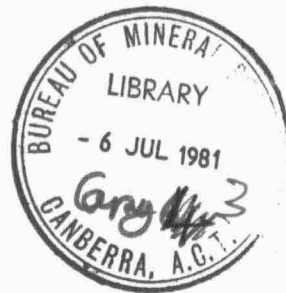


1981/9

~~11~~ C.03

084019

BMR PUBLICATIONS COMPACTUS  
(LENDING SECTION)



# BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS

## RECORD

RECORD 1981/9  
\*\*\*\*\*

*copy 3.*

LIBED  
\*\*\*\*\*

A RELOCATABLE LIBRARY MANAGEMENT PROGRAM  
\*\*\*\*\*

FOR HP-1000 COMPUTERS WITH  
\*\*\*\*\*

AN RTE OPERATING SYSTEM  
\*\*\*\*\*

by Frank Brassil

RECORD 1981/9  
\*\*\*\*\*

LIBED  
\*\*\*\*\*

A RELOCATABLE LIBRARY MANAGEMENT PROGRAM  
\*\*\*\*\*

FOR HP-1000 COMPUTERS WITH  
\*\*\*\*\*

AN RTE OPERATING SYSTEM  
\*\*\*\*\*

by Frank Brassil

# LIBED - RELOCATABLE LIBRARY MANAGER

## Table of Contents

|       |                                                |    |
|-------|------------------------------------------------|----|
| 1     | INTRODUCTION AND GENERAL DESCRIPTION . . . . . | 1  |
| 1.1   | Introduction . . . . .                         | 1  |
| 1.2   | Conventions . . . . .                          | 1  |
| 1.2.1 | Definitions . . . . .                          | 1  |
| 1.2.2 | File Names . . . . .                           | 2  |
| 1.2.3 | File Types . . . . .                           | 2  |
| 1.3   | HP 1000 RTE SYSTEM RELOCATABLE FILES . . . . . | 2  |
| 1.4   | COMPILER ERRORS . . . . .                      | 3  |
| 1.5   | LIMITATIONS . . . . .                          | 3  |
| 2     | Execution of LIBED . . . . .                   | 4  |
| 2.1   | Update Mode . . . . .                          | 4  |
| 2.2   | List Mode . . . . .                            | 5  |
| 2.3   | Command mode - interactive form . . . . .      | 5  |
| 2.4   | Command File Mode . . . . .                    | 5  |
| 2.5   | Error Handling . . . . .                       | 6  |
| 2.6   | LIBED Report . . . . .                         | 6  |
| 3     | LIBED COMMANDS . . . . .                       | 7  |
| 3.1   | ABORT . . . . .                                | 7  |
| 3.2   | ATTACH, RFILE . . . . .                        | 7  |
| 3.3   | COPY(, NAME1, NAME2) . . . . .                 | 7  |
| 3.4   | DELETE, NAME1(, NAME2....., NAMEn) . . . . .   | 8  |
| 3.5   | END . . . . .                                  | 8  |
| 3.6   | LIBRARY, NEWLB . . . . .                       | 8  |
| 3.7   | LL, LU . . . . .                               | 8  |
| 3.8   | TRANSFER(, NAMR) . . . . .                     | 9  |
| 3.9   | REWIND . . . . .                               | 9  |
| 4     | PROGRAM ORGANISATION . . . . .                 | 10 |
| 4.1   | UPDATE MODE . . . . .                          | 10 |
| 4.2   | LIST MODE . . . . .                            | 11 |
| 4.3   | COMMAND MODE . . . . .                         | 11 |
| A     | SAMPLE FMGR TRANSFER FILE                      |    |
| B     | SAMPLE PRINTOUTS                               |    |
| C     | FLOWCHARTS                                     |    |
| D     | PROGRAM DESCRIPTIONS                           |    |
| E     | PROGRAM SOURCE LISTINGS                        |    |

## INTRODUCTION AND GENERAL DESCRIPTION

## CHAPTER 1

## 1.1 Introduction

The HEWLETT-PACKARD HP1000 computer system running under RTE-IVB is a disc based general purpose mini-computer system which provides many features to make the task of the programmer easier. Programs may be written in FORTRAN or ASSEMBLER language and converted into relocatable code by the compiler or assembler. The program source code almost always exists as a disc file which is created and edited by the programmer. This disc file is used as input to the compiler or assembler and a relocatable file is produced. One or more relocatable files are input to the relocating loader and an executable program module is produced. When large numbers of subprograms are used it is convenient to collect them into a single file called a relocatable library. The relocating loader has the ability to scan a library file for those relocatable subprogram modules which it needs. The advantages of using libraries are that commonly used subprograms may be collected together and loaded in a single library search, load time is reduced because there are fewer files to handle, and disc space is conserved when many small files are grouped into a single larger file. LIBED is a program which enables the programmer to maintain relocatable subprogram libraries on the HP 1000 RTE system.

## 1.2 Conventions

The RTE operating system has a number of conventions which are adopted in this document, or used by LIBED.

## 1.2.1 Definitions

**SOURCE FILE** - A file containing program code used as input to a compiler or assembler.

**RELOCATABLE FILE** - A file produced by a compiler or assembler containing one or more relocatable program modules.

**LIBRARY FILE** - A file containing one or more relocatable subprogram modules.

**PROGRAM MODULE** - A unit of program code which is loaded by the relocating loader. In FORTRAN a program module is begun with a PROGRAM, SUBROUTINE, FUNCTION or BLOCK DATA statement and terminated by an END

9:33 AM TUE., 14 APR., 1981

## LIBED - RELOCATABLE LIBRARY MANAGER

statement. In assembler a program module is begun by a NAM pseudoinstruction and terminated by an END pseudoinstruction.

FMGR - The HP supplied disc file management package.

### 1.2.2 File Names

All file names referred to in this document adopt the convention used in FMGR, i.e. NAMR = INAME:ISC:ICR

where INAME is from one to six characters, the first character being non-numeric and printable; ISC is the security code and ICR is the cartridge number. If ISC is not provided, zero is used as default, and if ICR is not provided, the first file called INAME will be used.

Certain conventions have been adopted with regard to the first character of a file name and, while not essential, they are useful. Source files have an &, relocatable files have a % and library files have a \$.

### 1.2.3 File Types

A number of file types are defined in the HP system, and each type is designated by a number. In particular, an ASCII file such as a user would make at a terminal is a type 3 file, and relocatable files and library files are type 5 files. Any attempt to use other than a type 5 file as a relocatable or library file in LIBED will cause an error.

## 1.3 HP 1000 RTE SYSTEM RELOCATABLE FILES

The output from an HP compiler or assembler is a relocatable file containing relocatable records which the loader processes to produce an absolute program module which can be executed under the RTE operating system.

The NAM record is produced by the NAM pseudoinstruction in ASSEMBLER or by the PROGRAM, SUBROUTINE, FUNCTION or BLOCK DATA statements in FORTRAN. It contains the module name, type, priority, time schedule parameters and a comment field. Every NAM record signals the start of a relocatable program module.

The ENT record contains the symbolic names and relocatable addresses of entry points within a module and is generated by the ENT pseudoinstruction in ASSEMBLER and generated automatically by the FORTRAN compiler.

The EXT record contains a list of the symbolic names of entry points of other program modules required by the program. It is generated by the EXT pseudoinstruction in ASSEMBLER and generated automatically by the FORTRAN compiler.

DBL and EMA records are the other kinds of relocatable records and contain program code and data definition code.

The END record terminates a relocatable module.

9:33 AM TUE., 14 APR., 1981

## LIBED - RELOCATABLE LIBRARY MANAGER

There is no functional difference between a relocatable file and a relocatable library. The term "library" is used to imply that the file is intended to be maintained as a collection of relocatable program modules.

### 1.4 COMPILER ERRORS

LIBED will not allow subprogram modules containing compiler errors to be added to a library. When the compiler detects an error it generates an external reference called .BAD. and LIBED scans every EXT record for this reference, and aborts if it is found.

### 1.5 LIMITATIONS

The RTE-IV relocating loader will accept libraries generated and maintained by LIBED. However at present the loader does not permit the programmer to selectively relocate program modules (type 2, 3 or 4) from within a relocatable library. Thus there is little value in maintaining main program modules in libraries. If HP ever provide this facility (it exists in the on-line generator) then main program modules could also be saved in libraries.

## Execution of LIBED

## CHAPTER 2

LIBED enables the programmer to manage libraries of relocatable subprograms. There are three modes of execution of LIBED: UPDATE mode, LIST mode and COMMAND mode. The mode is determined by the way in which LIBED is scheduled.

UPDATE mode enables the programmer to replace a program module within a library with a new version of that module; or to add a new program module onto the end of a library. UPDATE mode requires no interaction with the user. LIBED is designed to run in update mode as part of a FMGR transfer file. An example of this is shown in Appendix A which shows a transfer file which will compile a user specified source file, make a temporary relocatable file and update a user specified library, and then purge the temporary file.

LIST mode enables the programmer to examine the contents of a relocatable library by producing a listing of all NAM, ENT and EXT records in a library.

COMMAND mode enables the programmer to modify, concatenate and reorganise libraries and to delete subprogram modules from libraries. LIBED is controlled by commands which may be input interactively through a terminal or the commands may be saved as a disc file and LIBED controlled from that file. The commands which are used to control LIBED are described in chapter 3.

## 2.1 Update Mode

LIBED is scheduled in update mode as follows:

```
*RU,LIBED,RFILE,OLDLB,NEWLB,LL
```

|       |                                         |
|-------|-----------------------------------------|
| RFILE | is the name of a relocatable file.      |
| OLDLB | is a relocatable library to be updated. |
| NEWLB | is the name of the output library.      |
| LL    | is the list logical unit.               |

LIBED will scan RFILE and enter every subprogram name into a table in memory. It will then copy OLDLB to NEWLB, replacing any subprogram whose name appears in the table with the version on RFILE. When the copy is complete, any subprogram in RFILE which was not in OLDLB will be added to the end of NEWLB. LL is the list output logical unit, which defaults to

## LIBED - RELOCATABLE LIBRARY MANAGER

the scheduling terminal. The time, date and output library name will automatically be inserted into the comment field of each subprogram as it is copied from RFILE. This will be displayed on the load map by the loader and provides the user with a check as to the source and version of a module being loaded.

If NEWLB is not given, then the output library will be called OLDLB, and the previous version of OLDLB will be purged.

### 2.2 List Mode

LIBED may be scheduled in list mode as follows:

```
*RU,LIBED,,LIBRY,,LL
```

This is interpreted as a request to list the contents of the relocatable file LIBRY. Each NAM, EXT and ENT record is listed showing subprogram name, type, priority, length, blank common requirements, comment field, external references and entry points. The listing is dated and paged. LL is the list logical unit. It defaults to 6 only when LIBED is scheduled this way.

### 2.3 Command mode - interactive form

LIBED can be scheduled interactively as follows:

```
*RU,LIBED
```

LIBED will accept commands directly from the terminal after issuing the prompt:

```
LIBED?
```

The commands available are described in chapter 3.

### 2.4 Command File Mode

LIBED may be scheduled in command file mode as follows:

9:33 AM TUE., 14 APR., 1981

## LIBED - RELOCATABLE LIBRARY MANAGER

\*RU,LIBED,AFILE

AFILE is a file containing LIBED commands. The commands are identical to those used in interactive mode except that no prompt is issued.

### 2.5 Error Handling

All errors are reported on the list device and should be self explanatory. Any error in execution will cause LIBED to abort except in interactive mode, when an error will cause the message:

LIBED: INPUT ERROR

to be printed on the terminal and the prompt to be reissued.

### 2.6 LIBED Report

In update or command mode LIBED produces a report on the scheduling terminal which gives a detailed description of the execution process. In update mode the report shows the the subprogram modules which were replaced or added during the update process. In interactive or command file mode, a list of every subprogram module transfer or deletion is given. The list mode printout has been described earlier. The report can be redirected to another logical unit at scheduling or by use of the appropriate command in command mode. An example of a printout in each mode is given in Appendix B.

## LIBED - RELOCATABLE LIBRARY MANAGER

|                |           |
|----------------|-----------|
| LIBED COMMANDS | CHAPTER 3 |
|----------------|-----------|

In this section the commands that control LIBED in COMMAND mode are described. All parameters must be provided unless they are enclosed in curly brackets: { }.

### 3.1 ABORT

This causes LIBED to close all input files, purge the output file and stop. This command should always be used to abort LIBED in the interactive mode.

### 3.2 ATTACH,RFILE

This command causes RFILE to be opened as an input relocatable library, and is available to future commands until another input library is opened.

### 3.3 COPY(,NAME1,NAME2)

This command causes relocatable subprogram modules to be copied from the input library to the output library. NAME1 and NAME2 are optional. They are the names of relocatable subprogram modules on the input library. The effect of giving or not giving NAME1 and NAME2 is shown in the table below. This command can be used with the REWIND command to reorder libraries.

LIBED - RELOCATABLE LIBRARY MANAGER

```
*****
*      *      *
* NAME1 * NAME2 * MODULES COPIED FROM INPUT TO OUTPUT LIBRARY *
*      *      *
*****
*      *      *
* NOT GIVEN * NOT GIVEN * FROM NEXT INPUT TO END OF FILE *
* NOT GIVEN * GIVEN * FROM NEXT INPUT TO NAME2 *
* GIVEN * NOT GIVEN * NAME1 *
* GIVEN * GIVEN * NAME1 TO NAME2 INCLUSIVE *
*      *      *
*****
```

3.4 DELETE,NAME1(,NAME2.....,NAMEn)

The named subprograms are entered into a table in memory. When a COPY command is executed the named subprograms will not be copied.

3.5 END

This command causes the input files to be closed and the output library to be catalogued. An end-of-file on a command file will have the same effect, but the command must be used to terminate processing in interactive mode.

3.6 LIBRARY,NEWLB

This command causes the output library to be named NEWLB. This command must be input before a COPY command is executed. NEWLB may have the same name as the input library, in which case the input library will be purged at the end of the processing.

3.7 LL,LU

This command assigns the list logical unit to LU. Listing can be suppress by assigning the list logical unit to a bit bucket.

## LIBED - RELOCATABLE LIBRARY MANAGER

### 3.8 TRANSFER(,NAMR)

This command transfers control to or from a terminal logical unit and a command file. Note that transfers cannot be nested. If NAMR is a file name then control is transferred to that file. If NAMR is a logical unit number, then control is transferred to that Logical unit. If NAMR is not provided, then the response depends upon whether the TRANSFER command was recieved from a logical unit or a file. If the command was recieved from a file, then control is transferred to the terminal at which LIBED was scheduled. If the command was recieved from a logical unit, then command is transferred to a command file which is already open.

### 3.9 REWIND

This command rewinds the input library file.

|                      |           |
|----------------------|-----------|
| PROGRAM ORGANISATION | CHAPTER 4 |
|----------------------|-----------|

LIBED is a FORTRAN program which consists of a main program, a number of subprograms and a table in memory. The main program gets the system time, sets logical units and picks up the schedule string. LIBED executes in 3 modes - UPDATE mode, LIST mode and COMMAND mode. The mode is determined by which parameters are present in the schedule string. LIBED determines the mode and schedules an appropriate subroutine. A flowchart is given in Appendix C. The schedule string consists of the information provided after the program name when the program is scheduled. It consists of three parameters separated by commas. These parameters are referred to as P1, P2 and P3 in this chapter.

LIBED is a FORTRAN program which consists of a main program, a number of subprograms and a table in memory. The main program gets the system time, sets logical units and picks up the schedule string. LIBED executes in 3 modes - UPDATE mode, LIST mode and COMMAND mode. The mode is determined by which parameters are present in the schedule string. LIBED determines the mode and schedules an appropriate subroutine. A flowchart is given in Appendix C. The schedule string consists of the information provided after the program name when the program is scheduled. It consists of three parameters separated by commas. These parameters are referred to as P1, P2 and P3 in this chapter.

LIBED opens the relocatable file named in P1 and the relocatable library file named in P2. LIBED then creates a scratch relocatable library file with the same security code and cardridge number as given in P3 (note that P3 defaults to P2). LIBED then scans the input relocatable file and saves each subprogram module name in a table in memory. LIBED next starts to copy relocatable subprogram modules from the input library to the scratch file. When the name of a subprogram module is found in the table then that module is copied from the input relocatable file rather than the input library and a flag is set in the table. When a subprogram module is copied from the relocatable file the date, time and output library names are inserted into the comment field of the NAM record. When the copy of the input library is complete LIBED scans the table for any subprogram modules which were not on the input library and adds them onto the end of the output library from the input relocatable library's scratch file. When the name of a subprogram module is found in the table then that module is copied from the input relocatable file rather than the input library and a flag is set in the table. When a subprogram module is copied from the relocatable file the date, time and output library names are inserted into the comment field of the NAM record. When the copy of the input library is complete LIBED scans the table for any subprogram modules which were not on the input library and adds them onto the end of the output library from the input relocatable library's scratch file. When the name of a subprogram module is found in the table then that module is copied from the input relocatable file rather than the input library and a flag is set in the table. When a subprogram module is copied from the relocatable file the date, time and output library names are inserted into the comment field of the NAM record. When the copy of the input library is complete LIBED scans the table for any subprogram modules which were not on the input library and adds them onto the end of the output library from the input relocatable library's scratch file.

Once all subprogram modules are copied onto the end of the output library LIBED truncates the output scratch file to the correct size, and closes the input library and input relocatable file. If the output library has the same name as the input library then the input library is renamed by replacing the first two characters of its name with %%. The output scratch file is then renamed and if this is successful the input library is purged. If the output library name is different from the input library then the output scratch file is renamed, and the input library is intact. If the rename of the scratch file fails, the input library remains with %% as the first two characters of its name. A flowchart for LIBED in update mode is given in Appendix C.

LIBED truncates the output scratch file to the correct size, and closes the input library and input relocatable file. If the output library has the same name as the input library then the input library is renamed by replacing the first two characters of its name with %%. The output scratch file is then renamed and if this is successful the input library is purged. If the output library name is different from the input library then the output scratch file is renamed, and the input library is intact. If the rename of the scratch file fails, the input library remains with %% as the first two characters of its name. A flowchart for LIBED in update mode is given in Appendix C.

## LIBED - RELOCATABLE LIBRARY MANAGER

### 4.2 LIST MODE

In LIST mode LIBED opens a relocatable library and reads each record. The details of any NAM, ENT or EXT record are printed on the list logical unit. The listing is dated and paged. LIST mode can be entered upon scheduling LIBED with P2 only or by the list command in COMMAND mode.

### 4.3 COMMAND MODE

In COMMAND mode LIBED responds to commands either from a terminal logical unit or in a disc file. If input is from a terminal LIBED will prompt for each input. COMMAND mode uses an input library, an output library and a table in memory. The input relocatable library is defined using the ATTACH command; the output library is defined using the LIBRARY command. A scratch output library file is created and renamed at the end of the job. Input and output libraries may have the same names but the input library will be purged at the end of the job if this is the case. Program modules are transferred from the input library to the output library using the COPY command. The DELETE command saves the names of programs to be deleted in a table in memory. When the next COPY command is executed, the program modules whose names are referred to in the table will not be copied. The file closing procedure is the same as for UPDATE mode. Control of LIBED in COMMAND mode may be transferred between a terminal and a file by the use of the TRANSFER command. COMMAND mode can be entered either by scheduling without parameters or by scheduling with P1 only, P1 being a command file NAMR.

# LIBED - RELOCATABLE LIBRARY MANAGER

SAMPLE FMGR TRANSFER FILE

APPENDIX A

This transfer file is used to compile a specified source file, make a relocatable file call %TEMP and use this to update a relocatable library using LIBED. This file handles both assemblies and compilations. This file can be used by the following FMGR command:

TR,U,SFILEN,LFILEN,LL,SC,CR

Where:

SFILEN = The name of a FORTRAN or ASSEMBLER source file

LFILEN = The name of a relocatable library (optional)

LL = The logical unit for the compiler listing (default = 6)

SC = The security code of the relocatable library (default = 40)

CR = The cartridge number of the relocatable library (default = 8)

```

:SV,2,,IH
:TE,***  UPDATE JOB  ***
:* CHECK THAT SOURCE FILE GIVEN
:IF,-36P,EQ,3,2
:TE,***  NO SOURCE FILE GIVEN - JOB ABORTED  ***
:TR
:IF,-28P,GT,0
:SE,,,6
:RU,COMPL,1G,3G,%TEMP::2
:IF,-32P,NE,3,6
:IF,-24P,EQ,1
:SE,,,40
:IF,-20P,EQ,1
:SE,,,,8
:* RUN LIBED IN UPDATE MODE
:RU,LIBED,%TEMP,2G:4G:5G
:PU,%TEMP
:SV,0

```

9:33 AM TUE., 14 APR., 1981

A-1

# LIBED - RELOCATABLE LIBRARY MANAGER

|                  |            |
|------------------|------------|
| SAMPLE PRINTOUTS | APPENDIX B |
|------------------|------------|

Three examples of printouts produced by LIBED are included. The first is a LIST mode printout and was produced in response to the command:

```
RU,LIBED,, $$UTIL:40:8
```

The second is a command mode printout and was produced in response to the following dialogue:

```
RU,LIBED
LIBED? LIBRARY,$$UTIL:40:8
LIBED? ATTACH,$LIBED
LIBED? DELETE,CHARM
LIBED? COPY
LIBED? END
```

The third printout was produced by executing the FMGR transfer file in Appendix A with the following FMGR instruction:

```
TR,U,&CHARM,$$UTIL
```

LIBED - RELOCATABLE LIBRARY MANAGER

LIST OF RELOCATABLE LIBRARY FILE: \$\$UTIL 1:48 PM TUE 5 AUG 1980 PAGE: 1  
\*\*\*\*\*

PROGRAM: INT32 TYPE= 7, PRIORITY= 99, LENGTH= 74, COMMON= 0,  
ENTRY POINTS: DIADD DISUB DIMUL DIDIV DIFIX DIFLT SHTOL LTOSH  
EXTERNALS: .ENTR .DAD .DSB .DMP .DDI .FIXD .FLTD IFIX

PROGRAM: IRAR TYPE= 7, PRIORITY= 99, LENGTH= 7, COMMON= 0,  
ENTRY POINTS: IRAR  
EXTERNALS: .ENTR

PROGRAM: MEM TYPE= 7, PRIORITY= 99, LENGTH= 71, COMMON= 0,  
ENTRY POINTS: FLPST INTST MEMOV

PROGRAM: ISHIF TYPE= 0, PRIORITY= 0, LENGTH= 20, COMMON= 0,  
ENTRY POINTS: ISHIF  
EXTERNALS: .ENTR

PROGRAM: CHARM TYPE= 7, PRIORITY= 99, LENGTH= 93, COMMON= 0,  
ENTRY POINTS: CMOVE ICSCN ICBYT  
EXTERNALS: .ENTR

PROGRAM: CREPL TYPE= 7, PRIORITY= 99, LENGTH= 31, COMMON= 0,  
ENTRY POINTS: CREPL  
EXTERNALS: .ENTR ICSCN CMOVE

PROGRAM: IUNIT TYPE= 7, PRIORITY= 99, LENGTH= 51, COMMON= 0,  
ENTRY POINTS: IUNIT  
EXTERNALS: .ENTR EXEC JSHIF

Sample printout from List Mode

9:33 AM TUE., 14 APR., 1981

B-2

LIBED - RELOCATABLE LIBRARY MANAGER

PROGRAM: INT32 COPIED FROM FILE \$\$UTIL  
PROGRAM: IRAR COPIED FROM FILE \$\$UTIL  
PROGRAM: MEM COPIED FROM FILE \$\$UTIL  
PROGRAM: ISHIF COPIED FROM FILE \$\$UTIL  
PROGRAM: CHARM DELETED  
PROGRAM: CREPL COPIED FROM FILE \$\$UTIL  
PROGRAM: IUNIT COPIED FROM FILE \$\$UTIL

NEW LIBRARY CATALOGUED AS: \$\$UTIL  
OLD LIBRARY FILE PURGED

Sample Command Mode Report

PROGRAM: CHARM ADDED TO LIBRARY  
NEW LIBRARY CATALOGUED AS: \$\$UTIL  
OLD LIBRARY FILE PURGED

Sample Update Mode Report

9:33 AM TUE., 14 APR., 1981

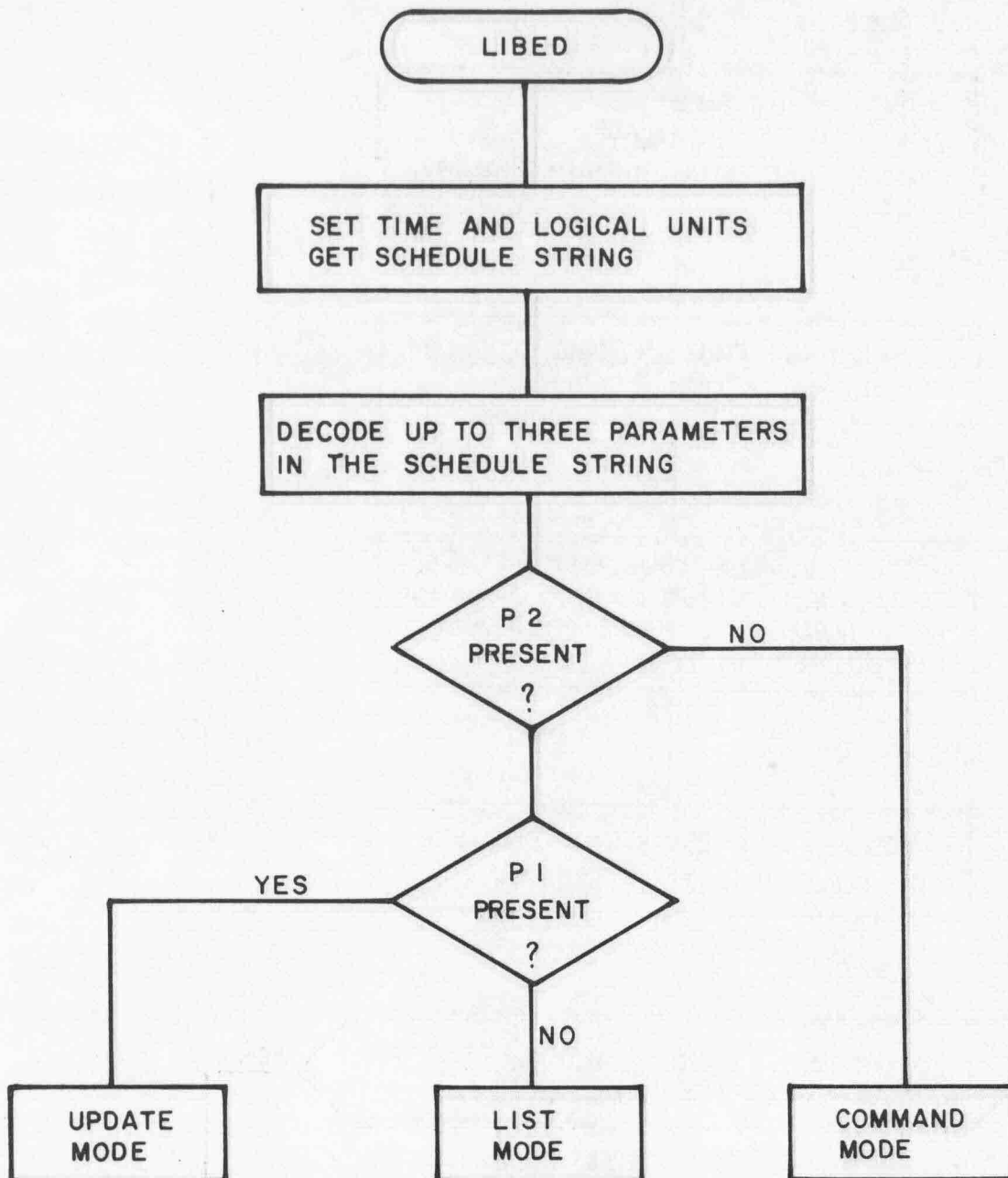


Fig. 1 Flowchart for LIBED main program

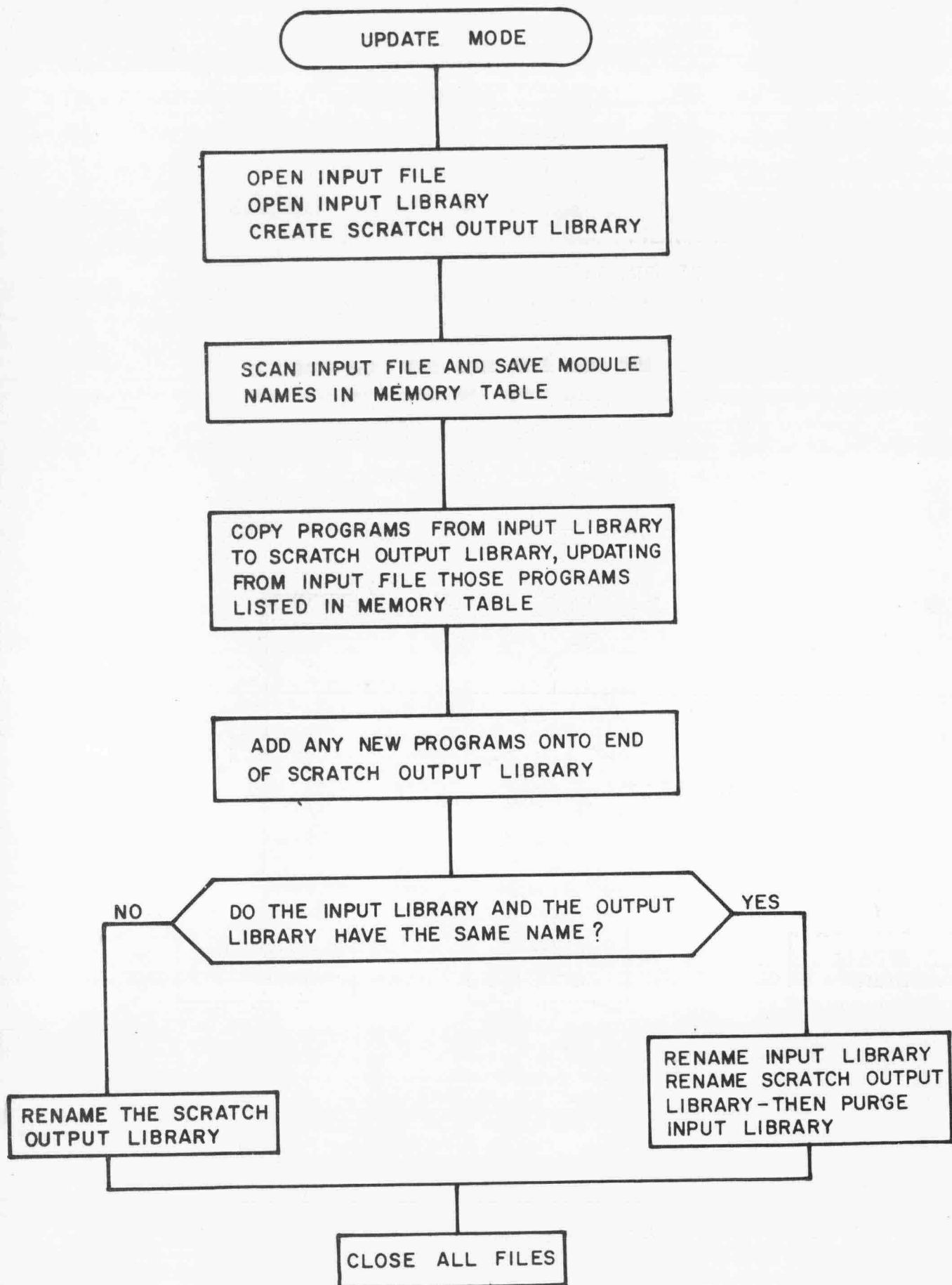


Fig.2 Flowchart for UPDATE MODE

# LIBED - RELOCATABLE LIBRARY MANAGER

## PROGRAM DESCRIPTIONS

## APPENDIX D

C\*\*\* LIBED  
C\*\*  
C\*\* LIBED IS A HP RTE RELOCATABLE SUBROUTINE LIBRARY  
C\*\* MANAGEMENT PROGRAM. ITS SCHEDULING AND USE ARE DESCRIBED  
C\*\* IN A SEPARATE DOCUMENT.  
C\*\*

C\*\*\* LUNS  
C\*\*  
C\*\* LUNS CONTAINS THE INPUT AND LIST LOGICAL UNITS  
C\*\* THEY ARE INITIALISED ON SCHEDULING OF LIBED.  
C\*\*

C\*\*\* FILES  
C\*\*  
C\*\* FILES IS A COMMON BLOCK WHICH CONTAINS THE FILE CONTROL  
C\*\* BUFFERS FOR LIBED. THERE ARE THREE FILES AVAILABLE,  
C\*\* AND THE ARRAYS ARE SET UP TO BE COMPATABLE WITH  
C\*\* NAMR DECODING OF AN INPUT STRING. AS WELL, THE NAME AND  
C\*\* SIZE OF THE OUTPUT SCRATCH FILE ARE SAVED.  
C\*\*

C\*\*\* TABLE  
C\*\*  
C\*\* TABLE IS A COMMON BLOCK CONTAINING THE TABLE FOR LIBED  
C\*\* IT IS USED TO RECORD INPUT LIBRARY ROUTINES AND  
C\*\* ROUTINES TO BE DELETED DURING COPIES. THERE ARE 500  
C\*\* ENTRIES IN THE TABLE, THOUGH THIS CAN BE ALTERED AT WILL.  
C\*\*

C\* N1 => TABLE POINTER  
C\* NMAX => MAXIMUM NUMBER OF TABLE ENTRIES (500)  
C\* NAME1 => TABLE - FOUR WORDS PER ENTRY.  
C\* WORD1 = FLAG WORD. WORDS 2-4 = ROUTINE NAME  
C\*

9:33 AM TUE., 14 APR., 1981

# LIBED - RELOCATABLE LIBRARY MANAGER

```
****    TODAY
***
***    THE TODAY BLOCK CONTAINS THE FORMATTED TIME GIVEN BY THE
***    RTE 'FTIME' ROUTINE.
***

****    WORKS
***
***    WORKS IS A COMMON BLOCK WHICH CONTAINS THE STRING
***    DECODING PARAMETERS FOR INPUT TO LIBED, AND ITS DECODING
***    BY THE NAMR ROUTINE
***

****    BADR
***
***    BADR SCANS AN EXTERNAL RECORD FOR THE REFERENCE
***    .BAD. THIS REFERENCE IS GENERATED IF COMPILER ERRORS
***    OCCUR DURING THE CREATION OF THE RELOCATABLE FILE.
***    IF SUCH AN EXTERNAL IS FOUND THEN LIBED WILL ABORT.
***    NOTE THAT THIS REFERENCE IS NOT GENERATED FOR COMPILER
***    WARNINGS.
***

****    COPIN
***
***    COPIN COPIES PROGRAMS FROM AN INPUT RELOCATABLE FILE
***    TO AN OUTPUT RELOCATABLE FILE. PROGRAMS MAY BE SPECIFIED BY
***    NAME OR IMPLICITLY.
***
***    FORMAL PARAMETERS:
***          IFLAG => ERROR FLAG
***          IFANY => FLAG THAT THERE IS SOMETHING ON THE
***                   OUTPUT FILE
***

****    COPLB
***
***    COPLB SUPERVISES THE COPYING OF RELOCATABLE LIBRARIES
***    FROM THE OLD LIBRARY TO THE NEW LIBRARY, REPLACING OR
***    DELETING AS NECESSARY
***
```

9:33 AM TUE., 14 APR., 1981

D-2

# LIBED - RELOCATABLE LIBRARY MANAGER

C\*\*\* COPYR  
C\*\*  
C\*\* COPYR IS USED TO COPY RELOCATABLE PROGRAMS FROM THE INPUT  
C\*\* FILES TO THE OUTPUT LIBRARY FILE.  
C\*\*

C\*\*\* CRELF  
C\*\*  
C\*\* CRELF CREATES A TYPE 5 RELOCATABLE SCRATCH FILE  
C\*\* THE DCB, SECURITY CODE AND CARTRIDGE  
C\*\* ARE AS SUPPLIED BY THE USER IN THE /FILES/  
C\*\* COMMON BLOCK  
C\*\*

C\*\*\* FCLOS  
C\*\*  
C\*\* FCLOS HANDLES THE CLOSING OF FMP FILES FOR NORMAL OR ERROR  
C\*\* RETURNS. ON A NORMAL RETURN THE OUTPUT LIBRARY FILE IS  
C\*\* TRUNCATED AND CLOSED AND THEN GIVEN ITS REQUESTED NAME.  
C\*\* IF THIS IS THE SAME AS THE INPUT LIBRARY FILE, THE INPUT  
C\*\* LIBRARY FILE IS RENAMED BY SUBSTITUTING % FOR THE FIRST TWO  
C\*\* CHARACTERS OF THE FILE NAME. THEN THE OUTPUT FILE  
C\*\* IS RENAMED AND THEN THE INPUT LIBRARY FILE IS PURGED.  
C\*\* IF THE OUTPUT LIBRARY AND THE INPUT LIBRARY FILES HAVE  
C\*\* DIFERENT NAMES THE INPUT LIBRARY IS NOT PURGED.  
C\*\* IFLAG IS NEGATIVE IF THE CLOSE IS AS A RESULT OF AN ERROR.  
C\*\* IN THIS CASE THE INPUT FILE AND LIBRARY ARE CLOSED  
C\*\* AND THE OUTPUT SCRATCH FILE IS PURGED.  
C\*\* WHAT MORE COULD YOU ASK FOR?  
C\*\*

C\*\*\* FINDR  
C\*\*  
C\*\* FINDR SEARCHES A RELOCATABLE FILE FOR THE NEXT NAM RECORD  
C\*\* AND LEAVES IT IN THE WORKSPACE BUFFER IBUF. IFLAG IS SET TO 1  
C\*\* IF A NAM RECORD IS FOUND, TO 0 IF AN EOF IS FOUND AND TO -1  
C\*\* IF AN ERROR OCCURS.  
C\*\*

C\*\*\* GETIN  
C\*\*  
C\*\* GETIN GETS THE NEXT INSTRUCTION FOR UDOIT TO EXECUTE.  
C\*\* INSTRUCTIONS MAY COME FROM A LOGIL UNIT OR A FILE.  
C\*\* IN EITHER CASE IT IS TRANSPARENT TO UDOIT. A COMMAND  
C\*\* TO TRANSFER FROM ONE INPUT SOURCE TO ANOTHER IS HANDLED BY GET??  
C\*\*  
C\* FORMAL PARAMETERS:  
C\* IFLAG : IF SET TO -1 AN ERROR HAS OCCURRED  
C\*

9:33 AM TUE., 14 APR., 1981

LIBED - RELOCATABLE LIBRARY MANAGER

C\*\*\*       INSRT  
C\*\*  
C\*\*       INSRT INSERTS THE CURRENT DATE AND OUTPUT LIBRARY  
C\*\*       NAME IN THE COMMENT REGION OF THE NAM RECORD OF AN INPUT  
C\*\*       RELOCATABLE ROUTINE.  
C\*\*  
C\*       FORMAL PARAMETER:  
C\*               LEN IS THE LENGTH OF THE NAM RECORD  
C\*               IT IS RESET BY THIS ROUTINE.  
C\*

C\*\*\*       LIBPR  
C\*\*  
C\*\*       LIBPR LISTS THE CONTENTS OF A RELOCATABLE LIBRARY.  
C\*\*       ALL NAM, ENT AND EXT RECORDS ARE LISTED.  
C\*\*

C\*\*\*       LISTE  
C\*\*  
C\*\*       LISTE LISTS DETAILS OF RELOCATABLE RECORD TYPES ON THE LIST  
C\*\*       LOGICAL UNIT. IF ITYPE = 1, THE RECORD IN THE WORKSPACE BUFFER  
C\*\*       IS A NAM RECORD, AND THE NAME, PROGRAM TYPE ETC. ARE LISTED.  
C\*\*       IF ITYPE = 2, THE RECORD IS AN ENT RECORD, AND IF ITYPE = 4  
C\*\*       THE RECORD IS AN EXT RECORD. THE ENTRY POINTS OR EXTERNALS  
C\*\*       ARE LISTED.  
C\*\*

C\*\*\*       NEWRL  
C\*\*  
C\*\*       THIS SUBROUTINE SUPERVISES THE CREATION OF A NEW  
C\*\*       PROGRAM LIBRARY FROM AN INPUT RELOCATABLE FILE AND A  
C\*\*       RELOCATABLE LIBRARY FILE.  
C\*\*

C\*\*\*       ROPEN  
C\*\*  
C\*\*       THIS SUBROUTINE OPENS A RELOCATABLE FILE "NAME"  
C\*\*       WITH SECURITY CODE ISECU ON CARTRIDGE ICR. THE FILE  
C\*\*       IS ATTACHED TO DATA CONTROL BLOCK "IDCB". IFLAG IS  
C\*\*       SET TO -1 IF THERE IS AN ERROR. THE FILE TYPE IS CHECKED  
C\*\*       TO ENSURE THAT IT IS TYPE 5.  
C\*\*  
C\*       FORMAL PARAMETERS:  
C\*               NAME   => RELOCATABLE FILE TO BE OPENED.  
C\*               IDCB   => DCB TO BE USED.  
C\*               ISECU  => SECURITY CODE  
C\*               ICR    => CARTRIDGE NUMBER  
C\*               IFLAG  => ERROR FLAG  
C\*

LIBED - RELOCATABLE LIBRARY MANAGER

```
C***      SCANR
C***
C***      SCANR SCANS THE INPUT RELOCATABLE FILE AND ENTERS EACH PROGRAM
C***      NAME IN A TABLE IN COMMON.
C***
C*      FORMAL PARAMETERS:
C*          IFLAG => ERROR FLAG
C*

C***      SRCHN
C***
C***      SRCHN SCANS THE TABLE FOR A GIVEN PROGRAM NAME.
C***
C*      FORMAL PARAMETERS:
C*          NAM    IS THE PROGRAM NAME
C*          N      IS THE TABLE POINTER WHERE NAM IS FOUND
C*          IFLAG IS THE STATUS FLAG OF THE PROGRAM IN
C*          TABLE, OR ZERO OF NOT FOUND.
C*

C***      SRCHR
C***
C***      SRCHR SEARCHES A RELOCATABLE FILE FOR A RELOCATABLE PROGRAM
C***
C*      FORMAL PARAMETERS:
C*          IFLAG IS RETURNED AS: 1 IF THE PROGRAM WAS FOUND
C*                                0 IF THE PROGRAM WAS NOT FOUND
C*                                -1 IF AN FMP ERROR OCCURED
C*          NAME IS THE PROGRAM NAME
C*          IDCB IS THE DCB OF THE RELOCATABLE FILE
C*          LEN IS THE LENGTH OF THE NAM RECORD, WHICH IS LEFT I??
C*          THE COMMON BUFFER.
C*

C***      UDOIT
C***
C***      UDOIT SUPERVISES EXECUTION OF LIBED IN THE COMMAND MODE.
C***
```

1

9:33 AM TUE., 14 APR., 1981

D-5

LIBED - RELOCATABLE LIBRARY MANAGER

|                         |            |
|-------------------------|------------|
| PROGRAM SOURCE LISTINGS | APPENDIX E |
|-------------------------|------------|

This appendix is attached as a microfiche.

9:33 AM TUE., 14 APR., 1981

E-1

.BAD., 3

A

ABORT, 7  
ASSEMBLER, 1, 2  
ATTACH, 7, 11

B

BIT BUCKET, 8  
BLOCK DATA STATEMENT, 2

C

COMMAND FILE, 4, 5  
COMMAND MODE, 4, 5, 6, 10, 11  
COMMANDS, 4, 7  
COMMENT FIELD, 5, 10  
COMPILER ERRORS, 3  
COPY, 7, 8, 11

D

DBL RECORD, 2  
DELETE, 8, 11

E

EMA RECORD, 2  
END, 8  
END OF FILE, 8  
END RECORD, 2  
ENT RECORD, 2, 4, 5, 11  
EXT RECORD, 2, 3, 4, 5, 11

## INDEX

## LIBED - RELOCATABLE LIBRARY MANAGER

### F

FMGR, 2  
FMGR FILE TYPES, 2  
FORTRAN, 1, 2, 3, 10  
FUNCTION STATEMENT, 2

### I

INPUT LIBRARY, 10, 11

### L

LIBED, 1  
LIBRARY, 1, 8, 11  
LIBRARY FILE, 2  
LIST LU, 4, 5, 8, 11  
LIST MODE, 4, 5, 6, 10, 11  
LL, 8  
LOAD MAP, 5  
LOADER, 1

### M

MAIN PROGRAM MODULES, 3

### N

NAM RECORD, 2, 4, 5, 10, 11  
NAMR, 2, 9  
NEW LIBRARY, 8

### O

ON-LINE GENERATOR, 3  
OUTPUT LIBRARY, 4, 8, 10, 11

P

PROGRAM STATEMENT, 2  
PROMPT, 5, 6, 11

R

RELOCATABLE FILE, 1, 2, 4, 10  
RELOCATABLE LIBRARIES, 3  
RELOCATABLE LIBRARY, 3, 4, 7, 10  
REPORT, 6  
REWIND, 9  
RTE OPERATING SYSTEM, 1

S

SCHEDULE STRING, 10  
SCRATCH FILE, 10, 11  
SOURCE FILE, 1, 2

T

TRANSFER, 9, 11  
TRANSFER FILE, 4

U

UPDATE MODE, 4, 6, 10

BMR RECORD 1981/9  
-----

COMPUTER PROGRAMS FOR LIBED  
RELOCATABLE LIBRARY MANAGEMENT PROGRAM  
FOR HP-1000 COMPUTERS WITH RTE OPERATING SYSTEMS

BY FRANK BRASSIL

```

2 00000      PROGRAM LIBED(), 09/04/81
3 00000 C
4 00000 C***      LIBED
5 00000 C**
6 00000 C**      LIBED IS A HP RTE RELOCatable SUBROUTINE LIBRARY
7 00000 C**      MANAGEMENT PROGRAM. ITS SCHEDULING AND USE ARE DESCRIBED
8 00000 C**      IN A SEPARATE DOCUMENT.
9 00000 C**
10 00000      COMMON IBUF(128)
11 00000      COMMON/LUWS / INPUT,LIST
12 00000      COMMON/FILES/ INAME(3),ITYPE,ISC,ICR,IDCB(144),
13 00000      1 JNAME(3),JTYPE,JSC,JCR,JDCB(144),
14 00000      1 KNAME(3),KTYPE,KSC,KCR,KDCB(144),NAMES(3),KSIZE
15 00000      COMMON/TABLE/ N1,NMAX,NAME1(4,500)
16 00000      COMMON/TODAY/ IDATE(15)
17 00000      COMMON/WORKS/ KBUF(40),JBUF(10),LOG,ISTRC
18 00000 C
19 00000 C      SET SYSTEM TIME, GET SCHEDULING UNITS AND THE SCHEDULE STRING
20 00000 C
21 00000      CALL FTIME(IDATE) $ NMAX = 500
22 00010      INPUT = LOGLU(IDUM) $ LIST = INPUT $ CALL GETST(KBUF,-80,LOG)
23 00023 C
24 00023 C      DECODE THE SCHEDULE STRING ONE ELEMENT AT A TIME.
25 00023 C
26 00023      ISTRC = 1
27 00025      IF (NAMR(INAME,KBUF,LOG,ISTRC)) 100,10
28 00035      10 IT = IAND(ITYPE,3B)
29 00040      IF (NAMR(JNAME,KBUF,LOG,ISTRC)) 100,20
30 00050      20 JT = IAND(JTYPE,3B)
31 00053      IF (NAMR(KNAME,KBUF,LOG,ISTRC)) 100,30
32 00063      30 KT = IAND(KTYPE,3B) $ IF (KDCB(2).NE.0) KSIZE = KDCB(2)
33 00073      IF (NAMR(JBUF,KBUF,LOG,ISTRC)) 100,40
34 00103      40 IF (IAND(JBUF(4),3B).EQ.1) LIST = JBUF(1)
35 00110 C
36 00110 C      WORK OUT THE EXECUTION MODE AND DO IT
37 00110 C
38 00112      100 IF (KTYPE.EQ.0) CALL MEMOV(JNAME,KNAME,6)
39 00122      IF (JT.NE.3) GO TO 130 $ IF (IT.EQ.3) 120,150
40 00132 C
41 00132 C      UPDATE MODE
42 00132 C
43 00134      120 CALL NEWRL $ GO TO 200
44 00136 C
45 00136 C      INTERACTIVE MODE
46 00136 C
47 00137      130 CALL UDOIT $ GO TO 200
48 00141 C
49 00141 C      LIST MODE
50 00141 C
51 00142      150 CALL ROPEN(KNAME,KDCB,KSC,KCR,IFLAG,0) $ IF (IFLAG) 200,160
52 00155      160 LIST = 6 $ CALL LIBPR
53 00161      200 CALL EXEC(6)
54 00164      END

```

PAGE 2 LIBED OPTS: QTYI 3:41 PM FRI., 10 APR., 1981

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 135 COMMON: 128

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION       |
|-------|---------|------------------|---------|----------------|
| @10   | 00035R  | STATEMENT NUMBER |         |                |
| @100  | 00112R  | STATEMENT NUMBER |         |                |
| @120  | 00134R  | STATEMENT NUMBER |         |                |
| @130  | 00137R  | STATEMENT NUMBER |         |                |
| @150  | 00142R  | STATEMENT NUMBER |         |                |
| @160  | 00155R  | STATEMENT NUMBER |         |                |
| @20   | 00050R  | STATEMENT NUMBER |         |                |
| @200  | 00161R  | STATEMENT NUMBER |         |                |
| @30   | 00063R  | STATEMENT NUMBER |         |                |
| @40   | 00103R  | STATEMENT NUMBER |         |                |
| EXEC  |         | INTRINSIC        |         |                |
| FILES | 00012X  | COMMON LABEL     |         | EXTERNAL       |
| FTIME | 00002X  | SUBROUTINE       |         | EXTERNAL       |
| GETST | 00007X  | SUBROUTINE       |         | EXTERNAL       |
| IAND  |         | INTRINSIC        | GENERIC |                |
| IBUF  | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON         |
| ICR   | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDATE | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON TODAY |
| IDCB  | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IDUM  | 00172R  | VARIABLE         | INTEGER | LOCAL          |
| IFLAG | 00166R  | VARIABLE         | INTEGER | LOCAL          |
| INAME | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| ISC   | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ISTRC | 00063+  | VARIABLE         | INTEGER | L COMMON WORKS |
| IT    | 00167R  | VARIABLE         | INTEGER | LOCAL          |
| ITYPE | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| JBUF  | 00050+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON WORKS |
| JCR   | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB  | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JNAME | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JSC   | 00232+  | VARIABLE         | INTEGER | L COMMON FILES |
| JT    | 00170R  | VARIABLE         | INTEGER | LOCAL          |
| JTYPE | 00231+  | VARIABLE         | INTEGER | L COMMON FILES |
| KBUF  | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON WORKS |
| KCR   | 00461+  | VARIABLE         | INTEGER | L COMMON FILES |
| KDCB  | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KNAME | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KSC   | 00460+  | VARIABLE         | INTEGER | L COMMON FILES |
| KSIZE | 00705+  | VARIABLE         | INTEGER | L COMMON FILES |
| KT    | 00171R  | VARIABLE         | INTEGER | LOCAL          |
| KTYPE | 00457+  | VARIABLE         | INTEGER | L COMMON FILES |
| LIBPR | 00017X  | SUBROUTINE       |         | EXTERNAL       |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| LOG   | 00062+  | VARIABLE         | INTEGER | L COMMON WORKS |
| LOGLU | 00005X  | FUNCTION         | INTEGER | EXTERNAL       |
| LUNS  | 00006X  | COMMON LABEL     |         | EXTERNAL       |
| MEMOV | 00013X  | SUBROUTINE       |         | EXTERNAL       |
| N1    | 00000+  | VARIABLE         | INTEGER | L COMMON TABLE |
| NAME1 | 00002+  | ARRAY, 2 DIMEN.  | INTEGER | L COMMON TABLE |

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|-------|---------|-----------------|---------|----------|-------|
| NAMES | 00702+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| NAMR  | 00011X  | FUNCTION        | INTEGER | EXTERNAL |       |
| NEWRL | 00014X  | SUBROUTINE      |         | EXTERNAL |       |
| NMAX  | 00001+  | VARIABLE        | INTEGER | L COMMON | TABLE |
| ROPEN | 00016X  | SUBROUTINE      |         | EXTERNAL |       |
| TABLE | 00004X  | COMMON LABEL    |         | EXTERNAL |       |
| TODAY | 00003X  | COMMON LABEL    |         | EXTERNAL |       |
| UD01T | 00015X  | SUBROUTINE      |         | EXTERNAL |       |
| WORKS | 00010X  | COMMON LABEL    |         | EXTERNAL |       |

PAGE 5 FTN. OPTS: QTYI 3:41 PM FRI., 10 APR., 1981

```
55 00000 BLOCK DATA LUNS, 10/04/81
56 00000 C
57 00000 C*** LUNS
58 00000 C**
59 00000 C** LUNS CONTAINS THE INPUT AND LIST LOGICAL UNITS
60 00000 C** THEY ARE INITIALISED ON SCHEDULING OF LIBED.
61 00000 C**
62 00000 COMMON/LUNS / INPUT,LIST
63 00000 END
```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: (NONE) COMMON: (NONE)

BLOCK COMMON LUNS SIZE: 2

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE        | TYPE    | LOCATION      |
|-------|---------|--------------|---------|---------------|
| INPUT | 00000+  | VARIABLE     | INTEGER | L COMMON LUNS |
| LIST  | 00001+  | VARIABLE     | INTEGER | L COMMON LUNS |
| LUNS  | 00002X  | COMMON LABEL |         | EXTERNAL      |

```
64 00000      BLOCK DATA FILES, 10/04/81
65 00000 C
66 00000 C***      FILES
67 00000 C**
68 00000 C**      FILES IS A COMMON BLOCK WHICH CONTAINS THE FILE CONTROL
69 00000 C**      BUFFERS FOR LIBED. THERE ARE THREE FILES AVAILABLE,
70 00000 C**      AND THE ARRAYS ARE SET UP TO BE COMPATABLE WITH
71 00000 C**      NAMR DECODING OF AN INPUT STRING. AS WELL, THE NAME AND
72 00000 C**      SIZE OF THE OUTPUT SCRATCH FILE ARE SAVED.
73 00000 C**
74 00000      COMMON/FILES/ INAME(3), ITYPE, ISC, ICR, IDCB(144),
75 00000      1          JNAME(3), JTYPE, JSC, JCR, JDCB(144),
76 00000      2          KNAME(3), KTYPE, KSC, KCR, KDCB(144), NAMES(3), KSIZE
77 00000      END
```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: (NONE) COMMON: (NONE)

BLOCK COMMON FILES SIZE: 454

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|-------|---------|-----------------|---------|----------|-------|
| FILES | 00706X  | COMMON LABEL    |         | EXTERNAL |       |
| ICR   | 00005+  | VARIABLE        | INTEGER | L COMMON | FILES |
| IDCB  | 00006+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| INAME | 00000+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| ISC   | 00004+  | VARIABLE        | INTEGER | L COMMON | FILES |
| ITYPE | 00003+  | VARIABLE        | INTEGER | L COMMON | FILES |
| JCR   | 00233+  | VARIABLE        | INTEGER | L COMMON | FILES |
| JDCB  | 00234+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| JNAME | 00226+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| JSC   | 00232+  | VARIABLE        | INTEGER | L COMMON | FILES |
| JTYPE | 00231+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KCR   | 00461+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KDCB  | 00462+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| KNAME | 00454+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| KSC   | 00460+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KSIZE | 00705+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KTYPE | 00457+  | VARIABLE        | INTEGER | L COMMON | FILES |
| NAMES | 00702+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |

```

78 00000      BLOCK DATA TABLE, 10/04/81
79 00000 C
80 00000 C***      TABLE
81 00000 C**
82 00000 C**      TABLE IS A COMMON BLOCK CONTAINING THE TABLE FOR LIBED
83 00000 C**      IT IS USED TO RECORD INPUT LIBRARY ROUTINES AND
84 00000 C**      ROUTINES TO BE DELETED DURING COPIES. THERE ARE 500
85 00000 C**      ENTRIES IN THE TABLE, THOUGH THIS CAN BE ALTERED AT WILL.
86 00000 C**
87 00000 C*      N1      => TABLE POINTER
88 00000 C*      NMAX     => MAXIMUM NUMBER OF TABLE ENTRIES (500)
89 00000 C*      NAME1    => TABLE - FOUR WORDS PER ENTRY.
90 00000 C*      WORD1 = FLAG WORD. WORDS 2-4 = ROUTINE NAME
91 00000 C*
92 00000      COMMON/TABLE/ N1,NMAX,NAME1(4,500)
93 00000      DATA      N1,NMAX
94 00000      1          / 0, 500/
95 00000      END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: (NONE) COMMON: (NONE)

BLOCK COMMON TABLE SIZE: 2002

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|-------|---------|-----------------|---------|----------|-------|
| N1    | 00000+  | VARIABLE        | INTEGER | L COMMON | TABLE |
| NAME1 | 00002+  | ARRAY, 2 DIMEN. | INTEGER | L COMMON | TABLE |
| NMAX  | 00001+  | VARIABLE        | INTEGER | L COMMON | TABLE |
| TABLE | 03722X  | COMMON LABEL    |         | EXTERNAL |       |

PAGE 11 FTN. OPTS: QTYI 3:41 PM FRI., 10 APR., 1981

```
96 00000 BLOCK DATA TODAY, 10/04/81
97 00000 C
98 00000 C*** TODAY
99 00000 C**
100 00000 C** THE TODAY BLOCK CONTAINS THE FORMATTED TIME GIVEN BY THE
101 00000 C** RTE 'FTIME' ROUTINE.
102 00000 C**
103 00000 COMMON/TODAY/ IDATE(15)
104 00000 END
```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: (NONE) COMMON: (NONE)

BLOCK COMMON TODAY SIZE: 15

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|-------|---------|-----------------|---------|----------|-------|
| DATE  | 00000+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | TODAY |
| TODAY | 00017X  | COMMON LABEL    |         | EXTERNAL |       |

PAGE 13 F.W. OPTS: QTY1 3:41 PM FRI., 10 APR., 1981

```
105 00000 BLOCK DATA WORKS, 10/04/81
106 00000 C
107 00000 C*** WORKS
108 00000 C**
109 00000 C** WORKS IS A COMMON BLOCK WHICH CONTAINS THE STRING
110 00000 C** DECODING PARAMETERS FOR INPUT TO LIBED, AND ITS DECODING
111 00000 C** BY THE NAMR ROUTINE
112 00000 C**
113 00000 COMMON/WORKS/ KBUF(40),JBUF(10),LOG,ISTRC
114 00000 END
```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: (NONE) COMMON: (NONE)

BLOCK COMMON WORKS SIZE: 52

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|-------|---------|-----------------|---------|----------|-------|
| ISTRC | 00063+  | VARIABLE        | INTEGER | L COMMON | WORKS |
| JBUF  | 00050+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | WORKS |
| KBUF  | 00000+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | WORKS |
| LOG   | 00062+  | VARIABLE        | INTEGER | L COMMON | WORKS |
| WORKS | 00064X  | COMMON LABEL    |         | EXTERNAL |       |

```

115 00034      SUBROUTINE BADR(IFLAG), 09/06/81
116 00034 C
117 00034 C***      BADR
118 00034 C**
119 00034 C**      BADR SCANS AN EXTERNAL RECORD FOR THE REFERENCE
120 00034 C**      .BAD. THIS REFERENCE IS GENERATED IF COMPILER ERRORS
121 00034 C**      OCCUR DURING THE CREATION OF THE RELOCATABLE FILE.
122 00034 C**      IF SUCH AN EXTERNAL IS FOUND THEN LIBED WILL ABORT.
123 00034 C**      NOTE THAT THIS REFERENCE IS NOT GENERATED FOR COMPILER
124 00034 C**      WARNINGS.
125 00034 C*
126 00034      DIMENSION IBAD(3)
127 00034      COMMON IBUF(128)
128 00034      COMMON/LUNS /INPUT,LIST
129 00034      DATA IBAD/ '.BAD.' /
130 00034      NENT'S = IAND(IBUF(2),37B)
131 00043      DO 10 I=1,NENTS
132 00045      IF (ICBYT(IBUF(3*I+1),0,IBAD,0,5).NE 0) GO TO 20
133 00064      10 CONTINUE
134 00073      IFLAG = 0 $ RETURN
135 00077      20 WRITE(LIST,1000) $ IFLAG = -1
136 00113      1000 FORMAT(5X5(1H*),5X'COMPILER ERRORS IN RELOCATABLE CODE')
137 00113      END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 86 COMMON: 128

SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION         |
|-------|---------|------------------|---------|------------------|
| @10   | 00064R  | STATEMENT NUMBER |         |                  |
| @1000 | 00003K  | STATEMENT NUMBER |         |                  |
| @20   | 00077R  | STATEMENT NUMBER |         |                  |
| I     | 00116R  | VARIABLE         | INTEGER | LOCAL            |
| !AND  |         | INTRINSIC        | GENERIC |                  |
| IBAD  | 00000R  | ARRAY, 1 DIMEN.  | INTEGER | LOCAL            |
| IBUF  | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON           |
| ICBYT | 00002X  | FUNCTION         | INTEGER | EXTERNAL         |
| IFLAG | 00034R  | VARIABLE         | INTEGER | DUMMY            |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON    LUNS |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON    LUNS |
| LUNS  | 00003X  | COMMON LABEL     |         | EXTERNAL         |
| NENTS | 00117R  | VARIABLE         | INTEGER | LOCAL            |

```

138 00110 SUBROUTINE COPIN(IFLAG,IFANY), 09/06/81
139 00110 C
140 00110 C*** COPIN
141 00110 C**
142 00110 C** COPIN COPIES PROGRAMS FROM AN INPUT RELOCATABLE FILE
143 00110 C** TO AN OUTPUT RELOCATABLE FILE. PROGRAMS MAY BE SPECIFIED BY
144 00110 C** NAME OR IMPLICITLY.
145 00110 C**
146 00110 C* FORMAL PARAMETERS:
147 00110 C* IFLAG => ERROR FLAG
148 00110 C* IFANY => FLAG THAT THERE IS SOMETHING ON THE
149 00110 C* OUTPUT FILE
150 00110 C*
151 00110 DIMENSION IPGM(3),NAME(3)
152 00110 COMMON IBUF(128)
153 00110 COMMON/WORKS/ KBUF(40),JBUF(10),LOG,ISTRC
154 00110 COMMON/LUNS / INPUT,LIST
155 00110 COMMON/FILES/ INAME(3),ITYPE,ISC,ICR,IDCB(144),
156 00110 1 JNAME(3),JTYPE,JSC,JCR,JDCB(144),
157 00110 2 KNAME(3),KTYPE,KSC,KCR,KDCB(144),NAMES(3)
158 00110 C
159 00110 C PRESET FLAG AND TEST WHETHER PROGRAM 1 IS GIVEN. IF
160 00110 C IT IS, THEN SEARCH FOR IT. OTHERWISE START FROM THE NEXT
161 00110 C PROGRAM.
162 00110 C
163 00110 JFLAG1 = -1 $ JFLAG2 = -1
164 00121 IF (NAMR(JBUF,KBUF,LOG,ISTRC)) 10,2
165 00131 2 IF (IAND(JBUF(4),3B).EQ.3) 3,5
166 00140 3 JFLAG1 = 1 $ CALL MEMOV(JBUF,NAME,3) $ CALL MEMOV(JBUF,IPGM,3)
167 00154 5 IF (NAMR(JBUF,KBUF,LOG,ISTRC)) 10,6
168 00164 6 IF (IAND(JBUF(4),3B).EQ.3) 7,10
169 00173 7 JFLAG2 = 1 $ CALL MEMOV(JBUF,IPGM,3)
170 00202 10 IF (JFLAG1) 30,15
171 00205 15 CALL SRCHR(IFLAG,NAME,JDCB,LEN) $ IF (IFLAG) 25,20,45
172 00220 20 WRITE(LIST,1020) NAME,JNAME $ IFLAG = -1
173 00242 1020 FORMAT(5X5(1H*),5X PROGRAM: ',3A2,' NOT FOUND ON FILE ',3A2)
174 00242 25 RETURN
175 00244 30 CALL FINDR(JDCB,IFLAG,LEN) $ IF (IFLAG) 25,35,40
176 00256 35 IF (JFLAG2.LT.0) WRITE(LIST,1020) IPGM,JNAME $ GO TO 25
177 00302 40 CALL MEMOV(IBUF(4),NAME,3)
178 00307 45 CALL SRCHN(NAME,N,IFLAG) $ IF (IFLAG) 46,50
179 00317 46 WRITE(LIST,1040) NAME $ GO TO 30
180 00335 1040 FORMAT(10X PROGRAM: ',3A2,' DELETED )
181 00335 50 CALL COPYR(2,IFLAG,LEN) $ IF (IFLAG.LT.0) RETURN $ IFANY = 1
182 00351 55 WRITE(LIST,1050) NAME,JNAME $ IF (JFLAG2) 55,60
183 00374 1050 FORMAT(10X PROGRAM: ',3A2,' COPIED FROM FILE ',3A2)
184 00374 55 IF (JFLAG1) 30,25
185 00400 60 IF (ICBYT(NAME,0,IPGM,0,5).LT.0) GO TO 30
186 00411 END

```

PAGE 18 COPIN OPTS: QTYI 3:41 PM FRI., 10 APR., 1981

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 277 COMMON: 128

## SYMBOL TABLE

| NAME   | ADDRESS | USAGE            | TYPE    | LOCATION       |
|--------|---------|------------------|---------|----------------|
| a10    | 00202R  | STATEMENT NUMBER |         |                |
| a1020  | 00006R  | STATEMENT NUMBER |         |                |
| a1040  | 00041R  | STATEMENT NUMBER |         |                |
| a1050  | 00061R  | STATEMENT NUMBER |         |                |
| a15    | 00205R  | STATEMENT NUMBER |         |                |
| a2     | 00131R  | STATEMENT NUMBER |         |                |
| a20    | 00220R  | STATEMENT NUMBER |         |                |
| a25    | 00242R  | STATEMENT NUMBER |         |                |
| a3     | 00140R  | STATEMENT NUMBER |         |                |
| a30    | 00244R  | STATEMENT NUMBER |         |                |
| a35    | 00256R  | STATEMENT NUMBER |         |                |
| a40    | 00302R  | STATEMENT NUMBER |         |                |
| a45    | 00307R  | STATEMENT NUMBER |         |                |
| a46    | 00317R  | STATEMENT NUMBER |         |                |
| a5     | 00154R  | STATEMENT NUMBER |         |                |
| a50    | 00335R  | STATEMENT NUMBER |         |                |
| a55    | 00374R  | STATEMENT NUMBER |         |                |
| a6     | 00164R  | STATEMENT NUMBER |         |                |
| a60    | 00400R  | STATEMENT NUMBER |         |                |
| a7     | 00173R  | STATEMENT NUMBER |         |                |
| COPYR  | 00016X  | SUBROUTINE       |         | EXTERNAL       |
| FILES  | 00006X  | COMMON LABEL     |         | EXTERNAL       |
| FINDR  | 00014X  | SUBROUTINE       |         | EXTERNAL       |
| IAND   |         | INTRINSIC        | GENERIC |                |
| IBUF   | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON         |
| ICBYT  | 00017X  | FUNCTION         | INTEGER | EXTERNAL       |
| ICR    | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDCB   | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IFANY  | 00111R  | VARIABLE         | INTEGER | DUMMY          |
| IFLAG  | 00110R  | VARIABLE         | INTEGER | DUMMY          |
| INAME  | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| INPUT  | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| IPGM   | 00003R  | ARRAY, 1 DIMEN.  | INTEGER | LOCAL          |
| ISC    | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ISTR   | 00063+  | VARIABLE         | INTEGER | L COMMON WORKS |
| ITYPE  | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| JBUF   | 00050+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON WORKS |
| JCR    | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB   | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JFLAG1 | 00414R  | VARIABLE         | INTEGER | LOCAL          |
| JFLAG2 | 00415R  | VARIABLE         | INTEGER | LOCAL          |
| JNAME  | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JSC    | 00232+  | VARIABLE         | INTEGER | L COMMON FILES |
| JTYPE  | 00231+  | VARIABLE         | INTEGER | L COMMON FILES |
| KBUF   | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON WORKS |
| KCR    | 00461+  | VARIABLE         | INTEGER | L COMMON FILES |
| KDCB   | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KNAME  | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KSC    | 00460+  | VARIABLE         | INTEGER | L COMMON FILES |
| KTYPE  | 00457+  | VARIABLE         | INTEGER | L COMMON FILES |

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|-------|---------|-----------------|---------|----------|-------|
| LEN   | 00416R  | VARIABLE        | INTEGER | LOCAL    |       |
| LIST  | 00001+  | VARIABLE        | INTEGER | COMMON   | LUNS  |
| LOG   | 00062+  | VARIABLE        | INTEGER | COMMON   | WORKS |
| LUNS  | 00007X  | COMMON LABEL    |         | EXTERNAL |       |
| MEMOV | 00004X  | SUBROUTINE      |         | EXTERNAL |       |
| N     | 00417R  | VARIABLE        | INTEGER | LOCAL    |       |
| NAME  | 00000R  | ARRAY, 1 DIMEN. | INTEGER | LOCAL    |       |
| NAMES | 00702+  | ARRAY, 1 DIMEN. | INTEGER | COMMON   | FILES |
| NAMR  | 00002X  | FUNCTION        | INTEGER | EXTERNAL |       |
| SRCHN | 00015X  | SUBROUTINE      |         | EXTERNAL |       |
| SRCHR | 00005X  | SUBROUTINE      |         | EXTERNAL |       |
| WORKS | 00003X  | COMMON LABEL    |         | EXTERNAL |       |

```

187 00122      SUBROUTINE COPLB(IFLAG), 09/04/81
188 00122 C
189 00122 C***      COPLB
190 00122 C**
191 00122 C**      COPLB SUPERVISES THE COPYING OF RELOCATABLE LIBRARIES
192 00122 C**      FROM THE OLD LIBRARY TO THE NEW LIBRAR , REPLACING OR
193 00122 C**      DELETING AS NECESSARY
194 00122 C**
195 00122      DIMENSION NAME(3)
196 00122      COMMON IBUF(128)
197 00122      COMMON/LUNS / INPUT,LIST
198 00122      COMMON/FILES/ INAME(3),IYPE,ISC,ICR,IDCB(144),
199 00122      1 JNAME(3),JTYPE,JSC,JCR,JDCB(144),
200 00122      2 KNAME(3),KTYPE,KSC,KCR,KDCB(144),NAMES(3)
201 00122      COMMON/TABLE/ N1,NMAX,NAME1(4,500)
202 00122 C
203 00122 C      PRESET COUNTERS AND SEARCH FOR THE NEXT RELOCATABLE
204 00122 C      PROGRAM, THEN TEST FOR STATUS OF SEARCH.
205 00122 C
206 00122      KOUNT1 = 0 $ KOUNT2 = 0
207 00132      IDEL = 0 $ IREPL = 0 $ IADD = 0
208 00140      10 CALL FINDR(JDCB,IFLAG,LEN) $ IF (IFLAG) 100,50,15
209 00145 C
210 00145 C      BUMP INPUT COUNTER AND SCAN THE TABLE
211 00145 C
212 00152      15 KOUNT1 = KOUNT1 + 1 $ CALL MEMOV(IBUF(4),NAME,3)
213 00162      CALL SRCHN(NAME,N,IFLAG) $ IF (IFLAG) 20,25,30
214 00167 C
215 00167 C      DELETE PROGRAM
216 00167 C
217 00174      20 WRITE(LIST,1020) NAME $ IDEL = IDEL+1 $ GO TO 10
218 00215      1020 FORMAT(10X'PROGRAM: ',3A2,' DELETED ')
219 00215 C
220 00215 C      COPY PROGRAM FROM OLD LIBRARY TO NEW LIBRARY
221 00215 C
222 00215      25 KOUNT2 = KOUNT2 + 1
223 00220      CALL COPYR(2,IFLAG,LEN) $ IF (IFLAG) 100,10
224 00225 C
225 00225 C      REPLACE PROGRAM. SEARCH FOR PROGRAM ON INPUT AND INSERT
226 00225 C      DATE IN NAM FIELD IF THERE IS ROOM, AND THEN COPY THE
227 00225 C      NEW PROGRAM TO THE NEW LIBRARY
228 00225 C
229 00231      30 IREPL = IREPL + 1
230 00234      CALL SRCHR(IFLAG,NAME,IDCB,LEN) $ IF (IFLAG) 100,45,35
231 00247      35 CALL INSRT(LEN) $ CALL COPYR(1,IFLAG,LEN) $ IF (IFLAG) 100,40
232 00262      40 WRITE(LIST,1040) NAME $ NAME1(1,N) = -1 $ GO TO 10
233 00305      1040 FORMAT(10X'PROGRAM: ',3A2,' REPLACED')
234 00305      45 WRITE(LIST,1041) NAME,INAME $ GO TO 10
235 00326      1041 FORMAT(5X5(1H+),5X'PROGRAM: ',3A2,' NOT IN INPUT FILE ',3A2)
236 00326 C
237 00326 C      ADD REMAINING ROUTINES TO NEW LIBRARY
238 00326 C
239 00326      50 DO 75 I=1,N1
240 00330      IF (NAME1(1,I).EQ.-1) GO TO 75
241 00336      CALL SRCHR(IFLAG,NAME1(2,I),IDCB,LEN) $ IF (IFLAG) 100,55,60

```

```
242 00355 55 WRITE(LIST,1041) (NAME1(J,1),J=2,4),INAME $ GO TO 75
243 00412 60 CALL INSRT(LEN) $ CALL COPYR(1,IFLAG,LEN) $ IF (IFLAG) 100,65
244 00425 65 WRITE(LIST,1060) (NAME1(J,1),J=2,4) $ IADD = IADD + 1
245 00461 1060 FORMAT(10X"PROGRAM: ",3A2," ADDED TO LIBRARY")
246 00461 75 CONTINUE
247 00461 D
248 00461 D WRITE MESSAGES TO LIST
249 00461 D
250 00461 D KOUNT2 = KOUNT2 + IADD + IREPL
251 00461 D WRITE(LIST,80) KOUNT1
252 00461 D 80 FORMAT(10X15," ROUTINES FOUND ON INPUT LIBRARY")
253 00461 D IF (IDEL.GT.0) WRITE(LIST,1080) IDEL
254 00461 D1080 FORMAT(10X15," ROUTINES DELETED")
255 00461 D IF (IREPL.GT.0) WRITE(LIST,1081) IREPL
256 00461 D1081 FORMAT(10X15," ROUTINES REPLACED")
257 00461 D IF (IADD.GT.0) WRITE(LIST,1082) IADD
258 00461 D1082 FORMAT(10X15," ROUTINES ADDED")
259 00461 D WRITE(LIST,1083) KOUNT2
260 00461 D1083 FORMAT(10X15," ROUTINES WRITTEN ONTO OUTPUT LIBRARY FILE.")
261 00470 100 RET. RN
262 00472 END
```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 334 COMMON: 128

## SYMBOL TABLE

| NAME   | ADDRESS | USAGE            | TYPE    | LOCATION       |
|--------|---------|------------------|---------|----------------|
| a10    | 00140R  | STATEMENT NUMBER |         |                |
| a100   | 00470R  | STATEMENT NUMBER |         |                |
| a1020  | 00003R  | STATEMENT NUMBER |         |                |
| a1040  | 00023R  | STATEMENT NUMBER |         |                |
| a1041  | 00043R  | STATEMENT NUMBER |         |                |
| a1060  | 00076R  | STATEMENT NUMBER |         |                |
| a15    | 00152R  | STATEMENT NUMBER |         |                |
| a20    | 00174R  | STATEMENT NUMBER |         |                |
| a25    | 00215R  | STATEMENT NUMBER |         |                |
| a30    | 00231R  | STATEMENT NUMBER |         |                |
| a35    | 00247R  | STATEMENT NUMBER |         |                |
| a40    | 00262R  | STATEMENT NUMBER |         |                |
| a45    | 00305R  | STATEMENT NUMBER |         |                |
| a50    | 00326R  | STATEMENT NUMBER |         |                |
| a55    | 00355R  | STATEMENT NUMBER |         |                |
| a60    | 00412R  | STATEMENT NUMBER |         |                |
| a65    | 00425R  | STATEMENT NUMBER |         |                |
| a75    | 00461R  | STATEMENT NUMBER |         |                |
| COPYR  | 00013X  | SUBROUTINE       |         | EXTERNAL       |
| FILES  | 00003X  | COMMON LABEL     |         | EXTERNAL       |
| FINDR  | 00002X  | SUBROUTINE       |         | EXTERNAL       |
| I      | 00476R  | VARIABLE         | INTEGER | LOCAL          |
| IADD   | 00474R  | VARIABLE         | INTEGER | LOCAL          |
| IBUF   | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON         |
| ICR    | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDCB   | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IDEL   | 00503R  | VARIABLE         | INTEGER | LOCAL          |
| IFLAG  | 00122R  | VARIABLE         | INTEGER | DUMMY          |
| INAME  | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| INPUT  | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| INSRT  | 00015X  | SUBROUTINE       |         | EXTERNAL       |
| IREPL  | 00501R  | VARIABLE         | INTEGER | LOCAL          |
| ISC    | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ITYPE  | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| J      | 00475R  | VARIABLE         | INTEGER | LOCAL          |
| JCR    | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB   | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JNAME  | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JSC    | 00232+  | VARIABLE         | INTEGER | L COMMON FILES |
| JTYPE  | 00231+  | VARIABLE         | INTEGER | L COMMON FILES |
| KCR    | 00461+  | VARIABLE         | INTEGER | L COMMON FILES |
| KDCB   | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KNAME  | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KOUNT1 | 00504R  | VARIABLE         | INTEGER | LOCAL          |
| KOUNT2 | 00502R  | VARIABLE         | INTEGER | LOCAL          |
| KSC    | 00460+  | VARIABLE         | INTEGER | L COMMON FILES |
| KTYPE  | 00457+  | VARIABLE         | INTEGER | L COMMON FILES |
| LEN    | 00477R  | VARIABLE         | INTEGER | LOCAL          |
| LIST   | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| LUNS   | 00006X  | COMMON LABEL     |         | EXTERNAL       |

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|-------|---------|-----------------|---------|----------|-------|
| MEMOV | 00004X  | SUBROUTINE      |         | EXTERNAL |       |
| N     | 00500R  | VARIABLE        | INTEGER | LOCAL    |       |
| N1    | 00000+  | VARIABLE        | INTEGER | L COMMON | TABLE |
| NAME  | 00000R  | ARRAY, 1 DIMEN. | INTEGER | LOCAL    |       |
| NAME1 | 00002+  | ARRAY, 2 DIMEN. | INTEGER | L COMMON | TABLE |
| NAMES | 00702+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| NMAX  | 00001+  | VARIABLE        | INTEGER | L COMMON | TABLE |
| SRCHN | 00005X  | SUBROUTINE      |         | EXTERNAL |       |
| SRCHR | 00014X  | SUBROUTINE      |         | EXTERNAL |       |
| TABLE | 00017X  | COMMON LABEL    |         | EXTERNAL |       |

```

263 00044 SUBROUTINE COPYR(IDC,F,IFLAG,LEN), 09/04/81
264 00044 C
265 00044 C*** COPYR
266 00044 C**
267 00044 C** COPYR IS USED TO COPY RELOCATABLE PROGRAMS FROM THE INPUT
268 00044 C** FILES TO THE OUTPUT LIBRARY FILE.
269 00044 C**
270 00044 COMMON IBUF(128)
271 00044 COMMON/LUNS / INPUT,LIST
272 00044 COMMON/FILES/IDCB(150,2),KNAME(3),KTYPE,KSC,KCR,KDCB(144),NAMES(3)
273 00044 COMMON/TABLE/ N1,NMAX,NAME1(4,500)
274 00044 C
275 00044 C SEARCH FOR THE NEXT RELOCATABLE ROUTINE AND TEST FOR END OF
276 00044 C FILE OR ERROR. OTHERWISE SEARCH THE TABLE TO SEE IF
277 00044 C ITS NAME OCCURS. IF SO GO AND SEARCH FOR THE NEXT ROUTINE.
278 00044 C IF NOT, COPY THE NAM RECORD AND THE REST OF THE ROUTINE
279 00044 C TO THE OUTPUT RELOCATABLE LIBRARY.
280 00044 C
281 00044 10 ITYPE = IAND(ISHIF(IBUF(2),3B),7B) $ IF (ITYPE.EQ.4B) 15,20
282 00066 15 CALL BADR(IFLAG) $ IF (IFLAG.LT.0) RETURN
283 00076 20 CALL WRITF(KDCB,IER,IBUF,LEN) $ IF (IER.GE.0) GO TO 30
284 00107 25 WRITE(LIST,1000) IER $ IFLAG = -1 $ RETURN
285 00127 1000 FORMAT(5X5(1H*),5X"EMP ERROR: ",I4," ON COPY")
286 00127 30 IF (ITYPE.EQ.5B) RETURN
287 00135 CALL READF(IDCB(7,IDCF),IER,IBUF,128,LEN)
288 00151 IF (LEN) 35,40
289 00154 35 WRITE(LIST,1010) $ IFLAG = -1 $ RETURN
290 00172 1010 FORMAT(5X5(1H*),5X"EOF ERROR ON COPY")
291 00172 40 IF (IER) 25,10
292 00176 END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 138 COMMON: 128

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION |
|-------|---------|------------------|---------|----------|
| a10   | 00052R  | STATEMENT NUMBER |         |          |
| a1000 | 00000R  | STATEMENT NUMBER |         |          |
| a1010 | 00024R  | STATEMENT NUMBER |         |          |
| a15   | 00066R  | STATEMENT NUMBER |         |          |
| a20   | 00076R  | STATEMENT NUMBER |         |          |
| a25   | 00107R  | STATEMENT NUMBER |         |          |
| a30   | 00127R  | STATEMENT NUMBER |         |          |
| a35   | 00154R  | STATEMENT NUMBER |         |          |
| a40   | 00172R  | STATEMENT NUMBER |         |          |
| BADR  | 00003X  | SUBROUTINE       |         | EXTERNAL |
| FILES | 00005X  | COMMON LABEL     |         | EXTERNAL |
| IAND  |         | INTRINSIC        | GENERIC |          |
| IBUF  | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON   |
| IDCB  | 00000+  | ARRAY, 2 DIMEN.  | INTEGER | L COMMON |
| IDCF  | 00044R  | VARIABLE         | INTEGER | DUMMY    |
| IER   | 00200R  | VARIABLE         | INTEGER | LOCAL    |
| IFLAG | 00045R  | VARIABLE         | INTEGER | DUMMY    |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON |
| ISHIF | 00002X  | FUNCTION         | INTEGER | EXTERNAL |
| ITYPE | 00201R  | VARIABLE         | INTEGER | LOCAL    |
| KCR   | 00461+  | VARIABLE         | INTEGER | L COMMON |
| KDCB  | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| KNAME | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| KSC   | 00460+  | VARIABLE         | INTEGER | L COMMON |
| KTYPE | 00457+  | VARIABLE         | INTEGER | L COMMON |
| LEN   | 00046R  | VARIABLE         | INTEGER | DUMMY    |
| LI:T  | 00001+  | VARIABLE         | INTEGER | L COMMON |
| LUNS  | 00006X  | COMMON LABEL     |         | EXTERNAL |
| N1    | 00000+  | VARIABLE         | INTEGER | L COMMON |
| NAME1 | 00002+  | ARRAY, 2 DIMEN.  | INTEGER | L COMMON |
| NAMES | 00702+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| NMAX  | 00001+  | VARIABLE         | INTEGER | L COMMON |
| REFDF | 00013X  | SUBROUTINE       |         | EXTERNAL |
| TABLE | 00000X  | COMMON LABEL     |         | EXTERNAL |
| W:TF  | 00004X  | SUBROUTINE       |         | EXTERNAL |

```

293 00050      SUBROUTINE CRELF(IFLAG), 09/04/81
294 00050 C
295 00050 C***      CRELF
296 00050 C**
297 00050 C**      CRELF CREATES A TYPE 5 RELOCATABLE SCRATCH FILE
298 00050 C**      THE DCB, SECURITY CODE AND CARTRIDGE
299 00050 C**      ARE AS SUPPLIED BY THE USER IN THE /FILES/
300 00050 C**      COMMON BLOCK
301 00050 C**
302 00050      INTEGER*4 ISIZE(2)
303 00050      COMMON/LUNS / INPUT,LIST
304 00050      COMMON/FILES/ INAME(3),ITYPE,ISC,ICR,IDCB(144),
305 00050      1      JNAME(3),JTYPE,JSC,JCR,JDCB(144),
306 00050      2      KNAME(3),KTYPE,KSC,KCR,KDCB(144),NAMES(3),KSIZE
307 00050      DATA NUM,ISIZE/1,96,0/
308 00050 C
309 00050 C      PRESET IFLAG TO -1 AND ATTEMPT TO CREATE THE FILE.
310 00050 C
311 00050      IFLAG = 0 $ IF (KSIZE.NE.0) ISIZE = KSIZE
312 00066      10 CALL CRETS(KDCB,IER,NUM,NAMES,ISIZE,5,KSC,KCR)
313 00100      IF (IER.GE.0) RETURN
314 00105      35 WRITE(LIST,1000) IER
315 00121      1000 FORMAT(5X5(1H*),5X"ATTEMPT TO CREATE SCRATCH FILE FAILED.",
316 00121      1      " FMP ERROR:",I4)
317 00121      END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 87 COMMON: (NONE)

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION       |
|-------|---------|------------------|---------|----------------|
| @10   | 00066R  | STATEMENT NUMBER |         |                |
| @1000 | 00005R  | STATEMENT NUMBER |         |                |
| @35   | 00105R  | STATEMENT NUMBER |         |                |
| CRETS | 00003X  | SUBROUTINE       |         | EXTERNAL       |
| FILES | 00002X  | COMMON LABEL     |         | EXTERNAL       |
| ICR   | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDCB  | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IER   | 00124R  | VARIABLE         | INTEGER | LOCAL          |
| IFLAG | 00050R  | VARIABLE         | INTEGER | DUMMY          |
| INAME | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| ISC   | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ISIZE | 00000R  | ARRAY, 1 DIMEN.  | DBL INT | LOCAL          |
| JTYPE | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| JCR   | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB  | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JNAME | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JSC   | 00232+  | VARIABLE         | INTEGER | L COMMON FILES |
| JTYPE | 00231+  | VARIABLE         | INTEGER | L COMMON FILES |
| KCR   | 00461+  | VARIABLE         | INTEGER | L COMMON FILES |
| KDCB  | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KNAME | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KSC   | 00460+  | VARIABLE         | INTEGER | L COMMON FILES |
| KSIZE | 00705+  | VARIABLE         | INTEGER | L COMMON FILES |
| KTYPE | 00457+  | VARIABLE         | INTEGER | L COMMON FILES |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| LUNS  | 00004X  | COMMON LABEL     |         | EXTERNAL       |
| NAMES | 00702+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| NUM   | 00004R  | VARIABLE         | INTEGER | LOCAL          |

```

318 00236 SUBROUTINE FCLOS(IFLAG), 09/04/81
319 00236 C
320 00236 C*** FCLOS
321 00236 C**
322 00236 C** FCLOS HANDLES THE CLOSING OF FMP FILES FOR NORMAL OR ERROR
323 00236 C** RETURNS. ON A NORMAL RETURN THE OUTPUT LIBRARY FILE IS
324 00236 C** TRUNCATED AND CLOSED AND THEN GIVEN ITS REQUESTED NAME.
325 00236 C** IF THIS IS THE SAME AS THE INPUT LIBRARY FILE, THE INPUT
326 00236 C** LIBRARY FILE IS RENAMED BY SUBSTITUTING XX FOR THE FIRST TWO
327 00236 C** CHARACTERS OF THE FILE NAME. THEN THE OUTPUT FILE
328 00236 C** IS RENAMED AND THEN THE INPUT LIBRARY FILE IS PURGED.
329 00236 C** IF THE OUTPUT LIBRARY AND THE INPUT LIBRARY FILES HAVE
330 00236 C** DIFFERENT NAMES THE INPUT LIBRARY IS NOT PURGED.
331 00236 C** IFLAG IS NEGATIVE IF THE CLOSE IS AS A RESULT OF AN ERROR.
332 00236 C** IN THIS CASE THE INPUT FILE AND LIBRARY ARE CLOSED
333 00236 C** AND THE OUTPUT SCRATCH FILE IS PURGED.
334 00236 C** WHAT MORE COULD YOU ASK FOR?
335 00236 C**
336 00236 DIMENSION N1(3)
337 00236 COMMON/LUNS / INPUT,LIST
338 00236 COMMON/FILES/ INAME(3),ITYPE,ISC,ICR,IDCB(144),
339 00236 1 JNAME(3),JTYPE,JSC,JCR,JDCB(144),
340 00236 2 KNAME(3),KTYPE,KSC,KCR,KDCB(144),NAMES(3)
341 00236 DATA N1/ 'XX' , ' ' /
342 00236 C
343 00236 C TEST FOR ERROR CONDITION
344 00236 C
345 00236 IF (IFLAG) 10,40
346 00242 C
347 00242 C PURGE THE SCRATCH FILE AND TEST FOR ERROR, THEN CLOSE
348 00242 C THE INPUT LIBRARY AND FILE.
349 00242 C
350 00245 10 CALL PURGE(KDCB,IER,NAMES,KSC,'R')
351 00254 IF (IER.EQ.-6) GO TO 25 $ 1. (IER) 15,20
352 00262 15 WRITE(LIST,1000) IER,NAME $ CALL CLOSE(KDCB) $ GO TO 25
353 00304 1000 FORMAT(5X5(1H*),5X"FMP ERROR:",14," ON ATTEMPTED PURGE OF ",
354 00304 1 3A2)
355 00304 20 WRITE(LIST,1010)
356 00316 1010 FORMAT(5X5(1H*),5X"SCRATCH FILE PURGED")
357 00316 25 CALL CLOSE(JDCB)
358 00321 30 CALL CLOSE(IDCB) $ RETURN
359 00324 C
360 00324 C TRUNCATE THE SCRATCH FILE, CHECKING FOR ERROR.
361 00324 C
362 00326 40 CALL LOCF(KDCB,IER,1,IRB,1,JSEC)
363 00336 IF (IER) 45,48
364 00341 45 WRITE(LIST,1020) IER $ GO TO 25
365 00356 1020 FORMAT(5X5(1H*),5X"FMP ERROR:",14," ON LOCF CALL")
366 00356 48 ITRUN = JSEC/2 - (IRB + 1) $ CALL CLOSE(KDCB,IER,ITRUN)
367 00374 IF (IER) 50,55
368 00377 50 WRITE(LIST,1030) IER $ GO TO 25
369 00414 1030 FORMAT(5X5(1H*),5X"FMP ERROR:",14," ON FILE TRUNCATION CLOSE")
370 00414 55 KILL = 1 $ IF (ICBYT(JNAME,0,KNAME,0,6).LT.0) GO TO 65
371 00427 KILL = -1 $ N1(2) = JNAME(2) $ N1(3) = JNAME(3)
372 00433 C

```

```

373 00433 C      RENAME THE INPUT LIBRARY
374 00433 C
375 00435      CALL NAMF(JDCB,IER,JNAME,N1,KSC,KCR)
376 00445      IF (IER) 60,65
377 00450      60 WRITE(LIST,1040) IER,JNAME $ GO TO 10
378 00470      1040 FORMAT(5X5(1H*),5X"FMP ERROR:",I4," ON RENAME OF ",3A2)
379 00470 C
380 00470 C      RENAME THE SCRATCH FILE
381 00470 C
382 00470      65 CALL NAMF(KDCB,IER,NAMES,KNAME,KSC,KCR)
383 00500      IF (IER) 66,70
384 00503      66 WRITE(LIST,1040) IER,KNAME $ GO TO 10
385 00523      70 WRITE(LIST,1050) KNAME $ IF (KILL) 80,25
386 00543      1050 FORMAT(10X"NEW LIBRARY CATALOGUED AS: ",3A2)
387 00543 C
388 00543 C      PURGE THE RENAMED INPUT LIBRARY.
389 00543 C
390 00543      80 CALL PURGE(JDCB,IER,N1,KSC,KCR) $ IF (IER.LT.0) GO TO 15
391 00555      WRITE(LIST,1060) $ GO TO 30
392 00570      1060 FORMAT(10X"OLD LIBRARY FILE PURGED")
393 00570      END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 388 COMMON: (NONE)

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION       |
|-------|---------|------------------|---------|----------------|
| a10   | 00245R  | STATEMENT NUMBER |         |                |
| a1000 | 00003R  | STATEMENT NUMBER |         |                |
| a1010 | 00040R  | STATEMENT NUMBER |         |                |
| a1020 | 00061R  | STATEMENT NUMBER |         |                |
| a1030 | 00107R  | STATEMENT NUMBER |         |                |
| a1040 | 00143R  | STATEMENT NUMBER |         |                |
| a1050 | 00174R  | STATEMENT NUMBER |         |                |
| a1060 | 00217R  | STATEMENT NUMBER |         |                |
| a15   | 00262R  | STATEMENT NUMBER |         |                |
| a20   | 00304R  | STATEMENT NUMBER |         |                |
| a25   | 00316R  | STATEMENT NUMBER |         |                |
| a30   | 00321R  | STATEMENT NUMBER |         |                |
| a40   | 00326R  | STATEMENT NUMBER |         |                |
| a41   | 00341R  | STATEMENT NUMBER |         |                |
| a43   | 00356R  | STATEMENT NUMBER |         |                |
| a50   | 00377R  | STATEMENT NUMBER |         |                |
| a55   | 00414R  | STATEMENT NUMBER |         |                |
| a60   | 00450R  | STATEMENT NUMBER |         |                |
| a65   | 00470R  | STATEMENT NUMBER |         |                |
| a66   | 00503R  | STATEMENT NUMBER |         |                |
| a70   | 00523R  | STATEMENT NUMBER |         |                |
| a80   | 00543R  | STATEMENT NUMBER |         |                |
| CLOSE | 00011X  | SUBROUTINE       |         | EXTERNAL       |
| FILES | 00003X  | COMMON LABEL     |         | EXTERNAL       |
| I     | 00576R  | VARIABLE         | INTECER | LOCAL          |
| ICBYT | 00013X  | FUNCTION         | INTEGER | EXTERNAL       |
| ICR   | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDCB  | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IER   | 00571R  | VARIABLE         | INTEGER | LOCAL          |
| IFLAG | 00236R  | VARIABLE         | INTEGER | DUMMY          |
| INAME | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| IRB   | 00574R  | VARIABLE         | INTEGER | LOCAL          |
| ISC   | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ITRUN | 00573R  | VARIABLE         | INTEGER | LOCAL          |
| ITYPE | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| JCR   | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB  | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JNAME | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JSC   | 00232+  | VARIABLE         | INTEGER | L COMMON FILES |
| JSEC  | 00575R  | VARIABLE         | INTEGER | LOCAL          |
| JTYPE | 00231+  | VARIABLE         | INTEGER | L COMMON FILES |
| KCR   | 00461+  | VARIABLE         | INTEGER | L COMMON FILES |
| KDCB  | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KILL  | 00572R  | VARIABLE         | INTEGER | LOCAL          |
| KNAME | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KSC   | 00460+  | VARIABLE         | INTEGER | L COMMON FILES |
| KTYPE | 00457+  | VARIABLE         | INTEGER | L COMMON FILES |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| LOCF  | 00012X  | SUBROUTINE       |         | EXTERNAL       |

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|-------|---------|-----------------|---------|----------|-------|
| LUNS  | 00004X  | COMMON LABEL    |         | EXTERNAL |       |
| N1    | 00000R  | ARRAY, 1 DIMEN. | INTEGER | LOCAL    |       |
| NAME  | 00577R  | VARIABLE        | INTEGER | LOCAL    |       |
| NAMES | 00702+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| NAMF  | 00014X  | SUBROUTINE      |         | EXTERNAL |       |
| PURGE | 00002X  | SUBROUTINE      |         | EXTERNAL |       |

```

394 00031      SUBROUTINE FINDR(IDCIB,IFLAG,LEN), 09/04/81
395 00031 C
396 00031 C***      FINDR
397 00031 C**
398 00031 C**      FINDR SEARCHES A RELOCATABLE FILE FOR THE NEXT NAM RECORD
399 00031 C**      AND LEAVES IT IN THE WORKSPACE BUFFER IBUF. IFLAG IS SET TO 1
400 00031 C**      IF A NAM RECORD IS FOUND, TO 0 IF AN EOF IS FOUND AND TO -1
401 00031 C**      IF AN ERROR OCCURS.
402 00031 C**
403 00031      DIMENSION IDCIB(144)
404 00031      COMMON IBUF(128)
405 00031      COMMON/LUNS / INPUT,LIST
406 00031 C
407 00031 C      PRESET IFLAG, READ THE NEXT RECORD. TEST FOR EOF OR ERROR.
408 00031 C      TEST RECORD TYPE. IF NAM RECORD RETURN, OTHERWISE READ NEXT
409 00031 C      RECORD
410 00031 C
411 00031      10 CALL READF(IDCIB,IER,IBUF,128,LEN)
412 00051      IFLAG = 0 $ IF (LEN.LT.0) RETURN
413 00060      IF (IER.GE.0) GO TO 20
414 00063      WRITE(LIST,1000) IER $ IFLAG = -1 $ RETURN
415 00103 1000 FORMAT(5X5(1H*),5X FMP ERROR: ,14, DURING FILE SEARCH )
416 00103      20 IDENT = 1AND(ISHIF(IBUF(2),3B),7B)
417 00111      IF (IDENT-1) 10,30,10
418 00115      30 IFLAG = 1
419 00117      END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 89 COMMON: 128

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION      |
|-------|---------|------------------|---------|---------------|
| a10   | 00042R  | STATEMENT NUMBER |         |               |
| a1000 | 00000R  | STATEMENT NUMBER |         |               |
| a20   | 00103R  | STATEMENT NUMBER |         |               |
| a30   | 00115R  | STATEMENT NUMBER |         |               |
| IAND  |         | INTRINSIC        | GENERIC |               |
| IBUF  | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON        |
| IDCB  | 00031R  | ARRAY, 1 DIMEN.  | INTEGER | DUMMY         |
| IDENT | 00123R  | VARIABLE         | INTEGER | LOCAL         |
| IER   | 00124R  | VARIABLE         | INTEGER | LOCAL         |
| IFLAG | 00032R  | VARIABLE         | INTEGER | DUMMY         |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS |
| ISHIF | 00010X  | FUNCTION         | INTEGER | EXTERNAL      |
| LEN   | 00033R  | VARIABLE         | INTEGER | DUMMY         |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS |
| LUNS  | 00003X  | COMMON LABEL     |         | EXTERNAL      |
| READF | 00002X  | SUBROUTINE       |         | EXTERNAL      |

```

420 00106      SUBROUTINE GETIN(IFLAG), 09/04/81
421 00106 C
422 00106 C***      GETIN
423 00106 C**
424 00106 C**      GETIN GETS THE NEXT INSTRUCTION FOR UDOIT TO EXECUTE.
425 00106 C**      INSTRUCTIONS MAY COME FROM A LOGIL UNIT OR A FILE.
426 00106 C**      IN EITHER CASE IT IS TRANSPARENT TO UDOIT. A COMMAND
427 00106 C**      TO TRANSFER FROM ONE INPUT SOURCE TO ANOTHER IS HANDLED BY GETIN
428 00106 C**
429 00106 C**      FORMAL PARAMETERS:
430 00106 C**          IFLAG : IF SET TO -1 AN ERROR HAS OCCURRED
431 00106 C*
432 00106      INTEGER*4 END,JT,TRAN
433 00106      DIMENSION IA(2)
434 00106      COMMON/LJNS / INPUT,LIST
435 00106      COMMON/FILES/ INAME(3),ITYPE,ISC,ICR,IDCB(144),
436 00106      1 JNAME(3),JTYPE,JSC,JCR,JDCB(144),
437 00106      2 KNAME(3),KTYPE,KSC,KCR,KDCB(144)
438 00106      COMMON/WORKS/ KBUF(40),JBUF(10),LOG,ISTRC
439 00106      EQUIVALENCE (IA,REG),(IB,IA(2)),(JT,JBUF)
440 00106      DATA JFLAG,LUCOM/1,0/, TRAN,END/ 'TRAN' , 'END ' /
441 00106 C
442 00106 C      TEST FOR ERROR CONDITION, AND THEN SELECT EITHER
443 00106 C      INITIALISATION, L.U. INPUT OR FILE INPUT
444 00106 C
445 00106      IF (IFLAG) 350,10,20
446 00117      10 IF (LUCOM) 200,445,100
447 00125      20 IF (IAND(ITYPE,3B)-1) 445,30,425
448 00134      30 LUCOM = INAME
449 00134 C
450 00134 C      L.U.INPUT
451 00134 C
452 00136      100 CALL INTST(KBUF,40,2H )
453 00143      IF (ITTY.LT.0) WRITE(LUCOM,1010)
454 00160      1010 FORMAT(2X'LIBED? ')
455 00160      REG = EXEC(1,LUCOM+400B,KBUF,-80) $ LOG = IB $ GO TO 300
456 00175 C
457 00175 C      FILE INPUT
458 00175 C
459 00176      200 CALL INTST(KBUF,40,2H )
460 00203      CALL READF(IDCB,IER,KBUF,40,LOG)
461 00212      IF (LEN) 205,210
462 00215      205 JT = END $ RETURN
463 00223      210 IF (IER) 450,220
464 00226      220 LOG = LOG*2
465 00226 C
466 00226 C      PARSE THE STRING, CHECK FOR A TRANSFER INSTRUCTION
467 00226 C      WHICH IS HANDLED IN THIS PROGRAM, AND RETURN
468 00226 C
469 00231      300 ISTRC = 1
470 00233      IF (NAMR(JBUF,KBUF,LOG,ISTRC)) 350,310
471 00243      310 IF (IAND(JBUF(4),3B).NE.3) GO TO 350
472 00250      IF (JT.EQ.TRAN) GO TO 400 $ IFLAG = 0 $ RETURN
473 00260 C
474 00260 C      ERROR SECTION. IF INPUT IS FROM AN INTERACTIVE DEVICE

```

```

475 00260 C      THEN REPORT AND GET NEW INPUT. OTHERWISE ABORT.
476 00260 C
477 00262      350 IF (LUCOM.GT.C.AND.ITTY.LT.0) GO TO 360
478 00271      WRITE(LIST,1020) $ GO TO 460
479 00304      1020 FORMAT(5X5(1H*),5X"INPUT ERROR!")
480 00304      360 WRITE(LUCOM,1030) $ IFLAG = 0 $ GO TO 100
481 00321      1030 FORMAT(" LIBED: INPUT ERROR")
482 00321 C
483 00321 C      THIS SECTION HANDLES INPUT SOURCE CHANGES
484 00321 C
485 00321      400 IF (NAMR(KBUF,JBUF,LOG,ISTR)) 440,410
486 00331      410 IF (IAND(JBUF(4),3B)-1) 440,430,420
487 00340      420 CALL CLOSE(IDCIB) $ CALL MEMOV(JBUF,INAME,5)
488 00350      425 CALL OPEN(IDCIB,IER,INAME,0,ISC,ICR)
489 00360      IF (IER) 450,441
490 00364      430 LUCOM = JBUF(1)
491 00366      435 ITTY = IFTTY(LUCOM) $ GO TO 100
492 00373      440 IF (LUCOM.LT.0) GO TO 445
493 00376      441 LUCOM = -1 $ GO TO 200
494 00401      445 LUCOM = INPUT $ GO TO 435
495 00403 C
496 00403 C      FMP ERRORS
497 00403 C
498 00404      450 WRITE(LIST,1040) IER,INAME,ISC,ICR
499 00427      1040 FORMAT(5X5(1H*),5X"FMP ERROR:",I4," ON FILE ",3A2,"",I5,"",I5)
500 00427      460 CALL FCLOS(-1) $ CALL EXEC(6)
501 00435      END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 303 COMMON: (NONE)

## SYMROL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION       |
|-------|---------|------------------|---------|----------------|
| a10   | 00117R  | STATEMENT NUMBER |         |                |
| a100  | 00136R  | STATEMENT NUMBER |         |                |
| a1010 | 00010R  | STATEMENT NUMBER |         |                |
| a1020 | 00017R  | STATEMENT NUMBER |         |                |
| a1030 | 00035R  | STATEMENT NUMBER |         |                |
| a1040 | 00051R  | STATEMENT NUMBER |         |                |
| a20   | 00125R  | STATEMENT NUMBER |         |                |
| a200  | 00176R  | STATEMENT NUMBER |         |                |
| a205  | 00215R  | STATEMENT NUMBER |         |                |
| a210  | 00223R  | STATEMENT NUMBER |         |                |
| a220  | 00226R  | STATEMENT NUMBER |         |                |
| a30   | 00134R  | STATEMENT NUMBER |         |                |
| a300  | 00231R  | STATEMENT NUMBER |         |                |
| a310  | 00243R  | STATEMENT NUMBER |         |                |
| a350  | 00262R  | STATEMENT NUMBER |         |                |
| a360  | 00304R  | STATEMENT NUMBER |         |                |
| a400  | 00321R  | STATEMENT NUMBER |         |                |
| a410  | 00331R  | STATEMENT NUMBER |         |                |
| a420  | 00340R  | STATEMENT NUMBER |         |                |
| a425  | 00350R  | STATEMENT NUMBER |         |                |
| a430  | 00364R  | STATEMENT NUMBER |         |                |
| a435  | 00366R  | STATEMENT NUMBER |         |                |
| a440  | 00373R  | STATEMENT NUMBER |         |                |
| a441  | 00376R  | STATEMENT NUMBER |         |                |
| a445  | 00401R  | STATEMENT NUMBER |         |                |
| a450  | 00404R  | STATEMENT NUMBER |         |                |
| a460  | 00427R  | STATEMENT NUMBER |         |                |
| CLOSE | 00015X  | SUBROUTINE       |         | EXTERNAL       |
| END   | 00006R  | VARIABLE         | DBL INT | LOCAL          |
| EXEC  |         | INTRINSIC        | REAL    |                |
| FCLOS | 00023X  | SUBROUTINE       |         | EXTERNAL       |
| FILES | 00002X  | COMMON LABEL     |         | EXTERNAL       |
| IA    | 00000R  | ARRAY, 1 DIMEN.  | INTEGER | LOCAL          |
| IAND  |         | INTRINSIC        | GENERIC |                |
| IB    | 00001R  | VARIABLE         | INTEGER | LOCAL          |
| ICR   | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDCB  | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IER   | 00441R  | VARIABLE         | INTEGER | LOCAL          |
| IFLAG | 00106R  | VARIABLE         | INTEGER | DUMMY          |
| IFTTY | 00020X  | FUNCTION         | INTEGER | EXTERNAL       |
| INAME | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| INTST | 00003X  | SUBROUTINE       |         | EXTERNAL       |
| ISC   | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ISTRC | 00063+  | VARIABLE         | INTEGER | L COMMON WORKS |
| ITTY  | 00442R  | VARIABLE         | INTEGER | LOCAL          |
| ITYPE | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| JBUF  | 00050+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON WORKS |
| JCR   | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB  | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|-------|---------|-----------------|---------|----------|-------|
| JFLAG | 00002R  | VARIABLE        | INTEGER | LOCAL    |       |
| JNAME | 00226+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| JSC   | 00232+  | VARIABLE        | INTEGER | L COMMON | FILES |
| JT    | 00050+  | VARIABLE        | DBL INT | L COMMON | WORKS |
| JTYPE | 00231+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KBUF  | 00000+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | WORKS |
| KCR   | 00461+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KDCB  | 00462+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| KNAME | 00454+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| KSC   | 00460+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KTYPE | 00457+  | VARIABLE        | INTEGER | L COMMON | FILES |
| LEN   | 00443R  | VARIABLE        | INTEGER | LOCAL    |       |
| LIST  | 00001+  | VARIABLE        | INTEGER | L COMMON | LUNS  |
| LOG   | 00062+  | VARIABLE        | INTEGER | L COMMON | WORKS |
| LUCOM | 00003R  | VARIABLE        | INTEGER | LOCAL    |       |
| LUNS  | 00014X  | COMMON LABEL    |         | EXTERNAL |       |
| MEMOV | 00016X  | SUBROUTINE      |         | EXTERNAL |       |
| NAMR  | 00012X  | FUNCTION        | INTEGER | EXTERNAL |       |
| OPEN  | 00017X  | SUBROUTINE      |         | EXTERNAL |       |
| READF | 00011X  | SUBROUTINE      |         | EXTERNAL |       |
| REG   | 00000R  | VARIABLE        | REAL    | LOCAL    |       |
| TRAN  | 00004R  | VARIABLE        | DBL INT | LOCAL    |       |
| WORKS | 00004X  | COMMON LABEL    |         | EXTERNAL |       |

```

502 00001        SUBROUTINE INSRT(LEN), 10/04/81
503 00001 C
504 00001 C***        INSRT
505 00001 C**
506 00001 C**        INSRT INSERTS THE CURRENT DATE AND OUTPUT LIBRARY
507 00001 C**        NAME IN THE COMMENT REGION OF THE NAM RECORD OF AN INPUT
508 00001 C**        RELOCATABLE ROUTINE.
509 00001 C**
510 00001 C*        FORMAL PARAMETER:
511 00001 C*        LEN IS THE LENGTH OF THE NAM RECORD
512 00001 C*        IT IS RESET BY THIS ROUTINE.
513 00001 C*
514 00001        COMMON IBUF(128)
515 00001        COMMON/FILES/ INAME(3), ITYPE, ISC, ICR, IDCB(144),
516 00001        1        JNAME(3), JTYPE, JSC, JCR, JDCB(144),
517 00001        2        KNAME(3), KTYPE, KSC, KCR, KDCB(144)
518 00001        COMMON/TODAY/ IDATE(15)
519 00001        DATA IBLNK/ ' ' /
520 00001 C
521 00001 C        TEST IF RECORD IS TOO LONG. THIS PROGRAM INSERTS 15
522 00001 C        WORDS AND THE MAXIMUM LENGTH OF A NAM RECORD IS 60
523 00001 C        WORDS. IF THERE IS NOT ROOM FOR ALL THE INFORMATION, NO
524 00001 C        INSERTION IS DONE
525 00001 C
526 00001        IF (LEN-45) 10,20
527 00011 10 CALL CMOVE(IBLNK, 0, IBUF(LEN), 2,2)
528 00023        CALL CMOVE(IDATE, 0, IBUF(LEN), 4,8)
529 00035        CALL CMOVE(IDATE, 15, IBUF(LEN), 12,3)
530 00047        CALL CMOVE(IDATE, 19, IBUF(LEN), 15,4)
531 00061        CALL CMOVE(IDATE, 25, IBUF(LEN), 19,5)
532 00073        CALL CMOVE(IBLNK, 0, IBUF(LEN), 24,2)
533 00105        CALL CMOVE(KNAME, 0, IBUF(LEN), 26,6)
534 00117        LEN = LEN + 15
535 00117 C
536 00117 C        NOW RESET THE LENGTH WORDS AND CHECKSUM
537 00117 C
538 00122        IBUF(1) = LEN*256 $ IBUF(3) = IBUF(2)
539 00127        DO 18 I=4,LEN
540 00131        18 IBUF(3) = IBUF(3) + IBUF(1)
541 00145        20 END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 119 COMMON: 128

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION |
|-------|---------|------------------|---------|----------|
| a10   | 00011R  | STATEMENT NUMBER |         |          |
| a18   | 00131R  | STATEMENT NUMBER |         |          |
| a20   | 00145R  | STATEMENT NUMBER |         |          |
| CMOVE | 00002X  | SUBROUTINE       |         | EXTERNAL |
| FILES | 00004X  | COMMON LABEL     |         | EXTERNAL |
| I     | 00147R  | VARIABLE         | INTEGER | LOCAL    |
| IBLNK | 00000R  | VARIABLE         | INTEGER | LOCAL    |
| IBUF  | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON   |
| ICR   | 00005+  | VARIABLE         | INTEGER | L COMMON |
| IDATE | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| IDCB  | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| INAME | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| ISC   | 00004+  | VARIABLE         | INTEGER | L COMMON |
| ITYPE | 00003+  | VARIABLE         | INTEGER | L COMMON |
| JCR   | 00233+  | VARIABLE         | INTEGER | L COMMON |
| JDCB  | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| JNAME | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| JSC   | 00232+  | VARIABLE         | INTEGER | L COMMON |
| JTYPE | 00231+  | VARIABLE         | INTEGER | L COMMON |
| KCR   | 00461+  | VARIABLE         | INTEGER | L COMMON |
| KDCB  | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| KNAME | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON |
| KSC   | 00460+  | VARIABLE         | INTEGER | L COMMON |
| KTYPE | 00457+  | VARIABLE         | INTEGER | L COMMON |
| LEN   | 00001R  | VARIABLE         | INTEGER | DUMMY    |
| TODAY | 00003X  | COMMON LABEL     |         | EXTERNAL |

```

542 00104      SUBROUTINE LIBPR, 18/06/80
543 00104 C
544 00104 C***      LIBPR
545 00104 C*-
546 00104 C**      LIBPR LISTS THE CONTENTS OF A RELOCATABLE LIBRARY.
547 00104 C**      ALL NAM, ENT AND EXT RECORDS ARE LISTED.
548 00104 C**
549 00104      COMMON IBUF(128)
550 00104      COMMON/LUNS / INPUT,LIST
551 00104      COMMON/FILES/ INAME(3),ITYPE,ISC,ICR,IDCB(144),
552 00104      1      JNAME(3),JTYPE,JSC,JCR,JDCB(144),
553 00104      2      KNAME(3),KTYPE,KSC,KCR,KDCB(144)
554 00104      COMMON/TODAY/ IDATE(15)
555 00104 C
556 00104 C      PRESET THE LINE COUNT AND PAGE NUMBER PARAMETERS, AND
557 00104 C      OPEN THE FILE AND CHECK FOR ERROR.
558 00104 C
559 00104      IPAGE = 1
560 00111      LKNT = 60
561 00113      20 CALL READF(KDCB,IER,IBUF,128,LEN)
562 00122          IF (LEN.LT.0) RETURN
563 00127          IF (IER) 25,30
564 00132      25 WRITE(LIST,26) IER
565 00146      26 FORMAT(5X5(1H*),5X"FMP ERROR:",I4," ON LIBRARY READ")
566 00146          RETURN
567 00150      30 IF (LKNT.LT.45) GO TO 40
568 00157          IF (IPAGE.GT.1) WRITE(LIST,31)
569 00200      31 FORMAT(1H1)
570 00200          WRITE(LIST,35)KNAME,IDATE,IPAGE
571 00222      35 FORMAT(///30X"LIST OF RELOCATABLE LIBRARY FILE: ",3A2,
572 00222      1      10X15A2,5X"PAGE:",I3/30X40(1H*)//)
573 00222          LKNT = 1
574 00224          IPAGE = IPAGE + 1
575 00227      40 ITYPE = IAND(ISHIF(IBUF(2),3B),7B)
576 00235          CALL LISTE(ITYPE,LEN)
577 00241          IF (ITYPE.EQ.2.OR.ITYPE.EQ.4) LKNT = LKNT + 1
578 00257          IF (ITYPE.EQ.1) LKNT = LKNT + 2
579 00266          GO TO 20
580 00267          END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 197 COMMON: 128

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION       |
|-------|---------|------------------|---------|----------------|
| a20   | 00113R  | STATEMENT NUMBER |         |                |
| a25   | 00132R  | STATEMENT NUMBER |         |                |
| a26   | 00000R  | STATEMENT NUMBER |         |                |
| a30   | 00150R  | STATEMENT NUMBER |         |                |
| a31   | 00030R  | STATEMENT NUMBER |         |                |
| a35   | 00033R  | STATEMENT NUMBER |         |                |
| a40   | 00227R  | STATEMENT NUMBER |         |                |
| FILES | 00002X  | COMMON LABEL     |         | EXTERNAL       |
| IAND  |         | INTRINSIC        | GENERIC |                |
| IBUF  | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON         |
| ICR   | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDATE | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON TODAY |
| IDCB  | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IER   | 00273R  | VARIABLE         | INTEGER | LOCAL          |
| INAME | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| IPAGE | 00272R  | VARIABLE         | INTEGER | LOCAL          |
| ISC   | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ISHIF | 00012X  | FUNCTION         | INTEGER | EXTERNAL       |
| ITYPE | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| JCR   | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB  | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JNAME | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JSC   | 00232+  | VARIABLE         | INTEGER | L COMMON FILES |
| JTYPE | 00231+  | VARIABLE         | INTEGER | L COMMON FILES |
| KCR   | 00461+  | VARIABLE         | INTEGER | L COMMON FILES |
| KDCB  | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KNAME | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KSC   | 00460+  | VARIABLE         | INTEGER | L COMMON FILES |
| KTYPE | 00457+  | VARIABLE         | INTEGER | L COMMON FILES |
| LEN   | 00271R  | VARIABLE         | INTEGER | LOCAL          |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| LISTE | 00013X  | SUBROUTINE       |         | EXTERNAL       |
| LKNT  | 00270R  | VARIABLE         | INTEGER | LOCAL          |
| LUNS  | 00003X  | COMMON LABEL     |         | EXTERNAL       |
| READ  | 00001X  | SUBROUTINE       |         | EXTERNAL       |
| TODAY | 00011X  | COMMON LABEL     |         | EXTERNAL       |

```

581 00173      SUBROUTINE LISTE(ITYPE,LEN), 18/06/80
582 00173 C
583 00173 C***      LISTE
584 00173 C**
585 00173 C**      LISTE LISTS DETAILS OF RELOCATABLE RECORD TYPES ON THE LIST
586 00173 C *      LOGICAL UNIT. IF ITYPE = 1, THE RECORD IN THE WORKSPACE BUFFER
587 00173 C *      IS A NAM RECORD, AND THE NAME, PROGRAM TYPE ETC. ARE LISTED.
588 00173 C**      IF ITYPE = 2, THE RECORD IS AN ENT RECORD, AND IF ITYPE = 4
589 00173 C**      THE RECORD IS AN EXT RECORD. THE ENTRY POINTS OR EXTERNALS
590 00173 C**      ARE LISTED.
591 00173 C**
592 00173      COMMON IBUF(128)
593 00173      COMMON/LUNS / INPUT,LIST
594 00173 C
595 00173 C      TRANSFER TO THE APPROPRIATE CODE ACCORDING TO ITYPE.
596 00173 C
597 00173      GO TO (100,200,250,400,250) ITYPE
598 00200 C
599 00200 C      LIST NAM RECORD
600 00200 C
601 00210      100 MPL = IAND(IBUF(7),77777B)
602 00213      IF (LEN.LT.18) 101,105
603 00222      101 WRITE(LIST,110) (IBUF(1),I=4,6),IBUF(10),IBUF(11),MPL,IBUF(9)
604 00261      RETURN
605 00263      105 WRITE(LIST,111) (IBUF(1),I=4,6),IBUF(10),IBUF(11),MPL,IBUF(9),
606 00316      1      (IBUF(1),I=18,LEN)
607 00340      RETURN
608 00342      110 FORMAT(/2X"PROGRAM: ",2A2,A1,2X"TYPE=",I3," , PRIORITY=",
609 00342      1      I5," , LENGTH=",I5," , COMMON=",I5)
610 00342      111 FORMAT(/2X"PROGRAM: ",2A2,A1,2X"TYPE=",I3," , PRIORITY=",
611 00342      1      I5," , LENGTH=",I5," , COMMON=",I5," ,",30A2)
612 00342 C      LIST ENTRY POINTS
613 00342 C
614 00342      200 NENTS = IAND(IBUF(2),17B)
615 00345      WRITE(LIST,210) (IBUF(4*J),IBUF(4*J+1),IBUF(4*J+2),J=1,NENTS)
616 00414      210 FORMAT(2X"ENTRY POINTS: ",14(1X2A2,A1))
617 00414      250 RETURN
618 00416      400 NENTS = IAND(IBUF(2),37B)
619 00421      WRITE(LIST,410) (IBUF(3*J+1),IBUF(3*J+2),IBUF(3*J+3),J=1,NENTS)
620 00474      410 FORMAT(2X,"EXTERNALS: ",19(1X2A2,A1))
621 00474      END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRA: 334 COMMON: 128

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION      |
|-------|---------|------------------|---------|---------------|
| a100  | 00210R  | STATEMENT NUMBER |         |               |
| a101  | 00222R  | STATEMENT NUMBER |         |               |
| a105  | 00263R  | STATEMENT NUMBER |         |               |
| a110  | 00000R  | STATEMENT NUMBER |         |               |
| a111  | 00052R  | STATEMENT NUMBER |         |               |
| a200  | 00342R  | STATEMENT NUMBER |         |               |
| a210  | 00131R  | STATEMENT NUMBER |         |               |
| a250  | 00414R  | STATEMENT NUMBER |         |               |
| a400  | 00416R  | STATEMENT NUMBER |         |               |
| a410  | 00152R  | STATEMENT NUMBER |         |               |
| I     | 00501R  | VARIABLE         | INTEGER | LOCAL         |
| IAND  |         | INTRINSIC        | GENERIC |               |
| IBUF  | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON        |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS |
| ITYPE | 00173R  | VARIABLE         | INTEGER | DUMMY         |
| J     | 00500R  | VARIABLE         | INTEGER | LOCAL         |
| LEN   | 00174R  | VARIABLE         | INTEGER | DUMMY         |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS |
| LUNS  | 00003X  | COMMON LABEL     |         | EXTERNAL      |
| MPL   | 00502R  | VARIABLE         | INTEGER | LOCAL         |
| NENTS | 00477R  | VARIABLE         | INTEGER | LOCAL         |

```
622 00000 SUBROUTINE NEWRL, 09/04/81
623 00000 C
624 00000 C*** NEWRL
625 00000 C**
626 00000 C** THIS SUBROUTINE SUPERVISES THE CREATION OF A NEW
627 00000 C** PROGRAM LIBRARY FROM AN INPUT RELOCATABLE FILE AND A
628 00000 C** RELOCATABLE LIBRARY FILE.
629 00000 C**
630 00000 COMMON/LUNS / INPUT,LIST
631 00000 COMMON/FILES/ INAME(3),ITYPE,ISC,ICR,IDCB(144),
632 00000 1 JNAME(3),JTYPE,JSC,JCR,JDCB(144),
633 00000 2 KNAME(3),KTYPE,KSC,KCR,KDCB(144)
634 00000 C
635 00000 C OPEN THE INPUT FILE AND THE LIBRARY FILE, AND CREATE
636 00000 C THE OUTPUT LIBRARY FILE WITH DUMMY NAME.
637 00000 C
638 00000 ILST = 0
639 00005 CALL ROPEN(INAME,IDCB,ISC,ICR,IFLAG)
640 00014 IF (IFLAG.LT.0) GO TO 100
641 00017 CALL ROPEN(JNAME,JDCB,JSC,JCR,IFLAG)
642 00026 IF (IFLAG.LT.0) GO TO 100
643 00031 CALL CRELF(IFLAG) $ IF (IFLAG.LT.0) GO TO 100
644 00037 C
645 00037 C SCAN THE INPUT FILE THEN
646 00037 C COPY FROM THE LIBRARY FILE.
647 00037 C
648 00037 CALL SCANR(IFLAG) $ IF (IFLAG.LT.0) GO TO 100
649 00045 CALL COPLB(IFLAG)
650 00050 100 CALL FCLOS(IFLAG)
651 00053 END
```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO EPRORS \*\* PROGRAM: 47 COMMON: (NONE)

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION       |
|-------|---------|------------------|---------|----------------|
| @100  | 00050R  | STATEMENT NUMBER |         |                |
| COPLB | 00005X  | SUBROUTINE       |         | EXTERNAL       |
| CRELF | 00003X  | SUBROUTINE       |         | EXTERNAL       |
| FCLOS | 00006X  | SUBROUTINE       |         | EXTERNAL       |
| FILES | 00002X  | COMMON LABEL     |         | EXTERNAL       |
| ICR   | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDCB  | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IFLAG | 00055R  | VARIABLE         | INTEGER | LOCAL          |
| ILST  | 00056R  | VARIABLE         | INTEGER | LOCAL          |
| INAME | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| ISC   | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ITYPE | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| JCR   | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB  | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JNAME | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JSC   | 00232+  | VARIABLE         | INTEGER | L COMMON FILES |
| JTYPE | 00231+  | VARIABLE         | INTEGER | L COMMON FILES |
| KCR   | 00461+  | VARIABLE         | INTEGER | L COMMON FILES |
| KDCB  | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KNAME | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KSC   | 00460+  | VARIABLE         | INTEGER | L COMMON FILES |
| KTYPE | 00457+  | VARIABLE         | INTEGER | L COMMON FILES |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| LUNS  | 00000X  | COMMON LABEL     |         | EXTERNAL       |
| ROPEN | 00001X  | SUBROUTINE       |         | EXTERNAL       |
| SCANR | 00004X  | SUBROUTINE       |         | EXTERNAL       |

```

652 00144      SUBROUTINE ROPEN(NAME,IDCB,ISECU,ICR,IFLAG), 09/04/81
653 00144 C
654 00144 C***      ROPEN
655 00144 C**
656 00144 C**      THIS SUBROUTINE OPENS A RELOCATABLE FILE "NAME"
657 00144 C**      WITH SECURITY CODE ISECU ON CARTRIDGE ICR. THE FILE
658 00144 C**      IS ATTACHED TO DATA CONTROL BLOCK "IDCB". IFLAG IS
659 00144 C**      SET TO -1 IF THERE IS AN ERROR. THE FILE TYPE IS CHECKED
660 00144 C**      TO ENSURE THAT IT IS TYPE 5.
661 00144 C**
662 00144 C**      FORMAL PARAMETERS:
663 00144 C**          NAME => RELOCATABLE FILE TO BE OPENED.
664 00144 C**          IDCB => DCB TO BE USED.
665 00144 C**          ISECU => SECURITY CODE
666 00144 C**          ICR  => CARTRIDGE NUMBER
667 00144 C**          IFLAG => ERROR FLAG
668 00144 C**
669 00144      DIMENSION NAME(3),IDCB(144)
670 00144      COMMON/LUNS /INPUT,LIST
671 00144 C
672 00144 C      PRESET IFLAG TO -1 AND ATTEMPT TO OPEN THE FILE.
673 00144 C      TEST FOR "FILE NOT FOUND" CONDITION
674 00144 C
675 00144      IFLAG = -1 $ CALL OPEN(IDCB,IER,NAME,0,ISECU,ICR)
676 00174      IF (IER.EQ.-6) 10,20
677 00202      10 WRITE(LIST,1000) NAME,ISECU,ICR $ RETURN
678 00225      1000 FORMAT(5X5(1H+),5X"FILE: ",3A2," ",15," ",15," NOT FOUND ")
679 00225 C
680 00225 C      CHECK FOR ANY OTHER ERROR CONDITION
681 00225 C
682 00225      20 IF (IER) 25,30
683 00230      25 WRITE(LIST,1010) IER,NAME,ISECU,ICR $ RETURN
684 00255      1010 FORMAT(5X5(1H+),5X"EMP ERROR: ",14," WHILE ATTEMPTING TO ",
685 00255      1      OPEN FILE: ",3A2," ",15," ",15)
686 00255 C
687 00255 C      CHECK THAT IT IS A RELOCATABLE FILE (TYPE 5).
688 00255 C
689 00255      30 IF (IDCB(3).NE.5) GO TO 40 $ IFLAG = 0 $ RETURN
690 00267      40 WRITE(LIST,1020) NAME,ISECU,ICR,IDCB(3)
691 00315      1020 FORMAT(5X5(1H+),5X"FILE: ",3A2," ",15," ",15,
692 00315      1      " IS A TYPE ",13," FILE")
693 00315      END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 216 COMMON: (NONE)

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION      |
|-------|---------|------------------|---------|---------------|
| a10   | 00202R  | STATEMENT NUMBER |         |               |
| a1000 | 00000R  | STATEMENT NUMBER |         |               |
| a1010 | 00032R  | STATEMENT NUMBER |         |               |
| a1020 | 00104R  | STATEMENT NUMBER |         |               |
| a20   | 00225R  | STATEMENT NUMBER |         |               |
| a25   | 00230R  | STATEMENT NUMBER |         |               |
| a30   | 00255R  | STATEMENT NUMBER |         |               |
| a40   | 00267R  | STATEMENT NUMBER |         |               |
| ICR   | 00147R  | VARIABLE         | INTEGER | DUMMY         |
| IDCB  | 00145R  | ARRAY, 1 DIMEN.  | INTEGER | DUMMY         |
| IER   | 00322R  | VARIABLE         | INTEGER | LOCAL         |
| IFLAG | 00150R  | VARIABLE         | INTEGER | DUMMY         |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS |
| ISECU | 00146R  | VARIABLE         | INTEGER | DUMMY         |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS |
| LUNS  | 00003X  | COMMON LABEL     |         | EXTERNAL      |
| NAME  | 00144R  | ARRAY, 1 DIMEN.  | INTEGER | DUMMY         |
| OPEN  | 00002X  | SUBROUTINE       |         | EXTERNAL      |

```

694 00000 SUBROUTINE SCANR(IFLAG), 09/04/81
695 00000 C
696 00000 C*** SCANR
697 00000 C**
698 00000 C** SCANR SCANS THE INPUT RELOCATABLE FILE AND ENTERS EACH PROGRAM
699 00000 C** NAME IN A TABLE IN COMMON.
700 00000 C**
701 00000 C* FORMAL PARAMETERS:
702 00000 C* IFLAG => ERROR FLAG
703 00000 C*
704 00000 COMMON IBUF(128)
705 00000 COMMON/FILES/ INAME(3), ITYPE, ISC, ICR, IDC(144),
706 00000 1 JNAME(3), JTYPE, JSC, JCR, JDC(144),
707 00000 1 KNAME(3), KTYPE, KSC, KCR, KDC(144), NAMES(3), KSIZE
708 00000 COMMON/TABLE/ N1, NMAX, NAME1(4,500)
709 00000 C
710 00000 C SEARCH FOR THE NEXT NAM RECORD AND INSERT THE NAME IN THE
711 00000 C TABLE. IF AN EOF OR ERROR IS FOUND, RETURN.
712 00000 C
713 00000 10 CALL FINDR(IDC, IFLAG, LEN) $ IF (IFLAG) 20, 15, 25
714 00016 15 CALL RWDF(IDC)
715 00021 20 RETURN
716 00023 25 N1 = N1 + 1 $ NAME1(1, N1) = 1
717 00033 CALL MEMOV(IBUF(4), NAME1(2, N1), 3) $ GO TO 10
718 00045 END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 41 COMMON: 128

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION       |
|-------|---------|------------------|---------|----------------|
| @10   | 00004R  | STATEMENT NUMBER |         |                |
| @15   | 00016R  | STATEMENT NUMBER |         |                |
| @20   | 00021R  | STATEMENT NUMBER |         |                |
| @25   | 00023R  | STATEMENT NUMBER |         |                |
| FILES | 00003X  | COMMON LABEL     |         | EXTERNAL       |
| FINDR | 00002X  | SUBROUTINE       |         | EXTERNAL       |
| IBUF  | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON         |
| ICR   | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDCB  | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IFL,3 | 00000R  | VARIABLE         | INTEGER | DUMMY          |
| INAME | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| ISC   | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ITYPE | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| JCR   | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB  | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JNAME | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JSC   | 00232+  | VARIABLE         | INTEGER | L COMMON FILES |
| JTYPE | 00231+  | VARIABLE         | INTEGER | L COMMON FILES |
| KCR   | 00461+  | VARIABLE         | INTEGER | L COMMON FILES |
| KDCB  | 00462+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KNAME | 00454+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| KSC   | 00460+  | VARIABLE         | INTEGER | L COMMON FILES |
| KSIZE | 00705+  | VARIABLE         | INTEGER | L COMMON FILES |
| KTYPE | 00457+  | VARIABLE         | INTEGER | L COMMON FILES |
| LEN   | 00045R  | VARIABLE         | INTEGER | LOCAL          |
| MEMOV | 00006X  | SUBROUTINE       |         | EXTERNAL       |
| N1    | 00000+  | VARIABLE         | INTEGER | L COMMON TABLE |
| NAME1 | 00002+  | ARRA, 2 DIMEN.   | INTEGER | L COMMON TABLE |
| NAMES | 00702+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| NMAX  | 00001+  | VARIABLE         | INTEGER | L COMMON TABLE |
| RWNDF | 00004X  | SUBROUTINE       |         | EXTERNAL       |
| TABLE | 00005X  | COMMON LABEL     |         | EXTERNAL       |

```

719 00000      SUBROUTINE SRCHN(NAM,N,IFLAG), 08/04/81
720 00000 C
721 00000 C***      SRCHN
722 00000 C**
723 00000 C**      SRCHN SCANS THE TABLE FOR A GIVEN PROGRAM NAME.
724 00000 C**
725 00000 C*      FORMAL PARAMETERS:
726 00000 C*          NAM    IS THE PROGRAM NAME
727 00000 C*          N      IS THE TABLE POINTER WHERE NAM IS FOUND
728 00000 C*          IFLAG IS THE STATUS FLAG OF THE PROGRAM IN
729 00000 C*          TABLE, OR ZERO OF NOT FOUND.
730 00000 C*
731 00000      DIMENSION NAM(3)
732 00000      COMMON/TABLE/ N1,NMAX,NAME(4,500)
733 00000 C
734 00000 C          PRESET IFLAG TO 1 AND SCAN THE LIST IN THREE WORD
735 00000 C          BLOCKS LOOKING FOR NAM.
736 00000 C
737 00000      DO 20 I=1,N1
738 00013          N = I $ IFLAG = NAME(1,I)
739 00022          IF (ICBYT(NAM,0,NAME(2,I),0,5).GE.0) RETURN
740 00041      20 CONTINUE
741 00050          IFLAG = 0
742 00052      END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 50 COMMON: (NONE)

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION |       |
|-------|---------|------------------|---------|----------|-------|
| @20   | 00041R  | STATEMENT NUMBER |         |          |       |
| :     | 00055R  | VARIABLE         | INTEGER | LOCAL    |       |
| ICRYT | 00002X  | FUNCTION         | INTEGER | EXTERNAL |       |
| IFLAG | 00002R  | VARIABLE         | INTEGER | DUMMY    |       |
| N     | 00001R  | VARIABLE         | INTEGER | DUMMY    |       |
| N1    | 00000+  | VARIABLE         | INTEGER | L COMMON | TABLE |
| NAM   | 00000R  | ARRAY, 1 DIMEN.  | INTEGER | DUMMY    |       |
| NAME  | 00002+  | ARRAY, 2 DIMEN.  | INTEGER | L COMMON | TABLE |
| NMAX  | 00001+  | VARIABLE         | INTEGER | L COMMON | TABLE |
| TABLE | 00003X  | COMMON LABEL     |         | EXTERNAL |       |

```

743 00000 SUBROUTINE SRCHR(IFLAG,NAME,IDCB,LEN), 08/04/81
744 00000 C
745 00000 C*** SRCHR
746 00000 C**
747 00000 C** SRCHR SEARCHES A RELOCATABLE FILE FOR A RELOCATABLE PROGRAM
748 00000 C**
749 00000 C* FORMAL PARAMETERS:
750 00000 C* IFLAG IS RETURNED AS: 1 IF THE PROGRAM WAS FOUND
751 00000 C* 0 IF THE PROGRAM WAS NOT FOUND
752 00000 C* -1 IF AN FMP ERROR OCCURED
753 00000 C* NAME IS THE PROGRAM NAME
754 00000 C* IDCB IS THE DCB OF THE RELOCATABLE FILE
755 00000 C* LEN IS THE LENGTH OF THE NAM RECORD, WHICH IS LEFT IN
756 00000 C* THE COMMON BUFFER.
757 00000 C*
758 00000 DIMENSION NAME(3),IDCB(144)
759 00000 COMMON IBUF(128)
760 00000 COMMON/LUNS / INPUT,LIST
761 00000 C
762 00000 C PRESET IFLAG TO ZERO AND LOOK FOR THE NEXT NAM RECORD
763 00000 C
764 00000 CALL RWNDF(IDCB) $ IFLAG = 0
765 00022 10 CALL FINDR(IDCB,IFLAG,LEN) $ IF (IFLAG) 25,25,15
766 00033 15 IF (ICBYT(IBUF,6,NAME,0,5).LT.0) GO TO '0' $ IFLAG = 1
767 00046 25 RETURN
768 00050 END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 45 COMMON: 128

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION      |
|-------|---------|------------------|---------|---------------|
| @10   | 00022R  | STATEMENT NUMBER |         |               |
| @15   | 00033R  | STATEMENT NUMBER |         |               |
| @25   | 00046R  | STATEMENT NUMBER |         |               |
| FINDR | 00003X  | SUBROUTINE       |         | EXTERNAL      |
| IBUF  | 00000C  | ARRAY, 1 DIMEN.  | INTEGER | COMMON        |
| ICBYT | 00004X  | FUNCTION         | INTEGER | EXTERNAL      |
| IDCB  | 00002R  | ARRAY, 1 DIMEN.  | INTEGER | DUMMY         |
| IFLAG | 00000R  | VARIABLE         | INTEGER | DUMMY         |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS |
| LEN   | 00003R  | VARIABLE         | INTEGER | DUMMY         |
| LIST  | 00001+  | VARIABLE         | INTEGER | L COMMON LUNS |
| LUNS  | 00000X  | COMMON LABEL     |         | EXTERNAL      |
| NAME  | 00001R  | ARRAY, 1 DIMEN.  | INTEGER | DUMMY         |
| RWPDF | 00002X  | SUBROUTINE       |         | EXTERNAL      |

```

769 00052 SUBROUTINE UDOIT, 08/04/81
770 00052 C
771 00052 C*** UDOIT
772 00052 C**
773 00052 C** UDOIT SUPERVISES EXECUTION OF LIBED IN THE COMMAND MODE.
774 00052 C**
775 00052 DIMENSION KWORD(3,9)
776 00052 COMMON/LUNS / INPUT,LIST
777 00052 COMMON/FILES/ INAME(3),ITYPE,ISC,ICR,IDCB(144),
778 00052 1 JNAME(3),JTYPE,JSC,JCR,JDCB(144),
779 00052 2 KNAME(3),KTYPE,KSC,KCR,KDCB(144)
780 00052 COMMON/WORKS/ KBUF(40),JBUF(10),LOG,ISTRC
781 00052 COMMON/TABLE/ N1,NMAX,NAME1(4,500)
782 00052 C
783 00052 C DEFINE THE KEYWORD TABLE
784 00052 C
785 00052 DATA KWORD / 'ABORT ' , 'ATTACH' , 'COPY ' , 'DELETE' ,
786 00052 1 'END ' , 'LIBRAR' , 'LIST ' , 'LL ' ,
787 00052 2 'REWIND' /
788 00052 DATA LIBOPN/-1/
789 00052 C
790 00052 C PRESET THE FLAG AND GET THE NEXT INSTRUCTION.
791 00052 C
792 00052 IFLAG = 1 $ IFANY = -1
793 00061 10 CALL GETIN(IFLAG)
794 00064 C
795 00064 C SCAN THE PARSED STRING FOR A KNOWN CODE AND TRANSFER
796 00064 C TO THE APPROPRIATE PORTION OF CODE. OTHERWISE ERROR
797 00064 C RETURN
798 00064 C
799 00064 DO 15 I=1,10
800 00066 IF (ICBYT(JBUF,0,KWORD(1,I),0,4).GE.0) GO TO (100,200,400,500,
801 00104 1 600,700,800,900,
802 00104 2 1000) I
803 00120 15 CONTINUE
804 00126 16 IFLAG = -1 $ GO TO 10
805 00130 C
806 00130 C ABORT! CLOSE FILES AND PRINT RUDE MESSAGE
807 00130 C
808 00131 100 CALL FCLOS(-1) $ WRITE(LIST,1100) $ CALL EXEC(6)
809 00151 1100 FORMAT(5X5(1H*),5X"LIBED ABORTED")
810 00151 C
811 00151 C ATTACH,XFILE:JSC:JCR
812 00151 C OPEN THE RELOCATABLE FILE TO JDCB
813 00151 C
814 00151 200 IF (NAMR(JBUF,KBUF,LOG,ISTRC)) 16,205
815 00161 205 CALL MEMOV(JBUF,JNAME,6)
816 00166 210 CALL CLOSE(JDCB) $ CALL ROPEN(JNAME,JDCB,JSC,JCR,IFLAG)
817 00200 N1 = 0 $ GO TO 10
818 00202 C
819 00202 C COPY,PGM1,PGM2
820 00202 C COPY FROM PGM1 TO PGM2 ON OLDLIB TO NEWLIB
821 00202 C
822 00203 400 CALL COPIN(IFLAG,IFANY) $ GO TO 10
823 00207 C

```

```

824 00207 C      DELETE,PGM1,PGM2,,,
825 00207 C      DELETE THE NAMED PROGRAMS
826 00207 C
827 00210 500 IF (NAMR(JBUF,KBUF,LOG,ISTR)) 10,505
828 00220 505 N1 = N1 + 1 $ IF (N1.GT.NMAX) GO TO 16
829 00233 NAME1(1,N1) = -1 $ CALL MEMOV(JBUF,NAME1(2,N1),3) $ GO TO 500
830 00251 C
831 00251 C      END PROCESSING.
832 00251 C
833 00252 600 CALL CLOSE(JDCB)
834 00255 CALL OPEN(JDCB,IER,KNAME) $ IF (IER) 610,605
835 00265 605 CALL MEMOV(KNAME,JNAME,3)
836 00272 610 CALL FCLOS(IFANY) $ CALL EXEC(6)
837 00300 C
838 00300 C      OPEN OUTPUT LIBRARY FILE
839 00300 C      IF THERE IS NO INPUT LIBRARY, OPEN AS INPUT ALSO.
840 00300 C      IF THE LIBRARY IS ALREADY OPEN, ERROR
841 00300 C
842 00300 700 IF (LIBOPN.GE.0) GO TO 16 $ LIBOPN = 1
843 00305 IF (NAMR(KNAME,KBUF,LOG,ISTR)) 16,705
844 00315 705 CALL CPRLF(IFLAG) $ IF(JDCB(10).NE.0) GO TO 10
845 00323 CALL MEMOV(KNAME,JNAME,6)
846 00330 CALL OPEN(JDCB,IER,JNAME,0,JSECU,JCR) $ IF (IER.GE.0) 210,10
847 00342 C
848 00342 C      LIST THE REQUESTED FILE, OR THE OUTPUT FILE
849 00342 C
850 00345 800 IF (NAMR(JBUF,KBUF,LOG,ISTR)) 820,810
851 00355 810 CALL CLOSE(KDCB) $ CALL MEMOV(JBUF,KNAME,6)
852 00365 CALL ROPEN(KNAME,KDCB,KSC,KCR,IFLAG) $ IF (IFLAG.LT.0) GO TO 16
853 00377 820 CALL RWNDF(KDCB) $ CALL LIBPR $ GO TO 10
854 00404 C
855 00404 C      LL,N
856 00404 C      REASSIGN LIST LOGICAL UNIT
857 00404 C
858 00405 900 IF (NAMR(JBUF,KBUF,LOG,ISTR)) 16,901
859 00415 901 IF (IAND(JBUF(4),3B).NE.1) GO TO 16
860 00422 LIST = JBUF(1) $ GO TO 10
861 00424 C
862 00424 C      REWIND THE INPUT LIBRARY
863 00424 C
864 00425 1000 CALL RWNDF(JDCB) $ GO TO 10
865 00431 END

```

FTN4X COMPILER: HP92834 REV.2030 (800821)

\*\* NO WARNINGS \*\* NO ERRORS \*\* PROGRAM: 298 COMMON: (NONE)

## SYMBOL TABLE

| NAME  | ADDRESS | USAGE            | TYPE    | LOCATION       |
|-------|---------|------------------|---------|----------------|
| a10   | 00061R  | STATEMENT NUMBER |         |                |
| a100  | 00131R  | STATEMENT NUMBER |         |                |
| a1000 | 00425R  | STATEMENT NUMBER |         |                |
| a1100 | 00034R  | STATEMENT NUMBER |         |                |
| a15   | 00120R  | STATEMENT NUMBER |         |                |
| a16   | 00126R  | STATEMENT NUMBER |         |                |
| a200  | 00151R  | STATEMENT NUMBER |         |                |
| a205  | 00161R  | STATEMENT NUMBER |         |                |
| a210  | 00166R  | STATEMENT NUMBER |         |                |
| a400  | 00203R  | STATEMENT NUMBER |         |                |
| a500  | 00210R  | STATEMENT NUMBER |         |                |
| a505  | 00220R  | STATEMENT NUMBER |         |                |
| a600  | 00252R  | STATEMENT NUMBER |         |                |
| a605  | 00265R  | STATEMENT NUMBER |         |                |
| a610  | 00272R  | STATEMENT NUMBER |         |                |
| a700  | 00300R  | STATEMENT NUMBER |         |                |
| a705  | 00315R  | STATEMENT NUMBER |         |                |
| a800  | 00345R  | STATEMENT NUMBER |         |                |
| a810  | 00355R  | STATEMENT NUMBER |         |                |
| a820  | 00377R  | STATEMENT NUMBER |         |                |
| a900  | 00405R  | STATEMENT NUMBER |         |                |
| a901  | 00415R  | STATEMENT NUMBER |         |                |
| CLOSE | 00016X  | SUBROUTINE       |         | EXTERNAL       |
| COPIN | 00021X  | SUBROUTINE       |         | EXTERNAL       |
| CREF  | 00023X  | SUBROUTINE       |         | EXTERNAL       |
| EXEC  |         | INTRINSIC        |         |                |
| FCLOS | 00005X  | SUBROUTINE       |         | EXTERNAL       |
| FILES | 00015X  | COMMON LABEL     |         | EXTERNAL       |
| GETIN | 00001X  | SUBROUTINE       |         | EXTERNAL       |
| I     | 00436R  | VARIABLE         | INTEGER | LOCAL          |
| IAND  |         | INTRINSIC        | GENERIC |                |
| ICBYT | 00002X  | FUNCTION         | INTEGER | EXTERNAL       |
| ICR   | 00005+  | VARIABLE         | INTEGER | L COMMON FILES |
| IDCB  | 00006+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| IER   | 00433R  | VARIABLE         | INTEGER | LOCAL          |
| IFANY | 00435R  | VARIABLE         | INTEGER | LOCAL          |
| IFLAG | 00432R  | VARIABLE         | INTEGER | LOCAL          |
| INAME | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| INPUT | 00000+  | VARIABLE         | INTEGER | L COMMON LUNS  |
| ISC   | 00004+  | VARIABLE         | INTEGER | L COMMON FILES |
| ISTR  | 00063+  | VARIABLE         | INTEGER | L COMMON WORKS |
| ITYPE | 00003+  | VARIABLE         | INTEGER | L COMMON FILES |
| JBUF  | 00050+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON WORKS |
| JCR   | 00233+  | VARIABLE         | INTEGER | L COMMON FILES |
| JDCB  | 00234+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JNAME | 00226+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON FILES |
| JSC   | 00232+  | VARIABLE         | INTEGER | L COMMON FILES |
| JSECU | 00434R  | VARIABLE         | INTEGER | LOCAL          |
| JTYPE | 00231+  | VARIABLE         | INTEGER | L COMMON FILES |
| KBUF  | 00000+  | ARRAY, 1 DIMEN.  | INTEGER | L COMMON WORKS |

## SYMBOL TABLE

| NAME   | ADDRESS | USAGE           | TYPE    | LOCATION |       |
|--------|---------|-----------------|---------|----------|-------|
| KCR    | 00461+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KDCB   | 00462+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| KNAME  | 00454+  | ARRAY, 1 DIMEN. | INTEGER | L COMMON | FILES |
| KSC    | 00460+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KTYPE  | 00457+  | VARIABLE        | INTEGER | L COMMON | FILES |
| KWORD  | 00000R  | ARRAY, 2 DIMEN. | INTEGER | LOCAL    |       |
| LIROPN | 00033R  | VARIABLE        | INTEGER | LOCAL    |       |
| LIBPR  | 00025X  | SUBROUTINE      |         | EXTERNAL |       |
| LIST   | 00001+  | VARIABLE        | INTEGER | L COMMON | LUNS  |
| LOG    | 00062+  | VARIABLE        | INTEGER | L COMMON | WORKS |
| LUNS   | 00006X  | COMMON LABEL    |         | EXTERNAL |       |
| MEMOV  | 00014X  | SUBROUTINE      |         | EXTERNAL |       |
| N1     | 00000+  | VARIABLE        | INTEGER | L COMMON | TABLE |
| NAME1  | 00002+  | ARRAY, 2 DIMEN. | INTEGER | L COMMON | TABLE |
| NAMR   | 00013X  | FUNCTION        | INTEGER | EXTERNAL |       |
| NMAX   | 00001+  | VARIABLE        | INTEGER | L COMMON | TABLE |
| OPEN   | 00022X  | SUBROUTINE      |         | EXTERNAL |       |
| ROPEN  | 00017X  | SUBROUTINE      |         | EXTERNAL |       |
| RWPDF  | 00024X  | SUBROUTINE      |         | EXTERNAL |       |
| TABLE  | 00020X  | COMMON LABEL    |         | EXTERNAL |       |
| WORKS  | 00003X  | COMMON LABEL    |         | EXTERNAL |       |

```

0001          ASMB,L
0002 00000    NAM CHARM,7 20/06/80
0003*
0004****    CHARM
0005****
0006****    CHARM CONTAINS FORTRAN CALLABLE ASSEMBLER ROUTINES FOR
0007****    CHARACTER OR BYTE MANIPULATION. ENTRY POINTS ARE:
0008****    CMOVE:  MOVE BYTE STRING
0009****    ICSCN:  SCAN FOR BYTE
0010****    ICBYT:  COMPARE BYTES
0011****
0012          EXT .ENTR
0013          ENT CMOVE,ICSCN,ICBYT
0014**
0015****    CMOVE: CALLING SEQUENCE IS:
0016****    CALL CMOVE(IFROM,IFLRF,ITO,ITLRF,NCHAR)
0017****    WHERE:
0018****    IFROM IS STARTING WORD OF SOURCE STRING
0019****    IFLRF IS THE CHARACTER OFFSET OF THE SOURCE STRING
0020****    ITO  IS STARTING WORD OF DESTINATION STRING
0021****    ITLRF IS CHARACTER OFFSET OF DESTINATION
0022****    NCHAR IS NUMBER OF CHARACTERS TO SHIFT
0023**
0024****    SOURCE FILE = &CHARM          LIBRARY = $$UTIL
0025**
0026 00000 000000 FROM BSS 1
0027 00001 000000 FLRF BSS 1
0028 00002 000000 TO  BSS 1
0029 00003 000000 TLRf BSS 1
0030 00004 000000 NCHAR BSS 1
0031 00005 000000 CMOVE NOP
0032 00006 016001X JSB .ENTR      RESOLVE PARAMETER ADDRESSES
0033 00007 000000R DEF FROM
0034 00010 062000R LDA FROM      GET START ADDRESS OF SOURCE ARRAY
0035 00011 000066  CLE,ELA      FORM CHARACTER ADDRESS BY DOUBLING
0036 00012 142001R ADA FLRF,I    ADD THE CHARACTER OFFSET TO
0037*      THE CHARACTER ADDRESS
0038 00013 066002R LDB TO        GET THE DESTINATION ARRAY ADDRESS
0039 00014 004066  CLE,ELB      CONVERT TO CHARACTER ADDRESS
0040 00015 146003R :DB TLRf,I    ADD THE CHARACTER OFFSET TO FORM
0041*      THE FINAL DESTINATION CHARACTER ADDRESS
0042 00016 072104R STA ACHAR      SAVE THE SOURCE ADDRESS
0043 00017 003004  CMA,INA      NEGATE IT
0044 00020 040001  ADA B        ADD THE DESTINATION ADDRESS
0045 00021 002003  SZA,RSS      EQUAL?
0046 00022 126005R JMP CMOVE,I    YES - RETURN
0047 00023 002020  SSA          NEGATIVE?
0048 00024 026033R JMP QUICK     YES - USE QUICK MOVE
0049 00025 062104R LDA ACHAR      GET SOURCE ADDRESS AGAIN
0050 00026 142004R ADA NCHAR,I    AND ADD THE NUMBER OF CHARACTERS
0051 00027 003004  CMA,INA      FORM THE NEGATIVE
0052 00030 040001  ADA B        IS THIS GREATER THAN THE DESTINATION ADD
0053 00031 002020  SSA
0054 00032 026040R JMP SLOW      NO - MUST USE SLOW MOVE

```

3:43 PM FRI., 10 APR., 1981

```

0055 00033 062104R QUICK LDA ACHAR RESTORE SOURCE ADDRESS
0056 00034 105765 MBT NCHAR,I MOVE BYTES
      00035 100004R
      00036 000000
0057 00037 126005R JMP CMOVE,I RETURN
0058 00040 062104R SLOW LDA ACHAR GET SOURCE ADDRESS
0059 00041 002004 INA AND INCREMENT IT (READ MBT DESCRIPTION)
0060 00042 006004 INB DESTINATION ADDRESS ALSO (SAME REASON)
0061 00043 142004R ADA NCHAR,I ADD NO OF CHARACTERS
0062 00044 146004R ADB NCHAR,I SAME TO B
0063 00045 036104R ISZ ACHAR BUMP ACHAR (USED AS LOOP TEST)
0064 00046 042057R NEXT ADA DM2 COMPUTE NEXT BYTE ADDRESS
0065 00047 046057R ADB DM2 LIKEWISE
0066 00050 105765 MBT D1 MOVE ONE BYTE
      00051 000056R
      00052 000000
0067 00053 052104R CPA ACHAR FINISHED?
0068 00054 126005R JMP CMOVE,I YES - RETURN
0069 00055 026046R JMP NEXT OTHERWISE LOOP
0070 00056 000001 D1 DEC 1
0071 00057 177776 DM2 DEC -2
0072*
0073**
0074** ICSCN: CALLING SEQUENCE:
0075** I = ICSCN(IBUF,IOFST,ICHAR)
0076** WHERE:
0077** IBUF IS THE BUFFER TO BE SCANNED
0078** IOFST IS THE CHARACTER OFFSET TO BEGIN THE SCAN
0079** ICHAR CONTAINS THE CHARACTER TO BE SCANNED FOR IN THE LOW
0080** BYTE AND THE TERMINATION CHARACTER IN THE HIGH BYTE
0081** I CONTAINS THE OFFSET OF THE CHARACTER FROM THE START OF
0082** THE BUFFER, OR -1 IF THE TERMINATION BYTE WAS ENCOUNTERED
0083**
0084**
0085 00060 000000 IBUF BSS 1
0086 00061 000000 IOFST BSS 1
0087 00062 000000 ICHAR BSS 1
0088 00063 000000 ICSCN NOP ENTRY POINT
0089 00064 016001X JSB .ENTR RESOLVE PARAMETER ADDRESSES
0090 00065 000060R DEF IBUF
0091 00066 066060R LDB IBUF GET INPUT BUFFER STARTING ADDRESS
0092 00067 004066 CLE,ELB MAKE INTO A CHARACTER ADDRESS
0093 00070 146061R ADB IOFST,I ADD THE CHARACTER OFFSET
0094 00071 076104R STB ACHAR AND SAVE IT
0095 00072 162062R LDA ICHAR,I GET THE CHARACTER
0096 00073 105767 SFB SCAN FOR THE BYTE
0097 00074 026100R JMP OK NORMAL RETURN - CONTINUE
0098 00075 062105R LDA DM1 BYTE NOT FOUND
0099 00076 126063R JMP ICSCN,I SO SET A REG TO -1 AND RETURN
0100 00077 126063R JMP ICSCN,I AND RETURN
0101 00100 062104R OK LDA ACHAR
0102 00101 003004 CMA,INA NEGATE IT
0103 00102 040001 ADA B AND ADD THE FINAL CHARACTER ADDRESS
0104 00103 126063R JMP ICSCN,I AND RETURN

```

3:43 PM FRI., 10 APR., 1981

```

0055 00033 062104R QUICK LDA ACHAR RESTORE SOURCE ADDRESS
0056 00034 105765 MBT NCHAR,I MOVE BYTES
      00035 100004R
      00036 000000
0057 00037 126005R JMP CMOVE,I RETURN
0058 00040 062104R SLOW LDA ACHAR GET SOURCE ADDRESS
0059 00041 002004 INA AND INCREMENT IT (READ MBT DESCRIPTION)
0060 00042 006004 INB DESTINATION ADDRESS ALSO (SAME REASON)
0061 00043 142004R ADA NCHAR,I ADD NO OF CHARACTERS
0062 00044 146004R ADB NCHAR,I SAME TO B
0063 00045 036104R ISZ ACHAR BUMP ACHAR (USED AS LOOP TEST)
0064 00046 042057R NEXT ADA DM2 COMPUTE NEXT BYTE ADDRESS
0065 00047 046057R ADB DM2 LIKEWISE
0066 00050 105765 MBT D1 MOVE ONE BYTE
      00051 000056R
      00052 000000
0067 00053 052104R CPA ACHAR FINISHED?
0068 00054 126005R JMP CMOVE,I YES - RETURN
0069 00055 026046R JMP NEXT OTHERWISE LOOP
0070 00056 000001 D1 DEC 1
0071 00057 177776 DM2 DEC -2
0072*
0073**
0074** ICSCN: CALLING SEQUENCE:
0075** I = ICSCN(IBUF,IOFST,ICHAR)
0076** WHERE:
0077** IBUF IS THE BUFFER TO BE SCANNED
0078** IOFST IS THE CHARACTER OFFSET TO BEGIN THE SCAN
0079** ICHAR CONTAINS THE CHARACTER TO BE SCANNED FOR IN THE LOW
0080** BYTE AND THE TERMINATION CHARACTER IN THE HIGH BYTE
0081** I CONTAINS THE OFFSET OF THE CHARACTER FROM THE START OF
0082** THE BUFFER, OR -1 IF THE TERMINATION BYTE WAS ENCOUNTERED
0083**
0084**
0085 00060 000000 IBUF BSS 1
0086 00061 000000 IOFST BSS 1
0087 00062 000000 ICHAR BSS 1
0088 00063 000000 ICSCN NOP ENTRY POINT
0089 00064 016001X JSB .ENTR RESOLVE PARAMETER ADDRESSES
0090 00065 000060R DEF IBUF
0091 00066 066060R LDB IBUF GET INPUT BUFFER STARTING ADDRESS
0092 00067 004066 CLE,ELB MAKE INTO A CHARACTER ADDRESS
0093 00070 146061R ADB IOFST,I ADD THE CHARACTER OFFSET
0094 00071 076104R STB ACHAR AND SAVE IT
0095 00072 162062R LDA ICHAR,I GET THE CHARACTER
0096 00073 105767 SFB SCAN FOR THE BYTE
0097 00074 026100R JMP OK NORMAL RETURN - CONTINUE
0098 00075 062105R LDA DM1 BYTE NOT FOUND
0099 00076 126063R JMP ICSCN,I SO SET A REG TO -1 AND RETURN
0100 00077 126063R JMP ICSCN,I AND RETURN
0101 00100 062104R OK LDA ACHAR
0102 00101 003004 CMA,INA NEGATE IT
0103 00102 040001 ADA B AND ADD THE FINAL CHARACTER ADDRESS
0104 00103 126063R JMP ICSCN,I AND RETURN

```

3:43 PM FRI., 10 APR., 1981

```

0055 00033 062104R QUICK LDA ACHAR RESTORE SOURCE ADDRESS
0056 00034 105765 MBT NCHAR,I MOVE BYTES
      00035 100004R
      00036 000000
0057 00037 126005R JMP CMOVE,I RETURN
0058 00040 062104R SLOW LDA ACHAR GET SOURCE ADDRESS
0059 00041 002004 INA AND INCREMENT IT (READ MBT DESCRIPTION)
0060 00042 006004 INB DESTINATION ADDRESS ALSO (SAME REASON)
0061 00043 142004R ADA NCHAR,I ADD NO OF CHARACTERS
0062 00044 146004R ADB NCHAR,I SAME TO B
0063 00045 036104R ISZ ACHAR BUMP ACHAR (USED AS LOOP TEST)
0064 00046 042057R NEXT ADA DM2 COMPUTE NEXT BYTE ADDRESS
0065 00047 046057R ADB DM2 LIKEWISE
0066 00050 105765 MBT D1 MOVE ONE BYTE
      00051 000056R
      00052 000000
0067 00053 052104R CPA ACHAR FINISHED?
0068 00054 126005R JMP CMOVE,I YES - RETURN
0069 00055 026046R JMP NEXT OTHERWISE LOOP
0070 00056 000001 D1 DEC 1
0071 00057 177776 DM2 DEC -2
0072*
0073**
0074** ICSCN: CALLING SEQUENCE:
0075** I = ICSCN(IBUF,IOFST,ICHAR)
0076** WHERE:
0077** IBUF IS THE BUFFER TO BE SCANNED
0078** IOFST IS THE CHARACTER OFFSET TO BEGIN THE SCAN
0079** ICHAR CONTAINS THE CHARACTER TO BE SCANNED FOR IN THE LOW
0080** BYTE AND THE TERMINATION CHARACTER IN THE HIGH BYTE
0081** I CONTAINS THE OFFSET OF THE CHARACTER FROM THE START OF
0082** THE BUFFER, OR -1 IF THE TERMINATION BYTE WAS ENCOUNTERED
0083**
0084**
0085 00060 000000 IBUF BSS 1
0086 00061 000000 IOFST BSS 1
0087 00062 000000 ICHAR BSS 1
0088 00063 000000 ICSCN NOP ENTRY POINT
0089 00064 016001X JSB .ENTR RESOLVE PARAMETER ADDRESSES
0090 00065 000060R DEF IBUF
0091 00066 066060R LDB IBUF GET INPUT BUFFER STARTING ADDRESS
0092 00067 004066 CLE,ELB MAKE INTO A CHARACTER ADDRESS
0093 00070 146061R ADB IOFST,I ADD THE CHARACTER OFFSET
0094 00071 076104R STB ACHAR AND SAVE IT
0095 00072 162062R LDA ICHAR,I GET THE CHARACTER
0096 00073 105767 SFB SCAN FOR THE BYTE
0097 00074 026100R JMP OK NORMAL RETURN - CONTINUE
0098 00075 062105R LDA DM1 BYTE NOT FOUND
0099 00076 126063R JMP ICSCN,I SO SET A REG TO -1 AND RETURN
0100 00077 126063R JMP ICSCN,I AND RETURN
0101 00100 062104R OK LDA ACHAR
0102 00101 003004 CMA,INA NEGATE IT
0103 00102 040001 ADA B AND ADD THE FINAL CHARACTER ADDRESS
0104 00103 126063R JMP ICSCN,I AND RETURN

```

3:43 PM FRI., 10 APR., 1981

```

0055 00033 062104R QUICK LDA ACHAR RESTORE SOURCE ADDRESS
0056 00034 105765 MBT NCHAR,I MOVE BYTES
      00035 100004R
      00036 000000
0057 00037 126005R JMP CMOVE,I RETURN
0058 00040 062104R SLOW LDA ACHAR GET SOURCE ADDRESS
0059 00041 002004 INA AND INCREMENT IT (READ MBT DESCRIPTION)
0060 00042 006004 INB DESTINATION ADDRESS ALSO (SAME REASON)
0061 00043 142004R ADA NCHAR,I ADD NO OF CHARACTERS
0062 00044 146004R ADB NCHAR,I SAME TO B
0063 00045 036104R ISZ ACHAR BUMP ACHAR (USED AS LOOP TEST)
0064 00046 042057R NEXT ADA DM2 COMPUTE NEXT BYTE ADDRESS
0065 00047 046057R ADB DM2 LIKEWISE
0066 00050 105765 MBT D1 MOVE ONE BYTE
      00051 000056R
      00052 000000
0067 00053 052104R CPA ACHAR FINISHED?
0068 00054 126005R JMP CMOVE,I YES - RETURN
0069 00055 026046R JMP NEXT OTHERWISE LOOP
0070 00056 000001 D1 DEC 1
0071 00057 177776 DM2 DEC -2
0072*
0073**
0074** ICSCN: CALLING SEQUENCE:
0075** I = ICSCN(IBUF,IOFST,ICHAR)
0076** WHERE:
0077** IBUF IS THE BUFFER TO BE SCANNED
0078** IOFST IS THE CHARACTER OFFSET TO BEGIN THE SCAN
0079** ICHAR CONTAINS THE CHARACTER TO BE SCANNED FOR IN THE LOW
0080** BYTE AND THE TERMINATION CHARACTER IN THE HIGH BYTE
0081** I CONTAINS THE OFFSET OF THE CHARACTER FROM THE START OF
0082** THE BUFFER, OR -1 IF THE TERMINATION BYTE WAS ENCOUNTERED
0083**
0084**
0085 00060 000000 IBUF BSS 1
0086 00061 000000 IOFST BSS 1
0087 00062 000000 ICHAR BSS 1
0088 00063 000000 ICSCN NOP ENTRY POINT
0089 00064 016001X JSB .ENTR RESOLVE PARAMETER ADDRESSES
0090 00065 000060R DEF IBUF
0091 00066 066060R LDB IBUF GET INPUT BUFFER STARTING ADDRESS
0092 00067 004066 CLE,ELB MAKE INTO A CHARACTER ADDRESS
0093 00070 146061R ADB IOFST,I ADD THE CHARACTER OFFSET
0094 00071 076104R STB ACHAR AND SAVE IT
0095 00072 162062R LDA ICHAR,I GET THE CHARACTER
0096 00073 105767 SFB SCAN FOR THE BYTE
0097 00074 026100R JMP OK NORMAL RETURN - CONTINUE
0098 00075 062105R LDA DM1 BYTE NOT FOUND
0099 00076 126063R JMP ICSCN,I SO SET A REG TO -1 AND RETURN
0100 00077 126063R JMP ICSCN,I AND RETURN
0101 00100 062104R OK LDA ACHAR
0102 00101 003004 CMA,INA NEGATE IT
0103 00102 040001 ADA B AND ADD THE FINAL CHARACTER ADDRESS
0104 00103 126063R JMP ICSCN,I AND RETURN

```

3:43 PM FRI., 10 APR., 1981

```

0055 00033 062104R QUICK LDA ACHAR RESTORE SOURCE ADDRESS
0056 00034 105765 MBT NCHAR,I MOVE BYTES
      00035 100004R
      00036 000000
0057 00037 126005R JMP CMOVE,I RETURN
0058 00040 062104R SLOW LDA ACHAR GET SOURCE ADDRESS
0059 00041 002004 INA AND INCREMENT IT (READ MBT DESCRIPTION)
0060 00042 006004 INB DESTINATION ADDRESS ALSO (SAME REASON)
0061 00043 142004R ADA NCHAR,I ADD NO OF CHARACTERS
0062 00044 146004R ADB NCHAR,I SAME TO B
0063 00045 036104R ISZ ACHAR BUMP ACHAR (USED AS LOOP TEST)
0064 00046 042057R NEXT ADA DM2 COMPUTE NEXT BYTE ADDRESS
0065 00047 046057R ADB DM2 LIKEWISE
0066 00050 105765 MBT D1 MOVE ONE BYTE
      00051 000056R
      00052 000000
0067 00053 052104R CPA ACHAR FINISHED?
0068 00054 126005R JMP CMOVE,I YES - RETURN
0069 00055 026046R JMP NEXT OTHERWISE LOOP
0070 00056 000001 D1 DEC 1
0071 00057 177776 DM2 DEC -2
0072*
0073**
0074** ICSCN: CALLING SEQUENCE:
0075** I = ICSCN(IBUF,IOFST,ICHAR)
0076** WHERE:
0077** IBUF IS THE BUFFER TO BE SCANNED
0078** IOFST IS THE CHARACTER OFFSET TO BEGIN THE SCAN
0079** ICHAR CONTAINS THE CHARACTER TO BE SCANNED FOR IN THE LOW
0080** BYTE AND THE TERMINATION CHARACTER IN THE HIGH BYTE
0081** I CONTAINS THE OFFSET OF THE CHARACTER FROM THE START OF
0082** THE BUFFER, OR -1 IF THE TERMINATION BYTE WAS ENCOUNTERED
0083**
0084**
0085 00060 000000 IBUF BSS 1
0086 00061 000000 IOFST BSS 1
0087 00062 000000 ICHAR BSS 1
0088 00063 000000 ICSCN NOP ENTRY POINT
0089 00064 016001X JSB .ENTR RESOLVE PARAMETER ADDRESSES
0090 00065 000060R DEF IBUF
0091 00066 066060R LDB IBUF GET INPUT BUFFER STARTING ADDRESS
0092 00067 004066 CLE,ELB MAKE INTO A CHARACTER ADDRESS
0093 00070 146061R ADB IOFST,I ADD THE CHARACTER OFFSET
0094 00071 076104R STB ACHAR AND SAVE IT
0095 00072 162062R LDA ICHAR,I GET THE CHARACTER
0096 00073 105767 SFB SCAN FOR THE BYTE
0097 00074 026100R JMP OK NORMAL RETURN - CONTINUE
0098 00075 062105R LDA DM1 BYTE NOT FOUND
0099 00076 126063R JMP ICSCN,I SO SET A REG TO -1 AND RETURN
0100 00077 126063R JMP ICSCN,I AND RETURN
0101 00100 062104R OK LDA ACHAR
0102 00101 003004 CMA,INA NEGATE IT
0103 00102 040001 ADA B AND ADD THE FINAL CHARACTER ADDRESS
0104 00103 126063R JMP ICSCN,I AND RETURN

```

3:43 PM FRI., 10 APR., 1981

```

0055 00033 062104R QUICK LDA ACHAR RESTORE SOURCE ADDRESS
0056 00034 105765 MBT NCHAR,I MOVE BYTES
      00035 100004R
      00036 000000
0057 00037 126005R JMP CMOVE,I RETURN
0058 00040 062104R SLOW LDA ACHAR GET SOURCE ADDRESS
0059 00041 002004 INA AND INCREMENT IT (READ MBT DESCRIPTION)
0060 00042 006004 INB DESTINATION ADDRESS ALSO (SAME REASON)
0061 00043 142004R ADA NCHAR,I ADD NO OF CHARACTERS
0062 00044 146004R ADB NCHAR,I SAME TO B
0063 00045 036104R ISZ ACHAR BUMP ACHAR (USED AS LOOP TEST)
0064 00046 042057R NEXT ADA DM2 COMPUTE NEXT BYTE ADDRESS
0065 00047 046057R ADB DM2 LIKEWISE
0066 00050 105765 MBT D1 MOVE ONE BYTE
      00051 000056R
      00052 000000
0067 00053 052104R CPA ACHAR FINISHED?
0068 00054 126005R JMP CMOVE,I YES - RETURN
0069 00055 026046R JMP NEXT OTHERWISE LOOP
0070 00056 000001 D1 DEC 1
0071 00057 177776 DM2 DEC -2
0072*
0073**
0074** ICSCN: CALLING SEQUENCE:
0075** I = ICSCN(IBUF,IOFST,ICHAR)
0076** WHERE:
0077** IBUF IS THE BUFFER TO BE SCANNED
0078** IOFST IS THE CHARACTER OFFSET TO BEGIN THE SCAN
0079** ICHAR CONTAINS THE CHARACTER TO BE SCANNED FOR IN THE LOW
0080** BYTE AND THE TERMINATION CHARACTER IN THE HIGH BYTE
0081** I CONTAINS THE OFFSET OF THE CHARACTER FROM THE START OF
0082** THE BUFFER, OR -1 IF THE TERMINATION BYTE WAS ENCOUNTERED
0083**
0084**
0085 00060 000000 IBUF BSS 1
0086 00061 000000 IOFST BSS 1
0087 00062 000000 ICHAR BSS 1
0088 00063 000000 ICSCN NOP ENTRY POINT
0089 00064 016001X JSB .ENTR RESOLVE PARAMETER ADDRESSES
0090 00065 000060R DEF IBUF
0091 00066 066060R LDB IBUF GET INPUT BUFFER STARTING ADDRESS
0092 00067 004066 CLE,ELB MAKE INTO A CHARACTER ADDRESS
0093 00070 146061R ADB IOFST,I ADD THE CHARACTER OFFSET
0094 00071 076104R STB ACHAR AND SAVE IT
0095 00072 162062R LDA ICHAR,I GET THE CHARACTER
0096 00073 105767 SFB SCAN FOR THE BYTE
0097 00074 026100R JMP OK NORMAL RETURN - CONTINUE
0098 00075 062105R LDA DM1 BYTE NOT FOUND
0099 00076 126063R JMP ICSCN,I SO SET A REG TO -1 AND RETURN
0100 00077 126063R JMP ICSCN,I AND RETURN
0101 00100 062104R OK LDA ACHAR
0102 00101 003004 CMA,INA NEGATE IT
0103 00102 040001 ADA B AND ADD THE FINAL CHARACTER ADDRESS
0104 00103 126063R JMP ICSCN,I AND RETURN

```

3:43 PM FRI., 10 APR., 1981

```

0055 00033 062104R QUICK LDA ACHAR RESTORE SOURCE ADDRESS
0056 00034 105765 MBT NCHAR,I MOVE BYTES
      00035 100004R
      00036 000000
0057 00037 126005R JMP CMOVE,I RETURN
0058 00040 062104R SLOW LDA ACHAR GET SOURCE ADDRESS
0059 00041 002004 INA AND INCREMENT IT (READ MBT DESCRIPTION)
0060 00042 006004 INB DESTINATION ADDRESS ALSO (SAME REASON)
0061 00043 142004R ADA NCHAR,I ADD NO OF CHARACTERS
0062 00044 146004R ADB NCHAR,I SAME TO B
0063 00045 036104R ISZ ACHAR BUMP ACHAR (USED AS LOOP TEST)
0064 00046 042057R NEXT ADA DM2 COMPUTE NEXT BYTE ADDRESS
0065 00047 046057R ADB DM2 LIKEWISE
0066 00050 105765 MBT D1 MOVE ONE BYTE
      00051 000056R
      00052 000000
0067 00053 052104R CPA ACHAR FINISHED?
0068 00054 126005R JMP CMOVE,I YES - RETURN
0069 00055 026046R JMP NEXT OTHERWISE LOOP
0070 00056 000001 D1 DEC 1
0071 00057 177776 DM2 DEC -2
0072*
0073**
0074** ICSCN: CALLING SEQUENCE:
0075** I = ICSCN(IBUF,IOFST,ICHAR)
0076** WHERE:
0077** IBUF IS THE BUFFER TO BE SCANNED
0078** IOFST IS THE CHARACTER OFFSET TO BEGIN THE SCAN
0079** ICHAR CONTAINS THE CHARACTER TO BE SCANNED FOR IN THE LOW
0080** BYTE AND THE TERMINATION CHARACTER IN THE HIGH BYTE
0081** I CONTAINS THE OFFSET OF THE CHARACTER FROM THE START OF
0082** THE BUFFER, OR -1 IF THE TERMINATION BYTE WAS ENCOUNTERED
0083**
0084**
0085 00060 000000 IBUF BSS 1
0086 00061 000000 IOFST BSS 1
0087 00062 000000 ICHAR BSS 1
0088 00063 000000 ICSCN NOP ENTRY POINT
0089 00064 016001X JSB .ENTR RESOLVE PARAMETER ADDRESSES
0090 00065 000060R DEF IBUF
0091 00066 066060R LDB IBUF GET INPUT BUFFER STARTING ADDRESS
0092 00067 004066 CLE,ELB MAKE INTO A CHARACTER ADDRESS
0093 00070 146061R ADB IOFST,I ADD THE CHARACTER OFFSET
0094 00071 076104R STB ACHAR AND SAVE IT
0095 00072 162062R LDA ICHAR,I GET THE CHARACTER
0096 00073 105767 SFB SCAN FOR THE BYTE
0097 00074 026100R JMP OK NORMAL RETURN - CONTINUE
0098 00075 062105R LDA DM1 BYTE NOT FOUND
0099 00076 126063R JMP ICSCN,I SO SET A REG TO -1 AND RETURN
0100 00077 126063R JMP ICSCN,I AND RETURN
0101 00100 062104R OK LDA ACHAR
0102 00101 003004 CMA,INA NEGATE IT
0103 00102 040001 ADA B AND ADD THE FINAL CHARACTER ADDRESS
0104 00103 126063R JMP ICSCN,I AND RETURN

```

| INDEX |         |     |         |     |
|-------|---------|-----|---------|-----|
| LOC   | CONTENT | LOC | CONTENT | LOC |
| C01   |         | D04 |         |     |
| D01   |         | P04 |         |     |
| E01   |         | C05 |         |     |
| F01   |         | D05 |         |     |
| G01   |         | E05 |         |     |
| H01   |         | F05 |         |     |
| I01   |         | G05 |         |     |
| J01   |         | H05 |         |     |
| K01   |         | I05 |         |     |
| L01   |         | J05 |         |     |
| M01   |         | K05 |         |     |
| N01   |         | L05 |         |     |
| O01   |         | M05 |         |     |
| P01   |         |     |         |     |
| C02   |         |     |         |     |
| D02   |         |     |         |     |
| E02   |         |     |         |     |
| F02   |         |     |         |     |
| G02   |         |     |         |     |
| H02   |         |     |         |     |
| I02   |         |     |         |     |
| J02   |         |     |         |     |
| K02   |         |     |         |     |
| L02   |         |     |         |     |
| M02   |         |     |         |     |
| N02   |         |     |         |     |
| O02   |         |     |         |     |
| P02   |         |     |         |     |
| C03   |         |     |         |     |
| D03   |         |     |         |     |
| E03   |         |     |         |     |
| F03   |         |     |         |     |
| G03   |         |     |         |     |
| H03   |         |     |         |     |
| I03   |         |     |         |     |
| J03   |         |     |         |     |
| K03   |         |     |         |     |
| L03   |         |     |         |     |
| M03   |         |     |         |     |
| N03   |         |     |         |     |
| O03   |         |     |         |     |
| P03   |         |     |         |     |
| C04   |         |     |         |     |
| D04   |         |     |         |     |
| E04   |         |     |         |     |
| F04   |         |     |         |     |
| G04   |         |     |         |     |
| H04   |         |     |         |     |
| I04   |         |     |         |     |
| J04   |         |     |         |     |
| K04   |         |     |         |     |
| L04   |         |     |         |     |
| M04   |         |     |         |     |
| N04   |         |     |         |     |

| INDEX |         |     |         |     |
|-------|---------|-----|---------|-----|
| LOC   | CONTENT | LOC | CONTENT | LOC |
| C01   |         | D04 |         |     |
| D01   |         | P04 |         |     |
| E01   |         | C05 |         |     |
| F01   |         | D05 |         |     |
| G01   |         | E05 |         |     |
| H01   |         | F05 |         |     |
| I01   |         | G05 |         |     |
| J01   |         | H05 |         |     |
| K01   |         | I05 |         |     |
| L01   |         | J05 |         |     |
| M01   |         | K05 |         |     |
| N01   |         | L05 |         |     |
| O01   |         | M05 |         |     |
| P01   |         |     |         |     |
| C02   |         |     |         |     |
| D02   |         |     |         |     |
| E02   |         |     |         |     |
| F02   |         |     |         |     |
| G02   |         |     |         |     |
| H02   |         |     |         |     |
| I02   |         |     |         |     |
| J02   |         |     |         |     |
| K02   |         |     |         |     |
| L02   |         |     |         |     |
| M02   |         |     |         |     |
| N02   |         |     |         |     |
| O02   |         |     |         |     |
| P02   |         |     |         |     |
| C03   |         |     |         |     |
| D03   |         |     |         |     |
| E03   |         |     |         |     |
| F03   |         |     |         |     |
| G03   |         |     |         |     |
| H03   |         |     |         |     |
| I03   |         |     |         |     |
| J03   |         |     |         |     |
| K03   |         |     |         |     |
| L03   |         |     |         |     |
| M03   |         |     |         |     |
| N03   |         |     |         |     |
| O03   |         |     |         |     |
| P03   |         |     |         |     |
| C04   |         |     |         |     |
| D04   |         |     |         |     |
| E04   |         |     |         |     |
| F04   |         |     |         |     |
| G04   |         |     |         |     |
| H04   |         |     |         |     |
| I04   |         |     |         |     |
| J04   |         |     |         |     |
| K04   |         |     |         |     |
| L04   |         |     |         |     |
| M04   |         |     |         |     |
| N04   |         |     |         |     |