DECUS
PROGRAM LIBRARY

DECUS NO.	8-928 (DA)
TITLE	VAXCOM: PDP-8/VAX-11 COMMUNICATIONS PROGRAM
AUTHOR	R.M. DOESBURG AND A.T. WALLIS
COMPANY	WELLCOME MEDICAL RESEARCH INSTITUTE DUNEDIN, NEW ZEALAND
DATE	MARCH 1982
SOURCE LANGUAGE	PAL-8

VAXCOM: PDP-8/VAX-11 Communications Program

Version: March 1982

Author: P.M. Doesburg, A.T. Wallis, Wellcome Medical Pesearch Institute, Dunedin, New Zealand

Operating System: OS/8 V3D

Source Language: PAL-8

Memory Pequired: 8KW

Special Hardware Pequired: KL8-J Terminal Control/Asynchronous Data Interface.

VAXCOM is a PDP-8 program which allows terminal operations with a remote VAX-11, and transfer of ASCII files. A standard DEC KL8-J serial interface board is used to connect the VAX communication line to the PDP-8. This program sends all input from the PDP-8 terminal display. ASCII files can be transmitted in either direction between computers, using any OS/8 device as source or destination. File transfer is initiated and synchronised by a VAX/VMS DCL Command Procedure, which transmits a code that causes the PDP-8 program to call the OS/8 command decoder. The XON/XOFF protocol is used to control the rate of data transmission.

Note: There are two DCL Command Procedures required to run on VAX that are supplied with the program.

VAXCOM: PDP8 - VAX 11 Communications Program

Version: March 82

Authors: RM Doesburg, AT Wallis

Wellcome Medical Research Institute,
University of Otago,
Dunedin,
New Zealand.

Abstract

VAXCOM is a PDP8 program that allows terminal operations with a remote VAX 11, and transfer of ASCII files. A standard DEC KL8-J serial interface board is used to connect the VAX communication line to the PDP8. The program sends all input from the PDP8 keyboard to the VAX and relays the VAX responses to the PDP8 terminal display. ASCII files can be transmitted in either direction between computers, using any OS/8 device as source or destination. File transfer is initiated and synchronised by a VAX/VMS DCL Command Procedure, which transmits a code that causes the PDP8 program to call the OS/8 command decoder. The XON/XOFF protocol is used to control the rate of data transmission.

Introduction

This is a description of a program to perform computer-to-computer communications between a DEC VAX-11 and PDP-8. Use is made of a data-communications line from the VAX terminal handler, which is connected by a KL8-J serial interface board to the PDP8. The problem of synchronising the transfer of files between the computers is simplified with the use of VAX/VMS Command Procedures to provide a software interface.

Program VAXCOM is designed to perform two functions:

1.Link the PDP8 terminal directly to the VAX. The program initially runs in this mode, in which all characters typed at the PDP8 keyboard are transmitted to the remote VAX 11 and responses from the VAX are displayed on the PDP8 terminal.

2.Transfer ASCII files between the VAX and PDP8 computers. Transfers are initiated by running a VAX command procedure, which sends a code character to the PDP8 so that an OS/8 file name can be requested from the operator. The command procedure also modifies the characteristics of the terminal interface for the duration of the file transfer, and sends an end-of-file character on completion of a transfer to the PDP8.

Using the program

Before running the program, the KL8-J interface board (or its equivalent) should be adjusted and installed. The device number of the board should be changed to the value used by the program; alternatively the source program can be re-assembled using IOT's altered to use the existing device number (see "Setting up", below).

When the VAX 11 communications line has been connected to the PDP8 interface, start the program by typing

```
R VAXCOM
```

Normal terminal operations with the VAX will now be available. An initial null character is sent by the program, and the response from the VAX depends on whether this terminal line is currently logged on.

-If not logged on, the VAX will ask for user name and password.

-If already logged on by other means, the VAX will now respond with the prompting '$'.

The PDP8 terminal acts as if it is directly connected to the VAX. With one exception, any typed character will be transmitted. The only character not passed on is the control character CTRL/B which forces an immediate jump to the OS/8 keyboard monitor. The VAX may be left in a logged-on state - this might be useful if using the VAX handler (DECUS 8-921).

File transfers

A file is sent as a stream of characters, using a set of OS/8 routines (ref 2) to pack and unpack normal blocks of OS/8 data. With a few exceptions, any ASCII character may be transmitted. The longest permitted line (a string terminated by line-feed) is 511 characters. Null characters are not transmitted, and the character CTRL/Z (ASCII <SUB>) is recognised as end-of-file.

To begin a file transfer, the program has to receive a special control character from the VAX. Receipt of the code character leads to a jump to one of the file transfer routines, which call the OS/8 command decoder.

During transfer from the VAX, all data received is sent to the specified OS/8 file until an end-of-file character (CTRL/Z, ASCII <SUB>) arrives. During transfer to the VAX, data from the named OS/8 file is transmitted and a CTRL/Z sent at the end-of-file.

File transfers are performed under the control of Command Procedures running on the VAX. A different procedure is used for each direction, and example listings are given below. Command files containing these procedures must be present on the VAX for transfers to take place. The VAX editor may be used to load these small files.

After the command procedures have been loaded to the VAX directory (with the filenames VAXTOPDP8.COM and PDP8TOVAX.COM) file transfers may be performed. A command procedure is run by using the DCL @ command in front of the procedure name, with the name of the source or destination VAX file typed afterwards. The proper formats are given below, and some examples. (Note that it is not necessary to type the .COM extension, which is the default assumed for a command procedure).

Transfer from PDP8 to VAX

```
Type @PDP8TOVAX vax-file-name <CR>
(OS/8 Command Decoder prompt '*' appears)
Type OS/8 file name or device name <CR>
(transfer takes place)
(VAX/VMS prompt '$' re-appears at completion)
```

Transfer from VAX to PDP8

```
Type @VAXTOPDP8 vax-file-name <CR>
(OS/8 command decoder prompt appears)
Type destination OS/8 file name or device <CR>
(transfer takes place)
('$' appears at completion)
```

For example, to transfer a file called DATA.DA from PDP8 device RXA1: into the VAX file DATA.DAT:

```
$ @PDP8TOVAX DATA.DAT  (return)  ('$' is printed by the VAX)
*RXA1:DATA.DA (return)
 ...transfer takes place
$
```

To print a file from the VAX on the PDP8 line-printer:

```
$ @VAXTOPDP8 PRINTOUT.LIS (return)
*LPT:< (return)
 ...transfer
$
```

Aborting transfer

The operator may stop a file transfer at any time by typing the control character CTRL/Y or CTRL/Z. The effect of CTRL/Y differs depending on the direction of transfer; the VAX will close the file as it exists, so no information is lost. A CTRL/Y does not close an output file on the PDP8. Typing CTRL/Z will ensure that a PDP8 output file is closed.

(The familiar CTRL/C does work, but may lead to a return to the OS/8 monitor if received while a device handler is running. It would then be necessary to run VAXCOM again, and type CTRL/Y to return to the VAX monitor)

Setting up

The program VAXCOM has been set up to use an interface board with device numbers 36 (input) and 37 (output). The KL8-J board can be given these values using jumpers, and the baud rate and data formats (parity, start and stop bits) set (ref 1). If the above device numbers are already in use, the PAL8 source program should be altered and re-assembled. The four IOT's to be modified are at the beginning of the source VAXCOM.PA.

VKSF	test input flag
VKRB	read input register
VTSF	test output flag
VTLS	write to output register

The keyboard and display are assumed to have the standard device numbers 03 and 04. The IOT's KSF, KRB, TSF and TLS are used to access these devices, and should be re-defined if the terminal interface has a non-standard device number. After assembly, the binary file VAXCOM.BN should be saved with the following addresses:

```
SAVE SYS VAXCOM 200-777,16000-17377;200
```

DCL Command Procedures

Listings are given below of two Command Procedures which are used to control the transfer of files between computers. They must be present on the VAX before a file transfer can be performed.

The functions of these procedures are:

Transfer to VAX (procedure PDP8TOVAX)

-obtain name of VAX file to be created
-alter line width of terminal to maximum allowed;
 suppress echoing of input and broadcasts to terminal
-send control character (ASCII <DC1> or <XON>) to
 indicate readiness for transfer from PDP8
-COPY input from PDP8 to the named file
-restore normal terminal characteristics at end-file

Transfer to PDP8 (procedure VAXTOPDP8)

-obtain name of VAX file to be copied
-set maximum line-width for terminal,
 allow form-feed characters,
 suppress broadcasts,
 insert nulls after line-feed
-send control character (ASCII <STX>) to
 signal impending transfer to PDP8
-COPY file to PDP8
-send end-of-file code (CTRL/Z, ASCII <SUB>)
-restore normal terminal characteristics

It is assumed that the XON/XOFF protocol is in effect (HOSTSYNC and TTSYNC set in the terminal characteristics)

These DCL procedures should be created with the .COM extension using the DCL CREATE command or the VAX editor. The second to last line in each, which restores the normal terminal characteristics, should be altered if the PDP8 terminal in use is different from a VT52. See the DCL "SET TERMINAL" command for information about alternative terminal types (ref 3). Information about Command Procedures may be found in the DEC manual "Guide to using Command Procedures" (ref 4)

NB (1) Each line of a command procedure begins with '$'.
(2) '%O' is the symbol for an Octal literal.

PDP8TOVAX.COM

```
$ ON ERROR THEN GOTO RESTORE
$ ON CONTROL_Y THEN GOTO RESTORE
$ IF P1 .EQS. "" THEN INQUIRE P1 "FILE NAME"
$ SET TERMINAL/NOECHO/WIDTH=511/NOBROADCAST
$ XON[0,32]=%021
$ WRITE SYS$OUTPUT XON
$ COPY SYS$OUTPUT 'P1'
$RESTORE:
$ SET TERMINAL/VT52
$ EXIT
```

VAXTOPDP8.COM

```
$ ON ERROR THEN GOTO RESTORE
$ ON CONTROL_Y THEN GOTO RESTORE
$ IF P1 .EQS. "" THEN INQUIRE P1 "FILE NAME"
$ SET TERM/FORM/WIDTH=511/LFFILL=2/NOBROADCAST
$ STX[0,32]=%02
$ SUB[0,32]=%032
$ WRITE SYS$OUTPUT STX
$ COPY 'P1' SYS$OUTPUT
$ WRITE SYS$OUTPUT SUB
$RESTORE:
$ SET TERMINAL/VT52
$ EXIT
```

Acknowledgements

This work was supported by the Medical Research Council of New Zealand. We would like to thank Professor FO Simpson for his encouragement and support of the project.

References

(1) KL8-J Engineering Drawings (C)1975 Digital Equipment Corporation

(2) OS/8 Software Support Manual DEC-S8OSSMA-A-D
Appendix F: OS/8 Input/Output Routines

(3) VAX/VMS Command Language User's Guide DEC AA-D023B-TE
(March 1980)

(4) VAX/VMS Guide to Using Command Procedures DEC AA-H782A-TE
(March 1980)

NOTE regarding possible bug in TECO-8

There appears to be a bug in TECO-8 (or VTEDIT). When stepping through a certain page, line-by-line, the system would hang in field 3, the keyboard ignored, and the edited page lost. This problem seemed to depend on the size of the page, because after several unsuccessful attempts at editing I had to use another editor to make the required alteration (insertion of a line). The page now was accepted by TECO.

A similar problem occurred on another occasion. Attempts to insert characters into a line would cause the system to stop responding to keyboard input (including ^C) and a loop in field 3 would be evident from the console display.

I haven't excluded some form of "interrupt sensitivity"; does TECO enable interrupts? However, the position dependence of the first problem mentioned (it would always hang as you stepped from Top down to the same line) suggests that a character count or pointer is becoming corrupt when a certain critical page size is reached.

R Doesburg

```
        /       VAX COMM: TTY AND FILE TRANSFERS
        /
        /       VERSION: MARCH 1982
        /
        /       AUTHORS: RM DOESBURG, AT WALLIS
        /                WELLCOME MEDICAL RESEARCH INSTITUTE
        /                UNIVERSITY OF OTAGO,
        /                DUNEDIN,
        /                NEW ZEALAND
        /
        /       THIS PROGRAM RUNS UNDER OS/8 VERSION 3D, IN CONJUNCTION
        /       WITH A REMOTE VAX-11/780 RUNNING THE VAX/VMS OPERATING
        /       SYSTEM, VERSION 2.3
        /
        /       THE PDP8 TERMINAL IS CONNECTED TO THE VAX FOR ORDINARY
        /       INTERACTIVE OPERATIONS, AND FILES CAN BE TRANSFERRED
        /       BETWEEN COMPUTERS. FILE TRANSFERS REQUIRE THE PRESENCE
        /       OF COMMAND PROCEDURES ON THE VAX TO SEND SYNCHRONISING
        /       CONTROL CHARACTERS BEFORE AND AFTER THE TRANSFER. THIS
        /       PROGRAM COULD RUN WITH ANY COMPUTER THAT SUPPORTS THE
        /       XON/XOFF PROTOCOL, WITH A SMALL SOFTWARE INTERFACE TO
        /       SEND THE SYNCHRONISING CONTROL CHARACTERS DESCRIBED BELOW.
        /
        /       THE FOLLOWING IOTS CONTROL A KL8 SERIAL INTERFACE
        /       CONNECTED TO OUR VAX LINE. (CURRENTLY IT RUNS AT
        /       4800 BAUD)
        /       DEVICE 36 (INPUT), 37 (OUTPUT).

6361            VKSF=6361               /TEST VAX INPUT FLAG
6366            VKRB=6366               /READ  "    "
6371            VTSF=6371               /TEST  "    "    "
6376            VTLS=6376               /READ  "    "

        /       TO GET THE PROGRAM RUNNING ON A DIFFERENT SYSTEM,
        /       THE FOUR IOTS MUST BE ADJUSTED TO THE CORRECT DEVICE
        /       CODE, AND THIS FILE ASSEMBLED USING PAL8. THE COMMANDS
        /       TO ASSEMBLE, LOAD AND SAVE THE CORE IMAGE ARE
        /
        /       .R PAL8
        /       *VAXCOM,LPT:<VAXCOM
        /       .R ABSLDR
        /       *VAXCOM$
        /       .SAVE SYS VAXCOM 200-777,16000-17377;200
        /
        /       THIS PROGRAM IS STARTED BY TYPING
        /       .R VAXCOM
        /
        /       THE VAX WILL RESPOND BY PROMPTING FOR USERNAME AND
        /       PASSWORD (UNLESS ALREADY LOGGED ON, WHEN A '$' SHOULD
        /       APPEAR).
        /
        /       NOW THE PDP8 ACTS AS A 'GO-BETWEEN' BETWEEN THE KEYBOARD,
        /       DISPLAY AND THE REMOTE VAX. ANY VAX COMMANDS CAN BE GIVEN
```

```
/       AND THE OUTPUT DISPLAYED ON THE TERMINAL. HOWEVER, A FILE
/       CAN BE TRANSFERRED BETWEEN THE TWO COMPUTERS BY INITIATING
/       A SPECIAL DCL COMMAND PROCEDURE ON THE VAX. THE VAX FILENAME
/       IS OBTAINED FIRST, THEN THE PDP8 COMMAND DECODER PROMPT APPEARS
/       FOR INPUT OF THE OS/8 FILENAME (FOLLOWED BY '<' FOR OUTPUT FILE).
```

```
                /
                /       THIS PROGRAM HAS 5 SECTIONS:
                /
                /        VAXTTY:        CONNECT PDP8 TERMINAL WITH VAX
                /        VAXCOPY:       COPY DATA FROM VAX TO OS/8 DEVICE
                /        PDPCOPY:        .     .   .   OS/8 DEVICE TO VAX
                /        USRCALL:       SET UP JUMPS TO CHARACTER I/O ROUTINES
                /        OS8IO:         OS/8 CHARACTER ORIENTED I/O INTERFACE
                /
                /       VAXTTY
                /       VAX - TTY COMMUNICATION LINK
                /
                /       CHARACTERS TYPED AT KEYBOARD ARE TRANSMITTED TO
                /       THE VAX, AND CHARACTERS RECEIVED FROM THE VAX ARE
                /       PASSED ON TO THE TERMINAL.
                /
                /       TYPE CTRL/B TO GO BACK TO THE OS/8 KEYBOARD MONITOR (THIS
                /       CHARACTER IS NOT PASSED ON TO THE VAX).
                /
                /       CERTAIN CHARACTERS FROM THE VAX ARE GIVEN SPECIAL TREATMENT.
                /       THE TWO CONTROL CHARACTERS RECOGNISED ARE:
                /
                /               ASCII 2 <STX>           PRECEDES FILE TRANSFER FROM VAX.
                /                                       CAUSES JUMP TO SECTION VAXCOPY.
                /
                /               ASCII 21(OCTAL) <XON>       REQUESTS A FILE TRANSFER TO THE VAX.
                /                                       CAUSES JUMP TO SECTION PDPCOPY.
                /
                /       A  VAX/VMS DCL COMMAND PROCEDURE CONTROLS FILE TRANSFERS BY
                /       SENDING THE REQUIRED CONTROL CHARACTER AND PERFORMING A VAX
                /       COPY TO OR FROM THE TERMINAL. THE COMMAND PROCEDURE SHOULD ALTER
                /       THE TERMINAL CHARACTERISTICS TO ENSURE SUCCESSFUL FILE TRANSFERS,
                /       IN ADDITION TO SENDING THE APPROPRIATE CONTROL CHARACTERS.
                /       CHARACTERISTICS ALTERED      ARE:
                /
                /       FILE TO PDP8:
                /         WIDTH SET TO 511    ALLOWS LONG RECORDS
                /         FORM  SET                 FORM-FEEDS TRANSMITTED WITHOUT TRANSLATION
                /         LFFILL SET TO 2     2 NULLS AFTER <LF> (BREATHING SPACE)
                /         BROADCAST RESET     FORBID SYSTEM BROADCASTS
                /
                /       FILE TO VAX:
                /         NOECHO SET                ONLY ERROR MESSAGES WILL BE SENT BY VAX
                /         WIDTH SET TO 511    ALLOW LONG RECORDS
                /         BROADCAST RESET     FORBID SYSTEM BROADCASTS
                /
                /       THE COMMAND PROCEDURE SHOULD ENSURE THAT NORMAL TERMINAL
                /       CHARACTERISTICS ARE RESTORED AT THE FINISH, AND AFTER A
                /       CONTROL_Y OR ERROR CONDITION.
                /
        0000            FIELD 0
        0200            *200
00200   7300    VAXTTY, CLA CLL             /INITIALISE FLAGS, ALL DEVICES
00201   6376            VTLS
00202   6046            TLS
```

```
00203  6366           VKRB
00204  6036           KRB
00205  7300           CLA CLL

00206  6031  KEYRD,   KSF                 /ANYTHING TYPED AT KEYBOARD?
00207  5223           JMP       VAXRD     /NO, LOOK TOWARDS VAX
00210  6036           KRB                 /YES
00211  0377           AND       (177      /REMOVE PARITY BIT
00212  3255           DCA       STORE
00213  1376           TAD       (2        /CHECK FOR ^B ON KEYBOARD
00214  7041           CIA                 /OPERATOR REQUESTED EXIT
00215  1255           TAD       STORE
00216  7450           SNA
00217  5656           JMP I     OSEXIT    /EXIT TO OS8
00220  7300           CLA CLL
00221  1255           TAD       STORE     /RECOVER KBD CHAR FOR TRANSMISSION

00222  4250  VAXWAIT, JMS       PUTVAX    /SEND ACC TO VAX
00223  6361  VAXRD,   VKSF                /LOOK FOR CHAR FROM VAX
00224  5206           JMP       KEYRD     /IF NOT, CHECK VDU KEYBOARD
00225  6366           VKRB                /GET CHAR FROM VAX
00226  3255           DCA       STORE
00227  1376           TAD       (2        /CHECK FOR <STX> FROM VAX
00230  7041           CIA                 /THIS IS CODE TO SIGNAL
00231  1255           TAD       STORE     /IMPENDING TRANSFER OF A FILE
00232  7450           SNA
00233  5775'          JMP       VAXCOPY   /READ FILE FROM VAX TO OS8

00234  7300  CONT,    CLA CLL
00235  1374           TAD       (21       /CHECK FOR <XON> FROM VAX
00236  7041           CIA                 /THIS MEANS VAX WAITS FOR
00237  1255           TAD       STORE     /A FILE FROM PDP8
00240  7450           SNA
00241  5773'          JMP       PDPCOPY   /WRITE FILE FROM OS8 TO VAX

00242  7300  CONT2,   CLA CLL
00243  1255           TAD       STORE     /PASS CHAR FROM VAX TO VDU
00244  6046           TLS
00245  6041  TTYWAIT, TSF
00246  5245           JMP       .-1
00247  5206           JMP       KEYRD     /LOOK FOR KEYBD CHAR AGAIN

00250  0000  PUTVAX,  0                   /SUBROUTINE SENDS 1 CHAR TO VAX
00251  6371           VTSF
00252  5251           JMP       .-1
00253  6376           VTLS
00254  5650           JMP I     PUTVAX    /RETURN FROM SUBROUTINE

00255  0000  STORE,   0
00256  7600  OSEXIT,  7600
00373  0600
00374  0021
00375  0400
00376  0002
00377  0177
```

```
0400            PAGE
```

```
                        /
                        /       VAXCOPY: COPY FILE FROM VAX TO PDP8
                        /
                        /       BRANCHES TO OS/8 COMMAND DECODER TO OBTAIN FILE-NAME
                        /       FROM OPERATOR FOR OUTPUT.
                        /
                        /       COPIES DATA FROM VAX TO A TEMPORARY MEMORY BUFFER
                        /       STARTING AT BOTTOM+1 (1000 OCTAL) UNTIL <LF> (LINE-
                        /       FEED). <XOFF> IS SENT TO HALT VAX TRANSMISSION WHILE
                        /       THE BUFFER IS PASSED CHARACTERWISE TO OS/8. <XON> IS
                        /       SENT TO ENABLE VAX TRANSMISSION AGAIN. END OF FILE
                        /       IS GIVEN BY CTRL/Z RECEIVED FROM VAX.
                        /
                        /       TYPE CTRL/Z TO FORCE IMMEDIATE END OF FILE.
                        /
                        /       TYPE CTRL/C OR CTRL/Y TO ABORT TRANSFER. CONTROL WILL RETURN
                        /       EITHER TO VAXTTY ABOVE, OR TO OS/8 IF CTRL/C WAS PICKED
                        /       UP BY THE DEVICE HANDLER. IN THIS CASE YOU WILL HAVE TO
                        /       RESTART THIS PROGRAM AND SEND CTRL/C TO THE VAX (WHICH
                        /       WILL STILL BE IN THE <XOFF> STATE).
                        /
00400  7300  VAXCOPY,CLA CLL
00401  1377          TAD     (23     /  <XOFF> CONTROL CHAR
00402  4776'         JMS     PUTVAX  /STOP INPUT TILL FILE OPENED

00403  6213          CIF CDF 10
00404  4775'         JMS     USRCALL /GET OS8 DESTINATION FILE NAME

00405  7300  RET1,    CLA CLL
00406  1336          TAD     BOTTOM
00407  7001          IAC
00410  3337          DCA     PTR
00411  1374  ENABLE, TAD     (21     / <XON>
00412  4776'         JMS     PUTVAX  /ACCEPT INPUT FROM VAX
00413  6361  NXTCHR, VKSF
00414  5274          JMP     TSTKBD  /NOTHING YET; CHECK KEYBOARD
00415  6366          VKRB
00416  0373          AND     (177    /CLEAR PARITY BIT
00417  7450          SNA
00420  5213          JMP     NXTCHR  /IF NULL,LOOK FOR ANOTHER CHAR
00421  3340          DCA     STORE2
00422  7300          CLA CLL
00423  1372          TAD     (32     /CONTROL-Z SENT AT END OF FILE?
00424  7041          CIA
00425  1340          TAD     STORE2
00426  7450          SNA
00427  5323          JMP     CLOS    /YES, CLOSE THE OS/8 FILE

00430  7300          CLA CLL
00431  1340          TAD     STORE2  /RECALL CHAR FOR ENTRY INTO LINE BUFFER
00432  3737          DCA I   PTR
00433  1337          TAD     PTR
00434  7001          IAC             /INCREMENT POINTER TO NEXT CHAR IN BUFFER
00435  3337          DCA     PTR
00436  1371          TAD     (12     /<LF>; AT END OF A LINE, INPUT WILL BE
```

```
00437  7041           CIA                  /DISABLED AND LINE BUFFER SENT TO OS/8
00440  1340           TAD      STORE2
00441  7450           SNA
00442  5244           JMP      LINESND /LAST CHAR WAS <LF>
00443  5213           JMP      NXTCHR

00444  7300  LINESND,CLA CLL
00445  1377           TAD      (23     / <XOFF>
00446  4776'          JMS      PUTVAX  /STOP INPUT
00447  7300           CLA CLL
00450  1336           TAD      BOTTOM  /SET POINTER TO START OF LINE BUFFER (LESS 1)
00451  3337           DCA      PTR

00452  7300  WRITE,   CLA CLL          /SEND BUFFER CHAR-BY-CHAR UNTIL <LF>
00453  7001           IAC              /TO OS/8 FILE
00454  1337           TAD      PTR
00455  3337           DCA      PTR
00456  1737           TAD I    PTR
00457  6213           CIF CDF 10
00460  5770'          JMP      TODSK   /1 CHAR TO FILE
00461  7300  RET4,    CLA CLL
00462  1371           TAD      (12     / <LF>
00463  7041           CIA
00464  1737           TAD I    PTR
00465  7440           SZA              /AFTER <LF> GET ANOTHER LINE
00466  5252           JMP      WRITE


00467  7300  NXTLINE,CLA CLL
00470  1336           TAD      BOTTOM  /RESET BUFFER POINTER
00471  7001           IAC
00472  3337           DCA      PTR
00473  5211           JMP      ENABLE  /READY FOR NEXT LINE

00474  6031  TSTKBD, KSF      .       /CHECK KEYBOARD
00475  5213           JMP      NXTCHR  /IF NOTHING, LOOK AT VAX AGAIN
00476  6036  KBD,    KRB
00477  0373           AND      (177    /CLEAR PARITY BIT
00500  3340           DCA      STORE2
00501  1372           TAD      (32     /CONTROL-Z: OPERATOR REQUESTS
00502  7041           CIA              /IMMEDIATE END OF FILE
00503  1340           TAD      STORE2
00504  7450           SNA
00505  5323           JMP      CLOS    /YES, CLOSE FILE AS IS
00506  7300           CLA CLL
00507  1367           TAD      (3      /CHECK IF ^C TO ABORT TRANSFER
00510  7041           CIA
00511  1340           TAD      STORE2
00512  7450           SNA
00513  5330           JMP      ABORT   /SEND CONTROL-C TO VAX
00514  7300           CLA CLL
00515  1366           TAD      (31     /CHECK IF ^Y (ALT. ABORT COMMAND)
00516  7041           CIA
00517  1340           TAD      STORE2
00520  7450           SNA
```

```
00521  5330           JMP     ABORT   /YES, STOP
00522  5211           JMP     ENABLE

00523  7300  CLOS,     CLA CLL
00524  1377           TAD     (23     /GET <XOFF> TO DISABLE INPUT
00525  4776'          JMS     PUTVAX
00526  6213           CIF CDF 10
00527  5765'          JMP     USRCL   /CLOSE FILE; RETURN VIA RET5

00530  7300  ABORT,    CLA CLL
00531  1367           TAD     (3      /CONTROL-C
00532  4776'          JMS     PUTVAX  /STOP FILE TRANSFER PERMANENTLY
00533  7300  RET5,     CLA CLL
00534  1374           TAD     (21     /ENABLE VAX TRANSMISSION
00535  5764'          JMP     VAXWAIT
00536  0777  BOTTOM,   777
00537  0000  PTR,      0
00540  0000  STORE2,   0
00564  0222
00565  6014
00566  0031
00567  0003
00570  6022
00571  0012
00572  0032
00573  0177
00574  0021
00575  6000
00576  0250
00577  0023
       0600           PAGE
```

```
                /
                /
                /       PDPCOPY: COPY FILE TO VAX
                /
                /       BRANCHES TO OS/8 COMMAND DECODER TO GET OS/8 INPUT FILE
                /       NAME FROM OPERATOR.
                /
                /       THE OS/8 CHARACTER-ORIENTED IO ROUTINES SUPPLY THE FILE
                /       AS A STREAM OF CHARACTERS WHICH ARE SENT TO THE VAX. IF
                /       THE VAX SENDS AN <XOFF>, THE PROGRAM WAITS FOR AN <XON>
                /       BEFORE CONTINUING. AN ERROR MESSAGE FROM THE VAX, WHICH
                /       IS RECOGNISED FROM THE LEADING '%', ABORTS THE TRANSFER.
                /       SINCE THE VAX INSERTS A LINE-FEED AFTER CARRIAGE-RETURN,
                /       LINE-FEEDS AREN'T SENT. NULL CHARACTERS ARE IGNORED.
                /
                /       TYPING CTRL/Z WILL FORCE IMMEDIATE END OF FILE AND RETURN TO
                /       VAXTTY.
                /       TYPING CTRL/Y ABORTS TRANSFER AND RETURNS CONTROL TO VAXTTY.
                /       (CTRL/C TERMINATES ALSO).
                /
                /
00600  7300     PDPCOPY,CLA CLL
00601  6213             CIF CDF 10
00602  4777'            JMS     UCAL2   /OPEN OS/8 INPUT FILE
00603  7300     RET10,  CLA CLL
00604  6366             VKRB            /CLEAR INPUT FLAG

00605  6371     NXT,    VTSF            /IS LINE READY FOR OUTPUT?
00606  5262             JMP     TKBD1   /NO! TEST FOR KEYBOARD CHAR
00607  6213             CIF CDF 10
00610  5776'            JMP     GDATA   /READ CHARACTER FROM OS/8 FILE
00611  0375     RET20,  AND     (177    /CLEAR PARITY BIT
00612  7450             SNA             /IF <NULL>, IGNORE
00613  5205             JMP     NXT
00614  3313             DCA     STORE3
00615  1374             TAD     (12     /IF <LF>, DON'T SEND ON
00616  7041             CIA
00617  1313             TAD     STORE3
00620  7450             SNA
00621  5205             JMP     NXT     /NEXT CHAR
00622  7300             CLA CLL
00623  1313             TAD     STORE3  /SEND ON TO VAX
00624  4773'            JMS     PUTVAX
00625  6361             VKSF            /HAS VAX SENT ANYTHING?
00626  5205             JMP     NXT     /NO

00627  6366             VKRB            /YES
00630  0375             AND     (177
00631  3313             DCA     STORE3
00632  1372             TAD     (23     /<XOFF> MEANS WAIT TILL <XON>
00633  7041             CIA
00634  1313             TAD     STORE3
00635  7450             SNA
00636  5246             JMP     IDLE    /GO AND WAIT FOR <XON>
00637  7300             CLA CLL
```

```
00640  1371           TAD     (45     /'%' PRECEDES AN ERROR MESSAGE
00641  7041           CIA
00642  1313           TAD     STORE3
00643  7450           SNA
00644  5770'          JMP     VAXWAIT /AN ERROR HAS OCCURRED
00645  5205           JMP     NXT     /IGNORE ANY OTHER DATA

00646  6361  IDLE,    VKSF            /WAIT FOR NEW INPUT FROM VAX
00647  5246           JMP     .-1
00650  6366           VKRB
00651  0375           AND     (177
00652  7041           CIA
00653  1367           TAD     (21     /CHECK FOR <XON>
00654  7440           SZA
00655  5246           JMP     IDLE    /<XON> IS ONLY WAY OUT!
00656  5205           JMP     NXT     /FREE TO CONTINUE

00657  7300  EOF,     CLA CLL
00660  1366           TAD     (32     /CONTROL-Z
00661  5770'          JMP     VAXWAIT /TELL VAX THAT TRANSFER IS FINISHED

00662  6031  TKBD1,   KSF             /ANYTHING TYPED BY OPERATOR?
00663  5205           JMP     NXT     /NO, CONTINUE TRANSFER
00664  6036           KRB
00665  0375           AND     (177    /CLEAR PARITY BIT
00666  3313           DCA     STORE3
00667  1366           TAD     (32     /^Z REQUESTS IMMEDIATE EOF
00670  7041           CIA
00671  1313           TAD     STORE3
00672  7450           SNA
00673  5257           JMP     EOF

00674  7300           CLA CLL
00675  1365           TAD     (3      /^C ABORTS TRANSFER
00676  7041           CIA
00677  1313           TAD     STORE3
00700  7450           SNA
00701  5310           JMP     ABORT1  /SEND ^C TO VAX
00702  7300           CLA CLL
00703  1364           TAD     (31     /^Y IS ALTERNATE ABORT CODE
00704  7041           CIA
00705  1313           TAD     STORE3
00706  7440           SZA             /JUMP TO ABORT1 IF ^Y
00707  5205           JMP     NXT     /IGNORE ANY OTHER INPUT

00710  7300  ABORT1,  CLA CLL
00711  1365           TAD     (3      /LOAD CONTROL-C FOR VAX
00712  5770'          JMP     VAXWAIT /TRANSFER ABORTED

               /
00713  0000  STORE3,  0
00764  0031
00765  0003
00766  0032
00767  0021
```

```
00770  0222
00771  0045
00772  0023
00773  0250
00774  0012
00775  0177
00776  6042
00777  6026
       1000          PAGE
```

```
                /
                /       USRCALL
                /       INTERFACE TO OS/8 CHARACTER IO ROUTINES
                /       (SEE OS8IO, FOLLOWING)
                /
                /
        0001            FIELD 1
        6000            *6000
16000   0000  USRCALL,0                /USR MANIPULATIONS
16001   7300            CLA CLL
16002   4653            JMS I ULOCK     /LOCK USR IN CORE
16003   0010            10              /PARAMETER
16004   4654  MONI,     JMS I MONIT     /CALL MONITOR FOR FILE NAMES ETC
16005   0005            5
16006   0000            0
16007   7300            CLA CLL
16010   4777'           JMS OOPEN       /OPEN FILE
16011   5204            JMP     MONI    /ERROR OPENING FILE; RETRY
16012   6203            CIF CDF 0       /BACK FIELD 0
16013   4600            JMS I   USRCALL /RETURN WITH FILE OPEN

16014   7300  USRCL,    CLA CLL         /GOT INSTRUCTION TO CLOSE FILE
16015   4776'           JMS OCLOSE      /CLOSE OUTPUT FILE
16016   7402            HLT             /ERROR CLOSING FILE
16017   7300            CLA CLL
16020   6203            CIF CDF 0
16021   5775'           JMP RET5        /RETURN TO VAXCOPY

16022   4774' TODSK,    JMS OCHAR       /OUTPUT A CHARACTER
16023   7402            HLT             /ERROR ON DSK WRITE
16024   6203            CIF CDF 0
16025   5773'           JMP RET4        /BACK TO VAXCOPY

16026   0000  UCAL2,    0               /INQUIRE FOR INPUT FILE
16027   7300            CLA CLL
16030   4653            JMS I   ULOCK
16031   0010            10
16032   4654  MON2,     JMS I   MONIT
16033   0005            5
16034   0000            0
16035   7300            CLA CLL
16036   4772'           JMS     IOPEN   /OPEN INPUT FILE
16037   7000            NOP
16040   6203            CIF CDF 0
16041   5626            JMP I   UCAL2

16042   7300  GDATA,    CLA CLL
16043   4771'           JMS     ICHAR   /READ ONE CHARACTER FROM OS/8
16044   5247            JMP     ENDFOR
16045   6203            CIF CDF 0
16046   5770'           JMP     RET20   /RETURN TO PDPCOPY
16047   7510  ENDFOR,   SPA
16050   7402            HLT
16051   6203            CIF CDF 0
16052   5767'           JMP     EOF
```

```
16053   7700  ULOCK,  7700
16054   0200  MONIT,  200
16167   0657
16170   0611
16171   6613
16172   6601
16173   0461
16174   7211
16175   0533
16176   7106
16177   7000
        6200          PAGE
```

```
                / OSBIO
                /THE COMPLETE INPUT AND OUTPUT ROUTINES AS USED IN THE OS/8
                /SYSTEM WHICH WILL INPUT AND OUTPUT SINGLE ASCII CHARACTERS
                /(OR BINARY) TO AND FROM FILES SPECIFIED AT RUN-TIME.
                /COPIED FROM DEC-S8OSSMA-A-D  (OS/8 SOFTWARE SUPPORT MANUAL)
                /26-JUL-73
                /SEE APPENDIX F FOR FULL DESCRIPTION

                /PARAMETER DEFINITION:

        4000    OUBUF=4000                      /OUTPUT BUFFER STARTS AT 04000
        4200    OUCTL=4200                      /AND IS 2 PAGES LONG
        1400    OUDEVH=1400                     /OUTPUT HANDLER IS LOADED AT 01400. WE
                                                /ALLOW 2 PAGE HANDLERS.
        4400    INBUF=4400                      /INPUT BUFFER STARTS AT 04400
        0200    INCTL=0200                      /    TWO PAGES LONG.
        0001    INRECS=1                        /2 PAGES = 1 RECORD
        1000    INDEVH=1000                     /ALLOW 2 PAGE INPUT HANDLER AT 01000
        6600    ORIGIN=6600                     /THESE SUBROUTINES RESIDE AT 16600
        7760    DCR=7760                        /DEVICE CONTROL TABLE

        0001            FIELD 1
        6600            *ORIGIN
        0000    INFLD=INCTL&70                  /INPUT BUFFER FIELD
        0000    OUFLD=OUCTL&70                  /OUTPUT BUFFER FIELD

16600   7400    IN7400, 7400
16601   0000    IOPEN,  0                       /INITIALISE INPUT
16602   7240            CLA CMA
16603   3333            DCA INCHCT              /SET TO READ FROM NEW DEVICE
16604   2210            ISZ INEOF               /FORCE A NEW INPUT FILE
16605   1377            TAD (7617               /POINT TO CD INPUT LIST
16606   3211            DCA INPTR
16607   5601            JMP I IOPEN
16610   0000    INEOF,  0
16611   0000    INPTR,  0
16612   0000    INPTR,  0

16613   0000    ICHAR,  0                       /INPUT A CHARACTER
16614   7600    IN7600, 7600
16615   6214            RDF                     /SAVE CALLING FIELD FOR RETURN
16616   1245            TAD INCDIF
16617   3331            DCA INRTRN
16620   6201    INCHAR, CDF INFLD               /DATA FIELD TO FIELD OF BUFFER
16621   2300            ISZ INJMP               /3-WAY UNPACKING SWITCH
16622   2333            ISZ INCHCT              /INPUT BUFFER EXHAUSTED?
16623   5300    INJMPP, JMP INJMP               /NO..UNPACK THE NEXT CHARACTER
16624   1210            TAD INEOF               /DID LAST READ GIVE EOF ON THIS DEVICE?
16625   7650            SNA CLA
16626   5231            JMP INGBUF              /NO..CONTINUE READING
16627   4333    GETNEW, JMS INNEWF              /YES..GET NEXT INPUT FILE IF IT EXISTS
16630   5277            JMP EOFERR              /TAKE EOF EXIT FROM ICHAR
16631   1201    INGBUF, TAD INCTR               /INCTR HOLDS THE CURRENT LENGTH OF
                                                /THE INPUT FILE. WHEN THE AMOUNT REMAINING
                                                /TO READ IS LESS THAN THE SIZE OF THE
```

```
                                        /INPUT BUFFER, AN EOF IS SIGNALLED.
16632  7100          CLL
16633  1376          TAD     (INRECS
16634  7420          SNL
16635  3201          DCA INCTR             /UPDATE REMAINING LENGTH
16636  7430          SZL
16637  2210          ISZ INEOF             /AND SIGNAL EOF FOR NEXT READ
16640  7172          CLL CML CMA RTR       /CONSTRUCT A CONTROL WORD FOR THIS
16641  7012          RTR
16642  7012          RTR
16643  1375          TAD (INCTL+1
16644  3250          DCA INCTLW

16645  6203  INCDIF, CIF CDF 0             /NOW DO A CALL TO THE INPUT HANDLER
16646  6211          CDF 10                /WE ARE IN FIELD 1, HANDLER IN FIELD 0
16647  4744          JMS I INHNDL
16650  0000  INCTLW, 0                     /INPUT CONTROL WORD
16651  4400  INBUFP, INBUF                /INPUT BUFFER ADDRESS
16652  0000  INREC,  0                     /POINTER TO INPUT RECORD
16653  5273          JMP INERRX
16654  1252  INBREC, TAD INREC
16655  1376          TAD (INRECS           /UPDATE POINTER INTO FILE
16656  3252          DCA INREC
16657  1250          TAD INCTLW            /NOW COMPUTE THE NUMBER OF CHARACTERS
16660  0214          AND IN7600            /IN THIS INPUT BUFFER
16661  7104          CLL RAL
16662  1250          TAD INCTLW
16663  0214          AND IN7600
16664  7040          CMA
16665  3333          DCA INCHCT            /NEW NUMBER OF CHARACTERS
16666  1223          TAD INJMPP            /RESET 3-WAY SWITCH
16667  3300          DCA INJMP
16670  1251          TAD INBUFP
16671  3212          DCA INPTR             /AND BUFFER POINTER
16672  5220          JMP INCHAR            /NOW READ THE BUFFER
16673  2210  INERRX, ISZ INEOF            /SET EOF JUST IN CASE
16674  7700          SMA CLA               /IF <0, A PHYSICAL ERROR
16675  5254          JMP INBREC            /EOF ON INPUT
16676  7330  INERR,  CLA CLL CML RAR       /FATAL
16677  5331  EOFERR, JMP INRTRN            /GET OUT
16700  7402  INJMP,  HLT                   /3-WAY UNPACK SWITCH
16701  5322          JMP ICHAR1            /GET FIRST OF 3
16702  5316          JMP ICHAR2            /SECOND
16703  1223  ICHAR3, TAD INJMPP
16704  3300          DCA INJMP             /SET FOR FIRST CHAR. NEXT
16705  1612          TAD I INPTR           /THE THIRD WORD IS MADE OF THE HIGH
16706  0200  IN200,  AND IN7400            /ORDER BITS OF THE FIRST
16707  7112          CLL RTR               /TWO.
16710  7012          RTR
16711  1250          TAD INCTLW
16712  7012          RTR
16713  7012          RTR
16714  2212          ISZ INPTR             /POINT TO NEXT WORD
16715  5323          JMP INCOMN            /GET OUT WITH CHARACTER IN AC
16716  1612  ICHAR2, TAD I INPTR
```

```
16717  0200           AND IN7400              /SAVE HIGH ORDER FOR THIRD WORD
16720  3250           DCA INCTLW
16721  2212           ISZ INPTR
16722  1612  ICHAR1,  TAD I INPTR
16723  0374  INCOMN,  AND (377
16724  1373           TAD (-232               /IS IT ^Z (EOF)?
16725  7450           SNA
16726  5227           JMP GETNEW              /YES..LOOK AT NEXT INPUT
16727  1372           TAD (232
16730  2213           ISZ ICHAR               /TAKE NORMAL RETURN
16731  0000  INRTRN,  0                       /CIF CDF N
16732  5613           JMP I ICHAR

16733  7777  INNEWF,  -1
       6733  INCHCT=INNEWF
16734  6211           CDF 10
16735  1371           TAD (INDEVH+1           /INITIALISE IN CASE WE NEED A NEW
16736  3344           DCA INHNDL              /MORE INPUT?
16737  1611           TAD I INFPTR
16740  7450           SNA
16741  5733           JMP I INNEWF            /NOPE
16742  4706           JMS I IN200             /CALL MONITOR TO GET HANDLER
16743  0001           1
16744  0000  INHNDL,  0
16745  7402           HLT                     /VERY BAD!
16746  1611           TAD I INFPTR
16747  0370           AND (7760               /GET INPUT FILE LENGTH
16750  7440           SZA
16751  1367           TAD (17
16752  7132           CLL CML RTR
16753  7012           RTR                     /NEGATIVE OF FILE LENGTH
16754  3201           DCA INCTR
16755  2211           ISZ INFPTR              /POINT TO STARTING BLOCK
16756  1611           TAD I INFPTR
16757  3252           DCA INREC               /STORE IN HANDLER CALL
16760  2211           ISZ INFPTR              /NEXT INPUT
16761  3210           DCA INEOF               /CLEAR EOF FLAG
16762  2333           ISZ INNEWF
16763  5733           JMP I INNEWF
       6601           INCTR=IOPEN
16767  0017
16770  7760
16771  1001
16772  0232
16773  7546
16774  0377
16775  0201
16776  0001
16777  7617
       7000           PAGE
```

```
17000  0000  OOPEN,  0                    /OPEN OUTPUT FILE
17001  7600  OU7600, 7600
17002  1347         TAD OU7601            /POINT TO OUTPUT FILE NAME IN CD AREA
17003  3221         DCA OUBLK
17004  1377         TAD (OUDEVH+1
17005  3214         DCA OUHNDL            /INITIALISE OUTPUT DEVICE HANDLER
17006  1601         TAD I OU7600          /PICKUP OUTPUT DEVICE NUMBER
17007  0376         AND (17
17010  7450         SNA                   /IS THERE ONE?
17011  5244         JMP ONOFIL            /NO..INHIBIT OUTPUT
17012  4775         JMS I (200            /FETCH OUTPUT HANDLER
17013  0001         1
17014  0000  OUHNDL, 0
17015  7402         HLT                   /BAD THING
17016  1601  OUENTR, TAD I OU7600
17017  4775         JMS I (200            /ENTER THE OUTPUT FILE
17020  0003         3
17021  7601  OUBLK,  7601                  /GETS STARTING BLOCK OF HOLE

17022  0000  OUELEN, 0                     /GETS SIZE OF HOLE AVAILABLE
17023  5232         JMP OEFAIL            /FAILURE. SEE WHAT WE DID.
17024  3350         DCA OUCCNT            /CLEAR CLOSING LENGTH
17025  3774         DCA I (OUTINH         /CLEAR OUTPUT INHIBIT
17026  4773         JMS I (OUSETP         /SET UP POINTERS
17027  2200         ISZ OOPEN
17030  6213  OORETN, CDF CIF 10
17031  5600         JMP I OOPEN           /RETURN O.K.
17032  1601  OEFAIL, TAD I OU7600          /IF LENGTH = 0, GIVE OPEN ERROR
17033  0372         AND (7760             /IF NOT, MAKE IT 0 AND TRY AGAIN
17034  7650         SNA CLA
17035  5242         JMP ONTERR            /WAS 0. FAILED.
17036  1601         TAD I OU7600
17037  0376         AND (17               /MAKE IT 0
17040  3601         DCA I OU7600
17041  5216         JMP OUENTR            /AND TRY AGAIN
17042  7330  ONTERR, CLA CLL CML RAR
17043  5230         JMP OORETN
17044  2774  ONOFIL, ISZ I (OUTINH        /INHIBIT OUTPUT
17045  5230         JMP OORETN
17046  0000  OUTDMP, 0                    /DUMP OUTPUT BUFFER
17047  3300         DCA OUCTLW            /STORE CONTROL WORD
17050  6211         CDF 10
17051  1774         TAD I (OUTINH         /IS OUTPUT INHIBITED?
17052  7640         SZA CLA
17053  5304         JMP OUNOWR            /YEP
17054  1350         TAD OUCCNT            /IF THIS IS FIRST WRITE, START THE
17055  7450         SNA                   /SEARCH FORWARD ON DECTAPE
17056  2300         ISZ OUCTLW
17057  1221         TAD OUBLK             /GET STARTING BLOCK OF THIS
17060  3302         DCA OUREC             /TRANSFER
```

```
17061  1300           TAD OUCTLW
17062  7106           CLL RTL
17063  7006           RTL
17064  7006           RTL                    /COMPUTE # OF RECORDS TO OUTPUT
17065  0376           AND (17
17066  1350           TAD OUCCNT             /UPDATE CLOSING LENGTH
17067  3350           DCA OUCCNT
17070  1350           TAD OUCCNT             /SEE IF CLOSING LENGTH WILL BE
17071  7120           CLL CML                /BIGGER THAN OUTPUT HOLE
17072  1222           TAD OUELEN
17073  7660           SNL SZA CLA
17074  5646           JMP I OUTDMP           /WILL BE TOO BIG
17075  6203 OUCDIF,   CIF CDF 0
17076  6211           CDF 10
17077  4614           JMS I OUHNDL           /DO THE WRITE
17100  0000 OUCTLW,   0
17101  4000           OUBUF
17102  0000 OUREC,    0
17103  7410           SKP                    /ERROR
17104  2246 OUNOWR,   ISZ OUTDMP             /TAKE NORMAL RETURN
17105  5646           JMP I OUTDMP
       0020           PTP=0020

17106  0000 OCLOSE,   0                      /CLOSE OUTPUT FILE
17107  6211           CDF 10
17110  1774           TAD I (OUTINH          /IF OUTPUT INHIBITED, CLOSE IS A NOP
17111  7640           SZA CLA
17112  5352           JMP OCISZ              /A NOP
17113  4771           JMS I (OTYPE           /DETERMINE IF OUTPUT IS TO PTP
17114  0370           AND (770               /IF IT IS, DON'T OUTPUT A ^Z
17115  1372           TAD (-PTP
17116  7640           SZA CLA
17117  1367           TAD (232               /NOT PTP, OUTPUT ^Z AS EOF
17120  4766           JMS I (OCHAR
17121  5353           JMP OCRET              /ERROR RETURN
17122  4766           JMS I (OCHAR           /FILL WITH 0 CHARACTERS
17123  5353           JMP OCRET
17124  4766 FILLIP,   JMS I (OCHAR           /FILL TO BOUNDARY WITH 0
17125  5353           JMP OCRET
17126  4771           JMS I (OTYPE           /IF OUTPUT IS DIRECTORY DEVICE, FILL
17127  7710           SPA CLA                /WHOLE RECORD, ELSE HALF RECORD
17130  1365           TAD (100
17131  1364           TAD (77
17132  0763           AND I (OUDWCT          /ARE WE UP TO BOUNDARY YET?
17133  7640           SZA CLA
17134  5324           JMP FILLIP             /NO
17135  1763           TAD I (OUDWCT
17136  1375           TAD (OUCTL&3700        /IS THERE A FULL WRITE LEFT?
17137  7450           SNA
17140  5344           JMP NODUMP             /YES, BUT DON'T DO IT, AS ^Z IS OUT
17141  1362           TAD (4000+OUFLD
17142  4246           JMS OUTDMP             /DUMP LAST BUFFER
17143  5353           JMP OCRET
17144  1601 NODUMP,   TAD I OU7600           /GET DEVICE NUMBER
```

```
17145  4775           JMS I (200              /CLOSE THE OUTPUT FILE
17146  0004           4
17147  7601  OU7601,  7601                    /POINTER TO FILE NAME
17150  0000  OUCCNT,  0                       /CLOSING FILE LENGTH HERE
17151  7410           SKP                     /ERROR
17152  2306  OCISZ,   ISZ OCLOSE             /NORMAL RETURN
17153  6213  OCRET,   CDF CIF 10             /RESTORE CALLING FIELDS
17154  5706           JMP I OCLOSE
17162  4000
17163  7272
17164  0077
17165  0100
17166  7211
17167  0232
17170  0770
17171  7274
17172  7760
17173  7200
17174  7273
17175  0200
17176  0017
17177  1401
       7200           PAGE
```

```
17200  0000  OUSETP, 0                      /INITIALISE OUTPUT POINTERS
17201  1377         TAD (OUCTL&3700
17202  7041         CIA
17203  3272         DCA OUDWCT              /DOUBLE WORD OUTPUT COUNT
17204  1376         TAD (OUBUF              /INITIALISE WORD POINTER
17205  3270         DCA OUPTR
17206  1271         TAD OUJMPE
17207  3224         DCA OUJMP               /3-WAY UNPACK SWITCH
17210  5600         JMP I OUSETP

17211  0000  OCHAR, 0                       /OUTPUT CHARACTER ROUTINE
17212  0375         AND (377                /ISOLATE EIGHT BITS
17213  3266         DCA OUTEMP
17214  6214         RDF                     /GET FIELD WE WERE CALLED
17215  1374         TAD (CIF CDF 0          /FROM
17216  3264         DCA OUCRET
17217  1273         TAD OUTINH              /OUTPUT INHIBITED?
17220  7640         SZA CLA
17221  5263         JMP OUCOMN              /YES, NOP.
17222  6201  OUCHAR, CDF OUFLD              /GOTO DATA FIELD OF BUFFER
17223  2224         ISZ OUJMP               /BUMP CHARACTER SWITCH
17224  7402  OUJMP,  HLT                    /GETS JMP.,JMP+1, ETC
17225  5261         JMP OCHAR1
17226  5256         JMP OCHAR2
17227  1266  OCHAR3, TAD OUTEMP              /THIRD CHARACTER
17230  7106         CLL RTL                 /HIGH ORDER BITS GO INTO THE
17231  7006         RTL                     /HIGH ORDER 4 BITS OF THE
17232  0373         AND (7400               /FIRST OF TWO WORDS
17233  1667         TAD I OUPOLD
17234  3667         DCA I OUPOLD
17235  1266         TAD OUTEMP              /THE SECOND DOUBLE WORD GETS
```

```
17236  7112            CLL RTR
17237  7012            RTR
17240  7010            RAR
17241  0373            AND (7400
17242  1670            TAD I OUPTR
17243  3670            DCA I OUPTR
17244  1271            TAD OUJMPE
17245  3224            DCA OUJMP
17246  2270            ISZ OUPTR
17247  2272            ISZ OUDWCT

17250  5263            JMP OUCOMN
17251  1372            TAD (OUCTL
17252  4771            JMS I (OUTDMP
17253  5264            JMP OUCRET
17254  4200            JMS OUSETP
17255  5263            JMP OUCOMN
17256  1270  OCHAR2,   TAD OUPTR
17257  3267            DCA OUPOLD
17260  2270            ISZ OUPTR
17261  1266  OCHAR1,   TAD OUTEMP
17262  3670            DCA I OUPTR
17263  2211  OUCOMN,   ISZ OCHAR
17264  7402  OUCRET,   HLT
17265  5611            JMP I OCHAR
17266  0000  OUTEMP,   0
17267  0000  OUPOLD,   0
17270  0000  OUPTR,    0
17271  5224  OUJMPE,   JMP OUJMP
17272  0000  OUDWCT,   0
17273  0000  OUTINH,   0
17274  0000  OTYPE,    0
17275  6214            RDF
17276  1374            TAD (CIF CDF 0
17277  3306            DCA OTRTN
17300  6211            CDF 10
17301  1770            TAD I (7600
17302  0367            AND (17
17303  1366            TAD (DCB-1
17304  3266            DCA OUTEMP
17305  1666            TAD I OUTEMP
17306  7402  OTRTN,    HLT
17307  5674            JMP I OTYPE
17366  7757
17367  0017
17370  7600
17371  7046
17372  4200
17373  7400
17374  6203
17375  0377
17376  4000
17377  0200
       7400
```

```
   /THE LOW ORDER BITS OF
   /THE THIRD CHARACTER



   /RESET CHARACTER SWITCH

   /POINT TO NEXT BUFFER WORD
   /BUMP DOUBLE COUNT AFTER
   /3 CHARS.
   /GET OUT
   /READY TO OUTPUT A BUFFER
   /OUTPUT IT
   /AN ERROR
   /RESET OUTPUT POINTERS

   /POINT TO FIRST DOUBLE WORD

   /POINT OUPTR TO SECOND


   /NORMAL EXIT









                   /OTYPE LOOKS AT THE OUTPUT DEVICE #



                   /AND LOOKS UP THE DCB WORD FOR
                   /THAT DEVICE


                   /GET DCB ENTRY
```

```
                /THIS IS A TEST PROGRAM WHICH TRANSFERS ONE FILE TO ANOTHER
                /
      0001              FIELD 1
      2000              *2000
12000 4777              JMS I (7700             /LOCK MONITOR INTO CORE
12001 0010              10
12002 4776  CALLCD,     JMS I (200              /CALL THE COMMAND DECODER
12003 0005              5
12004 0000              0
12005 4775              JMS I (IOPEN            /SETUP INPUT POINTERS
12006 4774              JMS I (OOPEN            /PEN OUTPUT FILE
12007 7700              SMA CLA                 /ERROR, IF AC<0, IT WAS FATAL,
12010 5220              JMP OK                  /NON FILE-STRUCTURED OUTPUT
12011 4263              JMS TERR
12012 1720              TEXT /OPEN FAILED/
12013 0516
12014 4006
12015 0111
12016 1405
12017 0400
12020 4773  OK,         JMS I (ICHAR            /READ A CHARACTER
12021 5227              JMP TSTEOF              /ERROR, SEE IF EOF
12022 7450              SNA                     /IGNORE BLANKS
12023 5220              JMP OK
12024 4772              JMS I (OCHAR            /AND OUTPUT THE CHARACTER
12025 5243              JMP OUTERR
12026 5220              JMP OK                  /CONTINUE TILL EOF
12027 7700  TSTEOF,     SMA CLA                 /WAS IT FATAL?
12030 5240              JMP CLOSE               /NO..EOF.. CLOSE OUTPUT
12031 4263              JMS TERR
12032 2205              TEXT /READ ERROR/
12033 0104
12034 4005
12035 2222
12036 1722
12037 0000
12040 4771  CLOSE,      JMS I (OCLOSE
12041 5253              JMP CLERR               /FILE CLOSE FAILED
12042 5202              JMP CALLCD              /NEW INPUT
12043 4263  OUTERR,     JMS TERR
12044 1725              TEXT /OUTPUT ERROR/
12045 2420
12046 2524
12047 4005
12050 2222
12051 1722
12052 0000
12053 4263  CLERR,      JMS TERR                /CLOSE FAILURE
12054 0314              TEXT /CLOSE FAILED/
12055 1723
12056 0540
12057 0601
12060 1114
12061 0504
12062 0000
```

```
12063  0000  TERR,   0
12064  1663         TAD I TERR
12065  7012         RTR!RTR!RTR
12066  7012
12067  7012

12070  4275         JMS TYPIT          /TYPE THE CHARACTER
12071  1663         TAD I TERR
12072  4275         JMS TYPIT
12073  2263         ISZ TERR
12074  5264         JMP TERR+1

12075  0000  TYPIT,  0
12076  0370         AND (77            /ISOLATE THE CHARACTER
12077  7450         SNA
12100  5304         JMP CRLF           /0 TERMINATES IT
12101  1367         TAD (300
12102  4311         JMS TTYOUT
12103  5675         JMP I TYPIT
12104  1366  CRLF,  TAD (215
12105  4311         JMS TTYOUT
12106  1365         TAD (212
12107  4311         JMS TTYOUT
12110  5202         JMP CALLCD

12111  0000  TTYOUT, 0
12112  6046         TLS
12113  6041         TSF
12114  5313         JMP .-1
12115  7200         CLA
12116  5711         JMP I TTYOUT
12165  0212
12166  0215
12167  0300
12170  0077
12171  7106
12172  7211
12173  6613
12174  7000
12175  6601
12176  0200
12177  7700
       $
```

```
ABORT  0530
ABORT1 0710
BOTTOM 0536
CALLCD 2002
CLERR  2053
CLOS   0523
CLOSE  2040
CONT   0234
CONT2  0242
CRLF   2104
DCB    7760
ENABLE 0411
ENDFOR 6047
EOF    0657
EOFERR 6677
FILLIP 7124
GDATA  6042
GETNEW 6627
ICHAR  6613
ICHAR1 6722
ICHAR2 6716
ICHAR3 6703
IDLE   0646
INBREC 6654
INBUF  4400
INBUFP 6651
INCDIF 6645
INCHAR 6620
INCHCT 6733
INCOMN 6723
INCTL  0200
INCTLW 6650
INCTR  6601
INDEVH 1000
INEOF  6610
INERR  6676
INERRX 6673
INFLD  0000
INFPTR 6611
INGBUF 6631
INHNDL 6744
INJMP  6700
INJMPP 6623
INNEWF 6733
INPTR  6612
INREC  6652
INRECS 0001
INRTRN 6731
IN200  6706
IN7400 6600
IN7600 6614
IOPEN  6601
KBD    0476
KEYRD  0206
LINESN 0444
MONI   6004
MONIT  6054
MON2   6032
NODUMP 7144
NXT    0605
NXTCHR 0413
NXTLIN 0467
OCHAR  7211
OCHAR1 7261
OCHAR2 7256
OCHAR3 7227
OCISZ  7152
OCLOSE 7106
OCRET  7153
OEFAIL 7032
OK     2020
ONOFIL 7044
ONTERR 7042
OOPEN  7000
OORETN 7030
ORIGIN 6600
OSEXIT 0256
OTRTN  7306
OTYPE  7274
OUBLK  7021
OUBUF  4000
OUCCNT 7150
OUCDIF 7075
OUCHAR 7222
OUCOMN 7263
OUCRET 7264
OUCTL  4200
OUCTLW 7100
OUDEVH 1400
OUDWCT 7272
OUELEN 7022
OUENTR 7016
OUFLD  0000
OUHNDL 7014
OUJMP  7224
OUJMPE 7271
OUNOWR 7104
OUFOLD 7267
OUPTR  7270
OUREC  7102
OUSETP 7200
OUTDMP 7046
OUTEMP 7266
OUTERR 2043
OUTINH 7273
OU7600 7001
OU7601 7147
PDPCOP 0600
PTP    0020
PTR    0537
PUTVAX 0250
RET1   0405
RET10  0603
RET20  0611
RET4   0461
RET5   0533
STORE  0255
STORE2 0540
STORE3 0713
TERR   2063
TKBD1  0662
TODSK  6022
TSTEOF 2027
TSTKBD 0474
TTYOUT 2111
TTYWAI 0245
TYPIT  2075
UCAL2  6026
ULOCK  6053
USRCAL 6000
USRCL  6014
VAXCOP 0400
VAXRD  0223
VAXTTY 0200
VAXWAI 0222
VKRB   6366
VKSF   6361
VTLS   6376
VTSF   6371
WRITE  0452
```

```
ERRORS DETECTED: 0
LINKS GENERATED: 26
```