

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53

REPEAT 0,< DSKRD - TRAVEL AT YOUR OWN RISK!

TERMINOLOGY:

IN THE FOLLOWING MONOLOG, ALL REFERENCES TO E REFER TO THE ADDRESS  
FIELD OF THE UO; AC TO THE ACCUMULATOR FIELD; AC <N>  
TO THAT BIT IN THE INSTRUCTION (I.E. AC 10 MEANS BIT 9, TO SET IT, TYPE  
UO 10,E); AND AC&7 REFER TO THE LOW ORDER 3 BITS OF THE AC FIELD.

UO DESCRIPTIONS:

STR SELECTION UO:

P FUTURE IO (UNTIL ANOTHER P UO) WILL BE DONE FROM THE DEVICE  
WHOSE NAME IS AT BLKTAB+E. OPENS ARE DONE AS RARELY AS  
POSSIBLE, SO THAT LOOKUP/ENTER STATUS IS RETAINED.

AC FEATURES:

AC 10: DON'T PRINT DEVICE NAME  
AC 4: PRINT DEVICE NAME IFF AN OPEN WAS DONE ON IT.

FILE SELECTION:

L DO A LOOKUP, ENTER, OR RENAME USING E AS THE ADDRESS OF THE  
E FILE BLOCK, PRINT AN ERROR MSG IF IT FAILS, RESTORE THE OWNERS  
PPN TO THE FILE BLOCK, AND PUT THE SIZE AT LOC E+4.

AC FEATURES:

AC 10: DON'T PRINT ERROR MESSAGE  
AC 2: DON'T RESTORE THE PPN OR MOVE THE SIZE WORD  
(USEFUL FOR EXTENDED LOOKUPS, CASES WHERE E+4  
HAS NEEDED DATA).  
AC 1: CONVERT THE ASCIZ STRING IN E+3,4 FROM A CMUPPN TO  
A BINARY DEC PPN.

IO REQUESTS:

R READ, READ & DUMP, OR WRITE USING BUFFER AC&7 AND  
RD BLOCK E, PRINT ERROR STATUS IF ERROR OCCURS, RESET  
W STATUS TO WHAT IT WAS BEFORE.

AC OPTION:

AC 10, DON'T PRINT STATUS, BUT STORE IN LOC STAT.

LPT DUMP REQUESTS:

FILE CLOSE, REOPEN LPT FILE (MUST BE DONE BEFORE USING LPT)

DUMP DUMP CONTENTS OF BUFFER AC&7

RD READ, THEN DUMP (ONLY IF NO ERROR), VIA BUFFER AC&7

LPT CLOSE, DON'T REOPEN LPT FILE

OTHER UOOS:

CL DO CLOSE ON CHANNEL

RL DO RELEASE ON CHANNEL. MUST BE DONE IF A FILE HAD BEEN LOOKED UP,  
AND NOW YOU WANT TO DO SUPER-USER IO.

THE "REENTER" COMMAND IS THE BEST WAY TO RESTART DSKRD. IT WILL RESTORE  
ACCUMULATORS FROM THE LAST UO & REENTER DDT. >

```
54          SUBTTL  VERSION 3D, APR 1974
55          ;!!! LOAD WITH BOTH DMPSUB AND DDT!!!
56
57          IFNDEF DSKFLG,<DSKFLG==0>          ;USERMODE VERSION
58          ;ASSIGNED BITS:
59          ;BIT 35: EXEC MODE VERSION
60
61          000001  USER==<DSKFLG+1>&1
62
63          IFE USER,<
64          TITLE  EXCDISK - STANDALONE DISK FREEING UTILITY
65          IFNDEF DSKS,<DSKS==1
66          FIRSTR==0>>
67          IFN USER,<
68          TITLE  DSKRD - DISK FREEING UTILITY
69          IFNDEF DSKS,<DSKS==6
70          FIRSTR==1>>
71          EXTERN  DDT,DMPOUT,DMPCHR,DMPLIN,JOBSYM,,JBSYM,,JBUUO,,JBREN
72          INTERN  DMPHED
73
74          ;VARIOUS AC DEFS..A PRELUDE TO ALL SYMBOL PROGRAM
75          000000  F=0          ;FLAGS
76          000011  D=11         ;DSKRD UUU..LOADED FROM .JBUUO BY UUUH
77          000013  UO=13        ;UUO..USED BY FILUUO
78          000014  U=14         ;CURRENT STR (ENTRY IN BLKTAB, ACTUALLY)
79          000015  B=15         ;CURRENT BUFFER
80          000016  W=16         ;IOWD
81          000001  T1=1         ;TEMPS
82          000002  T2=2
83          000003  T3=3
84          000004  T4=4
85          000005  P1=5         ;MORE PERMANENT SORT
86          000006  P2=6
87          000007  P3=7
88          000017  SP=17        ;STACK REG
89
90          ;FLAGS IN F:
91          000001  READOP==1
92          000002  HDRIO==2
93
94          020000  ENTRB=20000
95          040000  LOQKB=40000
96          000075  JDA=75       ;STUPID LOADER CAN'T FIND IT (ACTUALLY STUPID JOBDAT)
97          000075  .JBJDA=JDA
98          000072  JOBHCU=72
99          000072  .JBHCU=JOBHCU
100
101          ;DEFINE BITS AND OTHER GOODIES FOR STORING PACK #'S
102          000006  PSIZ==6      ;SAVE 6 BITS FOR PACK (MORE THAN ENOUGH)
103          000005  PLOC==5     ;PUT IT FAR LEFT JUSTIFIED
104          770000  PBITS==770000 ;LH BITS WHERE STR # IS PUT
```

```
105            254000 000000            OPDEF    PQRST    [QRST]
106                            IFE USER,<
107                            PURGE    TTCALL
108                            OPDEF    TTCALL    [TTCAL]
109                            OPDEF    OUTCHR    [TTCALL 1,]
110                            OPDEF    OUTSTR    [TTCALL 3,]
111            DEFINE UUOS<
112                            C R,
113                            C W,
114                            C RD,
115                            C DUMP,
116                            C P
117                            C TTCAL
118            >>
119                            IFN USER,<
120            047000 777776            OPDEF    CMUDEC    [CALLI -2]
121                            DEFINE UUOS<
122                            C R,
123                            C W,
124                            C RD,
125                            C DUMP,
126                            C FILE,
127                            C P,
128                            C L,
129                            C E,
130                            C PEN,
131                            C CL,
132                            C RL>
133            071040 000000            OPDEF    LPT        [RELEASE 1,]
134                            >
135
136                            ZY==0
137                            DEFINE C(ARG)< ZY==ZY+188
138                            OPDEF    ARG        [ZY]>
139                            UUOS"
140
```

```

141          SUBTTL  INITIALIZATION
142          IFE USER,<
143              JRST  DSK          ;DISPATCH TABLE... BE SURE DSKRD IS LOADED FIRST
144              JRST  RENAD
145              JRST  DDT##
146          >
147
148          000000'
149          000000' 047 00 0 00 000000
150
151          DSK:
152          IFN USER,<          CALLI  0          ;RESET EVERYTHING>
153          IFE USER,<
154              CONQ  APR,200000      ;MUCH BETTER THAN A CALLI!
155              MOVE  13,JOBSYM      ;EXEC DDT DOESN'T LIKE THIS HERE,
156              MOVEM 13,36          ;SO WE MOVE IT WHERE IT SHOULD BE
157              MOVE  13,[JRST DMPCHAR]
158              MOVEM 13,DMPCHR      ;PATCH IN FOR EXEC LPT OUTPUT>
159              MOVE  13,[PUSHJ SP,UOOR]
160              MOVEM 13,41          ;SETUP UOOR HANDLING ROUTINE
161              MOVE  SP,STACK       ;SETUP STACK
162
163          IFN USER,<
164              MOVEI  T1,RENAD       ;INITIALIZE THE REENTER COMMAND
165              HRRM  T1,.JBREN
166              SKIPE STRLST         ;HAVE WE BEEN THROUGH THIS ONCE?
167              JRST  DDT            ;YES..NO NEED TO DO IT AGAIN
168              MOVSI 1,-DSKS        ;AN AOBJN WORD
169              SETZ  2,             ;FIRST CALL TO SYSSTR MUST BE 0
170          STRLOP: SYSSTR 2,        ;GET NEXT STR IN SYSTEM
171              JRST  DDT            ;SHOULDN'T HAPPEN...
172              JUMPE 2,DDT          ;0 RETURN MEANS NO MORE
173              MOVEM 2,STRLST(1)    ;STORE IN TABLE
174              AOBJN 1,STRLOP       ;AND LOOP TIL DONE
175          >;END OF MONITOR INITS
176              JRST  DDT            ;ENTER DDT

```

```

174          SUBTTL  UOQ PROCESSORS
175
176 000020' 200 17 0 00 000512'  RENAD:  MOVE  SP,STACK      ;RESET STACK (ESP IF WAS IN DMPQUT)
177 000021' 261 17 0 00 002664'      PUSH  SP,[DDT]      ;TO RETURN TO DDT
178 000022' 270 17 0 00 002665'      ADD   SP,[17,,17]   ;POINT TO LAST SET OF ACS
179 000023' 254 00 0 00 000041'      JRST  RENAD1       ;RETURN TO LOWEST LEVEL OF WONDERFULNESS
180
181 000024'          UOQH:
182          IFE USER,<
183              CONSZ  TTY,TTYINR      ;TTY INPUT OKY IF SO, PLAY ^C
184              JRST  RENAD            ;WITH STANDARD REENTER CODE
185          >
186 000024' 261 17 0 00 000000      PUSH  SP,0          ;NEED A TEMP REG TO PUSH REST
187          IFE USER,<
188              HRRZM  SP,SVDACO        ;REMEMBER WHERE WE PUT ACO FOR EASY RETRIEVAL
189          >
190 000025' 201 00 0 00 000001      MOVEI  0,1          ;PUSH FROM 1 UP
191 000026' 261 17 1 00 000000      PUSH  SP,@0        ;SURE, BLT IS FASTER,
192 000027' 305 00 0 00 000016      CAIGE  0,16        ;BUT WON'T MAKE PDL TRAP!
193 000030' 344 00 0 00 000026'      ADJA  0,.-2        ;DO UP TO 16
194 000031' 200 14 0 00 000404'      MOVE  U,STR        ;COMMONLY USED VALUE
195 000032' 400 00 0 00 000000      SETZ  F,           ;INITIALLY NO FLAGS
196 000033' 200 11 0 00 000000*     MOVE  D,.JBUUD     ;IF UOQ ROUTINES LOOK AT D INSTEAD
197                                     ;OF .JBUUD, UOQS MAY CALL OTHER UOQS,
198                                     ;INCLUDING THEMSELVES (LET'S HEAR IT FOR RECURSION!)
199 000034' 135 01 0 00 002666'      LDB   T1,[POINT 9,D,8];GET OPCODE
200 000035' 307 01 0 00 000020      CAIG  T1,U2-U1     ;MAKE SURE IN RANGE
201 000036' 260 17 1 01 000045'      PUSHJ SP,@U1-1(T1) ;CALL IT
202 000037' 254 00 0 00 000041'      JRST  .+2         ;NON-SKIP
203 000040' 350 00 0 17 777761      AOS   -17(SP)     ;SKIP RET TO CALLER
204 000041' 201 00 0 00 000016      RENAD1: MOVEI  0,16 ;BETTER POP IN INVERSE ORDER
205 000042' 262 17 1 00 000000      POP   SP,@0       ;ONE OF THEM
206 000043' 367 00 0 00 000042'      SOJG  0,.-1       ;NO, DO NEXT
207 000044' 262 17 0 00 000000      POP   SP,0        ;AND, OF COURSE, 0
208 000045' 263 17 0 00 000000      POPJ  SP,         ;RETURN
209
210 000046'          U1:1  DEFINE C(ARG)< ARG'H>
211          UOQS^
212          C R,^    RH^
213          C W,^    WH^
214          C RD,^   RDH^
215          C DUMP,^ DUMPH^
216          C FILE,^ FILEH^
217          C P,^    PH^
218          C L,^    LH^
219          C E,^    EH^
220          C REN,^  RENH^
221          C CL,^   CLH^
222          C RL,^   RLH^
223          REPEAT 5,< CPOPJ>          ;FOR PATCHED IN UOQS
224 000061' 000000 000223'      CPOPJ
225 000062' 000000 000223'      CPOPJ
226 000063' 000000 000223'      CPOPJ
227 000064' 000000 000223'      CPOPJ
228 000065' 000000 000223'      CPOPJ

```

229 000065'

U2:1

```
230            IFE USER,<  
231    TICALL: MOVEI    T1,(D)            ;SEE IF WE SHOULD DIDDLE THE ADDRESS FIELD,  
232            CAIG    T1,17            ;IN CASE THEY GAVE US AN ACCUMULATOR  
233            ADD    D,SVDAC0        ;YES, ADD IN OFFSET  
234            LDB    T1,PUUOAC        ;GET TICALL CODE  
235            SKIPN   TUODSP(T1)       ;IS THERE ONE?  
236            HALT    BELL            ;NO!  
237            JRST   @TUODSP(T1)      ;DO IT  
238  
239    TUODSP: 0  
240            TICCHR                    ;OUTCHR  
241            0  
242            TTCSTR                    ;OUTSTR  
243            REPEAT 14,<            0>  
244  
245    TTCSTR: HRLI    D,(POINT 7,.)    ;MAKE A BYTE POINTER  
246    TTSTR1: ILDB    T1,D            ;NEXT CHARACTER  
247            JUMPE   T1,CPOPJ        ;QUIT WHEN DONE  
248            PUSHJ   SP,CTYTYP       ;PRINT IT  
249            JRST    TTSTR1           ;ANOTHER?  
250  
251    TICCHR: MOVE    T1,(D)            ;GET CHARACTER  
252            PURST   CTYTYP           ;AND PRINT IT  
253            >
```

```

254          SUBTTL  UUD CODE
255
256          IFE USER,<
257          RHDH:  TLOA      F,READOP+HDRIO  ;READING HEADERS
258          >
259          RH:    TLO      F,READOP      ;SAY WE'RE READING
260          PUSHJ  SP,SETPNT  ;SET UP POINTER
261          MOVEI  13,(D)     ;RH HAS BLOCK NUMBER
262          DPB    14,PPTR    ;AND 14 HAS THE PACK NUMBER
263          MOVEM  13,BLKTAB(15) ;GET AND SAVE PARAMETERS
264          TLZ    13,PBITS   ;REMOVE PACK
265          PJRST  TRYIO     ;TRY TRANSFER
266
267          IFE USER,<
268          WHDH:  TLO      F,HDRIO        ;WRITING HEADERS
269          >
270          WH:    TLZ      F,READOP      ;SAY WE'RE WRITING
271          PUSHJ  SP,SETPNT  ;SETUP POINTER
272          MOVEI  13,(D)     ;LIKE READH
273          DPB    14,PPTR    ;
274          CAME  13,BLKTAB(15) ;SAME ONE?
275          JRST   HUH1      ;NOPE DON'T LET HIM
276          TLZ    13,PBITS   ;REMOVE PACK
277          PJRST  TRYIO     ;TRY TRANSFER
278
279          RDH:   PUSHJ  SP,RH      ;READ A BLOCK
280          PJRST  DUMPH     ;DUMP AND RETURN
281          PJRST  CPOPJ1    ;ON ERROR
  
```



```

282 000110'
283
284 000110' 331 00 0 00 000076
285 000111' 264 00 0 00 000000*
286
287 000112' 260 17 0 00 000355'
288 000113' 200 01 0 15 000374'
289 000114' 202 01 0 00 000373'
290 000115' 201 16 0 16 000001
291 000116' 201 15 0 16 000000
292 000117' 275 15 0 00 000663'
293 000120' 201 14 0 00 000200
294 000121' 264 00 0 00 000000*
295 000122' 200 16 0 00 000017
296 000123' 201 07 0 00 000014
297 000124' 265 17 0 00 000000*
298
299
300
301
302 000125' 067 01 0 00 000000
303 000126' 200 17 0 00 000016
304 000127' 263 17 0 00 000000
305
306
307 000130' 264 00 0 00 000111*
308 000131' 263 17 0 00 000000
309
310
    
```

```

DUMPH:
IFN USER,<
    SKIPL .JBDATA1
    JSR   DMPFIL
>
    PUSHJ SP,SETPNT
    MOVE  T1,BLKTAB(B)
    MOVEM T1,BLKTAB-1
    MOVEI 16,1(16)
    MOVEI 15,(16)
    SUBI  15,BLK0
    MOVEI 14,200
    JSR   DMPQUT
    MOVE  16,17
PRTFRM: MOVEI 7,14
    JSP   17,DMPCHR
IFE USER,<
    CONSZ LPT,LPTBSY
    JRST  .-1
    CONO  LPT,LPTRES
;>
    OUTPUT 1,
    MOVE  17,16
    POPJ  SP,
IFN USER,<
FILEH: JSR   DMPFIL
    POPJ  SP,
    EXTERN DMPFIL
>
    
```

*T1, 1 ✓*  
*T2, 1 ✓*

```

;LPT IS (WAS?) ON CHAN 1. IS IT OPEN?
;OPEN IT

;SET UP IOWD
;PUT BLOCK POINTER WHERE IT CAN BE FOUND
;EASILY
;ALL WE WANT IS THE START ADDR
;WE CAN FIND BLK #
;KLUDGE AROUND ASSEMBLER'S SHORT COMINGS (=BLK0(16))
;PRINT 200 WORDS
;DUMP IT
;KLUDGY INTERFACE
;THROW OUT A FORMFEED
;OR SOMETHING PATCHED

;LPT STILL AT IT?
;YEP
;CLEAR IT, SO WE CAN GET THE OUTPUT OFF
;THIS ONCE WAS THE LPT CHANNEL

;RETURN
    
```

```

311
312 000132' 336 02 0 11 000653' IFN USER,<
313 000133' 254 00 0 00 000371' PH: SKIPN T2,STRLST(D) ;T2_DEVNAM
314 000134' 047 02 0 00 000064 DEVNAM T2, ;NO DEV IS NO GOODNESS
315 000135' 254 00 0 00 000371' JRST BELL ;GET REAL DEVICE NAME FROM LOGICAL NAME
316 000136' 201 03 0 11 000010 MOVEI T3,10(D) ;DEVICE DOESN'T EXIST?
317 000137' 047 03 0 00 000064 DEVNAM T3, ;CHANNEL #
318 000140' 254 00 0 00 000143' JRST PH1 ;SEE WHAT REALLY IS THERE
319 000141' 316 03 0 00 000002 CAMN T3,T2 ;ASSUME NO CHANNEL OPEN
320 000142' 254 00 0 00 000156' JRST PH2 ;AGREE?
321 000143' 621 11 0 00 000200 PH1: TLZ D,200 ;YES.. DON'T REINIT (AND WIPE OUT LOOKUP)
322 000144' 201 01 0 00 000017 MOVEI T1,17 ;FLAG THIS CONDITION FOR TYPEOUT
323 000145' 200 03 0 00 000002 MOVE T3,T2 ;DUMP MODE (ALWAYS)
324 000146' 047 03 0 00 000004 DEVCHR T3, ;NOW GET DEVICE NAME TO SEE IF DTA
325 000147' 603 03 0 00 000100 TLNE T3,(1B11) ;DO IT
326 000150' 660 01 0 00 000300 TRO T1,300 ;DECTAPE?
327 000151' 400 03 0 00 000000 SETZ T3, ;SET SPECIAL BITS
328 000152' 200 13 0 00 002667' MOVE UD,[OPEN 10,T1] ;NO BUFFERS
329 000153' 201 14 0 11 000000 MOVEI U,(D) ;WHAT TO DO
330 000154' 260 17 0 00 000217' PUSHJ SP,FILUUD ;TRY THIS UNIT
331 000155' 254 00 0 00 000366' PJRST HUH1 ;DO IT
332 000156' 607 11 0 00 000600 PH2: TLNN D,600 ;DIDN'T1
333 000157' 260 17 0 00 000325' PUSHJ SP,TYPSIX ;IF AC 10, DON'T1 PRINT
334 000160' 552 11 0 00 000404' HRRZM D,STR ;PRINT STR NAME NOW ON
335 000161' 254 00 0 00 000222' PJRST CPOPJ1 ;SAVE NEW STR NUMBER

```

```

336 >
337
338 IFE USER,< ;EXEC MODE - USE ASKED FOR PACK
339 PH: TRNE D,777770 ;ILLEGAL PACK #?
340 PJRST HUH ;YEP
341 MOVEI T1,(D) ;LET'S GO SEE HOW IT IS
342 SETZB T2,T4 ;NO BLOCK OR ATTNS
343 MOVEI T3,F,SLDR ;JUST SELECT IT
344 PUSHJ SP,DPCDO ;LOOK AT IT
345 PUSHJ SP,DPCCHK ;WONDERFUL?
346 JRST PH1 ;NO - LEAVE IT SELECTED, THOUGH
347 HRRZM D,PACK ;POINT TO THIS ONE
348 TLNE D,600 ;SHOULD WE RECALIBRATE IT?
349 PJRST CPOPJ1 ;NO
350 MOVEI T1,(D) ;GET PACK #
351 SETZB T2,T4 ;NO CYL, NO ICW
352 MOVEI T3,F,RECL ;RECALIBRATE REQUEST
353 PUSHJ SP,DPCDO ;DO IT
354 PUSHJ SP,DPCWAI ;WAIT FOR COMPLETION
355 PJRST TIMEOUT ;HUNG?
356 PUSHJ SP,DPCCHK ;HOW IS IT?
357 PH1: PJRST DING1 ;NOT VERY GOOD, COMPLAIN
358 JRST CPOPJ1 ;FINE AND DANDY.
359 >

```

```

360 IFN USER,<
361 000162' 205 13 0 00 055411 RENH: MOVSI UO,(RENAME 10,(D)) ;RENAME STUFF
362 000163' 254 00 0 00 000166' JRST LH+1 ;ENTER LOOKUP STUFF
363 000164' 334 13 0 00 002670' EH: SKIPA UO,[ENTER 10,(D)]
364 000165' 205 13 0 00 076411 LH: MOVSI UO,(LOOKUP 10,(D))
365 000166' 006 04 0 14 000000 F 4,(U) ;BE SURE WE HAVE THIS UNIT
366 000167' 254 04 0 00 000371' HALT BELL ;WE DIDN'T!
367 000170' 607 11 0 00 000040 TLNN D,40 ;CMUPPN IN IT?
368 000171' 254 00 0 00 000176' JRST LH1 ;NO..PROCESS NOW
369 000172' 201 01 0 11 000003 MOVEI T1,3(D) ;ADDR OF PPN
370 000173' 505 01 0 01 000000 HRLI T1,(T1) ;DOUBLE IT
371 000174' 047 01 0 00 777776 CMUDEC T1, ;CONVERT
372 000175' 254 00 0 00 000370' JRST HUH ;COMPLAIN VEHEMENTLY
373 000176' 200 06 0 11 000003 LH1: MOVE P2,3(D) ;GET PPN
374 000177' 260 17 0 00 000217' PUSHJ SP,FILUUD ;DO IT
375 000200' 661 13 0 00 400000 TLO UO,400000 ;FLAG A FAILURE
376 000201' 603 11 0 00 000100 TLNE D,100 ;AC2..DON'T CHANGE UO BLOCK
377 000202' 254 00 0 00 000205' JRST LH2
378 000203' 250 06 0 11 000003 EXCH P2,3(D) ;RESTORE PPN
379 000204' 202 06 0 11 000004 MOVEM P2,4(D) ;SAVE SIZE
380 000205' 325 13 0 00 000212' LH2: JUMPGE UO,PAUSE1 ;JUMP ON GOODNESS
381 000206' 550 01 0 11 000001 HRRZ T1,1(D) ;GET ERROR RETURN
382 000207' 607 11 0 00 000400 TLNN D,400 ;IF SET, NO MSG
383 000210' 260 17 0 00 000351' PUSHJ SP,TYPOCR ;PRINT "ERROR #"
384 000211' 254 00 0 00 000371' JRST BELL ;REINFORCE AND RETURN
385
386
387 000212' 400 01 0 00 000000 PAUSE1: SETZ T1, ;SLEEP A JIFFY TO FORCE JDA OUT
388 000213' 047 01 0 00 000031 SLEEP T1,
389 000214' 254 00 0 00 000222' PJRST CPOPJ1 ;AND SAY LOOKUP OR ENTER WORKED
390
391 000215' 334 13 0 00 002671' RLH: SKIPA UO,[RELEASE 10,] ;FOR A RELEASE,
392 000216' 205 13 0 00 070400 CLH: MOVSI 13,(CLOSE 10,) ;FOR A CLOSE
393 000217' 137 14 0 00 002672' FILUUD: DPB 14,[POINT 3,13,12] ;STORE THAT
394 000220' 256 00 0 00 000013 XCT 13 ;XCT IT
395 000221' 263 17 0 00 000000 POPJ SP, ;NON SKIP
396 ; PJRST CPOPJ1
397 >
398
399 000222' 350 00 0 17 000000 CPOPJ1: AOS (SP) ;FOR A SKIP RETURN
400 000223' 263 17 0 00 000000 CPOPJ: POPJ SP, ;OR NON-SKIP

```

SUBTTL SUBROUTINES

```
401
402
403 IFN USER,<
404 000224' 006 04 0 14 000000 TRYIO: P 4,(U) ;INSURANCE TO BE SURE CHN OPEN
405 000225' 254 04 0 00 000367' HALT DING ;SHOULDN'T HAPPEN
406 000226' 201 02 0 11 000000 MOVEI T2,(D) ;JUST THE BLOCK # (MONITOR WANTS WHOLE WORD)
407 000227' 200 16 0 14 000653' MOVE 16,STKLST(14) ;GET DEV NAME
408 000230' 047 16 0 00 000004 DEVCHR 16, ;FIND OUT WHAT SORT OF DEV
409 000231' 607 16 0 00 200000 TLNN 16,(1B1) ;DISK?
410 000232' 254 00 0 00 000240' JRST NMUSET ;NO..USET ALWAYS TAKES BLK #
411 000233' 201 16 0 00 000002 MOVEI 16,T2 ;USETI CH,T2 IF SUPER
412 000234' 205 01 0 00 040000 MOVSI T1,LOOKB ;SEE IF LOOKUP DONE
413 000235' 607 00 0 00 000001 TLNN F,READOP ;UNLESS WE'RE WRITING,
414 000236' 205 01 0 00 020000 MOVSI T1,ENTRB ;IN WHICH CASE WE CHECK FOR AN ENTER
415 000237' 612 01 0 14 000105 TDNE T1,.JBJDA+10(U) ;CHECK FOR LOOKUP OR ENTER DONE
416 000240' 201 16 0 02 000000 NMUSET: MOVEI 16,(T2) ;OR USETI CH,(T2) IF NORMAL
417 000241' 205 13 0 00 074416 MOVSI UO,(USETI 10,(16)) ;USETI?
418 000242' 607 00 0 00 000001 TLNN F,READOP ;NOT IF THIS BIT IS ON
419 000243' 205 13 0 00 075416 MOVSI UO,(USETO 10,(16)) ;THEN A USETO
420 000244' 260 17 0 00 000217' PUSHJ SP,FILUOO ;DO IT
421 000245' 205 13 0 00 056400 MOVSI UO,(IN 10,) ;IN?
422 000246' 607 00 0 00 000001 TLNN F,READOP
423 000247' 205 13 0 00 057400 MOVSI UO,(OUT 10,) ;OUT
424 000250' 541 13 0 00 000405' HRRI UO,PNT ;WHERE ALL DUMP IO LIVES
425 000251' 260 17 0 00 000217' PUSHJ SP,FILUOO ;TRY IT
426 000252' 263 17 0 00 000000 POPJ SP, ;ALL WONDERFUL
427 000253' 200 13 0 00 002673' MOVE UO,[GETSTS 10,P1]
428 000254' 260 17 0 00 000217' PUSHJ SP,FILUOO ;FIND OUT JUST WHAT WENT WRONG
429 000255' 201 01 0 05 000000 MOVEI T1,(P1) ;FOR TYPOCT (WHO HAPPILY DESTROYS IT)
430 000256' 607 11 0 00 000400 TLNN D,400 ;PRINT ERROR?
431 000257' 260 17 0 00 000351' PUSHJ SP,TYPOCR ;PRINT FILE STATUS
432 000260' 202 05 0 00 000413' MOVEM P1,STAT ;FOR ANYONE TO REFERENCE
433 000261' 620 05 0 00 760000 TRZ P1,760000 ;CLEAR ERROR FLAGS
434 000262' 205 13 0 00 060405 MOVSI UO,(SETSTS 10,(P1))
435 000263' 260 17 0 00 000217' PUSHJ SP,FILUOO ;CLEAR ERROR
436 000264' 254 00 0 00 000367' PJRST DING ;RETURN
437
```



>

```

438            IFE USER,<
439            TRYIO:    ADS            (SP)                    ;BE PESSIMISTIC.. BET IT WON'T WORK
440                    MOVE            T1,PACK                    ;PICK UP PACK #
441                    MOVE            T2,CURBUF                   ;FIND BLOCK #
442                    HRRZ            T2,BLKTAB(T2)               ;AND BLOCK #
443                    MOVEI           T3,F,SEEK                   ;SEEK FIRST
444                    SETZ            T4,                         ;NO ICW
445                    PUSHJ           SP,DPCDO                    ;SEEK AND YE SHALL FIND
446                    PUSHJ           SP,DPCWAI                   ;WAIT TO MAKE SURE
447                       PJRST        TIMEOUT                    ;DIDN'T FIND IT THIS TIME
448                    PUSHJ           SP,DPCCHK                   ;CHECK FOR OTHER ERRORS
449                       POPJ         SP,                         ;GIVE UP
450                    MOVE            T1,PACK                    ;NOW FOR THE OPERATION
451                    MOVE            T2,CURBUF                   ;GET BLOCK # AGAIN
452                    HRRZ            T2,BLKTAB(T2)               ;PACK AND BLOCK
453                    MOVEI           T3,F,READ                   ;PROBABLY A READ
454                    TLNN            F,READOP                    ;BETTER MAKE SURE
455                    MOVEI           T3,F,WRIT                   ;NOPE, WRITE
456                    TLNE            F,HDRI0                    ;PLAYING WITH HEADERS?
457                    ADDI            T3,F,RHDR-F,READ;YES, ADD OFFSET
458                    MOVEI           T4,DPCICW                   ;WHERE THERE'S A CCW LIST
459                    PUSHJ           SP,DPCDO                    ;START IT UP
460                    PUSHJ           SP,DPCWAI                   ;WAIT FOR IT
461                       PJRST        TIMEOUT                    ;NOT GOOD.
462                    PUSHJ           SP,DPCCHK                   ;CHECK FOR OTHER ERRORS
463                       JRST         .+2                         ;NO GOOD
464                    SOS            (SP)                         ;WOW! ALL THE WAY THROUGH AND NO ERRORS!
465                    POPJ            SP,                         ;ALL DONE
466
467            TIMEOUT: OUTSTR    [ASCIZ/TIMEOUT ERROR
468                    /]
469                    PUSHJ           SP,DPCCHK                   ;TAKE A CLOSER LOOK
470                       JFCL                                    ;BUT WE DON'T REALLY CARE
471                    PUSHJ           SP,CRLF                    ;LEAVE A LINE
472                    POPJ            SP,                         ;ASK FOR MORE
473                    >

```

```

474 000265' 000000 000000            DMPHED: 0                    ;JSR HERE FOR A HEADING
475 000266' 201 11 0 00 777777        MOVEI     11,-1                ;DON'T COME HERE AGAIN
476 000267' 202 11 0 00 000000*        MOVEM    11,DMP LIN            ;
477 000270' 200 11 0 00 002676'        MOVE     11,[POINT 7,[ASCIZ/        BLOCK /]]
478 000271' 201 17 0 00 000272'        MOVEI    17,.,+1               ;RETURN FOR DMPCHR
479 000272' 134 07 0 00 000011        ILDB     7,11                 ;GET A CHAR
480 000273' 327 07 0 00 000124*        JUMPG    7,DMPCHR             ;LOOP AND PRINT
481 000274' 200 13 0 00 000373'        MOVE     13,BLK TAB-1         ;GET BUFFER WORD
482 000275' 200 11 0 00 002677'        MOVE     11,[POINT 3,13,17]     ;POINTER TO BLK # TO PRINT
483 000276' 201 12 0 00 000006        MOVEI    12,6                 ;PRINT 6 NUMBERS
484 000277' 134 07 0 00 000011        ILDB     7,11                 ;GET 1 OF THEM
485 000300' 201 07 0 07 000060        MOVEI    7,60(7)              ;ASCII IT
486 000301' 265 17 0 00 000273*        JSP      17,DMPCHR             ;PRINT IT
487 000302' 367 12 0 00 000277'        SOJG     12,.-3               ;LOOP
488 000303' 200 11 0 00 002702'        MOVE     11,[POINT 7,[ASCIZ/ FROM /]]
489 000304' 201 17 0 00 000305'        MOVEI    17,.,+1               ;RETURN ADDR FOR DMPCHR
490 000305' 134 07 0 00 000011        ILDB     7,11                 ;GET A CHARACTER
491 000306' 327 07 0 00 000301*        JUMPG    7,DMPCHR             ;LOOP AND PRINT
492 000307' 135 13 0 00 000407'        LDB      13,PPTR              ;GET STR NUMBER
493 000310' 200 13 0 13 000653'        MOVE     13,STRLST(13)        ;NOW STR NAME
494 000311' 200 12 0 00 002703'        MOVE     12,[POINT 6,13]     ;TO PRINT IT
495 000312' 134 07 0 00 000012        DMPLOP: ILDB    7,12            ;ONE CHARACTER
496 000313' 201 07 0 07 000040        MOVEI    7,40(7)              ;ASCII IT
497 000314' 265 17 0 00 000306*        JSP      17,DMPCHR             ;PRINT IT
498 000315' 603 12 0 00 770000        TLNE     12,770000            ;DONE WHEN THESE ARE ALL ZERO
499 000316' 254 00 0 00 000312'        JRST     DMPLOP               ;BUT NOT YET
500 000317' 201 07 0 00 000015        MOVEI    7,15                 ;CR
501 000320' 265 17 0 00 000314*        JSP      17,DMPCHR             ;PRINT IT
502 000321' 201 07 0 00 000012        MOVEI    7,12                 ;LF
503 000322' 265 17 0 00 000320*        JSP      17,DMPCHR             ;PRINT IT TOO
504 000323' 265 17 0 00 000322*        JSP      17,DMPCHR             ;PRINT IT TWO
505 000324' 254 00 1 00 000265'        JRST     @DMPHED              ;RETURN
  
```

```

506 000325' 200 01 0 00 002704'   TYPSEX: MOVE    T1,[POINT 6,T2] ;T2 HAS SIXBIT DEVNAM
507 000326' 134 03 0 00 000001     ILDB     T3,T1      ;GET A CHAR
508 000327' 322 03 0 00 000223'   JUMPE   T3,CPOPJ   ;RETURN ON NULL (IF NOT FROM T2, THEN T3)
509 000330' 201 03 0 03 000040     MOVEI   T3,40(T3)  ;ASCII IT
510 000331' 051 01 0 00 000003     OUTCHR  T3         ;PRINT IT
511 000332' 254 00 0 00 000326'   JRST   TYPSEX+1   ;LOOP
512
513 000333' 261 17 0 00 000001     TYPHAF: PUSH   SP,T1      ;SAVE FOR AWHILE
514 000334' 554 01 0 00 000001     HLRZ   T1,T1      ;FIRST GET LH
515 000335' 260 17 0 00 000341'   PUSHJ  SP,TYPOCT  ;DONE.
516 000336' 051 01 0 00 002705'   OUTCHR [" "]     ;SEPARATE
517 000337' 262 17 0 00 000001     POP    SP,T1      ;GET BACK
518 000340' 201 01 0 01 000000     MOVEI  T1,(T1)    ;JUST RH
519
520 000341' 231 01 0 00 000010     TYPOCT: IDIVI   T1,10    ;STANDARD OCTAL # PRINTOUT
521 000342' 506 02 0 17 000000     HRLM   T2,(SP)    ;SAVE LAST CHARACTER
522 000343' 322 01 0 00 000345'   JUMPE  T1,TYPO1  ;PRINT IF ONLY ONE
523 000344' 260 17 0 00 000341'   PUSHJ  SP,TYPOCT  ;PRINT THE REST
524 000345' 554 01 0 17 000000     TYPO1: HLRZ   T1,(SP)  ;GET THE NUMBER BACK
525 000346' 201 01 0 01 000060     MOVEI  T1,60(T1)  ;ASCII IT
526 000347' 051 01 0 00 000001     OUTCHR T1         ;PRINT IT
527 000350' 263 17 0 00 000000     POPJ   SP,        ;AND RETURN
528
529 000351' 051 03 0 00 002706'   TYPOCR: OUTSTR  [ASCIZ/ERROR /]
530 000352' 260 17 0 00 000341'   PUSHJ  SP,TYPOCT  ;PRINT SUPPLIED CODE
531 000353' 051 03 0 00 002710'   CRLF:  OUTSTR  [BYTE (7) 15,12,0]
532 000354' 263 17 0 00 000000     POPJ   SP,

```

```

533 000355' SETPNT:
534 IFE USER,<
535 MOVEI 16,PNT
536 MOVEM 16,DPCICW ;SETUP CCW LIST
537 >
538 000355' 135 16 0 00 002711' LDR 16,[POINT 3,D,12]
539 000356' 201 15 0 16 000000 MOVEI 15,(16) ;SAVE BUFFER #
540 000357' 202 15 0 00 000412' MOVEM 15,CURBUF ;KEEP TRACK OF IT,,HANDY VALUE
541 000360' 242 16 0 00 000007 LSH 16,7 ;GET AC FIELD & MULTIPLY BY 200
542 000361' 201 16 0 16 000662' MOVEI 16,BLKO-1(16)
543 000362' 505 16 0 00 777600 HRLI 16,-200 ;FINISH IOWD
544 000363' 202 16 0 00 000405' MOVEM 16,PNT ;SAVE FOR USE
545 000364' 402 00 0 00 000406' SETZM PNT+1 ;JUST TO MAKE SURE
546 000365' 263 17 0 00 000000 POPJ SP, ;RETURN
547
548 000366' 354 00 0 17 000000 HUH1: AOSA (SP) ;BELL, ?, SKIP
549 000367' DING:
550 000367' 354 00 0 17 000000 DING1: AOSA (SP) ;JUST BELL, SKIP
551 000370' 051 01 0 00 002712' HUH: OUTCHR ["?"] ;JUST ?, BELL
552 000371' 051 03 0 00 002713' BELL: OUTSTR [BYTE (7) 7,177] ;JUST BELL (FOR TTY), RUBOUT (FOR ARDS)
553 000372' 263 17 0 00 000000 POPJ SP,
554
555 IFE USER,< ;SPECIAL STANDALONE GARBAGE
556 DMPCHA: CONSZ LPT,LPTERR ;IS IT UNHAPPY?
557 HALT .-1 ;YES, LET PEOPLE FIX AND TRY AGAIN
558 CONSZ LPT,LPTBSY ;OR IS IT BUSY?
559 JRST .-1 ;WAIT UNTIL IT'S NOT
560 LSH 7,1 ;MAKE IT ACCEPTABLE TO THE PRINTER
561 DATAO LPT,7 ;MAKE IT EAT AC7
562 LSH 7,-1 ;THIS WASN'T SUPPOSED TO CHANGE
563 JRST (17) ;GO AWAY
564
565 LPTDON==100 ;LPT DONE FLAG
566 LPTBSY=200 ;LPT BUSY CONDITION (WHAT ELSE)
567 LPTERR=400
568 LPTRES==2000
569
570 CTYTYP: CONSZ TTY,TTYBSY ;IS IT BUSY?
571 JRST .-1 ;YEAH, BETTER WAIT
572 DATAO TTY,T1 ;PRINT IT
573 TRNE T1,340 ;CONTROL CHAR?
574 POPJ SP, ;NO..GO AWAY
575 PUSH SP,T1 ;SAVE IT
576 MOVEI T1,177 ;FAKE SOME TIMING
577 PUSHJ SP,CTYTYP ;THROW IT OUT
578 POP SP,T1 ;RESTORE CHAR
579 POPJ SP, ;AND RETURN
580
581 TTYBSY=20 ;OUTPUT BUSY
582 TTYINR==40 ;INPUT READY (FOR ^C FAKING)
583 >

```



```
584                    SUBTTL    STANDALONE DISK SUPPORT
585
586                    IFE USER,<
587                    F.READ==0                                    ;PACK FUNCTIONS
588                    F.WRIT==1
589                    F.RHDR==2
590                    F.WHDR==3
591                    F.SEEK==4
592                    F.CATN==5
593                       %CLALL==776                                ;HOW TO CLEAR THEM ALL
594                    JSWAI:F,SLDR==6
595                    F.RECL==7
596                    DPCICW==20                                    ;WHERE CCW LIST STARTS
597                    D.DONE==10                                    ;INTERRUPT FLAG FROM RP10
598
599                    COMMENT \
600                    THIS ROUTINE WILL ASSEMBLE AN XCT A DPC INSTRUCTION. PARAMETERS:
601                    T1:        UNIT
602                    T2:        BLOCK ON UNIT
603                    T3:        FUNCTION
604                    T4:        ICW ADDR OR ATTN BITS
605                    T1 IS THE ONLY AC NOT CLOBBERED
606                    \
607
608                    DPCDO:    DPB        T3,FNCPTR                ;STORE FUNCTION
609                                DPB        T1,UNIPTR             ;AND UNIT
610                                DPB        T4,ATNPTR             ;AND ATTNS
611                                IDIV      T2,BLKPCL             ;T2_ CYLINDER
612                                DPB        T2,CYLPTR             ;SAVE IT
613                                IDIV      T3,BLKPSF             ;T3_SURFACE, T4_SECTOR
614                                DPB        T3,SURPTR             ;SAVE BOTH
615                                DPB        T4,SECPTR
616                                DATAQ    DPC,DSKINS             ;DO IT
617                                POPJ      SP,                     ;AND RETURN
618
619                    BLKPUN:    ^D40000                             ;SETUP BLOCK PER LOCATIONS
620                    BLKPCL:    ^D200                              ;LOCATIONS, SO THEY CAN BE CHANGED.
621                    BLKPSF:    ^D10                                ;NEED: PER UNIT, CYLIDER, SURFACE
622
623                    FNCPTR:    POINT 3,DSKINS,2                    ;ALSO NEED POINTERS TO DATAQ WORD
624                    UNIPTR:    POINT 3,DSKINS,5
625                    CYLPTR:    POINT 8,DSKINS,13
626                    SURPTR:    POINT 5,DSKINS,18
627                    SECPTR:    POINT 5,DSKINS,23
628                    ATNPTR:    POINT 9,DSKINS,35
629                    DSKINS:    0
630
```

```
631 ;ROUTINE TO WAIT FOR COMPLETION - ERROR IF HANGS
632 DPCWAI: MOVSI T1,1 ;WAIT A WHILE
633 CONSO DPC,D.DONE ;DONE YET?
634 SOJG T1,-1 ;WAIT
635 JUMPG T1,CPOPJ1 ;HAPPY IF STILL POS
636 POPJ SP, ;OTHERWISE AN ERROR RETURN
637
638 CLRAT1: AOS (SP) ;CLRATN WITH SKIP
639 CLRATN: MOVE T1,PACK ;LEAVE CURRENT DRIVE SELECTED
640 SETZ T2, ;NO BLOCK DATA
641 MOVEI T3,F,CATN ;CLEAR ATTNS
642 SETO T4, ;ALL OF THEM
643 PJRST DPCDO ;DO AND RETURN
644
645 SUBTTL DEVICE STATUS AND ERROR ROUTINES
646
647 DEVSTA:
648 DEVST1: OUTSTR [ASCIZ/CONI: /]
649 CONI DPC,P1 ;GET STATUS
650 MOVE P2,[IOWD 14,DPCCON];POINT TO TEXT EXPLANATIONS
651 MOVEI P3,4 ;IGNORE PICHAN
652 PUSHJ SP,STASUB ;SEE WHAT WE HAVE
653 OUTSTR [ASCIZ/DATAI: /]
654 DATAI DPC,P1 ;GET DATAI
655 MOVE P2,[IOWD 10,DPCDAI]
656 MOVEI P3,400000 ;DPC DATAI BITS IN LH ONLY
657
658 STASUB: MOVE T1,P1 ;PUT WHERE PRINTABLE
659 PUSHJ SP,TYPHAF ;PRINT
660 STAS1: AOBJP P2,CRLF ;STOP IF TOO FAR
661 LSH P3,1 ;ROTATE BIT
662 TDNN P1,P3 ;BIT ON?
663 JRST STAS1 ;NO LSH AND TRY AGAIN
664 OUTCHR [" "]
665 MOVE T2,(P2) ;WORD OF EXPLANATION
666 PUSHJ SP,TYPSIX ;PRINT
667 JRST STAS1 ;AND LOOP
668
```

```
669            DPCCHK: CONSO    DPC,377720        ;ANY ERRORS?  
670                            JRST    CLRAT1        ;NO,    GIVE SUCCESSFUL RETURN  
671                            PUSHJ   SP,DEVSTA     ;YES..COMPLAIN  
672                            PJRST   CLRATN        ;YES - ERROR  
673            DEFINE   ZZ(X)  
674            <IRP X< SIXBIT/X/>>  
675            DPCCON: ZZ <DONE,BUSY,CWXPB,SURFX,SECX,DATAO,ILL,W,N,RDY>  
676                            ZZ <PARER,NXM,LATE,SRCH,POWER,EDC,SDONE,PE,DW,PE,CD,PE,DS,PE,CW>  
677            DPCDA1: ZZ <HDRSW,RDQNL,NXDR,UNSAF,ONLIN,ONCYL,SEEKN>  
678            DPCBSP==<DPCCON-DPCDA1-1>,,DPCDA1-.-1  
679            >
```

```
680                    SUBTTL DATA BASE
681
682 000373' 000000 000000                    0                    ;WHERE DUMPH SAVES THE BLKTAB ENTRY FOR DMPHD
683 000374'                    BLKTAB: REPEAT 10,<-1> ;BLOCK # NOW IN
684 000374' 777777 777777                    -1
685 000375' 777777 777777                    -1
686 000376' 777777 777777                    -1
687 000377' 777777 777777                    -1
688 000400' 777777 777777                    -1
689 000401' 777777 777777                    -1
690 000402' 777777 777777                    -1
691 000403' 777777 777777                    -1
692
693 000404' 000000 000001                    STR:    PACK:    FIRSTR ;CONTAINS PACK NUMBER CURRENTLY ACTIVE
694 000405' 000000 000000                    PNT:    0                    ;USED FOR DUMP I/O IOWD
695 000406' 000000 000000                                       0                    ;STOP CODE FOR DUMP I/O
696 000407' 36 06 0 00 000013                PPTR:    POINT    PSIZ,13,PLOC ;THIS MAY UNREALISTIC, STORING IN AC13
697 000410' 27 04 0 00 000011                PDUUAC: POINT    4,D,12 ;FOR GETTING AC FROM UO
698
699 000411' 000000 000000                    SVDACO: 0                    ;WHERE AC 0 WAS PUSHED ON LAST UO
700 000412' 000000 000000                    CURBUF: 0                    ;CURRENT BUFFER WE'RE LOOKING AT
701 000413' 000000 000000                    STAT:    0                    ;LAST IO ERROR STATUS
702                    IFN USER,<
703
704 000414' 000001 000001                    LOOK:    1,,1                    ;FOR LOOKUPS AND ENTERS
705 000415' 65 46 44 00 00 00                                   SIXBIT/UFD/
706 000416' 000000 000000                                       0
707 000417' 000001 000001                                       1,,1
708 000420' 000000 000000                                       0                    ;FOR CMUDEC CONVERSION
709 >
```

```

710 IFN USER,< ;WON'T WORK STANDALONE
711 000421' 005 00 0 00 000000 PROG: FILE ;OPEN LPT FILE
712 000422' 205 02 0 00 777770 MOVSI 2,-10 ;ALL STRS
713 000423' 261 17 0 00 000404' PUSH SP,STR ;SAVE THIS
714 000424' 006 10 0 02 000000 MAINLP: P 10,(2) ;SET CURRENT PACK
715 000425' 254 00 0 00 000432' JRST EOF+1 ;NOT THERE..STOP
716 000426' 007 10 0 00 000414' L 10,LOOK ;LOOKUP THAT FILE
717 000427' 254 00 0 00 000431' JRST EOF ;WASN'T THERE
718 000430' 264 00 0 00 000435' JSR COPY ;DUMP FILE
719 000431' 253 02 0 00 000424' EOF: AOBJN 2,MAINLP ;NEXT FILE FOR DSKS TIMES
720 000432' 071 01 0 00 000000 LPT ;CLOSE LPT FILE
721 000433' 262 17 0 00 000404' POP SP,STR ;RESTORE
722 000434' 254 00 0 00 000017* JRST DDT ;RETURN
723 000435' 000000 000000 COPY: 0 ;JSR HERE TO COPY FILE
724 000436' 201 11 0 00 000000 MOVEI 11,0 ;THIS AC HAS CURRENT BLOCK NUMBER
725 000437' 003 10 0 11 000000 RD 10,(11) ;READ AND DUMP IT
726 000440' 344 11 0 00 000437' AOJA 11,-1 ;LOOP
727 000441' 254 00 1 00 000435' JRST @COPY ;ON AN ERROR, RETURN
728 >
729 000442' ST: BLOCK 50 ;WRITE YOUR OWN PROGRAM IN 50 WORD OR LESS..
730
731 000512' 777640 000512' STACK: -140,,, ;STACK STUFF
732 000513' BLOCK 140
733
734 000653' STRLST: BLOCK 10 ;DATA AREA OF STR NAMES
735 000663' BLK0: BLOCK 200 ;BUFFERS
736 001063' BLK1: BLOCK 200
737 001263' BLK2: BLOCK 200
738 001463' BLK3: BLOCK 200
739 001663' BLK4: BLOCK 200
740 002063' BLK5: BLOCK 200
741 002263' BLK6: BLOCK 200
742 002463' BLK7: BLOCK 200
743 002663' BLK8:
744
745 000000' END DSK
  
```

NO ERRORS DETECTED

PROGRAM BREAK IS 002714  
 CPU TIME USED 00:01.794

3K CORE USED

B		000015	LH2		000205'	TYPOCR		000351'
BELL		000371'	LOOK		000414'	TYPOCT		000341'
BLK0		000663'	LOOKB		040000	TYPSEX		000325'
BLK1		001063'	LPT	071040	000000	U		000014
BLK2		001263'	MAINLP		000424'	U1		000046' SPD
BLK3		001463'	NMASET		000240'	U2		000066' SPD
BLK4		001663'	OUTCHR	051040	000000	UO		000013
BLK5		002063'	OUTSTR	051140	000000	USER		000001 SPD
BLK6		002263'	P	006000	000000	UUDH		000024'
BLK7		002463'	P1		000005	W		000016
BLK8		002663'	P2		000006	W	002000	000000
BLKTAB		000374'	P3		000007	WH		000075'
CL	012000	000000	PACK		000404'	ZY	013000	000000 SPD
CLH		000216'	PAUSE1		000212'	.JBHCU		000072
CMUDEC	047000	777776	PBITS		770000 SPD	.JBJDA		000075
COPY		000435'	PH		000132'	.JBREN		000005' EXT
CPOPJ		000223'	PH1		000143'	.JBSYM		000000 EXT
CPOPJ1		000222'	PH2		000156'	.JBUUO		000033' EXT
CRLF		000353'	PJRST	254000	000000			
CURBUF		000412'	PLOC		000005 SPD			
D		000011	PNT		000405'			
DDT		002664' EXT	PPTR		000407'			
DEVCHR	047000	000004	PROG		000421'			
DEVNAM	047000	000064	PRTFRM		000123'			
DING		000367'	PSIZ		000006 SPD			
DING1		000367'	PUUOAC		000410'			
DMPCHR		000323' EXT	R	001000	000000			
DMPFIL		000130' EXT	RD	003000	000000			
DMPHED		000265' INT	RDH		000105'			
DMPLIN		000267' EXT	READOP		000001 SPD			
DMPLUP		000312'	REN	011000	000000			
DMPQUT		000121' EXT	RENAD		000020'			
DSK		000000'	RENAD1		000041'			
DSKFLG		000000 SPD	RENH		000162'			
DSKS		000006 SPD	RH		000066'			
DUMP	004000	000000	RL	013000	000000			
DUMPH		000110'	RLH		000215'			
E	010000	000000	SETPNT		000355'			
EH		000164'	SLEEP	047000	000031			
ENTRB		020000	SP		000017			
EOF		000431'	ST		000442'			
F		000000	STACK		000512'			
FILE	005000	000000	STAT		000413'			
FILEH		000130'	STR		000404'			
FILUOO		000217'	STRLOP		000012'			
FIRSTR		000001 SPD	STRLST		000653'			
HDRI0		000002 SPD	SVDACO		000411'			
HUH		000370'	SYSSTR	047000	000046			
HUH1		000366'	T1		000001			
JDA		000075	T2		000002			
JOBHCU		000072	T3		000003			
JOBSYM		000000 EXT	T4		000004			
L	007000	000000	TRYIO		000224'			
LH		000165'	TYPHAF		000333'			
LH1		000176'	TYP01		000345'			



P1	85#	427	429	432	433	434								
P2	86#	373	378	379										
P3	87#													
PACK	693#													
PAUSE1	380	387#												
PBITS	104#	264	276											
PH	217	312#												
PH1	318	321#												
PH2	320	332#												
PLOC	103#	696												
PNT	424	544	545	694#										
PPTR	262	273	492	696#										
PROG	711#													
PRTFRM	296#													
PSIZ	102#	696												
PUUOAC	697#													
RDH	214	279#												
READOP	91#	259	270	413	418	422								
RENAD	161	176#												
RENAD1	179	204#												
RENH	220	361#												
RH	212	259#	279											
RLH	222	391#												
SETPNT	260	271	287	533#										
SP	88#	156	158	176	177	178	186	191	201	203	205	207	208	260
	271	279	287	304	308	330	333	374	383	395	399	400	420	425
	426	428	431	435	513	515	517	521	523	524	527	530	532	546
	548	550	553	713	721									
ST	729#													
STACK	158	176	731#											
STAT	432	701#												
STR	194	334	693#	713	721									
STRLOP	167#	171												
STRLST	163	170	312	407	493	734#								
SVDACO	699#													
T1	81#	161	162	199	200	201	288	289	322	326	328	369	370	371
	381	387	388	412	414	415	429	506	507	513	514	517	518	520
	522	524	525	526										
T2	82#	312	314	319	323	406	411	416	506	521				
T3	83#	316	317	319	323	324	325	327	507	508	509	510		
T4	84#													
TRYIO	265	277	404#											
TYPHAF	513#													
TYP01	522	524#												
TYP0CR	383	431	529#											
TYP0CT	515	520#	523	530										
TYP5IX	333	506#	511											
U	78#	194	329	365	404	415								
U1	200	201	210#											
U2	200	229#												
U0	77#	328	361	363	364	375	380	391	417	419	421	423	424	427
	434													
USER	61#	63	67	106	119	142	149	150	160	182	187	230	256	267





