



DECUS

PROGRAM LIBRARY

DECUS NO.	12-136
TITLE	MOVE
AUTHOR	Larry Davis, Carl Ralston Washington University, St. Louis, Missouri
COMPANY	Submitted by: Robert Hassinger Liberty Mutual Research Center Hopkinton, Massachusetts
DATE	October 9, 1972
SOURCE LANGUAGE	PAL-8

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

MOVE

DECUS Program Library Write-up

DECUS NO. 8-634

ABSTRACT

MOVE IS AN OS/8 PROGRAM FOR TRANSFERRING FILES FROM ONE DIRECTORY DEVICE TO ANOTHER DIRECTORY DEVICE. IT IS EFFICIENT SINCE IT READS THE INPUT AND OUTPUT DEVICE DIRECTORIES ONLY ONCE.

USE

MOVE IS CALLED FROM THE KEYBOARD MONITOR LEVEL AS FOLLOWS.

. R MOVE <DEVO>:<DEVI>:<FILE1. E1>, <FILE2. E2>, <FILE3. E3>, ...

THIS WILL CAUSE FILES <FILE1. E1>, <FILE2. E2>, <FILE3. E3>, ... TO BE MOVED FROM DIRECTORY DEVICE <DEVI> TO DIRECTORY DEVICE <DEVO>.

IF ALL OF THE FILES ARE TRANSFERRED CORRECTLY, THE PROGRAM TYPES "DONE" ON THE TELETYPE.

NOTE: BOTH THE INPUT AND OUTPUT DEVICE NAMES MUST BE EXPLICITLY SPECIFIED.

MOVE ERRORS

**** *****

MESSAGE

ERROR AND ACTION TAKEN

<DEV> NOT AVAILABLE

INPUT OR OUTPUT DEVICE NAME WHICH WAS SPECIFIED IS NOT AN OS/8 DEVICE NAME. THE PROGRAM RETURNS TO THE OS/8 KEYBOARD MONITOR.

<DEV> DIR SEG

INPUT OR OUTPUT DIRECTORY WAS MORE THAN 6 SEGMENTS LONG. PROGRAM RETURNS TO THE OS/8 KEYBOARD MONITOR. (OS/8 RESTRICTS THE NUMBER OF DIRECTORY SEGMENTS TO 6. IF THIS ERROR OCCURS, YOUR DIRECTORY IS PROBABLY BAD OR NON-EXISTENT.)

<FILE> NOT FOUND

FILE SPECIFIED WAS NOT FOUND ON INPUT DEVICE. CONTINUE WITH NEXT FILE.

<FILE> TOO BIG

FILE SPECIFIED WAS TOO BIG TO FIT ON THE OUTPUT DEVICE. CONTINUE WITH NEXT FILE.

DIR OVERFLOW

OUTPUT DIRECTORY NEEDED TO BE SPLIT, BUT AFTER THE SPLIT, THE DIRECTORY WAS MORE THAN SIX SEGMENTS. PROGRAM RETURNS TO THE OS/8 KEYBOARD MONITOR WITH NOTHING MOVED.

NO!!

A ^C WAS TYPED DURING MOVE. NO INTERRUPTIONS ARE ALLOWED AFTER THE PROGRAM STARTS MOVING THE FIRST FILE.

NOTE: THE PROGRAM RETURNS TO THE KEYBOARD MONITOR WITHOUT PRINTING AN ERROR MESSAGE IN FOUR CASES:

- 1) NOTHING WAS SPECIFIED, I. E.
. R MOVE
- 2) ONLY AN OUTPUT DEVICE WAS SPECIFIED, I. E.
. R MOVE <DEV0>:
- 3) NO FILES WERE SPECIFIED, I. E.
. R MOVE <DEV0>:<DEV1>:
- 4) INPUT AND OUTPUT DEVICES WERE THE SAME, I. E.
. R MOVE <DEV>:<DEV>:<FILE1>...

DESCRIPTION

MOVE USES THE OS/8 GETNAME ROUTINE WHICH IS STILL IN CORE AFTER MOVE IS LOADED. IT CALLS GETNAME TO GET THE OUTPUT DEVICE NAME. THEN IT CALLS THE USER SERVICE ROUTINE TO LOAD THE HANDLER FOR THIS OUTPUT DEVICE. NEXT, THE PROGRAM CALLS THE GETNAME ROUTINE TO GET THE INPUT DEVICE NAME THEN IT CALLS THE USER SERVICE ROUTINE TO LOAD THE HANDLER FOR THE INPUT DEVICE. USING THE HANDLERS, THE INPUT DIRECTORY AND THE OUTPUT DIRECTORY ARE READ INTO CORE IN THEIR TOTALITY. NOW THE PROGRAM GOES INTO A LOOP. GETNAME IS CALLED TO GET THE NEXT FILE NAME. A LOOKUP ROUTINE IS CALLED TO FIND THE STARTING BLOCK NUMBER AND LENGTH OF THIS FILE, AND THIS INFORMATION IS PLACED IN A MOVE TABLE. THEN AN ENTER ROUTINE IS CALLED TO ENTER THIS FILE NAME IN THE OUTPUT DIRECTORY, AND PLACE THE OUTPUT BLOCK NUMBER IN THE MOVE TABLE. IF NECESSARY, THE ENTER ROUTINE WILL SPLIT THE OUTPUT DIRECTORY SEGMENTS. THIS LOOP CONTINUES UNTIL ALL OF THE FILE NAMES SPECIFIED HAVE BEEN LOOKED UP AND ENTERED. THEN, THE ACTUAL MOVE TAKES PLACE FROM THE OUTPUT DEVICE TO THE INPUT DEVICE. THE FILES ARE MOVED IN THE ORDER SPECIFIED, SO TO OPTIMIZE INPUT FROM A TAPE, TYPE THE FILE NAMES IN THE ORDER THAT THE FILES APPEAR ON THE INPUT DEVICE. AFTER ALL OF THE FILES HAVE BEEN TRANSFERRED, THE MODIFIED OUTPUT DIRECTORY IS WRITTEN TO THE OUTPUT DEVICE.

CHAINING TO MOVE

IF THERE IS A PARTICULAR SEQUENCE OF FILES WHICH NEED TO BE MOVED FROM ONE DEVICE TO ANOTHER, AND THIS OPERATION IS TO BE PERFORMED SEVERAL TIMES, IT IS POSSIBLE TO WRITE A PROGRAM WHICH CHAINS TO MOVE. IN CHAINING TO MOVE, THE CHARACTERS WHICH WOULD BE NORMALLY TYPED TO SPECIFY DEVICE AND FILE NAMES SHOULD BE PLACED STARTING AT LOCATION 01000 AND LOCATION 00013 SHOULD BE SET TO 777. NOTE THAT TWO ZEROS MUST BE AT THE END OF THE LIST OF FILE NAMES, AND THAT NO MORE THAN 42 FILES MAY BE SPECIFIED. ALSO, THE CHARACTERS STARTING AT 01000 SHOULD NOT EXTEND PAST 01177 (INCLUDING THE TWO ZEROS!).

EXAMPLE:

SUPPOSE YOU ARE USING AN OS/8 DISK SYSTEM AND HAVE AN OS/8 DECTAPE OR LINCTAPE WITH FILES ON THE TAPE THAT YOU WOULD LIKE TO MOVE TO THE DISK AS A GROUP. BY SAVING A PROGRAM, SAY BOOT, THE FILES CAN BE MOVED BY TYPING

.R BOOT

ATTACHED IS A LISTING OF A PROGRAM WHICH WILL COPY THE FILES PIP.SV, EDIT.SV, MOVE.SV, AND INDEX.SV FROM DEVICE DSK: TO DEVICE DTA0:. IT IS ASSEMBLED AS

.R PALS

*STORESTORE(9L)=2000\$

SA SYS:STORE

IT IS RUN BY TYPING:

.R STORE

/
 /CHAIN TO MOVE. SV TO MOVE SPECIFIED FILES
 /

```

                *2000
                LXR=13
2000  7200  START,  CLA
2001  1377          TAD      (777
2002  3013          DCA      LXR      /POINT TO TEXT-1
2003  1376          TAD      (NAME
2004  3212          DCA      BLKN     /SAVE POINTER TO FILE NAME.
  
```

/
 /LOOK UP STARTING BLOCK NUMBER OF MOVE.
 /

```

2005  7201          CLA IAC      /DEVICE 1
2006  6201          CDF      0
2007  6212          CIF      10
2010  4775          JMS I      (7700 /CALL USR
2011  0002          2          /LOOKUP
2012  0000  BLKN,  0          /POINTER TO FILE NAME
                               /REPLACED BY BLOCK NUMBER.
2013  0000          0
2014  7402          HLT          /ERROR RETURN.
2015  7200          CLA
2016  1212          TAD      BLKN
2017  3223          DCA      BLOCK
2020  6212          CIF      10
2021  4775          JMS I      (7700
2022  0006          6          /CHAIN
2023  0000  BLOCK,  0          /STARTING BLOCK OF MOVE. SV
2024  7402          HLT          /SHOULD NOT GET HERE.
2025  1517  NAME,  FILENAME  MOVE. SV
2026  2605
2027  0000
2030  2326
  
```

2175 7700
 2176 2025
 2177 0777

*1000
 /NAMES OF FILES TO MOVE.

1000	0304	"D; "T; "A; "0; " ; "D; "S; "K; " ; "P; "I; "P; " ; "S; "V;
1001	0324	
1002	0301	
1003	0260	
1004	0272	
1005	0304	
1006	0323	
1007	0313	
1010	0272	
1011	0320	
1012	0311	
1013	0320	
1014	0256	
1015	0323	
1016	0326	
1017	0254	" ; "E; "D; "I; "T; " ; "S; "V; " ; "M; "O; "W; "E; " ; "S; "V;
1020	0305	
1021	0304	
1022	0311	
1023	0324	
1024	0256	
1025	0323	
1026	0326	
1027	0254	
1030	0315	
1031	0317	
1032	0326	
1033	0305	
1034	0256	
1035	0323	
1036	0326	
1037	0254	" ; "I; "N; "D; "E; "X; " ; "S; "V;
1040	0311	
1041	0316	
1042	0304	
1043	0305	
1044	0330	
1045	0256	
1046	0323	
1047	0326	
1050	0000	0
1051	0000	0
		\$ \$ \$

Source changes required to make MOVE
work with PS/8:

X3 = 13

X1 = 15

T8 = 56

X7605 = 104

XMER3 = 133

PRMSG = 154

PRNAME = 155

CRLF = 160