AAEC/E209

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

SPECIAL PROGRAMS, (IPLTEXT AND AEBOOTØI) FOR THE DEVELOPMENT AND USE OF STAND-ALONE PROGRAMS FOR THE IBM 360 COMPUTER

by

P.L. SANGER

August 1970

ISBN 0 642 99371 8

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

SPECIAL PROGRAMS (IPLTEXT AND AEBOOTØ1) FOR THE DEVELOPMENT AND USE OF STAND–ALONE PROGRAMS FOR THE IBM 360 COMPUTER

by

P. L. SANGER

ABSTRACT

A special IPL disk program, IPLTEXT, and a stand-alone program, AEBOOTØ1, are described. Their combined use provides a powerful way of developing and running stand-alone programs. Both programs can be used on any IBM 360 computer that normally operates under OS/360.

A stand-alone program can be assembled and put onto disk in load-module form as a member of a partitioned data set by using standard IBM 360 Job Control Language. By specifying the appropriate DSNAME and member name using the computer console typewriter, this stand-alone program may subsequently be executed by loading it into core storage under the control of the IPLTEXT–AEBOOTØ1 system.

National Library of Australia card number and ISBN 0 642 99371 8

CONTENTS

1. INTRODUCTION

To test special hardware attached to an IBM 360 computer, such as the computer-to-computer link [1,2] between the IBM 360/50I and PDP-9/L computers at the A.A.E.C., it is important to have programs that can check the operation of this hardware in the absence of a complex operating system such as OS/360. Once the hardware responds correctly under the stand-alone system, standard IBM 360 programs can use these hardware facilities under OS control. In the event of subsequent difficulties occurring in the operation of the special hardware, it is also important that diagnostic stand-alone programs can be executed as quickly as possible.

The combined use of the two programs IPLTEXT and AEBOOTØ1 (the IPLTEXT–AEBOOTØ1 system) described in this report provides a powerful way of developing and running stand-alone programs. An example of the use of the system is given in Appendix 1. The stand-alone programs are assembled and put onto disk* by the use of IBM System/360 Job Control Language and can subsequently be loaded into core storage and executed by using the IPLTEXT–AEBOOTØ1 system.

This system allowed each stand-alone program designed to test the Link to be written and thoroughly tested in less than a day, with an individual program being loaded from disk into core storage in a matter of seconds.

2. THE IPLTEXT PROGRAM

IPLTEXT is a special IPL (Initial Program Loading) disk program written for an IBM 360 computer. It is DDØ hex (hexadecimal) bytes long satisfying the condition that an IPL disk program cannot be longer than E29 hex bytes. This length restriction obviously limits the functions that can be performed by an IPL disk program, and IPLTEXT is used to load into core storage and transfer control to a stand-alone program that is stored on the same disk in load module form as the member AEBOOTØ1 of a partitioned data set (PDS).

This means that only stand-alone programs that are stored on the IPL disk in a PDS and have the member name AEBOOTØ1 can be loaded into core storage and executed by using the IPLTEXT program.

However, the version of AEBOOTØ1 described in Section 3 of this report allows stand-alone programs that have any member name to be loaded into core storage and executed.

2.1 Loading the IPLTEXT Program

The IPLTEXT program must be put onto a special track of the disk [3,4] when a disk pack is initialized using the IBM utility programs IEHDASDR or IBCDASDI [5]. At the A.A.E.C. it has been added to all disks other than the System residence volume. IPLTEXT is loaded from disk into core storage when the IPL button on the computer console is pressed, once the appropriate disk address has been selected with the load-unit switches [6].

2.2 Description of Program

Once the IPLTEXT program is loaded into core storage it searches through the Volume Table of Contents (VTOC [7]) of the IPL disk for the first Data Set Control Block (DSCB) that refers to a data set of the right type, that is, a data set that has DSORG=PO (a PDS), occupies a contiguous area of disk storage (ONE EXTENT) and has an undefined record format (RECFM=U). The associated directory blocks of this PDS are then searched for the member AEBOOTØ1.

If AEBOOTØ1 is automatically located the program checks that it has the correct attributes – that is, it is a load module in block format, it does not have an overlay structure, it is executable and its length is suitable for loading into the available core storage. AEBOOTØ1 is finally loaded from disk into core storage from location zero onwards, and control is passed to the relevant entry point by a current Program Status Word (PSW) that has all the channels masked.

---

*Throughout this report the word "disk" can be replaced equally well by the words "2314-disk" or "2311-disk".

If AEBOOTØ1 is not a member of the data set located by the VTOC search, then the computer is put into the WAIT state with the hex address FFFFFF in the Instruction Address Register (IAR) on the computer console – the current PSW in this case is referred to as WAITPSW1 in this report. This indicates that the address of the console typewriter must be given to the IPLTEXT program by pressing the REQUEST button on the typewriter. (Interrupts from other devices are ignored at this stage since the program checks that ATTENTION has been signalled). Once the console typewriter address is specified in this way the message "SPECIFY DSNAME FOR AEBOOTØ1" is typed out after first spacing a line. The typewriter goes to a new line and waits with the PROCEED light on [8] for the name of the data set that has the member AEBOOTØ1 to be specified. The computer is now in the WAIT state with the hex address FFØØFF in the IAR – this current PSW is referred to as WAITPSW2 in this report. The data set name (DSNAME) can be typed on the keyboard using either upper or lower case letters and the input can be cancelled if a typing error is made. A check is made to see whether the specified data set is located in the VTOC, is of the right type and contains the member AEBOOTØ1. If any of these conditions are not satisfied, error messages (to be described in Section 2.4) are typed out and the program again asks for the DSNAME for AEBOOTØ1 to be specified. Three attempts to give a valid DSNAME are allowed before the job is terminated, with the message "JOB TERMINATED" being typed out and the computer put into the WAIT state with all channels masked and the hex address ØØØØØØ in the IAR. At this point, the IPLTEXT program may be restarted if necessary by pressing the PSW RESTART button on the computer console. The console typewriter address does not have to be respecified after restarting in this way.

When there are no data sets of the right type on the IPL disk, or if AEBOOTØ1 is automatically located but does not have the correct attributes, then WAITPSW1 is made the current PSW requiring that the console typewriter address be specified. Once this is done an appropriate error message is typed out and the job terminated. In this case a check should be made to see whether the IPL has been initiated from the correct disk. If so, this disk pack should be checked by running the IBM utility program IEHLIST [5] or the program AEDLIST [9] under OS.

2.3 Restrictions on the Size of AEBOOTØ1

The maximum size of an AEBOOTØ1 program that can be loaded into core storage by IPLTEXT is a function of the size of core storage on the particular IBM 360 computer being used. The IPLTEXT program checks that AEBOOTØ1 can fit into the available core storage before loading it. Calculation of the core storage available for loading AEBOOTØ1 is carried out in the following way, the numbers used being hexadecimal (the example given is for an IBM 360/50I computer):

Size of main core storage	= 80000 bytes
Length Constant for IPLTEXT program	= A12 bytes
Difference	= 7F5EE bytes
Last hex digit made an "E" (due to internal logic of program)	= 7F5EE bytes*
Length of PSW area	= 80 bytes
Difference = core storage available for loading of AEBOOTØ1	= 7F56E bytes

It is important to note that zeros are written through main core storage by the IPLTEXT program in the process of calculating the core size of the IBM 360 computer being used.

2.4 Error Messages from IPLTEXT

The error messages that can be typed out on the console typewriter by the IPLTEXT program can be grouped into two classes.

---

* It is a coincidence in this case that the first difference ended in an E

Class 1 error messages occur after a reply is given to the request to "SPECIFY DSNAME FOR AEBOOTØ1". If the DSNAME "SYSLIB" is specified the error message typed out can be:

"SYSLIB HAS NOT BEEN LOCATED IN VTOC" , or

"SYSLIB DOES NOT HAVE DSORG=PO, RECFM=U, ONE EXTENT", or

"AEBOOTØ1 NOT LOCATED IN SYSLIB".

As mentioned in Section 2.2, three attempts to specify a data set of the right type containing the member AEBOOTØ1 are allowed before the job is terminated.

Class 2 error messages occur if there are no data sets of the right type on the IPL disk, or if AEBOOTØ1 is located but does not have the correct attributes. In this case the job is terminated as soon as the error message is typed out. The Class 2 error messages are:

"NO DATA SETS WITH DSORG=PO, RECFM=U, ONE EXTENT LOCATED IN VTOC",

"AEBOOTØ1 LOCATED BUT IS NOT IN BLOCK FORMAT",

"AEBOOTØ1 LOCATED BUT HAS AN OVERLAY STRUCTURE",

"AEBOOTØ1 LOCATED BUT IS NOT EXECUTABLE", and

"AEBOOTØ1 EXCEEDS THE AVAILABLE STORAGE OF ..... HEX BYTES".

2.5 Contents of General Registers on Exit from IPLTEXT

On exit from IPLTEXT, parameters are passed to AEBOOTØ1 using the general registers as follows:

Register Ø contains the address of the IPL disk.

Register 1 contains the address of a full word that has the value zero and bit Ø set to indicate that there are no parameters to be passed via core storage.

Register 2 contains the console typewriter address or zero depending on whether REQUEST had to be pressed or not.

Register 3 contains the main storage requirements of AEBOOTØ1.

Register 4 contains the size of main core storage.

Register 13 contains the address of a Save area with the second full word showing there is no previous Save area.

Register 14 contains zero.

Register 15 contains the entry point of AEBOOTØ1, and hence can be used as an initial base register.

The address of the IPL disk and the console typewriter address (or zero) are also stored in the interruption code portion of the initial PSW and the I/O OLD PSW respectively.

3. THE AEBOOTØ1 PROGRAM

3.1 Loading the AEBOOTØ1 Program

The AEBOOTØ1 program can be any stand-alone program that is stored in load module form on a disk as a member of a data set of the right type. It must have the correct attributes and can be assembled and put onto disk in this form using IBM System/360 Job Control Language [10] in a

manner similar to that shown in Appendix 2. As discussed in Section 2, AEBOOTØ1 is loaded into core storage by the IPLTEXT program with the initial general register values described in Section 2.5.

3.2 Description of Program

While AEBOOTØ1 can be any stand-alone program, the version referred to in this report is used to load into core storage and transfer control to other stand-alone programs that are stored on the same disk in load module form as members of partitioned data sets.

When the AEBOOTØ1 program is loaded into core storage it checks the contents of register 2 to see if the console typewriter address has already been specified. If not, WAITPSW1 is made the current PSW requiring that the console typewriter address be specified.

Once the typewriter address is known, the typewriter spaces a line, types the message "SPECIFY PROGRAM TO BE EXECUTED" with "DSNAME =" on a new line. At this point WAITPSW2 is made the current PSW, and the program waits for the data set name of the program to be loaded into core storage to be specified.

The DSNAME may be typed using either upper or lower case letters and any errors may be cancelled in the usual way (8). A check is made to see that the specified data set is located in the VTOC of the IPL disk, and that it is of the right type. Three attempts to specify a valid DSNAME are allowed before the job is terminated. A job terminated at this point may be restarted by pressing PSW RESTART on the computer console.

Once a valid DSNAME has been given, the message "MEMBER =" is typed out on a new line, and WAITPSW2 made the current PSW. The member name of the program to be loaded into core storage should now be specified. The member name (up to eight characters long) can be typed using either upper or lower case letters, and any errors can be cancelled in the usual way. Three attempts to specify a valid member name are allowed before the job is terminated. The program can be restarted if necessary by pressing PSW RESTART on the computer console.

When a valid member name has been specified, a check is made to see that the program has the correct attributes. If not, an appropriate error message is typed out and the job terminated.

Finally the specified stand-alone program is loaded from disk into core storage from location zero onwards and control passed to the relevant entry point by loading a PSW with all channels masked.

3.3 Restriction on the Size of the Stand-alone Program to be Loaded by AEBOOTØ1

The maximum size of a stand-alone program that can be loaded by AEBOOTØ1 is calculated using the procedure described in Section 2.3. In this case, however, the length constant for the AEBOOTØ1 program is A38 hex bytes and the last hex digit of the difference between the size of core storage and the length constant must be made a zero. The core storage available for loading a stand-alone program using AEBOOTØ1 on an IBM 360/50I computer is therefore 7F54Ø hex bytes.

3.4 Error Messages from AEBOOTØ1

Three classes of error messages can be typed out on the console typewriter by the AEBOOTØ1 program.

Class 1 error messages occur after a reply has been given to the request "SPECIFY PROGRAM TO BE EXECUTED", "DSNAME =". If the DSNAME "SYSLIB" is specified the error message typed out can be:

"SYSLIB HAS NOT BEEN LOCATED IN VTOC",

or

"SYSLIB DOES NOT HAVE DSORG=PO, RECFM=U, ONE EXTENT".

Three attempts to specify a valid DSNAME are allowed before the job is terminated.

A Class 2 error message occurs after a reply is given to the request "MEMBER =". If the member name "PROG2Ø" is specified the error message typed out would be "PROG2Ø NOT LOCATED IN SYSLIB". Three attempts to specify a valid member name are allowed before the job is terminated.

Class 3 error messages occur if the specified stand-alone program does not have the correct attributes. In this case the job is terminated as soon as the error message is typed out. The Class 3 error messages are:

"PROG2Ø LOCATED BUT IS NOT IN BLOCK FORMAT"

"PROG2Ø LOCATED BUT HAS AN OVERLAY STRUCTURE"

"PROG2Ø LOCATED BUT IS NOT EXECUTABLE", or

"PROG2Ø EXCEEDS THE AVAILABLE STORAGE OF ..... HEX BYTES".

3.5 Contents of General Registers on Exit from AEBOOTØ1

On exit from AEBOOTØ1 parameters are passed to the specified stand-alone program using the general registers as follows:

Register Ø contains the address of the IPL disk.

Register 1 contains the address of a full word that has the value zero and bit Ø set to show that there are no parameters to be passed via core storage.

Register 2 contains the console typewriter address.

Register 3 contains the main storage requirements of the stand-alone program just loaded into core storage.

Register 4 contains the size of main core storage.

Register 13 contains the address of a save area with the second full word showing that there is no previous save area.

Register 14 contains zero.

Register 15 contains the entry point of the stand-alone program just loaded into core storage, and hence can be used as an initial base register.

The address of the IPL disk and the console typewriter address are also stored in the interruption code portion of the initial PSW and the I/O OLD PSW respectively.

4. CONCLUSIONS

The combined use of IPLTEXT and AEBOOTØ1 to load a stand-alone program from disk into core storage provides a powerful way of developing and running stand-alone programs. There is now no need to load these programs through the card reader preceded by an IPL bootstrap loader. The new method of loading stand-alone programs results in a considerable saving of computer time since the programs are brought into core storage from disk, which is much faster than loading them via the card reader. Time is also saved because the card reader stack need not be touched between the IPL needed to load the stand-alone program and the IPL needed to restart OS.

A stand-alone program written to run under the IPLTEXT–AEBOOTØ1 system can be assembled and put onto disk in load module form as a member of a PDS by using standard IBM 360 Job Control Language (see Appendix 2). The ease with which the stand-alone program can be re-assembled and re-run results in a considerable saving of program development time.

Both IPLTEXT and AEBOOTØ1 can be used on any IBM 360 computer that normally operates under OS/360. Details of the stand-alone program to be loaded into core storage are specified via the console typewriter and, where necessary, error messages are also typed out. The console typewriter address is defined by pressing the REQUEST button on the typewriter and this need be done only once when the IPLTEXT–AEBOOTØ1 system is used.

Between July 1969 and January 1970 both of the programs were tested on an IBM 360/50H computer with 2311 disks, an IBM 360/50H computer with 2314 disks and an IBM 360/50I computer with 2314 disks, as the result of the updating of the A.A.E.C.'s IBM 360/50 computer.

The IPLTEXT–AEBOOTØ1 system proved to be very effective when it was used to develop and run stand-alone programs designed to test the computer-to-computer link at the A.A.E.C. This application illustrated the need for this system in an environment where special hardware attached to an IBM 360 computer must be tested in the absence of a complex operating system.

5. ACKNOWLEDGEMENTS

The author wishes to thank Mr. D. J. Richardson for valuable discussions, and Mr. R.P. Backstrom for allowing one of his stand-alone programs to be used as a sample program in this report.

6. REFERENCES

(1) D.J. Richardson (1968). – A generalised computer-to-computer link. AAEC/TM485.

(2) D.J. Richardson (1969). – A generalised computer-to-computer link for an IBM 360 computer. Aust. Computer Journal, 1: 273.

(3) IBM System/360 Component Descriptions-2314 Direct Access Storage Facility and 2844 Auxiliary Storage Control, Form A26–3599–3.

(4) IBM System/360 Component Descriptions–DASD for 2841 (Includes IBM 2311 Disc Storage Drive), Form A26–5988–6.

(5) IBM System/360 Operating System – Utilities, Form C28–6586–9.

(6) IBM System/360 Principles of Operation, Form A22–6821–7.

(7) IBM System/360 Operating System – System Control Blocks, Form C28–6628–4.

(8) IBM System/360 Component Descriptions and Operating Procedures – IBM 1052 Printer – Keyboard Model 7 and IBM 2150 Console, Form A22–6877–4.

(9) D.J. Richardson (1969). – AEDLIST – Disk Analysis Program – Unpublished.

(10) IBM System/360 Operating System – Job Control Language, Form C28–6539–8.

APPENDIX 1

EXAMPLES OF THE COMBINED USE OF IPLTEXT AND AEBOOTØ1 TO LOAD STAND-ALONE PROGRAMS

Two examples of the combined use of IPLTEXT and AEBOOTØ1 to load the stand-alone program CLIP that is a member of the data set AAELIB.LMODA on disk AAEØØ4 (device address 134) are given below. AAELIB.LMODA will be assumed to be a load module having the correct attributes.

Example 1. This applies when AEBOOTØ1 is a member of the first data set of the right type pointed to by the VTOC of AAEØØ4. In this case the following sequence of operations are used to load CLIP into core storage:

(a) IPL from disk AAEØØ4 after selecting the device address 134 in the load-unit switches on the computer console.

(b) The computer will go into the WAIT state with hex address FFFFFF in the IAR on the computer console – WAITPSW1 is made the current PSW. Press the REQUEST button on the console typewriter to specify its address to the AEBOOTØ1 program.

(c) After spacing a line the message "SPECIFY PROGRAM TO BE EXECUTED", followed by "DSNAME = " on the next line, will be typed. The computer will go into the WAIT state with the hex address FFØØFF in the IAR – WAITPSW2 is made the current PSW. Type the data set name "AAELIB.LMODA" on the keyboard.

(d) The message "MEMBER =" will then be typed on a new line and WAITPSW2 made the current PSW. Type the member name "CLIP" on the keyboard.

(e) The stand-alone program CLIP will now be loaded into core storage and control passed to the relevant entry point.

Example 2. This applies when AEBOOTØ1 is not a member of the first data set of the right type pointed to by the VTOC of AAEØØ4. AEBOOTØ1 is assumed to be a member of the data set PLS.TESTLIB of the right type, and the following sequence of operations are used to load CLIP into core storage:

(a) IPL from disk AAEØØ4.

(b) The computer will go into the WAIT state with the hex address FFFFFF in the IAR–WAITPSW1 is made the current PSW. Press REQUEST on the console typewriter to specify the typewriter address to the IPLTEXT program.

(c) After spacing a line the message "SPECIFY DSNAME FOR AEBOOTØ1" is typed out on the console typewriter, followed by a skip to a new line. The computer will go into the WAIT state with the hex address FFØØFF in the IAR–WAITPSW2 is made the current PSW. Type the data set name "PLS.TESTLIB" on the keyboard.

(d) The AEBOOTØ1 program will be loaded into core storage. After spacing a line the message "SPECIFY PROGRAM TO BE EXECUTED", followed by "DSNAME = " on the next line, will be typed and WAITPSW2 made the current PSW. Type the data set name "AAELIB.LMODA" on the keyboard.

(e) The message "MEMBER =" will then be typed on a new line and WAITPSW2 made the current PSW. Type the member name CLIP on the keyboard.

(f) The stand-alone program CLIP will now be loaded into core storage and control passed to the relevant entry point.

APPENDIX 2

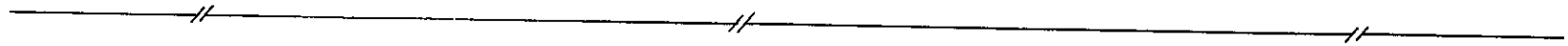
USE OF STANDARD JCL TO PUT A STAND–ALONE PROGRAM ONTO DISK

A listing of a sample program is given on the following pages, showing the use of standard JCL to put a stand-alone program onto disk as a member of a partitioned data set. The program was written by Mr. R.P. Backstrom and was one of the stand-alone programs used to test the link between the IBM 360/50I and PDP-9/L computers at the A.A.E.C.

Persons interested in writing stand-alone programs to run under the IPLTEXT–AEBOOTØ1 system should note the manner in which the first 80 hex bytes of the sample program are set up as a PSW area, the use of register 15 to load the program base register and the use of some of the parameters passed to this program via the general registers.

(continued)

```
//LINK     JOB '        ,0031',R.P.BACKSTROM,MSGLEVEL=1
**JST  70.252  18.01.02 00000 R.17
//AEBOOT01 EXEC ASMFCL,PARM.ASM='NODECK,LOAD'
XXASM    EXEC  PGM=IEUASM,PARM='LOAD'                                          00000010
XXSYSGO    DD  DSN=SYSOBJ,DISP=OLD,DCB=(RECFM=FB,BLKSIZE=80,LRECL=80)          00000020
XXSYSLIB   DD  DSN=SYS1.MACLIB,DISP=OLD                                        00000030
XXSYSPRINT DD  SYSOUT=A                                                        00000040
XXSYSPUNCH DD  SYSOUT=B                                                        00000050
XXSYSUT1   DD  DSN=SYSUT1,DISP=OLD                                             00000060
XXSYSUT2   DD  DSN=SYSUT3,DISP=OLD                                             00000070
XXSYSUT3   DD  DSN=SYSUT2,DISP=OLD                                             00000080
//ASM.SYSIN DD *
IEF236I ALLOC. FOR LINK     ASM      AEBOOT01
IEF237I SYSGO    ON 135
IEF237I SYSLIB   ON 137
IEF237I SYSUT1   ON 137
IEF237I SYSUT2   ON 135
IEF237I SYSUT3   ON 136
IEF237I SYSIN    ON 00C
```



```
SYMBOL   TYPE ID  ADDR  LENGTH LD ID          EXTERNAL SYMBOL DICTIONARY          PAGE    1
                                                                                 18.01  9/09/70

READ#     SD  01 000000 00048A
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000000                                1 READ#    CSECT
000000 0004000000000088               2          DC    X'00040000',A(RD1)      RESTART PSW
000008 0000000000000000               3          DC    8D'0'                   IPL CCW
000048 0000029800000000               4          DC    A(DEV),A(0)             INITIAL CAW
000050 FFFFFF0000000000               5          DC    X'FFFFFF0000000000'     TIMER
000058 0004000000000088               6          DC    X'00040000',A(RD1)      EXTERNAL NEW PSW
000060 0006000000000F00               7          DC    X'0006000000000F00'     SVC NEW PSW
000068 0006000000FF00FF               8          DC    X'0006000000FF00FF'     PROGRAM ERROR NEW PSW
000070 0002000000FFFFFF               9          DC    X'0002000000FFFFFF'     MACHINE CHECK PSW
000078 8004000000000AC               10          DC    X'80040000',A(RD5)      I/O NEW PSW
000080                               11          USING *,12                    LOAD BASE
000080 18CF                          12 RD       LR    12,15                   REGISTER
000082 18A2                          13          LR    10,2                    STORE TYPEWRITER
000084 4020 C33C          003BC      14          STH   2,TPAD                  ADDRESS
000088 8000 C2E5    00365            15 RD1      SSM   ZER                     TEMPORARILY NON INTERRUPTIBLE
00008C 4120 C218          00298      16          LA    2,DEV                   SET
000090 5020 0048          00048      17          ST    2,72(0,0)               CAW
000094 4120 C02C          000AC      18 RD2      LA    2,RD5                   SET I/O
000098 5020 007C          0007C      19          ST    2,124(0,0)              NEW PSW
00009C 9D00 A000    00000            20 RD3      TIO   0(10)                   WAIT FOR
0000A0 4770 C01C          0009C      21          BNZ   RD3                     TYPEWRITER
0000A4 9C00 A000    00000            22          SIO   0(10)                   START TYPING
0000A8 8200 C208    00288            23 RD4      LPSW  WPSW                    WAIT FOR TYPEWRITER
0000AC D501 003A C33C 003BC          24 RD5      CLC   58(2),TPAD              WAS IT THE TYPEWRITER ?
0000B2 4770 C028          000A8      25          BNE   RD4                     IF NOT, WAIT
0000B6 9180 0044    00044            26          TM    68(0),X'80'             WAS IT ATTENTION ?
0000BA 4710 C008          00088      27          BO    RD1                     IF SO, RESTART
0000BE 9109 0044    00044            28          TM    68(0),X'09'             WAS IT CANCEL ?
0000C2 4710 C008          00088      29          BO    RD1                     IF SO, RESTART ALSO
0000C6 4160 0000          00000      30          LA    6,0                     CLEAR REG 6
0000CA 4360 C2DD          0035D      31          IC    6,DL+9                  CONVERT
0000CE 4366 C340          003C0      32          IC    6,TABLE(6)              EBCDIC
0000D2 8C60 0004          00004      33          SRDL  6,4                     CHARACTERS
0000D6 4360 C2DC          0035C      34          IC    6,DL+8                  INTO
0000DA 4366 C340          003C0      35          IC    6,TABLE(6)              HEXADECIMAL
0000DE 8C60 0004          00004      36          SRDL  6,4                     DIGITS
0000E2 4360 C2DB          0035B      37          IC    6,DL+7                  ENSURE THAT
0000E6 4366 C340          003C0      38          IC    6,TABLE(6)              THE CHANNEL
0000EA 5960 C2B0          00330      39          C     6,MASK                  NUMBER IS
0000EE 47B0 C008          00088      40          BNL   RD1                     LESS THAN 7
0000F2 4356 C2DE          0035E      41          IC    5,SYSMSK(6)             CALCULATE APPROPRIATE
0000F6 4250 C210          00290      42          STC   5,CH                    INTERRUPT MASK (CHANNEL
0000FA 8D60 0008          00008      43          SLDL  6,8                     'X' PLUS TYPEWRITER)
0000FE 4060 C33E          003BE      44          STH   6,TPAD+2                STORE I/O ADDRESS
000102 1886                          45          LR    11,6
000104 41F0 0800          00800      46 RD6      LA    15,2048                 2**19 COUNTDOWN
000108 89F0 0008          00008      47          SLL   15,8                    FOR A 2.5 SECOND
00010C 46F0 C08C          0010C      48          BCT   15,*                    DELAY BEFORE
000110 8000 C2E5    00365            49          SSM   ZER                     READING 24 BYTES
000114 4120 C230          002B0      50          LA    2,READ                  TEMPORARILY NON INTERRUPTIBLE
000118 5020 0048          00048      51          ST    2,72(0,0)               SET
00011C 4120 C0B0          00130      52          LA    2,RD8                   CAW
000120 5020 007C          0007C      53          ST    2,124(0,0)              SET I/O
000124 9C00 B000    00000            54          SIO   0(11)                   NEW PSW
000128 4770 C198          00218      55          BNZ   MSG                     READ 24 BYTES
                                                                              PRINT MESSAGE IF UNSUCCESSFUL
```

```
00012C 8200 C210       00290          56 RD7      LPSW  CH                OTHERWISE, WAIT
000130 D501 003A C33C 0003A 0038C     57 RD8      CLC   58(2),TPAD         WAS IT THE TYPEWRITER ?
000136 4770 C0C6              00146    58          BNE   RD9               WAS IT THE EXPECTED DEVICE ?
00013A 9180 0044       00044          59          TM    68(0),X'80'       WAS IT ATTENTION ?
00013E 4710 C008              00088    60          BO    RD1               IF SO, RESTART
000142 47F0 C0AC              0012C    61          B     RD7               IF NOT, IGNORE IT
000146 D501 003A C33E 0003A 0038E     62 RD9      CLC   58(2),TPAD+2       END OF 24 BYTE TRANSMISSION ?
00014C 4770 C0AC              0012C    63          BNE   RD7               IF NOT, IGNORE IT
000150 F384 C308 C2E8 00388 00368     64          UNPK  LINE(9),LN(5)       UNPACK
000156 F384 C310 C2EC 00390 0036C     65          UNPK  LINE+8(9),LN+4(5)   SECTIONS
00015C F384 C318 C2F0 00398 00370     66          UNPK  LINE+16(9),LN+8(5)  OF THE
000162 F384 C320 C2F4 003A0 00374     67          UNPK  LINE+24(9),LN+12(5) 24 BYTES
000168 F384 C328 C2F8 003A8 00378     68          UNPK  LINE+32(9),LN+16(5) READ
00016E F384 C330 C2FC 003B0 0037C     69          UNPK  LINE+40(9),LN+20(5) TRANSLATE
000174 DC2F C308 C1C4 00388 00244     70          TR    LINE(48),TAB-240    INTO EBCDIC
00017A 4120 0000              00000    71          LA    2,0               CLEAR REG 2
00017E 4320 0047              00047    72          IC    2,71              LOAD RESIDUAL BYTE COUNT
000182 1322                            73          LCR   2,2               COMPLEMENT THIS NUMBER
000184 4120 2018              00018    74          LA    2,24(0,2)         ACTUAL NO OF BYTES READ
000188 1232                            75          LTR   3,2               IS IT ZERO ?
00018A 4780 C180              00200    76          BZ    RD16              PRINT 'NO TRANSMISSION'
00018E 8930 0001              00001    77          SLL   3,1               DOUBLE NO OF BYTES READ
000192 4230 C247              002C7    78          STC   3,PRINT+15        FOR PRINTING TWO HEX
000196 4E20 C200              00280    79          CVD   2,BTNO            DIGITS PER BYTE
00019A F342 C200 C206 00280 00286     80          UNPK  BTNO(5),BTNO+6(3)   CONVERT BYTE COUNT
0001A0 D201 C300 C201 00380 00281     81          MVC   BT(2),BTNO+1        TO A DECIMAL NUMBER
0001A6 95F0 C300       00380          82          CLI   BT,C'0'           1ST DIGIT ZERO ?
0001AA 4770 C132              001B2    83          BNE   RD10              IF NOT, CONTINUE
0001AE 9240 C300       00380          84          MVI   BT,C' '           OTHERWISE, BLANK THE ZERO
0001B2 9502 C247       002C7          85 RD10     CLI   PRINT+15,X'02'    NO OF BYTES = 1 ?
0001B6 4770 C142              001C2    86          BNE   RD11             IF NOT, SET 'S' IN LINE
0001BA 9240 C307       00387          87          MVI   BT+7,C' '         BLANK THE 'S'
0001BE 47F0 C146              001C6    88          B     RD12              BEGIN TYPING
0001C2 92E2 C307       00387          89 RD11     MVI   BT+7,C'S'         SET 'S' IN 'BYTES'
0001C6 8000 C2E5       00365          90 RD12     SSM   ZER               TEMPORARILY NON INTERRUPTIBLE
0001CA 4120 C238              002B8    91          LA    2,PRINT           SET
0001CE 5020 0048              00048    92          ST    2,72(0,0)         CAW
0001D2 4120 C16A              001EA    93          LA    2,RD15            SET I/O
0001D6 5020 007C              0007C    94          ST    2,124(0,0)        NEW PSW
0001DA 9D00 A000       00000          95 RD13     TIO   0(10)             WAIT FOR
0001DE 4770 C15A              001DA    96          BNZ   RD13              TYPEWRITER
0001E2 9C00 A000       00000          97          SIO   0(10)             START TYPING
0001E6 8200 C208       00288          98 RD14     LPSW  WPSW              WAIT
0001EA D501 003A C33C 0003A 003BC     99 RD15     CLC   58(2),TPAD        WAS IT THE TYPEWRITER ?
0001F0 4770 C166              001E6   100          BNE   RD14              IF NOT, IGNORE IT
0001F4 9180 0044       00044         101          TM    68(0),X'80'       WAS IT ATTENTION ?
0001F8 4710 C008              00088   102          BO    RD1               IF SO, RESTART
0001FC 47F0 C084              00104   103          B     RD6               OTHERWISE, READ ANOTHER 24 BYTES
000200 8000 C2E5       00365         104 RD16     SSM   ZER               TEMPORARILY NON INTERRUPTIBLE
000204 4120 C260              002E0   105          LA    2,TRA             SET
000208 5020 0048              00048   106          ST    2,72(0,0)         CAW
00020C 4120 C16A              001EA   107          LA    2,RD15            SET I/O
000210 5020 007C              0007C   108          ST    2,124(0,0)        NEW PSW
000214 47F0 C15A              001DA   109          B     RD13              TYPE 'NO TRANSMISSION'
000218 4730 C1C2              00242   110 MSG      BC    3,MSG2            NOT OPERATIONAL ?
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                F01FEB69   9/09/70

00021C 4720 C18A            0023A   111          BC    2,MSG1                    CHANNEL BUSY ?
000220 4120 C248            002C8   112          LA    2,MS1                     CSW STORED
000224 F384 C275 0040 002F5 00040   113          UNPK  ERRMSG1+13(9),64(5,0)     UNPACK AND
00022A F384 C27D 0044 002FD 00044   114          UNPK  ERRMSG1+21(9),68(5,0)     TRANSLATE CSW
000230 DC0F C275 C1C4 002F5 00244   115          TR    ERRMSG1+13(16),TAB-240    FOR PRINTING
000236 47F0 C1C6            00246   116          B     MSG3                      TYPE MESSAGE
00023A 4120 C250            002D0   117 MSG1     LA    2,MS2                     CHANNEL
00023E 47F0 C1C6            00246   118          B     MSG3                      BUSY
000242 4120 C258            002D8   119 MSG2     LA    2,MS3                     NOT OPERATIONAL
000246 8000 C2E5      00365         120 MSG3     SSM   ZER                       TEMPORARILY NON INTERRUPTIBLE
00024A 5020 0048            00048   121          ST    2,72(0,0)                 STORE CAW
00024E 4120 C1E6            00266   122          LA    2,MSG6                    SET I/O
000252 5020 007C            0007C   123          ST    2,124(0,0)                NEW PSW
000256 9D00 A000      00000         124 MSG4     TIO   0(10)                     WAIT FOR
00025A 4770 C1D6            00256   125          BNZ   MSG4                      TYPEWRITER
00025E 9C00 A000      00000         126          SIO   0(10)                     START TYPING
000262 8200 C208      00288         127 MSG5     LPSW  WPSW                      WAIT
000266 D501 003A C33C 0003A 003BC   128 MSG6     CLC   58(2),TPAD                TYPEWRITER ?
00026C 4770 C1E2            00262   129          BNE   MSG5                      IF NOT, WAIT
000270 9180 0044      00044         130          TM    68(0),X'80'               ATTENTION ?
000274 4710 C008            00088   131          BO    RD1                       IF SO, RESTART
000278 47F0 C084            00104   132          B     RD6                       READ ANOTHER 24 BYTES
00027C 00000000
000280 0000000000000000             133 BTNO     DC    D'0'                      NO OF BYTES
000288 8002000000000FF              134 WPSW     DC    X'80020000000000FF'       TYPEWRITER WAIT PSW
000290 0002000000000000             135 CH       DC    X'0002000000000000'       CHANNEL 'X' PLUS TYPEWRITER
000298 0900035360000001             136 DEV      CCW   9,NL,X'60',1
0002A0 0100035460000007             137          CCW   1,DL,X'60',7
0002A8 0A00035B20000003             138          CCW   X'0A',DL+7,X'20',3
0002B0 0200036820000018             139 READ     CCW   2,LN,X'20',24
0002B8 0900038060000008             140 PRINT    CCW   9,BT,X'60',8
0002C0 0900038820000018             141          CCW   9,LINE,X'20',24
0002C8 090002E820000001D            142 MS1      CCW   9,ERRMSG1,X'20',29
0002D0 0900030620000001B            143 MS2      CCW   9,ERRMSG2,X'20',27
0002D8 090003212000000F             144 MS3      CCW   9,ERRMSG3,X'20',15
0002E0 090003442000000F             145 TRA      CCW   9,TRN,X'20',15
0002E8 C3E2E640E2E3D6D9             146 ERRMSG1  DC    CL30'CSW STORED ='
000306 C3C8C1D5D5C5D340             147 ERRMSG2  DC    CL27'CHANNEL OR SUB CHANNEL BUSY'
000321 D5D6E340D6D7C5D9             148 ERRMSG3  DC    CL15'NOT OPERATIONAL'
000330 00000007                     149 MASK     DC    F'7'
000334 F0F1F2F3F4F5F6F7             150 TAB      DC    CL16'0123456789ABCDEF'
000344 D5D640E3D9C1D5E2             151 TRN      DC    CL15'NO TRANSMISSION'
000353 40                           152 NL       DC    C' '
000354 C4C5E5C9C3C54040             153 DL       DC    CL10'DEVICE'
00035E 80C0A090888482               154 SYSMSK   DC    X'80C0A090888482'         CH 'X' PLUS TYPEWRITER
000365 00                           155 ZER      DC    X'00'
000366 0000
000368 0000000000000000             156 LN       DC    6F'0'
000380 404040C2E8E3C5E2             157 BT       DC    CL8'   BYTES'
000388 0000000000000000             158 LINE     DC    13F'0'
0003BC 00000000                     159 TPAD     DC    F'0'
0003C0 0000000000000000             160 TABLE    DC    129X'00'
000441 0A0B0C0D0E0F                 161          DC    X'0A0B0C0D0E0F'           LOWER CASE A-F
000447 0000000000000000             162          DC    58X'00'
000481 0A0B0C0D0E0F                 163          DC    X'0A0B0C0D0E0F'           UPPER CASE A-F
```

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                    F01FEB69   9/09/70
000487 0000000000000000              164          DC    41X'00'
000480 0001020304050607              165          DC    X'000102030405060708090'
000080                               166          END   RD
```

RELOCATION DICTIONARY

POS.ID	REL.ID	FLAGS	ADDRESS
01	01	0C	000004
01	01	0C	000048
01	01	0C	00005C
01	01	0C	00007C
01	01	08	000299
01	01	08	0002A1
01	01	08	0002A9
01	01	08	0002B1
01	01	08	0002B9
01	01	08	0002C1
01	01	08	0002C9
01	01	08	0002D1
01	01	08	0002D9
01	01	08	0002E1

SYMBOL	LEN	VALUE	DEFN	REFERENCES							
BT	00008	000380	0157	0081	0082	0084	0087	0089	0140		
BTNO	00008	000280	0133	0079	0080	0080	0081				
CH	00008	000290	0135	0042	0056						
DEV	00008	000298	0136	0004	0016						
DL	00010	000354	0153	0031	0034	0037	0137	0138			
ERRMSG1	00030	0002E8	0146	0113	0114	0115	0142				
ERRMSG2	00027	000306	0147	0143							
ERRMSG3	00015	000321	0148	0144							
LINE	00004	000388	0158	0064	0065	0066	0067	0068	0069	0070	0141
LN	00004	000368	0156	0064	0065	0066	0067	0068	0069	0139	
MASK	00004	000330	0149	0039							
MSG	00004	000218	0110	0055							
MSG1	00004	00023A	0117	0111							
MSG2	00004	000242	0119	0110							
MSG3	00004	000246	0120	0116	0118						
MSG4	00004	000256	0124	0125							
MSG5	00004	000262	0127	0129							
MSG6	00006	000266	0128	0122							
MS1	00008	0002C8	0142	0112							
MS2	00008	0002D0	0143	0117							
MS3	00008	0002D8	0144	0119							
NL	00001	000353	0152	0136							
PRINT	00008	0002B8	0140	0078	0085	0091					
RD	00002	000080	0012	0166							
RD1	00004	000088	0015	0002	0006	0027	0029	0040	0060	0102	0131
RD10	00004	0001B2	0085	0083							
RD11	00004	0001C2	0089	0086							
RD12	00004	0001C6	0090	0088							
RD13	00004	0001DA	0095	0096	0109						
RD14	00004	0001E6	0098	0100							
RD15	00006	0001EA	0099	0093	0107						
RD16	00004	000200	0104	0076							
RD2	00004	000094	0018								
RD3	00004	00009C	0020	0021							
RD4	00004	0000A8	0023	0025							
RD5	00006	0000AC	0024	0010	0018						
RD6	00004	000104	0046	0103	0132						
RD7	00004	00012C	0056	0061	0063						
RD8	00006	000130	0057	0052							
RD9	00006	000146	0062	0058							
READ	00008	0002B0	0139	0050							
READ#	00001	000000	0001								
SYSMSK	00007	00035E	0154	0041							
TAB	00016	000334	0150	0070	0115						
TABLE	00001	0003C0	0160	0032	0035	0038					
TPAD	00004	0003BC	0159	0014	0024	0044	0057	0062	0099	0128	
TRA	00008	0002E0	0145	0105							
TRN	00015	000344	0151	0145							
WPSW	00008	000288	0134	0023	0098	0127					
ZER	00001	000365	0155	0015	0049	0090	0104	0120			

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
  250 PRINTED LINES

```
IEF285I   SYSOBJ                                                KEPT
IEF285I   VOL SER NOS= AAE003.
IEF285I   SYS1.MACLIB                                           KEPT
IEF285I   VOL SER NOS= AAE001.
IEF285I   SYSOUT                                                SYSOUT
IEF285I   VOL SER NOS= OUTPUT.
IEF285I   SYSUT1                                                KEPT
IEF285I   VOL SER NOS= AAE001.
IEF285I   SYSUT3                                                KEPT
IEF285I   VOL SER NOS= AAE003.
IEF285I   SYSUT2                                                KEPT
IEF285I   VOL SER NOS= AAE002.
**EOS ASM       18.02.02 00060 SECS 0000
//LKED   EXEC  PGM=IEWL,PARM=(XREF,LIST,NCAL)                                00000090
XXSYSLIN   DD  DSN=SYSOBJ,DISP=OLD                                          00000100
XX         DD  DDNAME=SYSIN                                                 00000110
//LKED.SYSLMOD DD DSN=PLS.LINKLIB,VOL=REF=PACK6,DISP=OLD
X/SYSLMOD  DD  DSN=&AAEC(USERSPGM),                                         00000120
XX             VOL=REF=PACK3,SPACE=(TRK,(40,20,01)),DISP=(NEW,PASS)         00000130
XXSYSPRINT DD  SYSOUT=A                                                     00000140
XXSYSUT1   DD  DSN=SYSUT2,DISP=OLD                                          00000150
//LKED.SYSIN DD *
IEF236I ALLOC. FOR LINK     LKED     AEBOOT01
IEF237I SYSLIN   ON 135
IEF237I          ON 00C
IEF237I SYSLMOD  ON 132
IEF237I SYSUT1   ON 136
```

```
F128-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED XREF,LIST,NCAL
           VARIABLE OPTIONS USED - SIZE=(204800,30720)              DEFAULT OPTION(S) USED
IEW0000     NAME LNKRD(R)
```

CROSS REFERENCE TABLE

CONTROL SECTION NAME	ORIGIN	LENGTH	ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
READ#	00	4BA								

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION	LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION

```
ENTRY ADDRESS       80
TOTAL LENGTH       4C0

****LNKRD     NOW REPLACED IN DATA SET
```

```
IEF285I   SYSOBJ                                          KEPT
IEF285I   VOL SER NOS= AAE003.
IEF285I   PLS.LINKLIB                                     KEPT
IEF285I   VOL SER NOS= AAE006.
IEF285I   SYSOUT                                          SYSOUT
IEF285I   VOL SER NOS= OUTPUT.
IEF285I   SYSUT2                                          KEPT
IEF285I   VOL SER NOS= AAE002.
**EOS LKED     18.02.18 00015 SECS 0000
//
**EOJ LINK     18.02.19 0.03 HOURS
```