SEMINAR NO. DA3015

CYBER ADVANCED FORTRAN WORKSHOP

STUDENT HANDOUT

REVISION RECORD	
REVISION	**DESCRIPTION**
A	Manual Release.
(01-23-83)	
B	Manual Update. Typographical errors.
(02-22-83)	
C	Manual Update. Typographical errors.
(03-22-83)	
D	Manual update for NOS V2.2; drop SEGLOAD support
(02-28-84)	
E	Manual Update. Drop all FTN V4 support including
(04-09-84)	conversion projects.
F	Manual Update.
(12-15-84)	
Publication No. DA3015-1	

REVISION LETTERS I, O, Q AND X ARE NOT USED

CYBER ADVANCED FORTRAN WORKSHOP

Address comments concerning this manual to:

CONTROL DATA CORPORATION
National Coordinator
Bloomington Facility (MNA02B)
5001 West 80th Street
Bloomington, Minnesota 55437

or use Comment Sheet in the back of this manual.

TABLE OF CONTENTS

CYBER ADVANCED FORTRAN WORKSHOP

DA3015 Length: 5 days

The advanced FORTRAN programming language workshop is intended to provide the experienced FORTRAN programmer with a working knowledge of the many features and extensions to the standard ANSI-77 FORTRAN provided by Control Data Corporation FORTRAN V5 compiler. Emphasis will be placed on the interfaces to the specific operating system supporting the compiler. Advanced file structures and access methods will be discussed as well as their relative efficiencies for implementing given applications. Attention will be placed on loader interfaces, especially in the context of segmentation for large applications subsystems, linkage to subprograms implemented in other host programming languages, and the generation of user libraries and their ability to consolidate the maintenance needs of large programming systems. Other topics to be covered include string manipulations and the future extensions mandated by the ANSI FORTRAN standards. Programming exercises will supplement the lectures.

Prerequisites: FORTRAN (DA3013) or a working knowledge of FORTRAN.

Seminar Objectives:

At the completion of the seminar the student should be able to:

1. Understand the basic concepts of the CYBER 170 hardware and how this may be of advantage to the FORTRAN programmer.
2. Understand how NOS and FORTRAN interface so as to take advantage of NOS when writing FORTRAN.
3. Understand the features of ANSI-77 FORTRAN.
4. Understand the parameters available on the compiler call.
5. Understand data representation within the CYBER 170.
6. Understand the optimization levels available with FORTRAN V5.
7. Understand how the various IF statements effect execution speed.
8. Understand how DO-loops may be optimized for fast execution.
9. Use the compilation listing to find clerical errors.
10. Use file name substitution at execution time.
11. Modify the maximum print output.
12. Read and use an XDMP from FORTRAN V5.
13. Request and use a DAYFILE when necessary.
14. Examine Loader Maps and use them with debugging.
15. Know what is available in Post Mortem Dumps with FORTRAN V5.

16. Use PDUMP for debugging.
17. Verify that the correct LGO file is being used.
18. Use both relocatable and absolute binaries.
19. Use the simple loader commands.
20. Create and maintain User Libraries for FORTRAN V5 programs.
21. Use Fast Dynamic Load (FDL) OVCAPs for FORTRAN applications.
22. Understand and use basic CRM concepts.
23. Find the end of a file using CRM.
24. Read and write explicit FORTRAN CRM files.
25. Use Indexed Sequential (IS) files with FORTRAN V5.
26. Understand Multiple Index Processing (MIP) FORTRAN V5 files.
27. Write FORTRAN V5 subroutines.
28. Use both parameters strings and COMMON for data pass pass both parameters strings and COMMON for data passage.
29. Interface FORTRAN V5 subroutines with FORTRAN V5 main programs.
30. Interface FORTRAN V5 subroutines with COBOL V5 main programs.
31. Interface COBOL V5 subprograms with FORTRAN V5 main programs.
32. Interface COMPASS subroutines with FORTRAN V5 calling routines.
33. Call and initialize Sort/Merge from within a FORTRAN V5 module.
34. Understand the concept of BUFFER IN and BUFFER OUT, and know why not to use them in new programs.
35. Understand how to efficiently allocate I/O buffers.
36. Understand how and when to change collating sequences.
37. Manipulate permanent files from within a FORTRAN V5 module.
38. Understand the concepts of Central Memory Manager (CMM).
39. Know what CYBER Data Control System (CDCS) is and when it may be used.
40. Have an awareness of the reasons for converting FORTRAN V4 Extended programs to FORTRAN V5.
41. Have an awareness for the problems in a conversion.
42. Know where to go for information on conversion.
43. Have a general understanding of what the future changes in ANSI FORTRAN are likely to be.

COURSE CHART

1ST DAY	2ND DAY	3RD DAY	4TH DAY	5TH DAY
INTRODUCTION OPERATING SYSTEM OVERVIEW	PROGRAM OPTIMIZATION	CYBER RECORD MANAGER (CRM)	LINKAGES AND INTERFACES	ADVANCED TECHNIQUES
	ERROR ANALYSIS			
			SORT/MERG CALLS	CONVERSION TO FORTRAN V5
ANSI FORTRAN REVIEW/OVERVIEW				
				FUTURE FORTRAN ENHANCE
LUNCH	LUNCH	LUNCH	LUNCH	LUNCH
ANSI FORTRAN REVIEW/OVERVIEW (CONT)	LOADER, AND FAST DYNAMIC LOADER (FDL)	LABORATORY 3	LABORATORY 4	LABORATORY 5
LABORATORY 1				
	LABORATORY 2			

LESSON OUTLINE

CYBER ADVANCE FORTRAN WORKSHOP

LESSONS

1 Operating System Review

2 ANSI FORTRAN Review

3 Sample XDMP and Loader Map

4 Post Mortem Dump

5 Creation of User Libraries

6 CYBER Record Manager

7 Sample FORTRAN Program with SORT Call

APPENDIXES

A Student Projects

B Student Problem Solutions

Lesson 1

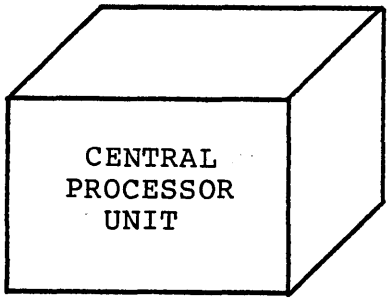
OPERATING SYSTEM REVIEW

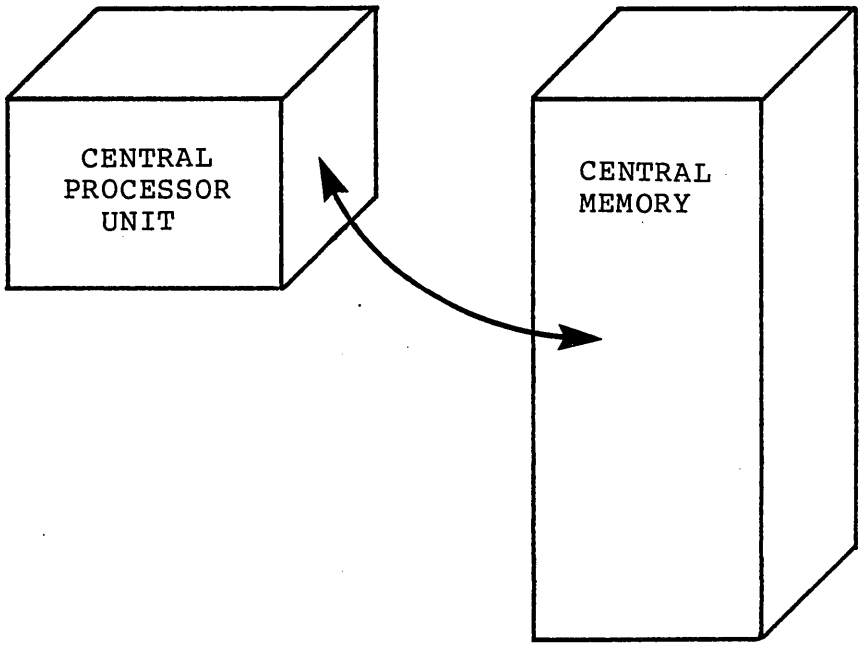
LESSON PREFACE:

OBJECTIVES:

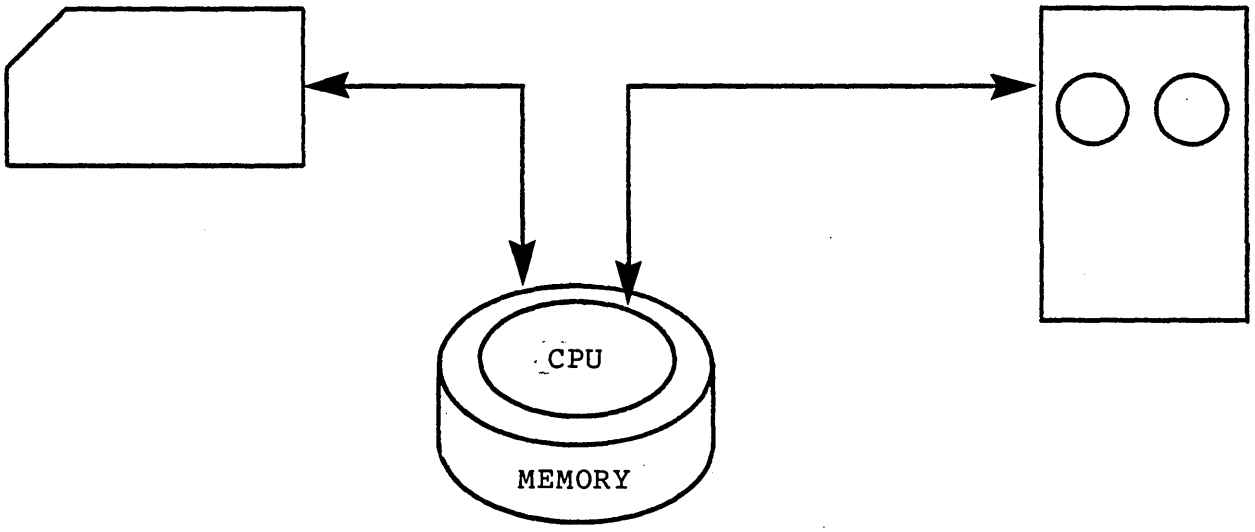
REFERENCES:

PROJECTS:

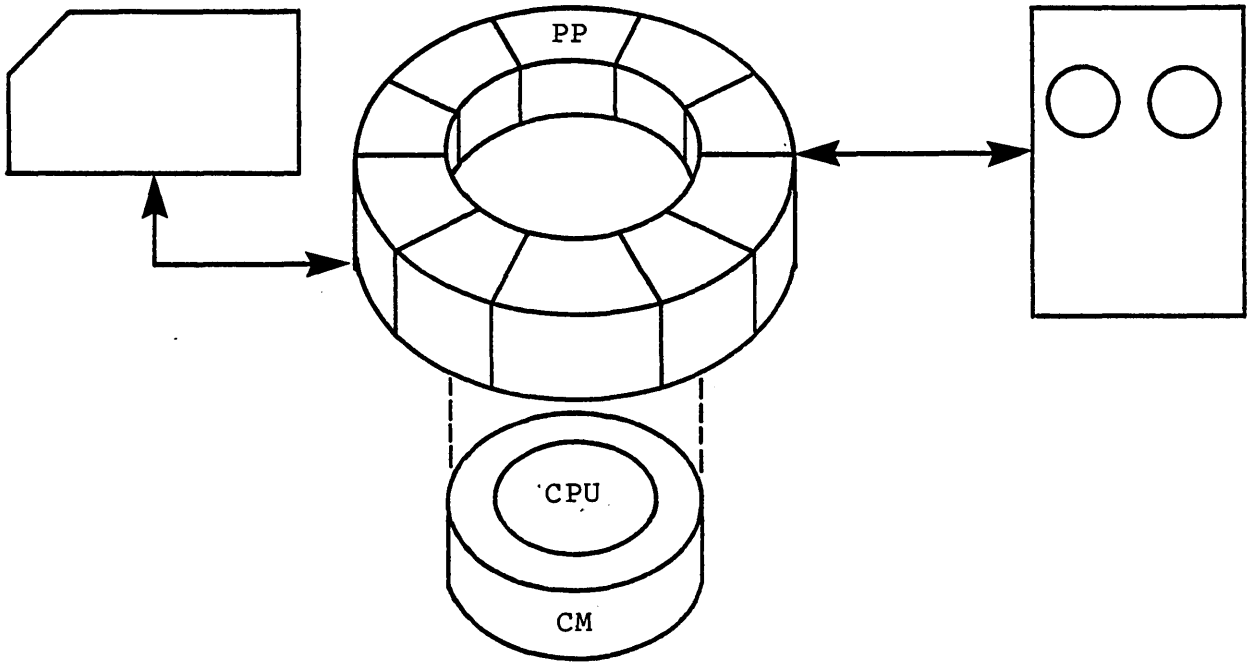

CENTRAL
PROCESSOR
UNIT

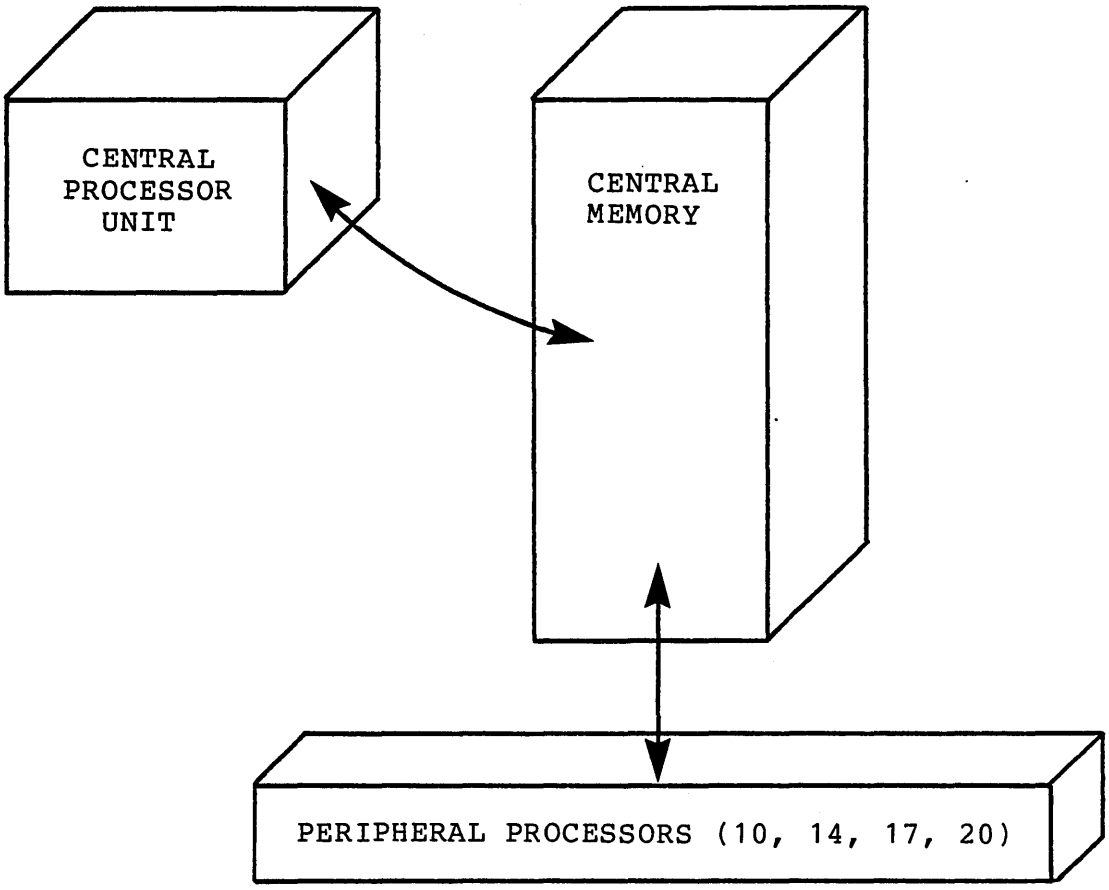

CENTRAL
PROCESSOR
UNIT
CENTRAL
MEMORY

TRADITIONAL I/O HANDLING

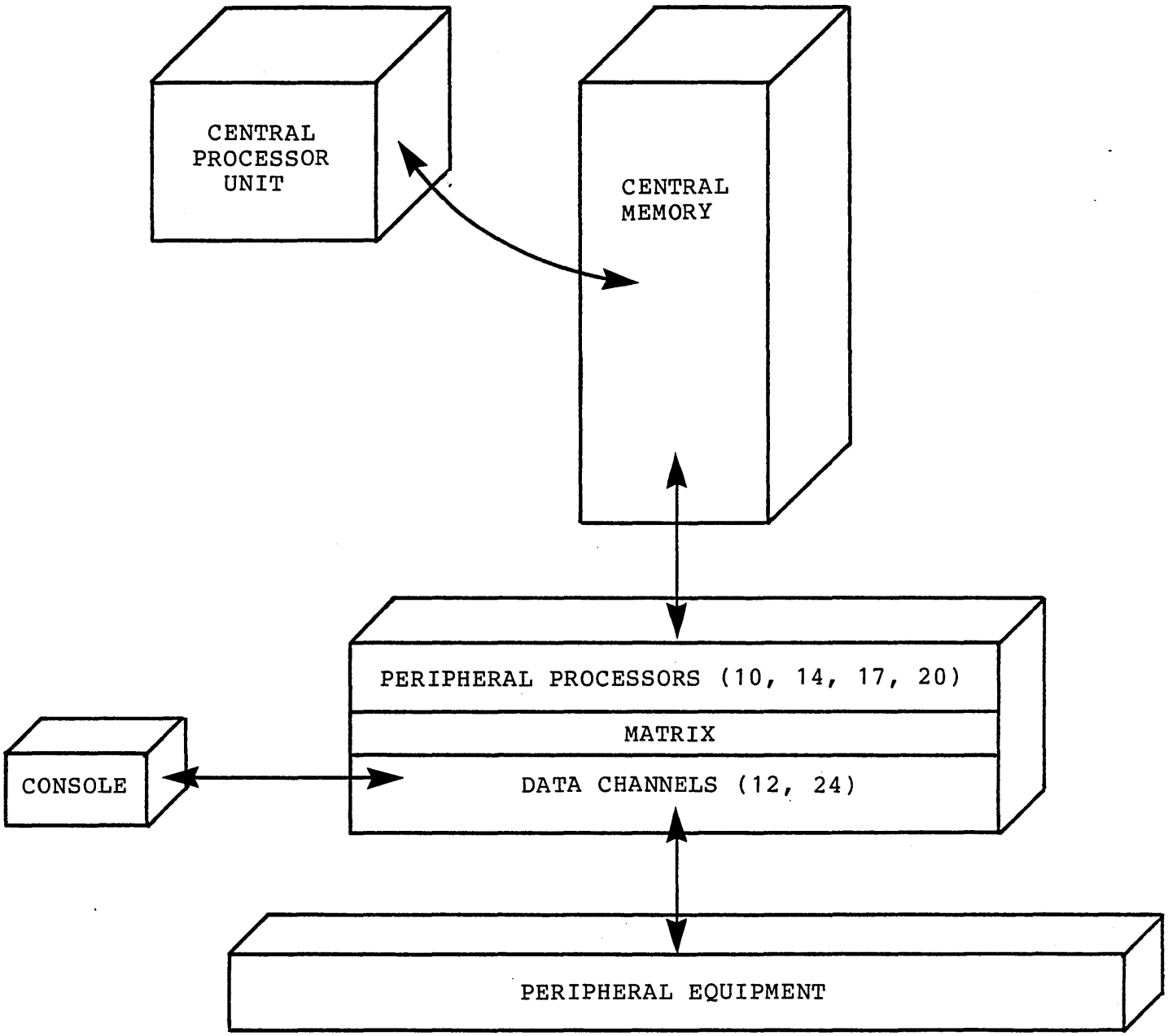


PERIPHERAL PROCESSOR CONCEPT

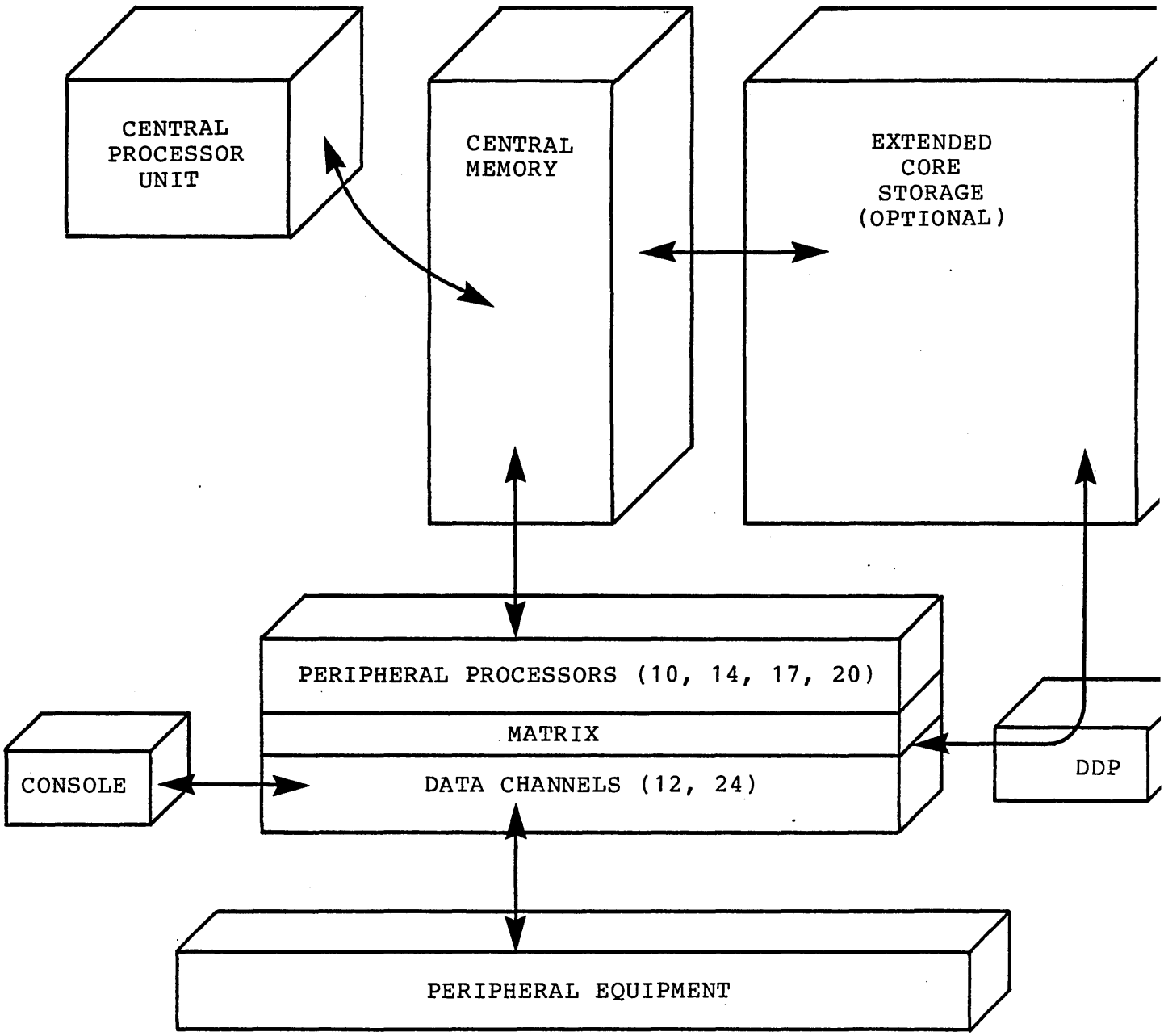


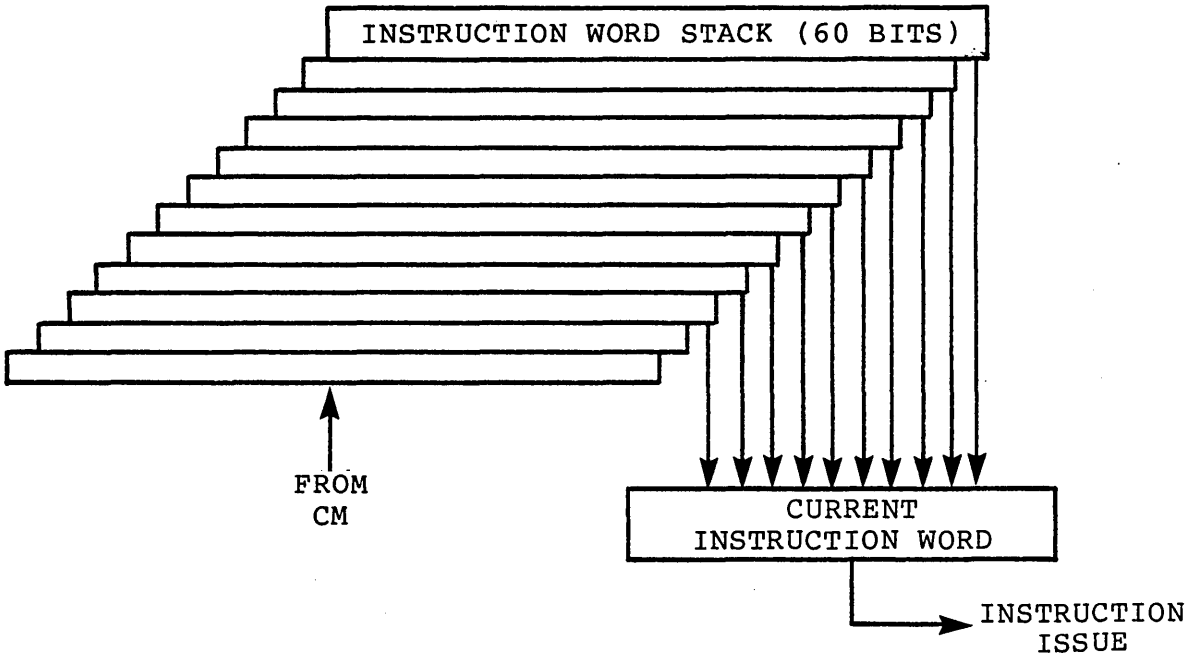

CENTRAL
PROCESSOR
UNIT
CENTRAL
MEMORY
PERIPHERAL PROCESSORS (10, 14, 17, 20)

CDC CYBER 170 SERIES

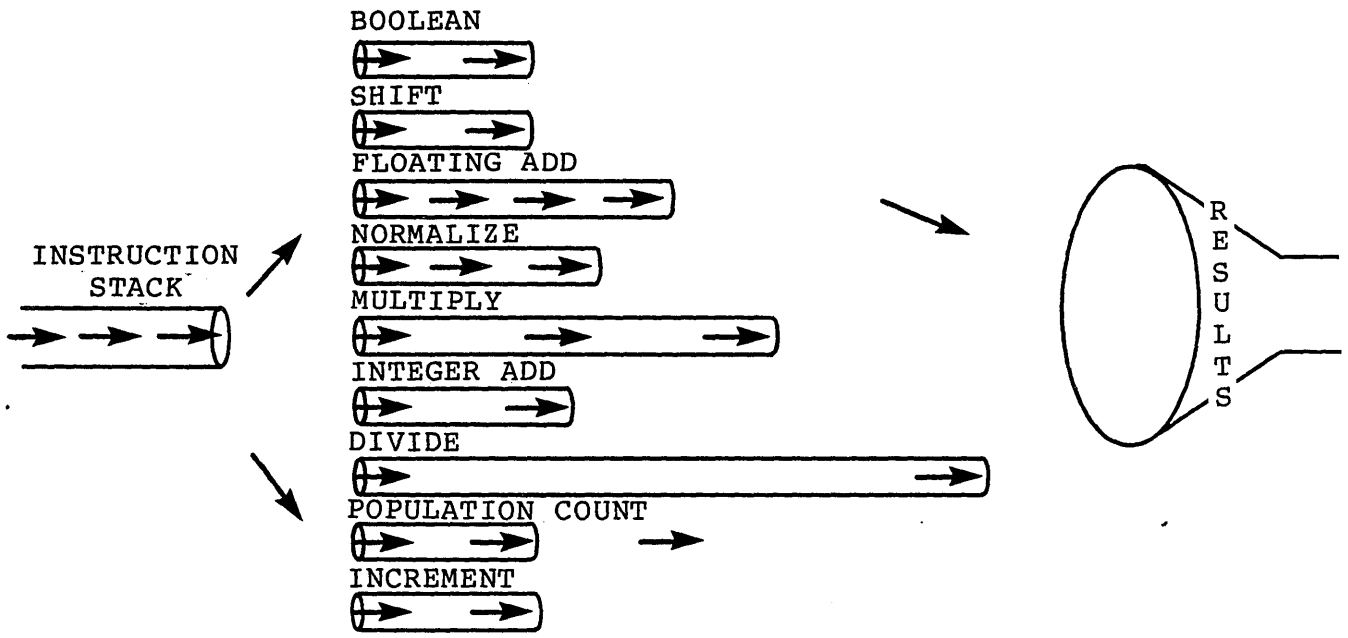


CDC CYBER 170 SERIES

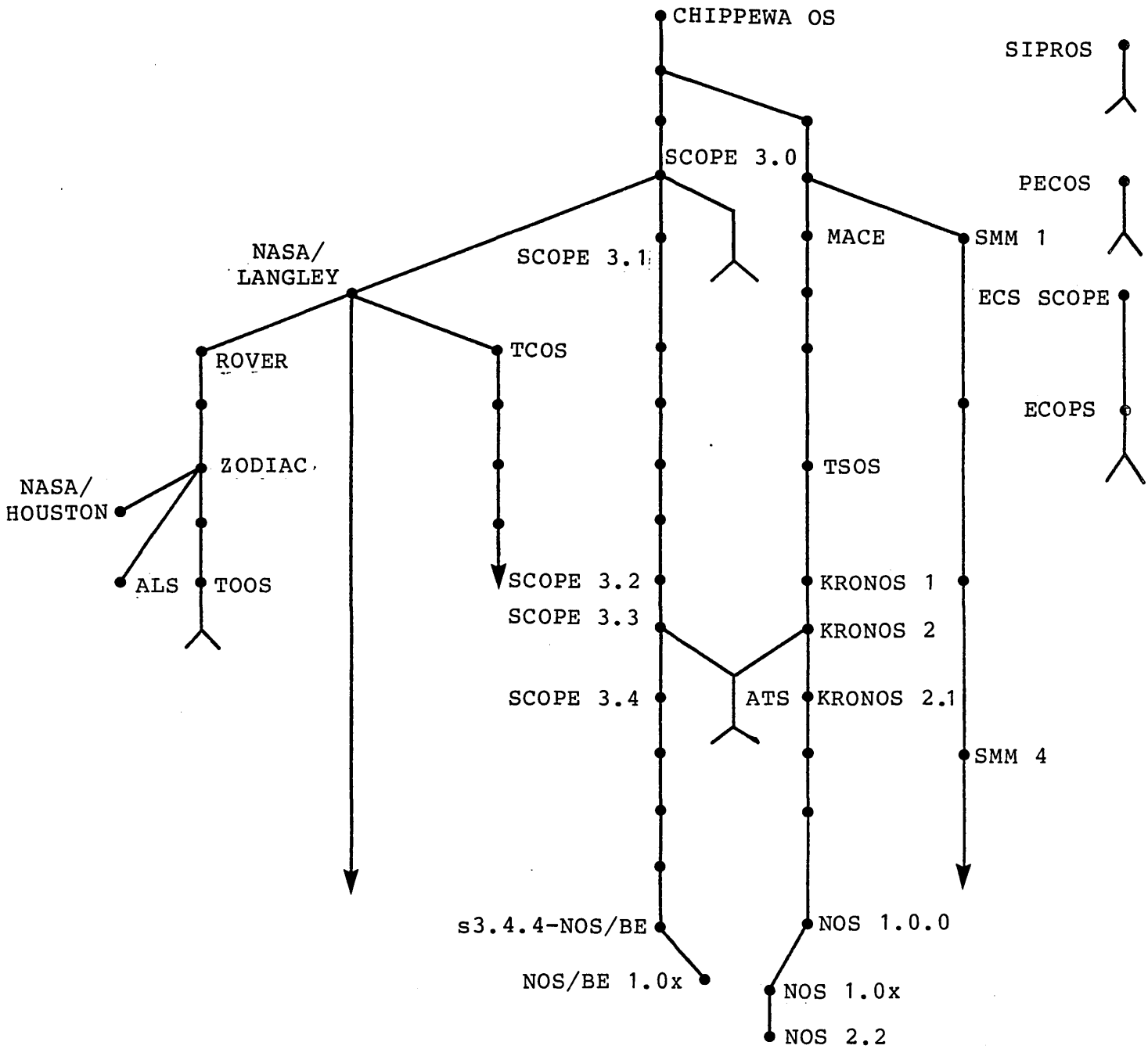



INSTRUCTION WORD STACK (60 BITS)
FROM
CM
CURRENT
INSTRUCTION WORD
INSTRUCTION
ISSUE

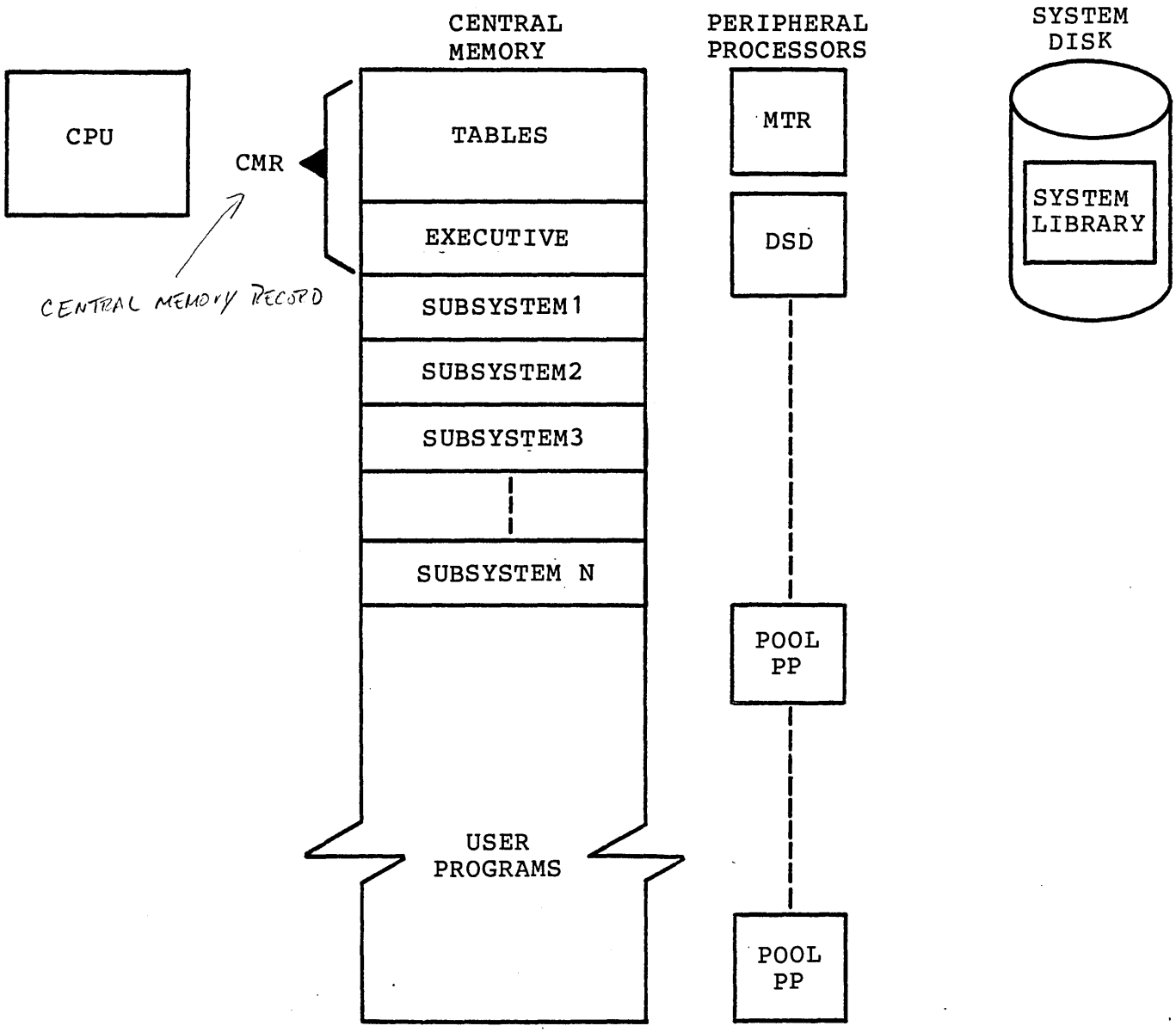
CDC CYBER 170 MODEL 175
PHASED FUNCTIONAL UNITS



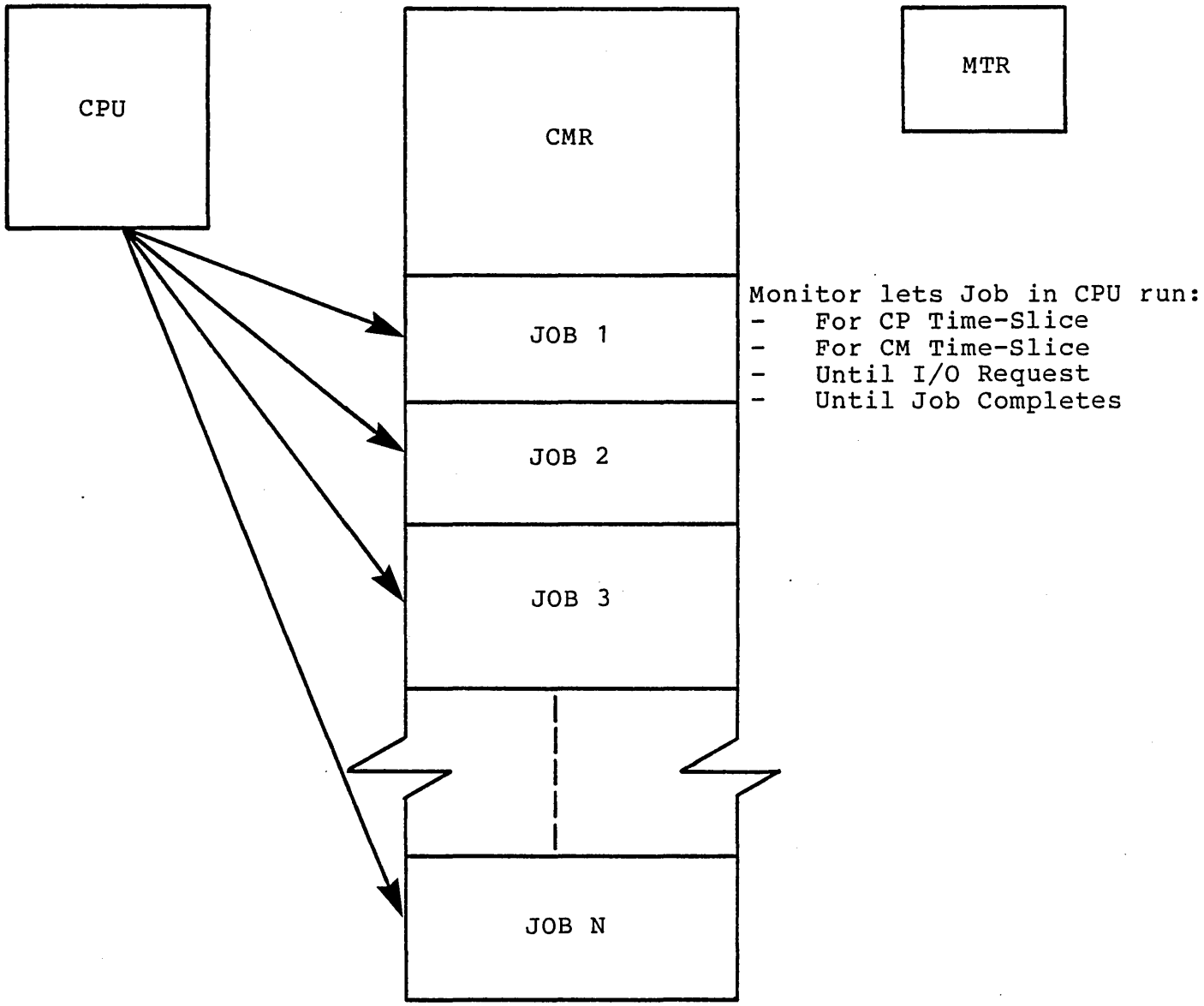
COMMON O/S MODULES
OPERATING SYSTEM GENEALOGY



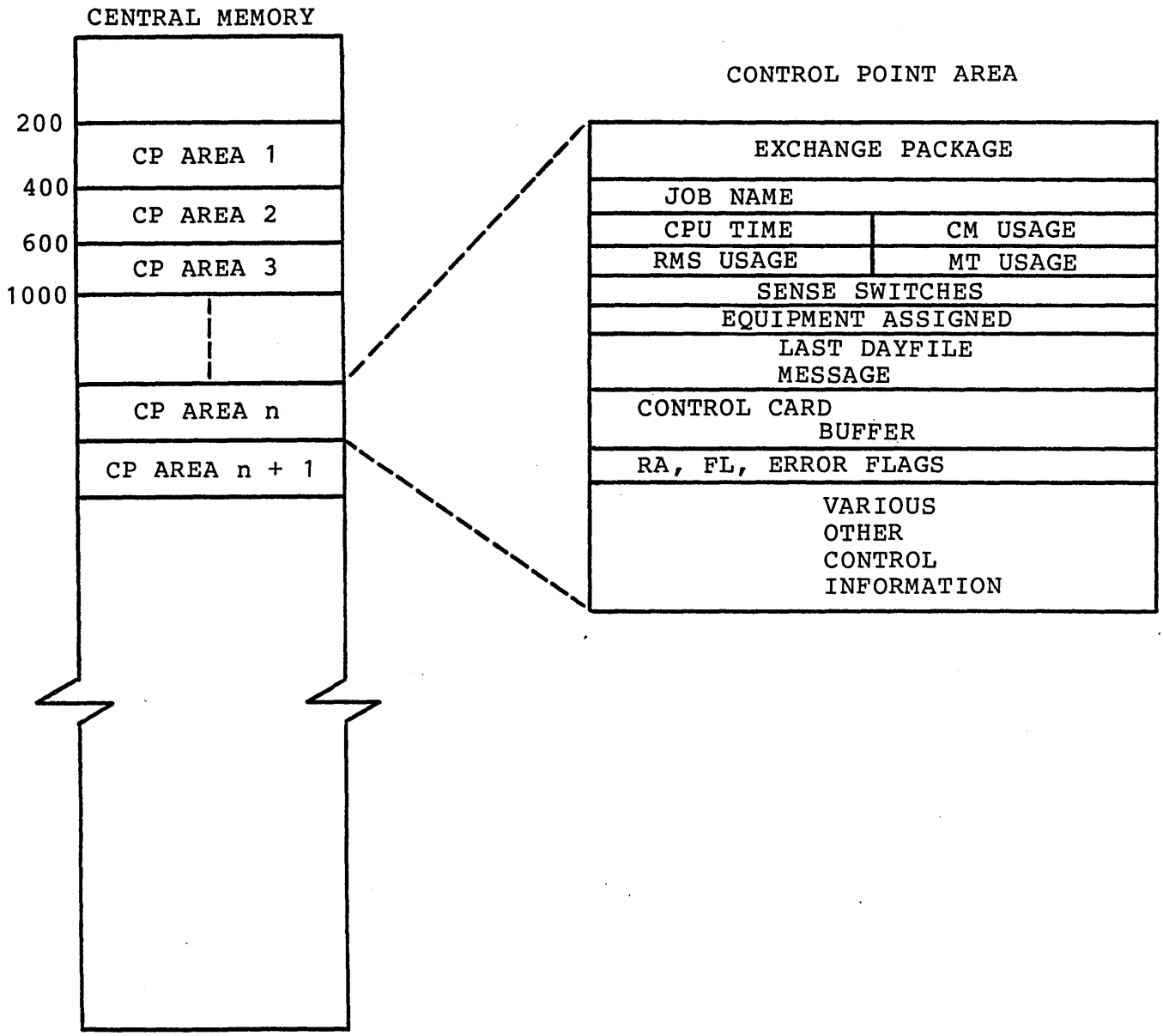
SYSTEM LAY-OUT



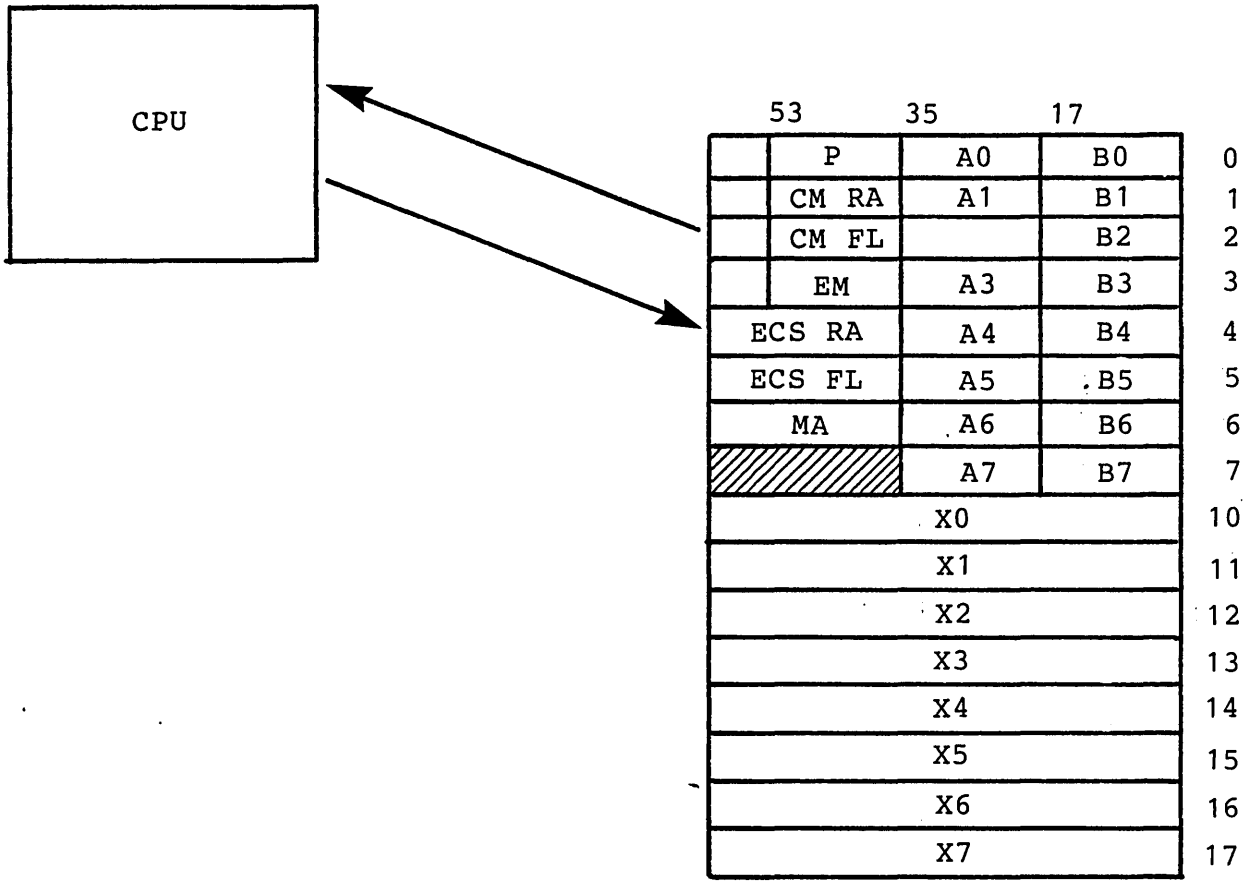
MULTI-PROGRAMMING



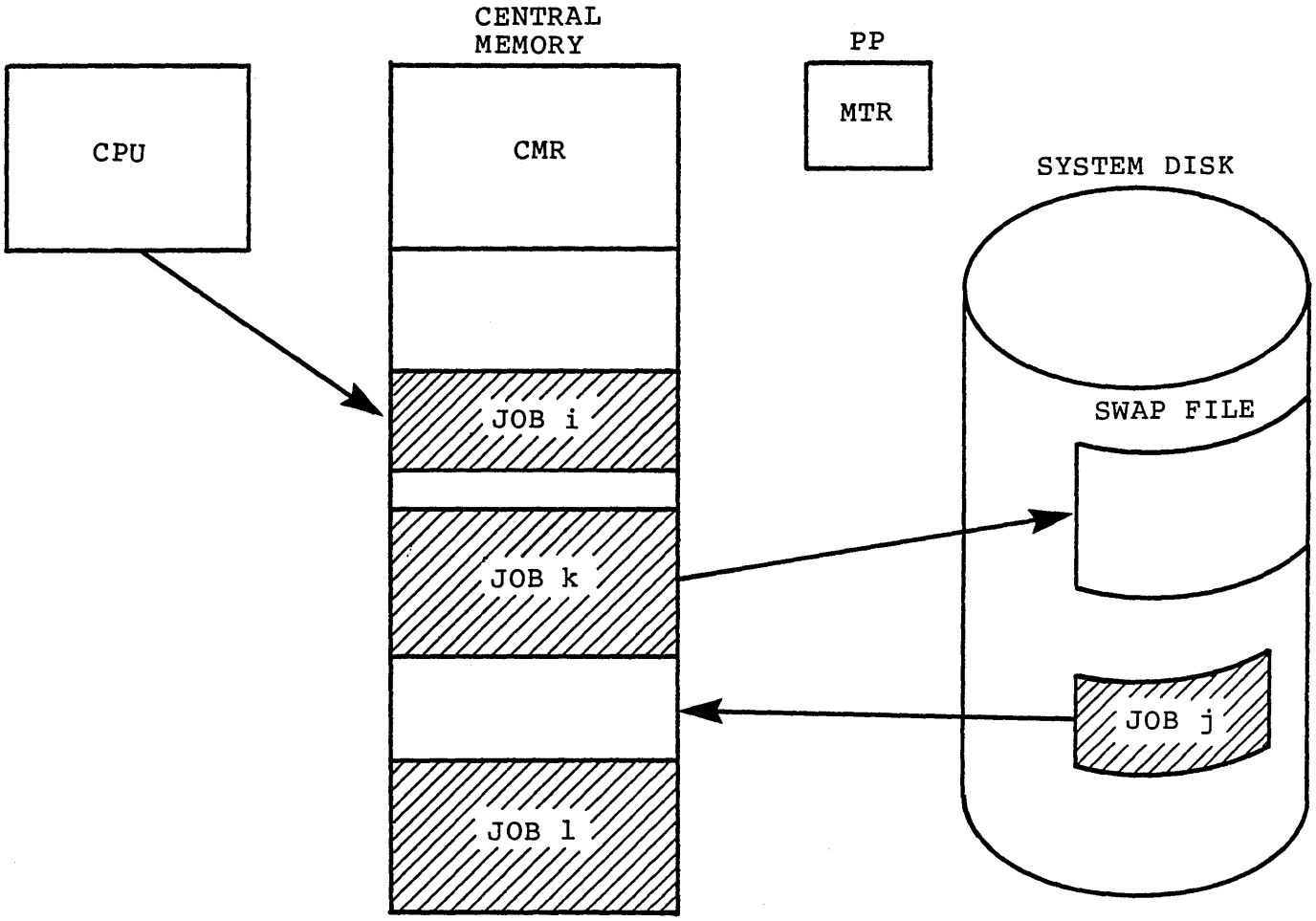
CONTROL POINT CONCEPT



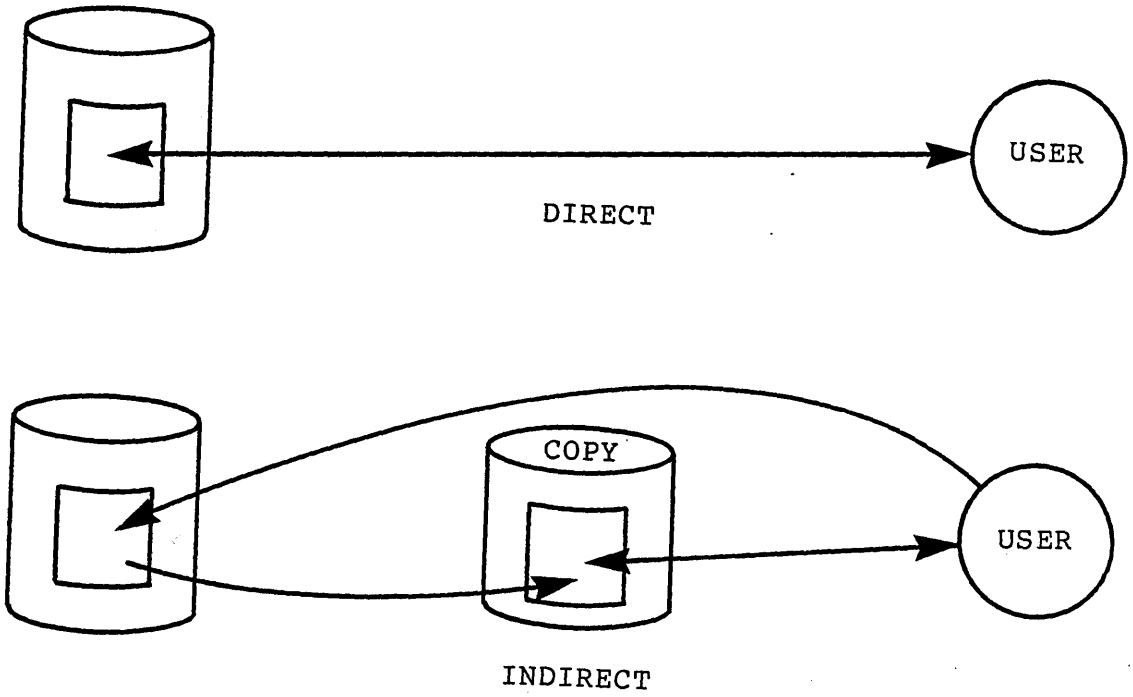
EXCHANGE PACKAGE



SWAPPING


CENTRAL
MEMORY
PP
CPU
CMR
MTR
SYSTEM DISK
SWAP FILE
JOB i
JOB k
JOB j
JOB l

PERMANENT FILES



Lesson 2

ANSI FORTRAN REVIEW

LESSON PREFACE:

OBJECTIVES:

REFERENCES:

PROJECTS:

NEW FEATURES OF ANSI 77 FORTRAN

PARAMETER STATEMENT

BLOCK IF

CHANGES TO THE DO STATEMENT

CHANGES IN ARRAYS

CHARACTER DATA

KEYWORDS IN READ AND WRITE

INTERNAL CHARACTER STRING FILES

DYNAMIC FILE ASSIGNMENT AND INQUIRY

DIRECT ACCESS FILES

CHANGES IN SUBROUTINE CALLS

GENERIC INTRINSIC FUNCTIONS

BLOCK IF

```
IF (CONDITIONAL) THEN
      }
ELSE IF (CONDITIONAL) THEN
      }
ELSE
      }
ENDIF
```

```
IF (A.GT.B) THEN
  |
  v
ENDIF
END
```

```
IF (A.GT.B) THEN
  1
  2 -CODE      TRUE
  3
ELSE
  1
  2 -CODE      NOT TRUE
  3
ENDIF
END
```

66

FORTRAN V4

```
    N=4
    M=2
    J=0
    DO 100 I=N,M
    J=J+1
100 CONTINUE
    PRINT *, J
```

RESULTS:

1

77

FORTRAN V5

```
    N=4
    M=2
    J=0
    DO 100 I=N,M
    J=J+1
100 CONTINUE
    PRINT *,J
```

RESULTS:

0

Using: FTN5,DO=OT....

will give the same results as FTN V4

CHARACTER DATA

- A NEW DATA TYPE THAT ALLOWS THE HANDLING OF CHARACTERS WITHOUT REFERENCE TO THE UNDERLYING MACHINE REPRESENTATION

- WHEN CHARACTER VARIABLES ARE FIRST DECLARED, THE LENGTH OF THE CHARACTER STRING MUST BE SPECIFIED

- OPERATORS ARE

  CONCATENATION (//)

  SUBSTRING SELECTION (:)

4600

0 ——— 46

4600 CHAR'S LONG

CHAR FICA

= 46

```
      CHARACTER PERSON(100)*46,SSN*9,SNAME*22,FNAME*14,INIT*1,PERSONS
      EQUIVALENCE(PERSON,PERSONS)
      ...
      OPEN(UNIT=5,FILE='INPUT',RECL=46)
100   READ(5,1000,END=110)(PERSON(I),I=1,100)
1000  FORMAT(A46)
110   IREC=I
      ...
      SSN=FICA
      J=INDEX(PERSONS,SSN)
      IF(J.EQ.0)GO TO 900
      IF(J.GT.((IREC-1)*46+1)) GO TO 900
      K=J+9
      L=J+31
      M=J+45
      SNAME=PERSONS(K:(K+21))
      FNAME=PERSONS(L:(L+13))
      INIT=PERSONS(M:M)
      ...
      I=1
150   READ(PERSON,'(A9,A22,A14,A1)',END=200) SSN,SNAME,FNAME,INIT
      IF(SSN.NE.FICA) GO TO 150
      ...
900   END
```

RECORD LENGTH IN CHAR'S

→ GO TO 110 WHEN EOF HAS BE ENCOUNTERED

OF CHAR/WORD

OF CHAR'S INTO THE ARRAY

INSURE YOU DID NOT READ PASS EOF

K IS START OF SS#

L IS START OF SIR NAME

M IS START OF FIRST NAME

(K:(K+21)) GET CHAR'S STARTING AT K TO K+21 INCLUSIVE

9 22 14 1

INTERNAL READ

* - READ FROM STANDARD INPUT DEVICE

READ *, A - READ FROM STANDARD INPUT DEVICE CHAR STRIN A

PERSON

4600

TO OUTPUT THE CHAR STRING AND OMIT BLANKS

CHAR LABEL1* 36

LABEL1 = Blank36

LABEL1 = FNAME//' '//INIT//' '// SNAME

CHAR TAB

TAB

READ (TAB,) CHAR

3635

I/O KEYWORDS

```
READ
      }  (UNIT=u, FMT=fmt,REC=rec#,END=sl,ERR=sl,IOSTAT=var name)
WRITE
```

I/O STATUS

IOSTAT = IERR

The first two may be indicated positionally, e.g.:

```
READ (u,fmt,END=statement)
```

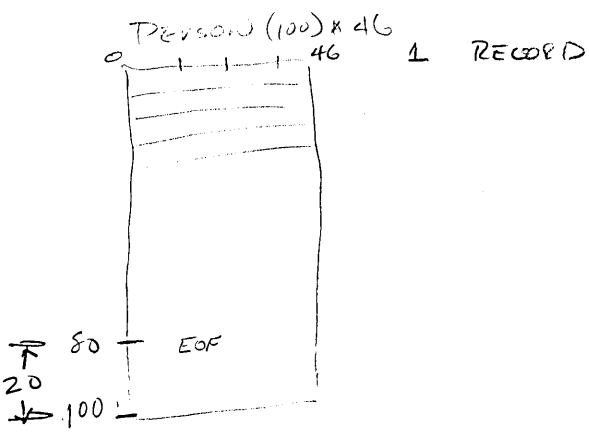
IF IERR .GT. 1000 sub 1000 FROM IERROR AND
go to THE CRM Cyber Record Manager MANUAL
TO FIND Error code

IF (IERR . LT. 1000) GO TO FTN MANUAL TO GET code.

RECORD 80 = EOF

IREC = 80

((IREC - 1) * 46 + 1) = 3635



$$\begin{array}{r} 4600 \\ -\ 3635 \\ \hline 985 \end{array}$$

$$46 \overline{)985} \quad 20$$

UNIT = u

u is

- * implying STANDARD INPUT on READ
  OUTPUT on WRITE

- an integer expression whose result is 0 - 999, the tape unit number

- a Boolean expression whose value is the display code file name, left justified and zero filled

FMT = fmt

where fmt may be

- a statement label pointing to a format statement
- a character expression containing a format specification
- a non-character array or array element containing the format specification
- an integer variable that has been given the value of a format statement label using an ASSIGN statement.
- * indicating list directed I/O
- a NAMELIST group name

END = statement label

The program jumps to the statement on end of file.

ERR = statement label

The program jumps here if I/O returns an error code.

IOSTAT = integer variable

If an error code is returned, this variable will contain its value.

REC = record number for direct access files

FILE STATUS

THE ATTRIBUTES OF FORTRAN I/O FILES CAN BE SPECIFIED IN TWO WAYS - IN THE PROGRAM STATEMENT, OR WITH THE OPEN, CLOSE, AND INQUIRE STATEMENTS. FILE SPECIFICATION IN THE PROGRAM STATEMENT IS A CONTROL DATA EXTENSION TO ANSI FORTRAN (FORTRAN-68 HAD NO STANDARD IN THIS AREA). FILE SPECIFICATION USING OPEN/CLOSE/INQUIRE CONFORMS TO THE ANSI-77 STANDARD, AND IS THUS COMPATIBLE WITH NON-CONTROL DATA MACHINES.

EXAMPLES:

I/O ON UNIT 9 . . . . . WILL BE DONE WITH FILE INSTR

```
OPEN(UNIT=9,ERR=110,FILE='INSTR',STATUS='OLD')
```

IF THERE IS AN ERROR AT OPEN, GO TO STATEMENT 110

THE FILE ALREADY EXISTS

```
CLOSE(9)
```

UNIT 9 IS DISCONNECTED

FORTRAN STANDARD DIRECT ACCESS FILES

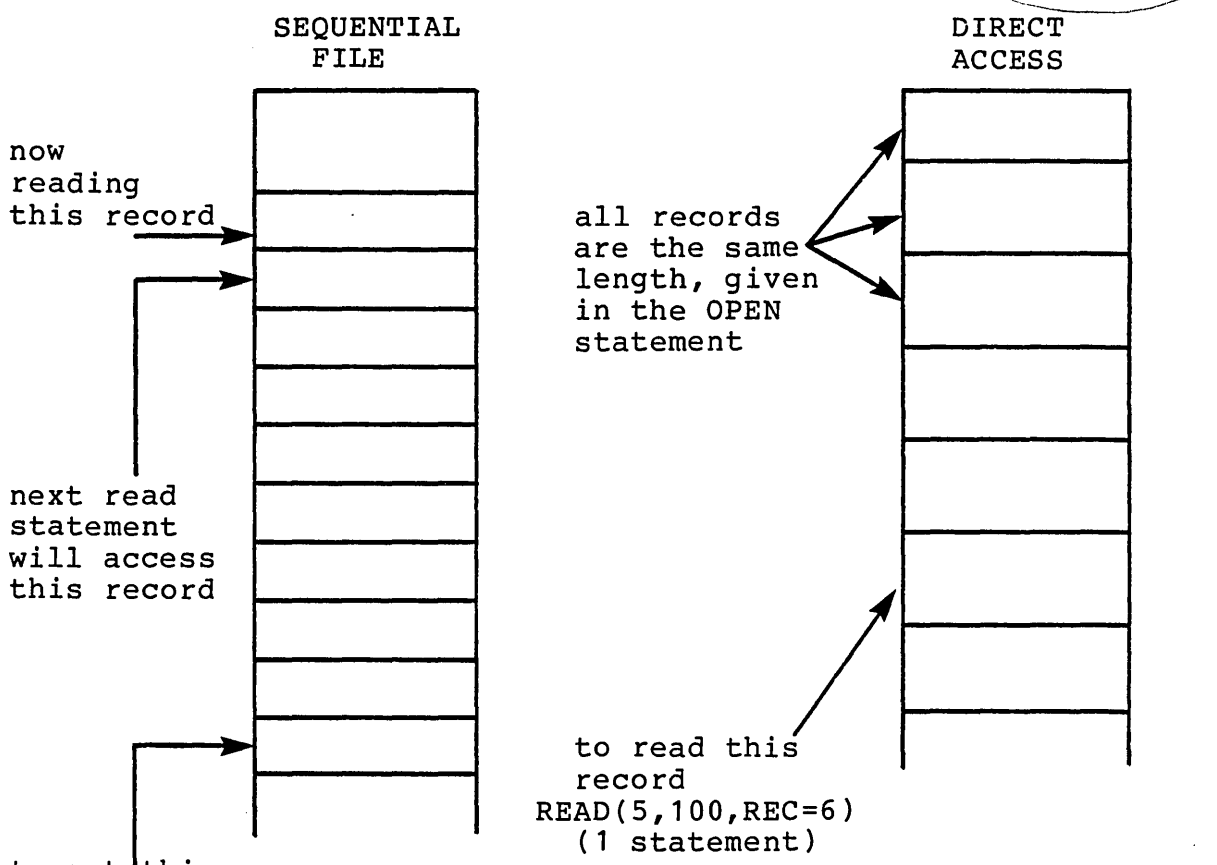
FORTRAN OPEN/CLOSE statements can be used to specify direct access I/O. Each record in a direct access file has a record number. All records must be fixed length. A read or write on the file specifies the record number, so the file can be accessed in any order.

EXAMPLE:

ordinal 5

```
OPEN(5, FILE='INVREC', ACCESS='DIRECT', RECL = 120)

READ (5, 100, REC = 7) WSA

100 FORMAT (1......
```

120 characters or 12 words



FOR POSITIONING:
- REWIND
- BACKSPACE
- ENDFILE
- READ/WRITE

FOR POSITIONING:
"REC = " IN READ STATEMENT, OTHERWISE POSITION IS MEANINGLESS

INTERNAL FILES

Internal files allow the programmer to reformat data by moving it in memory, in particular they are used to translate numeric data into its character representation and vice versa. There are two ways to handle internal files:

ENCODE/DECODE

The ENCODE/DECODE statements were non-standard Control Data extensions to FORTRAN IV which have been retained in FORTRAN 5 for compatibility. ANSI FORTRAN 77 provides a standard for internal files, so ENCODE/DECODE is no longer needed and its use is discouraged.

Standard FORTRAN Internal Files

Rely on formatted READ and WRITE statements, in which the unit identifier is a block of memory containing character data - a character variable, array, or substring. Data in this field is treated as a character string on an external device. If this is a character array, each element of the array is treated as a separate record.

```
DJM -  A(4)

ENCODE (# OF CHARS, FMT #, A) X, I
                 17      1000

1000  FORMAT (F10.2, I7)
```

EXAMPLE:

THE INTERNAL FILE
CONTAINS 20 RECORDS
OF 10 CHARACTERS EACH

```
CHARACTER * 10 CARRAY (20)

INTEGER  IAR  (20)

     .
     .
     .
     .

WRITE (CARRAY, 'I10')     IAR
```

INTEGER DATA
WILL BE WRITTEN....

... INTO THIS ARRAY

SUBROUTINES

SAVE Statement

Indicates that program entities this routine are to be saved for the next time the routine is called. FTN5 saves then anyway, although ANSI standard does not require it.

PARAMETER Lists

Alternate returns are indicated by an asterisk.

Each entry point has its own parameter list, and does not take the parameter list of the main entry point.

The number of actual arguments passed may be less than the number of dummy arguments in the entry point.

Arrays and character strings within a subprogram may be given an assumed size using an *.

ALTERNATE RETURN POINTS

```
      .
      .
      .
      CALL SUB1 (A,B,C, *100, D,*200)
      .
      .
      .
100   ____________________________
      .
      .
200   ____________________________
      .
      .
      .
      END
      SUBROUTINE SUB1 (X,Y,Z,*,A,*)
      .
      .
      .
      RETURN 1
      .
      .
      .
      RETURN 2
      .
      .
      .
      RETURN
      END
```

* = DUMMY ARG

* = DUMMY ARG IF YOU GOT TO RETURN 1 GO TO 100

IF YOU GOT TO RETURN 2 GO TO 200

RETURN WITH OUT A # DEFAULT and GO TO 100

GENERIC INTRINSIC FUNCTIONS

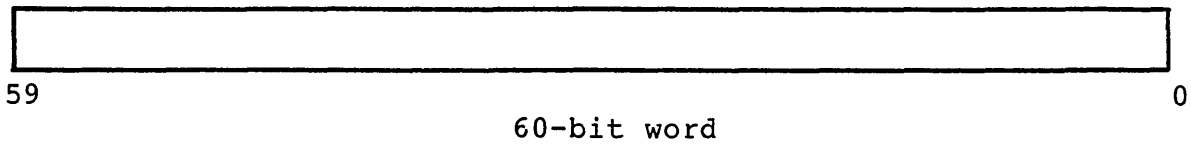
- The type of the argument determines the type of the result

EXAMPLE: AMIN1
DMIN1
MINO

May all be replaced by

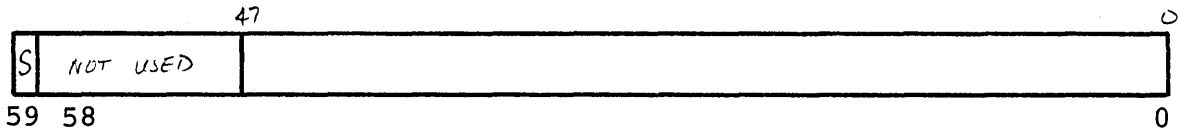
MIN

CYBER WORD STRUCTURE

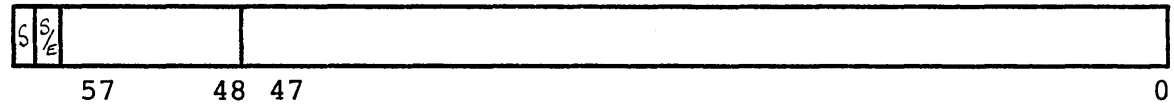


C1	C2	C3	C4	C5	C6	C7	C8	C9	C10

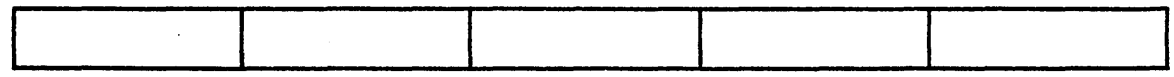
HOLERITH - 10, 6 bit characters



INTEGER - 59 sign bit
47-0 value in binary



REAL - 59 sign bit of value
58 sign bit of exponent
58-48 exponent
47-0 mantissa



ASCII and EBCDIC - 5 12 bit-bytes

RECURSION --

"THE ABILITY TO TALK TO YOURSELF AND GET AN ANSWER"

... BUT DID YOU GET THE RIGHT ANSWER?

Lesson 3

Sample XDMP and Loader Map

LESSON PREFACE:

OBJECTIVES:

REFERENCES:

PROJECTS:

```
get,lf121
/ftn5,i=lf121,lo=r/a/s/m/o,pw=132,b=0
1          PROGRAM IFS        73/176  OPT=0                          FTN 5.1+528        81/10/12. 16.36.58        PAGE     1

      1                PROGRAM IFS
      2                DIMENSION R(1000)
      3          C
      4          C     THIS PROGRAM DOES NOTHING -- IT IS FOR USE ONLY IN SHOWING THE
      5          C     OCTAL CODE GENERATED BY THE VARIOUS TYPE OF IF STATEMENTS.
      6          C
      7                A=-5.2
      8                B=4.3
      9                C=2.8
     10                I=5
     11                IF(A) 100,110,120
     12          100   GO TO (110,120,120,140,140) I
     13          110   IF(C.GT.A) GO TO 140
     14          120   DO 130 J=1,10
     15                A=(B+A)/J
     16          130   CONTINUE
     17          140   DO 150 I=1,1000
     18                R(I)=SQRT(B/C)+I
     19          150   CONTINUE
     20                STOP
     21                END

--VARIABLE MAP--(LO=M/A/R)                                                          A=ARGLIST, C=CTRL OF DO, I=DATA INIT,
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE---REFERENCES-        R=READ, S=STORE, U=I/O UNIT, W=WRITE

  A        2057B                              REAL                   7/S   11     13     15     15/S
  B        2060B                              REAL                   8/S   15     15     18/A
  C        2061B                              REAL                   9/S   13     18/A
  I        2062B                              INTEGER               10/S   12     17/C   18     18
  J        2063B                              INTEGER               14/C   15
  R         107B                              REAL        1000       2     18/S

--PROCEDURES--(LO=M/A/R)                                                            D=DEF LINE OF STMT FUNC
 -NAME----TYPE------ARGS----CLASS-----REFERENCES-                                   A=ACTUAL ARGUMENT

  SQRT    GENERIC      1    INTRINSIC    18

--STATEMENT LABELS--(LO=M/A/R)                                                      A=ASSIGN STMT, D=DO STMT,
 -LABEL--ADDRESS---PROPERTIES-----DEF--REFERENCES-                                  R=READ, W=WRITE

    100   INACTIVE                 12     11
    110     32B                    13     11     12
    120     35B                    14     11     12     12
    130   INACTIVE DO-TERM         16     14/D
    140     52B                    17     12     12     13
    150   INACTIVE DO-TERM         19     17/D
1          PROGRAM IFS        73/176  OPT=0                          FTN 5.1+528        81/10/12. 16.36.58        PAGE     2

--ENTRY POINTS--(LO=M/A/R)                                                          D=DEFINITION
 -NAME---ADDRESS--ARGS------REFERENCES-

  IFS        5B     0          1/D
```

```
--DO LOOPS--(LO=M/A/R)
 -LABEL --ADDRESS----PROPERTIES--------------INDEX---FROM-----TO

   130  *NO REFS*                                J        14      16
   150  *NO REFS*  XREF                          J        17      19
1          PROGRAM IFS          73/176  OPT=0                          FTN 5.1+528        81/10/12. 16.36.58        PAGE      3
           OBJECT LISTING.

           BLOCK       ADDRESS     LENGTH
           START.          0B          6B
           CODE.           6B         67B
           LITERL.        75B          5B
           FORMAT.       102B          0B
           TEMPS.        102B          1B
           APLST.        103B          1B
           IOAPL.        104B          0B
           NAMLST.       104B          0B
           VARS.         104B       1762B

           SUB.         2066B          0B
           SUBO.        2066B          0B
           BUFER.       2066B          0B

                                                     IDENT   IFS
     0B  1106235555555000005 +          IFS         TRACE.
     2B                                 FILVEC.     BSS     0
     2B  0000000000001000000                        ADDR    0,1
    3B  0000000000000011610                         PLIM    5000
                                                    USE     LITERL.
    75B  1722514631463146314 6          CON.        CON        1722514631463146314 6B        OR(-Y-Y-Y-
    76B  6055263146314631463 1                      CON        6055263146314631463 1B        * VY-Y-Y-Y
    77B  1722423146314631463 2                      CON        1722423146314631463 2B        OR7Y-Y-Y-Z
   100B  1721546314631463146 3                      CON        1721546314631463146 3B        OQ=ZLZLZLZ
   101B  0000000000000000000                        CON        0000000000000000000 0B
   101B                                             USE     FORMAT.
                                                    USE     APLST.
   103B                                 AP.1        BSS     0
   103B  0000000000000000101                        APL     0000,CON.+4
                                                    USE     IOAPL.
                                                    USE     NAMLST.
                                                    USE     CODE.
                                        *                       LINE 7
    6B  6102000007                                  SB0     B2+0+7
                 5110000076+                        SA1     CON.+1
    7B  10711                                       BX7     X1
            5170002057      +                       SA7     A
                                        *                       LINE 8
   10B  6102000010                                  SB0     B2+0+10B
                 5120000077+                        SA2     CON.+2
   11B  10622                                       BX6     X2
            5160002060      +                       SA6     B
                                        *                       LINE 9
   12B  6102000011                                  SB0     B2+0+11B
                 5130000100+                        SA3     CON.+3
   13B  10633                                       BX6     X3
            5160002061      +                       SA6     C
                                        *                       LINE 10
   14B  6102000012                                  SB0     B2+0+12B
                 7160000005                         SX6     0+5
   15B  5160002062          +                       SA6     I
                                        *                       LINE 11
   16B  6102000013                                  SB0     B2+0+13B
                 0307000032+                        ZR      X7,.110
   17B  0327000035          +                       PL      X7,.120
                                        *                       LINE 12
```

DA3015

3-3

TRM LIST
TMD LIST

LIST CODE E
LOG A

OBJECT LISTING.

```
20B  6102000014                          SB0    B2+0+14B
           7100000005                    BX0    0+5
21B  5110002062          +               SA1    I
           7211777776                    BX1    X1+0-1
22B  37210                               IX2    X1-X0
        16312                            BX3    -X2+X1
           21373                         AX3    73B
                15413                    BX4    -X3*X1
23B  11530                               BX5    X3*X0
        36045                            IX0    X4+X5
           63600                         SB6    X0
24B  0260000025          +               JP     B6+GL.1
25B                                GL.1  BSS    0
25B  0400000032          +               EQ     .110
26B  0400000035          +               EQ     .120
27B  0400000035          +               EQ     .120
30B  0400000052          +               EQ     .140
31B  0400000052          +               EQ     .140
                                   *                     LINE 13
32B                                .110  BSS    0
32B  6102000015                          SB0    B2+0+15B
           5110002061+                   SA1    C
33B  5120002057          +               SA2    A
           35021                         RX0    X2-X1
                24000                    NX0    X0
34B  0330000052          +               MI     X0,.140
                                   *                     LINE 14
35B                                .120  BSS    0
35B  6102000016                          SB0    B2+0+16B
           7170000001                    SX7    0+1
36B  5170002063          +               SA7    J
           7110000001                    SX1    0+1
37B  7120000012                          SX2    0+12B
           37021                         IX0    X2-X1
                10600                    BX6    X0
40B  5160002064          +               SA6    DC.0
41B                                .DD.0 BSS    0
                                   *                     LINE 15
41B  6102000017                          SB0    B2+0+17B
           5110002057+                   SA1    A
42B  5120002060          +               SA2    B
           34221                         RX2    X2+X1
                24002                    NX0    X2
43B  5130002063          +               SA3    J
           27303                         PX3    X3
                24603                    NX6    X3
44B  44706                               FX7    X0/X6
        54710                            SA7    A1
                                   *                     LINE 16
45B                                .130  BSS    0
45B  6102000020                          SB0    B2+0+20B
           5110002063+                   SA1    J
46B  7120000001                          SX2    0+1
           36021                         IX0    X2+X1
                10700                    BX7    X0
47B  54710                               SA7    A1
        5120002064       +               SA2    DC.0
```


OBJECT LISTING.

```
                                        *                       LINE 17
52B                                     .140    BSS     0
52B  6102000021                                 SB0     B2+0+21B
                7170000001                      SX7     0+1
53B  5170002062          +                      SA7     I
                7110000001                      SX1     0+1
54B  7120001750                                 SX2     0+1750B
                37021                           IX0     X2-X1
                     10600                      BX6     X0

55B  5160002065          +                      SA6     DC.1
56B                                     *.DO.1  BSS     0          LINE 18
56B  6102000022                                 SB0     B2+0+22B
                5110002061+                     SA1     C
57B  5120002060          +                      SA2     B
                44021                           FX0     X2/X1

                     10100                      BX1     X0
60B  10711                                      BX7     X1
          5170000102     +                      SA7     ST.
61B  0100000000                                 RJ      =XSQRT.
62B  5130002062          +                      SA3     I
                27103                           PX1     X3
                     24701                      NX7     X1
63B  34267                                      RX2     X6+X7
          24002                                 NX0     X2
                10600                           BX6     X0
64B  5263000106          +                      SA6     X3+R-1
                                        *                       LINE 19
65B                                     .150    BSS     0
65B  6102000023                                 SB0     B2+0+23B
                5110002062+                     SA1     I
66B  7120000001                                 SX2     0+1
                36021                           IX0     X2+X1
                     10700                      BX7     X0
67B  54710                                      SA7     A1
          5120002065     +                      SA2     DC.1
70B  7212777776                                 SX1     X2+0-1
                10611                           BX6     X1
                     54620                      SA6     A2
71B  0321000056          +                      PL      X1,.DO.1
                                        *                       LINE 20
72B  6102000024                                 SB0     B2+0+24B
                5110000103+                     SA1     AP.1
73B  01000000000024000000                       RJT     =XSTOP5.,24B
                                        *                       LINE 21
74B  6102000025                                 SB0     B2+0+25B
75B                                             BSS     0
75B                                             USE     START.
 4B  6102775711                                 SB0     B2+0-2066B
                6102000000+                     SB0     B2+TRACE.
 5B                                     IFS     BSS     0
 5B  5110000002          +                      SA1     FILVEC.
 5B              0100000000                     RJ      =XQSRPV.
```



```
 6B                                             BSS     0
 6B                                             USE     TEMPS.
102B                                    ST.     BSS     1
103B                                    CT.     BSS     0
103B                                    IT.     BSS     0
103B                                    OT.     BSS     0
103B                                    VD.     BSS     0
103B                                    LC.     BSS     0
```

5 | 5 | 1
8 | 13 | 5

```
                                                    END    IF8

--STATISTICS--

  PROGRAM-UNIT LENGTH            2066B =  1078
  SCM STORAGE USED              62600B =  25984
  COMPILE TIME                  0.322 SECONDS
    0.327 CP SECONDS COMPILATION TIME.
```

Lesson 4

POST MORTEM DUMP

LESSON PREFACE:

OBJECTIVES:

REFERENCES:

PROJECTS:

168,420
/010/001

17

```
begin,,lfipr2x
0ARGUMENT NEGATIVE
 FTN - INFORMATIVE ERROR NUMBER 39
 TRACEBACK INITIATED BY SYSERR AT REL(AIS) ADDRESS 122(11101).
 CALLED BY SQRT AT ADDRESS 1(2627) WITH NO ARGLIST.
   INDEFINITE VALUE IN QUAD     NEAR LINE 17
     EXCHANGE PACKAGE.
```

MUST ALWAYS = 0

P	0	A0	124	B0	0	(A0)	0000	0000	0000	0100	0113
RA	660000	A1	2637	B1	1	(A1)	6047	3237	7777	7777	7777
FL	16100	A2	2640	B2	0	(A2)	6047	3237	7777	7777	7776
EM	7007	A3	0	B3	775060	(A3)	0004	0001	6600	0000	0000
RAE	0	A4	1	B4	6	(A4)	0000	0000	0000	0000	0000
FLE	0	A5	271	B5	12520	(A5)	1723	5000	0000	0000	0000
MA	2200	A6	10572	B6	12520	(A6)	0000	0000	0600	0000	0514
		A7	13110	B7	1	(A7)	0000	0000	0000	0000	0000

X0	4462	4462	0000	0000	0000
X1	6047	3237	7777	7777	7777
X2	1777	0000	0000	0000	0000
X3	0000	0000	0000	0000	0000
X4	0000	0000	0000	0000	0000
X5	1723	5000	0000	0000	0000
X6	1777	0000	0000	0000	0000
X7	1777	0000	0000	0000	0000

166g IS WHERE program BLOWS UP

(RA)	0004	0001	6600	0000	0000
(RA+1)	0000	0000	0000	0000	0000

^ CM DUMP FROM 126 TO 226.

126	61027	77547	61020	00111	51100	00124	01000	02552
130	61020	00011	61000	46000	01000	00373	00110	00111
132	51600	00320	61000	46000	61020	00012	13777	46000
134	51700	00321	61000	46000	61020	00013	71607	77765
136	10166	51600	00322	46000	71200	00001	71300	00025
140	37032	10700	51700	00323	61020	00014	71607	77765
142	10466	51600	00324	46000	71500	00001	71100	00025
144	37215	10722	51700	00325	61020	00015	71607	77765
146	51600	00326	71200	00001	71300	00025	37332	10733
150	51700	00327	61000	46000	61020	00016	51100	00322
152	27401	24604	51600	00330	61020	00017	51200	00324
154	27502	24705	51700	00331	61020	00020	51300	00326
156	27003	24600	51600	00332	61020	00021	14077	41477
160	51500	00330	41165	46000	51200	00246	41112	35141
162	24401	10144	10711	46000	51700	00270	10600	46000
164	51600	00271	01000	00355	51500	00271	34265	24702
166	51100	00247	44071	46000	51200	00330	41702	46000
170	51700	00333	61000	46000	61020	00022	51300	00331
172	14633	41033	51400	00332	41542	51400	00246	41554
174	35405	24004	10100	46000	51600	00272	10611	46000
176	51600	00273	01000	00355	51100	00272	35516	24605
200	51200	00247	44762	46000	51300	00330	41773	46000
202	51700	00334	61000	46000	61020	00023	51100	00275
204	01000	02346	00230	00111	61020	00024	71100	00001
206	51200	00321	36721	54720	61020	00025	51100	00326
210	71200	00001	36021	10700	54710	51200	00327	46000
212	72127	77776	10611	54620	03210	00151	61000	46000
214	61020	00026	51100	00324	71200	00001	36021	10700
216	54710	51200	00325	46000	72127	77776	10611	54620
220	03210	00145	61000	46000	61020	00027	51100	00322
222	71200	00001	36021	10700	54710	51300	00323	46000

FORTRAN LINE # 21g = 1

ZZZDUMP
REWIND, *
COPY, ZZZDUMP

```
copy,lfimap
S
1          LOAD MAP - QUAD                                    CYBER LOADER 1.5-528        81/10/16. 17.23.29.        PAGE     1

           FWA OF THE LOAD              111
           LWA+1 OF THE LOAD           10551

           TRANSFER ADDRESS -- QUAD                127

           PROGRAM ENTRY POINTS --        QUAD                127


           PROGRAM AND BLOCK ASSIGNMENTS.

           BLOCK        ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL   HARDWARE   COMMENTS

           QUAD             111      230     LGO      81/10/16 FTN      5.1 528     767X I        PROGRAMOPT=0
           SQRT.            341       32 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 COMPUTE THE SQUARE ROOT OF X. OPT=ALL.
           SECOND=          373       13 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 FCL5 - SECOND INTRINSIC FUNCTION
           /Q5.IO./         406      302
           /AP.IO./         710       17
           /FCL.C./         727       34
           /FCL=ENT/        763       73
           FORSYS=         1056     1253 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 FORTRAN OBJECT LIBRARY UTILITIES.
           OUTC=           2331      221 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 FORMATTED WRITE FORTRAN RECORD.
           Q5RPV=          2552       14 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 FCL5 - ABORT RECOVERY INITIALIZATION
R.         SYS=1ST         2566       70 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 MATH LIBRARY LINK TO ERROR MESSAGE PROCESS
           /STP.END/       2656        1
           Q5NTRY=         2657       32 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 FCL5 - INITIALIZE FCL5 RUN TIME LIBRARY.
           CHMOVE=         2711      266 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 CHARACTER MOVE AND CONCATENATE
           COMIO=          3177       17 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 COMMON CODED I/O ROUTINES AND CONSTANTS.
           FCL=FDL         3216       63 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 FCL CAPSULE LOADING
           FMTAP=          3301      572 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 CRACK APLIST AND FORMAT FOR KODER/KRAKER.
           FORUTL=         4073      165 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 FCL MISC. UTILITIES.
           GETFIT=         4260      164 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 LOCATE AN FIT GIVEN A FILE NAME.
           KODER=          4444      654 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 OUTPUT FORMAT INTERPRETER.
           EXP.MSG         5320       16 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 COMMON ERROR MESSAGES FOR EXPONENTIATION.
E.         SYSAID=         5336        1 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 LINK BETWEEN SYS=AID AND INITIALIZATION CO
           FECMSK=         5337       41 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 INITIALIZE CONSTANTS.
           FLTOUT=         5400      321 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 COMMON FLOATING OUTPUT CODE
           OUTCOM=         5721      211 SL-FTN5LIB  81/06/05 COMPASS 3.6 528                 COMMON OUTPUT CODE
           CPU.SYS         6132       40 SL-SYSLIB   81/06/08 COMPASS 3.6 498                 PROCESS SYSTEM REQUEST.
           RECOVR          6172      166 SL-SYSLIB   81/06/08 COMPASS 3.6 498                 REPRIEVE INTERFACE
           /FDL.COM/       6360       14
           FDL.MMI         6374      222 SL-SYSLIB   81/06/09 COMPASS 3.6 528                 FDL MEMORY MANAGER, INTERFACE.
           CMF.ALF         6616      162 SL-SYSLIB   81/08/18 COMPASS 3.6 528                 CMM V1.1 - ALLOCATE FIXED.
           CMF.FRF         7000       36 SL-SYSLIB   81/08/18 COMPASS 3.6 528                 CMM V1.1 - FREE FIXED.
           CMF.GSS         7036       22 SL-SYSLIB   81/08/18 COMPASS 3.6 528                 CMM V1.1 - GET SUMMARY STATISTICS.
           CTL$RM          7060      605 SL-SYSLIB   81/08/19 COMPASS 3.6 528                 CRM CONTROLLING ROUTINE.
           CTL$WR          7665       44 SL-SYSLIB   81/08/19 COMPASS 3.6 528                 CRM CONTROLLER - WEOX, REWIND
ROCE       CPUCPM          7731        5 SL-SYSLIB   81/06/17 COMPASS 3.6 498                 73/06/12. 81/06/17. CONTROL POINT MANAGER
           FDL.RES         7736      211 SL-SYSLIB   81/06/09 COMPASS 3.6 528                 FAST DYNAMIC LOADER RESIDENT.
           CMF.CSF        10147        6 SL-SYSLIB   81/08/18 COMPASS 3.6 528                 CMM V1.1 - CHANGE SPECS FIXED.
           CMM.FFA        10155       14 SL-SYSLIB   81/08/18 COMPASS 3.6 528                 CMM V1.1 - FIXED FREE ALGORITHM.
           CMM.R          10171      206 SL-SYSLIB   81/08/18 COMPASS 3.6 528                 CMM V1.1 - RESIDENT SUBROUTINES.
           CMF.SLF        10377       22 SL-SYSLIB   81/08/18 COMPASS 3.6 528                 CMM V1.1 - SHRINK AT LWA FIXED.
           ERR$RM         10421       25 SL-SYSLIB   81/08/19 COMPASS 3.6 528                 CRM ERROR PROCESSOR ENTRY.
           LIST$RM        10446       67 SL-SYSLIB   81/08/19 COMPASS 3.6 528                 CRM - ALLOCATE SPACE FOR LIST OF FILES
           RM$SYS=        10535        5 SL-SYSLIB   81/08/19 COMPASS 3.6 528                 CRM - POST RA+1 REQUEST
1          LOAD MAP - QUAD                                    CYBER LOADER 1.5-528        81/10/16. 17.23.29.        PAGE     2
```

```
      PROGRAM FINIS
      IMPLICIT INTEGER (A-Z)
      DIMENSION REC (5), ISFIT (35)
C
C BUILD  FILE INFORMATION TABLE
C
      CALL FILEIS (ISFIT, 'LFN', 'ISFILE', 'ORG', 'NEW',
     X 'RT', 'T', 'HL', 35, 'TL', 5, 'CP', 30, 'CL', 1,
     X 'MRL', 50,
     X 'DP', 80, 'IP', 20, 'MBL', 600,
     X 'KT', 'S', 'KL', 10, 'RKW', 2, 'RKP', 0, 'EMK', 'YES',
     X 'KA', REC (3), 'KP', 0, 'WSA', REC(1),
     X 'EFC', 3)
      CALL OPENM (ISFIT, 'NEW')
      CALL FITDMP (ISFIT)
C
C COPY SEQUENTIAL FILE TO INDEXED SEQUENTIAL
C
      OPEN (1, FILE = 'CZFILE')
      REWIND 1
      PRINT *, ' WRITE -IS- FILE'
  100 READ (1, 901, END=200) REC
      CALL PUT (ISFIT)
      PRINT 902, IFETCH (ISFIT(1), 'FP'), IFETCH (ISFIT(1), 'RL'), REC
      GO TO 100
  200 CALL FITDMP (ISFIT)
      PRINT *, ' READ -IS- FILE'
      CALL CLOSEM (ISFIT)
      CALL OPENM (ISFIT, 'INPUT')
  300 CALL GETN (ISFIT)
      IF (IFETCH (ISFIT(1), 'FP') .EQ. O"100") GO TO 400
      PRINT 902, IFETCH (ISFIT(1), 'FP'), IFETCH (ISFIT(1), 'RL'), REC
      GO TO 300
  400 CONTINUE
      CALL CLOSEM (ISFIT)
  901 FORMAT (5A10)
  902 FORMAT (' FP = ', O4, ' RL = ', I3, ' REC = ', 5A10)
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ISFIT      306B                              INTEGER        35
  REC        301B                              INTEGER         5


--PROCEDURES  (LO=A)
 -NAME----- TYPE -------ARGS------CLASS-----        -NAME------TYPE--------ARGS------CLASS-----

  CLOSEM                  1       SUBROUTINE        IFETCH    INTEGER        2      FUNCTION
  FILEIS                 41       SUBROUTINE        OPENM                    2      SUBROUTINE
  FITDMP                  1       SUBROUTINE        PUT                      1      SUBROUTINE
  GETN                    1       SUBROUTINE
```

```
          PROGRAM FINIS      76/176  OPT=0                          FTN 5.0+528        80/11/06. 07.43.12       PAGE     2

 -STATEMENT LABELS- (LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES - -DEF

    100      22B                    22          400      62B                    34
    200      35B                    26          901     146B     FORMAT        36
    300      45B                    30          902     150B     FORMAT        37

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  FINIS       5B      0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------

  TAPE1    AUX/FMT/SEQ

--STATISTICS--

  PROGRAM-UNIT LENGTH            351B =  233
  SCM STORAGE USED             60700B =  25024
  COMPILE TIME                 0.040 SECONDS
```

```
LOAD MAP - FTNIS                                    CYBER LOADER 1.5-528        80/11/06. 07.43.15.        PAGE      1


FWA OF THE LOAD              111
LWA+1 OF THE LOAD          15461

TRANSFER ADDRESS -- FTNIS              116

PROGRAM ENTRY POINTS --        FTNIS              116


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

FTNIS            111      351    LGO       80/11/06 FTN     5.0 528    767X I        PROGRAMOPT=0
/Q5.IO./         462      300
/AP.IO./         762       17
/FCL.C./        1001       34
/FCL=ENT/       1035       73
FORSYS=         1130     1256 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FORTRAN OBJECT LIBRARY UTILITIES.
INPC=           2406      260 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FORMATTED READ FORTRAN RECORD.
OPEN=           2666       11 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL5 - EXECUTE FORTRAN OPEN STATEMENT.
OUTC=           2677      221 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FORMATTED WRITE FORTRAN RECORD.
OUTF=           3120      230 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  LIST DIRECTED OUTPUT CONTROL
Q5RPV=          3350       14 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL5 - ABORT RECOVERY INITIALIZATION
REWIND=         3364       55 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  POSITION FILE AT BEGINNING-OF-INFORMATION.
/STP.END/       3441        1
Q5NTRY=         3442       32 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL5 - INITIALIZE FCL5 RUN TIME LIBRARY.
CHMOVE=         3474      254 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  CHARACTER MOVE AND CONCATENATE
COMIO=          3750       17 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  COMMON CODED I/O ROUTINES AND CONSTANTS.
ECA=            3767      122 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL5 - ENCODE CHARACTER ARGUMENT.
FCL=FDL         4111       63 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL CAPSULE LOADING
FECMSK=         4174       41 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  INITIALIZE CONSTANTS.
FMTAP=          4235      572 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  CRACK APLIST AND FORMAT FOR KODER/KRAKER.
FORUTL=         5027      165 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL MISC. UTILITIES.
GETFIT=         5214      164 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  LOCATE AN FIT GIVEN A FILE NAME.
INCOM=          5400      174 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  COMMON INPUT FORMATTING CODE
KODER=          5574      654 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  OUTPUT FORMAT INTERPRETER.
KRAKER=         6450      566 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  PROCESS FORMATTED FORTRAN INPUT.
LDOUT=          7236      332 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  LIST DIRECTED OUTPUT FORMATTING
OUTCOM=         7570      211 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  COMMON OUTPUT CODE
SYSAID=        10001        1 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
CMC=           10002       16 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL5 - COMPARE MULTIPLE CHARACTER.
FLTIN=         10020      156 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  COMMON FLOATING INPUT CONVERTER.
FLTOUT=        10176      321 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  COMMON FLOATING OUTPUT CODE
BCT=           10517       12 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL5 - BURST COLLATE TABLE.
CCM=           10531       61 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL5 - COLLATED COMPARE MANAGER.
ICC=           10612       21 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL5 - INITIALIZE CHARACTER COMPARE.
DCC=           10633      176 SL-FTN5LIB  80/09/26 COMPASS 3.6 530                  FCL5 - DYNAMIC COLLATED COMPARE.
CPU.CPM        11031        5 SL-SYSLIB   80/02/19 COMPASS 3.6 518                  79/05/10. 80/02/18. CONTROL POINT MANAGER PROCE
CPU MVE        11036       64 SL-SYSLIB   80/09/02 COMPASS 3.6 528                  MOVE BLOCK OF DATA.
CPU.SYS        11122       40 SL-SYSLIB   80/09/02 COMPASS 3.6 528                  PROCESS SYSTEM REQUEST.
CMF ALF        11162      162 SL-SYSLIB   80/10/01 COMPASS 3.6 528                  CMM V1.1 - ALLOCATE FIXED.
CMM CIA        11344      105 SL-SYSLIB   80/10/01 COMPASS 3.6 528                  CMM V1.1 - CHANGE INTERNAL AREA.
CMF CSF        11451        6 SL-SYSLIB   80/10/01 COMPASS 3.6 528                  CMM V1.1 - CHANGE SPECS FIXED.
CMF DOE        11457       17 SL-SYSLIB   80/10/01 COMPASS 3.6 528                  CMM V1.1 - DELETE OVERFLOW ENTRY.
CMM FFA        11476       14 SL-SYSLIB   80/10/01 COMPASS 3.6 528                  CMM V1.1 - FIXED FREE ALGORITHM
```

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCSSR	VER	LEVEL	HARDWARE	COMMENTS
CMF.FRF	11512	36	SL-SYSLIB	80/10/01	COMPASS	3	6 528		CMM V1.1 - FREE FIXED.
CMM.GOA	11550	32	SL-SYSLIB	80/10/01	COMPASS	3.6	528		CMM V1.1 - GENERAL OVERFLOW ACTION.
CMF.GOS	11602	73	SL-SYSLIB	80/10/01	COMPASS	3.6	528		CMM V1.1 - GET OVERFLOW STATISTICS.
CMF.GSS	11675	22	SL-SYSLIB	80/10/01	COMPASS	3.6	528		CMM V1.1 - GET SUMMARY STATISTICS.
CMM.MEM	11717	7	SL-SYSLIB	80/10/01	COMPASS	3.6	528		
CMM.POA	11726	130	SL-SYSLIB	80/10/01	COMPASS	3.6	528		CMM V1.1 - PROCESS OVERFLOW ACTION.
CMF.POE	12056	22	SL-SYSLIB	80/10/01	COMPASS	3.6	528		CMM V1.1 - PUSH OVERFLOW ENTRY.
CMM.R	12100	206	SL-SYSLIB	80/10/01	COMPASS	3.6	528		CMM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF	12306	22	SL-SYSLIB	80/10/01	COMPASS	3.6	528		CMM V1.1 - SHRINK AT LWA FIXED.
CTLSRM	12330	435	SL-SYSLIB	80/10/02	COMPASS	3.6	528		CRM CONTROLLING ROUTINE.
CTLSWR	12765	44	SL-SYSLIB	80/10/02	COMPASS	3.6	528		CRM CONTROLLER - WEOX, REWIND
CTRLSAA	13031	141	SL-SYSLIB	80/10/02	COMPASS	3.6	528		CRM CONTROLLER - ADVANCED ACCESS.
ERRSRM	13172	25	SL-SYSLIB	80/10/02	COMPASS	3.6	528		CRM ERROR PROCESSOR ENTRY.
LIST$RM	13217	67	SL-SYSLIB	80/10/02	COMPASS	3.6	528		CRM - ALLOCATE SPACE FOR LIST OF FILES
RMSSYS*	13306	5	SL-SYSLIB	80/10/02	COMPASS	3.6	528		CRM - POST RA+1 REQUEST
RMSX	13313	2	SL-SYSLIB	80/10/02	COMPASS	3.6	528		
FILE$AA	13315	16	SL-SYSLIB	80/10/02	COMPASS	3.6	528		CRM FTN INTERFACE (FILEIS,FILEDA,FILEAK)
/FDL.COM/	13333	14							
FDL.RES	13347	211	SL-SYSLIB	80/10/02	COMPASS	3.6	528		FAST DYNAMIC LOADER RESIDENT.
FDL.MMI	13560	222	SL-SYSLIB	80/10/02	COMPASS	3.6	528		FDL MEMORY MANAGER INTERFACE.
RECOVR	14002	166	SL-SYSLIB	80/10/06	COMPASS	3.6	528		REPRIEVE INTERFACE
FIT$COM	14170	543	SL-BAMLIB	80/10/02	COMPASS	3.6	528		CRM FTN INTERFACE (IFETCH,STOREF,SETFIT)
ER$PROC	14733	75	SL-BAMLIB	80/10/02	COMPASS	3.6	528		CRM FTN INTERFACE ERROR PROCESSOR
OPN$CLS	15030	66	SL-BAMLIB	80/10/02	COMPASS	3.6	528		CRM FTN INTERFACE (CLOSEM,OPENM)
PUT	15116	17	SL-BAMLIB	80/10/02	COMPASS	3.6	528		CRM FTN INTERFACE (PUT)
GETN	15135	14	SL-BAMLIB	80/10/02	COMPASS	3.6	528		CRM FTN INTERFACE (GETN)
PLST$RM	15151	106	SL-BAMLIB	80/10/02	COMPASS	3.6	528		
RMSILD	15257	7	SL-BAMLIB	80/10/02	COMPASS	3.6	528		CRM FTN INTERFACE (SETUP COMMON PARAMETERS)
RMSSTUF	15266	122	SL-BAMLIB	80/10/02	COMPASS	3.6	528		CRM FTN INTERFACE (SETUP PARAMETER REGISTERS)
FITDMP	15410	24	SL-BAMLIB	80/10/02	COMPASS	3.6	528		CRM FTN INTERFACE (FITDMP)
CTL$LBL	15434	25	SL-SYSLIB	80/10/02	COMPASS	3.6	528		CRM CONTROLLER - LABEL PROCESSING.

.134 CP SECONDS. 34300B CM STORAGE USED 13 TABLE MOVES

```
 WRITE -IS- FILE
FP = 0020 RL =  40 REC =        MAMMAL    LION      1    LAND
FP = 0020 RL =  50 REC =        BIRD      DUCK      3    AIR  LAND WATER
FP = 0020 RL =  45 REC =        MAMMAL    SEAL      2    LAND WATER
FP = 0020 RL =  40 REC =        FISH      SHARK     1    WATER
FP = 0020 RL =  40 REC =        MAMMAL    WHALE     1    WATER
FP = 0020 RL =  45 REC =        BIRD      PENGUIN   2    LAND WATER
 READ -IS- FILE
FP = 0020 RL =  50 REC =        BIRD      DUCK      3    AIR  LAND WATER
FP = 0020 RL =  40 REC =        MAMMAL    LION      1    LAND LAND WATER
FP = 0020 RL =  45 REC =        BIRD      PENGUIN   2    LAND WATER
FP = 0020 RL =  45 REC =        MAMMAL    SEAL      2    LAND WATER
FP = 0020 RL =  40 REC =        FISH      SHARK     1    WATERWATER
FP = 0020 RL =  40 REC =        MAMMAL    WHALE     1    WATERWATER
```

```
STATISTICS FOR FILE ISFILE
ORGANIZATION------- IS
CREATION DATE------ 80/11/06.
DATE OF LAST CLOSE- 80/11/06.
TIME OF LAST CLOSE- 07.43.18.

FILE IS NOT MIPPED
COLLATION IS STANDARD

PRIMARY KEY INFORMATION
  STARTING WORD POSITION ------ 2
  STARTING CHARACTER POSITION - 0
  TYPE -- COLLATED SYMBOLIC
  LENGTH IN CHARACTERS -------- 10

MAXIMUM RECORD SIZE  50
MINIMUM RECORD SIZE  35

TOTAL TRANSACTIONS
  NUMBER OF PUTS ------ 6
  NUMBER OF GETS ------ 0
  NUMBER OF DELETES --- 0
  NUMBER OF REPLACES -- 0
  NUMBER OF GETNEXTS -- 6

CIO CALLS FOR FILE
  NUMBER OF READS ----- 3
  NUMBER OF WRITES ---- 4
  NUMBER OF RECALLS --- 0
  NUMBER OF REWRITES -- 2

NUMBER OF BLOCKS------- 3
NUMBER OF EMPTY BLOCKS- 0
BLOCK SIZE IN PRUS----- 2
NUMBER OF DATA RECORDS- 6

FILE LENGTH IN PRUS  8
NUMBER OF INDEX LEVELS IN USE   1
```

```
        CRMEP.LO.RU.                                                                                    80/11/06.        07.43.19.
RM NOTE  1000 ON LFN ISFILE   FILE OPENED
RM NOTE  1000 ON LFN ISFILE   FIT DUMP            (FIT AT 000417)
                   0 ASCII                   0 EO                  00000000 LBL                   0 PM
                   0 BAL              0000000001 EOIWA                   0 LCR            00000000 PNO
                   0 BBH                     000 ERL          11230611140500 LFN              00 POS
                   0 BCK                     000 ES                 000000 LGX            00000000 PTL
                   0 BFF                  000000 EX                     01 LL                0002 RB
              000000 BFS                       1 EXD                    00 LNG          0000000000 RC
          0000000000 BN                        0 FF                     01 LOP                   0 RDR
                   0 BT                 00000062 FL                     01 LOP5                  0 REL
              000417 BZF             7777777777 FLM                 00000036 LP                 00 RKP
                   0 B8F                       0 FNF                     2 LT                0002 RKW
              000000 CDT                       3 FO                     00 LVL           00000000 RL
                   0 CF                      020 FP                 000000 LX                  01 RMK
                  01 CL                        0 FPB               00002354 MBL                 05 RT
                   0 CM                       36 FTS            000000000000 MFN                 0 SB
                   1 CMPLT                 033462 FWB                 000 MKL                  0 SBF
                   0 CNF                        0 FWI            00000000 MNB                  0 SDS
            00000036 CP                         0 HB             00000043 MNR                 00 SES
              000000 CPA                 00000043 HL             00000062 MRL                  0 SOL
                   0 C1                  00000000 HMB                 00 MUL                  0 SPR
                  00 DC                    120000 HRL                  0 NDX           00000005 TL
              000000 DCA                 00000000 IBL                 03 NL                  00 TRC
              000000 DCT                      024 IP                   0 NOFCP                0 ULP
                   0 DFC                      000 IRS                  1 OC                   0 VF
                   0 DFLG                  000414 KA                   0 OF                  03 VNO
                   0 DKI                      012 KL                   1 ON            0000000000 WA
                 120 DP                         0 KNE                  1 ORG                  0 WPN
                0411 DVT                       00 KP                   0 OVF           00000412 WSA
              000000 DX                      0000 KR                  00 PC              000000 XBS
                 000 ECT                         1 KT                  2 PD       00000000000000 XN
                   3 EFC                   000000 LA                   0 PEF
                   1 EMK                       00 LAC             000000 PKA

RM NOTE  1000 ON LFN ISFILE   FIT DUMP            (FIT AT 000417)
                   0 ASCII                   0 EO                  00000000 LBL                   0 PM
                   0 BAL              0000000001 EOIWA                   0 LCR            00000000 PNO
                   0 BBH                     000 ERL          11230611140500 LFN              00 POS
                   0 BCK                     000 ES                 000000 LGX            00000000 PTL
                   0 BFF                  000000 EX                     01 LL                0002 RB
              000000 BFS                       1 EXD                    00 LNG          0000000000 RC
          0000000000 BN                        0 FF                     01 LOP                   0 RDR
                   0 BT                 00000062 FL                     01 LOP5                  0 REL
              000417 BZF             7777777777 FLM                 00000036 LP                 00 RKP
                   0 B8F                       0 FNF                     2 LT                0002 RKW
              000000 CDT                       3 FO                     00 LVL           00000055 RL
                   0 CF                      020 FP                 000000 LX                  01 RMK
                  01 CL                        0 FPB               00002354 MBL                 05 RT
                   0 CM                       36 FTS            000000000000 MFN                 0 SB
                   1 CMPLT                 032054 FWB                 000 MKL                  0 SBF
                   0 CNF                        0 FWI            00000000 MNB                  0 SDS
            00000036 CP                         0 HB             00000043 MNR                 00 SES
              000000 CPA                 00000043 HL             00000062 MRL                  0 SOL
                   0 C1                  00000000 HMB                 00 MUL                  0 SPR
                  00 DC                    120000 HRL                  0 NDX           00000005 TL
              000000 DCA                 00000000 IBL                 03 NL                  00 TRC
              000000 DCT                      024 IP                   0 NOFCP                0 ULP
                   0 DFC                      000 IRS                  1 OC                   0 VF
                   0 DFLG                  000414 KA                   0 OF                  03 VNO
                   0 DKI                      012 KL                   1 ON            0000000000 WA
                                                                                  ......... FTN -IS- - PAGE    6
```

```
                 120 DP                    0 KNE                    1 ORG                    0 WPN
                0411 DVT                  00 KP                     0 OVF             00000412 WSA
              000000 DX                 0000 KR                    00 PC               000000 XBS
                 000 ECT                   1 KT                     2 PD     00000000000000 XN
                   3 EFC              000000 LA                     0 PEF
                   1 EMK                  00 LAC                 000000 PKA

RM NOTE  1002 ON LFN ISFILE   FILE CLOSED
RM NOTE  1003 ON LFN ISFILE   NUMBER OF INDEX LEVELS          1
RM NOTE  1004 ON LFN ISFILE   ***NUMBER OF GETS THIS OPEN                0
RM NOTE  1005 ON LFN ISFILE   ***NUMBER OF PUTS THIS OPEN                6
RM NOTE  1006 ON LFN ISFILE   ***NUMBER OF REPLACES THIS OPEN            0
RM NOTE  1007 ON LFN ISFILE   ***NUMBER OF DELETES THIS OPEN             0
RM NOTE  1033 ON LFN ISFILE   ***NUMBER OF GET NEXTS THIS OPEN           0
RM NOTE  1010 ON LFN ISFILE       ***TOTAL DISKAREA***            512  WORDS
RM NOTE  1001 ON LFN ISFILE   FILE OPENED
RM NOTE  1011 ON LFN ISFILE   GETN REACHED EOI
RM NOTE  1002 ON LFN ISFILE   FILE CLOSED
RM NOTE  1003 ON LFN ISFILE   NUMBER OF INDEX LEVELS          1
RM NOTE  1004 ON LFN ISFILE   ***NUMBER OF GETS THIS OPEN                0
RM NOTE  1005 ON LFN ISFILE   ***NUMBER OF PUTS THIS OPEN                0
RM NOTE  1006 ON LFN ISFILE   ***NUMBER OF REPLACES THIS OPEN            0
RM NOTE  1007 ON LFN ISFILE   ***NUMBER OF DELETES THIS OPEN             0
RM NOTE  1033 ON LFN ISFILE   ***NUMBER OF GET NEXTS THIS OPEN           6
RM NOTE  1010 ON LFN ISFILE       ***TOTAL DISKAREA***            512  WORDS
```

```
.PROC,FTNIS,INP=*DATA.
ASSIGN,MS,OUTPUT.
COMMENT. THIS HANDOUT PREPARED BY -
COMMENT.    JOHN S. FOX
COMMENT.    PRODUCT SET FIELD SUPPORT
COMMENT.    CONTROL DATA CORP.
COMMENT.    SUNNYVALE, CALIFORNIA
COMMENT.BANNER.    1H
NOTE,OUTPUT,NR./XEROX1     ......... FTN -IS- -/
GET,CZFILE.
FTN5,I=INP,REW.
PURGE,ISFILE/NA.
RETURN,ISFILE.
DEFINE,ISFILE.
LDSET,MAP=SB.
LGO.
FLSTAT,ISFILE.
CRMEP,LO,RU.
REWIND,FTNIS.
COPYSBF,FTNIS,OUTPUT.
DAYFILE,FR=ASSIGN.
.DATA
      PROGRAM FTNIS
      IMPLICIT INTEGER (A-Z)
      DIMENSION REC (5), ISFIT (35)
C
C BUILD  FILE INFORMATION TABLE
C
      CALL FILEIS (ISFIT, 'LFN', 'ISFILE', 'ORG', 'NEW',
     X 'RT', 'T', 'HL', 35, 'TL', 5, 'CP', 30, 'CL', 1,
     X 'MRL', 50,
     X 'DP', 80, 'IP', 20, 'MBL', 600,
     X 'KT', 'S', 'KL', 10, 'RKW', 2, 'RKP', 0, 'EMK', 'YES',
     X 'KA', REC (3), 'KP', 0, 'WSA', REC(1),
     X 'EFC', 3)
      CALL OPENM (ISFIT, 'NEW')
      CALL FITDMP (ISFIT)
C
C COPY SEQUENTIAL FILE TO INDEXED SEQUENTIAL
C
      OPEN (1, FILE = 'CZFILE')
      REWIND 1
      PRINT *, ' WRITE -IS- FILE'
  100 READ (1, 901, END=200) REC
      CALL PUT (ISFIT)
      PRINT 902, IFETCH (ISFIT(1), 'FP'), IFETCH (ISFIT(1), 'RL'), REC
      GO TO 100
  200 CALL FITDMP (ISFIT)
      PRINT *, ' READ -IS- FILE'
      CALL CLOSEM (ISFIT)
      CALL OPENM (ISFIT, 'INPUT')
  300 CALL GETN (ISFIT)
      IF (IFETCH (ISFIT(1), 'FP') .EQ. O"100") GO TO 400
      PRINT 902, IFETCH (ISFIT(1), 'FP'), IFETCH (ISFIT(1), 'RL'), REC
      GO TO 300
  400 CONTINUE
      CALL CLOSEM (ISFIT)
  901 FORMAT (5A10)
  902 FORMAT (' FP = ', O4, ' RL = ', I3, ' REC = ', 5A10)
      END
```

```
PARTIAL  DAYFILE.  80/11/06. 07.43.20.*07.43.10* PAGE     1

07.43.10.ASSIGN,MS,OUTPUT.
07.43.10.MS. ASSIGNED TO OUTPUT.
07.43.11.COMMENT. THIS HANDOUT PREPARED BY -
07.43.11.COMMENT.    JOHN S. FOX
07.43.11.COMMENT.    PRODUCT SET FIELD SUPPORT
07.43.11.COMMENT.    CONTROL DATA CORP.
07.43.11.COMMENT.    SUNNYVALE, CALIFORNIA
07.43.11.COMMENT.BANNER.    1H
07.43.11.NOTE,OUTPUT,NR./XEROX1     ......... FTN -IS- -/
07.43.11.GET,CZFILE.
07.43.12.FTN5,I=ZZCCLAB,REW.
07.43.12.      60700 SCM STORAGE USED.
07.43.12.      0.042 CP SECONDS COMPILATION TIME.
07.43.12.PURGE,ISFILE/NA.
07.43.13.RETURN,ISFILE.
07.43.13.DEFINE,ISFILE.
07.43.13.LDSET,MAP=SB.
07.43.13.LGO.
07.43.18.      END  FTNIS
07.43.18.      52600 MAXIMUM EXECUTION FL.
07.43.18.       .062 CP SECONDS EXECUTION TIME.
07.43.18.FLSTAT,ISFILE.
07.43.19.CRMEP,LO,RU.
07.43.20.REWIND,FTNIS.
07.43.20.COPYSBF,FTNIS,OUTPUT.
07.43.20. EOI ENCOUNTERED.
07.43.20.DAYFILE,FR=ASSIGN.
```

```
 xedit,list
 XEDIT 3.1.00
?? p50
1 FTN 5.1+587          84/08/28. 16.47.09 PAGE      1
     PROGRAM BINSCH    74/176  OPT=0,ROUND= A/ S/ M/-D,-DS
          DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER
          /-FIXED,DB=-TB/-SB/-SL/ ER/-ID/-PMD/-ST,PL=5000
          FTN5,I=BINSCH2,L=LIST,PW=85,OPT=0.


        1          PROGRAM BINSCH(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT,TAPE10)
        2          CHARACTER NAMDAT*9,NAME*29,STARS*10,SSN*9
        3          CHARACTER NTAB(100)*29,NTABB*2900
        4          EQUIVALENCE (NTAB,NTABB)
        5          DATA NAMDAT /' NAME IS '/
        6          DATA STARS /'**********'/
        7          DATA I /1/
        8 100      READ(10,1000,END=110) NTAB(I)
        9          I=I+1
       10          IF(I.GT.100) GO TO 180
       11          GO TO 100
       12 110      IF(I.EQ.1)GO TO 170
       13          I=I+1
       14          DO 120 J=I,100
       15 120      NTAB(J)(1:10)=STARS
       16 130      READ(5,1003,END=150)SSN
       17          I=INDEX(NTABB,SSN)
       18          IF(I.EQ.0)GO TO 160
       19          I=((I-1)/29)+1
       20          NAME=NAMDAT//NTAB(I)(11:29)
       21 140      WRITE(6,1000)NAME
       22          GO TO 130
       23 150      CALL FITDMP(TAPE10,'TAPE10')
       24          CALL XXX
       25          STOP 'FIT DUMP TAKEN'
       26 160      NAME=STARS//STARS
       27          GO TO 140
       28 170      WRITE(6,1001)
       29          STOP
       30 180      WRITE(5,1002)
       31          STOP
       32 1000     FORMAT(A29)
       33 1001     FORMAT('NO DATA ON TAPE10')
       34 1002     FORMAT('OVER 100 RECORDS ON TAPE10')
       35 1003     FORMAT(A9)
       36          END


 --VARIABLE MAP--(LO=A)
  -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE

   I          710B                                INTEGER
   J          711B                                INTEGER
??
```

```
  NAMDAT      240B                                CHAR*9
  NAME        241B                                CHAR*29
  NTAB        246B            EQV                 CHAR*29        100
  NTABB       246B            EQV                 CHAR*2900
  SSN         245B                                CHAR*9
  STARS       244B                                CHAR*10
  TAPE10      713B            *S*                 REAL
1 FTN 5.1+587         84/08/28. 16.47.09 PAGE      2
     PROGRAM BINSCH   74/176  OPT=0,ROUND= A/ S/ M/-D,-DS

 --PROCEDURES--(LO=A)
  -NAME------TYPE--------ARGS------CLASS-----

  FITDMP                     2      SUBROUTINE
  INDEX      INTEGER         2      INTRINSIC
  XXX                        0      SUBROUTINE


 --STATEMENT LABELS--(LO=A)
  -LABEL-ADDRESS-----PROPERTIES----DEF

     100      23B                      8
     110      37B                     12
     120   INACTIVE   DO-TERM         15
     130      60B                     16
     140      77B                     21
     150     102B                     23
     160     110B                     26
     170     113B                     28
     180     117B                     30
    1000     137B     FORMAT          32
    1001     141B     FORMAT          33
    1002     145B     FORMAT          34
    1003     152B     FORMAT          35


 --ENTRY POINTS--(LO=A)
  -NAME---ADDRESS--ARGS---

  BINSCH      22B      0


 --I/O UNITS--(LO=A)
  -NAME--- PROPERTIES-------------

  TAPE10  FMT/SEQ
  TAPE5   FMT/SEQ
  TAPE6   FMT/SEQ


??
```

```
   P55

 --STATISTICS--

   PROGRAM-UNIT LENGTH                714B =  460
   SCM STORAGE USED                 62600B =  25984
   COMPILE TIME                     0.428 SECONDS
  END OF FILE
?? end
LIST     IS A LOCAL FILE
AEB ,       0.485UNTS.
/
```

```
 xedit,mlist
 XEDIT 3.1.00
?? p55
S
1          LOAD MAP - BINSCH                                        CYBER LOADER 1.5


           FWA OF THE LOAD              111
           LWA+1 OF THE LOAD          14445

           TRANSFER ADDRESS -- BINSCH            133

           PROGRAM ENTRY POINTS --        BINSCH            133


      ********* ERROR SUMMARY
           NE4100///  UNSATISFIED EXTERNAL REF -- XXX


           PROGRAM AND BLOCK ASSIGNMENTS.

           BLOCK        ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL   HA

           BINSCH           111      714     LGO      84/08/28 FTN      5.1 587    .76
           SYSAID=         1025        1 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
DE.
           /FCL.C./        1026       36
           /STP.END/       1064        1
           /Q5.IO./        1065      322
           Q5NTRY=         1407       31 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           CHMOVE=         1440      274 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           /FCL=ENT/       1734       75
           COMIO=          2031       17 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           DCC=            2050      177 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FAR=            2247       30 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FAS=            2277       16 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FCD=            2315       22 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FCL=FDL         2337       63 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FECMSK=         2422       41 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FEIFST=         2463        3 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FLTIN=          2466      156 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FLTOUT=         2644      327 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           /AP.IO./        3173       17
           FMTAP=          3212      676 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FORSYS=         4110     1564 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           FORUTL=         5674      200 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           GETFIT=         6074      200 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
PTOR
           INCOM=          6274      173 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           INDEX=          6467      111 SL-FTN5LIB  84/07/19 COMPASS 3.6 587

           INPC=           6600      261 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           KODER=          7061      710 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           KRAKER=         7771      576 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
DA3015     OUTC=          10567      230 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           OUTCOM=        11017      211 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
           Q5RPV=         11230       14 SL-FTN5LIB  84/07/19 COMPASS 3.6 587
 ??
```

```
   l/lof$rm/
            LOF$RM        12544    CTL$RM
?? n-10

?? p21

            ENTRY       ADDRESS    PROGRAM

            FDL.ULC       12160
            FDL=PCU       12165
            FDL.ROS       12202
            FDL.CGD       12205
            FDL.UGD       12232
            FDL=GGD       12361
            FDL=SGD       12533
  ---------> LOF$RM       12544    CTL$RM
            RM$PLL        12545
            RM$PL         12551
            RMP$FE        12565
            RM$PLB        12632
            CMMX.AA       12636
            OFCT$RM       12645
            RM$CIO        12647
            RM$RCLA       12661
            CHWR$RM       12667
            MOVE$RM       12675
?? end
MLIST    IS A LOCAL FILE
AEB ,       1.015UNTS.
/
```

```
xedit,zzzdump
XEDIT 3.1.00
?? p55
    EXCHANGE PACKAGE.

    P          0  A0     130  B0       0   (A0) 0000 0000 0000 0100 0113
    RA    332100  A1     311  B1       1   (A1) 0000 0000 0000 0000 0000
    FL     25000  A2   22153  B2   21761   (A2) 0000 0000 0001 0002 2147
    EM      7007  A3   11424  B3   11422   (A3) 4100 0201 0000 4300 1024
    RAE        0  A4   22150  B4   22147   (A4) 0000 0401 0000 0002 1707
    FLE        0  A5    1041  B5   22222   (A5) 0200 0000 0000 0000 0000
    MA      1000  A6   12645  B6   11403   (A6) 0000 0000 0000 0000 0000
                  A7    1041  B7       2   (A7) 0200 0000 0000 0000 0000

    X0    0000 0000 0000 0001 1422
    X1    0000 0000 0000 0000 0000
    X2    0000 0000 0000 0000 0000
    X3    4100 0201 0000 4300 1024
    X4    0000 0401 0000 0002 1707
    X5    0200 0000 0000 0000 0000
    X6    0000 0000 0000 0000 0003
    X7    0200 0000 0000 0000 0000

  (RA)    0001 4002 1700 0000 0000
  (RA+1)  0000 0000 0000 0000 0000


↑ CM DUMP FROM     157 TO     257.

    156       51700 01022 72167 77776       10611 51600 01023 46000
    160       03310 00171 61000 46000       61020 00017 51100 00266
    162       01000 02303 00170 00111       51600 00272 51100 00272
    164       01000 01446 00170 00111       51100 01022 71200 00001
    166       36021 10700 54710 46000       51200 01023 72127 77776
    170       10611 54620 03210 00161       61020 00020 51100 00326
    172       01000 06614 00200 00111       61020 00021 51100 00275
    174       01000 06472 00210 00111       51600 01021 61000 46000
    176       61020 00022 13111 37061       03000 00221 61000 46000
    200       61020 00023 71200 00001       37062 71300 00035 27103
    202       24401 27500 44054 26160       22761 36772 51700 01021
    204       61020 00024 51100 00277       01000 02303 00240 00111
    206       51600 00305 51100 00303       01000 01446 00240 00111
    210       61020 00025 51100 00334       01000 10604 00250 00111
    212       61020 00026 04000 00171       61020 00027 51100 00307
    214       01000 11404 00270 00111       61020 00030 51100 00311
    216       01004 00216 00300 00111       61020 00031 51100 00312
    220       01000 04420 00310 00111       61020 00032 51100 00313
    222       01000 01446 00320 00111       61020 00033 04000 00210
    224       61020 00034 51100 00340       01000 10604 00340 00111
    226       61020 00035 51100 00317       01000 04420 00350 00111
    230       61020 00036 51100 00343       01000 10604 00360 00111
    232       61020 00037 51100 00317       01000 04420 00370 00111
    234       61020 00044 61000 46000       00000 00000 00000 00012
    236       00000 00000 00000 00001       00000 00000 00000 00005
    240       00000 00000 00000 00013       00000 00000 00000 00035
    242       00000 00000 00000 00006       24012 00534 33555 55555
    244       06112 45504 25152 05524       01130 51655 55555 55555
    246       00000 00000 00000 00000       34333 33355 00000 00000
??
```

```
   250      51013 54452 55555 55555      34333 33455 00000 00000
   252      51344 21016 17550 40124      01551 71655 24012 00534
   254      33525 55555 55555 55555      34333 33555 00000 00000
   256      51354 11017 26052 25534      33335 52205 03172 20423
--EOR--
  CM DUMP FROM    1020 TO    1045.

  1020      00000 00000 00000 00000      00000 00000 00000 00000
  1022      00000 00000 00000 00145      77777 77777 77777 77776
  1024      00000 00000 00000 00000      01000 05051 00000 00000
  1026      55555 55555 55555 55555      40404 04040 40404 04040
  1030      11162 02524 00000 00000      17252 42025 24000 00000
  1032      17171 27432 14774 13155      20001 20726 42717 30565
  1034      00000 63146 31463 14632      00000 00000 00000 00003
  1036      00000 00000 00000 00000      00000 00000 00000 00000
  1040      00000 00000 00000 00113      02000 00000 00000 00000
  1042      00000 00000 00000 11610      00000 00000 00000 00000
  1044      00000 00000 00000 00000      00000 00000 00000 00000
--EOR--
  CM DUMP FROM   12540 TO   12550.

 12540      13341 11373 03130 12537      53346 54441 04000 12533
 12542      46551 40404 55052 22255      46551 40421 55052 22255
 12544      00000 00000 00000 14342      00000 00000 00000 12546
 12546      00000 00000 00000 12551      00000 00000 00000 12632
 12550      00000 00000 00000 00000      00210 06000 00000 00000
--EOR--
  CM DUMP FROM   14340 TO   14360.

 14340      63323 43600 04000 14323      00000 00000 00000 00000
 14342      00000 00000 00000 00065      77773 23232 17250 21150
 14344      77772 02524 00000 14447      77772 42025 24000 16404
 14346      77772 00534 33000 16456      77773 23232 05070 22147
 14350      00000 00000 00000 00000      34444 33456 34444 33557
 14352      00000 00000 00000 00000      00000 00000 00000 00000
          DUPLICATED LINES.
--EOR--
  CM DUMP FROM   14440 TO   14520.

 14440      12676 53620 71601 50515      27606 20652 71100 14444
 14442      12661 01000 11616 46000      53120 21136 04000 14436
 14444      00000 25000 00000 00001      60000 00000 00000 25000
 14446      00000 00001 44450 14520      11162 02524 00000 00031
 14450      24240 56000 00360 14510      00000 00000 00000 14510
 14452      00000 00000 00000 14510      64300 00000 01000 14512
 14454      00000 00000 00000 00000      00000 14457 00000 00000
 14456      00000 00000 00000 00000      00000 00000 00000 00001
 14460      00000 00000 00000 00000      00000 00044 00600 00000
 14462      00000 00065 03200 00000      00000 22600 00000 07017
 14464      14000 00000 00000 05030      00000 00000 00000 00000
 14466      00000 00006 00000 01065      00000 00000 00000 00000
 14470      03000 00000 00020 02003      00000 00000 00000 00000
 14472      00000 00000 00000 00000      00000 00000 00000 00000
 14474      00000 00000 00000 00000      00002 00100 00000 00000
 14476      00000 00000 00000 00000      04000 00000 00000 00000
 14500      00000 00000 00000 00000      00000 00000 00000 00000
          DUPLICATED LINES.
 14504      00000 00000 00000 00000      01040 14510 00000 00000
 14506      00000 00000 00000 00002      00000 00000 00000 00000
??
```

```
  P50
  14510      00014 51000 00000 00000      00000 00000 00000 00000
  14512      00000 00000 00000 00000      00000 00000 00000 00000
         DUPLICATED LINES.
  14520      60000 00001 44460 14651      00000 00000 00000 16232
--EOR--
  CM DUMP FROM   16400 TO   16530.

  16400      00234 34400 06610 01252      20010 50000 00000 00000
  16402      00000 00000 50000 00200      00000 00001 62540 16455
  16404      17252 42025 24000 00105      24240 56000 00360 16445
  16406      00000 00000 00000 16445      00000 00000 00000 16445
  16410      64300 00000 01000 16447      00000 00000 00000 00000
  16412      00000 16414 00000 00000      00000 00000 00000 00000
  16414      00000 00000 00000 00001      00000 00000 00000 00000
  16416      00000 00042 00600 00000      00000 06265 03200 00000
  16420      00000 22600 00000 00000      14000 00000 00000 05030
  16422      00000 00000 00000 00000      00000 00006 00000 01174
  16424      00000 00000 00000 00000      03000 00000 00020 02003
  16426      00000 00000 00000 00000      00000 00000 00000 00000
         DUPLICATED LINES.
  16432      00002 00100 00000 00000      00000 00000 00000 00000
  16434      00000 00000 00000 00000      00000 00000 00000 00000
  16436      00000 00000 00000 00000      02115 00000 00000 00000
  16440      00000 00000 00000 00000      00000 00000 00000 00000
  16442      01040 16445 00000 00000      00000 00000 00000 00062
  16444      00000 00000 00000 00000      00016 44500 00000 00000
  16446      00000 00000 00000 00000      00000 00000 00000 00000
         DUPLICATED LINES.
  16454      00000 00000 00000 00000      00000 00001 62540 16527
  16456      24012 00534 33000 00131      04210 46000 00360 16517
  16460      00000 00000 00000 16517      00000 00000 00000 16517
  16462      64360 00000 01000 16532      00000 00000 00000 00000
  16464      00000 16466 00000 00000      00000 00000 00000 00000
  16466      00000 00000 00000 00001      00000 00000 00000 00000
  16470      00000 00050 00600 00000      00000 00065 03200 00000
  16472      00000 22600 00000 07017      14000 00000 00000 05030
  16474      00000 00000 00000 00000      00000 00006 00000 01065
  16476      00000 00000 00000 00000      03000 00000 00000 02003
  16500      00000 00000 00000 00000      00012 00000 00000 00000
  16502      00000 00017 00000 00000      00000 00000 00000 00000
  16504      00004 00100 00000 00000      00000 00000 00000 00000
  16506      04000 00000 00000 00000      00000 00000 00000 00000
  16510      00000 00000 00000 00000      00000 00000 00000 00000
  16512      00000 00000 00000 01065      00000 00000 00000 00000
  16514      05040 16517 00012 00000      00000 00000 00000 00000
  16516      00000 00000 00000 00000      00016 51700 00000 00000
  16520      00000 00000 00000 00000      00000 00000 00000 00000
         DUPLICATED LINES.
  16526      00000 00000 00000 00000      00000 00001 62540 22413
  16530      33333 33333 33333 33355      27012 31011 16072 41716
--EOR--
  CM DUMP FROM   22140 TO   22220.
??
```

```
.
 22140      00000 00000 00000 00000        00000 00000 00000 00000
        DUPLICATED LINES.
 22146      00000 00000 00000 00000        32323 23232 05070 00025
 22150      00000 40100 00000 21707        00000 00000 00000 21761
 22152      00000 00000 00000 21761        00000 00000 01000 22147
 22154      05065 30322 15000 12641        05065 30322 15000 00001
 22156      51300 22155 13736 21722        03170 22160 50637 77776
 22160      51100 21706 03010 21707        51100 22151 63210 10055
 22162      43300 54500 56060 76500        61600 22246 01000 22344
 22164      52134 22164 10611 56620        73331 66221 05230 22167
 22166      01000 22344 61000 46000        72137 77774 03110 22164
 22170      10100 20153 03310 22204        51100 22350 10611 56620
 22172      61220 00001 05230 22174        01000 22344 61000 46000
 22174      50250 00007 61400 00006        70100 00000 03020 22177
 22176      73120 21222 03120 22176        51200 22351 43771 10622
 22200      15217 20706 67441 36662        20103 07040 22200 56620
 22202      61220 00001 05230 22204        01000 22344 61000 46000
 22204      10100 61600 22223 20155        03210 22212 10100 20152
 22206      53310 46000 61000 46000        01000 22344 61000 46000
 22210      03330 22212 50330 00001        04000 22207 61000 46000
 22212      10100 20152 03210 22217        71200 00043 20222 46000
 22214      71100 00045 20166 12121        76310 74250 20353 12131
 22216      12321 01000 22344 46000        51100 22352 10611 56620
 22220      61220 00001 61600 22246        01000 22344 61000 46000
  END OF FILE
?? end
~ZZDUMP IS A LOCAL FILE
.EB ,       0.850UNTS.
/
 07.43.35. WARNING


FOR IMPORTANT INFO TYPE EXPLAIN,WARNING.
```

POST MORTEM DUMP

Post mortem dump outputs the program state at the time of program failure, or when a call is made to PMDLOAD or PMDSTOP.

Output is in readable format, rather than an octal dump. The output includes:

- names and decimal values of variables
- the nature of the error that activated PMD
- activity on I/O files
- overlays in memory at the time of the error
- subroutine calling traceback, to the main program

Post mortem dump is enabled by specifying

```
db=pmd
```

on the FTN5 control card. It is then activated by an execution error or by a user callable routine, PMDLOAD (after which execution resumes) or PMDSTOP (after which the program is aborted).

USING POST MORTEM DUMP

The programmer must specify post mortem dump both at compilation and at execution.

- At compilation - symbol tables must be generated as part of the object deck. There are 3 ways to accomplish this:

  A) The control Statement

  DEBUG or DEBUG(ON)

  can be issued prior to compilation, or

  B) The DB parameter can be set to

  DB = PMD

  or DB

  on the FTN5 control statement

- At execution - the control statement

  DEBUG or DEBUG(ON)

  must be issued prior to load and execution.

POST MORTEM DUMP COMMANDS

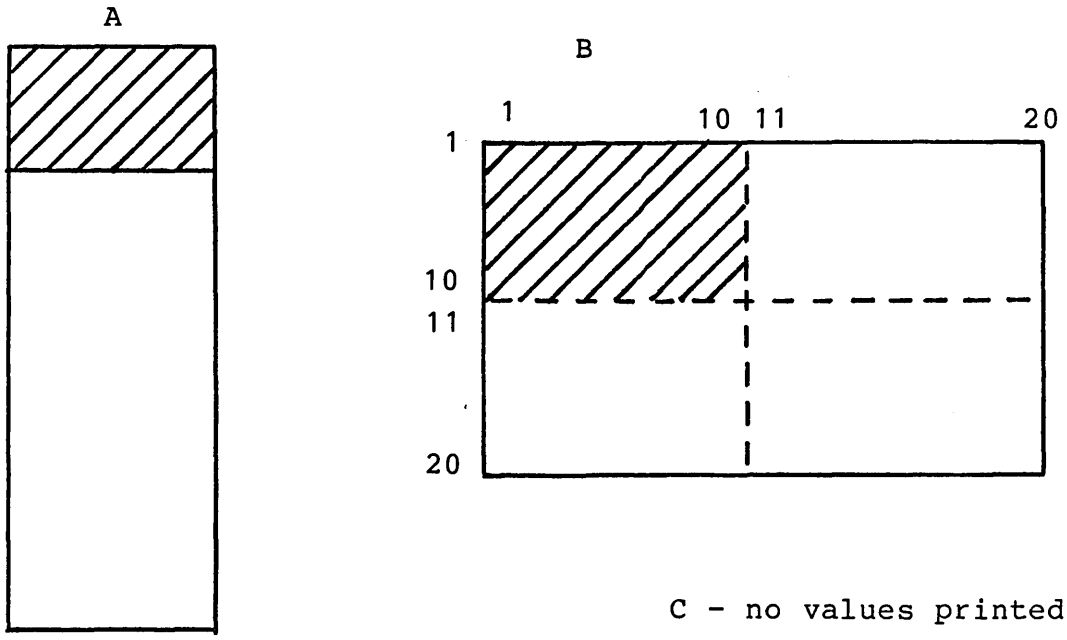
CALL PMDARRY (i,j,k,l,m,n,o)

- Limits printing of arrays to only those elements whose subscripts do not exceed what is shown.
- This command does not cause a dump, when a dump occurs the last executed PMDARRY call determines what array values are printed.
- default is (20,2,1,1,1,1,1)

EXAMPLE:

```
DIMENSION A(100),B(20,20),C(10,10,10)
CALL PMDARRY (10,10)
```

When a dump is activated, only arrays having 1 or 2 dimensions will be printed, and only those elements having subscripts less than or equal to 10, e.g., only the shaded areas of the given arrays:



C - no values printed

CALL PMDDUMP

- This call notifies the post mortem dump facility that the variables in the calling routine are to be included in any subsequent dump.

CALL PMDLOAD

- Causes an immediate dump, after which the program resumes execution.

CALL PMDSTOP

- Causes an immediate dump, after which the program is aborted.

CYBER INTERACTIVE DEBUG

CID is a supervisory program that allows a user working at a terminal to control and interact with a program during its execution. CID is available to users of any assembler or compiler under NOS and NOS/BE. Special commands are available to the FORTRAN 5 programmer if DEBUG mode is turned on during compilation.

When a program is executed under CID, the user can halt program execution at any specified program location (BREAKPOINT) or at the occurrence of a condition (TRAP) such as a reference to a particular variable. When execution is stopped, the programmer can display and alter variables and arrays, and resume execution at that or another location. Additional features enable the user to build sequences of CID commands, with loops and conditionals, and to suspend a debug session so that it can be resumed at a later time.

USE OF CID

CID does not require any changes to the user's source code. DEBUG mode must be turned on when the program is loaded and executed and, for any practical debugging, DEBUG mode should be turned on during compilation.

DEBUG MODE

Debug mode should be on during compilation and during loading and execution.

- During Compilation

  When a FORTRAN 5 program is compiled in DEBUG mode, symbol and address tables are retained which allow the programmer to interact with the program in terms of FORTRAN Line and statement numbers, and FORTRAN program variable names. Commands can be entered in FORTRAN syntax, such as

  ```
  PRINT *, A,B,C
  ```

  Debug mode is turned on for compilation by issuing the

  ```
  DEBUG (ON)
  ```

  control statement prior to execution, or by specifying

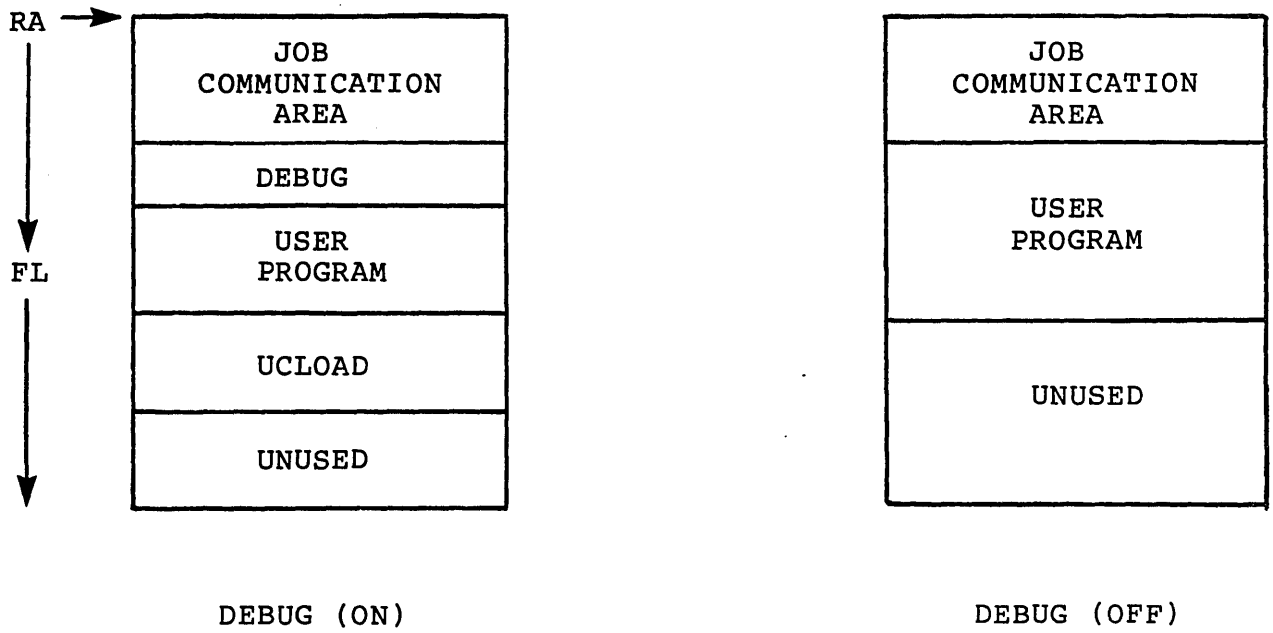
  ```
  DB = ID
  ```

  on the FTN5 control statement.

- During Loading and Execution

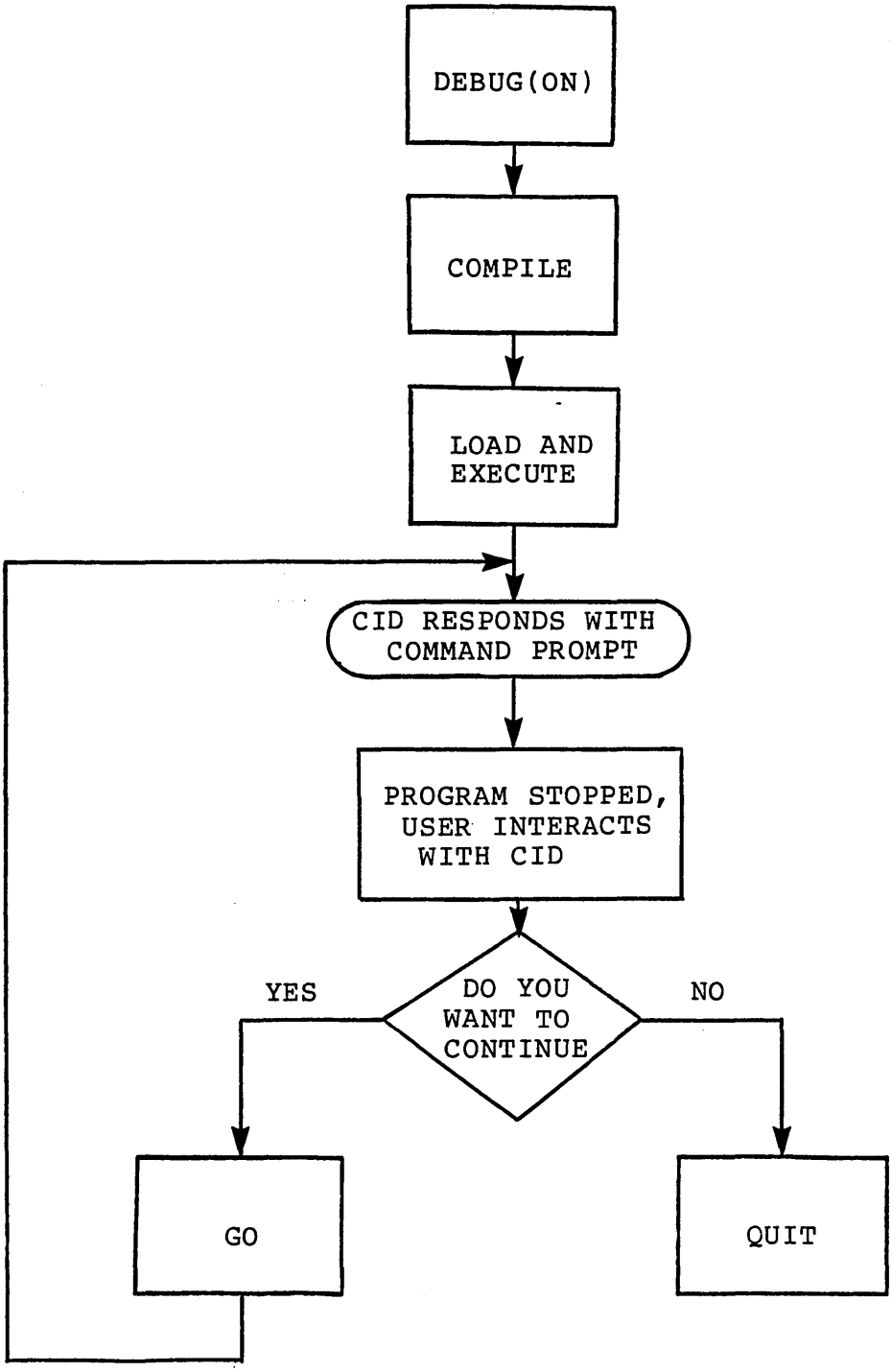
  IF DEGUG(ON) precedes loading, CID will be loaded along with the user's program into his field length. When the program is executed, control will transfer to CID, and execution will then depend on CID commands issued from the terminal.

PROGRAM LOAD

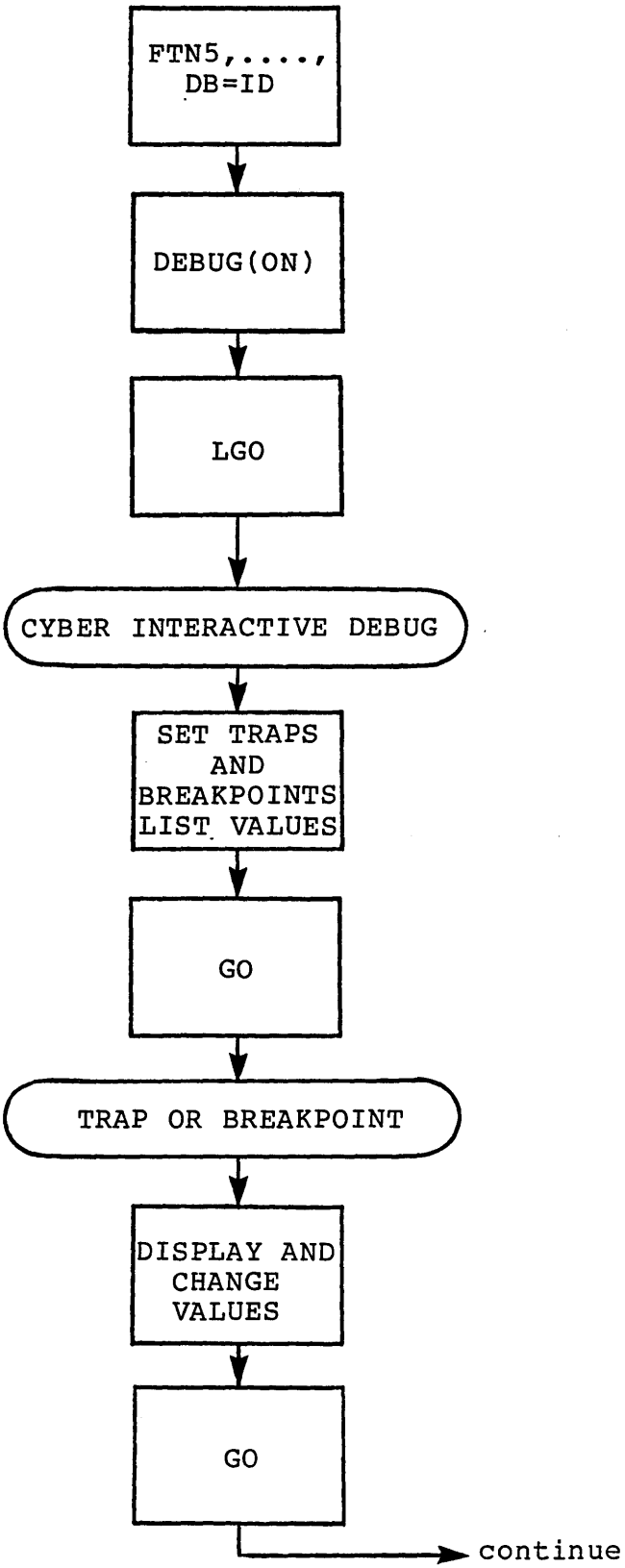


DEBUG (ON) DEBUG (OFF)

OVERVIEW OF A CID SESSION



CID INTERACTION



EXECUTION INTERRUPTS

There are two kinds of conditions under which CID will interrupt program execution, breakpoints and traps.

BREAKPOINTS

A breakpoint is a program location where execution is to be suspended. Execution is suspended before the instructions at that location are executed. A breakpoint location can be described as:

- an absolute address
- an address relative to a module or loader block
- an entry point
- an overlay
- a FORTRAN line number
- a FORTRAN statement label

TRAPS

A trap suspends execution when some condition occurs. A trap will be valid within a specified region of the program called the scope of the trap. The types of traps available are:

TRAP TYPE	STOPS EXECUTION ON
ABORT	program abort
END	normal program end
FETCH, STORE LINE	fetch or store on data before each line of FORTRAN source code
INSTRUCTION	before each machine instruction
JUMP	before any machine language jump is executed
RJ,XJ	before a machine return jump or exchange jump
INTERRUPT	when an interrupt is sent from the terminal

HOME PROGRAM

When CID suspends execution, the program that is suspended will be executing in one of its subroutines. This subroutine becomes the home program. It is more difficult to reference program locations if they are outside of the home program. The home program can be changed with the SET,HOME command.

CID COMMANDS

There are two formats for CID commands. Language independent command formats are established by CID. Language dependent commands are similar in format to commands in the programming language of the program being debugged.

LANGUAGE INDEPENDENT COMMANDS

These commands have a common format:

COMMAND NAME, OBJECT, TYPE, PARAMETERS

EXAMPLE:

SET, TRAP, JUMP, P.MAIN

For most instructions, a short form is available which combines the first two fields.

ST, JUMP, P.MAIN

FORTRAN LANGUAGE DEPENDENT COMMANDS

Four commands allow the programmer to interact with the program using FORTRAN syntax.

Assignment

EXAMPLE:

```
x = 5.0
```

GOTO statement label causes execution to resume at the specified location.

IF (Logical expression) CID command

The CID command is executed if the FORTRAN logical expression is true.

PRINT *, variable list
displays the current values of the listed variables.

EXECUTION INTERRUPT COMMANDS

SET - establishes a trap or breakpoint

EXAMPLES:

```
SET, BREAKPOINT, P.MAIN_L.1

     breakpoint in
     program main.
     line 1

SB, P.MAIN_L.1

SET, TRAP, FETCH, P.MAIN_AR1+9B

     trap on fetch
     from 10th element
     of array AR1
     in program main

ST,FETCH, P.MAIN_AR1+9B
```

CLEAR - erases a trap or breakpoint

EXAMPLE:

```
CLEAR,BREAKPOINT, P.MAIN _ S.110

     clear breakpoint
     a statement #110
     in program main

CB,P.MAIN _ S.110
```

LIST - displays traps, breakpoints, and other information about CID

EXAMPLES:

```
LIST, BREAKPOINT, P. MASSON

     lists all breakpoints in
     program masson

LB, P.MASSON

LIST, TRAP, FETCH, P.SORT

LT, FETCH, P.SORT

     list all fetch
     traps in program sort
```

SAVE - copies the definitions of traps and breakpoints to a local file

EXAMPLES:

```
SAVE, TRAP, FILEA *

SAVET, FILEA, *

     all traps
     are saved
     on "filea"

SAVE,*,F

     all breakpoint, trap,
     and group definitions
     are saved on file "f"
```

a READ command is available to restore these definitions

STRACE

- A subroutine which provides the current subroutine calling traceback to the main program
- The trace back is written to output

LEGVAR

- A function which checks the value of a variable and returns a -1 if the value is indefinite, +1 if out of range, and 0 otherwise.

Lesson 5

CREATION OF USER LIBRARIES

LESSON PREFACE:

OBJECTIVES:

REFERENCES:

PROJECTS:

CREATION OF USER LIBRARIES

One or more relocatable records are located on file FILBIN. The last of these records is BIN5. One or more absolute records are located on file FILEABS. One is ABS5. It is desired to place all the relocatable and absolute records together on a User Library called LIB1. The following job stream is for interactive use, but with a few small changes it could be used for batch.

```
   GET,FILBIN
   ATTACH,FILEABS
   LIBEDIT(B=FILBIN,P=O,U)
?  *BEFORE *,REL/*
?  (CR)
   DEFINE,NEW1=LIB1
   LIBEDIT(P=NEW,B=FILEABS,N=NEW1,U)
?  *INSERT REL/BIN5,ABS/*
?  (CR)
   DEFINE,LIB1
```

To execute from this library:

```
   ATTACH,LIB1
   LIBRARY(LIB1)
   LIBLOAD(LIB1,BIN5)
   EXECUTE,BIN5
```

Or:

```
   ATTACH,LIB1
   GTR(LIB1,ABS5)ABS/ABS5
   ABS5
```

Or:

```
   ATTACH,LIB1
   LIBRARY(LIB1/A)
   BIN5
   ABS5
```

Note that these examples are for NOS V2. They do NOT work at present on NOS V1.4 under COMSOURCE, CYBERNET, or MIPC.

UPDATING AN EXISTING USER LIBRARY

The following code will generate a "replacement" of an existing binary record on a User Library. XEDIT is used to generate the change-set. UPDATE is then used to update the Program Library. Finally, the User Library is modified.

```
   XEDIT,chgset
?  ...
   ...
   END,,SL
   ATTACH,OLDPL=pl
   UPDATE,I=chgset,L=A1
   FTN5,I,L,EL=T,LO,OPT=2
   DAYFILE,,FTN5                    This is to verify that there
   ATTACH,OLD=LIB1                  are no errors in the compile
   DEFINE(NEW=NEWLIB)
   LIBEDIT,U
?  (CR)                             Replacements are by default
```

To test the new library without execution:

```
   RETURN,MLIST
   ATTACH,NEWLIB
   LIBRARY(NEWLIB)
   LDSET(MAP=SB/MLIST)
   LIBLOAD(NEWLIB,ept)
   NOGO.
```

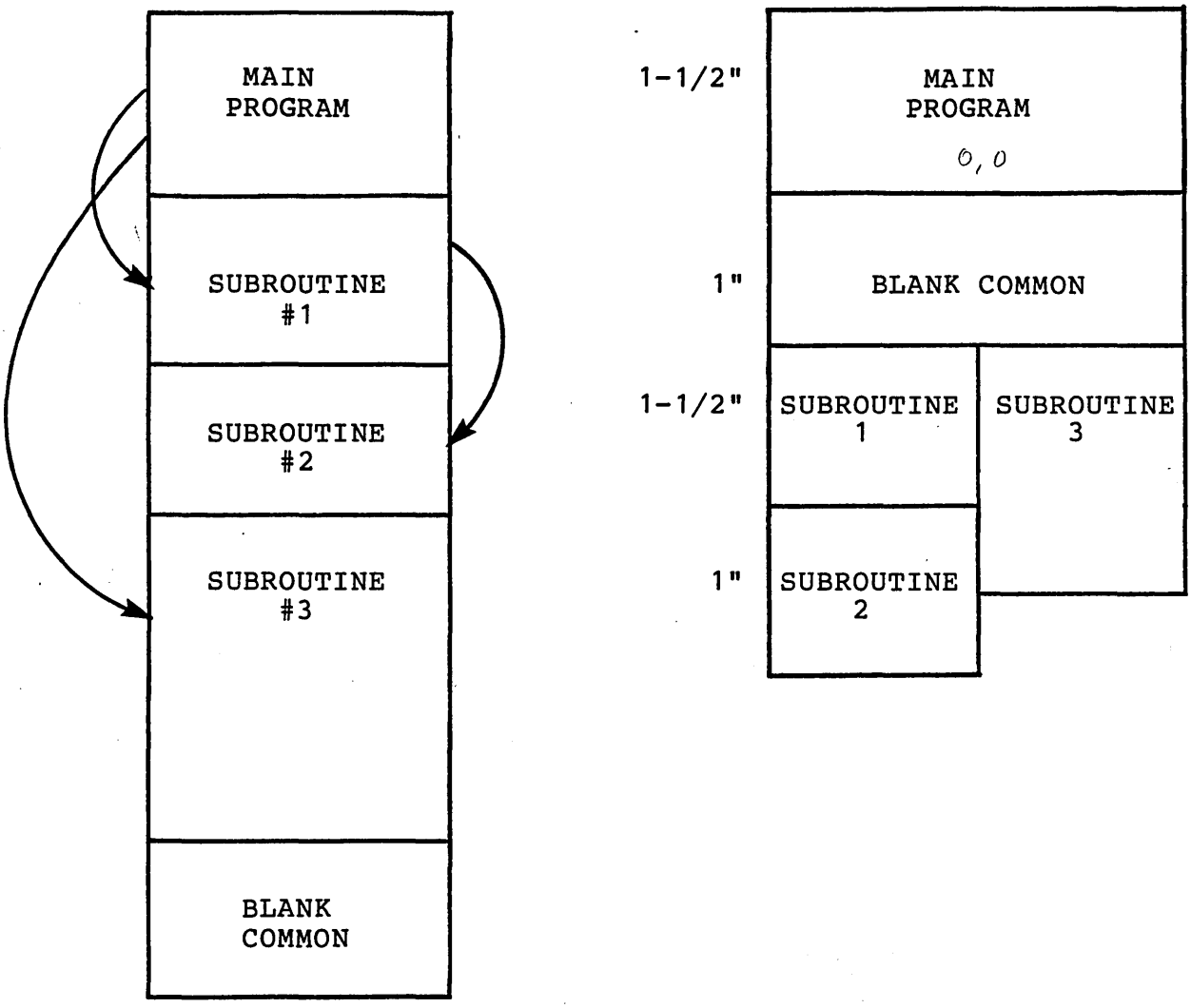
The above code will only work on NOS V2 and above; it will not work on present COMSOURCE, CYBERNET, or MIPC systems.

OVERLAYS

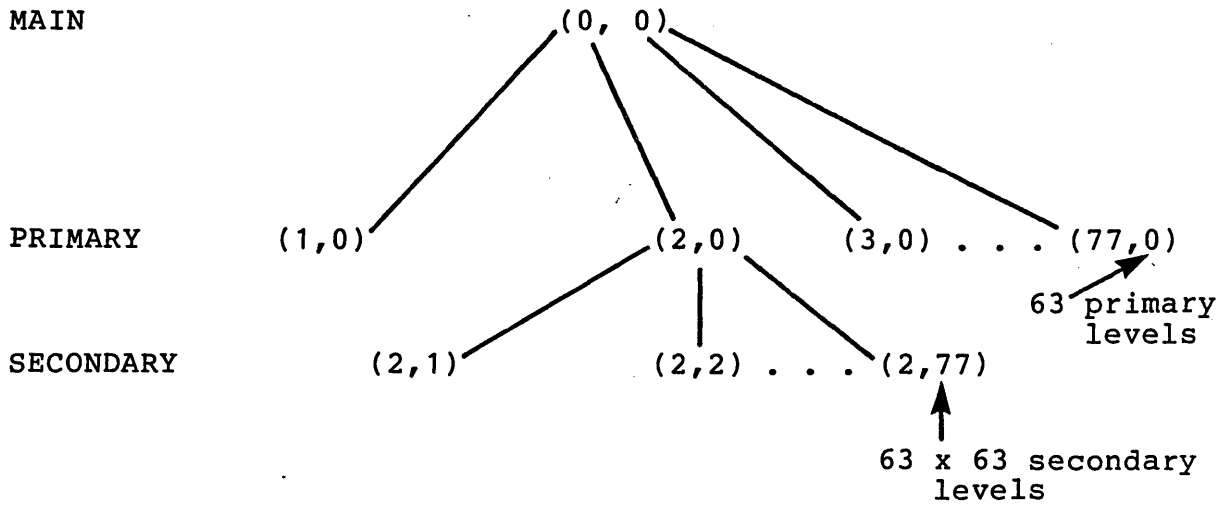
An overlay is a part of a program that does not need to be in memory for the whole time that the program is executing. Overlays are brought in from disc when they are needed, and written over when they are not needed. This reduces the amount of memory needed for job execution.

The programmer controls when the overlays are called in. This requires changes to the FORTRAN source.

FIELD LENGTH REDUCTION



OVERLAY LEVELS



NOTES:

- The main overlay is always loaded
- The origin of each primary overlay is immediately after the main overlay
- The origin of each secondary overlay is immediately following its primary overlay
- Only three overlays can be in memory at any one time

THE OVERLAY DIRECTIVE

OVERLAY (lfn, l_1, l_2, origin, OV = n)

optional

where

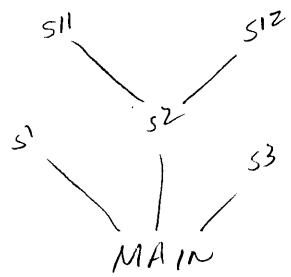
- lfn is the file onto which the overlay should be written
- l_1 is the primary level number
- l_2 is the secondary level number
- origin specifies the beginning of the overlay relative to the beginning of the main overlay, the beginning of blank common, or an entry point is a lower level overlay
- OV=n is an optional parameter for the (0,0) level overlay specifying that the fast overlay loader will be used. n is the number of higher level overlays in the structure.

```
ISON.
USER.COMML.RDWEST.
PROJECT,#RW1#104--01EK100D.
COPYCR..DEBG.
REWIND.DEBG.
ATTACH.VENXLIB,ID=ISON.
ATTACH.TAPE1.VENDATA.ID=ISON.
LIBRARY(VENXLIB)
MAP.ON.
SEGLOAD.
EXECUTE.DRIVER.

CS      DEBUG
CS      CALLS(MAIN1.MAIN4.SEEK.RCV1.INTL.INTL1)

            GLOBAL     CDATA,MGMTIO.CINPT,NEUTIN,VIERRS.XCTRL.VTROL
            GLOBAL     CNTRL.VCTRL.USRID.IOUNT.AFLUY.ADRES.LIMTS.FSNAP
            GLOBAL    DEASU.COMSAM,ADSUP.SAVED
            TREE       DRIVER-(CONTROL.IMPROSE,OVENTR.VENT4EU-(VENT,ADN1.AJ
.NT,EDIT,MAC1,PHIA.EASU,ORLX.OUTR-(ETR1.DOIN,FOUX,POUX,SOUY.INRX-(LEK1,L
.EK2.LEK3.LEK4.LEK5.LEKX)).JERT,PERT))
VENT        INCLUDE   VENT.CORE.CORI.CORP.GRAM,CORD.CORR.DDSP.DASU.JPRT
ADN1        INCLUDE   ADN1.ADN2.ADN3,DSDF.DIMS.AJDS.DIM1.DIM2.DIM3.DCID.ZVR
.V.CHGV.FLRD.FLMH
AJNT        INCLUDE   AJNT.COMC.LCAL.FLXP,FXSR.RSWV.REV1.PROS.ZIO3.SAV1.SAV
.3.SAV4,SAV6,SAV2
EDIT        INCLUDE   EDIT.RSQS.FISS.FLXW,JIMT.NBAL.PNOM.PIVL.PTZF.SOBL.POU
.T
PHIA        INCLUDE   PHIA.PHI1.PHI2.PHI3.PHI7.EDBN.NDBN.SDBN.GRXP.TOIP.RDR
.N
ORLX        INCLUDE   ORLX.BATG.CONS.ORLA.ORLB.ORLC.ORLD.ORLE.ORLF.ORLR.RCO
.V.MICK
OUTR        INCLUDE   OUTR.ATED.BALC.CHRF.FLUX.FSOR.JOSR.SSOR.PSOR.XTRP.ZIN
.S.BHAV.FFGG.CHEV.NEWB.DELX.RDUE.REEX.RDER.LTRG
ETR1        INCLUDE   ETR1.MUEX.SGDA
DOIN        INCLUDE   DOIN.PREC.RRES.WRES
FOUX        INCLUDE   FOUX.FOU1.FOU2.FOU3.FOU4.FOU5.FOU6
POUX        INCLUDE   POUX.POU1.POU2.POU3.POU4.POU5
SOUY        INCLUDE   SOUY.SOU1.SOU2.SOU3.SOU4.SOU5.SOU6
INRX        INCLUDE   INRX.INR1.INR2.INR3.INR4.INR5.INR6.DELX
LEK1        INCLUDE   LEK1.LOU1
LEK2        INCLUDE   LEK2.LOU2
LEK3        INCLUDE   LEK3.LOU3
LEK4        INCLUDE   LEK4.J1C4.LOU4.QUUF.DELX.SOUX
LEK5        INCLUDE   LEK5.J1C5.LOU5
LEKX        INCLUDE   LEKX.J1CX.LOUX.SOUZ
JERT        INCLUDE   JERT.JAFA.JAPS.JIFF.JUFY.JGET.JMAP.MRPT.PERO.RTUB.BRB
.1.BRB2.QOUT
PERT        INCLUDE   PERT.DAFA.LIFE.MAPS.TUFY.PMAP.MRPT.PERO.RTUB.RBB1.BBB
.2.QOUT
            END
```

OVERLAY CAPSULES

OVCAPS are a more flexible alternative to overlays. They are not specified by levels, so any combination can be in memory at the same time. When an OVCAP is loaded, space is made available by Common Memory Manager.

When OVCAPS are used, the (0,0) overlay must be present. All OVCAPS are considered to be logical extensions of this overlay. Communication between OVCAPS is carried out using common blocks in the (0,0) overlay.




```
TREE MAIN - (S1, S2 - (S11, S12), S3)
END
```

USING OVCAPS

- Source code must be changed to identify the OVCAPS, to call them into memory, and to release their memory.

- OVCAPS are identified with an OVCAP directive. Each OVCAP directives should be immediately followed by a SUBROUTINE statement. The OVCAP takes the name of that first subroutine.

- The OVCAP can contain other program units after the first subroutine.

- OVCAP's are controlled using three calls:

```
CALL LOVCAP (name) - load
CALL XOVCAP (name) - execute
CALL UOVCAP (name) - unload
```

- A program using OVCAPS must be generated by a load sequence which ends in a NOGO directive.

Lesson 6

CYBER RECORD MANAGER

LESSON PREFACE:

OBJECTIVES:

REFERENCES:

PROJECTS:

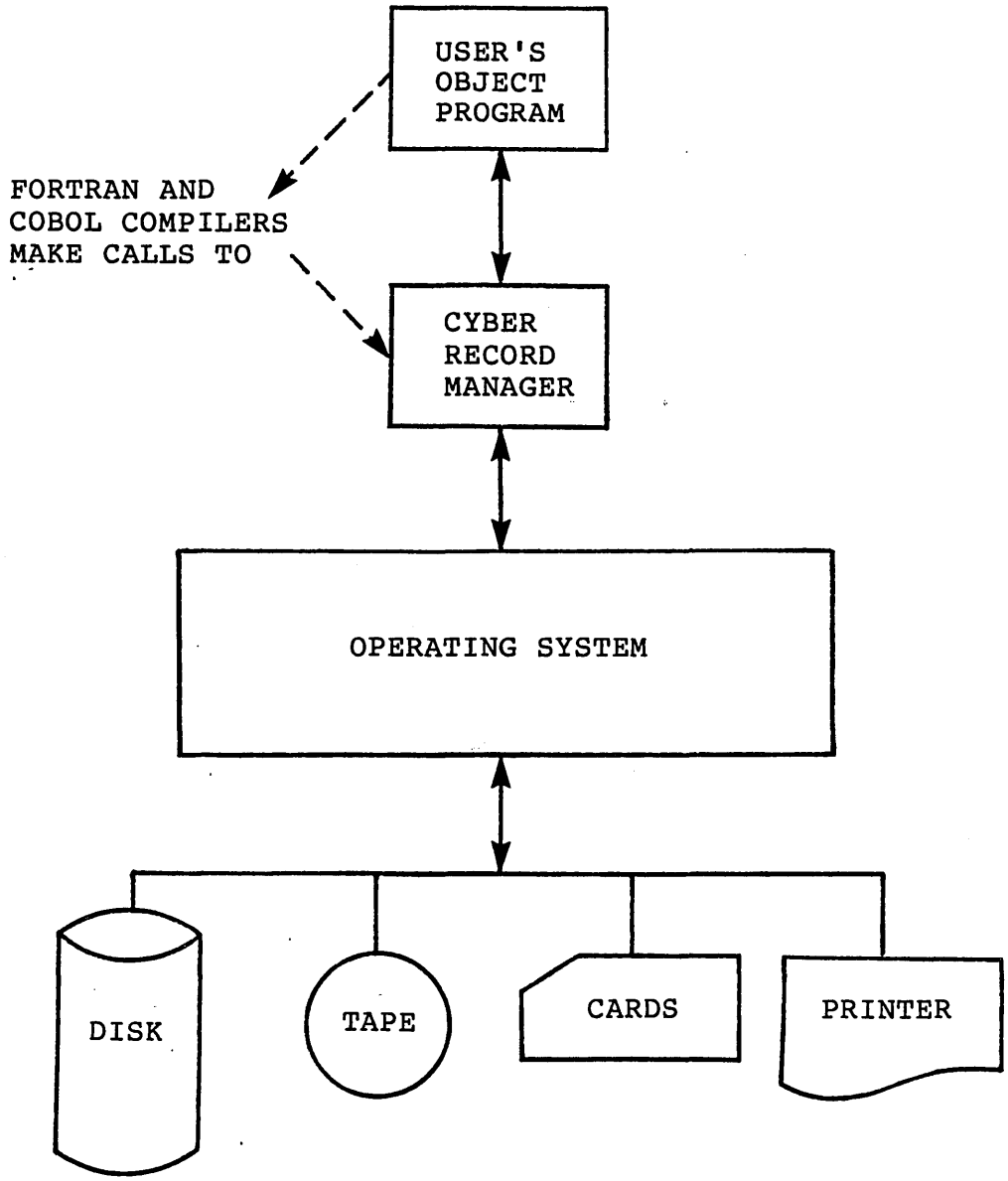
CYBER RECORD MANAGER

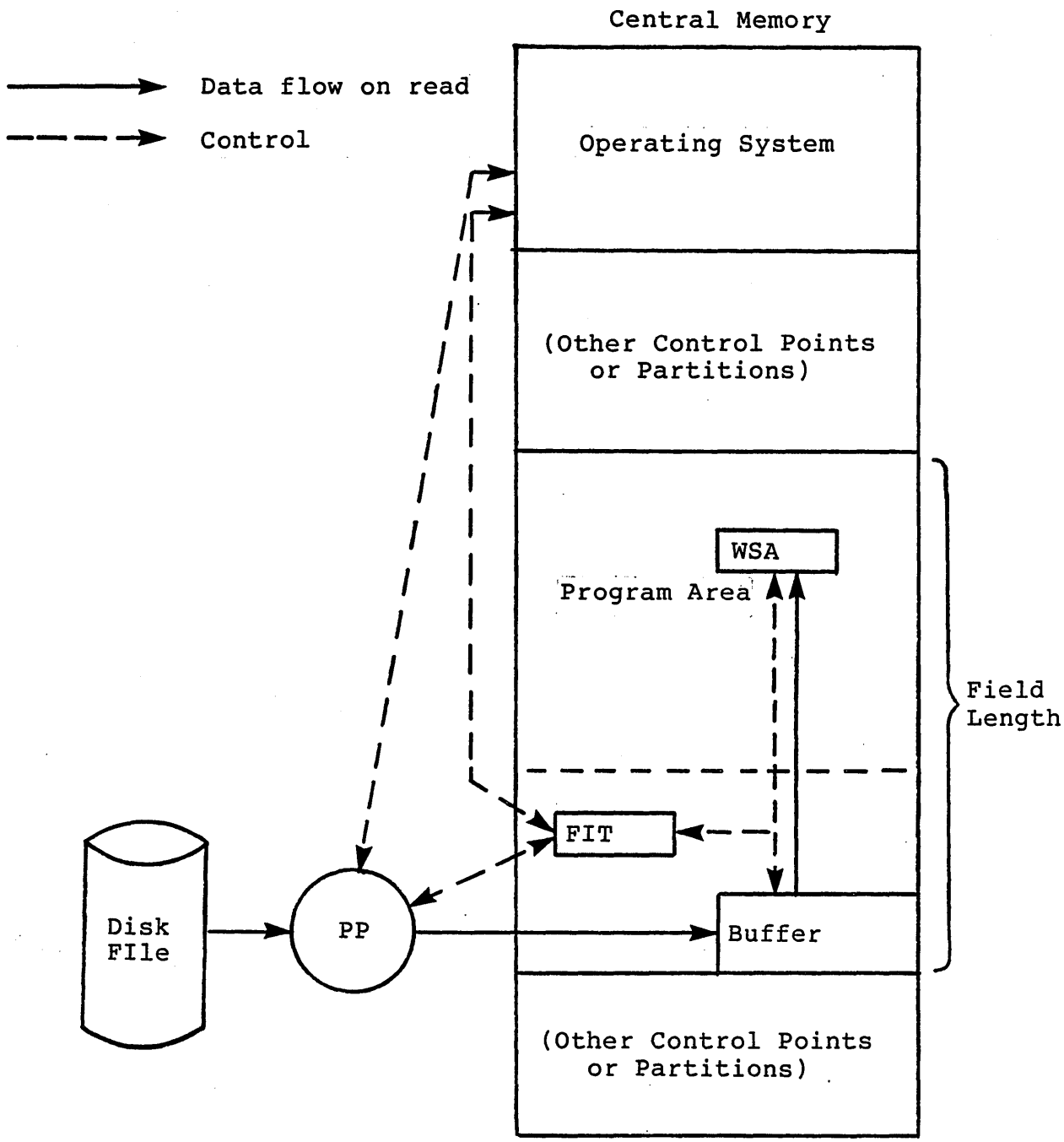
CYBER record manager is an input/output processor that provides an interface between a user's program and the system routines that processes external files. The user program may be written in any of the major CYBER compilers or database utilities so, for example, CRM allows the FORTRAN programmer to access files created by a COBOL program, using the file structures available in COBOL.

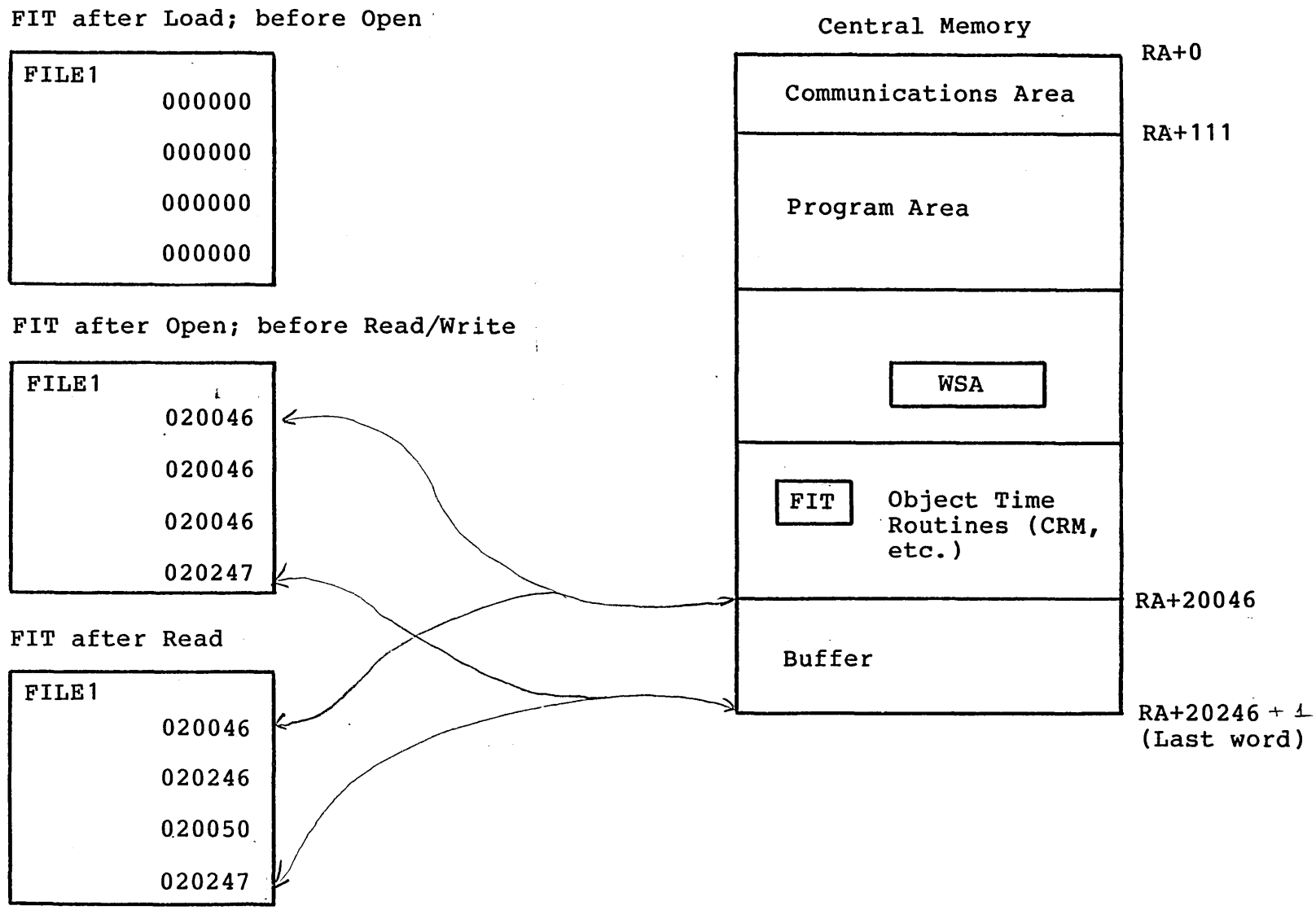
The FTN5 compiler uses CRM calls to handle FORTRAN I/O. The programmer can take advantage of the full capability of CRM by calling it directly using subroutine calls which correspond to the CRM COMPASS macros.

CYBER RECORD MANAGER

GENERAL PURPOSE INPUT-OUTPUT MANAGER, ALLOWING FILE COMPATIBILITY BETWEEN COBOL, FORTRAN, AND OTHER PROCESSORS




Central Memory
Data flow on read
Control
Operating System
(Other Control Points
or Partitions)
WSA
Program Area
Field
Length
FIT
Disk
FIle
PP
Buffer
(Other Control Points
or Partitions)


FIT after Load; before Open
FILE1
000000
000000
000000
000000
FIT after Open; before Read/Write
FILE1
020046
020046
020046
020247
FIT after Read
FILE1
020046
020246
020050
020247
Central Memory
RA+0
Communications Area
RA+111
Program Area
WSA
FIT
Object Time Routines (CRM, etc.)
RA+20046
Buffer
RA+20246 + 1
(Last word)

CRM COMMANDS

Establishing the FIT:

```
CALL FILExx(fit,keyword,value,keyword,value,,,)
```

where:

- xx identifies the file organization

  FILESQ
  FILEIS
  FILEDA
  FILEAK
  FILEWA

- keyword identifies the FIT field, value goes into the field in the actual FIT

- keywords and most values are specified as left justified character strings

- single FIT values may be read with IFETCH and written into the FIT using STOREF

- FIT values compiled into a FORTRAN program can be changed at execution time using a FILE card

EXAMPLES:

In a FORTRAN program:

```
CALL FILEWA(WFIT,'LFN','WAFILE','RT','F','EFC',3,'FL',30)
...
CALL STOREF(WFIT,'WA',34)
...
IWA=IFETCH(WFIT,'WA')
```

And by Control Statements:

```
FILE(WAFILE,FO=WA,FL=32)
```

OPENING THE FILE:

CALL OPENM(fit,pd)

- pd is a field indication processing direction:

  'I-O' or 'OUTPUT'

CLOSING THE FILE:

CALL CLOSEM(fit,cf,type)

- cf is a single field containing one of:

'R'	REWIND
'N'	NO REWIND
'U'	UNLOAD
'RET'	RETURN

- type specifies FILE or VOLUME

Example: CALL CLOSEM(WFIT,'N')

RECORD TYPES

THE RECORD TYPE IS DETERMINED BY THE 'RT' FIELD OF THE FIT. THERE ARE 8 DIFFERENT RECORD TYPES, SOME OF WHICH ARE USED BY FORTRAN, SOME BY COBOL, AND SOME BY OTHER SYSTEMS.

RT=Z
- ZERO BYTE TERMINATOR
- USED FOR INTERFACE WITH CARD READER/PUNCH, LINE PRINTERS, AND COMMUNICATIONS EQUIPMENT
- EACH RECORD STARTS ON A WORD BOUNDARY
- TERMINATED WITH AT LEAST 12 BITS OF OCTAL ZERO ENDING A WORD
- USED BY FORTRAN FOR FORMATTED READ AND WRITE

RT=W
- CONTROL WORD PREFIX TELLING THE LENGTH OF THE RECORD AND THE PREVIOUS RECORD
- USED BY FORTRAN FOR UNFORMATTED READ AND WRITE

RT=F
- FIXED LENGTH RECORDS
- LENGTH IS DECLARED IN THE 'FL' FIELD OF THE FIT
- MAY BE DECLARED EITHER BY:

```
CALL FILESQ(FIT1,....'RT','F','FL',40)

      OR

FILE(FILE1,....RT=F,FL=40)
```

RT=D
- DECIMAL COUNT RECORDS
- RECORDS ARE VARIABLE LENGTH
- A FIELD WITHIN THE RECORD CONTAINS A COUNT OF THE CHARACTERS IN THE RECORD
- USED BY COBOL; USEFUL FOR INTERFACING WITH COBOL PROGRAMS

RT=T - TRAILER COUNT RECORDS

- EACH RECORD CONSISTS OF A "HEADER" WHICH IS A FIXED LENGTH, AND A VARIABLE NUMBER OF TRAILERS, ALL OF WHICH ARE THE SAME LENGTH

- USED BY COBOL; USEFUL FOR INTERACE WITH COBOL PROGRAMS

RT=R - RECORD MARK RECORDS

- USED BY "OLD" IBM AND HONEYWELL; RETAINED FOR COMPATIBILITY

- A "RECORD MARK" IS APPENDED TO THE RECORD; THE OBJECT CODE RETURNS ALL DATA UNTIL AND INCLUDING THE RECORD MARK.

- DEFAULT RECORD MARK IS AN O"62"

RT=S - SYSTEM LOGICAL RECORD

- A SHORT PRU MARKS THE END OF THE RECORD

- NOT USED BY FORTRAN; USED BY NOS AND NOS/BE

RT=U - UNDEFINED RECORD

- THE PROGRAM READS OR WRITES THE QUANTITY OF CHARACTERS PRESENTLY INDICATED IN THE 'RL' FIELD OF THE FIT. THE USER MUST PERFORM ALL BLOCKING AND DEBLOCKING.

FILE ORGANIZATION

CRM recognizes five types of files

BASIC ACCESS METHODS

Records are accessed according to their position with regard to the beginning of the file.

- SEQUENTIAL

  A sequential file has a "window" into a current record. When this record is read, the window moves to the next record.

- WORD ADDRESSABLE

  A word addressable file is a collection of contiguous words stored on disc. Each word is 10 characters. Words are addressed using an index count of a number of words from the beginning of the file.

ADVANCED ACCESS METHODS

Records are accessed according to a key. Secondary keys may be defined.

- INDEXED SEQUENTIAL

  In an indexed sequential file, records are ordered in sequence according to the value in a primary key field. The records can be accessed sequentially or randomly according to the key value. Index tables relate the keys to the physical disc location of the record.

- DIRECT ACCESS

  Direct access files allow rapid access to the disc record with a user supplied key field. The disc block containing the record is found by applying a hashing algorithm to the key value.

- ACTUAL KEY

  An actual key record is composed of a number of data blocks. Each record occupies a slot within a block. When a record is written on an actual key file, CRM returns the block and slot number where the record has been stored. To read the record, the user requests it by actual block and slot.

Lesson 7

Sample FORTRAN Program with SORT Call

LESSON PREFACE:

OBJECTIVES:

REFERENCES:

PROJECTS:

```
get,tape10,lfimain,lfisort
/ftn5,i=lfimain,lo=-a
1 FTN 5.1+587          84/09/13. 13.18.11 PAGE      1
     PROGRAM MAIN      74/176  OPT=1,ROUND= A/ S/ M/-D,-DS
          DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER
          /-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,PL=5000
          FTN5,I=LFIMAIN,LO=-A.


        1        PROGRAM MAIN(OUTPUT)
        2        PRINT*,'STARTING SORT'
        3        CALL SORTIT(10,11,20,40)
        4        PRINT*,'SORT IS COMPLETE'
        5        STOP
        6        END
     0.097 CP SECONDS COMPILATION TIME.
/ftn5,i=lfisort,lo=-a
1 FTN 5.1+587          84/09/13. 13.18.45 PAGE      1
  SUBROUTINE SORTIT    74/176  OPT=1,ROUND= A/ S/ M/-D,-DS
          DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER
          /-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,PL=5000
          FTN5,I=LFISORT,LO=-A.


        1        SUBROUTINE SORTIT(P1,P2,P3,P4)
        2        INTEGER P1,P2,P3,P4
        3        CALL SMSORT(P4)
        4        CALL SMFILE('SORT','FORMATTED',P1,'REWIND')
        5        CALL SMFILE('OUTPUT','FORMATTED',P1,'REWIND')
        6        CALL SMKEY(P2,1,P3,0,'DISPLAY','COBOL6')
        7        CALL SMEND
        8        RETURN
        9        END
     0.188 CP SECONDS COMPILATION TIME.
/lgo
 STARTING SORT
 SORT IS COMPLETE
     0.281 CP SECONDS EXECUTION TIME.
/copy.,tape10
555555555 GARFIELD, JAMES ABRAM
999999999 GRANT, ULYSES S.
777777777 HOOVER, HERBERT
222222222 JACKSON, THOMAS
333333333 LEE, ROBERT EDWARD
111111111 LINCOLN, ABRAHAM
666666666 MONROE, JAMES
444444444 ROOSEVELT, THEODORE
000000000 WASHINGTON,GEORGE
888888888 WILSON, WOODROW
 EOI ENCOUNTERED.
```

WRITE (5, 1000) A, B C

APPENDIX A

STUDENT PROJECTS

```
CHAR  ARRAY (100) * 40, SSN1 * 3, SSN2 * 2, SSN3 * 4, NAME * 30, ALL * 40
OPEN (UNIT=5, FILE = 'TAPE10',) RECL= 40, RT = Z, BT = C)
READ (5, 1000,, END = 900)(ARRAY (I), I = 1, 100)
1000  FORMAT (
```

```
PROGRAM X
CHARACTER  X*10 , Y*20, Z*30
    ≀
CALL    SUB1 (X)
    ≀
CALL    SUB2 (Z)
```

Right Justify

DIMENSION R(*)

LEN IS FUNCTION BRING'S BACK I THE LENTH

EX: I = 10

```
SUBROUTINE  SUB1 (A)
CHARACTER   A * (*)
I = LEN (A)
A (1:I) = '  '
RETURN
```

SAY'S HOW DON'T KNOW HOW LONG A IS

PUT BLANKS IN ARRAY (X) IN POSITION 1 — 10

SIGN ON
PRESS FUNCTION # 1 RET
9 - 482 - 4708 (CR)
FAMILY (CR)
USER NAME ASE123G (CR)
PASSWORD TEACH86
PROMP

STUDENT PROJECT #1

A program is to be written, called BINSCH, which will perform the following tasks:

1. Read a file named TAPE10 into memory forming an array. The format of TAPE10 is:

   Positions 1-9 — Social security number
   Position 10 — ~~Unused~~
   ~~Positions 11-30~~ — Name, i.e., SMITH, John

   The file is BT=C, RT=Z, RL=40. (The extra 10 characters are required for the Z-byte terminator; they will not be returned to the user when using normal FORTRAN READs.) The file has no password. (DO NOT MAKE ANY CHAGES TO TAPE10.)

2. The array should allow for 100 names on the file; but the file may contain less names than 100.

3. The program will then query the user for a social security number. The program will then perform an INDEX function on the table looking for the SSN read. If matched, the program will return the name of the person having that SSN, as contained on TAPE10.

4. If no match is found, then the program will return an appropriate diagnostic message as determined by the student.

5. The program will allow for multiple entries of SSN's, and will contain an appropriate END= parameter and send a message back to the terminal station that the run is complete. A null carriage return will be used to signal the input is complete.

6. ~~Since the data on TAPE10 is word aligned for CYBER 170 and NOS V2, the student may use INTEGER and REAL for internal comparisons should they desire.~~ USE ALL CHAR STRINGS

After writing the program, the student should debug it as required, using their own test data from the terminal. (TAPE10 is already on the User Name as an Indirect file.)

Save your code; you will use it again later in the seminar.

A solution will be provided later in the seminar.

GET, TAPE10 /UN=ASE1222

STUDENT PROJECT #2

1. File LFI2D contains two source modules, a main called DEBUG and a subroutine called QUAD. Both are written in FORTRAN V5. When executed DEBUG asks for the three parameters for a quadratic equation. Upon receipt of these parameters, which it expects in format F6.4, it calls QUAD and computes the real roots for that quadratic equation. Complex roots, if any, are not returned.

2. For those who may not remember their high school algebra, the general format of a quadratic, in FORTRAN terminology, is:

   Y=(-B+SQRT(B**2-4*A*C))/(2*A) for one root, and the same with a negative sign in front of the SQRT for the other root.

3. There are at least five errors in the two modules, depending on what one calls an error. The project is to find the errors - and any others we may have missed. Use any of the debugging measures discussed, plus any others you may know.

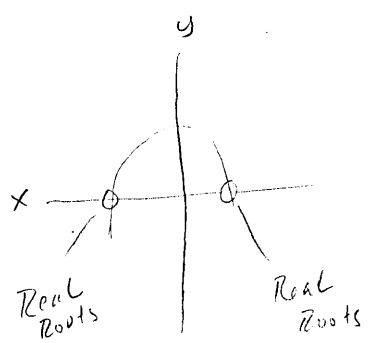
4. Save your code -- you will need it again later.

OPTIONAL:

Place all the binaries and PROC's thus far on a User Library and execute from it.

OPTIONAL:

Using FDL, place the main on the root, and QUAD on a capsule. Execute it.



$$x = \frac{\pm\sqrt{b^2 - 4ac}}{2a}$$

STUDENT PROJECT #3

In Student Project #1 we wrote a program called BINSCH. Now modify BINSCH as follows:

(BEFORE STARTING THE FOLLOWING, MAKE A COPY OF YOUR BINSCH PROGRAM FROM PROJECT #1. YOU WILL NEED THIS "ORIGINAL" COPY LATER IN ANOTHER PROJECT. MAKE THE COPY PERMANENT.)

1. Change all the I/O from READ and WRITE to explicit CRM calls. Change TAPE10 to BT=C, RT=F file with a record length of 30.

2. Remove the array. Instead, as TAPE10 is read, write a file, TAPE20, that is FO=IS and using the SSN as the key.

3. Change the queries so that it is TAPE20 that is matched for an equal key. If a match if found, produce the previous name message. If no match is found, produce the previous diagnostic.

The instructor will supply a different file called TAPE30 which is the same data as before but in RT=F format. Inside your program continue to call it TAPE10. This may be done by a JCL statement: DON'T DO

GET(TAPE10=TAPE30)

Do the required debugging as necessary. You may need to make some changes in your control cards depending on how you wrote BINSCH.

A solution is provided later.

OPTIONAL:

Place this newly modified program on the ULIB from the Optional Problem 2 and execute from it.

LFN
BT
RT
FL

EFC
ORG
ZMK
RT
RKW
RKP
KA
KP
WSA

STUDENT PROJECT #4

In Student Project #2 we had a program named DEBUG which called a subroutine QUAD. Modify your corrected DEBUG so it will call a COBOL V5 subprogram called COBQUAD. Parameters are passed to COBQUAD in a COMMON block called CCOMMON in the same order as they were passed to QUAD.

A listing of COBQUAD follows.

Note that COBQUAD in turn calls a FORTRAN V5 subroutine named ROOT. ROOT does the actual square root computations. The student is to write ROOT. ROOT uses parameters passing from COBQUAD. TEMP is the input to ROOT and is the value for which the square root will be determined. ROOT is the returned parameter. Both parameters will be REAL.

Write ROOT, modify DEBUG as required, execute, and debug if necessary. Use dumps, loader maps, etc., as required.

A solution will be provided later.

OPTIONAL:

Place these new modifications on your ULIB and execute from it.

```
/xedit,list
 XEDIT 3.1.00
?? p50
S
1CDC COBOL 5.3 - LEVEL 587        SOURCE LISTING OF COBQUAD
   AOPT= 76/CDC/CDCS2            84/09/12. 15.29.57.    PAGE        1


 1                IDENTIFICATION DIVISION.
 2                PROGRAM-ID.        COBQUAD.
 3               *
 4               *    THIS PROGRAM IS THE COBOL5 SUBROUTINE TO BE CALLED BY THE
 5               *    STUDENT IN PROJECT 5 OF THE CYBER ADVANCED FORTRAN WORKSHOP.
 6               *
 7                ENVIRONMENT DIVISION.
 8                DATA DIVISION.
 9                COMMON-STORAGE SECTION.
 10               01  A   COMP-2.
 11               01  B   COMP-2.
 12               01  C   COMP-2.
 13               01  X1  COMP-2.
 14               01  X2  COMP-2.
 15               WORKING-STORAGE SECTION.
 16               01  TEMP        COMP-2.
 17               01  ROT         COMP-2.
 18               PROCEDURE DIVISION.
 19               P1.
 20                   COMPUTE TEMP = (B ** 2 - 4 * A * C).
 21                   ENTER FTN5 "ROOT" USING TEMP ROT.
 22                   COMPUTE X1 = ( - B + ROT) / 2 * A.
 23                   COMPUTE X2 = ( - B - ROT) / 2 * A.
 24               P2.
 25                   EXIT PROGRAM.
0          COLUMN   1         2         3         4         5         6         7
  >>>>              8
           12345678901234567890123456789012345678901234567890123456789012345678901
  >>>>     234567890
--EOR--
0


-CDC COBOL 5.3 - LEVEL 587      CROSS REFERENCE FOR COBQUAD
   AOPT= 76/CDC/CDCS2
   REFERENCED DATA-NAMES         LINE COLUMN            REFERENCE(S)

 A                               10    12            20    22    23
 B                               11    12            20    22    23
 C                               12    12            20
 ROT                             17    12            21    22    23
 TEMP                            16    12            20    21
 X1                              13    12            22
 X2                              14    12            23
0
   REFERENCED DATA-NAMES         LINE COLUMN            REFERENCE(S)
```

STUDENT PROJECT #5

Using the program, BINSCH, as prior to the change for FO=IS, modify the program so that it performs the following editing of data:

1. Check that the social security numbers that are inputted on the terminal are truly numeric, and exactly nine digits long.

2. Verify that the data coming in on TAPE10 is correct, i.e., no leading blanks; contain only A through Z, and not more than one hyphen, comma, or period. If may contain one numerical digit, as JONES, John 3rd.

Suggestion: Do each of these steps one at a time. Don't start on the next until the previous works correctly. This way if you inadvertently insert a bug you will know which portion caused the bug, and it will be easier to correct.

Debug as required.

A solution is provided.

OPTIONAL:

Add this code to your ULIB and execute it.

```
1 ....   SUBROUTINE QUAD        73/176  OPT=0                          FTN 5.1+528        81/10/12. 16.30.18        PAGE      1

      1                 SUBROUTINE QUAD
      2                 COMMON /PARAMS/A,B,C,D,X1,X2
      3                 IF((B**2).LT.(4*A*C)) GO TO 100
      4                 X1=(-B+SQRT(B**2-4*A*C))/2*A
      5                 X2=(-B-SQRT(B**2-4*A*C))/2*A
      6                 RETURN
      7           100   WRITE(6,9000)
      8                 RETURN
      9           9000  FORMAT(' INVALID PARAMETERS -- REQUIRES SQ.RT OF NEGATIVE')
     10                 END

--VARIABLE MAP--(LO=A/R)                                                    A=ARGLIST, C=CTRL OF DO, I=DATA INIT,
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE---REFERENCES-   R=READ, S=STORE, U=I/O UNIT, W=WRITE

 A          0B  /PARAMS/                  REAL                  2     3     4/A   4     5/A   5
 B          1B  /PARAMS/                  REAL                  2     3     3     4     4     4/A   5     5     5/A
 C          2B  /PARAMS/                  REAL                  2     3     4/A   5/A
 D          3B  /PARAMS/                  REAL                  2
 X1         4B  /PARAMS/                  REAL                  2     4/S
 X2         5B  /PARAMS/                  REAL                  2     5/S

--PROCEDURES--(LO=A/R)                                                     D=DEF LINE OF STMT FUNC
 -NAME----TYPE------ARGS----CLASS-----REFERENCES-                          A=ACTUAL ARGUMENT

 SQPT    REAL          1    FUNCTION      5
 SQRT    GENERIC       1    INTRINSIC     4

--STATEMENT LABELS--(LO=A/R)                                               A=ASSIGN STMT, D=DO STMT,
 -LABEL--ADDRESS---PROPERTIES-----DEF--REFERENCES-                         R=READ, W=WRITE

   100       0B                     7      3
  9000       4B   FORMAT            9      7/W

--ENTRY POINTS--(LO=A/R)                                                   D=DEFINITION
 -NAME---ADDRESS--ARGS------REFERENCES-

 QUAD        0B     0          1/D

--I/O UNITS--(LO=A/R)                                                      R=READ, W=WRITE
 -NAME---PROPERTIES------------REFERENCES-

 TAPE6   FMT/SEQ                    7/W

1 ....   SUBROUTINE QUAD        73/176  OPT=0                          FTN 5.1+528        81/10/12. 16.30.18        PAGE      2

--STATISTICS--

 PROGRAM-UNIT LENGTH              34B =  28
 SCM LABELLED COMMON LENGTH        6B =  6
 SCM STORAGE USED               60400B =  24832
 COMPILE TIME                   0.152 SECONDS
   0.300 CP SECONDS COMPILATION TIME.
```

Code of PROC LFIPR2D:

```
.PROC,LFIPR2D,SOURCE.
RETURN,SOURCE,LGO,ZZMLIST,LIST.
GET,SOURCE.
FTN5,I=SOURCE,L,LO,PW=132.
LDSET(MAP=BS/ZZMLIST)
LGO.
REVERT.CCL.
EXIT.
DAYFILE,,$$$BEGIN$.
REVERT,ABORT.
```

STUDENT PROJECT #6

In Student Project #1 we had a program called BINSCH which we converted from FORTRAN V4 to V5. Now take the converted BINSCH and change all the I/O from READ and WRITE to explicit CRM calls. Make TAPE10 into a BT=C, RT=F file with a record length of 30.

The instructor will supply a file called TAPE20 which is the same data as before but in RT=F format. Inside your program continue to call it TAPE10. This may be done by a JCL statement:

```
GET(TAPE10=TAPE20)
```

Do the required debugging as necessary. You may need to make some changes in your control cards depending on how you did the conversion.

A solution will be provided later.

OPTIONAL:

Place this newly modified program on the ULIB from the Optional Problem 2 and execute from it.

STUDENT PROJECT #7

In Student Project #2 we had a program named DEBUG which called a subroutine QUAD. Modify your corrected DEBUG so it will call a COBOL V5 subprogram call COBQUAD. Parameters are passed to COBQUAD in a COMMON block called CCOMMON in the same order as they were passed to QUAD.

COBQUAD is on STAFFAA. A listing also follows.

Note that COBQUAD in turn calls a FORTRAN V5 subroutine named ROOT. ROOT does the actual square root computations. The student is to write ROOT. ROOT uses parameters passing from COBQUAD. TEMP is the input to ROOT and is the value for which the square root will be determined. ROOT is the returned parameter. Both parameters will be REAL.

Write ROOT, modify DEBUG as required, execute, and debug if necessary. Use dumps, loader maps, etc., as required.

A solution will be provided later.

OPTIONAL:

Place these new modifications on your ULIB and execute from it.

OPTIONAL:

Modify BINSCH so that instead of using the binary search by SSN it is a linear search by name.

```
get,lf14o
/cobol5,i=lf14o,lo,b=0,al=t
```

```
CDC COBOL 5.3 - LEVEL 528        SOURCE LISTING OF COBQUAD      AOPT= 76/CDC/CDCS2          81/10/13. 16.29.31.      PAGE      1

 1              IDENTIFICATION DIVISION.
 2              PROGRAM-ID. COBQUAD.
 3             *
 4             *THIS PROGRAM IS THE COBOL5 SUBROUTINE TO BE CALLED BY THE STUDENT
 5             *WRITTEN PROGRAM FOR LABORATORY 5 OF THE FORTRAN5 ADVANCED WORKSHOP.
 6             *
 7              ENVIRONMENT DIVISION.
 8              DATA DIVISION.
 9              COMMON-STORAGE SECTION.
10              01  A                        COMP-2.
11              01  B                        COMP-2.
12              01  C                        COMP-2.
13              01  X1                       COMP-2.
14              01  X2                       COMP-2.
15              WORKING-STORAGE SECTION.
16              01  TEMP     COMP-2.
17              01  ROT      COMP-2.
18              PROCEDURE DIVISION.
19              P1.
20                  COMPUTE TEMP = (B ** 2 - 4 * A * C).
21                  ENTER FORTRAN-X "ROOT" USING TEMP ROT.
22                  COMPUTE X1 = ( - B + ROT) / 2.0 * A.
23                  COMPUTE X2 = ( - B - ROT) / 2.0 * A.
24              P2.
25                  EXIT PROGRAM.

         COLUMN   1         2         3         4         5         6         7         8
         12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
CDC COBOL 5.3 - LEVEL 528      CROSS REFERENCE FOR COBQUAD      AOPT= 76/CDC/CDCS2
  REFERENCED DATA-NAMES         LINE COLUMN            REFERENCE(S)

A                               10    12             20    22    23
B                               11    12             20    22    23
C                               12    12             20
ROT                             17    12             21    22    23
TEMP                            16    12             20    21
X1                              13    12             22
X2                              14    12             23

  REFERENCED DATA-NAMES         LINE COLUMN            REFERENCE(S)
```

```
CDC COBOL 5.3 - LEVEL 528      CROSS REFERENCE FOR COBQUAD      AOPT= 76/CDC/CDCS2          81/10/13. 16.29.31.      PAGE      2
  UNREFERENCED DATA-NAMES       LINE COLUMN
```

```
  REFERENCED PROCEDURE-NAMES   LINE COLUMN          REFERENCE(8)


CDC COBOL 5.3 - LEVEL 528     CROSS REFERENCE FOR COBQUAD     AOPT= 76/CDC/CDCS2
  UNREFERENCED PROCEDURE-NAMES LINE COLUMN

P1                             19    8
P2                             24    8

  UNREFERENCED PROCEDURE-NAMES LINE COLUMN


CDC COBOL 5.3 - LEVEL 528                  MAP OF COBQUAD     AOPT= 76/CDC/CDCS2

                                      *** DATA MAP (ADDR/BCP IN OCTAL, SZ IN DECIMAL) ***

                             COMMON-STORAGE SECTION
   *  01  A                          BLOCK=/CCOMMON/ ADDR/BCP=000000/00 SZ=10       COMP-2                LNR=10
   *  01  B                                                 000001/00    10       COMP-2                    11
   *  01  C                                                 000002/00    10       COMP-2                    12
   *  01  X1                                                000003/00    10       COMP-2                    13
   *  01  X2                                                000004/00    10       COMP-2                    14
                             WORKING-STORAGE SECTION
   *  01  TEMP                                PROGRAM       000073/00    10       COMP-2                    16
   *  01  ROT                                               000074/00    10       COMP-2                    17


                                      *** PROCEDURE MAP (ADDR IN OCTAL) ***
                  P1                                  ADDR=000004                                          LNR=19
                  P2                                        000044                                             24


                                                   *** END MAP ***
066000B CM,    1.542 CPS, 000000B ECS
```

STUDENT PROJECT #8

Using BINSCH as modified by Project #3 (and perhaps by #4) modify it to eliminate all debug codes.

Now remove all normal arrays (not FITs) and substitute character strings. This will require other coding changes -- how many depends upon how you do the changes above.

Also remove the code for the binary search (or linear search) and use INDEX instead.

Suggestion: Do each of these steps one at a time. Don't start on the next until the previous works correctly. This way if you inadvertently insert a bug you will know which portion caused the bug, and it will be easier to correct.

A solution will be provided separately.

OPTIONAL:

Add this code to your ULIB and execute from it.

APPENDIX B

STUDENT PROBLEM SOLUTIONS

A SOLUTION TO PROBLEM #1

```
 xedit,list
 XEDIT 3.1.00
?? p59
1 FTN 5.1+587          84/08/30. 16.37.32 PAGE      1
     PROGRAM BINSCH   74/176  OPT=1,ROUND= A/ S/ M/-D,-DS
          DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER
          /-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,PL=5000
          FTN5,I=BINSCH1,L=LIST,LO,PW=85.

        1          PROGRAM BINSCH(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT,TAPE10)
        2          CHARACTER NAMDAT*9,NAME*29,STARS*10,SSN*9
        3          CHARACTER NTAB(100)*30,NTABB*3000
        4          EQUIVALENCE (NTAB,NTABB)
        5          DATA NAMDAT /' NAME IS '/
        6          DATA STARS /'**********'/
        7          DATA I /1/
        8 100      READ(10,1000,END=110) NTAB(I)
        9          I=I+1
       10          IF(I.GT.100) GO TO 180
       11          GO TO 100
       12 110      IF(I.EQ.1)GO TO 170
       13          I=I+1
       14          DO 120 J=I,100
       15 120      NTAB(J)(1:10)=STARS
       16 130      READ(5,1003,END=150)SSN
       17          I=INDEX(NTABB,SSN)
       18          IF(I.EQ.0)GO TO 160
       19          I=((I-1)/29)+1
       20          NAME=NAMDAT//NTAB(I)(11:30)
       21 140      WRITE(6,1000)NAME
       22          GO TO 130
       23 150      STOP
       24 160      NAME=STARS//STARS
       25          GO TO 140
       26 170      WRITE(6,1001)
       27          STOP
       28 180      WRITE(5,1002)
       29          STOP
       30 1000     FORMAT(A29)
       31 1001     FORMAT('NO DATA ON TAPE10')
       32 1002     FORMAT('OVER 100 RECORDS ON TAPE10')
       33 1003     FORMAT(A9)
       34          END

 --VARIABLE MAP--(LO=A/R)
  -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE---REFERENCES-

   I          666B                              INTEGER                  7/I     8
                                                                         9       9/S
                                                                        10      12
                                                                        13      13/S
                                                                        14/C    17/S
                                                                        18      19
                                                                        19/S    20
   J          667B                              INTEGER                 14/C    15
   NAMDAT     204B                              CHAR*9                   2       5/I
                                                                        20
   NAME       205B                              CHAR*29                  2      20/S
                                                                        21/W    24/S
   NTAB       212B            EQV               CHAR*30        100       3       4
??
```

```
                                                                                  8/R    15/S
1 FTN 5.1+587          84/08/30. 16.37.32 PAGE      2
     PROGRAM BINSCH    74/176  OPT=1,ROUND= A/ S/ M/-D,-DS

                                                                                 20
   NTABB       212B              EQV                 CHAR*3000                   3       4
                                                                                17/A
   SSN         211B                                  CHAR*9                      2      16/R
                                                                                17/A
   STARS       210B                                  CHAR*10                     2       6/I
                                                                                15      24
                                                                                24


 --PROCEDURES--(LO=A/R)
  -NAME----TYPE------ARGS----CLASS-----REFERENCES-

   INDEX   INTEGER       2    INTRINSIC    17


 --STATEMENT LABELS--(LO=A/R)
  -LABEL--ADDRESS---PROPERTIES-----DEF--REFERENCES-

     100      22B                      8      8/L   11
     110      34B                     12      8/R   12/L
     120   INACTIVE DO-TERM           15     14/D   15/L
     130      50B                     16     16/L   22
     140      64B                     21     21/L   25
     150      67B                     23     16/R   23/L
     160      70B                     24     18     24/L
     170      73B                     26     12     26/L
     180      76B                     28     10     28/L
    1000     112B   FORMAT            30      8/R   21/W   30/L
    1001     114B   FORMAT            31     26/W   31/L
    1002     120B   FORMAT            32     28/W   32/L
    1003     125B   FORMAT            33     16/R   33/L


 --ENTRY POINTS--(LO=A/R)
  -NAME---ADDRESS--ARGS------REFERENCES-

   BINSCH     21B      0          1/D


 --I/O UNITS--(LO=A/R)
  -NAME--- PROPERTIES-------------REFERENCES-

   TAPE10  FMT/SEQ                     8/R
   TAPE5   FMT/SEQ                    16/R    28/W
   TAPE6   FMT/SEQ                    21/W    26/W


 --STATISTICS--

   PROGRAM-UNIT LENGTH              670B =  440
   SCM STORAGE USED               61000B =  25088
   COMPILE TIME                   0.510 SECONDS
  END OF FILE
??
   end
LIST    IS A LOCAL FILE
AEB :       0.433UNTS.
/
```

20 - 1
19/24

```
 rewind,*
   8 FILE(S) PROCESSED.
/lgo
? 333333333
 NAME IS LEE, ROBERT EDWARD
? 777777777
 NAME IS HOOVER, HERBERT
? 999999999
 NAME IS GRANT, ULYSES S.
? 000000000
 NAME IS WASHINGTON, GEORGE
? 123456789
********************
?
     0.143 CP SECONDS EXECUTION TIME.
/


 rewind,*
   8 FILE(S) PROCESSED.
/copy,tape10
000000000 WASHINGTON, GEORGE
111111111 LINCOLN, ABRAHAM
222222222 JACKSON, THOMAS
333333333 LEE, ROBERT EDWARD
444444444 ROOSEVELT, THEODORE
555555555 GARFIELD, JAMES ABRAM
666666666 MONROE, JAMES
777777777 HOOVER, HERBERT
888888888 WILSON, WOODROW
999999999 GRANT, ULYSES S.
 EOI ENCOUNTERED.
/
```

A SOLUTION TO PROJECT #2

```
PROGRAM DEBUG      73/176  OPT=0                                  FTN 5.1+528        81/10/13. 16.18.18        PAGE     1

 1                 PROGRAM DEBUG(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT)
 2                 COMMON /PARAMS/ A,B,C,X1,X2
 3          100    READ(5,9000,END=110) A,B,C
 4                 CALL QUAD
 5                 WRITE(6,9001) X1,X2
 6                 GO TO 100
 7          110    WRITE(6,9002)
 8                 STOP
 9          9000   FORMAT(3F10.4)
10          9001   FORMAT(' X1 = ',F10.4,' X2 = ',F10.4)
11          9002   FORMAT(' END OF JOB')
12                 END

--VARIABLE MAP--(LO=A/R)                                                    A=ARGLIST, C=CTRL OF DO, I=DATA INIT,
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE---REFERENCES-  R=READ, S=STORE, U=I/O UNIT, W=WRITE

  A          0B  /PARAMS/                    REAL                  2      3/R
  B          1B  /PARAMS/                    REAL                  2      3/R
  C          2B  /PARAMS/                    REAL                  2      3/R
  X1         3B  /PARAMS/                    REAL                  2      5/W
  X2         4B  /PARAMS/                    REAL                  2      5/W

--PROCEDURES--(LO=A/R)                                                     D=DEF LINE OF STMT FUNC
 -NAME----TYPE------ARGS----CLASS-----REFERENCES-                          A=ACTUAL ARGUMENT

  QUAD                 0    SUBROUTINE    4

--STATEMENT LABELS--(LO=A/R)                                               A=ASSIGN STMT, D=DO STMT,
 -LABEL--ADDRESS---PROPERTIES-----DEF--REFERENCES-                         R=READ, W=WRITE

    100       0B                    3      6
    110       0B                    7      3/R
   9000       4B   FORMAT           9      3/R
   9001       6B   FORMAT          10      5/W
   9002      13B   FORMAT          11      7/W

--ENTRY POINTS--(LO=A/R)                                                   D=DEFINITION
 -NAME---ADDRESS--ARGS------REFERENCES-

  DEBUG       0B     0          1/D

--I/O UNITS--(LO=A/R)                                                      R=READ, W=WRITE
 -NAME--- PROPERTIES-------------REFERENCES-

  TAPE5   FMT/SEQ                    3/R
  TAPE6   FMT/SEQ                    5/W    7/W
1           PROGRAM DEBUG      73/176  OPT=0                              FTN 5.1+528        81/10/13. 16.18.18        PAGE     2

--STATISTICS--

  PROGRAM-UNIT LENGTH            54B =  44
  SUM LABELLED COMMON LENGTH      5B =  5
  SUM STORAGE USED             60400B =  24832
```

```
 LOAD  JE TIME                    0.171 SECONDS
1          SUBROUTINE QUAD        73/176  OPT=0                          FTN 5.1+528        81/10/13. 16.18.18        PAGE     1

        1                  SUBROUTINE QUAD
        2                  COMMON /PARAMS/A,B,C,X1,X2
        3                  IF((B**2.0).LT.(4.0*A*C)) GO TO 100
        4                  X1=(-B+SQRT((B**2.0)-(4.0*A*C)))/2.0*A
        5                  X2=(-B-SQRT((B**2.0)-(4.0*A*C)))/2.0*A
        6                  RETURN
        7            100   WRITE(6,9000)
        8                  RETURN
        9           9000   FORMAT(' INVALID PARAMETERS -- REQUIRES SQ.RT OF NEGATIVE')
       10                  END

 --VARIABLE MAP--(LO=A/R)                                                          A=ARGLIST, C=CTRL OF DO, I=DATA INIT,
  -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE---REFERENCES-      R=READ, S=STORE, U=I/O UNIT, W=WRITE

   A           0B  /PARAMS/                    REAL                  2     3     4     4     5     5
 5/AB          1B  /PARAMS/                    REAL                  2     3     3     4     4     4/A   4     5     5
                                                                     5
   C           2B  /PARAMS/                    REAL                  2     3     4     5
   X1          3B  /PARAMS/                    REAL                  2     4/S
   X2          4B  /PARAMS/                    REAL                  2     5/S

 --PROCEDURES--(LO=A/R)                                                           D=DEF LINE OF STMT FUNC
  -NAME----TYPE------ARGS----CLASS-----REFERENCES-                                A=ACTUAL ARGUMENT

   SQRT    GENERIC      1    INTRINSIC     4     5

 --STATEMENT LABELS--(LO=A/R)                                                     A=ASSIGN STMT, D=DO STMT,
  -LABEL--ADDRESS---PROPERTIES-----DEF--REFERENCES-                               R=READ, W=WRITE

     100       0B                     7     3
    9000       4B   FORMAT            9     7/W

 --ENTRY POINTS--(LO=A/R)                                                         D=DEFINITION
  -NAME---ADDRESS--ARGS------REFERENCES-

   QUAD        0B    0          1/D

 --I/O UNITS--(LO=A/R)                                                            R=READ, W=WRITE
  -NAME--- PROPERTIES-------------REFERENCES-

   TAPE6   FMT/SEQ                     7/W

 --STATISTICS--

   PROGRAM-UNIT LENGTH              32B =  26
   SCM LABELLED COMMON LENGTH        5B =  5
   SCM STORAGE USED              60400B =  24832
   COMPILE TIME                  0.174 SECONDS
     0.354 CP SECONDS COMPILATION TIME.
```

A SOLUTION TO PROJECT #3

```
 xedit,list
 XEDIT 3.1.00
?? p55
1 FTN 5.1+587          84/09/03. 10.58.20 PAGE      1
     PROGRAM BINSCH   74/176  OPT=1,ROUND= A/ S/ M/-D,-DS
          DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER
          /-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,PL=5000
          FTN5,I=BINSCH3,LO,L=LIST,PW=85,EL=F.

        1        PROGRAM BINSCH
        2        COMMON /FITS/ FIT5(35),FIT6(35),FIT10(35),FIT20(35)
        3        CHARACTER NAMDAT*9,NAME*29,STARS*10,SSN*9
        4        CHARACTER WSA*30,DIAG1*17,DIAG2*23
        5        DATA NAMDAT /' NAME IS '/
        6        DATA STARS /'**********'/
        7        DATA DIAG1 /'NO DATA ON TAPE10'/
        8        DATA DIAG2 /'SSN NOT FOUND ON TAPE10'/
        9        DATA I /1/
       10        CALL FILESQ(FIT5,'LFN','INPUT','BT','C','RT','Z','FL',80,
       11       1     'EFC',3,'BFS',129)
       12        CALL FILESQ(FIT6,'LFN','OUTPUT','BT','C','RT','Z','FL',80,
       13       1     'EFC',3,'BFS',129)
       14        CALL FILESQ(FIT10,'LFN','TAPE10','BT','C','RT','F','FL',30,
       15       1     'EFC',3,'BFS',129)
       16        CALL FILEIS(FIT20,'LFN','TAPE20','FO','IS','BT','C','RT','Z',
       17       1     'FL',40,'EFC',3,'EMK','YES','ORG','NEW','KT','S','KL',9,
       18       2     'RKW',0,'RKP',0,'KA',SSN,'KP',0,'WSA',WSA)
       19        CALL OPENM(FIT10,'INPUT')
       20        CALL OPENM(FIT6,'OUTPUT')
       21        CALL OPENM(FIT20,'NEW')
       22 100    CALL GET(FIT10,WSA,30)
       23        IF(IFETCH(FIT10,'FP').EQ.O"10") GO TO 100
       24        IF(IFETCH(FIT10,'FP').GE.O"40") GO TO 110
       25        I=I+1
       26        CALL PUT(FIT20,WSA,30,WSA)
       27        GO TO 100
       28 110    IF(I.EQ.1) GO TO 170
       29        CALL CLOSEM(FIT10)
       30        CALL CLOSEM(FIT20)
       31        CALL OPENM(FIT20,'I-O')
       32        CALL OPENM(FIT5,'INPUT')
       33 130    CALL GET(FIT5,SSN,9)
       34        IF(IFETCH(FIT5,'FP').GE.O"40") GO TO 150
       35        CALL GET(FIT20,WSA,SSN)
       36        IF(IFETCH(FIT20,'FP').GE.O"40") THEN
 1     37            CALL PUT(FIT6,DIAG2,23)
 1     38            GO TO 150
 1     39        ENDIF
       40        IF(SSN.NE.WSA(1:9)) GO TO 160
       41        NAME=NAMDAT//WSA(11:30)
       42 140    CALL PUT(FIT6,NAME,29)
       43        GO TO 130
       44 150    CALL CLOSEM(FIT20)
       45        CALL CLOSEM(FIT5)
       46        CALL CLOSEM(FIT6)
       47        STOP
       48 160    NAME=NAMDAT//STARS//STARS
??
```

```
      49        GO TO 140
      50 170    CALL PUT(FIT6,DIAG1,17)
      51        GO TO 150
      52        END
1 FTN 5.1+587        84/09/03. 10.58.20 PAGE     2
     PROGRAM BINSCH   74/176  OPT=1,ROUND= A/ S/ M/-D,-DS

 --VARIABLE MAP--(LO=A/R)
  -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE---REFERENCES-

   DIAG1       417B                             CHAR*17                  4       7/I
                                                                       50/A
   DIAG2       421B                             CHAR*23                  4       8/I
                                                                       37/A
   FIT10       106B  /FITS/                     REAL            35       2      14/A
                                                                       19/A   22/A
                                                                       23/A   24/A
                                                                       29/A
   FIT20       151B  /FITS/                     REAL            35       2      16/A
                                                                       21/A   26/A
                                                                       30/A   31/A
                                                                       35/A   36/A
                                                                       44/A
   FIT5          0B  /FITS/                     REAL            35       2      10/A
                                                                       32/A   33/A
                                                                       34/A   45/A
   FIT6         43B  /FITS/                     REAL            35       2      12/A
                                                                       20/A   37/A
                                                                       42/A   46/A
                                                                       50/A
   I           424B                             INTEGER                  9/I    25
                                                                       25/S   28
   NAMDAT      406B                             CHAR*9                   3       5/I
                                                                       41     48
   NAME        407B                             CHAR*29                  3      41/S
                                                                       42/A   48/S
   SSN         413B                             CHAR*9                   3      16/A
                                                                       33/A   35/A
                                                                       40
   STARS       412B                             CHAR*10                  3       6/I
                                                                       48     48
   WSA         414B                             CHAR*30                  4      16/A
                                                                       22/A   26/A
                                                                       26/A   35/A
                                                                       40     41


 --PROCEDURES--(LO=A/R)
  -NAME----TYPE------ARGS----CLASS-----REFERENCES-

   CLOSEM                 1    SUBROUTINE   29     30     44     45     46
   FILEIS                31    SUBROUTINE   16
   FILESQ                13    SUBROUTINE   10     12     14
   GET                    3    SUBROUTINE   22     33     35
   IFETCH   INTEGER       2    FUNCTION     23     24     34     36
??
```

```
  OPENM                    2    SUBROUTINE   19     20     21     31     32
  PUT                      4    SUBROUTINE   26     37     42     50
1 FTN 5.1+587          84/09/03. 10.58.20 PAGE     3
     PROGRAM BINSCH   74/176  OPT=1,ROUND= A/ S/ M/-D,-DS

 --STATEMENT LABELS--(LO=A/R)
  -LABEL--ADDRESS---PROPERTIES-----DEF--REFERENCES-

     100      23B                    22     22/L   23     27
     110      40B                    28     24     28/L
     130      52B                    33     33/L   43
     140      74B                    42     42/L   49
     150      77B                    44     34     38     44/L   51
     160     106B                    48     40     48/L
     170     111B                    50     28     50/L


 --ENTRY POINTS--(LO=A/R)
  -NAME---ADDRESS--ARGS------REFERENCES-

  BINSCH      4B     0          1/D


 --STATISTICS--

  PROGRAM-UNIT LENGTH            425B =  277
  SCM LABELLED COMMON LENGTH     214B =  140
  SCM STORAGE USED             61000B =  25088
  COMPILE TIME                 0.811 SECONDS
 END OF FILE
??
```

SAMPLE EXECUTION OF PROJECT #3

```
 lgo
? 333333333
 NAME IS LEE, ROBERT EDWARD
? 777777777
 NAME IS HOOVER, HERBERT
? 000000000
 NAME IS WASHINGTON, GEORGE
? 002266880
 NAME IS ********************
? 999999999
 NAME IS GRANT, ULYSES S.
?
     0.530 CP SECONDS EXECUTION TIME.
/
```

CREMP OUTPUT FOR PROJECT #3

```
 rewind,*
  14 FILE(S) PROCESSED.
/copy,crmlist
1         CRMEP(LO,L=CRMLIST)
 RM NOTE  1001 ON LFN TAPE20   FILE OPENED
 RM NOTE  1002 ON LFN TAPE20   FILE CLOSED
 RM NOTE  1003 ON LFN TAPE20   NUMBER OF INDEX LEVELS          0
 RM NOTE  1004 ON LFN TAPE20   ***NUMBER OF GETS THIS OPEN                    0
 RM NOTE  1005 ON LFN TAPE20   ***NUMBER OF PUTS THIS OPEN                   10
 RM NOTE  1006 ON LFN TAPE20   ***NUMBER OF REPLACES THIS OPEN                0
 RM NOTE  1007 ON LFN TAPE20   ***NUMBER OF DELETES THIS OPEN                 0
 RM NOTE  1033 ON LFN TAPE20   ***NUMBER OF GET NEXTS THIS OPEN               0
 RM NOTE  1010 ON LFN TAPE20       ***TOTAL DISKAREA***              640  WORDS
 RM NOTE  1001 ON LFN TAPE20   FILE OPENED
 RM ERROR 0445 ON LFN TAPE20   EXIT ADDRESS 005652 KEY NOT FOUND - FILE POSITION M
 RM NOTE  1002 ON LFN TAPE20   FILE CLOSED
 RM NOTE  1003 ON LFN TAPE20   NUMBER OF INDEX LEVELS          0
 RM NOTE  1004 ON LFN TAPE20   ***NUMBER OF GETS THIS OPEN                    4
 RM NOTE  1005 ON LFN TAPE20   ***NUMBER OF PUTS THIS OPEN                    0
 RM NOTE  1006 ON LFN TAPE20   ***NUMBER OF REPLACES THIS OPEN                0
 RM NOTE  1007 ON LFN TAPE20   ***NUMBER OF DELETES THIS OPEN                 0
 RM NOTE  1033 ON LFN TAPE20   ***NUMBER OF GET NEXTS THIS OPEN               0
 RM NOTE  1010 ON LFN TAPE20       ***TOTAL DISKAREA***              640  WORDS
 EOI ENCOUNTERED.
/
```

FLSTAT OUTPUT FOR PROJECT #3

```
 flstat,tape20
1STATISTICS FOR FILE TAPE20
-ORGANIZATION------- IS
 CREATION DATE------ 84/09/03.
 DATE OF LAST CLOSE- 84/09/03.
 TIME OF LAST CLOSE- 10.59.37.

 FILE IS NOT MIPPED
 COLLATION IS STANDARD

 PRIMARY KEY INFORMATION
   STARTING WORD POSITION ------ 0
   STARTING CHARACTER POSITION - 0
   TYPE -- COLLATED SYMBOLIC
   LENGTH IN CHARACTERS -------- 9

 MAXIMUM RECORD SIZE  40
 MINIMUM RECORD SIZE  9

 TOTAL TRANSACTIONS
   NUMBER OF PUTS ------ 10
   NUMBER OF GETS ------ 4
   NUMBER OF DELETES --- 0
   NUMBER OF REPLACES -- 0
   NUMBER OF GETNEXTS -- 0

 CIO CALLS FOR FILE
   NUMBER OF READS ----- 2
   NUMBER OF WRITES ---- 2
   NUMBER OF RECALLS --- 0
   NUMBER OF REWRITES -- 2

 NUMBER OF BLOCKS------- 1
 NUMBER OF EMPTY BLOCKS- 0
 BLOCK SIZE IN PRUS----- 8
 NUMBER OF DATA RECORDS- 10

 FILE LENGTH IN PRUS  10
 NUMBER OF INDEX LEVELS IN USE   0
FLSTAT,TAPE20.
/
```

A SOLUTION TO PROJECT #4

```
get,lf14.
/ftn5,i=lf14m,lo,el=t,pw=132,b=0
1            PROGRAM DEBUG       73/176  OPT=0                                  FTN 5.1+528        81/10/13. 16.34.46        PAGE      1

      1                 PROGRAM DEBUG(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT)
      2                 COMMON /CCOMMON/ A,B,C,X1,X2
      3           100   READ(5,9000,END=110) A,B,C
      4                 CALL CORQUAD
      5                 WRITE(6,9001) X1,X2
      6                 GO TO 100
      7           110   WRITE(6,9002)
      8                 STOP
      9           9000  FORMAT(3F10.4)
     10           9001  FORMAT(' X1 = ',F10.4,' X2 = ',F10.4)
     11           9002  FORMAT(' END OF JOB')
     12                 END

---VARIABLE MAP--(LO=A/R)                                                          A=ARGLIST, C=CTRL OF DO, I=DATA INIT,
 -NAME----ADDRESS--BLOCK------PROPERTIES--------TYPE----------SIZE---REFERENCES-   R=READ, S=STORE, U=I/O UNIT, W=WRITE

  A           0B  /CCOMMON/                    REAL                    2      3/R
  B           1B  /CCOMMON/                    REAL                    2      3/R
  C           2B  /CCOMMON/                    REAL                    2      3/R
  X1          3B  /CCOMMON/                    REAL                    2      5/W
  X2          4B  /CCOMMON/                    REAL                    2      5/W

---PROCEDURES--(LO=A/R)                                                           D=DEF LINE OF STMT FUNC
 -NAME-----TYPE-------ARGS----CLASS-----REFERENCES-                                A=ACTUAL ARGUMENT

  CORQUAD                0     SUBROUTINE     4

--STATEMENT LABELS--(LO=A/R)                                                       A=ASSIGN STMT, D=DO STMT,
 -LABEL--ADDRESS----PROPERTIES------DEF--REFERENCES-                               R=READ, W=WRITE

    100       0B                       3       6
    110       0B                       7       3/R
   9000       4B   FORMAT              9       3/R
   9001       6B   FORMAT             10       5/W
   9002      13B   FORMAT             11       7/W

---ENTRY POINTS--(LO=A/R)                                                          D=DEFINITION
 -NAME----ADDRESS--ARGS-------REFERENCES-

  DEBUG       0B      0           1/D

---I/O UNITS--(LO=A/R)                                                             R=READ, W=WRITE
 -NAME---- PROPERTIES--------------REFERENCES-

  TAPE5   FMT/SEQ                     3/R
  TAPE6   FMT/SEQ                     5/W     7/W
            PROGRAM DEBUG       73/176  OPT=0                                  FTN 5.1+528        81/10/13. 16.34.46

--STATISTICS--

  PROGRAM-UNIT LENGTH              540 =  44
  SCM LABELLED COMMON LENGTH        50 =   5
```

```
get,lf14f
/ftn5,i=ftn   lf14f,lo,pw=132,b=0,el=t
1          SUBROUTINE ROOT        73/176  OPT=0                                FTN 5.1+528        81/10/13. 16.33.09        PAGE      1

        1                  SUBROUTINE ROOT (TEMP, ROT)
        2                  ROT=SQRT(TEMP)
        3                  RETURN
        4                  END

 --VARIABLE MAP--(LO=A/R)                                                                    A=ARGLIST, C=CTRL OF DO, I=DATA INIT,
  -NAME----ADDRESS--BLOCK------PROPERTIES--------TYPE----------SIZE---REFERENCES-            R=READ, S=STORE, U=I/O UNIT, W=WRITE

   ROT          2   DUMMY-ARG                    REAL                    1/A    2/S
   TEMP         1   DUMMY-ARG                    REAL                    1/A    2/A

 --PROCEDURES--(LO=A/R)                                                                      D=DEF LINE OF STMT FUNC
  -NAME-----TYPE-------ARGS-----CLASS------REFERENCES-                                      A=ACTUAL ARGUMENT

   SQRT     GENERIC       1     INTRINSIC      2

 --ENTRY POINTS--(LO=A/R)                                                                    D=DEFINITION
  -NAME----ADDRESS--ARGS-------REFERENCES-

   ROOT         0B      2           1/D

 --STATISTICS--

   PROGRAM-UNIT LENGTH              20B =  16
   SCM STORAGE USED               60400B =  24832
   COMPILE TIME                   0.057 SECONDS
     0.063 CP SECONDS COMPILATION TIME.
```

A SOLUTION TO PROJECT #5

```
 xedit,. list
 XEDIT 3.1.00
?? p54
1 FTN 5.1+587          84/09/04. 08.35.01 PAGE      1
     PROGRAM BINSCH   74/176  OPT=1,ROUND= A/ S/ M/-D,-DS
          DO=-LONG/-OT,ARG=-COMMON/-FIXED,CS= USER
          /-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,PL=5000
          FTN5,I=BINSCH2,L=LIST,LO,PW=85.

        1        PROGRAM BINSCH(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT,TAPE10)
        2        CHARACTER NAMDAT*9,NAME*29,STARS*10,SSN*10
        3        CHARACTER NTAB(100)*30,NTABB*3000
        4        EQUIVALENCE (NTAB,NTABB)
        5        DATA NAMDAT /' NAME IS '/
        6        DATA STARS /'**********'/
        7        DATA I /1/
        8 100    ICOMMA=0
        9        INUMBER=0
       10        ISWITCH=0
       11        IHYPHEN=0
       12        IPERIOD=0
       13        READ(10,1000,END=110) NTAB(I)
       14        IF(NTAB(I)(21:21).EQ.' ') THEN
  1    15            PRINT*,'*** LEADING BLANK IN TAPE10 ***'
  1    16            PRINT*,'*** ',NTAB(I)
  1    17            GO TO 100
  1    18        ENDIF
       19        DO 114 J=11,30
       20        IF(NTAB(I)(J:J).LT.'A' .OR. NTAB(I)(J:J).GT.'Z') GO TO 200
       21 114    CONTINUE
       22        IF(INUMBER.GT.1) THEN
  1    23            PRINT*,'*** TOO MANY DIGITS IN TAPE10 NAME'
  1    24            PRINT*,NTAB(I)
  1    25            INUMBER=0
  1    26            GO TO 100
  1    27        ENDIF
       28        IF(IHYPHEN.GT.1) THEN
  1    29            PRINT*,'*** TOO MANY HYPHENS IN NAME ***'
  1    30            PRINT*,NTAB(I)
  1    31            IHYPHEN=0
  1    32            GO TO 100
  1    33        ENDIF
       34        IF(ICOMMA.GT.1) THEN
  1    35            PRINT*,'*** TOO MANY COMMAS IN NAME'
  1    36            PRINT*,NTAB(I)
  1    37            ICOMMA=0
  1    38            GO TO 100
  1    39        ENDIF
       40        IF(IPERIOD.GT.1) THEN
  1    41            PRINT*,'*** TOO MANY PERIODS IN NAME'
  1    42            PRINT*,NTAB(I)
  1    43            IPERIOD=0
  1    44            GO TO 100
  1    45        ENDIF
       46        IF(ISWITCH.GT.1) THEN
  1    47        PRINT*,'*** BAD TAPE10 INPUT ***'
??
```

```
 1     48            PRINT*,'*** ',NTAB(I)(21:30),' ***'
 1     49            ISWITCH=0
 1     50            GO TO 100
 1     51        ENDIF
       52        I=I+1
       53        IF(I.GT.100) GO TO 180
1 FTN 5.1+587         84/09/04. 08.35.01 PAGE      2
     PROGRAM BINSCH   74/176  OPT=1,ROUND= A/ S/ M/-D,-DS

       54        GO TO 100
       55 110    IF(I.EQ.1)GO TO 170
       56        I=I+1
       57        DO 120 J=I,100
       58 120    NTAB(J)(1:10)=STARS
       59 130    READ(5,1003,END=150)SSN
       60        IF(SSN(10:10).NE.' ') THEN
 1     61            SSN(10:10)=' '
 1     62            PRINT*,'SSN CONTAINS MORE THAN 9 DIGITS'
 1     63            GO TO 130
 1     64        ENDIF
       65        DO 135 J=1,9
       66        IF(SSN(J:J).LT.'0'.OR.SSN(J:J).GT.'9') THEN
 1     67            PRINT*,'DIGIT ',J,' OF SSN IS NOT A DIGIT, BUT A "',
 1     68      1         SSN(J:J),'"'
 1     69            ISWITCH=1
 1     70        ENDIF
       71 135    CONTINUE
       72        IF(ISWITCH.NE.0) THEN
 1     73            ISWITCH=0
 1     74            GO TO 130
 1     75        ENDIF
       76        I=INDEX(NTABB,SSN)
       77        IF(I.EQ.0)GO TO 160
       78        I=((I-1)/29)+1
       79        NAME=NAMDAT//NTAB(I)(11:30)
       80 140    WRITE(6,1000)NAME
       81        GO TO 130
       82 150    STOP
       83 160    NAME=STARS//STARS
       84        GO TO 140
       85 170    WRITE(6,1001)
       86        STOP
       87 180    WRITE(5,1002)
       88        STOP
       89 200    IF(NTAB(I)(J:J).EQ.',') THEN
 1     90            ICOMMA=ICOMMA+1
 1     91            GO TO 114
 1     92        ENDIF
       93        IF(NTAB(I)(J:J).EQ.'.') THEN
 1     94            IPERIOD=IPERIOD+1
 1     95            GO TO 114
 1     96        ENDIF
       97        IF(NTAB(I)(J:J).EQ.'-') THEN
 1     98            IHYPHEN=IHYPHEN+1
??
```

```
 1    99            GO TO 114
 1   100        ENDIF
     101        IF(NTAB(I)(J:J).LT.'9'  .AND. NTAB(I)(J:J).GT.'0')THEN
 1   102            INUMBER=INUMBER+1
 1   103            GO TO 114
 1   104        ENDIF
     105        IF(NTAB(I)(J:J).EQ.' ') GO TO 114
     106        PRINT*,'*** INVALID CHARACTER IN TAPE10'
     107        PRINT*, '*** ',NTAB(I)
     108        GO TO 114
     109 1000   FORMAT(A29)
     110 1001   FORMAT('NO DATA ON TAPE10')
1 FTN 5.1+587         84/09/04. 08.35.01 PAGE      3
    PROGRAM BINSCH   74/176  OPT=1,ROUND= A/ S/ M/-D,-DS

     111 1002   FORMAT('OVER 100 RECORDS ON TAPE10')
     112 1003   FORMAT(A10)
     113        END


 --VARIABLE MAP--(LO=A/R)
  -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE---REFERENCES-

   I          1352B                              INTEGER                 7/I    1
                                                                         14     1
                                                                         20     2
                                                                         24     3
                                                                         36     4
                                                                         48     5
                                                                         52/S   5
                                                                         55     5
                                                                         56/S   5
                                                                         76/S   7
                                                                         78     7
                                                                         79     8
                                                                         93     9
                                                                        101    10
                                                                        105    10
   ICOMMA     1353B                              INTEGER                 8/S    3
                                                                         37/S   9
                                                                         90/S
   IHYPHEN    1356B                              INTEGER                 11/S   2
                                                                         31/S   9
                                                                         98/S
   INUMBER    1354B                              INTEGER                 9/S    2
                                                                         25/S  10
                                                                        102/S
   IPERIOD    1357B                              INTEGER                 12/S   4
                                                                         43/S   9
                                                                         94/S
   ISWITCH    1355B                              INTEGER                 10/S   4
                                                                         49/S   6
                                                                         72     7
   J          1360B                              INTEGER                 19/C   2
??
```

DA3015

```
                                                                                  20      20
                                                                                  20      57/C
                                                                                  58      65/C
                                                                                  66      66
                                                                                  66      66
                                                                                  67/W    67
                                                                                  67      89
                                                                                  89      93
                                                                                  93      97
                                                                                  97     101
                                                                                 101     101
                                                                                 101     105
                                                                                 105
   NAMDAT      670B                                CHAR*9                          2       5/I
                                                                                  79
   NAME        671B                                CHAR*29                         2      79/S
                                                                                  80/W   83/S
   NTAB        676B              EQV               CHAR*30        100              3       4
1 FTN 5.1+587          84/09/04. 08.35.01 PAGE      4
     PROGRAM BINSCH    74/176  OPT=1,ROUND= A/ S/ M/-D,-DS

                                                                                  13/R    14
                                                                                  16/W    20
                                                                                  20      24/W
                                                                                  30/W    36/W
                                                                                  42/W    48/W
                                                                                  58/S    79
                                                                                  89      93
                                                                                  97     101
                                                                                 101     105
                                                                                 107/W
   NTABB       676B              EQV               CHAR*3000                       3       4
                                                                                  76/A
   SSN         675B                                CHAR*10                         2      59/R
                                                                                  60      61/S
                                                                                  66      66
                                                                                  67/W    76/A
   STARS       674B                                CHAR*10                         2       6/I
                                                                                  58      83
                                                                                  83


 --PROCEDURES--(LO=A/R)
  -NAME----TYPE------ARGS----CLASS-----REFERENCES-

   INDEX   INTEGER       2     INTRINSIC     76


 --STATEMENT LABELS--(LO=A/R)
  -LABEL--ADDRESS---PROPERTIES-----DEF--REFERENCES-

     100       22B                      8      8/L    17      26      32      38      44
                                                     50     54
     110      167B                     55     13/R   55/L
??
```

```
    114       64B   DO-TERM          21      19/D   21/L   91      95      99     103
                                                105    108
    120   INACTIVE DO-TERM           58      57/D   58/L
    130      203B                    59      59/L   63     74      81
    135   INACTIVE DO-TERM           71      65/D   71/L
    140      255B                    80      80/L   84
    150      260B                    82      59/R   82/L
    160      261B                    83      77     83/L
    170      264B                    85      55     85/L
    180      267B                    87      53     87/L
    200      272B                    89      20     89/L
   1000      437B   FORMAT          109      13/R   80/W  109/L
   1001      441B   FORMAT          110      85/W  110/L
   1002      445B   FORMAT          111      87/W  111/L
   1003      452B   FORMAT          112      59/R  112/L
1 FTN 5.1+587          84/09/04. 08.35.01 PAGE      5
     PROGRAM BINSCH    74/176  OPT=1,ROUND= A/ S/ M/-D,-DS

 --ENTRY POINTS--(LO=A/R)
  -NAME---ADDRESS--ARGS------REFERENCES-

   BINSCH     21B      0          1/D


 --I/O UNITS--(LO=A/R)
  -NAME--- PROPERTIES-------------REFERENCES-

   TAPE10  FMT/SEQ                    13/R
   TAPE5   FMT/SEQ                    59/R   87/W
   TAPE6   FMT/SEQ                    80/W   85/W


 --STATISTICS--

   PROGRAM-UNIT LENGTH            1361B =  753
   SCM STORAGE USED              61100B =  25152
   COMPILE TIME                  1.545 SECONDS
  END OF FILE
?? end
LIST    IS A LOCAL FILE
AEB ,      0.563UNTS.
/
```

Seminar Evaluation

Your feedback will be very useful in improving the quality of our seminars!

Seminar Title _____

Dates ___/___/___ to ___/___/___ Location _____

Instructor _____

	Strongly Disagree	Disagree	Not Sure	Agree	Strongly Agree
CONTENT					
1. The objectives were clearly stated at the beginning of the seminar. Please explain.	SD	D	NS	A	SA
2. All of the seminar objectives were adequately covered. Please explain.	SD	D	NS	A	SA
FACILITIES					
3. The seminar facilities were comfortable. Please explain.	SD	D	NS	A	SA
INSTRUCTOR					
4. The instructor demonstrated a thorough knowledge of the content. Please explain.	SD	D	NS	A	SA
5. The instructor presented the information in a clear and understandable manner. Please explain.	SD	D	NS	A	SA
PARTICIPANT					
6. I had the necessary prior knowledge for this seminar. Please explain.	SD	D	NS	A	SA

7. Indicate your knowledge level of the content prior to this seminar. (Circle one)

No knowledge ← 1 2 3 4 5 6 7 8 9 10 → Knew everything covered in seminar

8. Indicate your knowledge level of the content after this seminar. (Circle one)

No knowledge ← 1 2 3 4 5 6 7 8 9 10 → Knew everything covered in seminar

GENERAL COMMENTS

9. Considering all aspects of the seminar, how would you rate it overall? (Circle one)

Poor ← 1 2 3 4 5 6 7 8 9 10 → Excellent

10. What did you like best about the seminar?

_____

_____

_____

11. What did you like least about the seminar?

_____

_____

_____

12. Please use the space below to make any other general comments about the seminar.

_____

_____

_____

13. Please list colleagues who should receive advance notices of similar seminars.

Name _____ Name _____

Title _____ Title _____

Organization _____ Organization _____

Address _____ Address _____

_____

Phone No. _____ Phone No. _____

14. Would there be enough interested people in your organization to warrant offering the seminar at your site?

Yes ☐ No ☐

If yes, who should be contacted to manage it?

Name _____ Title _____

Organization _____ Phone No. _____

Address _____

Signature _____ Company _____

PARTICIPANT INFORMATION FORM

In order for our seminars/courses to be most effective, they need to take into account the characteristics, needs and objectives of the people who attend them. The information asked for below will assist us in keeping our presentations relevant to the participants and in developing and scheduling new presentations that will meet participant needs. Please complete this form and leave it with the presenter at the next break.

Seminar/Course Title _____ **Date of Presentation** _____

Name _____ **Field or Type of Business** _____

Title _____ **Years of Experience** _____

Business Address _____ **Supervisor's Title** _____

_____ **Last professional degree** _____

_____

List your three primary objectives in attending this seminar.

1. _____
2. _____
3. _____

Will this course/seminar be credited toward certification/training requirements? _____

Rank in order of importance in your choice of this seminar session.

Instructor _____ **Date** _____ **Location** _____ **Employer's Preference** _____

Previous courses/seminars attended relating to this topic.

1. _____
2. _____
3. _____

Topics for additional courses/seminars in which you would be interested.

1. _____
2. _____
3. _____

PARTICIPANT INFORMATION FORM

What trade journals/magazines do you regularly read or subscribe to in order to keep abreast in your profession?

1. _____
2. _____
3. _____

How did you become aware of this course/seminar?

Schedule/Catalogue _____,

Direct Mail Brochure _____,

Recommendations of Supervisor _____,

Recommendation of Colleague _____,

Corporate Training Department _____,

Other _____ .

COMMENT SHEET

MANUAL TITLE: CYBER ADVANCED FORTRAN WORKSHOP

PUBLICATION NO.: DA3015-1 REVISION: F

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ STATE: _____ ZIP CODE: _____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

CUT ALONG LINE

AA3419 REV. 4/79 PRINTED IN U.S.A.

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

National Coordinator
Bloomington Facility (MNA02B)
5001 West 80th Street
Bloomington, Minnesota 55437

CUT ALONG LINE

FOLD

FOLD

CONTROL DATA
BUSINESS ADVISORS

CORPORATE HEADQUARTERS
P.O. BOX 0
MINNEAPOLIS, MINNESOTA 55440