27
1-27-76
.6 NTIS

IS-3773

USERS MANUAL FOR OS910, THE SDS-910 OPERATING SYSTEM

MASTER

Leonard C. Moon

AMES LABORATORY, USERDA
IOWA STATE UNIVERSITY
AMES, IOWA

Date Transmitted: December 1975

PREPARED FOR THE U. S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
UNDER CONTRACT W-7405-eng-82

DISCLAIMER

IS-3773

USERS MANUAL FOR OS910, THE SDS-910 OPERATING SYSTEM

By

Leonard C. Moon

Ames Laboratory, ERDA
Iowa State University
Ames, Iowa 50010

Date Transmitted: December 1975

PREPARED FOR THE U.S. ENERGY AND RESEARCH AND DEVELOPMENT ADMINISTRATION UNDER CONTRACT NO. W-7405-eng-82

Available from: National Technical Information Service
U. S. Department of Commerce
P.O. Box 1553
Springfield, VA 22161

Price: Microfiche $2.25
Paper Copy $5.00

TABLE OF CONTENTS

ABSTRACT

This document is designed for the users of the OS910 real-time operating system for the SDS-910 -- PDP-15 computer network. It describes what information the user must provide to the system for successful operation. Included is a description of all programmed operators (POPS) that will be of use to the user.

Distribution List

Chicago Pantent Office	1
OSERDA-TIC	27
Ames Laboratory Library	9
L. C. Moon	53
Total	90

1. INTRODUCTION

This document contains a description of those facilities needed by the user to interact with the OS910 system for the SDS-910 -- PDP-15 computer network. All the programmed operators (POPS) are discussed so the user can know what is needed by the POP and what will be returned (if anything) by the POP.

2. USER PROGRAM STRUCTURE

The structure of a user's program is as follows:

ID/KEY TABLE

$TASK_1$

$TASK_2$

.
.
.
.

$TASK_n$

The ID/KEY TABLE, which is supplied by the user, must have the form:

BIT	0——8	9	10————23
WORD 1	ID #	a/i	# KEY ENTRIES
2	ADDRESS OF $INTERRUPT_1$		
3	ADDRESS OF $INTERRUPT_1$ HANDLER		
4	ADDRESS OF $INTERRUPT_2$		
5	ADDRESS OF $INTERRUPT_2$ HANDLER		
6	OPEN ENTRY POINT		
7	CLOSE ENTRY POINT		
8	KEY 1 BUFFER LOCATION		
9	KEY 1 ENTRY POINT		
	. . .		
2N+6	KEY N BUFFER LOCATION		
2N+7	KEY N ENTRY POINT		

Taking each section in order, we have:

1. WORD 1:

   bits 0-8: the ID number (in the range 511-537) assigned by the system staff.

   bit 9: the a/i (active/inactive) flag for the user. If this bit is set to zero, then the user is assumed to be in the active (running) state.

bits 10-23: maximum key value used.

2. WORDS 2 and 3:

   Addresses of the user's first interrupt location and the user interrupt handler for it. If the user does not have an interrupt for his or her experiment, then put:

```
WORD2    OCT    77777777
WORD3    BRM    2.
```

3. WORDS 4 and 5:

   Addresses of the user's second interrupt location and the user interrupt handler for it. If the user does not have an interrupt or does not have two interrupts for his or her experiment, then put:

```
WORD4    OCT    77777777
WORD5    BRM    2.
```

4. WORD 6:

   User's OPEN routine entry point.

5. WORD 7:

   User's CLOSE routine entry point.

6. THE ENTRIES:

   These are pairs of data buffer location and entry point associated with each key used in the PDP-15 ALECS program. To conserve space in the SDS-910 the user is asked to assign the keys in numerical order.

This table is used in the following manner.

1. When the task that resides in the PDP-15 performs an OPEN for the EXPERIMENT file, the a/i bit is set to zero for that user. The user's interrupt(s) are connected through the system and is(are) set active. Finally, the user's open routine is scheduled for execution.

2. When a task is to be dispatched, the a/i bit is checked. If the bit is set to one (inactive) the task is flushed from the system.

3. When information is sent to the SDS-910 for the user, the key, say i, associated with the data, is used to find the address of that area of the user's storage which will contain the data (the storage area must be large enough to contain both control information i.e., two words of storage for a read return, and the data). Associated with the buffer address is a service routine entry point address. Using the address of the (user supplied) buffer, the entry point address for the service routine, and the user's ID value, a task queue node is created, thus scheduling the execution of the user's service routine.

4. When the user incurs a real-time (RT) error (by using an invalid interface (EOM) address for his experiment) the RT error servicing routine resets the a/i bit, sends information to the PDP-15 that the user has incurred an RT error and then allows the user to finish execution of the task.

5. When the user's code in the PDP-15 performs a CLOSE of the EXPERIMENT file, a record is transmitted to the SDS-910 to indicate this. The SDS-910 system then resets the a/i bit to inactive, turns off the interrupts associated with this user, and schedules the user's CLOSE routine.

The ID/KEY TABLE is user generated. The table is easy to create. For a ID number of 524, and a total of two keys, the table can be constructed as follows:

OCT	52440002	
OCT	242	ADDRESS of $INTERRUPT_1$
BRM	CHAR	ADDRESS of $INTERRUPT_1$ HANDLER
OCT	243	ADDRESS of $INTERRUPT_2$
BRM	CHAR	ADDRESS of $INTERRUPT_2$ HANDLER
HLT	OPEN	OPEN ENTRY POINT
HLT	CLOSE	CLOSE ENTRY POINT
HLT	BLOCK	KEY ONE BUFFER AREA
HLT	READ	KEY ONE ENTRY POINT
HLT	DATA	KEY TWO BUFFER AREA
HLT	SEND	KEY TWO ENTRY POINT

The first word, declared OCT 52440002, has the field values: bits 0-8 set to 524_8 i.e., the user ID number, bit 9 set to 1 (inactive) and bits 10-23 set to 00002, the total number of keys. The routine names have been arbitrarily chosen. The declarations using

```
        HLT       name
```

create a word containing the storage address for 'name'.

If a user does not desire to have a task automatically scheduled for a specific key, a simple artifice must be used. The service routine entry point field must reference a (user supplied) routine consisting of nothing more than:

```
        RTNE      PZE
                  BRR       *-1
```

Thus, no work will be done by the "servicing routing", but rather will just return. This can be done for OPEN and CLOSE routines if the user doesn't need to initialize the experiment in some way.

3. USER INTERRUPT PROCESSING

All user interrupts are handled through the system. The system will turn-on the user's interrupts when an OPEN of the EXPERIMENT file for the user is executed in the corresponding PDP-15 code. The system will turn-off the user's interrupts when a CLOSE of the EXPERIMENT file is executed in the corresponding PDP-15 code.

The user can turn-on and turn-off his or her interrupts with the use of the POPS, PON and POFF, respectively. PON needs the address of the user interrupt servicing routine in the A register. This gives a user the flexibility to have more then one interrupt code per interrupt by just furnishing the system the address of the appropriate interrupt routine address.

```
EXAMPLE:  POFF       242           TURN-OFF INTERRUPT 242
          LDA        INTHAN        LOAD ADDRESS OF INTERRUPT HANDLER
          PON        241           TURN-ON INTERRUPT 241
           .
           .
           .
           .
     INTHAN HLT      INT           ADDRESS OF INTERRUPT HANDLER
```

Each user is assigned a particular priority for his or her experiment in the SDS-910 computer. This means that when an interrupt for the user comes into the SDS-910 from their experiment, the code that is executed to handle the interrupt is executed in a priority mode. Therefore, all interrupt handlers should only do those things necessary to "field" the interrupt, i.e., turn-off the interrupt with a POFF and schedule a task to handle what the interrupt is saying needs to be done. Then exit to leave the priority mode and the system will give control to the task you have scheduled.

There is one interrupt error that affects the user, but is not within his sphere of control. This is the real-time (RT) error interrupt. An RT error is incurred by incorrectly addressing (EOM) a user's experiment. In all cases, an interrupt, with the ONCODE set to zero, is scheduled for transmission to the PDP-15. This will cause the PDP-15 system to shutdown all of the user's codes in the PDP-15 and the PDP-15 system will send a CLOSE of the EXPERIMENT file to the SDS-910 system.

4. USER TASK SCHEDULING

Users have the ability to effect the scheduling of tasks. There are two classes of user scheduled tasks: normal, and clock dependent.

A normal task is one that will do, say, angle derivation for one of the experiments, or a task that OS910 schedules for the user, to treat data sent to the user from the DEC PDP-15. To effect the scheduling of such a task, the user executes the POP:

```
SKDTSK    0
```

Here, the operand field is not used, but cannot be eliminated due to assembly language restrictions. The contents of the A and B registers supply the required parameters. The A register must contain the entry point address of the task being scheduled, and the B register must contain either the address of the (possible) set of parameters, or be set to -1 (indicating no parameters). The parameter list address is passed back to the user in the A register when he is given control.

There are three clock queues that the user may have tasks scheduled on. The associated clocks have approximate periods of 12.8 msec. (fast clock), 102.4 msec. (medium clock), and 1.6384 sec. (slow clock). The clocks may be profitably employed when tasks must be executed after performing a delay operation. This could happen when a shaft on a test bed is to be swung thru an arc, and the amount of time required for the operation is both known and at least as much as the fast clock. The user could then schedule a task on the appropriate clock, and "delay" processing.

To perform the scheduling, the user:

1. Sets the contents of the A register to the entry point address of the task to be scheduled;

2. Sets the B register contents to reference any parameters;

3. Executes one of:

SKDFST	0	Fast Clock
SKDMED	0	Medium Clock
SDKSLO	0	Slow Clock

where the operand, although required by the assembler, is not used, and is arbitrary.

When the appropriate clock ticks, the user is given control. The parameter list address is passed to the user in the A register.

5. DEC PDP-15 TO SDS-910 COMMUNICATION

It should be kept in mind that programs residing in the SDS-910 have an associated major program residing in the DEC PDP-15. The DEC PDP-15 program treats the SDS-910 as nothing more than a special purpose file structure, and interrupt generator. Thus, it will perform OPEN's, CLOSE's, READ's, and WRITE's, to the EXPERIMENT file (which is the SDS-910, and the user's servicing programs residing in the SDS-910).

An OPEN of the EXPERIMENT file will cause a message to be transmitted to the SDS-910 with a key field value of zero. Upon receipt of a key zero record, the system activates the user's interrupts and schedules the user's OPEN routine for execution.

A CLOSE of the EXPERIMENT file causes a "shutdown" message to be transmitted to the SDS-910. Upon receipt of this message, OS910 will schedule the shutdown of the user's SDS-910 resident code. The system will also turn-off the user's interrupts.

A WRITE to the EXPERIMENT file will transmit data to the buffer area in the SDS-910 user's code. Because of this action, a user supplied task will also be scheduled to treat the data. If the user wants to schedule a task in the SDS-910 from the PDP-15 a "dummy" write can be executed which will schedule the task associated with the key in the PDP-15. Note: The user must still supply a data area for the SDS-910 system to transfer the "dummy" information.

A READ of the EXPERIMENT file does not transmit any data. Rather, it requests the SDS-910 resident user's program to return some information (thus, the term "read request"). Upon receipt of this message, OS910 will store, in the user supplied buffer area, the length parameters, and the field termed "key location in CAL". Thus the data area must be two words longer than the length of data that will be sent back to the PDP-15 program. It is recommended that the user have an event variable that can be waited on in his or her PDP-15 code associated with the READ of the experiment file. Failing to do this can lead to use of a data area that doesn't have the proper information in it.

Please note that when numeric values are transmitted to a user's program in the SDS-910, the 18 bit data values are expanded to 24 bit form. The 18 bits of data are placed in bits 6-23 of the word, and bit 6 is expanded, to the left, throughout bits 0-5 of the SDS-910 word. Thus, if the user is transmitting other than FIXED data values the user will have to correct the values in his or her data area.

6. SDS-910 TO DEC PDP-15 COMMUNICATION

There are two methods of communication that an OS910 user has available. First, one is able to initiate a read return (a reply to a DEC PDP-15 initiated READ). Secondly, the user may transmit an ONCODE corresponding to an interrupt signal. Indirectly, the user may cause an RT error interrupt signal (ONCODE = 0) to be transmitted by executing an invalid address for a user's experiment.

There are three types of read returns: FIXED, FLOAT, and DMA returns. Each of these requires a specific data structure. These are:

1. One word read return (FIXED):

WORD 1:	Length of transfer
WORD 2:	"Key location in CAL" value
WORD 3:	FIXED data value

where the first word of this area is the same as that sent with the read request, initially.

2. Two word read return (FLOAT):

WORD 1:	Length of transfer
WORD 2:	"Key location in CAL" value
WORD 3:	Data value one, EXPONENT
WORD 4:	Data value two, MANTISSA

NOTE: If you are going to send back a FLOAT number, you must first compact it into its 18 bit form because the SDS-910 system will just truncate the first 6 high order bits to make-up the data transfer buffer. The user can send float numbers from the SDS-910 to the PDP-15 into a fixed or bit array that is twice as long as the number of words being sent.

3. DMA read return (FIXED or FLOAT):

WORD 1:	Number of data values to send
WORD 2:	"Key location in CAL" value
WORD 3:	Data value one
.	.
.	.
.	.
WORD (n+2)	Data value n

where the "number of data values to SEND" does not include the "key location in CAL" field.

These data areas are referenced by the operand field values of the POPS:

1. SNDFIX ref:

   The reference is to a FIXED read return area;

2. SNDFLT ref:

   The reference is to FLOAT read return area;

3. SENDMA ref:

   The reference is to a DMA read return area.

The invoked POP will:

1. Create a regular send queue (SNDQUE) entry that references the appropriate transmission routine and a buffer (or list, thereof) containing the data;

2. Compress the data from 24 bit format to 18 bit format, as required by the DEC PDP-15;

3. Insert the data into the buffer node(s) as required.

This action, which amounts to dynamic buffering, is done to ensure the integrity of the user's data. If the data were to be left in the user's area while awaiting transmission, there would need to be some notification technique developed to ensure that the user did not write over his or her own data.

There is one other send POP, SNDINT. This POP sends an interrupt to the user's associated PDP-15 code ON EXPT block. There, the ONCODE built-in function can be used to obtain the value sent from the SDS-910.

USE:

```
        LDA       ONCODE
        SNDINT    0
```

Here ONCODE contains the ONCODE that will be sent to the PDP-15 program.

7. FLOATING POINT POPS

All of the floating point routines, except the conversion routines, have been taken from the SDS-910 POPS Manual and modified to work with OS910. The ones which are available are addition (FSA), subtraction (FSS), multiplication (FSM), division (FSD), and negation (FSN). Two other POPS fixed to float conversion (FFL) and float to fixed conversion (FFI) (for SDS-910 numbers only) are available. All of the above POPS are for single precision floating point numbers.

The conversion routines assume the value to be converted is in the A register (for fixed values) or the A and B registers (for float values) and return with the value converted in the A or A and B registers.

The A and B registers should be loaded with the value the user wants negated (in the case of FSN) or with the first operand, A register should contain the mantissa and the B register should contain the exponent. The address field specified in the POP instruction should reference the second operand field (EXPONENT, MANTISSA). The result of the operation will be in the A and B registers.

EXAMPLE A/B:

```
        LDB     A
        LDA     A+1
        FSD     B
         .
         .
         .
         .
         .
A       OCT                 /EXPONENT for A
        OCT                 /MANTISSA for A
B       OCT                 /EXPONENT for B
        OCT                 /MANTISSA for B.
```

The result of the division will be in the A and B registers when control is given back to the user's program.

8. FIXED POINT POP

All of the fixed point routines, have been taken from the SDS-910 POPS manual and modified to work with OS910. Multiplication (MUL), Division (DIV) and a branching routine, skip if equal (SKE) are available to the user.

The first operand should be in the A register and the second operand in the address field of the POP.

EXAMPLE A/B:

```
        LDA     A
        DIV     B
         .
         .
         .
         .
    A   OCT                 /FIXED VALUE for A
    B   OCT                 /FIXED VALUE for B.
```

The result is in the A register when control is given back to the user.

9. PAUSE POP

This routine is designed to allow a user to cause a delay in execution. The address field will allow the user to return to some other point in his or her code when control is restored to the program.

EXAMPLE:

```
        SKN     VALUE
        BRU     *+2
        PAUSE   *-2
         .
         .
         .
```

This example will cause a delay in execution until some other code (presumably in an interrupt routine) sets VALUE negative. PAUSE is a variable delay depending on the number of users in the system at the time.

The user is asked to use the PAUSE POP instead of an infinite loop so other users of the system can be executing codes while the user is waiting for the desired event to happen. NOTE: The PAUSE POP cannot be used in an interrupt routine.

TABLE 1

OS910 POPS

LOCATION	POP	PURPOSE
100	SKDFST	Schedule a task on the fast clock.
101	SKDMED	Schedule a task on the medium clock.
102	SKDSLO	Schedule a task on the slow clock.
103	SKDTSK	Schedule a task for execution.
104	SNDFIX	Send PDP-15 a fixed value on a read return.
105	SNDFLT	Send PDP-15 a float value on a read return.
106	SENDMA	Send PDP-15 a DMA transfer on a read return.
107	SNDINT	Send PDP-15 an interrupt.
110	FFL	Fixed to float conversion.
111	DIV	Fixed divide.
112	MUL	Fixed multiply.
113	FSN	Float negate.
114	FSD	Float divide.
115	FSM	Float multiply.
116	FSS	Float subtract.
117	FSA	Float add.
120	FFI	Float to fixed conversion.
121	PAUSE	Create pause in execution.
122	SKE	Skip on equal.
123	PON	Turn-on user interrupt.
124	POFF	Turn-off user interrupt.

TABLE 2

SYSTEM USER CONSTANTS AND ADDRESSES

LOCATION	PURPOSE
4	Special clock time counter.
13	Constant -1.
14	Constant 0.
15	Constant 1.
16	Constant 2.
17	Constant 3.
20	Constant 4.
22	System cold start routine address.
23	System tasker address.
24	RAID entry point address.

APPENDIX A

SDS-910 SYSTEM AND PROGRAM LOADER ROUTINES:

These routines load SDS-910 program into the SDS-910 computer when used in conjunction with the PDP-15 LOADER. There are two modes of operation;

1. initial (system and program) load,
2. load under system control.

When the initial load mode is chosen, the program zeroes core, sets-up the interrupt vectors, and loads the system and users programs into the SDS-910. If a load under system control is done the program just loads the user program into the SDS-910. The program is on both paper tape and PDP-15 disk.

HOW TO USE INITIAL SYSTEM LOADER (IPL):

1. Read in the SDS-910 system loader paper tape on the SDS-910.
   a. Put RUN/IDLE switch to idle.
   b. Turn register select switch to A register. Push start button.
   c. Turn register select switch to C register. Push start button.
   d. Turn register select switch to B register. Push start button.
   e. Turn register select switch to X register. Push start button.
   f. Turn register select switch to C register.
   g. Put paper tape loader in paper tape reader. Make sure the load run knob on the paper tape reader is in LOAD when loading and RUN when you move to the next step.

h. Turn power on to paper tape reader.

i. Move RUN/IDLE switch to RUN.

j. Lift FILL switch.

Paper tape should read in.

l. Turn power off to paper tape reader.

m. Re-wind paper tape.

2. On the PDP-15 side:

a. Zero PDP-15 core. See systems operation book.

b. Set address register to 17720_8 by using the console switches. Read in PDP-15 system using the system loader tape.

c. On the PDP-15 console:

1. J100 CR

2. Press continue on the console.

d. CNTL A, the system will respond with

REQ, INQ, OR ZAP?

e. Reply REQ CNTL D, the system will respond with

TYPE IN USER LIBRARY NAME?

f. Reply with UNIVERSAL CNTL D.

The system will respond with

TYPE IN TASK NAME?

g. Reply LOADER CNTL D.

The system will respond

USER ID IS ##### CORE ADDRESS IS #####
TASK DISPATCHABLE

h. The loader routine on the PDP-15 side will ask:

DATA SET NAME?

Reply SYSTEM 910 CNTL D.

i. The loader will load the system into the SDS-910 and then return with:

NORMAL END, NEXT DATA SET NAME?

Reply with your program data set name, CNTL D.

j. The loader will load that program into the SDS-910 and return again with:

NORMAL END, NEXT DATA SET NAME?

Reply with your next program (if you have one) CNTL D. If you don't have any more programs then reply CNTL D. This will cause the loader in both computers to end normally.

Now you can start any program by entering CNTL A. The system will respond again

REQ, INQ, OR ZAP?

Reply REQ CNTL D and reply to the rest of the questions with the library and data set that you want to execute.

TO USE THE LOADER UNDER SYSTEM CONTROL:

On the PDP-15 side:

1. On any teletype press CNTL A.

The system will respond

REQ, INQ, OR ZAP?

Reply REQ CNTL D.

The system will respond

TYPE IN USER LIBRARY NAME?

Reply with the library name that the SDS-910 program resides in, CNTL D. The system will respond with

TYPE IN TASK NAME?

Reply LOADER CNTL D. The system will respond with

USER ID IS ##### CORE ADDRESS IS #####
TASK DISPATCHABLE

The PDP-15 loader will ask:

DATA SET NAME?

Reply with the name of the SDS-910 program data set, CNTL D.

The loader will load the data set with the SDS-910 and return with:

NORMAL END, NEXT DATA SET NAME?

If you have more data sets to be loaded across the interface, specify them at this time. Otherwise, reply with a CNTL D. This will end the loader program.

NOTE: It is assumed that the user will be using the system location which has a zero in it and not have a location in his own code for this purpose. Therefore, the loader does not place zeroes into a user's area. This can be a problem if the user is loading over an area that has not been zeroed and he wants the constant zero in his own area.

APPENDIX B

HOW TO SET-UP A DATA SET ON THE PDP-15 DISK TO BE LOADED ACROSS THE INTERFACE INTO THE SDS-910:

This routine is used to build a program data set which will be executed in the SDS-910.

HOW TO USE:

1. On the PDP-15 computer use DISKORG to create a norm data set with record length = 44. The number of sectors needed is figured by counting the number of cards of the SDS-910 program to be put out on disk and dividing by 5. Add 1 to the number of sectors if there is any fraction left over from the division.

On any PDP-15 teletype type in a

	CNTL A
System responds	REQ, INQ, OR ZAP?
Reply	REQ CNTL D
System responds	TYPE IN USER LIBRARY NAME?
Reply	DISK CNTL D
System responds	TYPE IN TASK NAME?
Reply	DISKORG CNTL D
System responds	USER ID IS ##### CORE ADDRESS IS ##### TASK DISPATCHABLE
DISKORG asks	TYPE IN DATE-YYMMDD?
Reply	RESPOND WITH CURRENT YEAR, MONTH, DAY CNTL D
DISKORG asks	TYPE IN OPCODE?
Reply	ADS CNTL D

DISKORG asks	DIRECTORY NAME?
Reply	RESPOND WITH YOUR LIBRARY NAME CNTL D
DISKORG asks	DATA SET NAME?
Reply	RESPOND WITH THE NAME OF YOUR SDS-910 PROGRAM CNTL D
DISKORG asks	DATA SET TYPE?
Reply	NORM CNTL D
DISKORG asks	RECORD LENGTH?
Reply	44 CNTL D
DISKORG asks	NUMBER OF SECTORS?
Reply	RESPOND WITH NUMBER CALCULATED ABOVE CNTL D
DISKORG asks	CREATOR'S NAME?
Reply	YOUR INITIALS CNTL D
DISKORG asks	UPDATE?
Reply	YES CNTL D
DISKORG asks	GET INPUT FROM CARD READER?
Reply	NO CNTL D
DISKORG asks	TYPE IN OPCODE?
Reply	If you want to create more data sets reply ADS CNTL D and go through the sequence again. If not reply CNTL D.
	DISKORG should terminate normally at this point.

2. Use CARDREAD routine to read in SDS-910 program cards through the PDP-15 card reader.

On a PDP-15 teletype type in a

	CNTL A
System responds	REQ, INQ, OR ZAP?
Reply	REQ CNTL D
System responds	TYPE IN USER LIBRARY NAME?
Reply	Respond with name of library that the SDS-910 program will reside in CNTL D
System responds	TYPE IN TASK NAME?
Reply	CARDREAD CNTL D
System responds	USER ID IS ##### CORE ADDRESS IS ##### TASK DISPATCHABLE
CARDREAD	DATA SET NAME?
Reply	Reply with above created data set name CNTL D

Put SDS-910 program cards in the PDP-15 card reader hopper. Push the start button on the card reader. CARDREAD will then read in the cards.

If any errors are encountered while reading the cards the program will respond with the appropriate error message. If the READ CHECK light comes on on the PDP-15 card reader, simply take all of the cards out of the stacker and put them on the front of the hopper again. The program will respond with

DATA SET NAME?

just reply with the appropriate program data set name, CNTL D. On successful completion the program will ask

DATA SET NAME?

If you have more programs to load onto the PDP-15 disk reply with the next program name, CNTL D. If not reply CNTL D.

The program should end normally (completion code 000000).