HDRSW /SWITCH TELLS WHETHER HEADER HAS BEEN WRITTEN
4232 SNA CLA /SKIP IF NOT WRITTEN YET
4233 JMP SB140 /BUT IF ITS DONE, DONT RE-WRITE
4234 DCA HDRSW /SET THE SWITCH
4235 CIF H18FLD
4236 JMS I WRITE /WRITE THE HEADER
4237 HDRIO
4238 CIF H18FLD
4239 JMS I WRITE /WRITE BINARY BLOCKS
4240 OUTFIL
4241 CLA
4242 TAD MBCNT /GET MINUS REMAINING COUNT
4243 SNA CLA /SKIP NOT END OF SAVBIN
4244 JMP EDRTN+1 /RETURN TO EDITOR
4245
/
/
/ COME HERE TO RE-INITIALIZE CONTROL BLOCKS
/
/
/ IF MORE BINARY REMAINS TO BE DONE
/
/
/
4247 TAD BFAD /BUFFER START ADDRESS
4248 DCA INFIL+1
4249 TAD BFLN /BLOCKS WRITTEN...
4250 TAD OUTFIL+2 /...PLUS START BLOCK...
4251 DCA OUTFIL+2 /...IS NEW START BLOCK
4252 ISZ INFIL+2 /GO TO NEXT BLOCK IN BWA
4253 JMP SB110 /AND LOOP TO END
4254
0357 EJECT

```

```

0360 /
0361 /
0362 /
0363 /
0364 /
0365 /
0366 /
0367 /
0370 /
0371 /
0372 /
0373 /
0374 /
0375 /
0376 /
0377 /
0400 /
0401 /
0402 /
0403 /
0404 /
0405 /
0406 /
0407 /
0410 /
0411 /
0412 /
0413 /
0414 /
0415 /
0416 /
0417 /
0420 /
0421 /
0422 /
0423 /
0424 /
0425 /
0426 /
0427 /
0430 /
0431 /
0432 /
0433 /
0434 /
0435 /
0436 /
0437 /
0440 /
0441 /
0442 /
0443 /
0444 /
0445 /
0446 /
0447 /
0450 /
0451 /
0452 /

COPY LOOP FOR SAVE PROGRAM (SP)

0002 /ENTER 8-MODE FOR COPY
1331 TAD BFLN /ADD TO MINUS BLOCK COUNT
1327 TAD MBCNT /SKIP IF 16 OR LESS TO GO
7510 SPA
5271 JMP SP120
7200 CLA
1327 TAD MBCNT
7041 CIA
3316 DCA INFIL+3
1316 TAD INFIL+3
3322 DCA OUTFIL+3
5327 DCA MBCNT
6212 CIF HI8FLD
4421 JMS I READ
4313 INFIL
6212 CIF HI8FLD
4422 JMS I WRITE
4317 OUTFIL
7200 CLA
1327 TAD MBCNT
7650 SNA CLA
5336 JMP EDTN+1
1331 TAD BFLN
1315 TAD INFIL+2
3315 DCA INFIL+2
1331 TAD BFLN
1321 TAD OUTFIL+2
3321 DCA OUTFIL+2
5257 JMP SP110

I/O CONTROL BLOCKS

0110 INFIL, SWA /UNIT FOR INPUT
0020 BUFFER /MEM ADDR=1 0000, SHIFTED RIGHT 8
0000 /START BLOCK
0016 BUFLN /NO OF BLOCKS

0000 OUTFIL, /UNIT FOR OUTPUT
0020 BUFFER /MEM ADDR=1 0000
0000 /STARTING BLOCK
0016 BUFLN /COUNT

0111 HDRIO, BWA /HEADER IS ON BIN WORK AREA UNIT
0013 13 /MEM ADDR = 5400 (3400 IF AB)
0057 HDRBLK /BLOCK 47 WITHIN BWA
0001 1 /JUST ONE

0000 MBCNT, /
7777 HDRSW, 7777
0016 BFLN, BUFLN
0020 BFAD, BUFFER
7760 MBFAD, -BUFFER
7742 MBFEND, -BUFFER-BUFLN
5257 EJECT

```

```

0453 0453 RETURN TO EDITOR
0454 0454
0455 0455
0456 0456 /RETURN IN 8-MODE
0457 0457 /DIAL IS IN UPPER MEMORY FIELD
0460 0460
0461 0461 /START OF DIAL RESTART
0462 0462
0463 0463
0464 0464
0465 0465
0466 0466
0467 0467
0470 0470
0471 0471
0472 0472
0473 0473
0474 0474
0475 0475
0476 0476
0477 0477
0500 0500
0501 0501
0502 0502
0503 0503
0504 0504
0505 0505
0506 0506
0507 0507
0510 0510
0511 0511
0512 0512
0513 0513
0514 0514
0515 0515
0516 0516
0517 0517
0520 0520
0521 0521
0522 0522
0523 0523
0524 0524
0525 0525
0526 0526
0527 0527

EDRTN, 0002 /RETURN IN 8-MODE
CIF M18FLO /DIAL IS IN UPPER MEMORY FIELD
JMP I,*1
DRSTRT
LMODE

/SAVE PROGRAM (SP) ENTER HERE
/ LAST BLOCK NO OF WORKING SOURCE IS IN AC
/ FILE NAME AND UNIT IN MC PARAMETER LIST AT E6

/
/
/
/
/
/
/
/ SAVPRG, ADA I
/ -WA=5000*2
/ COM
/ STA I
/ - NUM OF TBLKS
/ GET TWOS COMP
/ HOLD FOR COPY
/ FORCE DATA FIELD FOR REF TO E6
/ CHECK FOR PARTIAL SAVE
/ PARTIAL SAVE ILLEGAL
/ CHK FOR INDEX
/ NO INDEX -- MAKE ONE
/ DO NAME SEARCH
/ NO NAME MATCH
/ CHECK FOR SOURCE
/ ASK REPLACE
/ REPLACE WAS REQUESTED
/ PUT NAME IN INDEX
/ NO ROOM IN INDEX
/ SAVE PTR TO START AND LEN FIELDS
/ DO GAP SEARCH
/ RE-WRITE INDEX
/ OUTPUT UNIT
/ .., TO OUTPUT PARAMETERS
/ GO DO THE COPY
/ EJECT

2341 1120 SAVPRG, ADA I
2342 2411 -WA=5000*2
2343 0017 COM
2344 1060 STA I
2345 0000 /- NUM OF TBLKS
2346 1120 /GET TWOS COMP
2347 0001 /HOLD FOR COPY
2350 4327 /FORCE DATA FIELD FOR REF TO E6
2351 0641 /CHECK FOR PARTIAL SAVE
2352 1000 /PARTIAL SAVE ILLEGAL
2353 2371 /CHK FOR INDEX
2354 0450 /NO INDEX -- MAKE ONE
2355 7071 /DO NAME SEARCH
2356 6464 /NO NAME MATCH
2357 0505 /CHECK FOR SOURCE
2360 6517 /ASK REPLACE
2361 6365 /REPLACE WAS REQUESTED
2362 6436 /PUT NAME IN INDEX
2363 6543 /NO ROOM IN INDEX
2364 6367 /SAVE PTR TO START AND LEN FIELDS
2365 6622 /DO GAP SEARCH
2366 7071 /RE-WRITE INDEX
2367 0044 /OUTPUT UNIT
2370 0002 /.., TO OUTPUT PARAMETERS
2371 6647 /GO DO THE COPY
2372 6775 /EJECT
2373 3017
2374 4317
2375 6256

```

```

0530 /
0531 /
0532 / PRINT SOURCE
0533 /
0534 /
0535 / PRNPRG, LOF DATSEG
0536 / LOF
0537 / E6*2
0540 / SHD I
0541 / 7700
0542 /
0543 / JMP PSWA
0544 / JMP CHKIND
0545 / JMP SAYNO
0546 / JMP LOOKUP
0547 / JMP SAYNO
0550 / JMP SROBN
0551 / JMP GETPS
0552 / JMP SAYNO
0553 /
0554 / COME HERE IF SWA TO BE PRINTED
0555 /
0556 / PSWA, LDA I
0557 / SWA
0558 / STA
0559 / E6*6
0560 / SET I 2
0561 / ZERO
0562 /
0563 / READ PS MAIN CODE -- BLOCKS 63, 4 ON DIAL
0564 /
0565 /
0566 / GETPS, POP
0567 / PMODE
0570 / CIF HI8FLD
0571 / JMS I READ
0572 / PSIN
0573 / LINC
0574 / LMODE
0575 / LOF DATSEG
0576 / LDA 2
0577 / JMP PSENT
0600 /
0601 / PARAMS FOR READING PS AND TTY
0602 /
0603 / PSIN, DIALU
0604 / 12
0605 / PSBLK
0606 / 2
0607 /
0610 / ZERO,
0611 / EJECT

```

```

/CHK MC PARAM
/TABLE
/FILE ENTRY
/OR WA ?
/PRINT THE WA
/NAMED FILE -- READ AND CHECK THE INDEX
/NO INDEX
/INDEX OK -- FIND THE NAME
/NO NAME MATCH
/IS THERE SOURCE?
/YES -- LOAD PS
/BINARY ONLY, CANT DO IT

/SOURCE WA PSEUDO-UNIT
/...TO MC PARAM LIST
/...AS I/O UNIT
/STARTING BLOCK IS ZERO

/PMODE FOR READ

/READ PS AND TELETYPE INTO 5002

/SET DATA FIELD FOR PS
/PICK UP STARTING BLOCK NO OF DESIRED SOURCE
/GO TO PS MAIN PROCESSING

/CONSTANT ZERO TO USE AS STARTING BLOCK IF SWA PRINTED

```

[illegible]

[illegible]

```

0757 /
0760 /
0761 /
0762 /
0763 /
0764 /
0765 /
0766 /
0767 /
0770 /
0771 /
0772 /
0773 /
0774 /
0775 /
0776 /
0777 /
1000 /
1001 /
1002 /
1003 /
1004 /
1005 /
1006 /
1007 /
1010 /
1011 /
1012 /
1013 /
1014 /
1015 /
1016 /
1017 /
1020 /
1021 /
1022 /
1023 /
1024 /
1025 /
1026 /
1027 /
1030 /
1031 /
1032 /
1033 /
1034 /
1035 /
1036 /
1037 /
1040 /
1041 /
1042 /
1042 /
1042 /
1042 /
1042 /
1042 /
1043 /
1044 /

REPLAC, SET 3
0
LDA I
0200
ESF
SET I 1
234
SET I 4
RPLSTR-4000
LDH I 4
SHD I
CR1
JMP K2
ROL I
ADA I
A6
STC 5
DSC 5
DSC I 5
LDA I
2
ADD 1
STC I
JMP REPLAC+11

TEST FOR KEY STRUCK

K2,
0415
0574 6550
0575 0011
0576 0004
0577 0500
0600 6036
0601 0500
0602 6046
0603 1460
0604 0322
0605 6335
0606 0045
0607 0002
0610 1020
0611 5757
0612 1045
0613 1065
0614 6003

0615 2205
0616 2014
0617 0103
0620 0577

0621 4377

RPLSTR, TEXT "REPLACE?"
4377
EJECT

/ASK REPLACE
/ENABLE FULL SIZE CHARACTERS
/SET X-COORDINATE
/POINT TO REPLACE TEXT STRING
/END OF MESSAGE ?
/YES CHK KBD
/ADDRESS DOUBLE-WORD DISPLAY TABLE
/DIAL DISPLAY CHARACTER TABLE
/DISPLAY FIRST HALF OF CHARACTER
/,,,AND SECOND HALF
/INCREMENT Y-COORDINATE
/NO KBD -- REFRESH SCOPE
/RESET TO HALF SIZE
/GET CHAR
/KRB
/ECHO CHAR
/R ?
/NO RETURN
/POINT TO OLD INDEX ENTRY
/SLASHES TO FILL OLD SPACE
/ZAP OLD INDEX ENTRY START BLOCK
/,,,AND LENGTH
/RETURN TO DO REPLACE

```

1045	/	INDEX SLOT SEARCH	
1046	/		
1047	/		
1050	ENTER,	SET I 4	/PTR TO
1051		E6+1	/REQ NAME
1052		SET 1	/RTN JMP
1053		0	
1054		SET I 2	/PTR TO INDEX
1055		INDEX	
1056		JMP ADVIND	
1057		SET 5	
1060		2	
1061		LDA I 2	/FOUND A
1062		SAE I	/BLANK SLOT ?
1063		5757	
1064		JMP ENTER+6	/NO TRY NXT SLOT
1065		LDA I 4	/PUT REQ NAME
1066		STA I 5	/IN INDEX
1067		SRO I	/SKP OUT AFT
1070		3567	/4 PASSES
1071		JMP I-4	
1072		SET 2	/TO END OF
1073		5	/NAME
1074		JMP BUMP-1	/BUMP PTR AND EXIT
1075			
1076	/		
1077	/		
1100	/	FIND THE BEST SPACE FOR A FILE	
1101		FINDSP, SET 10	/B6 IS
1102		0	/DYNAMICALLY
1103		SET I 6	/RESET TO
1104		-FILE	/NEAREST TBLK
1105		LDA I	/SWITCH OFF
1106		7777	/LOWER FILE
1107		STC SWITCH	
1110		JMP GAPSR	/RTN WITH B6 SET
1111			/TO -CLOSEST BLK
1112		SET 7	
1113		6	
1114		CLR	/SWITCH ON
1115		STC SWITCH	/LOWER FILE
1116		SET I 6	
1117		FREE	
1120		JMP GAPSR	
1121		EJECT	

Address	Instruction	Comment
1122	/	
1123	/	
1124	/	FIND NEAREST SPACE OF TWO
1125	/	
1126	0666	1000 LDA
1127	0667	0007 7
1130	0670	0017 COM
1131	0671	2006 ADD 6
1132	0672	1120 ADA 1
1133	0673	7063 -XBLK-XBLK
1134	0674	0451 APO
1135	0675	6700 JMP ,+3
1136	0676	6454 JMP H1
1137	0677	6010 JMP 10
1140	0700	1020 LDA 1
1141	0701	1000 1000
1142	0702	2007 ADD 7
1143	0703	6450 JMP H1+2
1144	0704	6710 JMP ,+4
1145	0705	6454 JMP H1
1146	0706	6010 JMP 10
1147	0707	7071 JMP SAYNO
1150	0710	1000 LDA
1151	0711	0007 7
1152	0712	0017 COM
1153	0713	6010 JMP 10
1154		
1155	0714	0041 GAPSR, SET 1
1156	0715	0000 0
1157	0716	0062 SET 1 2
1160	0717	7000 INDEX:4000
1161	0720	6444 JMP ADVIND
1162	0721	1322 LDH 1 2
1163	0722	1420 SHD 1
1164	0723	5700 5700
1165	0724	6720 JMP ,+4
1166	0725	0222 XSK 1 2
1167	0726	0222 XSK 1 2
1170	0727	0222 XSK 1 2
1171	0730	6735 JMP GETBN
1172		
1173	0731	6757 JMP COMBN
1174	0732	6735 JMP GETBN
1175	0733	6757 JMP COMBN
1176	0734	6720 JMP GAPSR+4
1177		EJECT

ADDRESS	GETBN	DATA	OPERATION	BN	FILE
1200				/GET BN	
1201	0735	1022	LOA I 2	/TBLK NUM	
1202	0736	1560	BCL I	/IN LOW ORDER	
1203	0737	7000		/9 BITS	
1204	0740	1040	STA		
1205	0741	0011	11		
1206	0742	1122	ADA I 2	/NUM OF TBLKS	
1207	0743	0451	AP0	/INACTIVE ENTRY	
1210	0744	0220	XSK I 0		
1211	0745	0017	COM		
1212	0746	1040	STA		
1213	0747	0012	12		
1214	0750	1520	SRO I		
1215	0751	0000	SWITCH, 0	/FOR UPPER OR	
1216	0752	6000	JMP 0	/LOWER FILE	
1217	0753	0052	SET 12		
1220	0754	0011	11		
1221	0755	4011	STC 11		
1222	0756	6000	JMP 0	/BN COMPARE	
1223				/WITH B6	
1224	0757	1000	COMBN, LOA		
1225	0760	0011	11		
1226	0761	2006	ADD 6		
1227	0762	0470	AZE I		
1230	0763	0017	COM		
1231	0764	0451	AP0		
1232	0765	6000	JMP 0		
1233	0766	2345	ADD F1		
1234	0767	0451	AP0		
1235	0770	0470	AZE I		
1236	0771	6000	JMP 0		
1237	0772	0046	SET 6		
1240	0773	0012	12		
1241	0774	6716	JMP GAPSR+2		
1242			EJECT		

1243	/					
1244	/					
1245	/			RE-WRITE THE INDEX		
1246	/			ENTERED WITH STARTING TBLK IN AC		
1247	/					
1250		0775	0053	RSTRX, SET 13		
1251		0776	0000	0		/STORE STARTING BLOCK IN INDEX
1252		0777	1044	STA 4		
1253		1000	1560	BCL I		
1254		1001	7000	7000		
1255		1002	4321	STC OUTFIL+2		/START BLOCK TO PARAMETER LIST
1256		1003	2345	ADD F1		/MINUS FILE LEN
1257		1004	0017	COM		
1260		1005	1064	STA I 4		/STORE LEN IN INDEX
1261		1006	0435	KST I		/LAST CHANCE TO
1262						/INHIBIT COMMAND
1263		1007	6335	JMP EDRTN		
1264		1010	0002	PDP		/CALL WRITE IN 8-MODE
1265				Pmode		
1266		5011	6201	COF L08FLD		
1267		5012	6212	CIF H18FLD		
1270		5013	4422	JMS I WRITE		/RE-WRITE THE INDEX
1271		5014	5017	XIO		
1272		5015	6141	LINC		
1273				Lmode		
1274		1016	6013	JMP 13		
1275	/					
1276	/			I/O PARAMETERS FOR INDEX MANIPULATION		
1277	/					
1300		1017	0000	XIO, 0		/UNIT NUMBER
1301		1020	0006	6		/MEM ADDR = 3000
1302		1021	0346	XBLK		/INDEX TBLK = 346
1303		1022	0002	2		/LENGTH IS 2 BLOCKS
1304				EJECT		

```

1305 /
1306 / LI, QL, OR AS (BY NAME) ENTER HERE
1307 /
1310 /
1311 /
1312 /
1313 /
1314 /
1315 /
1316 /
1317 /
1320 /
1321 /
1322 /
1323 /
1324 /
1325 /
1326 /
1327 /
1330 /
1331 /
1332 /
1333 /
1334 /
1335 /
1336 /
1337 /
1340 /
1341 /
1342 /
1343 /
1344 /
1345 /
1346 /
1347 /
1352 /
1351 /
1352 /
1353 /
1354 /
1355 /
1356 /
1357 /

1023 0011
1024 0641
1025 0057
1026 2371
1027 1440
1030 2372
1031 7035
1032 1440
1033 2371
1034 7071
1035 1000
1036 2371
1037 0017
1040 1100
1041 2372
1042 1200
1043 0017
1044 0452
1045 7071
1046 6464
1047 7071
1050 6517
1051 7071
1052 6436
1053 0456
1054 7071
1055 1000
1056 2377

      CLR          DATSEG
      SET 17
      E6
      SAE
      E6+1
      JMP ,+4
      SAE
      E6
      JMP SAYNO
      LDA
      E6
      COM
      ADA
      E6+1
      LAM
      17
      LZE
      JMP SAYNO
      JMP CHKIND
      JMP SAYNO
      JMP LOOKUP
      JMP SAYNO
      JMP SRORBN
      SKP
      JMP SAYNO
      LDA
      E6+6

      / NO INDEX
      / SEARCH FOR NAME MATCH
      / NO NAME MATCH
      / SOURCE OR BINARY ?
      / SOURCE IS THERE, CONTINUE
      / BINARY ONLY

      / LN2 = LN1
      / LN2 TO 17
      / SKP IF LN2 GE LN1

      / WARNING -- THIS CODE STORES IN THIS SEGMENT

      STC UNITNO
      LDA
      2000
      STC UNITNO-3
      LDA 2
      LIF KBDSEG
      JMP KBDOPR-1
      EJECT

1057 4777
1060 1000
1061 2000
1062 4774
1063 1002
1064 0603
1065 7377

```

```

1360 /
1361 /
1362 /
1363 NOPE, 0000
1364 1066 6141
1365 1067 7071
1366 1070 0050
1367 1071 0050
1368 1072 0000
1369 1073 1020
1370 1074 0200
1371 1075 0004
1372 1076 0011
1373 1077 0061
1374 1077 0061
1375 1100 0344
1376 1101 1760
1377 1102 3077
1378 1103 1760
1379 1104 7706
1380 1105 0061
1381 1106 0374
1382 1107 1760
1383 1110 4177
1384 1111 1760
1385 1112 7741
1386 1113 0415
1387 1114 7077
1388 1115 0011
1389 1116 0004
1390 1117 6335
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415

```

ADDRESS	DATA	INSTRUCTIONS	REMARKS
1416		/	
1417		/	
1420		/	ADD BINARY ENTERS HERE
1421	0641	ADDBIN, LDF DATSEG	/FORCE DATA FIELD TO GET E6
1422	1121	LDF	/FIRST CHAR OF NAME
1423	1122	E6+2	
1424	1123	SHD I	/WAS NAME SPECIFIED?
1425	1124	7700	
1426	1125	JMP SAYNO	/NO - CANT DO IT
1427	1126	LDA I	/SET BUMP CONST
1430	1127	3	
1431	1130	STC BUMP	/...IN LOOKUP
1432	1131	JMP CHKIND	/GET INDEX, CHECK FOR VALIDITY
1433	1132	JMP SAYNO	/NO INDEX - TOO BAD
1434	1133	JMP LOOKUP	/FIND THE NAME
1435	1134	JMP SAYNO	/THE FILE AINT THERE
1436	1135	JMP SRORBN	/IS THERE A BINARY?
1437	1136	SKP	/YES - READ IT
1440	1137	JMP SAYNO	/NO - TOO BAD
1441		/	
1442		/	WHEN WE GET HERE WEVE FOUND THE FILE TO ADD
1443		/	
1444	1140	LDA 2	/FIRST BLOCK OF FILE...
1445	1141	STA	/...IS HEADER
1446	1142	FHDRIN+2	
1447	1143	ADD P1	/SECOND BLOCK
1450	1144	STC BINIPT+2	/...IS START OF BINARY
1451	1145	LDA I 2	/LENGTH...
1452	1146	ADD M1	/...MINUS ONE...
1453	1147	STA	/...IS BINARY BLOCK COUNT
1454	1150	BLOCKS	
1455	1151	COM	
1456	1152	ADD P1	/TWO'S COMPLEMENT
1457	1153	STC MBLKS	/...FOR RELOCATION CONTROL
1460	1154	ADD X10	/INDEX UNIT...
1461	1155	STA	/...IS ALSO FILE UNIT
1462	1156	FHDRIN	
1463	1157	STC BINIPT	
1464	1160	ADD P7	/3400 SHIFTED 8
1465	1161	STC HDRIO+1	/READ HEADER AT 3400
1466	1162	JMP AB000	/GO TO NEXT PAGE
1467		/	
1470	1163	M1, -1	
1471	1164	FHDRIN, 0	/UNIT FOR FILE
1472	1165	6	/HEADER TO 3000
1473	1166	0	
1474	1167	1	
1475	1167	EJECT	

1476	1200	0002	AB000,	*1200	
1477	1500			PDP	
1500				PMODE	
1501	5201	6212		CIF H18FLD	
1502	5202	4421		JMS I READ	/READ BINARY FILE HEADER
1503	5203	5164		FHDRIN	
1504	5204	4700		JMS I PFILBF	/...AND UP TO 15 BLOCKS OF FILE
1505	5205	6212		CIF H18FLD	
1506	5206	4421		JMS I READ	/READ BWA HEADER
1507	5207	4323		HDR10	
1510	5210	4263		JMS SCANHD	/SCAN HEADER FOR STANDARD RELOCATION
1511	5211	1674		TAD I PE6A	/GET RELOCATION FIELD FROM E6
1512	5212	7110		CLL RAR	/MOVE TO LINK
1513	5213	7440		SZA	/SKIP IF FIELD 0 OR 1
1514	5214	4672		JMS I PNOPE	/OTHERWISE ERROR
1515	5215	1673		TAD I PE6	/GET RELOCATION ADDR
1516	5216	7440		SZA	/IF RELOC ADDR,...
1517	5217	5222		JMP ,+3	
1520	5220	7420		SNL	/...AND FIELD ARE ZERO,...
1521	5221	5232		JMP AB010	/...DO NOT RELOCATE
1522	5222	0347		AND P377	/ADDR WITHIN BLOCK
1523	5223	3355		DCA RELADR	
1524	5224	1673		TAD I PE6	
1525	5225	0350		AND P7400	/BLOCK NO - HIGH ORDER BIT IN LINC
1526	5226	7006		RTL	/ MOVE
1527	5227	7006		RTL	/ TO
1530	5230	7004		RAL	/ LOW ORDER
1531	5231	3356		DCA RELBLK	/RELOCATED BLOCK ADDR
1532				EJECT	

ADDRESS	INSTR	OPERAND	COMMENT
1533	0000		NOW BEGINS THE REAL STUFF
1534	0000		SEARCH FOR NON-ZERO IN INPUT
1535	0000		
1536	0000		
1537	0000		
1540	0000		
1541	0000		
1542	0000		
1543	0000		
1544	0000		
1545	0000		
1546	0000		
1547	0000		
1550	0000		
1551	0000		
1552	0000		
1553	0000		
1554	0000		
1555	0000		
1556	0000		
1557	0000		
1560	0000		
1561	0000		
1562	0000		
1563	0000		
1564	0000		
1565	0000		
1566	0000		
1567	0000		
1570	0000		
1571	0000		
1572	0000		
1573	0000		
1574	0000		
1575	0000		
1576	0000		
1577	0000		
1600	0000		
1601	0000		
1602	0000		
1603	0000		
1604	0000		
1605	0000		
1606	0000		
1607	0000		
1610	0000		
1611	0000		
1612	0000		
1613	0000		
1614	0000		
1615	0000		
1616	0000		
1617	0000		
1620	0000		

```

1621 /
1622 /
1623 /
1624 /
1625 /
1626 /
1627 /
1628 /
1629 /
1630 /
1631 /
1632 /
1633 /
1634 /
1635 /
1636 /
1637 /
1640 /
1641 /
1642 /
1643 /
1644 /
1645 /
1646 /
1647 /
1650 /
1651 /
1652 /
1653 /
1654 /
1655 /
1656 /
1657 /
1660 /
1661 /
1662 /
1663 /
1664 /
1665 /
1666 /
1667 /
1670 /
1671 /
1672 /
1673 /
1674 /
1675 /
1676 /
1677 /
1700 /
1701 /
1702 /
1703 /
1704 /
1705 /
1706 /
1707 /

5301 0000 /MOVEWD, 0
5302 1351 TAD PINDAT
5303 0347 AND P377
5304 1355 TAD RELADR
5305 3352 DCA POUTDT
5306 1352 TAD POUTDT
5307 0350 AND P7400
5310 7650 SNA CLA
5311 5316 JMP MW010
5312 1352 TAD POUTDT
5313 0347 AND P377
5314 3352 DCA POUTDT
5315 7001 IAC
5316 1356 TAD RELBLK
5317 3353 DCA OUTBLK
5320 1353 TAD OUTBLK
5321 7041 CIA
5322 1354 TAD CURBLK
5323 7650 SNA CLA
5324 5341 JMP MOVEIT
5325 6201 CDF LOBFLD
5326 1354 TAD CURBLK
5327 7700 SMA CLA
5330 4745 JMS I PWRBLK
5331 1353 TAD OUTBLK
5332 1346 TAD M40
5333 7700 SMA CLA
5334 5247 JMP AB030
5335 1353 TAD OUTBLK
5336 3354 DCA CURBLK
5337 1353 TAD OUTBLK
5340 4744 JMS I PGETBK
5341 1751 /
5342 3752 /
5343 5701 /
5344 5400 /
5345 5427 /
5346 7740 /
5347 0377 /
5350 7400 /
5351 0000 /
5352 0000 /
5353 0000 /
5354 7777 /
5355 0000 /
5356 7777 /

/WEVE GOT A NON-ZERO WORD
/GET ADDR OF WORD
/BLOCK-RELATIVE ADDRESS
/ADD RELOCATION
/OUTPUT DATA PTR
/TEST FOR CARRY
/SKIP ON CARRY
/BLOCK ADDR...
/...TRUNCATED TO 8 BITS
/SET CARRY
/BLOCK RELOCATION
/GIVES BIN BLOCK
/PREP TO COMPARE
/...TO CURRENT BLOCK IN BUFFER
/IN THIS BLOCK?
/YES - DO THE MOVE NOW
/NO - GET THE RIGHT ONE
/ANYTHING IN BUFFER?
/SKIP IF NOT
/ELSE OUTPUT
/NEXT BLOCK OF INTEREST
/IS IT LEGAL?
/SKIP IF OK
/ELSE ABORT
/IF OK, IT IS...
/...NEW CURRENT
/GET BWA BLOCK INTO BUFFER
NOW, MOVE ONE WORD
TAD I PINDAT
DCA I POUTDT
JMP I MOVEWD
/AND RETURN
PGETBK, GETBLK
PWRBLK, WRTBLK
M40, -40
P377, 377
P7400, 7400
PINDAT, 0
POUTDT, 0
OUTBLK, 0
CURBLK, -1
RELADR, 0
RELBLK, -1
EJECT

```

```

1710 1710 00000 GETBLK, 0 /
1711 1711 3254 DCA BWAIO+2 /BLOCK WANTED
1712 1712 1254 TAD BWAIO+2 /ADDRESS CTL WD
1713 1713 1251 TAD POUTH0 /
1714 1714 3246 DCA TMP2 /GET CONTROL WD
1715 1715 1646 TAD I TMP2 /SKIP IF BLK USE
1716 1716 7650 CLA SNA /ELSE CLEAR BUFF
1717 1717 5216 JMP CLRBUF /
1718 1718 / /
1719 1719 / /
1720 1720 / /
1721 1721 / /
1722 1722 / /
1723 1723 / /
1724 1724 / /
1725 1725 / /
1726 1726 / /
1727 1727 / /
1728 1728 / /
1729 1729 / /
1730 1730 6212 CIF HI8FLD /GET IT
1731 1731 4421 JMS I READ /
1732 1732 5452 BWAIO /SET UPPER FIELD FOR MOVE
1733 1733 6211 CDF HI8FLD /
1734 1734 7200 CLA /
1735 1735 5600 JMP I GETBLK /RETURN
1736 1736 / /
1737 1737 / /
1738 1738 / /
1739 1739 / /
1740 1740 / /
1741 1741 / /
1742 1742 / /
1743 1743 / /
1744 1744 / /
1745 1745 / /
1746 1746 / /
1747 1747 / /
1748 1748 / /
1749 1749 / /
1750 1750 / /
1751 1751 / /
1752 1752 / /
1753 1753 / /
1754 1754 / /
1755 1755 / /
1756 1756 / /
1757 1757 / /
1758 1758 / /
1759 1759 / /
1760 1760 / /
1761 1761 / /
1762 1762 / /
1763 1763 / /
1764 1764 / /
1765 1765 / /
1766 1766 / /
1767 1767 / /
1768 1768 / /
1769 1769 / /
1770 1770 / /
1771 1771 / /
1772 1772 / /
1773 1773 / /
1774 1774 / /
1775 1775 / /
1776 1776 / /
1777 1777 / /
1778 1778 / /
1779 1779 / /
1780 1780 / /
1781 1781 / /
1782 1782 / /
1783 1783 / /
1784 1784 / /
1785 1785 / /
1786 1786 / /
1787 1787 / /
1788 1788 / /
1789 1789 / /
1790 1790 / /
1791 1791 / /
1792 1792 / /
1793 1793 / /
1794 1794 / /
1795 1795 / /
1796 1796 / /
1797 1797 / /
1798 1798 / /
1799 1799 / /
1800 1800 / /
1801 1801 / /
1802 1802 / /
1803 1803 / /
1804 1804 / /
1805 1805 / /
1806 1806 / /
1807 1807 / /
1808 1808 / /
1809 1809 / /
1810 1810 / /
1811 1811 / /
1812 1812 / /
1813 1813 / /
1814 1814 / /
1815 1815 / /
1816 1816 / /
1817 1817 / /
1818 1818 / /
1819 1819 / /
1820 1820 / /
1821 1821 / /
1822 1822 / /
1823 1823 / /
1824 1824 / /
1825 1825 / /
1826 1826 / /
1827 1827 / /
1828 1828 / /
1829 1829 / /
1830 1830 / /
1831 1831 / /
1832 1832 / /
1833 1833 / /
1834 1834 / /
1835 1835 / /
1836 1836 / /
1837 1837 / /
1838 1838 / /
1839 1839 / /
1840 1840 / /
1841 1841 / /
1842 1842 / /
1843 1843 / /
1844 1844 / /
1845 1845 / /
1846 1846 / /
1847 1847 / /
1848 1848 / /
1849 1849 / /
1850 1850 / /
1851 1851 / /
1852 1852 / /
1853 1853 / /
1854 1854 / /
1855 1855 / /
1856 1856 / /
1857 1857 / /
1858 1858 / /
1859 1859 / /
1860 1860 / /
1861 1861 / /
1862 1862 / /
1863 1863 / /
1864 1864 / /
1865 1865 / /
1866 1866 / /
1867 1867 / /
1868 1868 / /
1869 1869 / /
1870 1870 / /
1871 1871 / /
1872 1872 / /
1873 1873 / /
1874 1874 / /
1875 1875 / /
1876 1876 / /
1877 1877 / /
1878 1878 / /
1879 1879 / /
1880 1880 / /
1881 1881 / /
1882 1882 / /
1883 1883 / /
1884 1884 / /
1885 1885 / /
1886 1886 / /
1887 1887 / /
1888 1888 / /
1889 1889 / /
1890 1890 / /
1891 1891 / /
1892 1892 / /
1893 1893 / /
1894 1894 / /
1895 1895 / /
1896 1896 / /
1897 1897 / /
1898 1898 / /
1899 1899 / /
1900 1900 / /
1901 1901 / /
1902 1902 / /
1903 1903 / /
1904 1904 / /
1905 1905 / /
1906 1906 / /
1907 1907 / /
1908 1908 / /
1909 1909 / /
1910 1910 / /
1911 1911 / /
1912 1912 / /
1913 1913 / /
1914 1914 / /
1915 1915 / /
1916 1916 / /
1917 1917 / /
1918 1918 / /
1919 1919 / /
1920 1920 / /
1921 1921 / /
1922 1922 / /
1923 1923 / /
1924 1924 / /
1925 1925 / /
1926 1926 / /
1927 1927 / /
1928 1928 / /
1929 1929 / /
1930 1930 / /
1931 1931 / /
1932 1932 / /
1933 1933 / /
1934 1934 / /
1935 1935 / /
1936 1936 / /
1937 1937 / /
1938 1938 / /
1939 1939 / /
1940 1940 / /
1941 1941 / /
1942 1942 / /
1943 1943 / /
1944 1944 / /
1945 1945 / /
1946 1946 / /
1947 1947 / /
1948 1948 / /
1949 1949 / /
1950 1950 / /
1951 1951 / /
1952 1952 / /
1953 1953 / /
1954 1954 / /
1955 1955 / /
1956 1956 / /
1957 1957 / /
1958 1958 / /
1959 1959 / /
1960 1960 / /
1961 1961 / /
1962 1962 / /
1963 1963 / /
1964 1964 / /
1965 1965 / /
1966 1966 / /
1967 1967 / /
1968 1968 / /
1969 1969 / /
1970 1970 / /
1971 1971 / /
1972 1972 / /
1973 1973 / /
1974 1974 / /
1975 1975 / /
1976 1976 / /
1977 1977 / /
1978 1978 / /
1979 1979 / /
1980 1980 / /
1981 1981 / /
1982 1982 / /
1983 1983 / /
1984 1984 / /
1985 1985 / /
1986 1986 / /
1987 1987 / /
1988 1988 / /
1989 1989 / /
1990 1990 / /
1991 1991 / /
1992 1992 / /
1993 1993 / /
1994 1994 / /
1995 1995 / /
1996 1996 / /
1997 1997 / /
1998 1998 / /
1999 1999 / /
2000 2000 / /
2001 2001 / /
2002 2002 / /
2003 2003 / /
2004 2004 / /
2005 2005 / /

```

ADDRESS	INSTR	OPERAND	COMMENT
2006	/		FILL BUFFERS FROM THE BINARY FILE
2007	/		
2010	/		
2011		FILBUF, 0	
2012		5456 0000	/GET REMAINING COUNT
2013		5457 1304	/SUBTRACT BUFFER LENGTH
2014		5460 1501	/SKIP IF NOT ENOUGH TO FILL BUFFER
2015		5461 7500	/ELSE READ 15
2016		5462 5266	
2017		5463 7200	
2018		5464 1504	/REMAINING BLKS
2019		5465 3310	/TO I/O LIST
2020		5466 3504	/STORE NEW REMAINING COUNT
2021		5467 6212	
2022		5470 4421	/READ UP TO 15
2023		5471 5505	
2024		5472 7200	
2025		5473 1307	/ADJUST STRT BLK
2026		5474 1510	/BY LENGTH
2027		5475 3307	
2028		5476 1302	/START OF FIRST BLOCK READ
2029		5477 3703	/RESET POINTER
2030		5500 5656	/RETURN
2031			
2032			
2033			
2034	/		
2035		M15, -15	
2036		5501 7763	
2037		5502 0400	/UNIT WITH FILE
2038		5503 5351	/ADDR = 1 0400
2039		5504 0000	/START BLOCK
2040		5505 0000	/NO TO READ
2041		5506 0021	
2042		5507 0000	
2043		5510 0015	
2044			
2045			

— — — — —

— — — — —

```

WA=370
FILE=470
E6=2371
UNITNO=777
XBLK=346
INDEX=3000
FREE=270
A6=2001
HEADER=1400
USENO=HEADER+337 /COUNT OF BLOCKS USED IN BWA
MAP=HEADER+340 /START OF BLOCK USAGE MAP IN HEADER
KBDOPR=1400 /GO HERE AFTER ASSEMBLY PRE-PROCESSING
PSENT=1000 /ENTRY POINT OF PS MAIN CODE
DRSTRT=777 /UPPER SEGMENT DIAL RESTART ADDR
CRI=4300
DIALU=100
PSBLK=63
SWA=110
BWA=111
HDR8LK=57
AUTO=10
READ=21
WRITE=22
BUFFER=20
BUFLN=16
HI8FLD=10
LO8FLD=0
DATSEG=1
KBDSSEG=3

ASMIFM 6000-.
WARNING // IF THIS ROUTINE EXCEEDS 2000 WORDS OCTAL, EDITOR MUST BE MODIFIED
// TO READ MORE BLOCKS, AND LAP6-DIAL MUST HAVE SPACE FOR THEM.
//
//
//
//
END OF FCOMSV4
```

NO EICIS

SYMBOL	VALUE	DEF	REFERENCES
AB000	5200	1477	1466
AB010	5232	1537	1521 1575
AB020	5233	1540	1547
AB030	5247	1557	1657
ADDRIN	5120	1421	0044
ADVIND	4444	0634	0742 1056 1161
AUTO	0010	2077	0263 0271 1744 1746
A6	2001	2062	1003
BFAD	4332	0447	0350
BFLN	4331	0446	0352 0366 0413 0416
BINIPT	5505	2041	1450 1463 2020 2024 2026 2027 2030
BLOCKS	5504	2040	1454 2012 2017 2021
BUFEND	5276	1615	1571
BUFFER	0020	2102	0426 0433 0447 0450 0451 2002 2042
BUFLEN	0016	2103	0430 0435 0446 0451
BUMP	4536	0751	0223 1074 1431
BWA	0111	2075	0242 0437
BWAI0	5452	2001	1715 1716 1731 1760 1762
CHKIND	4464	0663	0224 0507 0543 1334 1432
CLRBUF	5416	1741	1723
COMPBN	4757	1224	1173 1175
CR1	4300	2071	0777
CURBLK	5354	1704	1645 1651 1661
DATSEG	0001	2106	0067 0502 0535 0575 0677 1312 1421
DIALU	0100	2072	0603
DRSTRT	7777	2070	0461
EDRTN	4335	0456	0343 0412 1031 1263 1414 1563
ENTER	4622	1050	0233 0516 1064
E6	2371	2055	0073 0112 0123 0143 0153 0162 0171 0504 0537 0560 0666 0741 1051 1314 1316 1321 1324 1327 1344 1423 1612 1
FB010	5466	2021	2015
FCSA	4020	0043	
FHORIN	5164	1471	1446 1462 1503
FILBUF	5456	2011	1617 2033
FILE	0470	2054	1104
FINDSP	4647	1101	0237 0522
FREE	0270	2001	1117
F1	4345	0476	0217 0651 1233 1256
GAPSR	4714	1155	1110 1120 1176 1241
GETBLK	5400	1714	1674 1734 1751
GETBN	4735	1201	1171 1174
GETPS	4421	0566	0550
G1	4666	1126	
HDRBLK	0057	2076	0441
HDRIO	4523	0437	0064 0246 0255 0334 1465 1507 1561
HORSW	4330	0445	0326 0331
HEADER	1400	2063	0071 0121 0126 0136 0141 0151 0154 2064 2065
H18FLD	0010	2104	0062 0274 0332 0335 0401 0404 0457 0570 0672 1267 1501 1505 1537 1557 1727 1732 1745 1756 2022
H1	4454	0647	1136 1143 1145
INDEX	3000	2060	0701 0716 0735 1055 1160
INFIL	4313	0425	0243 0252 0276 0306 0314 0315 0317 0351 0355 0375 0376 0403 0414 0415
INSTRT	5502	2036	2031
KBDOPR	1400	2066	1356
KBDSEG	0003	2107	1355
K2	4573	1017	1000
L10RAS	5023	1311	0047
LOOKUP	4517	0732	0226 0511 0545 0745 1336 1434
L08FL	0000	2105	0061 1266 1553 1650
MAKIN	4505	0713	0225 0510
MAP	1740	2065	0176 0260 2000
MBCNT	4327	0444	0221 0300 0341 0367 0373 0400 0410 0501

SYMBOL	VALUE	DEF	REFERENCES
MBFEND	4334	0451	0316
MBLKS	5277	1616	1457 1554
MOVEIT	5341	1667	1647
MOVEWD	5301	1624	1542 1671
MW010	5316	1641	1634
M1	5163	1470	1452
M15	5501	2035	2013
M40	5346	1676	1655
M400	5447	1776	1741
NOPE	5066	1363	1611
NXTBLK	5254	1567	1555
OUTBLK	5353	1703	1642 1643 1654 1660 1662
OUTFIL	4317	0432	0244 0250 0256 0307 0337 0353 0354 0377 0406 0417 0420 0525 1255
PBUSE	5450	1777	1772
POPMOD	4070	0134	0100
PE6	5273	1612	1515 1524
PE6A	5274	1613	1511
PFILBF	5300	1617	1504 1573
PGETBK	5344	1674	1663
PINDAT	5351	1701	1540 1543 1544 1570 1625 1667 2037
PINHDR	5275	1614	1603 1604
PNOPE	5272	1611	1514
POUTDT	5352	1702	1630 1631 1635 1637 1670
POUTHD	5451	2000	1717 1763
PPINDT	5503	2037	2032
PRNPRG	4376	0535	0050
PSBLK	0063	2073	0605
PSENT	1000	2067	0577
PSIN	4431	0603	0572
PSWA	4413	0555	0542
PWRBLK	5345	1675	1556 1653
P1	4347	0500	0214 0247 0251 1447 1456
P2	4161	0236	0220
P3	5127	1430	0222
P377	5347	1677	1522 1545 1626 1636
P7	4667	1127	1464
P7400	5350	1700	1525 1632
READ	0021	2100	0063 0275 0402 0571 0673 1502 1506 1730 2023
RELAOR	5355	1705	1523 1627
RELBLK	5356	1706	1531 1602 1641
REPLAC	4543	0764	0231 0514 1013 1020
RPLSTR	4615	1042	0774
RSTRX	4775	1250	0240 0523
SAVBIN	4026	0057	0043
SAVPRG	4341	0472	0046
SAYNO	5071	1366	0167 0203 0213 0234 0506 0517 0544 0546 0551 1147 1322 1333 1335 1337 1342 1411 1426 1433 1435 1440
SB010	4111	0161	0127
SB020	4134	0211	0076
SB030	4156	0233	0227
SB040	4160	0235	0232
SB100	4164	0241	
SB110	4206	0271	0356
SB120	4223	0314	0301
SB130	4231	0326	0310
SB140	4240	0335	0330
SB160	4254	0355	0273 0321
SCANHO	5263	1601	1510 1574 1606 1607
SETFLD	4060	0120	0110
SP020	4365	0316	0312

SYMBOL	VALUE	DEF	REFERENCES
SP120	4271	0400	0371
SRORBN	4436	0621	0230 0513 0547 1340 1436
SWA	0110	2074	0425 0556
SWITCH	4751	1215	1107 1115
TMP2	5446	1775	1720 1721 1742 1747 1764 1765 1771
UNITNO	0777	2056	1350 1353
USENO	1737	2064	0211 1777
WA	0370	2053	0473
WRITE	0022	2101	0333 0336 0405 1270 1560 1757
WRTBLK	5427	1755	1675 1767 1773
XBLK	0346	2057	1133 1133 1302
XIO	5017	1300	0253 0524 0667 0674 1271 1460
ZERO	4435	0610	0562