

```

1      ;***COPYRIGHT 1969, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.***
2
3      ;THIS SUB-PROGRAM ASSEMBLED WITH SYSTEM PARAMETER FILE - S,MAC(V414)
4      XLIST
5      LIST
6      ;THIS SUB-PROGRAM ASSEMBLED WITH CONFIGURATION DEPENDENT FEATURE SWITCHES - FT50SB,MAC(
7      V003)
8      XLIST
9      LIST
10     IFN FTRA10, <
11     TITLE DSKSER -- DEVICE-INDEPENDENT DISK SERVICE ROUTINES,
12     >
13     IF FTRA10, <
14     IFN FTRC10, <
15     TITLE DSKSRB - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (BURROUGHS)
16     >>
17     IF FTRC10, <
18     TITLE DSKSRD - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (DATA PRODUCTS)
19     >
20
21     SUBTTL A, BLACKINGTON/CMF/TH/CHW/RCC/AF TS 02 JUN 69 V424
22     XP VDSKSR,424*
23
24     ;PUT VERSION NUMBER IN GLOB LISTING AND LOADER STORAGE MAP
25
26     INTERNAL FTRA10      ;TO ASSEMBLE DSKSER AS A
27                          ; DEVICE INDEPENDENT ROUTINE SET
28                          ; THIS SWITCH TO -1, NOTE THAT M0FINT (THE BRYANT
29                          ; DSKINT) WAS WRITTEN SO AS TO KEEP DSKSER DEVICE-INDEPENDENT
30                          ; THIS SWITCH AND CODE ENCLOSED IN ITS IFE'S WILL ULTIMATELY
31                          ; BE REMOVED FROM DSKSER.
32
33
34     INTERNAL      FTRC10      ;TO ASSEMBLE DSKSER FOR THE PDP-10 DISK
35                          ; (MODEL RC-10), SET THIS SWITCH TO -1.
36                          ; FOR OLD (270) DISK SET IT = 0.
37
38     IFN FTRA10, <
39     ENTRY DSKSER
40     DSKSER:
41     >
42     IF FTRA10, <
43     IFN FTRC10, <
44     ENTRY DSKSRB
45     DSKSRB:
46     >
47     IF FTRC10, <
48     ENTRY DSKSRD
49     DSKSRD:
50     >>
51
52     EXTERNAL JIFMIN
53
54     INTERNAL WLBIT,DSKFGS,CHKCNT
55
56     000012 TA=BUFPNT      ;USED BY RENAME

```

57
58
59
60
61

000013 TB=BUFWRD
IFE FTRA10, <
INTERNAL MFDBLK,SATXWD
IFN FTRC10, <

USED BY RENAME

DSKSRB - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (BURROUGHS) MACRO,V36 19:05 4-JUN-69 PAGE 13-1
A. BLACKINGTON/CMF/TH/CHW/RCC/AF TS 02 JUN 69 V424

```

62 000000 000000 000066 MFDBLK: 66
63 000001 777774 000067 SATXWD: XWD -NUMSAT,67
64
65 >
66 IFE FTRC10, <
67 MFDBLK: 13000
68 SATXWD: XWD -NUMSAT,13001
69
70 >>

```

ION PDP-10 DISK, MFD RETRIEVAL INFORMATION AND SAT
BLOCKS GO AT THESE ABSOLUTE ADDRESSES
UNLESS USER SPECIFIES OTHERWISE DURING "ONCE".

THIS BLOCK CONTAINS RETRIEVAL INFO. OF MFD
LH=NUMBER OF SAT BLOCKS,
RH=FIRST SAT BLOCK NUMBER.

```

71      000200  CHKCNT=200      ;NUMBER OF WORDS TO CHECK-SUM
72      400000  W8BIT=400000    ;WAIT INDICATION FOR DDB
73
74      ; BITS IN LH OF IOS
75
76      000100  NMP=100         ;NO MORE INPUT POINTERS (LH OF IOS)
77      000200  UBFS=200        ;USING BUFFER IN FREE STORAGE (LH OF IOS)
78      000400  NORELB=400      ;DO NOT RELOCATE BUFFER ON I/O (LH IOS)
79      001000  UBFU=1000       ;USING BUFFER IN USER AREA
80      002000  NCTRLC=2000     ;DISABLE *C DURING CRUCIAL SEQUENCES OF NON-
81                                     ; INTERRUPTABLE OPERATIONS
82
83      004000  DAFLG=4000
84      010000  AUFLG=10000
85      016200  DSKFGS=AUFLG+DAFLG+UBFS+NCTRLC ;USED TO SUPPRESS ACTION OF *C, SEE STOPCK
86      020000  PRCHG=20000     ;PROTECTION CHANGED
87      000040  LIR=IOEND       ;LAST INPUT REQUESTED
88
89      ;MISCELLANEOUS CONSTANTS
90      000003  TRIES=3         ;NUMBER OF TIMES TO READ OR WRITE BAD BLOCK
91      000200  BLKSIZ=200      ;SIZE OF DATA BLOCK, DUMP MODE REQUIRES THIS
92      777600  NBLKSZ=-BLKSIZ ;TO BE A POWER OF 2,
93      000007  BLKP2=7        ;BLKSIZ = 2*BLKP2
94
95      ;TEST BITS PUT IN LH OF DEVOAD
96
97      200000  PNTDIF=200000   ;POINTERS IN CORE DO NOT MATCH POINTERS ON DISC
98      100000  VRGPTR=100000   ;FLAG TO SIGNAL THAT POINTERS HAVE NEVER BEEN WRITTEN
99      020000  RENBIT=20000    ;"RENAMING" INDICATION FOR SETLE ROUTINE
100     040000  WPRO=40000      ;LOOKUP DONE, FILE IS IN WRITE PROTECT
101     010000  CKSUPH=10000     ;FLAG SET IF READING DUMP FILES IN NON-DUMP MODE
102
103

```

```

104          )CONSTANTS USED IN CONNECTION WITH ACCESS TABLE ENTRIES
105
106          000000  ATPP=0
107          000001  ATNAME=1
108          000002  ATEXT=2
109          000002  TBITS=2
110          000003  ATBLOK=3
111          000003  ATLINK=3
112
113          400000  WTBIT=400000
114          200000  DTBIT=200000
115          100000  RTBIT=100000
116          500000  ATBITS=500000
117          040000  ATIND=40000
118          020000  ATCLO=20000
119          010000  RWTBIT=10000
120          007777  RCOUNT=7777
121
122          )CONSTANTS USED IN CONNECTION WITH SAT BLOCKS
123
124          XP NUMBIT,*D36*BLKSIZ
125
126          IFE FTRA10, <
127          IFN      FTRC10, <
128          XP      NUMSAT,4
129          >
130          IFE      FTRC10, <
131          XP NUMSAT,16
132
133          >>
134          XP WLBIT,400000
135
          )PROJECT,PROGRAMMER NUMBERS
          )OWNERS NAME
          )EXTENSION
          )TEST BITS AND RCOUNT
          )BLOCK NUMBER OF RETRIEVAL INFO OF UFD IN LH
          )LINK TO NEXT ENTRY IN RH
          )FILE IS BEING CREATED
          )DELETE WHEN THRU READING
          )FILE HAS BEEN RENAMED
          )WTBIT+RTBIT
          )NAME IS ALREADY IN DIRECTORY
          )OUTPUT CLOSE TO BE DONE
          )FILE BEING READ AND WRITTEN, BUT NOT CREATED,
          )MASK FOR THE COUNT OF READERS
          )NUMBER OF BITS IN ONE BLOCK
          )ONLY FOUR SAT BLOCKS ON PDP-10 DISK.
          )NUMBER OF SAT BLOCKS FOR 5.76 MILLION
          )WORD DATA PRODUCTS MODEL 270 DISK.
          )IF BIT IS ON IN FIRST SATENT ENTRY
          )WORD, THE DISK IS WRITE-LOCKED.

```

```
136          EXTERNAL DSKOFF
137
138          DEFINE NOSCHEDULE <
139              CONO PI,DSKOFF
140          >
141
142          EXTERNAL DSKON
143          DEFINE SCHEDULE <
144              CONO PI,DSKON
145          >
146
```

```

147      INTERNAL FTCHECK,FTMONP
148      IFN FTCHECK+FTMONP,<
149      EXTERNAL DDBPTR,DSKCOR,PTRN,MOPTR,MIPTR,RUNUSR,SAVPRG,USRCNT
150      EXTERNAL SATPTR,SAT,SATBK2,DIRSZ
151      EXTERNAL DSKDOB,DEVACC,DEVcnt,DEVBLK,DEVPRG,DSKBUF
152      EXTERNAL DSKCNT,SETCNT,PTR1,FPNTR,FPNTR1,FAT,DFBUSY,LOCORE,GTGOR3
153      EXTERNAL GTCOR2,CORBIT,CKSMCT,DSKSIZ,DSKACC,DSKAPP,CORBSZ,MONBUF
154      >
155      IFE FTCHECK+FTMONP,<
156
157      ;DISK DEVICE DATA BLOCKS & MISCELLANEOUS CONSTANTS, WORKING STORAGE
158      000002 000000 000000 DDBPTR: 2 ;LOCATION OF NEXT DDB TO LOOK AT
159      ;DURING DISK INTERRUPT
160
161      ;MORE MISCELLANEOUS FOR DISK
162
163      INTERNAL SATPTR,SAT,SATBK2,REFLAG,DFBUSY,CKSMCT
164      MOPTR: BLOCK 1 ;POINTER TO NEXT TASK TO DO
165      MIPTR: BLOCK 1 ;POINTER TO NEXT FREE QUEUE ENTRY
166
167      RUNUSR: BLOCK 1 ;LH=DEVDAT OF USER JOB RUNNING
168      ;RH= UNUSED
169      USRCNT: BLOCK 1 ;COUNT OF WAITING USER JOBS
170      DFBUSY: 0 ;-1 IF DISK CONTROL IN USE, 0 IF FREE
171      FAT: 0 ;POINTER TO FIRST ACCESS TABLE ENTRY
172      CKSMCT: 0 ;COUNT OF VARIOUS CHECKSUM ERRORS
173      REFLAG: 0 ;CONTAINS A PPN TO ALLOW LOGIN AFTER REFRESH
174      UXFERS: 0 ;TOTAL USER TRANSFERS ATTEMPTED
175      ECOUNT: 0 ;TOTAL HARDWARE ERRORS ON USER TRANSFERS
176      ;DISK STORAGE ALLOCATION CONTROL
177
178      SATPTR: BLOCK 1 ;POINTER TO CURRENT SATENT ENTRY
179      SATPIK: BLOCK 1 ;POINTER TO NON-FULL SAT ENTRY
180      SATCHG: BLOCK 1 ;0 IF SAT BLOCK IN CORE=BLOCK ON DISK, -1 OTHERWISE,
181      SATTEM: BLOCK 1 ;TEMPORARY SAVE LOCATIONS USED DURING THE SAT READ
182      SATTMP: BLOCK 1 ;ROUTINE TO AVOID PUSH-DOWN LIST OVERFLOW,
183
184      IFN FTRA10, <
185      EXTERNAL MFDBLK,SATXWD,NUMBLK,SATENT,SENTSZ,SATTOP
186      >
187      IFE FTRA10, <
188      INTERNAL SATENT
189
190      IFN FTRC10, <
191      XP NUMBLK,*04000 ;NUMBER OF DATA BLOCKS PER SAT BLOCK ON THE
192      ; PDP-10 MODEL RC-10 DISK FILE,
193      >
194      IFE FTRC10, <
195      XP NUMBLK,5400 ;NUMBER OF DATA BLOCKS REPRESENTED BY
196      ;ONE SAT BLOCK ON THE DATA PRODUCTS 270 DISK.
197      >
198      XP SENTSZ,3 ;SIZE OF EACH SATENT ENTRY
199      IFN FTRC10, <
200      DSKDOB=0 ;ON PDP-10 DISK FILE STORAGE BEGINS IN LOGICAL

```

DSKSRB - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (BURROUGHS) MACRO,V36 19:05 4-JUN-69 PAGE 17-1
A. BLACKINGTON/CMF/TH/CHW/RCC/AF TS 02 JUN 69 V424

```
200                                     ; BLOCK NO. 0,  
201                                     >  
202                                    IFE        FTRC10, <  
203                                   DSKXDB=5400  
204                                     ;FILE STORAGE BEGINS ON SECOND DISK OF DID (MOD 270)  
205                                     ; DISK FILE SINCE FIRST DISK IS ALLOCATED  
206                                     ; TO SWAPPING,  
                                     >
```

```

207
208          SATENT: REPEAT  NUMSAT,<
209          XWD  DSKXDB,0      ;RH CONTAINS WLBIT AND COUNT OF BLOCK USED
210          BLOCK 1            ;BIT MASK, SINGLE ROTATING BIT
211          BLOCK 1            ;XWD LENGTH OF SAT TABLE POINTER TO WORD
212                                ; IN SAT BLOCK
213          DSKXDB=DSKXDB+NUMBLK>
214 000022 000000 000000      XWD  DSKXDB,0      ;RH CONTAINS WLBIT AND COUNT OF BLOCK USED
215 000023                                ;BIT MASK, SINGLE ROTATING BIT
216 000024                                ;XWD LENGTH OF SAT TABLE POINTER TO WORD
217                                ; IN SAT BLOCK
218                                007640
219          DSKXDB=DSKXDB+NUMBLK
220 000025 007640 000000      XWD  DSKXDB,0      ;RH CONTAINS WLBIT AND COUNT OF BLOCK USED
221 000026                                ;BIT MASK, SINGLE ROTATING BIT
222 000027                                ;XWD LENGTH OF SAT TABLE POINTER TO WORD
223                                ; IN SAT BLOCK
224                                017500
225          DSKXDB=DSKXDB+NUMBLK
226 000030 017500 000000      XWD  DSKXDB,0      ;RH CONTAINS WLBIT AND COUNT OF BLOCK USED
227 000031                                ;BIT MASK, SINGLE ROTATING BIT
228 000032                                ;XWD LENGTH OF SAT TABLE POINTER TO WORD
229                                ; IN SAT BLOCK
230                                027340
231          DSKXDB=DSKXDB+NUMBLK
232 000033 027340 000000      XWD  DSKXDB,0      ;RH CONTAINS WLBIT AND COUNT OF BLOCK USED
233 000034                                ;BIT MASK, SINGLE ROTATING BIT
234 000035                                ;XWD LENGTH OF SAT TABLE POINTER TO WORD
235                                ; IN SAT BLOCK
236                                037200
237          DSKXDB=DSKXDB+NUMBLK
238          XP      SATTOP,SATENT+SENTSZ*NUMSAT-SENTSZ+
239
240          >      ;END OF FTRA10 CONDITIONAL
241
242
243 000036          SAT:  BLOCK  200      ;CURRENT SAT BLOCK
244
245          XP      SATM2,SAT-2      ;LOWEST ABS. ADR-1 ALLOWED IN CHANNEL COMMAND
246                                ; LIST - USED TO KEEP FROM WIPING OUT MONITOR.
247 000236 777600 000036: SATBK2: XWD  -200,SAT

```

```

248          XP      DIRSIZ,4          ;NO, OF WORDS OF RET. INFO WHICH ARE
249                                     ;NOT POINTERS,
250
251          ;THE DUMMY DEVICE DATA BLOCK
252          000010 DSKCOR=10
253          000040 DSKSIZ=4*DSKCOR          ;NUMBER OF WORDS ALLOCATED FOR DDB
254                                     ;MUST BE A MULTIPLE OF FOUR (SEE GETCOR)
255
256          INTERN  DSKDDB
257          DSKDDB: SIXBIT  /DSK/
258          XWD      +D5*HUNGST,201
259          Z
260          EXP      DSKDSP
261          XWD      DVOUT+DVIN+DVDIR+DVDSK+DVLNG,154403
262          Z
263          XWD      PROG,
264          XWD      PROG,
265          XP      DEVFIL,.-DSKDDB ;FILE NAME
266          Z          ;SIXBIT/FILE? - FILE NAME
267          XP      DEVEXT,.-DSKDDB ;FILE EXTENSION
268          Z          ;XWD SIXBIT/EXT/,BLOCK# OF THE RIB
269          XP      DEVPPN,.-DSKDDB ;PROJ,PROGRAMMER FOR CURRENT
270          Z          ; (OR MOST RECENT) LOOKUP,ENTER,RENAME
271          000252 000000 000000      Z
272          XP      DEVACC,.-DSKDDB ;C(LH)=LOCATION OF CCESS TABLE ENTRY
273          Z          ;C(LH)=ADDRESS OF ACCESS TABLE ENTRY
274          XP      DEVCNT,.-DSKDDB ;C(RH)=ADDRESS OF CURRENT POINTER IN DDB
275          Z          ;C(LH)USED DURING LOOKUP,ENTER
276          000254 000000 000000      Z          ;C(LH)=RELATIVE BLOCK# WITHIN UFD OPTIMIZE UFD SEARCHES
277          000252 000000 000000      Z
278          XP      DEVBLK,.-DSKDDB ;C(RH)=COUNT OF BLOCKS IN FILE
279          Z          ;C(LH)=BLOCK NUMBER OF CURRENT POINTER
280          000255 000000 000000      Z          ;C(LH)=LOGICAL BLOCK# OF RIB
281          Z          ;C(RH)=RELATIVE LOC. OF NEXT
282          Z          ;POINTER IN THAT BLOCK
283          XP      DEVBK0,.-DSKDDB ;LIKE DEVBLK, BUT FOR BEGINNING
284          Z          ;SIMILAR TO DEVBLK - SAYS WHERE CURRENT
285          000256 000000 000000      Z          ;POINTERS COME FROM
286          XP      DSKBUF,.-DSKDDB ;C(LH)=-SIZE OF MONITOR READ/WRITE
287          Z          ;C(LH)=-SIZE OF READ/WRITE
288          000257 000000 000000      Z          ;C(RH)=BUFFER IN USER AREA
289          XP      DSKCNT,.-DSKDDB ;C(LH)=BLOCK NUMBER TO READ/WRITE
290          Z          ;C(LH)=LOGICAL BLOCK# FOR READ OR WRITE
291          000260 000000 000000      Z          ;C(RH)=ERROR COUNT OR ERROR BITS
292          XP      SETCNT,.-DSKDDB ;LH CONTAINS POINTER TO R,I, OF
293          Z          ;C(LH)=BLOCK# OF UFD RIB
294          000261 000000 000000      Z          ;UFD, RH IS SET0, SET1 COUNTER.
295          XP      PTR1,.-DSKDDB ;RETRIEVAL INFO STARTS HERE
296          Z          ; DEFINE THE AREA CONTAINING RETRIEVAL POINTERS
297          XP      FPNTR,PTR1+DIRSIZ          ;LOC. OF FIRST POINTER TO RET. INFO,
298          XP      FPNTR1,FPNTR-1*
299          XP      PTRN,DSKSIZ-1*
300

```

```

301          ;DEFINE 3 WORD ENTRY CONTROLLING BIT SEARCHING IN DDBTAB
302          INTERNAL LOCORE,CRINIT
303          EXTERNAL DDBTAB
304
305 000262 000000 000000 LOCORE: EXP 0          ;ADDRESS OF FIRST 4 WORD BLOCK IN FREE CORE AREA
306                                     ; SET BY LINKDB
307 000263 000000 000000 CORBIT: 0             ;FLOATING 1 BIT USED IN SEARCHES
308 000264 000000 000000 CORIWD: XWD 0,0       ;A0BJN POINTER TO FIRST WORD IN DDBTAB CONTAINING A 0
309 000265 000000 000000 CRINIT: XWD 0,DDBTAB  ;INITIAL A0BJN POINTER FOR CORIWD
310                                     ; LH SET BY LINKDB
311 MONBUF: BLOCK 200                         ;MONITOR BUFFER, USED TO READ RETRIEVAL POINTERS, ETC.,
312
313 >

```

```

314                                ;ERROR CODES FOR LOOKUP, RENAME AND/OR ENTER
315
316                                000000 NOTINU=0                                ;NO SUCH FILE
317                                000001 NOTINM=1                             ;NO SUCH USER
318                                000002 PROT=2                               ;PROTECTION FAILURE
319                                000003 NORITE=3                             ;MORE THAN 1 WRITE TO A FILE
320                                000004 RENFAL=4                             ;TRIED TO RENAME FILE TO EXISTING NAME OR ENTER
321                                ; A NULL FILE NAME
322                                000005 NOFILE=5                             ;TRIED TO RENAME WITH NO FILE SELECTED
323                                000001 ERBIT=1                             ;BITS GO IN WORD 1 OF THE ENTRY
324
325
326                                ;DISK DISPATCH TABLE
327
328                                INTERNAL DSKDSP
329
330                                000466 254000 000524' JRST DSKINI                ;INITILIZE
331                                000467 254000 000000' JRST CPOPJ1             ;DSK HUNG TIMEOUT, NO ERROR MESSAGE
332                                000470 254000 003355' DSKDSP: JRST DFREL        ;RELEASE
333                                000471 254000 001453' JRST DFCLSO             ;CLOSE OUTPUT
334                                000472 254000 003073' JRST DFOUT              ;OUTPUT
335                                000473 254000 002757' JRST DFIN               ;INPUT
336                                000474 254000 000625' JRST DFENTR             ;ENTER
337                                000475 254000 000766' JRST DFLOOK             ;LOOKUP
338                                000476 254000 002215' JRST DFDMP0             ;DUMP OUTPUT
339                                000477 254000 002431' JRST DFDMPI             ;DUMP INPUT
340                                000500 254000 002607' JRST DFSET              ;USET0
341                                000501 254000 002607' JRST DFSET              ;USETI
342                                000502 263140 000000' POPJ PDP,              ;UGETF
343                                000503 254000 001103' JRST DFREN              ;RENAME
344                                000504 254000 001601' JRST DFCLSI             ;CLOSE INPUT
345                                000505 263140 000000' POPJ PDP,              ;UTPCLR
346                                000506 263140 000000' POPJ PDP,              ;MTAPE

```

```

347
348
349      ;ACCINI - CALLED AT 140 START, 143 RESTART
350      ; RESTORE DEVICE DATA BLOCK CHAIN
351      ; INITIALIZE ODBTAB BIT SEARCHING ENTRY
352
353      INTERNAL ACCINI
354      000507 205040 446353 ACCINI: MOVSJ TAC,(SIXBIT /DSK/)
355      000510 201300 000237' MOVEI DEVDAT,DSKDOB ;POINT TO PROTOTYPE DDB
356      000511 554306 000003 ACCINI: HLRZ DEVDAT,DEVSER(DEV DAT)
357      000512 322300 000515' JUMPE DEVDAT,ACCIN2
358      000513 316046 000000 CAMN TAC,DEVNAM(DEV DAT)
359      000514 254000 000511' JRST ACCINI
360      000515 201040 000237' ACCIN2: MOVEI TAC,DSKDOB
361      000516 506301 000003 HRLM DEVDAT,DEVSER(TAC) ;LINK PROTOTYPE TO FIRST NON DSK DDB
362      000517 205040 400000 MOVSJ TAC,400000
363      000520 202040 000263' MOVEM TAC,CORBIT ;INITIALIZE FLOATING 1 BIT
364      000521 200040 000265' MOVE TAC,CRINIT
365      000522 202040 000264' MOVEM TAC,CORIND ;RESET AOBJN WORD
366      000523 263140 000000 POPJ PDP,
367
368
369      ;DSKINI - CALLED AT 140 START, RESTART, 143 RESTART
370      ; REMOVE ACCESS ENTRIES, REMOVE DDB'S NOT ASSIGNED BY CONSOLE. CLEAR THE MONITOR QUE
371      E.
372      INTERNAL DSKINI, SETSAT
373      EXTERNAL DISKUP,DFWUNS
374
375      000524
376      000524 402000 000000 DSKINI: IFN FTSWAP, <
377      000525 402000 000000 SETZM SQREQ ;NO SWAPPING REQUEST
378      >
379      IFE FTRA10, <
380      EXTERNAL DSKCON
381      000526 513000 000000 HLLZS DSKCON ;CLEAR INTERRUPT FLAGS FOR CONSO
382      >
383      000527 402000 000007' SETZM DFBUSY ;DISK CONTROL AVAILABLE
384      000530 402000 000005' SETZM RUNUSR ;NO USER I/O IN PROGRESS
385      000531 476000 000006' SETOM USRCNT ;NO USER TRANSFERS WAITING
386      000532 403000 000002' SETZM DDBPTR ;NO NEXT DDB
387      ; CLEAR IOS (SUPERSTITIOUS, BUT POSSIBLY NECESSARY)
388      000533 201300 000237' MOVEI DEVDAT,DSKDOB ;PREPARE TO RELEASE DISK DDB'S
389      000534 554306 000003 DSKINI: HLRZ DEVDAT,DEVSER(DEV DAT) ;GET DDB LINK
390      000535 322300 000553' JUMPE DEVDAT,DSKINI ;JUMP IF END REACHED
391      000536 205040 446353 MOVSJ TAC, (SIXBIT /DSK/) ;DEVICE NAME
392      000537 312046 000000 CAMN TAC,DEVNAM(DEV DAT) ;IS THIS A DISK DDB?
393      000540 254000 000553' JRST DSKINI ;NO, DONE
394      000541 554046 000014 HLRZ TAC, DEVACC(DEV DAT) ;GET POINTER TO ACCESS ENTRY
395      000542 332000 000001 SKIPE TAC ;ACCESS ENTRY EXIST?
396      000543 260140 002163' PUSHJ PDP,CLRAT ;YES, REMOVE IT
397      000544 205040 400000 MOVSJ TAC,W8BIT ;PREPARE TO CLEAR "TRANSFER WAITING" BIT
398      000545 412046 000007' ANDCAM TAC,DEVIAD(DEV DAT) ;CLEAR INPUT WAITING
399      000546 412046 000010 ANDCAM TAC,DEVODD(DEV DAT) ;CLEAR OUTPUT WAITING

```

DSKSRB - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (BURROUGHS) MACRO,V36 19:05 4-JUN-69 PAGE 22-1
A. BLACKINGTON/CMF/TH/CHW/RCC/AF TS 02 JUN 69 V424

400	000547	200046	000004	MOVE TAC,DEVMOD(DEV DAT)	
401	000550	606040	400000	TRNN TAC,ASSCON	!DDB ASSIGNED BY ASSIGN COMMAND?
402	000551	260140	003555'	PUSHJ PDP,CLR DDB	!NO. REMOVE DDB
403	000552	254000	000534'	JRST DSKIN0	!CONTINUE
404					
405	000553	402000	000010'	DSKIN1: SETZM FAT	!NO ACCESS ENTRIES
406	000554	201040	000001	MOVEI TAC,MQUEUE+1	!DESTINATION FOR BLT
407	000555	202040	000004'	MOVEM TAC,MIPTR	
408	000556	202040	000003'	MOVEM TAC,MOPTR	!RESET IN AND OUT POINTERS
409	000557	505040	000000	HRLI TAC,MQUEUE	
410	000560	402000	000557'	SETZM MQUEUE	!SOURCE FOR BLT
411	000561	251040	777777	BLT TAC,MQTOP-1	!CLEAR THE MONITOR QUEUE
412	000562	260140	000000	PUSHJ PDP,DISKUP	!DETERMINE IF DISK IS FUNCTIONING
413	000563	352000	000566'	SETSAT: AOSE STARTS	!SKIP FIRST TIME ONLY
414	000564	254000	000577'	JRST SAT01	!WRITE SAT BLOCK IN CORE
415	000565	254000	000000	JRST DFWUNS	!SCAN ALL SAT BLOCKS
416	000566	777777	777777	STARTS: -1	!COUNT STARTS, RESTARTS

```

417                                     ;WRITE OUT SAT BLOCK, ENTER AT DSKSTP WITH FIXED START 147,
418                                     ;AT SAT01 UPON RE-ENTRY OF INITIALIZATION,
419
420                                     INTERN DSKSTP,SAT05,JSAT06
421
422                                     EXTERNAL DSKRIT,WSYNC,STOIOS,DCBRIT
423
424 000567 551040 000000 DSKSTP: HRRZI TAC,DSKBIT
425 000570 435040 000000 IORI TAC,DCBBIT ;DCBBIT NON-ZERO ONLY FOR DATA-PRODUCTS n$IK
426 000571 435040 002200 IORI TAC,2200 ;CLEAR PI SYSTEM
427 000572 700601 010000 CONO PI,10000(TAC)
428 000573 331000 000566' SKIPL STARTS ;HAS A SAT BLOCK BEEN READ IN?
429 000574 260140 000577' PUSHJ PDP,SAT01 ;YES, WRITE IT OUT
430 000575 700600 010000 CONO PI,10000
431 000576 263140 000000 POPJ PDP, ;YOU ARE SAFELY DONE.
432
433 000577 201300 000237' SAT01: MOVEI DEVDAT,DSKDDB
434 000600 403006 000002 SETZR IOS,DEVIOS(DEV DAT)
435 000601 201340 000000 MOVEI PROG,0
436 000602 261140 000000 PUSH PDP,WSYNC ;CHANGE WSYNC ROUTINE
437 000603 200040 000624' MOVE TAC,JSAT06
438 000604 202040 000602' MOVEM TAC,WSYNC
439 000605 201200 000003 SAT05: MOVEI ITEM,TRIES ;SET UP COUNT FOR ERRORS
440 000606 260140 003546' SAT02: PUSHJ PDP,SATBLK ;SET UP TAC,TAC1
441 000607 260140 004112' PUSHJ PDP,MQOUT ;WRITE IT,
442 000610 254000 000615' JRST SAT04 ;ERROR
443 000611 402000 000017' SETZM SATCHG ;SHOW SAT BLOCK IN CORE = BLOCK ON DISK.
444
445 000612 262140 000604' POP PDP,WSYNC ;NO ERRORS, SO RETURN.
446 000613 402006 000002 SETZM DEVIOS(DEV DAT)
447 000614 263140 000000 POPJ PDP,
448
449 000615 367200 000606' SAT04: SOJG ITEM,SAT02 ;ERRORS, SO TRY AGAIN
450 000616 262140 000612' POP PDP,WSYNC ;RESTORE WSYNC ROUTINE
451 000617 254200 000577' JRST 4,SAT01
452
453 000620 201040 010000 SAT06: MOVEI TAC,IOACT ;THIS IS SUBSTITUE FOR WSYNC
454 000621 612046 000002 TDNE TAC,DEVIOS(DEV DAT)
455 000622 254000 000621' JRST ,-1
456 000623 263140 000000 POPJ PDP,
457
458 000624 254000 000620' JSAT06: JRST SAT06

```

```

459          SUBTTL ENTER, LOOKUP, RENAME, CLOSE
460          ;ENTER UUO
461
462          EXTERNAL TIME,PRJPRG,CPOPJ1,THSDAT
463
464
465 000625 336020 000014 DFENTR: SKIPN @UUO          ;ZERO FILE NAME ?
466 000626 254000 004440' JRST DFERR12
467 000627 260140 001071' PUSHJ PDP,ALTMFD          ;ENTER TO MFD?
468 000630 603300 040000' TLNE DEVDAT,LOOKR          ;NO, HAS LOOKUP BEEN DONE?
469 000631 254000 000736' JRST DFENT5              ;YES
470 000632 661000 000020' TLO IOS,IO
471 000633 260140 001340' PUSHJ PDP,SETLE          ;RETURNS WITH PP ON PDL
472 000634 254000 004402' JRST DFERR4              ;NO UFD, CLOCK ON
473 000635 254000 000651' JRST DFENT2              ;NO FILE, CLOCK ON.
474 000636 254000 004372' JRST DFERR6              ;FILE BEING WRITTEN
475 000637 254000 000763' JRST DFENT7              ;FILE BEING READ,CHECK FOR WRITERS
476
477 000640          DFENT1: SCHEDULE*
478 000640 700600 000000' CONO PI,DSKON
479 000641 550046 000012' HRRZ TAC,DEVEXT(DEV DAT) ;BLOCK# OF FIRST RIB
480 000642 260140 003316' PUSHJ PDP,SETPTR          ;READ FIRST RETRIEVAL POINTERS TO DOB
481 000643 205240 100000' MOVSJ DAT,100000          ;WRITE PROTECTION BIT
482 000644 260140 001423' PUSHJ PDP,PROTEK          ;WRITE PROTECTED?
483 000645 254000 004360' JRST DFERR3              ;NAME WRONG
484 000646 254000 004400' JRST DFERR5              ;PROTECTION FAILURE.
485 000647 201100 460000' MOVEI TAC1,WTBIT+ATIND*ATCLO ;BITS FOR ACCESS ENTRY-FILE EXISTS
486 000650 625000 020000' TLZA IOS,PRCHG          ;PREPARE FOR PROTECTION CHECK
487 000651 201100 420000' DFENT2: MOVEI TAC1,WTBIT+ATCLO ;BITS FOR ACCESS ENTRY - FILE CREATION
488 000652 513006 000012' HLLZS DEVEXT(DEV DAT)          ;SET TO 0 RIB BLOCK#
489          NOSCHEDULE*
490 000653 700600 000000' CONO PI,DSKOFF
491 000654 260140 002143' PUSHJ PDP,SCANAT          ;SCAN ACCESS TABLE FROM BEGINNING
492 000655 254000 000664' JRST DFENT3              ;NOT THERE

```

```

493 000656 200241 000002 DFEN2A: MOVE DAT,TBITS(TAC) ;WRITE BIT ON?
494 000657 602240 400000 TRNE DAT,WTBIT
495 000660 254000 004372' JRST DFERR6 ;YES,ERROR
496 000661 260140 002157' PUSHJ PDP,SCNAT2
497 000662 254000 000664' JRST DFENT3
498 000663 254000 000656' JRST DFEN2A
499
500 000664 623000 020000 DFENT3: TLZE IOS,PRCHG ;PROTECTION FAILURE?
501 000665 254000 004400' JRST DFERR5 ;YES, GIVE ERROR RETURN,
502 SCHEDULE+
503 000666 700600 000640' CONO PI,DSKON
504 000667 261140 000002 PUSH PDP,TAC1 ;SAVE ACCESS BITS
505 000670 260140 003371' PUSHJ PDP,DFGETF ;GET A BLOCK FOR RETRIEVAL POINTERS. # IN TAC
506 000671 262140 000002 POP PDP,TAC1 ;RESTORE ACCESS BITS
507 000672 516046 000016 HRLZM TAC,DEVBLK(DEV DAT) ;SET CURRENT RIB# AND INDEX IN DDB
508 000673 516046 000017 HRLZM TAC,DEVBKO(DEV DAT)
509 000674 542046 000012 HRRM TAC,DEVEXT(DEV DAT) ;STORE FIRST RIB#
510 NOSCHEDULE+
511 000675 700600 000653' CONO PI,DSKOFF

512 000676 200240 000002 MOVE DAT,TAC1 ;ACCESS ENTRY BITS
513 000677 260140 002201' PUSHJ PDP,SETAT ;CREATE ACCESS ENTRY. RETURN POINTER IN TAC1
514 000700 506106 000014 HRLM TAC1,DEVACC(DEV DAT) ;STORE POINTER TO ACCESS ENTRY
515 000701 201246 000023 MOVEI DAT,PTR1(DEV DAT) ;SET UP RETRIEVAL INFO
516 000702 200040 000005 MOVE TAC,DAT ;DESTINATION (DDB)
517 000703 505060 000014 HRLI TAC,0U00 ;SOURCE (U00 PAPAMETER BLOCK)
518 000704 251045 000002 RLT TAC,2(DAT) ;NAME & EXTENSION
519 SCHEDULE+
520 000705 700600 000666' CONO PI,DSKON
521 000706 200140 000000 MOVE TAC1,THSDAT
522 000707 435100 400000 ORI TAC1,400000 ;MAKE SURE ACCESS DATE NON-ZERO,
523 000710 542105 000001 HRRM TAC1,1(DAT) ;STORE THSDAT AS ACCESS DATE
524 000711 200105 000002 MOVE TAC1,2(DAT)
525 000712 621100 777740 TLZ TAC1,777740 ;IS THERE A DATE ALREADY?
526 000713 326170 000720' JUMPN TAC1,DFENT4 ;YES IF JUMP,
527 000714 200040 000000 MOVE TAC,TIME ;NO, GET TIME,
528 000715 231040 000000 IDIVI TAC,JIFMIN
529 000716 550100 000706' HRRZ TAC1,THSDAT ;AND DATE,
530 000717 137040 004577' DPB TAC,[POINT 11,TAC1,23]
531 000720 661100 055000 DFENT4: TLO TAC1,55000 ;PROTECTION
532 000721 137000 004600' DPR IOS,[POINT 4,TAC1,12] ;MODE
533 000722 202105 000002 MOVEM TAC1,2(DAT) ;STORE PROTECTION, MODE,TIME, DATE IN DDB
534 000723 200200 000000 MOVE ITEM,JOB
535 000724 200044 000000 MOVE TAC,PRJPRG(ITEM)
536 000725 202045 000003 MOVEM TAC,3(DAT) ;STORE PROGRAMMER# IN DDB
537 000726 402005 000004 SETZM 4(DAT) ;CLEAR FIRST POINTER
538 000727 205000 000100 MOVSI IOS,NMP ;SET "NO MORE POINTERS"
539 000730 437006 000002 ORB IOS,DEVIOS(DEV DAT)
540 000731 621300 001000 TLZ DEV DAT,OCLOS8
541 000732 513006 000015 HLLZS DEV CNT(DEV DAT) ;CLEAR FILE SIZE
542 000733 205040 100000 MOVSI TAC,VRGPTH ;SET "VIRGIN POINTER" FLAG (SO DFO4A SUBROUTINE
543 000734 436046 000010 IORM TAC,DEVQAD(DEV DAT) ;WILL CREATE A NEW RIB WHEN JUST CALLED
544 000735 254000 001044' JRST ALLOK

```

```

545                                     ;AN ENTER AFTER A LOOKUP. SET "RWTBIT" FLAG IN ACCESS ENTRY
546
547 000736 200046 000010 DFENT5: MOVE TAC,DEV0AD(DEV DAT) ;CHECK PROTECTION
548 000737 603040 040000 TLNE TAC,WPRO
549 000740 260140 004400' PUSHJ PDP,DFERR5 ;FAILURE, THIS WILL NOT RETURN HERE
550 000741 554046 000014 HLRZ TAC,DEVACC(DEV DAT) ;POINT TO ACCESS ENTRY
551 NOSCHEDULE+
552 000742 700600 000675' CONO PI,DSKOFF
553 000743 200101 000002 MOVE TAC1,TBITS(TAC) ;GET ACCESS BITS
554 000744 662100 010000 TROE TAC1,RWTBIT ;FILE BEING READ AND WRITTEN ALREADY?
555 000745 260140 004372' PUSHJ PDP,DFERR6 ;YES, THIS WILL NOT RETURN HERE
556 000746 202101 000002 MOVEM TAC1,TBITS(TAC) ;NO, SET IT
557 SCHEDULE+
558 000747 700600 000705' CONO PI,DSKON
559
560 000750 200060 000014 MOVE TAC,@UUO ;ISAME FILE?
561 000751 312046 000011 CAME TAC,DEVFIL(DEV DAT)
562 000752 254000 004400' JRST DFERR8 ;NO, NAME WRONG
563 000753 271600 000001 ADDI UUO,1
564 000754 510050 000014 HLLZ TAC,@UUO ;PICKUP EXTENSION FROM USERS ENTER BLOCK
565 000755 430046 000012 XOR TAC,DEVEXT(DEV DAT) ;COMPARE WITH LOOKUP EXTENSION(AND PICKUP
566 ; RETRIEVAL INFORMATION
567 000756 603040 777777 TLNE TAC,-1 ;EXTENSION THE SAME?
568 000757 254000 004400' JRST DFERR8 ;NO, EXTENSION WRONG
569 000760 370000 000014 SGS UUO ;YES, RETRIEVE LATEST RETRIEVAL INFORMATION AND
570 000761 260140 003316' PUSHJ PDP,SETPTR ;STORE CURRENT SIZE IN THE 4TH WORD OF THE 4-WORD
571 000762 260140 001025' PUSHJ PDP,DFLUK4 ;ENTER BLOCK (IN CASE SOME OTHER USER HAS APPENDED
572 ; DATA TO THIS FILE BETWEEN THE TIME THIS USER DID
573 ; HIS LOOK-UP AND THIS ENTER.) NOTE: THIS LAST
574 ; PUSHJ WILL EXIT TO UUOCON--NOT RETURN HERE!
575
576                                     ;FILE IS BEING READ
577
578 000763 260140 002157' DFENT7: PUSHJ PDP,SCNAT2 ;FIND ANOTHER
579 000764 254000 000640' JRST DFENT1 ;NO WRITERS, PERMIT THIS ENTER
580 000765 254000 004372' JRST DFERR6 ;MUST BE BEING WRITTEN OR RENAMED

```

```

581          ;LOOKUP UUU
582
583
584          EXTERNAL UDLKC,TPOPJ
585
586 000766 336020 000014 DFL00K: SKIPN @UUU          ;ZERO FILE NAME ?
587 000767 254000 004440' JRST DFER12          ;IF SO, ERROR,
588 000770 201040 000002 MOVEI TAC,CL5IN        ;SUPPRESS INPUT CLOSE IF UDLKC IS CALLED
589 000771 603300 020000 TLNE DEVDAT,ENTR8      ;ENTER DONE YET?
590 000772 260140 000000 PUSHJ PDP,UDLKC        ;CLOSE OUTPUT
591 000773 621000 000020 TLZ IOS,IO          ;INPUT STATE
592 000774 260140 001340' PUSHJ PDP,SETLE      ;RETURNS WITH PP ON PDL
593 000775 254000 004402' JRST DFERR4          ;NO UFD, CLOCK ON
594 000776 254000 004404' JRST DFERR7          ;NO FILE, CLOCK ON.
595 000777 254000 001066' JRST DFLUK8          ;FILE BEING WRITTEN
596 001000 254000 001004' JRST DFLUK3          ;FILE BEING READ
597
598 001001 201240 000000 DFLUK2: MOVEI DAT,0
599 001002 260140 002201' PUSHJ PDP,SETAT          ;BUILD AN ACCESS ENTRY
600 001003 200040 000002 MOVE TAC,TAC1          ;TAC,TAC1 POINT TO ACCESS ENTRY
601
602 001004 350001 000002 DFLUK3: AOS TBITS(TAC)      ;INCREMENT RCOUNT
603 001005 506046 000014 HRLM TAC,DEVACC(DEVDAT) ;STORE POINTER TO ACCESS ENTRY
604          SCHEDULE+
605 001006 700600 000747' CONO PI,OSKON
606 001007 200046 000012 MOVE TAC,DEVEXT(DEVDAT) ;FIRST RIB#
607 001010 260140 003316' PUSHJ PDP,SETPTR      ;GET IN THE RETRIEVAL INFO
608
609 001011 623000 020000 TLZE IOS,PRCHG          ;PROTECTION CHANGE FAILURE?
610 001012 254000 001064' JRST DFLUK7          ;YES, ERROR RETURN
611 001013 205240 200000 MOVSI DAT,200000      ;READ PROTECTION BIT
612 001014 260140 001423' PUSHJ PDP,PROTEK      ;READ PROTECTED?
613 001015 254000 004360' JRST DFERR3          ;NAME WRONG
614 001016 254000 001064' JRST DFLUK7          ;PROTECTION FAILURE
615
616 001017 200106 000010 MOVE TAC1,DEV0AD(DEVDAT) ;CHECK WRITE PROTECTION FOR
617 001020 205240 100000 MOVSI DAT,100000      ;POSSIBLE SURSEQUENT ENTER
618 001021 260140 001434' PUSHJ PDP,PROTKX
619 001022 665100 040000 TLOA TAC1,WPRO          ;WRITE PROTECTED, SET FLAG
620 001023 621100 040000 TLZ TAC1,WPRO          ;WRITE OK--RESET FLAG
621 001024 436106 000010 IORM TAC1,DEV0AD(DEVDAT)
622

```

623	001025	205046	000023	DFLUK4:	MOVSI TAC, PTR1(DEV DAT)	ICOPY INFO TO LOOKUP PARAMETER BLOCK
624					NOSCHEDULE*	
625	001026	700600	000742		CONO PI, DSKOFF	
626	001027	541060	000014		HRR I TAC, @UUO	
627	001030	200100	000001		MOVE TAC1, TAC	IDESTINATION IS UUO PARAMETER BLOCK
628	001031	251042	000002		RLT TAC, 2(TAC1)	
629					SCHEDULE*	
630	001032	700600	001006		CONO PI, DSKON	
631	001033	574046	000026		HLRE TAC, FPNT R1(DEV DAT)	IFILE LENGTH (USUALLY -WORDS)
632	001034	271600	000003		ADDI UUO, 3	
633	001035	516060	000014		HRLZM TAC, @UUO	ISSET UP 4TH WORD OF DIRECTORY HDR,
634	001036	275600	000003		SUBI UUO, 3	IRESET UUO
635	001037	217000	000001		MOVMS TAC	ISSET FILE SIZE (WORDS) IN DDB
636	001040	542046	000015		HRRM TAC, DEV CNT(DEV DAT)	
637	001041	621300	002000		TLZ DEV DAT, ICLOS B	
638	001042	205040	100000		MOVSI TAC, VRGPTR	IPPOINTERS DO EXIST ON THE DISK SO
639	001043	412046	000010		ANDCAM TAC, DEV DAD(DEV DAT)	ICLEAR THE VIRGIN POINTERS FLAG,

```

640 001044 201246 000027 ALLOK: MOVEI DAT,FPNTR(DEVDAT) ;SKIP FILE INFO IN FIRST RIB
641 001045 542246 000014 HRRM DAT,DEVACC(DEVDAT) ;SET VERTICAL POINTER POINTER
642 001046 201040 000001 MOVEI TAC,1
643 001047 542046 000022 HRRM TAC,SETCNT(DEVDAT) ;"POISITONED" AT RELATIVE BLOCK 1 OF FILE
644 001050 205040 637760 MOVSI TAC,777760-WPRO-VRGPTR ;CLEAR FLAGS
645 001051 412046 000010 ANDCAM TAC,DEVOAD(DEVDAT)
646
647 IFG CHKCN<, <
648 ;TO GET AROUND CHECKSUM FAILURE WHILE READING DUMP FILES IN NON-DUMP MODE
649
650 001052 135100 004601' LDB TAC1,[POINT 4,-2(DAT),12]
651 001053 305100 000015 CAIGE TAC1,15 ;IS FILE IN DUMP MODE?
652 001054 254000 001061' JRST ALLXIT ;NO
653 001055 135100 004602' LDB TAC1,[POINT 4,DEVIOS(DEVDAT),35]
654 001056 205040 010000 MOVSI TAC,CKSUPR
655 001057 305100 000015 CAIGE TAC1,15 ;IS CURRENT MODE DUMP?
656 001060 436046 000010 IORM TAC,DEVOAD(DEVDAT) ;NO, SUPPRESS CHECKSUM
657 >
658 001061 262140 000001 ALLXIT: POP PDP,TAC ;REMEMBER 0(PDP) WAS XWD PROJ,PROG
659 001062 350003 000000 AOS (PDP) ;RETURN TO CALL+2
660 001063 254000 003654' JRST CLRBUF ;RELEASE MONITOR BUFFER
661
662 001064 260140 001603' DFLUK7: PUSHJ PDP,DFCL21 ;PROTECTION FAILURE,CLEAR ACCESS
663 001065 254000 004400' JRST DFERR5
664
665 ;FILE IS BEING WRITTEN
666
667 001066 260140 002157' DFLUK8: PUSHJ PDP,SCNAT2 ;LOOK FOR ANOTHER
668 001067 254000 001001' JRST DFLUK2 ;NO MORE
669 001070 254000 001004' JRST DFLUK3 ;FOUND WHAT MUST BE READ
670
671 ;TEST FOR MFD ALTERATION
672
673 EXTERNAL SYSPP,JOB,PRJPRG
674
675 001071 201054 000003 ALTMFD: MOVEI TAC,3(UU0) ;IS PROJ. PROG. = 1,1?
676 001072 661040 000007 TLO TAC,PROG ;RELOCATE
677 001073 200060 000001 MOVE TAC,@TAC ;XWD PROJ,PROG
678 001074 312040 000000 CAME TAC,SYSPP
679 001075 263140 000000 POPJ PDP, ;NO,RETURN
680
681 001076 200200 000723' MOVE ITEM,JOB
682 001077 430044 000724' XOR TAC,PRJPRG(ITEM)
683 001100 607040 777777 TLNN TAC,-1 ;PRIVILEGED USER?
684 001101 263140 000000 POPJ PDP, ;YES, RETURN
685
686 001102 254000 004400' JRST DFERR5

```

```

687                                ;RENAME UUO
688
689                                EXTERNAL CPOPJ1
690                                EXTERNAL THSDAT
691 001103 260140 001106' DFREN: PUSHJ PDP,DFRENX
692 001104 263140 000000      POPJ PDP,
693 001105 254000 000467'      JRST CPOPJ1
694
695 001106 336006 000011 DFRENX: SKIPN DEVFIL(DEVDAT) ;IS THERE AN OLD FILE?
696 001107 254000 004436' JRST OFER11 ;NO, UNDEFINED FILE
697 001110 661000 000020      TLO IOS,IO ;YES, LOOK FOR THE FILE
698 001111 205040 020000      MOVSI TAC,RENBIT ;RESET BIT
699 001112 412046 000010      ANDCAM TAC,DEV0AD(DEVDAT)
700 001113 271600 000003      ADDI UUO,3 ;POINT TO XWD PROJ,PROG
701 001114 260140 001352' PUSHJ PDP,SETLE9 ;RETURNS WITH PP ON PDL
702 001115 254000 004402' JRST OFERR4 ;NO UFO, CLOCK ON
703 001116 254000 004404' JRST OFERR7 ;NO SUCH FILE, CLOCK ON,
704 001117 254000 004372' JRST OFERR6 ;FILE BEING WRITTEN
705 001120 254000 001124' JRST DFREN1 ;ALREADY HAS ACCESS ENTRY
706
707 001121 201240 000000      MOVEI DAT,0 ;BUILD ACCESS ENTRY
708 001122 260140 002201' PUSHJ PDP,SETAT
709 001123 200040 000002      MOVE TAC,TAC1 ;TAC,TAC1 POINT TO ACCESS ENTRY
710
711 001124 350001 000002 DFREN1: AOS TBITS(TAC) ;INCREMENT RCOUNT
712 001125 506046 000014      HRLM TAC,DEVACC(DEVDAT) ;SAVE POINTER TO ACCESS ENTRY
713                                SCHEDULE+
714 001126 700600 001032' CONO PI,DSKON
715
716 001127 550046 000012      HRRZ TAC,DEVEXT(DEVDAT) ;CAN THIS FILE BE RENAMED?
717 001130 260140 003316' PUSHJ PDP,SETPTR ;READ IN FIRST SET OF RETRIEVAL POINTERS
718 001131 201660 000014      MOVEI AC1,0UUO ;IS THE PROTECTION TO BE CHANGED ?
719 001132 201046 000023      MOVEI TAC,PTR1(DEVDAT) ;POINT TO RETRIEVAL INFO IN DOB
720 001133 200655 000002      MOVE AC1,2(AC1) ;USER SPECIFIED PROTECTION
721 001134 430641 000002      XOR AC1,2(TAC) ;COMPARE WITH STORED PROTECTION
722 001135 621000 020000      TLZ IOS,PRCHG
723 001136 607640 777000      TLNN AC1,777000 ;ANY DIFFERENCES?
724 001137 254000 001144' JRST DFRN2A
725 001140 205240 400000      MOVSI DAT,400000 ;YES, CAN IT BE?
726 001141 260140 001434' PUSHJ PDP,PROTKX
727 001142 254000 001310' JRST DFREN8 ;NO, ERROR
728 001143 661000 020000      TLO IOS,PRCHG ;YES
729 001144 205240 100000 DFREN2A: MOVSI DAT,100000 ;CAN THIS FILE BE RE-WRITTEN?
730 001145 260140 001423' PUSHJ PDP,PROTEK
731 001146 254000 004360' JRST OFERR3 ;BAD RETREIVAL INFO
732 001147 254000 001306' JRST DFRN8A ;WRITE PROTECTED
733 001150 621000 020000      TLZ IOS,PRCHG ;ARBITRARY CHANGE OK.
734

```

735	001151			DFREN2: NOSCHEDULE*	
736	001151	700600	001026'	CONO PI,DSKOFF	
737	001152	260140	002143'	PUSHJ PDP,SCANAT	
738	001153	254000	001162'	JRST DFRN1B	ISCAN ACCESS TABLE FROM BEGINNING
739					INO
740	001154	200241	000002	DFRN1A: MOVE DAT,TBITS(TAC)	
741	001155	602240	500000	TRNE DAT,ATBITS	YES, BEING ALTERED?
742	001156	254000	001312'	JRST DFREN9	IFILE BEING CREATED OR RENAMED?
743	001157	260140	002157'	PUSHJ PDP,SCANAT2	YES, YOU LOSE,
744	001160	254000	001162'	JRST DFRN1B	INO, CONTINUE SCANNING
745	001161	254000	001154'	JRST DFRN1A	
746					IFOUND ANOTHER
747	001162	554046	000014	DFRN1B: HLRZ TAC,DEVACC(DEV DAT)	
748	001163	201640	100000	MOVEI AC1,RTBIT	ISSET RENAME FLAG IN FIRST ENCOUNTERED ACCESS ENTRY
749	001164	436641	000002	ORM AC1,TBITS(TAC)	IRENAME BIT
750				SCHEDULE*	
751	001165	700600	001126'	CONO PI,DSKON	

```

752 001166 200060 000014 MOVE TAC,@UUO ;GET NEW FILENAME
753 001167 312046 000011 CAME TAC,DEVFIL(DEV DAT) ;SAME AS DDB?
754 001170 254000 001213' JRST DFREN3 ;NO
755 001171 271600 000001 ADDI UUO,1 ;YES, SAME EXTENSION?
756 001172 200060 000014 MOVE TAC,@UUO ;NEW EXTENSION
757 001173 430046 000012 XOR TAC,DEVEXT(DEV DAT) ;COMPARE WITH DDB?
758 001174 603040 777777 TLNE TAC,-1 ;SAME?
759 001175 364620 001213' SOJA UUO,DFREN3 ;NO
760
761
762 ;FILENAME AND EXTENSION ARE IDENTICAL, CHANGE ONLY PROTECTION
763
764 001176 621000 020000 TLZ IOS,PRCHG ;CLEAR PRTOECTION CHANGE FLAG
765 001177 271600 000001 ADDI UUO,1 ;YES, CHANGE PROTECTION
766 001200 135640 004603' LDB AC1,CPOINT 9,@UUO,8J ;NEW PROTECTION
767 001201 550106 000020 HRRZ TAC1,DSKBUF(DEV DAT) ;POINT TO BUFFER,CONTAINING RETRIEVAL INFO
768 001202 271100 000002 ADDI TAC1,2 ;POINT TO PROTECTION
769 001203 607000 000200 TLNN IOS,UBFS ;BUFFER IN MONITOR CORE?
770 001204 661100 000007 TLO TAC1,PROG ;NO, RELOCATE,
771 001205 137640 004604' DPB AC1,CPOINT 9,@TAC1,8J ;STORE NEW PROTECTION IN RIB
772
773 001206 260140 003274' PUSHJ PDP,WRI ;RE-WRITE THE BLOCK
774
775 001207 350003 777777 DFREXT: AOS -1(PDP) ;LEAVE
776 001210 262140 000001 POP PDP,TAC
777 001211 260140 003654' PUSHJ PDP,CLRBUF ;CLEAR ANY DUMP BUFFER
778 001212 254000 001603' JRST DFCL21 ;CLEAR ACCESS ENTRY AND EXIT
779
780 001213 623000 020000 DFREN3: TLZE IOS,PRCHG ;IS ONLY PROTECTION CHANGE PERMITTED?
781 001214 254000 001310' JRST DFREN8 ;YES, ERROR SINCE TRYING TO CHANGE NAME,FXT
782 001215 262140 000001 POP PDP,TAC ;BACK UP PDP
783 001216 260140 001071' PUSHJ PDP,ALTMFD ;ALTERING MFD?
784 001217 336020 000014 SKIPN @UUO ;RENAMING TO 0 (I.E. DELETING)?
785 001220 254000 001315' JRST DFREN7 ;YES
786
787 001221 200046 000014 MOVE TAC,DEVACC(DEV DAT) ;SAVE INFO ON OLD FILE THAT WILL BE CHANGED BY SETLE
788 001222 544046 000022 HLR TAC,SETCNT(DEV DAT) ;BLOCK# OF UFD RIB
789 001223 261140 000001 PUSH PDP,TAC ;SAVE XWD ACCESS POINTER, BLOCK#
790 001224 200506 000012 MOVE TA,DEVEXT(DEV DAT) ;BLOCK# OF FIRST FILE RIB
791 001225 500506 000015 HLL TA,DEV CNT(DEV DAT) ;RELATIVE TO UFD BLOCK#
792 001226 200546 000011 MOVE TB,DEVFIL(DEV DAT) ;HOLD FILE NAME
793
794 001227 205040 020000 MOVSI TAC,RENBIT
795 001230 436046 000010 ORM TAC,DEV OAD(DEV DAT) ;FOR "SETLE"
796 001231 260140 001342' PUSHJ PDP,SETLE0 ;LOOK FOR THIS NEW FILE NAME
797 001232 254000 004402' JRST DFERR4 ;NO UFD (SHOULD NOT HAPPEN)
798 001233 254000 001242' JRST DFREN5 ;NEW FILE NOT THERE
799 001234 255000 000000 JFCL ;NEW NAME ALREADY THERE, ERROR.
800 001235 255000 000000 JFCL
801 001236 262140 000001 POP PDP,TAC
802 SCHEDULE*
803 001237 700600 001165' CONO PI,DSKON
804 001240 260140 001603' PUSHJ PDP,DFCL21

```

DSKSRB - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (BURROUGHS)
ENTER, LOOKUP, RENAME, CLOSE

MACRO,V36 19:05 4-JUN-69 PAGE 32-1

805 001241 254000 004434'

JRST DFER10

```

806                                     ;ALTER RETRIEVAL INFORMATION
807
808 001242 262140 000005 DFREN5: POP PDP,DAT          ;BACK UP PDP
809
810 001243 202546 000011      MOVEM TB,DEVFIL(DEVDAT) ;RESTORE OLD FILE NAME TO DDB
811 001244 542506 000012      HRRM TA,DEVEXT(DEVDAT) ;RESTORE BLOCK# OF RIB
812 001245 502506 000015      HLLM TA,DEVcnt(DEVDAT) ;RESTORE UFD RELATIVE BLOCK
813
814 001246 200043 000000      MOVE TAC,(PDP)          ;XWD ACCESS POINTER, BLOCK#
815 001247 502046 000014      HLLM TAC,DEVACC(DEVDAT) ;RESTORE POINTER TO ACCESS ENTRY
816 001250 506046 000022      HRLM TAC,SETCNT(DEVDAT) ;RESTORE BLOCK# OF UFD RIB
817 001251 550040 000012      HRRZ TAC,TA
818 001252 206046 000016      MOVSM TAC,DEVBLK(DEVDAT) ;RESET BLOCK#, INDEX TO RETRIEVAL POINTFRS
819 001253 260140 003263'    PUSHJ PDP,RR1A          ;GET RETRIEVAL INFO BLOCK
820 001254 260140 003314'    PUSHJ PDP,SET000        ;POINT TAC1 TO WORD 0
821
822 001255 200060 000014      MOVE TAC,@UUO          ;GET NEW NAME
823 001256 202060 000002      MOVEM TAC,@TAC1        ;STORE IN RIB
824 001257 271100 000001      ADDI TAC1,1            ;CHANGE EXTENSION
825 001260 271600 000001      ADDI UUO,1
826 001261 200060 000002      MOVE TAC,@TAC1        ;GET OLD EXTENSION
827 001262 502046 000012      HLLM TAC,DEVEXT(DEVDAT) ;RESET EXTENSION IN DDB
828 001263 200060 000014      MOVE TAC,@UUO          ;NEW EXTENSION
829 001264 540040 000716'    HRR TAC,THSOAT         ;AND ACCESS DATE
830 001265 435040 400000      ORI TAC,400000         ;INSURE IT IS NON-ZERO
831 001266 202060 000002      MOVEM TAC,@TAC1        ;STORE IN RIB
832 001267 271100 000001      ADDI TAC1,1            ;CHANGE PROTECTION
833 001270 271600 000001      ADDI UUO,1
834 001271 135040 004603'    LDB TAC,[POINT 9,@UUO,8] ;GET NEW PROTECTION
835 001272 137040 004604'    OPB TAC,[POINT 9,@TAC1,8] ;STORE IN RIB
836
837 001273 260140 003274'    PUSHJ PDP,WRI          ;RE-WRITE RIB

```

```

838          ;ALTER UFD BLOCK
839
840 001274 260140 001720'      PUSHJ PDP,FINDE          ;BRING IN CORRECT BLOCK
841
842 001275 275600 000001      SUBI UUO,1              ;REPLACE EXTENSION IN UFD
843 001276 200060 000014      MOVE TAC,@UUO
844 001277 502060 000002      HLLM TAC,@TAC1
845 001300 275600 000001      SUBI UUO,1              ;AND NAME
846 001301 275100 000001      SUBI TAC1,1
847 001302 200060 000014      MOVE TAC,@UUO
848 001303 202060 000002      MOVEM TAC,@TAC1
849
850 001304 260140 002105'      PUSHJ PDP,WUFD          ;RE-WRITE THE BLOCK
851 001305 254000 001207'      JRST DFREXT            ;LEAVE
852
853          ;FILE COULD NOT BE RENAMED BECAUSE OF PROTECTION
854
855 001306 603000 020000 DFRN8A: TLNE IOS,PRCHG          ;PROTECTION CHANGE?
856 001307 254000 001151'      JRST DFREN2            ;YES
857
857 001310 260140 001603' DFRN8: PUSHJ PDP,DFCL21        ;TAKE OUT ACCESS ENTRY
858 001311 254000 004400'      JRST DFERR5            ;AND LEAVE
859
860          ;FILE COULD NOT BE RENAMED BECAUSE IT WAS BEING CREATED.
861
862 001312          DFRN9: SCHEDULE+
863 001312 700600 001237'      CONO PI,DSKON
864 001313 260140 001603'      PUSHJ PDP,DFCL21        ;TAKE OUT ACCESS ENTRY
865 001314 254000 004372'      JRST DFERR6

```

866					DELETE A FILE FROM A UFD	
867						
868	001315	260140	001720'	DFREN7:	PUSHJ PDP,FINDE	FIND THE CORRECT BLOCK
869						
870					NOSCHEDULE	*
871	001316	700600	001151'		CONO PI,DSKOFF	
872	001317	201120	000002		MOVEI TAC1,@TAC1	
873	001320	275100	000001		SUBI TAC1,1	DESTINATION
874	001321	505102	000002		HRLI TAC1,2(TAC1)	SOURCE IS ENTRY BEYOND DESTINATION
875						
876	001322	550706	000020	HRRZ	AC2,DSKBUF(DEVDAT)	
877	001323	271700	000176	ADDI	AC2,BLKSIZE-2	SET LIMIT OF BLT
878	001324	607000	000200	TLNN	10S,UDFS	BUFFER IN MONITOR ONE?
879	001325	271707	000000	ADDI	AC2,(PROG)	NO, RELOCATE
880						
881	001326	251116	000000	BLT	TAC1,(AC2)	MOVE SOME WORDS UP (ONE TOO MANY BECAUSE I COULD BE DELETING 64TH ENTRY IN UFD BLOCK)
882						
883						
884	001327	402020	000016	SETZM	@AC2	MAKE SURE OF ZERO
885	001330	402016	000001	SETZM	1(AC2)	
886					SCHEDULE*	
887	001331	700600	001312'	CONO	PI,DSKON	
888						
889	001332	260140	002105'	PUSHJ	PDP,WUFD	RE-WRITE
890	001333	350003	000000	AOS	(PDP)	SKIP RETURN
891	001334	554046	000014	HLRZ	TAC,DEVACC(DEVDAT)	POINT TO ACCESS ENTRY
892	001335	201100	200000	MOVEI	TAC1,DBIT	
893	001336	436101	000002	IORM	TAC1,TBITS(TAC)	SET TO DELETE WHEN THROUGH READING
894	001337	254000	001603'	JRST	DFCL21	CLOSE ACCESS ENTRY AND LEAVE

```

895      ;SET UP ARGUMENT LIST FOR LOOKUP, ENTER, AND RENAME
896      ;SEARCH THRU MFD AND UFD FOR THE FILE-NAME.
897      ;EXIT TO CALL+1 IF A UFD CANNOT BE FOUND.
898      ;EXIT TO CALL+2 IF THE FILE IS NOT FOUND.
899      ;EXIT TO CALL+3 IF FILE IS BEING WRITTEN
900      ;EXIT TO CALL+4 IF FILE IS BEING READ
901      ;EXIT TO CALL+5 IF THE FILE DOES NOT HAVE ACCESS ENTRY.
902
903      ;UPON EXIT,      (PDP) = OWNER PROJ,-PROG. NUMBER
904
905      EXTERNAL SYSPP,JOB,PRJPRG,CPOPJ2,CPOPJ1
906
907      001340 205040 000000 SETLE:  MOVSI TAC,RENBIT      ;TURN OFF THE FLAG
908      001341 412046 000010      ANDCAM TAC,DEVOAD(DEV DAT)
909
910      001342 202006 000002 SETLE0: MOVEM IOS,DEVIOS(DEV DAT)
911      001343 661600 000007      TLO U00,PROG
912      001344 200660 000014      MOVE AC1,@U00      ;PICK UP FILE NAME
913      001345 202646 000011      MOVEM AC1,DEVFIL(DEV DAT) ;STORE FILE NAME AND EXT IN DOB
914
915      001346 271600 000001      ADDI U00,1
916      001347 510660 000014      HLLZ AC1,@U00
917      001350 202646 000012      MOVEM AC1,DEVEXT(DEV DAT)
918      001351 271600 000002      ADDI U00,2      ;U00 POINTS TO PP
919      001352 200200 001076' SETLE9: MOVE ITEM,JOB      ;TRY CURRENT JOB PP.
920      001353 337260 000014      SKIPG DAT,@U00      ;IS IT A PP?
921      001354 200244 001077'      MOVE DAT,PRJPRG(ITEM) ;NO, GET PP FROM TABLE.
922      001355 603300 000100      TLNE DEV DAT,SYSDEV      ;SYSTEM DEVICE?
923      001356 200240 001074'      MOVE DAT,SYSPP      ;YES, USE SYSTEM (CUSP) PROJ,PROG #
924      001357 202260 000014      MOVEM DAT,@U00
925      001360 202246 000013      MOVEM DAT,DEVPPN(DEV DAT) ;SAVE PRJ,PRG FOR USE BY U00CON
926      001361 275600 000003      SUBI U00,3      ;U00 POINTS TO FILE NAME
927      001362 250243 000000      EXCH DAT,(PDP)      ;PUT PP-NUMBER IN PD LIST
928      001363 261140 000005      PUSH PDP,DAT      ;PDP POINTS TO CALL+1
  
```

```

928                                     ;SEARCH MFD FOR THE OWNER OF THE FILE
929
930 001364 200243 777777             MOVE DAT,-1(PDP)             ;XWD PROJ,PROG
931 001365 205640 654644             MOVSI AC1,(SIXBIT/UFD/)      ;EXTENSION "UFD"
932 001366 200040 000000'           MOVE TAC,MFDBLK             ;BLOCK# OF MFD RIB
933 001367 260140 001634'           PUSHJ PDP,DIRSRC
934 001370 263140 000000             POPJ PDP,                     ;UFD NOT THERE
935
936                                     ;SEARCH UFD FOR THE FILE
937
938 001371 350003 000000             AOS (PDP)                   ;PDP POINTS TO CALL+2
939 001372 271040 000001             ADDI TAC,1                  ;POINT TO XWD SIXBIT/EXT/,BLOCK#
940 001373 550060 000001             HRRZ TAC,@TAC               ;BLOCK# OF UFD RIB
941 001374 506046 000022             HRLM TAC,SETCNT(DEVDAT)      ;SAVE BLOCK#
942 001375 402006 000015             SETZM DEVCNT(DEVDAT)        ;LW WILL BE USED TO COUNT UFD BLOCKS
943 001376 200646 000012             MOVE AC1,DEVEXT(DEVDAT)
944 001377 260140 001704'           PUSHJ PDP,DSRC10             ;SEARCH UFD
945 001400 263140 000000             POPJ PDP,                     ;FILE NOT THERE
946
947 001401 271040 000001             ADDI TAC,1                  ;POINT TO XWD SIXBIT/EXT/,BLOCK#
948 001402 200760 000001             MOVE AC3,@TAC
949 001403 542746 000012             HRRM AC3,DEVEXT(DEVDAT)      ;SAVE BLOCK# OF FIRST RIB
950 001404 350003 000000             AOS (PDP)                   ;PDP POINTS TO CALL+3
951 001405 200703 777777             MOVE AC2,-1(PDP)            ;XWD PROJ,PROG
952                                     NOSCHEDULE*
953 001406 700600 001316'           CONO PI,DSKOFF
954 001407 260140 002144'           PUSHJ PDP,SCNAT0              ;LOOK IN ACCESS TABLE
955 001410 254000 000000             JRST CPOPJ2                  ;NOT THERE. RETURN TO CALL+5
956 001411 200241 000002             MOVE DAT,TBITS(TAC)         ;RENAMING?
957 001412 602240 400000             TRNE DAT,WTBIT              ;BEING WRITTEN?
958 001413 263140 000000             POPJ PDP,                     ;YES. RETURN TO CALL+3
959 001414 606240 100000             TRNN DAT,RTBIT
960 001415 254000 001105'           JRST CPOPJ1                  ;NO, ALL OK. RETURN TO CALL+4
961 001416 200046 000010             MOVE TAC,DEVOAD(DEVDAT)      ;ARE WE RENAMING?
962 001417 603040 020000             TLNE TAC,RENBIT
963 001420 262140 000001             POP PDP,TAC                  ;YES
964 001421 262140 000001             POP PDP,TAC
965 001422 254000 004372'           JRST DFERR6

```

```

966          ;CHECK RETRIEVAL INFO FOR ELIGIBILITY
967          ;ENTER WITH C(DAT) = PROTECTION TO CHECK IN BITS 0-2
968          ;EXIT TO CALL+1 IF NAME OR EXTENSION WRONG
969          ;EXIT TO CALL+2 IF PROTECTION WRONG
970          ;EXIT TO CALL+3 IF ALL OK
971
972          EXTERNAL JOB,PRJPRG,CPOPJ1,DUMPPP
973
974 001423 201046 000023  PROTK: MOVEI TAC,PTR1(DEVDAT) ;POINT TO RETRIEVAL INFO
975 001424 200660 000001      MOVE AC1,@TAC          ;NAME OK?
976 001425 312646 000011      CAME AC1,DEVFIL(DEVDAT)
977 001426 263140 000000      POPJ PDP,              ;NO
978 001427 510706 000012      HLLZ AC2,DEEXT(DEVDAT)
979 001430 510641 000001      HLLZ AC1,1(TAC)          ;EXTENSION FROM RIB
980 001431 312640 000016      CAME AC1,AC2
981 001432 263140 000000      POPJ PDP,              ;EXTENSION NOT RIGHT
982
983 001433 350003 000000      AOS (PDP)
984 001434 200643 777777  PROTKX: MOVE AC1,-1(PDP)

985 001435 200200 001352'  PROTKY: MOVE ITEM,JOB
986 001436 430644 001354'      XOR AC1,PRJPRG(ITEM)
987 001437 606640 777777      TRNN AC1,-1          ;SAME PROGRAMMER?
988 001440 254000 001444'      JRST PROT1          ;YES
989
990 001441 241240 777775      ROT DAT,-3          ;ROTATE TO PROJECT FIELD
991 001442 603640 777777      TLNE AC1,-1          ;NO, SAME PROJECT?
992 001443 241240 777775      ROT DAT,-3          ;NO, ROTATE TO UNIVERSAL FIELD
993
994 001444 621240 600000  PROT1: TLZ DAT,600000      ;DO NOT TEST OWNER READ OR PROTECT PROTECTION
995 001445 616241 000002      TDNN DAT,2(TAC)
996 001446 254000 001415'      JRST CPOPJ1
997 001447 200644 001436'      MOVE AC1,PRJPRG(ITEM) ;PROTECTION O.K.
998 001450 316640 000000      CAMN AC1,DUMPPP      ;EQUAL TO DUMPPER PROJECT
999                                     ;PROGRAMMER NUMBER?
1000 001451 350003 000000      AOS (PDP)          ;YES, ALL FILES AVAILABLE
1001 001452 263140 000000      POPJ PDP,

```

```

1002          ;CLOSE UUO
1003          ;CLOSE AN OUTPUT FILE
1004
1005          EXTERNAL PIOMOD, WAIT1, OUT
1006
1007 001453 607300 020000 DFCLSO: TLNN DEVDAT, ENTRB      ;ENTER DONE YET?
1008 001454 263140 000000 POPJ PDP,                    ;NO, FORGET IT
1009 001455 554046 000014 HLRZ TAC, DEVACC(DEV DAT) ;SHOULD WE CLOSE? (POINT TO ACCESS ENTRY)
1010          IFN FTRCHK, <
1011 001456 336000 000001 SKIPN TAC                ;ACCESS TABLE POINTER EXISTS?
1012 001457 254200 001457' HALT .                    ;IF NOT, HALT, NO RE-START POSSIBLE.
1013
1014 001460 200041 000002 > MOVE TAC, TBITS(TAC)
1015          IFN FTRCHK, <
1016 001461 606040 030000 TRNN TAC, ATCLO+RWTBIT      ;CREATION OR UPDATE OF FILE IN PROGRESS?
1017 001462 254200 001462' HALT .                    ;NO, ERROR, CAN'T CONTINUE
1018
1019 001463 135040 000000 > LDB TAC, PIOMOD
1020 001464 305040 000016 CAIGE TAC, DR              ;DUMP MODE?
1021 001465 603300 000020 TLNE DEVDAT, DSKRLB        ;RESET UUO IN PROGRESS?
1022 001466 254000 001506' JRST DFCL2                ;YES TO EITHER QUESTION
1023 001467 554046 000006 HLRZ TAC, DEVBUFF(DEV DAT) ;NO, GET ADDRESS OF OUTPUT
1024          ; BUFFER HEADER BLOCK,
1025 001470 661040 000007 TLO TAC, PROG              ;RELOCATE
1026 001471 337120 000001 SKIPG TAC1, @TAC          ;VIRGIN BUFFERS (NO RING SET-UP) ?
1027 001472 254000 001506' JRST DFCL2                ;YES, DON'T OUTPUT
1028 001473 350000 000001 AOS TAC                    ;TAC POINTS TO OUTPUT BYTE POINTER
1029 001474 270100 004605' ADD TAC1, CXWD PROG, 11    ;TAC1 POINTS TO WORD COUNT PRECEDING
1030          ; USER DATA BUFFER
1031 001475 550660 000001 HRRZ AC1, @TAC              ;PICK UP OUTPUT BYTE POINTER
1032 001476 332000 000015 SKIPE AC1                  ;DON'T CALCULATE WORD COUNT IF BYTE POINTER
1033          ; NOT SET UP
1034 001477 275642 000000 SUBI AC1, (TAC1)             ;CALCULATE NUMBER OF WORDS USER HAS FILLED
1035 001500 602000 000020 TRNE IOS, IOWC             ;USER KEEPING HIS OWN WORD COUNT ?
1036 001501 550660 000002 HRRZ AC1, @TAC1           ;YES, SUBSTITUTE HIS COUNT FOR
1037          ; COMPUTED WORD COUNT
1038 001502 336000 000015 SKIPN AC1                  ;WORD COUNT EQUAL TO 0?
1039 001503 254000 001506' JRST DFCL2                ;YES, DON'T OUTPUT 0-WORD FINAL BLOCK.
1040 001504 260140 000000 PUSHJ PDP, OUT            ;NO, GO WRITE LAST PARTIAL BUFFER
1041 001505 260140 000000 PUSHJ PDP, WAIT1          ;WAIT FOR IT TO FINISH
1042
1043 001506 260140 003620' DFCL2: PUSHJ PDP, SETBUF    ;CHOOSE A BUFFER AREA FOR RETRIEVAL POINTERS
1044 001507 661000 002100 TLO IOS, NMP:NCTRLC        ;SET NMP, MUST NOT INTERRUPT POINTER WRITING
1045 001510 202006 000002 MOVEM IOS, DEVIOS(DEV DAT) ;SO DF04A WON'T READ NEW POINTERS IN
1046 001511 260140 003131' PUSHJ PDP, DF04A          ;WRITE OUT LAST BLOCK OF POINTERS
1047 001512 554646 000016 HLRZ AC1, DEVBLK(DEV DAT)
1048 001513 430646 000012 XOR AC1, DEVEXT(DEV DAT)
1049 001514 602640 777777 TRNE AC1, -1              ;ONLY ONE BLOCK OF POINTERS?
1050 001515 260140 003261' PUSHJ PDP, RRB           ;NO, READ FIRST BLOCK INTO BUFFER
1051 001516 260140 003314' PUSHJ PDP, SET000
1052 001517 271100 000003 ADDI TAC1, 3
1053 001520 210646 000015 MOVN AC1, DEVCNT(DEV DAT) ;SET TAC1 TO POINT TO 4TH WORD OF BUFFER
1054 001521 506660 000002 HRLM AC1, @TAC1          ;PICK UP AND NEGATE SIZE OF FILE
          ;NEGATIVE WORD COUNT INTO FOURTH WORD OF FIRST RIB

```

DSKSRB - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (BURROUGHS) MACRO,V36 19:05 4-JUN-69 PAGE 39-1
ENTER, LOOKUP, RENAME, CLOSE

1055 001522 260140 003273'

PUSHJ PDP,WRIB

WRITE OUT FIRST BLOCK OF RETRIEVAL INFORMATION

```

1056                                     ;CLOSE UUD CONTINUED.
1057                                     ;THE FILE AND ALL ITS POINTERS HAVE BEEN PUT ON DISK.
1058                                     ;NOW PUT ENTRY IN DIRECTORY.
1059
1060 001523 621300 0020000 TLZ DEVDAT,ENTRB
1061 001524 554046 000014 HLRZ TAC,DEVACC(DEVDAT) ;PICK UP ACCESS TABLE POINTER
1062 001525 200241 000002 MOVE DAT,TBITS(TAC) ;GET STATUS BITS AND READ COUNT
1063 001526 622240 010000 TRZE DAT,RWTRIT ;CLOSING AN UPDATED FILE?
1064 001527 254000 001567' JRST DFCLU1 ;YES
1065 001530 603300 000020 TLNE DEVDAT,DSKRLB ;RESET UUD IN PROGRESS ?
1066 001531 254000 001556' JRST DFC16A ;YES
1067 001532 606240 040000 TRNN DAT,ATIND ;IS NAME ALREADY IN DIRECTORY?
1068 001533 254000 001562' JRST DFCL20 ;NO, INSERT IT
1069 001534 202006 000002 MOVEM IOS,DEVINS(DEVDAT)
1070 001535 260140 001720' PUSHJ PDP,FINDC ;BRING IN BLOCK WITH THIS ENTRY.
1071 001536 200046 000012 MOVE TAC,DEVEXT(DEVDAT) ;CHANGE BLOCK POINTER
1072 001537 250060 000002 EXCH TAC,@TAC1 ;TAC1 POINTS TO ENTRY IN UFD BLOCK
1073 001540 202046 000012 MOVEM TAC,DEVEXT(DEVDAT) ;SAVE POINTER TO RIB OF OLD VERSION FOR RECLAM
1074 001541 260140 002105' PUSHJ PDP,WUFD ;WRITE THE BLOCK BACK OUT
1075 001542 554046 000014 HLRZ TAC,DEVACC(DEVDAT) ;ACCESS TABLE POINTER
1076 NOSCHEDULE*
1077 001543 700600 001406' CONO PI,DSKOFF
1078 001544 261141 000000 PUSH PDP,ATPP(TAC) ;SAVE XWD PROJ,PROG
1079 001545 260140 002163' PUSHJ PDP,CLRAT ;REMOVE THE ACCESS ENTRY
1080 001546 262140 000016 POP PDP,AC2 ;XWD PROJ,PROG
1081 001547 260140 002144' PUSHJ PDP,SCNAT0 ;SCAN FOR ANY OTHERS READING OLD VERSION
1082 001550 254000 001557' JRST DFCL17 ;NONE
1083
1084 001551 201640 200000 DFCL16: MOVEI AC1,DTBIT ;DELETE WHEN THRU READING SINCE IT WAS JUST UPDATED
1085 001552 436641 000002 IORM AC1,TBITS(TAC)
1086 001553 260140 002157' PUSHJ PDP,SCNAT2 ;LOOK FOR MORE
1087 001554 254000 003654' JRST CLRBUF ;NO MORE
1088 001555 254000 001551' JRST DFCL16
1089 001556 260140 002163' DFC16A: PUSHJ PDP,CLRAT ;UPON RESET UUD WITH PARTIALLY WRITTEN FILE.
1090 ; CLEAR ACCESS TABLE ENTRY RECLAIM DISK SPACE,
1091
1092 001557 DFCL17: SCHEDULE*
1093 001557 700600 001331' CONO PI,DSKON
1094 001560 550046 000012 HRRZ TAC,DEVEXT(DEVDAT) ;RIB# OF FILE
1095 001561 254000 003327' JRST RECLAM
1096
1097 001562 260140 001756' DFCL20: PUSHJ PDP,INSUIR ;INSERT THE NAME
1098 001563 554046 000014 HLRZ TAC,DEVACC(DEVDAT) ;ACCESS TABLE POINTER
1099 NOSCHEDULE*
1100 001564 700600 001543' CONO PI,DSKOFF
1101 001565 260140 002163' PUSHJ PDP,CLRAT ;REMOVE THE ACCESS ENTRY
1102 001566 254000 003654' JRST CLRBUF ;CLEAR ANY BUFFER IN FREE STORAGE
1103 001567 202241 000002 DFCLU1: MOVEM DAT,TBITS(TAC) ;STORE STATUS BITS (WITH UPDATE MARKER CLEARED)
1104 001570 603300 040000 TLNE DEVDAT,LOOKB ;WAS INPUT SIDE OF FILE ALSO CLOSED?
1105 001571 263140 000000 POPJ PDP, ;NO, LET THE USER KEEP READING IT
1106 001572 602240 007777' TRNE DAT,RCOUNT ;YES, ANYBODY ELSE STILL READING IT?
1107 001573 254000 001577' JRST DFCLU2 ;YES
1108 001574 602240 200000 TRNE DAT,DTBIT ;NO, WAS IT MARKED FOR DELETION?

```

DSKSRB - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (BURROUGHS) MACRO,V36 19:05 4-JUN-69 PAGE 40-1
ENTER, LOOKUP, RENAME, CLOSE

1109	001575	254000	001556'	JRST DFC16A	YES, GO DELETE IT AND RECLAIM DISK SPACE
1110	001576	260140	002163'	PUSHJ PDP,CLRAT	NO, CLEAR ACCESS TABLE ENTRY
1111	001577	553006	000014	DFCLU2: HRRZS DEVACC(DEV DAT)	CLEAR POINTER TO ACCESS TABLE
1112	001600	254000	003654'	JRST CLRBUF	CLEAR ANY BUFFER AREA AND EXIT.

```

1113                                     ;CLOSE UUO CONTINUED.
1114                                     ;CLOSE AN INPUT FILE.
1115
1116 001601 627300 040000 DFCLSI: TLZN DEVDAT,LOOKR
1117 001602 263140 000000 POPJ PDP,
1118 001603 554046 000014 DFCL21: HLRZ TAC,DEVACC(DEVDAT) ;POINT TO ACCESS ENTRY
1119 001604 322040 000000 JUMPE TAC,CPOPJ ;EXIT IF ACCESS TABLE ALREADY CLOSED
1120                                     ; (PROBABLY *C DURING EARLIER ATTEMPT)
1121 001605 200101 000002 MOVE TAC1,TBITS(TAC) ;GET STATUS BITS
1122 IFN FTRCHK,<
1123 001606 602100 020000 TRNE TAC1,ATCLO ;IS THIS FILE SIMPLY BEING CREATED (NOT UPDATED)?
1124 001607 254200 001604' HALT CPOPJ ;IF SO, SOMEBODY IS VERY CONFUSED.
1125 >
1126 NOSCHEDULE+
1127 001610 700600 001564' CONO PI,OSKOFF
1128 001611 370101 000002 SOS TAC1,TBITS(TAC) ;DECREMENT RCOUNT
1129 001612 602100 017777 TRNE TAC1,RCOUNT+RWTBIT ;ANY MORE READS? OR OUTPUT CLOSE TO DO?
1130 001613 254000 001617' JRST DFCL23 ;YES, LEAVE
1131
1132 001614 602100 200000 TRNE TAC1,DTBIT ;TO BE DELETED?
1133 001615 254000 001556' JRST DFC16A ;YES
1134 001616 260140 002163' PUSHJ POP,CLRAT ;CLEAR THE ENTRY AND EXIT
1135 001617 DFCL23: SCHEDULE+
1136 001617 700600 001557' CONO PI,OSKON
1137 001620 627300 010000 TLZN DEVDAT,INPB ;ANY INPUT UUO?
1138 001621 254000 003654' JRST CLRBUF ;NO, JUST EXIT, RETURNING MON BUF
1139 001622 260140 003620' PUSHJ PDP,SETBUF ;YES, UPDATE ACCESS DATE AS FOLLOWS:
1140 001623 260140 003261' PUSHJ PDP,RRIB ;READ IN JUST BLOCK OF RETRIEVAL INFORMATION.
1141 001624 275100 000176 SUBI TAC1,BLSIZ-2 ;SET TAC1 TO 2ND WORD OF BLOCK
1142 001625 135640 004606' LDB AC1,[POINT 17,@TAC1,35] ;GET PREVIOUS ACCESS DATA.
1143 001626 316640 001264' CAMN AC1,THSDAT ;ALREADY ACCESSED EARLIER TODAY?
1144 001627 254000 003654' JRST CLRBUF ;YES, JUST CLEAR BUFFER AND EXIT.
1145 001630 200640 001626' MOVE AC1,THSDAT ;NO, STORE TODAY(S DATE AS NEW ACCESS DATE.
1146 001631 137640 004606' DPB AC1,[POINT 17,@TAC1,35] ;
1147 001632 260140 003273' PUSHJ PDP,WRIB ;REWRITE FIRST BLOCK OF RETRIEVAL INFORMATION.
1148 001633 254000 003654' JRST CLRBUF ;CLEAR BUFFER AND EXIT.

```

```

1149          SUBTTL DIRECTORY SEARCHING
1150          ;SEARCH DIRECTORY FOR A PARTICULAR ENTRY.
1151          ;ENTER BY:      MOVE DAT,NAME
1152          ;              HLLZ AC1,EXTENSION
1153          ;              HRRZ TAC,BLOCK NUMBER FOR POINTERS
1154          ;              PUSHJ PDP,DIRSRC
1155          ;              EXIT1      NAME NOT FOUND
1156          ;              EXIT2      NAME FOUND, TAC POINTS TO ENTRY
1157
1158
1159 001634 261140 000015 DIRSRC: PUSH PDP,AC1
1160 001635 261140 000005      PUSH PDP,DAT      ;0(PDP) IS FILENAME,-1(PDP) IS EXTENSION
1161 001636 260140 003316'      PUSHJ PDP,SETPTR      ;READ UFD RETRIEVAL POINTERS
1162 001637 550046 000014 DRSRC0: HRRZ TAC,DEVACC(DEV DAT)
1163 001640 303046 000037      CAILE TAC,PTRN(DEV DAT)      ;MORE POINTERS IN CORE?
1164 001641 254000 001667'      JRST DRSRC7      ;NO, GET MORE,
1165 001642 550041 000000      HRRZ TAC,(TAC)
1166 001643 322040 001701'      JUMPE TAC,DRSRC6      ;0 SIGNALS END OF POINTERS
1167 001644 350006 000014      AOS DEVACC(DEV DAT)      ;NEXT POINTER

1168 001645 350006 000015      AOS DEVCNT(DEV DAT)      ;COUNT UFD BLOCKS SEARCHED
1169 001646 260140 004110'      PUSHJ PDP,MQIN      ;READ THE DATA BLOCK
1170 001647 254000 004457'      JRST RERA      ;ERROR, YOU LOSE.
1171
1172 001650 200106 000020      MOVE TAC1,DSKBUF(DEV DAT) ;XWD=L, POINTER
1173 001651 550040 000002      HRRZ TAC,TAC1
1174 001652 607000 000200      TLNN IOS,UBFS      ;DUMP MODE?
1175 001653 505040 000007      HRLI TAC,PROG      ;NO, RELOCATE,
1176 001654 510643 777777      HLLZ AC1,-1(PDP)      ;SIXBIT/EXT/
1177 001655 336760 000001 DRSRC3A: SKIPN AC3,@TAC      ;BLOCK ENTIRELY SEARCHED?
1178 001656 254000 001637'      JRST DRSRC0      ;YES
1179 001657 271040 000001      ADDI TAC,1
1180 001660 316743 000000      CAMN AC3,(PDP)      ;NAMES MATCH?
1181 001661 254000 001673'      JRST DRSRC8      ;YES
1182
1183 001662 271040 000001 DRSRC4: ADDI TAC,1
1184 001663 550740 000001      HRRZ AC3,TAC
1185 001664 302742 000200      CAIE AC3,BLKSIZE(TAC1)      ;IS BLOCK ENTIRELY SEARCHED?
1186 001665 254000 001655'      JRST DRSRC3A      ;NO
1187 001666 254000 001637'      JRST DRSRC0
1188
1189 001667 603000 000100 DRSRC7: TLNE IOS,NMP      ;ANY MORE POINTERS ON DISK?
1190 001670 254000 001701'      JRST DRSRC6      ;NO
1191 001671 260140 003012'      PUSHJ PDP,GETPTR      ;YES, GET SOME
1192 001672 254000 001637'      JRST DRSRC0
1193
1194 001673 510760 000001 DRSRC8: HLLZ AC3,@TAC      ;DOES EXTENSION MATCH
1195 001674 312740 000015      CAMC AC3,AC1
1196 001675 254000 001662'      JRST DRSRC4      ;NO
1197 001676 275040 000001      SUBI TAC,1      ;POINT TO FIRST WORD OF ENTRY
1198
1199 001677 350003 777776      AOS -2(PDP)
1200 001700 207006 000015      MOVSS DEVCNT(DEV DAT)      ;YES, ENTRY IS FOUND.
1201                                ;SAVE COUNT OF UFD BLOCKS FOR USE AT CLOSE

```

1202				EXIT.	
1203					
1204	001701	262140	000005	DRSRC6: POP PDP,DAT	
1205	001702	262140	000015	POP PDP,AC1	
1206	001703	263140	000000	POPJ PDP,	;"FILE" NOT FOUND
1207				;"CHECK PROTECTION ON UFD	
1208					
1209	001704	261140	000015	DRSRC10: PUSH PDP,AC1	;"PUSH EXTENSION
1210	001705	260140	003316	PUSHJ PDP,SETPTR	
1211	001706	205240	200000	MOVSI DAT,200000	;"SET UP TO TEST READ BIT
1212	001707	603000	000020	TLNE IOS,IO	;"IS IT LOOKUP?
1213	001710	241240	777777	ROT DAT,-1	;"NO,TEST WRITE PROTECTION
1214	001711	201046	000023	MOVEI TAC,PTR1(DEVDAT)	
1215	001712	550643	777775	HRRZ AC1,-3(PDP)	;"XWD PROJ,PROG
1216	001713	621000	020000	TLZ IOS,PRCHG	
1217	001714	260140	001435	PUSHJ PDP,PROTKY	;"FLAG PROTECTION CHANGE
1218	001715	661000	020000	TLO IOS,PRCHG	
1219	001716	261146	000011	PUSH PDP,DEVFIL(DEVDAT)	;"PUSH FILENAME TO MAKE PDL LOOK LIKE DIRSRC
1220	001717	254000	001637	JRST DRSRC0	

```

1221                                     ;FIND A UFD ENTRY, IT MUST BE THERE.
1222
1223                                     EXTERNAL AUREQ,AUWAIT
1224
1225 001720 352000 000000 FINDE: AOSE AUREQ
1226 001721 260140 000000      PUSHJ PDP,AUWAIT
1227 001722 661000 010000      TLO IOS,AUFLG
1228 001723 554046 000022      HLRZ TAC,SETCNT(DEVDAT) ;SET TAC TO UFD POINTERS
1229 001724 554246 000015      HLRZ DAT,DEVcnt(DEVDAT) ;SET DAT TO COUNT COMPUTED BY DIRSRC
1230 001725 271240 000004      ADDI DAT,DIRSZ
1231
1232 001726 206046 000016 FINDE1: MOVSM TAC,DEVBLK(DEVDAT);READ A POINTER BLOCK
1233 001727 260140 003263      PUSHJ PDP,RR1A
1234 001730 303240 000176      CAILE DAT,BLKSIz-2          ;POINTER IN THIS BLOCK?
1235 001731 254000 001752      JRST FINDE2          ;NO, GET NEXT ONE
1236 001732 275240 000001      SUBI DAT,1          ;YES, BRING THEM IN
1237 001733 542246 000016      HRRM DAT,DEVBLK(DEVDAT) ;SET INDEX INTO R1B
1238 001734 260140 003016      PUSHJ PDP,DFIN4        ;COPY POINTERS INTO DDB
1239 001735 550046 000023      HRRZ TAC,PTR1(DEVDAT)   ;GET RETRIEVAL POINTER FOR UFD BLOCK

1240 001736 260140 004110      PUSHJ PDP,MQIN        ;READ THE UFD BLOCK
1241 001737 254000 004457      JRST RERA          ;ERROR
1242 001740 260140 003314      PUSHJ PDP,SET000        ;FIND THE ENTRY
1243 001741 510646 000012      HLLZ AC1,DEVEXT(DEVDAT) ;GET EXTENSION FROM DDB
1244
1245 001742 200060 000002 SRCU1: MOVE TAC,@TAC1          ;GET FILE NAME FROM UFD
1246 001743 271100 000001      ADDI TAC1,1
1247 001744 312046 000011      CAME TAC,DEVFIL(DEVDAT) ;SAME?
1248 001745 344100 001742      AOJA TAC1,SRCU1        ;NAMES DO NOT MATCH
1249 001746 510060 000002      HLLZ TAC,@TAC1        ;GET EXTENSION FROM UFD
1250 001747 312040 000015      CAME TAC,AC1          ;SAME?
1251 001750 344100 001742      AOJA TAC1,SRCU1        ;EXTENSIONS DO NOT MATCH
1252 001751 263140 000000      POPJ PDP,              ;FOUND IT
1253
1254
1255 001752 260140 003311 FINDE2: PUSHJ PDP,SET176        ;GET THE NEXT POINTER BLOCK
1256 001753 554060 000002      HLRZ TAC,@TAC1
1257 001754 275240 000176      SUBI DAT,BLKSIz-2        ;BACK UP DAT
1258 001755 254000 001726      JRST FINDE1
    
```

```

1259          ;INSERT AN ENTRY IN A DIRECTORY,
1260
1261          EXTERNAL AUREQ,AUWAIT,AUAVAIL
1262
1263 001756 352000 001720' INSDIR: AOSE AUREQ
1264 001757 260140 001721' PUSHJ PDP,AUWAIT
1265 001760 661000 010000 TLO IOS,AUFLG
1266 001761 402006 000015 SETZM DEVCNT(DEVDAT) ;CLEAR BEFORE COUNTING BLOCKS OF UFD
1267 001762 554046 000022 HLRZ TAC,SETCNT(DEVDAT) ;BLOCK# OF UFD RIB
1268 001763 260140 003316' PUSHJ PDP,SETPTR ;SET UP FIRST POINTERS,
1269 001764 370006 000014 SOS DEVACC(DEVDAT) ;SYNC
1270
1271 001765 350046 000014 INSD1: AOS TAC,DEVACC(DEVDAT) ;GET NEXT POINTER,
1272 001766 553000 000001 HRRZS TAC
1273 001767 350036 000015 AOS DEVCNT(DEVDAT) ;COUNT EACH UFD BLOCK
1274 001770 303046 000037 CAILE TAC,PTRN(DEVDAT) ;ANY MORE IN CORE?
1275 001771 254000 002073' JRST INSD10 ;NO, GET SOME MORE.
1276
1277 001772 550041 000000 INSD2: HRRZ TAC,(TAC) ;GET RETRIEVAL POINTER
1278 001773 322040 002023' JUMPE TAC,INSD5 ;0 SIGNALS END OF POINTERS
1279 001774 260140 004110' PUSHJ PDP,MQIN ;READ UFD BLOCK
1280 001775 254000 004457' JRST RERA ;ERROR,
1281
1282 001776 260140 003311' PUSHJ PDP,SET176 ;SET TAC1 TO WORD 127
1283 001777 332020 000002 SKIPE @TAC1 ;IS IT ZERO?
1284 002000 254000 001765' JRST INSD1 ;NO, BLOCK FULL,
1285
1286 002001 207006 000015 MOVSS DEVCNT(DEVDAT) ;SAVE RELATIVE# OF BLOCK CONTAINING NEW ENTRY
1287 002002 260140 002004' PUSHJ PDP,INSD3 ;INSERT ENTRY,
1288
1289 002003 254000 002105' JRST WUFD ;WRITE BLOCK AND LEAVE

```

```

1290
1291
1292                ;INSERT THE ENTRY,
1293 002004 550646 000020 INSD3: HRRZ AC1,DSKBUF(DEVDAT)
1294 002005 306642 000000 INSD3B: CAIN AC1,(TAC1)          ;BEGINNING OF BLOCK?
1295 002006 254000 002013' JRST INSD3A                  ;YES
1296 002007 275100 000002      SUBI TAC1,2
1297 002010 336020 000002      SKIPN @TAC1                ;SEARCH UFD BLOCK BACKWARDS
1298                                           ; FOR LAST NON-ZERO FILENAME.
1299
1300 002011 254000 002005'      JRST INSD3B
1301
1302 002012 271100 000002      ADDI TAC1,2                ;MOVE IN NAME.
1303 002013 200046 000011 INSD3A: MOVE TAC,DEVFIL(DEVDAT)
1304 002014 202060 000002      MOVEM TAC,@TAC1
1305 002015 271100 000001      ADDI TAC1,1                ;MOVE IN EXTENSION & POINTER.
1306 002016 200046 000012      MOVE TAC,DEVEXT(DEVDAT)
1307                                IFN FTRCHK,<
1308 002017 606040 777777      TRNN TAC,-1                ;POINTER TO BLOCK 0
1309 002020 254200 002021'      HALT ,+1                  ;FOR RETRIEVAL INFO
1310                                           ; CONTINUE-GET BAD INFO MESSAGE
1311
1312 002021 202060 000002      > MOVEM TAC,@TAC1
1313 002022 263140 000000      POPJ PDP,
    
```

```

1314                                     ;CREATE NEW UFD BLOCK AND ADD ENTRY. THEN FIND THE LAST OR ONLY
1315 ;POINTER BLOCK, ADD POINTER IF POSSIBLE. IF NOT POSSIBLE,
1316 ;CREATE NEW POINTER BLOCK AND PUT THE POINTER IN IT.
1317
1318 002023 260140 003314' INS05: PUSHJ PDP,SET000          ;@TAC1 POINTS TO WORD 0 OF BUFFER
1319                                NOSCHEDULE*
1320 002024 700600 001610' CONO PI,DSKOFF
1321 002025 201060 000002 MOVEI TAC,@TAC1          ;POINT TO WORD 0 OF BUFFER
1322 002026 402001 000002 SETZM 2(TAC)          ;CLEAR FOR BLT
1323 002027 205241 000002 MOVSI DAT,2(TAC)          ;SOURCE
1324 002030 541241 000003 HRR1 DAT,3(TAC)          ;DESTINATION
1325 002031 251241 000177 BLT DAT,BLKSI2-1(TAC)      ;CLEAR WORDS 2,-127.
1326                                SCHEDULE*
1327 002032 700600 001617' CONO PI,DSKON
1328
1329 002033 260140 002013' PUSHJ PDP,INS03A          ;INSERT ENTRY IN WORDS 0,1
1330 002034 207006 000015 MOVSS DEVCNT(DEVDAT)      ;SAVE RELATIVE BLOCK# OF THIS UFD BLOCK
1331 002035 260140 003371' PUSHJ PDP,DFGETF          ;GET A FREE BLOCK
1332 002036 202046 000022 MOVEM TAC,SETCNT(DEVDAT) ;SAVE IT
1333
1334                                IFG CHKCNT,<
1335 002037 200046 000020 MOVE TAC,DSKRUF(DEVDAT)
1336 002040 260140 004473' PUSHJ PDP,CHKSUM
1337 002041 506176 000022 HRLM TAC1,SETCNT(DEVDAT)
1338 002042 200046 000022 MOVE TAC,SETCNT(DEVDAT)
1339                                >
1340
1341 002043 260140 004112' INS05A: PUSHJ PDP,MQOUT          ;WRITE THE UFD BLOCK OUT
1342 002044 254000 002100' JRST INS011          ;ERROR
1343
1344 002045 260140 003262' PUSHJ PDP,RR1
1345 002046 260140 003311' INS06: PUSHJ PDP,SET176          ;READ POINTERS BACK IN.
1346 002047 554060 000002 HLRZ TAC,@TAC1          ;LAST OR ONLY BLOCK?
1347 002050 322040 002053' JUMPE TAC,INS07
1348
1349 002051 260140 003263' PUSHJ PDP,RR1A          ;NO, READ NEXT ONE.
1350 002052 254000 002046' JRST INS06
1351
1352 002053 275100 000001 INS07: SUBI TAC1,1          ;YES, ROOM FOR ANOTHER POINTER?
1353 002054 336020 000002 SKIPN @TAC1
1354 002055 364100 002070' SOJA TAC1,INS08          ;YES.
1355
1356 002056 260140 003202' PUSHJ PDP,NEWRI8          ;CREATE NEW POINTER BLOCK.
1357 002057 260140 003314' PUSHJ PDP,SET000          ;SET TAC1 TO FIRST WORD
1358 002060 200046 000022 INS07A: MOVE TAC,SETCNT(DEVDAT) ;GET POINTER SAVED ABOVE
1359 002061 202060 000002 MOVEM TAC,@TAC1          ;STORE POINTER TO NEW UFD BLOCK
1360 002062 260140 003274' PUSHJ PDP,WRI          ;WRITE BLOCK
1361 002063 371000 001756' SOSL AUREQ          ;RELEASE UFD RESOURCE
1362 002064 476000 000000 SETOM AUAVAL
1363 002065 621000 010000 TLZ IOS,AUFLG
1364 002066 202006 000002 MOVEM IOS,DEVIOS(DEVDAT)
1365 002067 263140 000000 POPJ PDP,

```

```

1366                                ;ROOM FOR ANOTHER POINTER IN THIS BLOCK.  FIND LAST POINTER,
1367                                ;INSERT NEW ONE, AND WRITE BLOCK OUT.
1368
1369 002070 336020 000002 INSD8:  SKIPN @TAC1
1370 002071 364100 002070'      SOJA TAC1,INSD8          ;SCAN BACKWARDS FOR LAST POINTER
1371 002072 344100 002060'      AOJA TAC1,INSD7A         ;FOUND IT,  ADD NEW ONE,
1372
1373                                ;GET MORE POINTERS INTO CORE.
1374
1375 002073 603000 000100 INSD10: TLNE IOS,NMP          ;ANY MORE ON DISC?
1376 002074 254000 002023'      JRST INSD5              ;NO
1377 002075 260140 003012'      PUSHJ PDP,GETPTR
1378 002076 200046 000014      MOVE TAC,DEVACC(DEV DAT)
1379 002077 254000 001772'      JRST INSD2
1380
1381                                ;WRITE ERROR, IF WRITE-LOCK, FIX IT,
1382
1383 002100 606000 400000 INSD11: TRNN IOS,IOIMPM
1384 002101 254000 004443'      JRST WERA                ;IT WAS NOT, YOU LOSE.
1385 002102 260140 004562'      PUSHJ PDP,WLERA
1386 002103 542046 000022      HRRM TAC,SETCNT(DEV DAT) ;GET ANOTHER BLOCK
1387 002104 254000 002043'      JRST INSD5A
    
```

```

1388                                     ;OUTPUT A UFD BLOCK, PERFORM CHECKSUM IF NEEDED, WRITE THE BLOCK OUT,
1389                                     ;IF WRITE-LOCK ERROR, TRY ANOTHER BLOCK, FINALLY, IF POINTERS WERE
1390                                     ;ALTERED, WRITE THEM OUT,
1391
1392                                     EXTERNAL AUREQ,AUAVAL
1393
1394 002105                                WUFD:  IFG CHKCNT,<
1395 002105 200046 000020                MOVE TAC,DSKBUF(DEVDAT) ;POINTER TO BUFFER
1396 002106 260140 004473'            PUSHJ PDP,CHKSUM          ;RETURN CHECKSUM IN TAC1
1397                                     >
1398 002107 200046 000014                MOVE TAC,DEVACC(DEVDAT) ;GET POINTER LOCATION
1399                                     IFG CHKCNT,<
1400 002110 506101 000000                HRLM TAC1,(TAC)          ;STORE CHECKSUM IN RETRIEVAL POINTER
1401 002111 205100 200000                MOVSI TAC1,PNTDIF
1402 002112 436106 000010                ORM TAC1,DEVOAD(DEVDAT) ;NOTE THAT POINTERS IN DOB NOT=DISK
1403                                     >
1404                                     IFLE CHKCNT,<
1405                                     MOVE TAC1,DEVOAD(DEVDAT)
1406                                     TLO TAC1,PTROUT
1407                                     TLZ TAC1,PNTDIF
1408                                     MOVEM TAC1,DEVOAD(DEVDAT)
1409                                     >
1410 002113 550041 000000                HRRZ TAC,(TAC)          ;GET BLOCK# FROM RETRIEVAL POINTER
1411
1412 002114 260140 004112' WUFD1:  PUSHJ PDP,MQOUT          ;WRITE UFD BLOCK
1413 002115 254000 002130'            JRST WUFD3              ;WRITE ERROR
1414 002116 200046 000010                MOVE TAC,DEVOAD(DEVDAT)
1415 002117 661000 000100                TLO IOS,NMP           ;SET NMP
1416 002120 202006 000002                MOVEM IOS,DEVIOS(DEVDAT) ;SO DFO4A WON'T READ POINTERS
1417 002121 603040 200000                TLNE TAC,PNTDIF       ;WERE POINTERS CHANGED?
1418 002122 260140 003131'            PUSHJ PDP,DFO4A         ;YES, WRITE THEM OUT,
1419 002123 371000 002063'            SOSL AUREQ              ;RELEASE UFD RESOURCE
1420 002124 476000 002064'            SETOM AUAVAL
1421 002125 621000 010000                TLZ IOS,AUFLG         ;NOTE THAT UFD RELEASED
1422 002126 202006 000002                MOVEM IOS,DEVIOS(DEVDAT)
1423 002127 263140 000000                POPJ PDP,
1424
1425 002130 200106 000021 WUFD3:  MOVE TAC1,DSKCNT(DEVDAT);WAS ERROR WRITE-LOCK?
1426 002131 606100 400000                TRNN TAC1,IOIMPM
1427 002132 254000 004443'            JRST WERA
1428 002133 620000 400000                TRZ IOS,IOIMPM        ;YES
1429 002134 202006 000002                MOVEM IOS,DEVIOS(DEVDAT)
1430 002135 205100 200000                MOVSI TAC1,PNTDIF
1431 002136 436106 000010                ORM TAC1,DEVOAD(DEVDAT)
1432 002137 260140 004562'            PUSHJ PDP,WLERA         ;FREE CURRENT ASSIGNED BLOCK AND GET A NEW ONE
1433 002140 200106 000014                MOVE TAC1,DEVACC(DEVDAT)
1434 002141 542042 000000                HRRM TAC,(TAC1)
1435 002142 254000 002114'            JRST WUFD1
    
```

```

1436          SUBTTL ACCESS TABLE PROCESSING
1437          ;SCAN ACCESS TABLE,
1438          ;EXIT TO CALL+1 WITH TAC SET TO THE LAST ENTRY IF NOT THERE,
1439          ;EXIT TO CALL+2 WITH TAC SET TO CORRECT ENTRY IF THERE,
1440
1441          EXTERNAL CPOPJ1,CPOPJ
1442
1443 002143 200703 777777 SCANAT: MOVE AC2,-1(PDP)
1444 002144 336040 000010' SCANAT0: SKIPN TAC,FAT          ;ANY AT ALL?
1445 002145 263140 000000 SCNHLT: POPJ PDP,                ;NO, LEAVE (HALT PC HERE IF ATTEMPT
1446                                     ; TO CLEAR NON-EXISTANT ACCESS TABLE)
1447
1448 002146 312701 000000 SCNAT1: CAME AC2,ATPP(TAC)          ;PROJ,-PROG, THE SAME?
1449 002147 254000 002157' JRST SCNAT2                      ;NO
1450
1451 002150 200246 000011          MOVE DAT,DEVFIL(DEV DAT) ;NAMES THE SAME?
1452 002151 312241 000001          CAME DAT,ATNAME(TAC)
1453 002152 254000 002157' JRST SCNAT2
1454
1455 002153 510241 000002          HLLZ DAT,ATEXT(TAC)      ;EXTENSION THE SAME?
1456 002154 510746 000012          HLLZ AC3,DEVEXT(DEV DAT)
1457 002155 316240 000017          CAMN DAT,AC3
1458 002156 254000 001446' JRST CPOPJ1                      ;YES
1459
1460          ;CONTINUE SCAN FROM CURRENT ENTRY
1461
1462
1463 002157 550241 000003 SCNAT2: HRRZ DAT,ATLINK(TAC)        ;ANY MORE ENTRIES?
1464 002160 322240 001607' JUMPE DAT,CPOPJ                  ;NO
1465 002161 200040 000005          MOVE TAC,DAT              ;YES, GO TO THE NEXT ONE
1466 002162 254000 002146' JRST SCNAT1
1467

```

```

1468          ;CLEAR AN ACCESS ENTRY,
1469          ;IT IS ASSUMED THAT NO SCHEDULING WILL TAKE PLACE,
1470          ;ENTER WITH TAC POINTING TO THE ENTRY TO WIPE OUT,
1471
1472
1473 002163          CLRAT: IFN      FTRCHK,<
1474                  SKIPN TAC          ;ATTEMPT TO CLEAR NON EXISTENT ACCESS TABLE IS AN ERROR
1475 002163 336000 000001          ;CONTINUE WILL POPJ AND EXIT,
1476 002164 254200 002145'      HALT SCNHLT
1477          >
1478 002165 553006 000014          HRRZS DEVACC(DEV DAT)      ;CLEAR ACCESS TABLE POINTER IN DEVICE DATA BLOCK
1479 002166 201700 000010'      MOVEI AC2,FAT              ;START AT BEGINNING OF LINKED ACCESS TABLE ENTRIES
1480 002167 275700 000003          SURI AC2,ATLINK
1481
1482 002170 200640 000016          CLRAT1: MOVE AC1,AC2          ;PICKUP THE NEXT ACCESS TABLE
1483 002171 550715 000003          HRRZ AC2,ATLINK(AC1)
1484          IFN      FTRCHK,<
1485 002172 336000 000016          SKIPN AC2          ;ZERO LINK INDICATES END OF TABLE WITHOUT
1486          ;FINDING ENTRY TO BE CLEARED
1487 002173 254200 002214'      HALT SETHLT          ;CONTINUING AFTER THIS ERROR WILL CONTINUE
1488          ; NORMALLY (IE POPJ)
1489          >
1490 002174 312700 000001          CAME AC2,TAC              ;DOES THIS ENTRY LINK TO THE ONE TO BE CLOBBERED?
1491 002175 254000 002170'      JRST CLRAT1          ;NO,CONTINUE SEARCH
1492
1493 002176 200701 000003          MOVE AC2,ATLINK(TAC)      ;YES, FOUND IT
1494 002177 542715 000003          HRRM AC2,ATLINK(AC1)      ;LINK AROUND IT
1495 002200 254000 004025'      JRST CLR COR          ;CLEAR THE CORE
1496
1497

```

```

1498                                ;FIND A HOLE FOR ACCESS TABLE ENTRY AND SET A LINK,
1499                                ;BUILD UP THE ENTRY
1500                                ;UPON ENTRY, DAT SHOULD CONTAIN TEST BITS TO SET
1501
1502
1503 002201 260140 003731' SETAT:  PUSHJ PDP,GETFCR                                ;TAC1 POINTS TO 4 WORD BLOCK
1504 002202 500246 000012          HLL DAT,DEVEXT(DEVDAT)
1505 002203 202242 000002          MOVEM DAT,ATEXT(TAC1)
1506 002204 200246 000011          MOVE DAT,DEVFIL(DEVDAT)                        ;STORE EXTENSION
1507 002205 202242 000001          MOVEM DAT,ATNAME(TAC1)
1508 002206 200243 777777          MOVE DAT,-1(PDP)                                ;STORE FILE NAME
1509 002207 202242 000000          MOVEM DAT,ATPP(TAC1)                          ;XWD PROJ,PROG
1510 002210 500246 000022          HLL DAT,SETCNT(DEVDAT)                        ;STORE XWD PROJ,PROG
1511 002211 540240 000010'          HRR DAT,FAT                                ;LOGICAL BLOCK# OF UFD RIB
1512 002212 202242 000003          MOVEM DAT,ATLINK(TAC1)                       ;POINTER TO NEXT ENTRY
1513 002213 552100 000010'          HRRZM TAC1,FAT                            ;STORE LINK
1514 002214 263140 000000          SETHLT: POPJ PDP,                            ;POINT FAT TO THIS NEW ENTRY
1515                                ;HALT PC HERE IF END OF TABLE WITHOUT FINDING
                                ;ENTRY TO BE CLEARED

```

```

1516          SUBTTL DUMP INPUT/OUTPUT
1517          IDUMP MODE OUTPUT
1518
1519          EXTERNAL COMCHK,ADRERR,SAVDOL
1520
1521 002215 607300 020000 DFMPO: TLNN DEVDAT,ENTRB      IENTER YET?
1522 002216 254000 004355' JRST DFERR2              INO, UNDEFINED FILE
1523
1524 002217 661000 000020      TLO IOS,IO              IOUTPUT STATUS
1525 002220 202006 000002      MOVEM IOS,DEVIOS(DEVDAT)
1526 002221 260140 000000      PUSHJ PDP,COMCHK          ICHECK IOWDS
1527 002222 254000 000000      JRST ADRERR              IERROR
1528
1529 002223 336060 000014 DFD00: SKIPN TAC,@UO          IPICK UP NEXT IOWD
1530 002224 263140 000000      POPJ PDP,                IIT WAS ZERO, LEAVE
1531 002225 603040 777777      TLNE TAC,-1             ILH ZERO?
1532 002226 254000 002231' JRST DFD00A              INO
1533 002227 540600 000001      HRR UO,TAC              IYES, ANOTHER TABLE
1534 002230 254000 002223' JRST DFD00
1535
1536 002231 271040 000001 DFD00A: ADDI TAC,1             IFORM XWD - LENGTH, FIRST-ADDRESS
1537 002232 261140 000001      PUSH PDP,TAC             ISAVE IT
1538
1539 002233 260140 003414' DFD01: PUSHJ PDP,UPDA          ISET DAREQ INTERLOCK (SAT BLOCKS MAY ONLY
1540                                I BE MANIPULATED BY ONE USER AT A TIME!)
1541 002234 260140 003421' DFD01A: PUSHJ PDP,SATGET        IACQUIRE A NON-FULL SAT BLOCK
1542                                I ITEM IS DESTROYED IF SATGET CALLS MQIN,MQOUT
1543 002235 574203 000000      HLRE ITEM,(PDP)           IITEM = CLH OF IOWD/200
1544 002236 217000 000004      MOVMS ITEM
1545 002237 271200 000177      ADDI ITEM,BLKSIZE-1       IMAKE E.G. 129 WORDS USE 2 DISK BLOCK
1546 002240 240200 777771      ASH ITEM,-BLKP2          INUMBER OF BLOCKS REQUIRED FOR THIS WRITE

```

1547	002241	550640	000015'	DFD02A:	HRRZ AC1,SATPTR	
1548	002242	200700	000236'		MOVE AC2,SATBK2	
1549	002243	260140	003744'	DFD01B:	PUSHJ PDP,GETBIT	IASK FOR C(ITEM BITS)
1550	002244	254000	002402'		JRST DFD06	I NOT AVAILABLE
1551						
1552	002245	272220	000015'		ADDM ITEM,@SATPTR	I INCREMENT COUNT
1553	002246	476000	000017'		SETOM SATCHG	ISAT BLOCK HAS BEEN CHANGED
1554	002247	554260	000015'		HLRZ DAT,@SATPTR	I COMPUTE BLOCK NUMBER LESS 1.
1555					SCHEDULE+	
1556	002250	700600	002032'		CONO PI,DSKON	
1557	002251	260140	003404'		PUSHJ PDP,DOWNDA	I RESET DAREQ INTERLOCK.
1558	002252	271242	777776		ADDI DAT,-2(TAC1)	
1559	002253	261140	000005		PUSH PDP,DAT	I AND SAVE BLOCK NUMBER LESS 1
1560	002254	261140	000004		PUSH PDP,ITEM	ISAVE COUNT OF BLOCKS WRITTEN
1561	002255	200103	777776		MOVE TAC1,-2(PDP)	I SET UP AN IOWD (SAVED AT DFD00A)
1562	002256	202106	000020		MOVEM TAC1,DSKBUF(DEVDAT)	
1563	002257	210040	000004		MOVN TAC,ITEM	
1564	002260	242040	000007		LSH TAC,BLKP2	
1565	002261	577000	000002		HLRES TAC1	
1566	002262	311040	000002		CAML TAC,TAC1	
1567	002263	506046	000020		HRLM TAC,DSKBUF(DEVDAT)	I ONLY A PARTIAL WRITE
1568	002264	574106	000020		HLRE TAC1,DSKBUF(DEVDAT)	
1569	002265	217000	000002		MOVMS TAC1	
1570	002266	260140	004534'		PUSHJ PDP,UPDEV	I UPDATE SIZE OF FILE
1571	002267	272206	000022		ADDM ITEM,SETCNT(DEVDAT)	I NON UPDATE RELATIVE BLOCK# WITHIN THE FILE

```

1572 002270 200106 000020      MOVE TAC1,DSKBUF(DEVDAT) ;PICK UP ADDRESS AND WORD COUNT
1573 002271 627100 000177      TLZN TAC1,BLKSI2-1      ;AN EVEN NUMBER OF DISK BLOCKS TO BE WRITTEN?
1574 002272 254000 002327'     JRST DF002G      ;YES, DON'T WORRY ABOUT PARTIAL BLOCK
1575                                ;HANDLING PROBLEMS
1576 002273 261140 000002      PUSH PDP,TAC1      ;NO, SAVE ADDRESS WITH COUNT OF FULL BLOCKS-1
1577                                ; TO BE WRITTEN
1578 002274 261146 000020      PUSH PDP,DSKBUF(DEVDAT) ;SAVE ORIGINAL ADDRESS ;EXACT WORD COUNT ALSO
1579 002275 260140 003620'     PUSHJ PDP,SETBUF      ;GET MONITOR BUFFER
1580 002276 200126 000020      MOVE TAC1,DSKBUF(DEVDAT) ;TAC1 GETS ADDRESS OF MONITOR BUFFER
1581                                ; (RELOCATION NOT NECESSARY)
1582 002277 262140 000001      POP PDP,TAC      ;USER'S INITIAL ADDR AND WC
1583 002300 260140 000000      PUSHJ PDP,SAVDDL      ;ADJUST IN CASE THIS IS A SAVE OF A
1584                                ; HIGH SEGMENT, WHICH WAS MOVED IN CORE
1585                                ; DURING I/O WAIT,
1586 002301 200640 000001      MOVE AC1,TAC
1587 002302 550043 777776      HRRZ TAC,-2(PDP)      ;GET INITIAL BLOCK NUMBER-1
1588 002303 270640 004607'     DF002C: ADD AC1,CXWD BLKSI2,PLKSI2 ;SET UP ADDRESS, WORD COUNT, AND LOGICAL
1589 002304 350000 000001      AOS TAC      ;BLOCK NUMBER FOR THE LAST (PARTIAL) BLOCK
1590 002305 321640 002303'     JUMPL AC1,DF002C      ;OF THIS DUMP MODE OUTPUT REQUEST,
1591 002306 274640 004607'     SUB AC1,CXWD BLKSI2,PLKSI2
1592 002307 500700 000015      HLL AC2,AC1      ;SAVE LH
1593 002310 271647 000000      ADDI AC1,(PROG)      ;RELOCATE USER'S ADDRESS
1594 002311 500640 000016      HLL AC1,AC2      ;IN CASE OF OVERFLOW INTO LH(2 SEG SAVE)
1595 002312 200715 000000      DF002D: MOVE AC2,(AC1)      ;MOVE DATA FROM END OF USER'S
1596 002313 202702 000000      MOVEM AC2,(TAC1)      ;OUTPUT AREA OT MONITOR BUFFER
1597 002314 252100 002314'     AOBJP TAC1,,
1598 002315 253640 002312'     AOBJN AC1,DF002D
1599 002316 402002 000000      DF002E: SETEM (TAC1)      ;ZERO OUT THE REMAINDER OF THE MONITOR BUFFER
1600 002317 253100 002316'     AOBJN TAC1,DF002E
1601 002320 260140 004112'     PUSHJ PDP,MQOUT      ;WRITE OUT THE PARTIALLY FULL MONITOR BUFFER
1602                                ; WHICH CONTAINS THE LAST BLOCK OF THE DUMP MODE
1603                                ; MODE OUTPUT REQUEST
1604 002321 255000 000000      JFCL      ;IGNORE ERROR RETURN
1605 002322 260140 003654'     PUSHJ PDP,CLRBUF      ;RELINQUISH THE MONITOR BUFFER
1606 002323 262146 000020      POP PDP,DSKBUF(DEVDAT) ;RETRIEVE ORIGINAL ADDRESS
1607 002324 205100 000200      MOVSI TAC1,BLKSI2      ;WITH WORD COUNT THAT HAS LOW-ORDER BITS CLEARED
1608 002325 273106 000020      ADDB TAC1,DSKBUF(DEVDAT) ;ADD IN BLOCK SIZE TO DETERMINE NUMBER
1609                                ; NUMBER OF FULL BLOCKS YET TO BE WRITTEN
1610 002326 327100 002336'     JUMPG TAC1,DF002B      ;IF NO FULL BLOCKS TO BE WRITTEN SKIP OVFR
1611                                ; OUTPUT SEQUENCE
1612 002327 200046 000020      DF002G: MOVE TAC,DSKBUF(DEVDAT) ;ORIGINAL ADDRESS AND WORD COUNT
1613 002330 260140 002300'     PUSHJ PDP,SAVDDL      ;ADJUST IN CASE OF SAVE OF HIGH SEG
1614                                ; WHICH HAS MOVED DURING IO WAIT
1615 002331 202046 000020      MOVEM TAC,DSKBUF(DEVDAT) ;STORE BACK IN DDB
1616 002332 200043 777777      MOVE TAC,-1(PDP)      ;WRITE THE BLOCKS
1617 002333 271040 000001      ADDI TAC,1
1618 002334 260140 004112'     PUSHJ PDP,MQOUT
1619 002335 254000 002407'     JRST DF007      ;ERROR
1620 002336 262140 000004      DF002B: POP PDP,ITEM      ;RESTORE COUNT
1621                                ;TOSS OUT A SERIES OF CONSECUTIVE POINTERS
1622
1623
1624 002337 550706 000014      HRRZ AC2,DEVACC(DEVDAT) ;GET RETRIEVAL POINTER POINTER

```

```

1625
1626 002340 200660 000016 DFD03A: MOVE AC1,@AC2      ;IS POINTER ALREADY THERE?
1627 002341 271700 000001      ADDI AC2,1
1628 002342 602640 777777      TRNE AC1,-1
1629 002343 254000 002422'      JRST DFD09          ;YES
1630
1631 002344 307706 000037      CAIG AC2,PTRN(DEVDAT) ;NO, IS THERE A NEXT ONE?
1632 002345 402020 000016      SETZM @AC2          ;CLEAR NEXT POINTER
1633 002346 661000 000100      TLO IOS,NMP        ;FLAG END OF POINTERS
1634 002347 202006 000002      MOVEM IOS,DEVIOS(DEVDAT)
1635 002350 350243 000000      DFD03B: AOS DAT,(PDP)      ;PUT A POINTER IN CORE
1636 002351 552256 777777      HRRZM DAT,-1(AC2)      ;STORE NEW POINTER IN DOB
1637 002352 205240 200000      MOVSI DAT,PNTDIF
1638 002353 436246 000010      ORM DAT,DEVOAD(DEVDAT) ;SHOW THAT POINTERS IN DOB NOT=DISK
1639
1640      IFG CHKCNT,<          ;GET CHECKSUM
1641 002354 200043 777777      MOVE TAC,-1(PDP)
1642 002355 260140 002330'      PUSHJ PDP,SAVDDL      ;ADJUST ADR IN CASE OF HIGH SEG SAVE
1643 002356 260140 004473'      PUSHJ PDP,CHKSUM
1644 002357 506116 777777      HRLM TAC1,-1(AC2)
1645      >
1646
1647 002360 303706 000037      CAILE AC2,PTRN(DEVDAT) ;MORE POINTERS?
1648 002361 254000 002373'      JRST DFD04          ;LAST POINTER IN CORE
1649
1650 002362 200100 004607'      DFD03C: MOVE TAC1,[XWD BLKSIZ,BLKSIZ];INCREMENT IOWD
1651 002363 272103 777777      ADDM TAC1,-1(PDP)
1652 002364 367200 002340'      SOJG ITEM,DFD03A      ;ANY MORE?
1653 002365 262140 000005      POP PDP,DAT
1654 002366 542706 000014      HRRM AC2,DEVACC(DEVDAT)
1655 002367 335003 000000      SKIPGE (PDP)
1656 002370 254000 002233'      JRST DFD01          ;NOT FINISHED, GET NEXT SET
1657 002371 262140 000001      POP PDP,TAC          ;BACK UP PDP
1658 002372 344600 002223'      AOJA UUO,DFD00      ;GO BACK FOR MORE

```

```

1659                                     ;WRITE POINTERS ONTO DISK
1660
1661 002373 506203 000000 DFD04: HRLM ITEM,(PDP)          ;SAVE COUNT OF BLOCKS TO WRITE.
1662 002374 260140 003620' PUSHJ PDP,SETBUF          ;FIND SOME 200 WORD AREA
1663 002375 260140 003131' PUSHJ PDP,DF04A          ;WRITE THEM
1664 002376 260140 003654' PUSHJ PDP,CLRBUF          ;CLEAR THE AREA
1665 002377 201706 000023 MOVEI AC2,PTR1(DEVDAT) ;RESET AC2 TO FIRST POINTER
1666 002400 554203 000000 HLRZ ITEM,(PDP)          ;RESTORE COUNT OF BLOCKS TO WRITE
1667 002401 254000 002362' JRST DFD03C
1668
1669 002402 242200 777777 DFD06: LSH ITEM,-1          ;TRY FOR 1/2 EARLIER REQUEST
1670 002403 326200 002243' JUMPN ITEM,DFD01B        ;CONTINUE IF NON-ZERO
1671 002404 201040 011000 MOVEI TAC,NUMBIT        ;BITS/SAT BLOCK
1672 002405 542060 000015' HRRM TAC,@SATPTR        ;MARK SAT ENTRY FULL
1673 002406 254000 002234' JRST DFD01A          ;TRY ANOTHER SAT BLOCK
1674
1675                                     ;ERROR WHILE WRITING, IF WRITE-LOCK, FIX IT.
1676
1677 002407 550046 000021 DFD07: HRRZ TAC,DSKCNT(DEVDAT) ;WRITE-LOCK?
1678 002410 606040 400000 TRNN TAC,I0IMPM
1679 002411 254000 002336' JRST DFD02B          ;NO, TOO BAD.
1680
1681 002412 554046 000021 HLRZ TAC,DSKCNT(DEVDAT)
1682 002413 260140 004565' PUSHJ PDP,SETWL          ;SET WRITE-LOCK BIT IN SAT ENTRY.
1683
1684 002414 350043 777777 DFD08: AOS TAC,-1(PDP)
1685 002415 260140 003464' PUSHJ PDP,SETFRE
1686 002416 373003 000000 SOSLE (PDP)
1687 002417 254000 002414' JRST DFD08
1688 002420 274140 004610' SUB PDP,[XWD 2,2]
1689 002421 254000 002233' JRST DFD01          ;TRY AGAIN.
1690
1691                                     ;WE ARE WRITING IN MIDDLE OF FILE, FREE THE OLD BLOCK.
1692
1693 002422 506203 000000 DFD09: HRLM ITEM,(PDP)          ;SAVE BLOCK COUNT WITH BLOCK#
1694 002423 261140 000016 PUSH PDP,AC2          ;SAVE POINTER TO RETRIEVAL INFO IN DOB
1695 002424 550056 777777 HRRZ TAC,-1(AC2)          ;GET OLD BLOCK# FROM DOB
1696 002425 260140 003464' PUSHJ PDP,SETFRE
1697 002426 262140 000016 POP PDP,AC2          ;RESTORE POINTER INTO RETRIEVAL INFO
1698 002427 554203 000000 HLRZ ITEM,(PDP)          ;RESTORE COUNT OF BLOCKS TO WRITE THIS ITERATION
1699 002430 254000 002350' JRST DFD03B

```

```

1700                                ;DUMP MODE INPUT.
1701
1702                                EXTERNAL COMCHK,ADRERR
1703
1704 002431 607300 040000 DFDMP1: TLNN DEVDAT,LOOKB      ;LOOKUP YET?
1705 002432 254000 004355' JRST DFERR2                ;NO, UNDEFINED FILE
1706 002433 603000 000040' TLNE IOS,IOEND              ;END-FILE?
1707 002434 254000 002501' JRST DFD11C                ;YES
1708 002435 621000 000020' TLZ IOS,IO                  ;INPUT STATUS
1709 002436 202006 000002' MOVEM IOS,DEVIOS(DEV DAT)    ;CHECK IOWDS
1710 002437 260140 002221' PUSHJ PDP,COMCHK            ;ERROR
1711 002440 254000 002222' JRST ADRERR                  ;PICK UP NEXT IOWD
1712 002441 336060 000014' DFD10: SKIPN TAC,@UUO        ;NO MORE, LEAVE.
1713 002442 263140 000000' POPJ PDP,                    ;LH ZERO?
1714 002443 603040 777777' TLNE TAC,-1                ;NO
1715 002444 254000 002447' JRST DFD10A                 ;YES, ANOTHER LIST.
1716 002445 540600 000001' HRR UUO,TAC
1717 002446 254000 002441' JRST DFD10
1718
1719 002447 271040 000001' DFD10A: ADDI TAC,1            ;FROM XWD -L,FIRST-ADDRESS
1720 002450 261140 000001' PUSH PDP,TAC                ;SAVE IT
1721
1722 002451 554100 000001' HLRZ TAC1,TAC                ;GET LENGTH FROM IOWD
1723 002452 305100 777600' CAIGE TAC1,NBLKSZ           ;LESS THAN 1 BLOCK?
1724 002453 505040 777600' HRLI TAC,NBLKSZ            ;YES, MUST WRITE AT LEAST 1 FULL BLOCK
1725 002454 261140 000001' PUSH PDP,TAC                ;IOWD FOR FIRST BLOCK
1726
1727 002455 550706 000014' HRRZ AC2,DEVACC(DEV DAT)    ;PICK UP FIRST POINTER
1728 002456 303706 000037' CAILE AC2,PTRN(DEV DAT)     ;ANY LEFT IN CORE?
1729 002457 260140 002574' PUSHJ PDP,DFD19             ;NO, GET SOME MORE
1730 002460 336020 000016' SKIPN @AC2                 ;IS A POINTER THERE?
1731 002461 254000 002500' JRST DFD11B                ;NO, EXIT
1732 002462 261160 000016' PUSH PDP,@AC2              ;SAVE THE POINTER
1733
1734 002463 200040 004607' MOVE TAC,[XWD BLKSIZ,BLKSIZ] ;INCREMENT IOWD
1735 002464 272043 777776' ADDM TAC,-2(PDP)
1736 002465 350006 000014' AOS DEVACC(DEV DAT)         ;NEXT POINTER
1737 002466 350006 000022' AOS SETCNT(DEV DAT)         ;NEXT RELATIVE BLOCK

```

```

1738 002467 335003 777776 DFDI1: SKIPGE -2(PDP)      ;MORE TO GO?
1739 002470 254000 002504 JRST DFDI2          ;YES
1740
1741 002471 200103 777777 DFDI1A: MOVE TAC1,-1(PDP)    ;NO, PROCESS THE FIRST BLOCK
1742 002472 202106 000020 MOVEM TAC1,DSKBUF(DEVDAT) ;SAY WHERE TO READ DATA
1743 002473 200043 000000 MOVE TAC,(PDP)      ;GET FIRST LOGICAL BLOCK#
1744 002474 260140 004110 PUSHJ PDP,MQIN
1745 002475 255000 000000 JFCL
1746                                ;ERRORS IGNORED.
1747
1748 REPEAT 0, < ;TEMPORARILY DELETE THIS CODE DUE TO SAVE-GET PROBLEM.
1749 IFG CHKCNT,< ;COMPARE CHECKSUM
1750 MOVE TAC,-1(PDP)
1751 PUSHJ PDP,CHKSUM
1752 HLRZ TAC,(PDP)
1753 MOVEI AC1,IODTER
1754 CAME TAC,TAC1
1755 ORM AC1,DEVIOS(DEVDAT)
1756 MOVE IOS,DEVIOS(DEVDAT)
1757 >
1758
1759 002476 274140 004611 SUB PDP,[XWD 3,3]      ;REMOVE INTERMEDIATE STORAGE FROM PDL
1760 002477 344600 002441 AOJA UUO,DFDI0      ;GO BACK FOR MORE,
1761
1762 002500 274140 004610 DFDI1B: SUB PDP,[XWD 2,2]    ;BACK UP PDP, REMOVE XWD AND IOWD
1763 002501 200000 004612 DFDI1C: MOVE IOS,[XWD IOEND,IOEND] ;NO MORE INPUT
1764 002502 437006 000002 ORB IOS,DEVIOS(DEVDAT)
1765 002503 263140 000000 POPJ PDP,

```

```

1766                                ;DUMP INPUT CONTINUED,
1767                                ;FIND A SERIES OF CONSECUTIVE BLOCKS TO READ AT ONCE
1768
1769 002504 550706 000014 DFD12: HRRZ AC2,DEVACC(DEVDAT) ;GET NEXT POINTER
1770 002505 303706 000037 CAILE AC2,PTRN(DEVDAT) ;ANY MORE POINTERS IN CORE?
1771 002506 260140 002574' PUSHJ PDP,DFD19 ;NO GET SOME MORE
1772 002507 336020 000016 SKIPN @AC2 ;YES
1773 002510 254000 002571' JRST DFD18 ;END-FILE
1774 002511 261140 000016 PUSH PDP,AC2 ;SAVE PONTNER TO POINTER
1775 002512 554043 777775 HLRZ TAC,-3(PDP) ;GET LENGTH OF REMAINING INPUT
1776 002513 550116 000000 HRRZ TAC1,(AC2) ;GET BLOCK# FROM DOB
1777
1778 002514 271040 000200 DFD13: ADDI TAC,BLKSI2 ;ANY MORE TO READ?
1779 002515 603040 000001 TLNE TAC,1
1780 002516 254000 002531' JRST DFD15 ;NO
1781 002517 271700 000001 ADDI AC2,1
1782 002520 307706 000037 CAIG AC2,PTRN(DEVDAT)
1783 002521 336760 000016 SKIPN AC3,@AC2 ;END-FILE?
1784 002522 254000 002525' JRST DFD14 ;YES

1785 002523 306117 777777 CAIN TAC1,-1(AC3) ;STILL CONSECUTIVE?
1786 002524 344100 002514' AOJA TAC1,DFD13 ;YES, LOOP
1787
1788 002525 200043 000000 DFD14: MOVE TAC,(PDP) ;COMPUTE COUNT
1789 002526 274040 000016 SUB TAC,AC2 ;DIFFERENCE BETWEEN POINTERS IS # OF BLOCKS
1790 002527 242040 000007 LSH TAC,BLKP2 ;COMPUTE WORDS
1791 002530 254000 002533' JRST DFD16
1792 002531 574043 777775 DFD15: HLRZ TAC,-3(PDP) ;GET LENGTH OF INPUT
1793 002532 271700 000001 ADDI AC2,1
1794 002533 550106 000014 DFD16: HRRZ TAC1,DEVACC(DEVDAT)
1795 002534 275116 000000 SUBI TAC1,(AC2)
1796 002535 217000 000002 MOVMS TAC1
1797 002536 272106 000022 ADDM TAC1,SETCNT(DEVDAT)
1798 002537 542706 000014 HRRM AC2,DEVACC(DEVDAT) ;STORE NEW POINTER POINTER
1799 002540 204100 000001 MOVS TAC1,TAC
1800 002541 506043 000000 HRLM TAC,(PDP)
1801 002542 540103 777775 HRR TAC1,-3(PDP)
1802 MOVE TAC,(PDP) ;GET POINTER TO FIRST RETRIEVAL POINTER FOR THIS INPUT
1803
1804 002543 200043 000000 MOVE TAC,(TAC) ;GET FIRST LOGICAL BLOCK#
1805 002545 202106 000020 MOVEM TAC1,DSKBUF(DEVDAT) ;STORE CONTROLLING IOWD
1806 002546 260140 004110' PUSHJ PDP,MQIN ;REQUEST THE INPUT
1807 002547 255000 000000 JFCL ;ERRORS IGNORED

```

```

1808                                ;DUMP INPUT CONTINUED.
1809                                ;IF THERE IS CHECKSUMMING, CHECK IT, ELSE INCREMENT IOWD
1810                                ;AND LOOP,
1811
1812 002550 262140 000016          POP PDP,AC2
1813                                IFLE CHKCNT,<          ;NO CHECKSUMMING
1814                                HLRS AC2
1815                                SETCA AC2,
1816                                AOBJN AC2,,+1
1817                                ADDM AC2,-2(PDP)
1818                                >
1819
1820                                IFG CHKCNT,<          ;CHECK CHECKSUMS
1821 002551 574100 000016          HLRE TAC1,AC2          ;CHANGE WORD COUNT TO BLOCKS
1822 002552 217000 000002          MOVMS TAC1
1823 002553 271100 000177          ADDI TAC1,BLKSI2-1
1824 002554 242100 777771          LSH TAC1,-BLKP2
1825 002555 213000 000002          MOVNS TAC1
1826 002556 504700 000002          HRL AC2,TAC1
1827 002557 200043 777776          DFDI7: MOVE TAC,-2(PDP)
1828 002560 260140 004473'        PUSHJ PDP,CHKSUM
1829 002561 554056 000000          HLRZ TAC,(AC2)
1830 002562 201740 100000          MOVEI AC3,IODTER
1831 002563 312040 000002          CAME TAC,TAC1
1832 002564 260140 004511'        PUSHJ PDP,CKREC2
1833 002565 200740 004607'        MOVE AC3,CXWD BLKSI2,BLKSI2]
1834 002566 272743 777776          ADDM AC3,-2(PDP)
1835 002567 253700 002557'        AOBJN AC2,DFDI7
1836                                >
1837
1838 002570 254000 002467'        JRST DFDI1          ;SEE IF MORE INPUT REQUIRED
1839
1840 002571 200000 004612'        DFDI8: MOVE IOS,CXWD IOEND,IOEND] ;NO MORE INPUT
1841 002572 437006 000002          ORB IOS,DEVIOS(DEVDAT)
1842 002573 254000 002471'        JRST DFDI1A
1843
1844 002574 603000 000100        DFDI9: TLNE IOS,NMP
1845 002575 254000 002603'        JRST DFDI10          ;NO
1846 002576 260140 003620'        PUSHJ PDP,SETBUF      ;GET BUFFER
1847 002577 260140 003012'        PUSHJ PDP,GETPTR      ;READ AND COPY POINTERS INTO DOB
1848 002600 260140 003654'        PUSHJ PDP,CLRBUF      ;CLEAR BUFFER
1849 002601 550706 000014          HRRZ AC2,DEVACC(DEVDAT)
1850 002602 263140 000000          POPJ PDP,
1851
1852 002603 201706 000023        DFDI10: MOVEI AC2,PTR1(DEVDAT) ;NO MORE POINTERS ON DISK
1853 002604 542706 000014          HRRM AC2,DEVACC(DEVDAT)
1854 002605 402020 000016          SETZM @AC2
1855 002606 263140 000000          POPJ PDP,

```

```

1856          SUBTTL USETO/USETI
1857          ISETO AND SETI UUOS
1858
1859          EXTERNAL WAIT1
1860
1861 002607 260140 002611' DFSET: PUSHJ PDP,+,+2
1862 002610 263140 000000      POPJ PDP,
1863 002611 603300 060000      TLNE DEVDAT,ENTRB+LOOKB ;FILE OPEN?
1864 002612 606600 777777      TRNN UUO,-1          ;NON-ZERO BLOCK NUMBER ARGUMENT?
1865 002613 254000 004355'      JRST DFERR2          ;NO, ERROR IN EITHER CASE
1866 002614 260140 001505'      PUSHJ PDP,WAIT1        ;WAIT FOR I/O TO COMPLETE
1867 002615 260140 003647'      PUSHJ PDP,SETB*6        ;SET UP A BUFFER
1868 002616 200000 004612'      MOVE IOS,[XWD IOEND,IOEND] ;CLEAR EOF
1869 002617 413006 000002      ANDCAB IOS,DEVIOS(DEV DAT)
1870
1871 002620 550106 000014      HRRZ TAC1,DEVACC(DEV DAT)
1872 002621 550046 000022      HRRZ TAC,SETCNT(DEV DAT) ;C(UUO) RH > C(SETCNT) RH?
1873 002622 301054 000000      CAIL TAC,(UUO)          ;I.E, SHOULD WE GO FORWARD?
1874 002623 254000 002676'      JRST DFSET6          ;NO, GO BACKWARD
1875
1876 002624 275106 000037      SUBI TAC1,PTRN(DEV DAT) ;YES, DESIRED POINTER IN CORE?
1877 002625 274040 000002      SUB TAC,TAC1
1878 002626 301054 000000      CAIL TAC,(UUO)
1879 002627 254000 002660'      JRST DFSET3          ;YES
1880
1881 002630 550106 000015      HRRZ TAC1,DEV CNT(DEV DAT)
1882 002631 271100 000177      ADDI TAC1,BLKSI2-1
1883 002632 242100 777771      LSH TAC1,-BLKP2
1884 002633 315100 000001      CAMGE TAC1,TAC          ;ANY MORE POINTERS ON DISK?
1885 002634 254000 002660'      JRST DFSETX          ;NO
1886 002635 542046 000022      HRRM TAC,SETCNT(DEV DAT)
1887 002636 200046 000010      MOVE TAC,DEVQAD(DEV DAT) ;YES, ARE POINTERS IN CORE THE SAME
1888 002637 607040 200000      TLNN TAC,PNTDIF        ;AS ON DISC?
1889 002640 254000 002645'      JRST DFSET1          ;YES
1890 002641 661000 000100      TLO IOS,NMP            ;SET "DO NOT READ NEXT POINTERS"
1891 002642 202006 000002      MOVEM IOS,DEVIOS(DEV DAT)
1892 002643 260140 003131'      PUSHJ PDP,DF04A        ;NO, WRITE THEM OUT
1893 002644 625000 000100      TLZA IOS,NMP            ;NO NEED TO READ
1894
1895 002645 260140 003262' DFSET1: PUSHJ PDP,RR1          ;READ THEM IN
1896 002646 202006 000002      MOVEM IOS,DEVIOS(DEV DAT)
1897 002647 260140 003311'      PUSHJ PDP,SET176        ;IS THIS THE FIRST POINTER BLOCK?
1898 002650 200060 000002      MOVE TAC,@TAC1
1899 002651 606040 777777      TRNN TAC,-1
1900 002652 254000 002707'      JRST DFSETB          ;YES
1901 002653 550046 000016      HRRZ TAC,DEVBLK(DEV DAT) ;IN BLOCK WERE CURRENT ONE.
1902 002654 275040 000001      SUBI TAC,1
1903 002655 213000 000001      MOVNS TAC
1904 002656 272046 000022      ADDM TAC,SETCNT(DEV DAT)
1905 002657 254000 002711'      JRST DFSETC

```

```

1906
1907
1908
1909
1910
1911 002660
1912
1913 002660 550046 000015
1914 002661 271040 000177
1915 002662 242040 777771
1916 002663 301054 000000
1917 002664 254000 002670
1918 002665 205000 000040
1919 002666 437006 000002
1920 002667 354000 000001
1921 002670 550040 000014
1922 002671 550106 000022
1923 002672 276040 000002
1924 002673 272106 000014
1925 002674 542046 000022
1926 002675 254000 003654
1927
1928
1929
1930 002676 275106 000023
1931 002677 274040 000002
1932 002700 307054 000000
1933 002701 254000 002670
1934
1935
1936
1937 002702 200046 000010
1938 002703 603040 200000
1939 002704 260140 003131
1940 002705 550046 000012
1941 002706 260140 003316

;THE DESIRED POINTER MAY BE IN CORE. IF THERE ARE MORE POINTERS ON
;DISC, THEN CORE IS FULL, AND THE POINTER IS IN CORE, OTHERWISE
;A CHECK IS MADE TO INSURE THAT THE END FILE IS AFTER THE DESIRED
;POINTER.

DFSET3:
DFSETX: HRRZ TAC,DEVcnt(DEVDAT) ;C(UUO)RH > C(DEVcnt) RH?
        ADDI TAC,BLKSIz-1      ;CONVERT TO BLOCKS
        LSH TAC,-BLKP2
        CAIL TAC,(UUO)
        JRST DFSET5
        MOVSI IOS,IOEND        ;NO, ALL OK.
        IORB IOS,DEVIOS(DEVDAT) ;YES, END-FILE.
        AOSA TAC
DFSET5: HRRZ TAC,UUO            ;SET NEXT BLOCK AS ONE BEYOND LAST EXISTING ONE
        HRRZ TAC1,SETCNT(DEVDAT) ;SET DESIRED BLOCK NUMBER INTO TAC
        SUBM TAC,TAC1
        ADDM TAC1,DEVACC(DEVDAT)
        HRRM TAC,SETCNT(DEVDAT) ;RESET SETCNT
        JRST CLRBUF            ;CLEAR ANY DUMP BUFFER AND LEAVE

;THE DESIRED POINTER IS BEFORE THE CURRENT ONE. SEE IF IT IS IN CORE.
DFSET6: SUBI TAC1,PTR1(DEVDAT)
        SUB TAC,TAC1
        CAIG TAC,(UUO)
        JRST DFSET5            ;POINTER IS IN CORE

;THE DESIRED BLOCK IS BEFORE THE CURRENT ONE. SEARCH FROM THE BEGINNING.
        MOVE TAC,DEVOAD(DEVDAT) ;ARE POINTERS IN CORE SAME AS ON DISC?
        TLNE TAC,PNTDIF
        PUSHJ PDP,DF04A
        HRRZ TAC,DEVEXT(DEVDAT) ;GET BLOCK# OF FIRST RIB
        PUSHJ PDP,SETPTR        ;READ FIRST RIB
    
```

```

1942 002707 201040 000001 DFSETB: MOVEI TAC,1      IRESET SETCNT
1943 002710 542046 000022 HRRM TAC,SETCNT(DEVDAT) I"POSITIONED" AT RELATIVE BLOCK 1
1944
1945 002711 550046 000020 DFSETC: HRRZ TAC,DSKBUF(DEVDAT) ISET TAC TO FIRST POINTER
1946 002712 260140 003311' PUSHJ PDP,SET176 IIS THIS THE FIRST POINTER BLOCK?
1947 002713 200260 000002 MOVE DAT,@TAC1
1948 002714 606240 777777 TRNN DAT,-1
1949 002715 271040 000004 ADDI TAC,DIRSIZ IYES, SKIP OVER 4 WORDS
1950 002716 554260 000002 HLRZ DAT,@TAC1 IBLOCK FULL?
1951 002717 275100 000001 SUBI TAC1,1
1952 002720 326240 002723' JUMPN DAT,DFSETD IYES
1953 002721 336020 000002 SKIPN @TAC1 INO, MOVE TAC1 TO LAST POINTER
1954 002722 364100 002721' SOJA TAC1,,-1
1955
1956 002723 550646 000022 DFSETD: HRRZ AC1,SETCNT(DEVDAT)
1957 002724 275641 000000 SUBI AC1,(TAC) IIS DESIRED POINTER HERE?
1958 002725 550700 000002 HRRZ AC2,TAC1
1959 002726 270700 000015 ADD AC2,AC1
1960 002727 301714 000000 CAIL AC2,(UUO)

1961 002730 254000 002737' JRST DFSET7 IYES
1962 002731 271700 000001 ADDI AC2,1 INO, GET NEXT BLOCK
1963 002732 542706 000022 HRRM AC2,SETCNT(DEVDAT)
1964 002733 322240 002747' JUMPE DAT,DFSET8 IANY MORE POINTER BLOCKS?
1965 002734 206246 000016 MOVSM DAT,DEVBLK(DEVDAT) IYES, SET UP TO READ NEXT BLOCK.
1966 002735 260140 003262' PUSHJ PDP,RR1
1967 002736 254000 002711' JRST DFSETC
1968
1969 ITHE BLOCK CONTAINING THE POINTER IS FOUND.
1970 002737 550040 000014 DFSET7: HRRZ TAC,UUO IRESET RIGHT HALF OF DEVBLK
1971 002740 274040 000015 SUB TAC,AC1
1972 002741 550106 000020 HRRZ TAC1,DSKBUF(DEVDAT)
1973 002742 274040 000002 SUB TAC,TAC1
1974 002743 542046 000016 HRRM TAC,DEVBLK(DEVDAT)
1975
1976 002744 542606 000022 HRRM UUO,SETCNT(DEVDAT) IRESET SETCNT
1977 002745 260140 003016' PUSHJ PDP,DFIN4 IGET POINTERS
1978 002746 254000 003654' JRST CLRBUF ICLEAR ANY DUMP BUFFER AND LEAVE
1979
1980 IRAN OUT OF POINTERS
1981
1982 002747 271100 000001 DFSET8: ADDI TAC1,1
1983 002750 550046 000020 HRRZ TAC,DSKBUF(DEVDAT)
1984 002751 274100 000001 SUB TAC1,TAC
1985 002752 542106 000016 HRRM TAC1,DEVBLK(DEVDAT)
1986 002753 661000 000140 TLO IOS,NMPIIOEND ISET END
1987 002754 202006 000002 MOVEM IOS,DEVIOS(DEVDAT)
1988 002755 260140 003162' PUSHJ PDP,DF06A ICLEAR POINTERS
1989 002756 254000 003654' JRST CLRBUF ICLEAR ANY DUMP BUFFER AND LEAVE

```

```

1990          SUBTTL INPUT/OUTPUT UUD'S, RETRIEVAL POINTER PROCESSING
1991          ;INPUT UUD
1992          ;* INDICATES INTERRUPT LEVEL
1993
1994          EXTERNAL ADV8FF
1995
1996
1997 002757 607300 040000 DFIN:  TLNN DEVDAT,LOOKB      ;FILE OPEN?
1998 002760 254000 004355 JRST DFERR2          ;NO, UNDEFINED FILE
1999 002761 621000 000020 TLZ IOS,IO          ;SET INPUT INDICATION
2000
2001 002762 603000 000040 DFIN1: TLNE IOS,LIR          ;*ANY MORE INPUT?
2002 002763 263140 000000 POPJ PDP,              ;*NO, LEAVE
2003
2004 002764 550106 000014 HRRZ TAC1,DEVACC(DEVDAT) ;*
2005 002765 303106 000037 CAILE TAC1,PTRN(DEVDAT) ;*POINTER LIST EMPTY?
2006 002766 254000 003005 JRST DFIN2          ;*YES
2007 002767 336020 000002 SKIPN @TAC1          ;*NO--END OF FILE?
2008 002770 254000 002777 JRST DFIN1E         ;*YES
2009
2010 002771 306106 000037 CAIN TAC1,PTRN(DEVDAT) ;*NO--IS THIS THE LAST POINTER ?
2011 002772 254000 003002 JRST DFIN1B         ;*YES
2012
2013 002773 336002 000001 DFIN1D: SKIPN 1(TAC1)        ;*NO
2014 002774 661000 000040 TLO IOS,LIR          ;*SET LAST INPUT FLAG
2015 002775 202006 000002 DFIN1A: MOVEM IOS,DEVIOS(DEVDAT) ;*
2016 002776 254000 004066 JRST QIN           ;*
2017
2018 002777 661000 000040 DFIN1E: TLO IOS,LIR          ;*EMPTY FILE
2019 003000 202006 000002 MOVEM IOS,DEVIOS(DEVDAT) ;*
2020 003001 263140 000000 POPJ PDP,              ;*
2021
2022
2023 003002 603000 000100 DFIN1B: TLNE IOS,NMP          ;*ANY MORE POINTERS ON DISK?
2024 003003 661000 000040 TLO IOS,LIR          ;*NO, THIS IS THE LAST READ
2025 003004 254000 002775 JRST DFIN1A         ;*
2026
2027 003005 202006 000002 DFIN2: MOVEM IOS,DEVIOS(DEVDAT) ;
2028 003006 260140 003630 PUSHJ PDP,SETBFI      ;CHOOSE NEXT INPUT BUFFER TO HOLD RETRIEVAL POINTERS
2029 003007 260140 003012 PUSHJ PDP,GETPTR      ;READ RIB AND COPY POINTERS INTO DDB
2030 003010 550106 000014 HRRZ TAC1,DEVACC(DEVDAT) ;
2031 003011 254000 002773 JRST DFIN1D
    
```

```

2032                                ;NEED NEW POINTERS FROM DISK
2033
2034 003012 200706 000010 GETPTR: MOVE AC2,DEV0AD(DEV DAT) ;POINTERS DIFFER?
2035 003013 603700 200000          TLNE AC2,PNTDIF
2036 003014 254000 003131'          JRST DF04A                      ;YES, WRITE THEN READ AND RETURN
2037 003015 260140 003262'          PUSHJ PDP,RR1                      ;NO, READ BLOCK
2038
2039 003016 201706 000023 DFIN4:  MOVEI AC2,PTR1(DEV DAT) ;RESET DEVACC
2040 003017 542706 000014          HRRM AC2,DEVACC(DEV DAT)
2041 003020 200046 000016          MOVE TAC,DEVBLK(DEV DAT) ;RESET DEVBKO
2042 003021 202046 000017          MOVEM TAC,DEVBKO(DEV DAT) ;BLOCK # WORD INDEX OF A CURRENT RETRIEVAL PACKET
2043 003022 260140 003314' GTPTR1: PUSHJ PDP,SET000 ;SET TAC1 TO FIRST WORD OF RIB
2044 003023 200006 000002          MOVE IOS,DEVIOS(DEV DAT)
2045 003024 550746 000016          HRRZ AC3,DEVBLK(DEV DAT) ;WORD INDEX OF RETRIEVAL POINTER PACKET
2046 003025 270740 000002          ADD AC3,TAC1 ;AC3 POINTS TO NEXT POINTER WITHIN RIB
2047 003026 271100 000176          ADDI TAC1,BLKSIZE-2 ;TAC1 POINTS TO WORD 127
2048
2049 003027 201646 000037          MOVEI AC1,PTRN(DEV DAT) ;ODD END CHECK
2050 003030 311740 000002 DFIN5:  CAML AC3,TAC1 ;POINTER BLOCK EMPTY?

2051 003031 254000 003053'          JRST DFIN7                      ;YES, GET SOME MORE
2052
2053 003032 200060 000017          MOVE TAC,@AC3 ;GET RETRIEVAL POINTER
2054 003033 202060 000016          MOVEM TAC,@AC2 ;TASH IN ODB
2055 003034 271740 000001          ADDI AC3,1
2056 003035 322040 003063'          JUMPE TAC,DFIN8A ;0 MEANS NO MORE RETRIEVAL POINTERS
2057 003036 302656 000000          CAIE AC1,(AC2) ;ODB FULL?
2058 003037 344700 003030'          AOJA AC2,DFIN5 ;NO, GET ANOTHER
2059
2060 003040 311740 000002          CAML AC3,TAC1 ;YES, ANY MORE FOR NEXT TIME?
2061 003041 513020 000017          HLLZS @AC3 ;END OF RIB REACHED, ACE(LH) IS LINK TO NEXT RIB
2062 003042 336020 000017          SKIPN @AC3 ;NEXT POINTER=0 OR LINK=0.
2063 003043 661000 000100          TLO IOS,NMP ;SET "NO MORE POINTER"
2064
2065 003044 202006 000002 DFIN6A: MOVEM IOS,DEVIOS(DEV DAT)
2066 003045 200106 000020          MOVE TAC1,DSKBUF(DEV DAT);POINT TO FIRST WORD OF RIB
2067 003046 275742 000000          SUBI AC3,(TAC1) ;COMPUTE NEW INDEX
2068 003047 542746 000016          HRRM AC3,DEVBLK(DEV DAT) ;STORE INDEX TO NEXT RETRIEVAL POINTER PACKET
2069 003050 205640 200000          MOVSI AC1,PNTDIF ;CLEAR "POINTERS DIFFER"
2070 003051 412646 000010          ANDCAM AC1,DEV0AD(DEV DAT) ;SINCE ODB IS COPY OF RETRIEVAL DATA.
2071 003052 263140 000000          POPJ PDP,

```

```

2072 003053 554060 000002 DFIN7: HLRZ TAC,@TAC1          ;RIB LINK TO TAC RH
2073 003054 322040 003062' JUMPE TAC,DFIN8          ;DONE IF LINK=0
2074 003055 206046 000016 MOVSM TAC,DEVBLK(DEV DAT) ;SET BLOCK#, WORD INDEX FOR NEXT PACKET
2075 003056 261140 000016 PUSH PDP,AC2
2076 003057 260140 003262' PUSHJ PDP,RR1          ;READ THE RIR
2077 003060 262140 000016 POP PDP,AC2
2078 003061 254000 003022' JRST GTPTR1
2079 003062 202060 000016 DFIN8: MOVEM TAC,@AC2          ;0 WORD IN DOB MARKS END
2080 003063 661000 000100 DFIN8A: TLO IOS,NMP          ;NO MORE POINTERS ON DISK
2081 003064 254000 003044' JRST DFIN6A
2082
2083
2084          ;ENTER HERE AT INTERRUPT LEVEL TO READ ANOTHER BLOCK
2085 003065 260140 000000 DFINX: PUSHJ PDP,ADVBBF          ;*ANY MORE EMPTY BUFFERS?
2086 003066 263140 000000 POPJ PDP,          ;*
2087
2088 003067 550106 000014 HRRZ TAC1,DEVACC(DEV DAT) ;*POINTER LIST EMPTY?
2089 003070 303106 000037 CAILE TAC1,PTRN(DEV DAT) ;*
2090 003071 263140 000000 POPJ PDP,          ;*
2091
2092 003072 254000 002762' JRST DFIN1          ;*NO, FILL NEXT ONE

```

```

2093          ;OUTPUT UUD
2094
2095          EXTERNAL WSYNC,ADVBFE
2096
2097 003073 607300 020000 DFOUT:  TLNN DEVDAT,ENTRB      ;FILE OPEN?
2098 003074 254000 004355' JRST DFERR2          ;NO, UNDEFINED FILE
2099 003075 661000 000020      TLO IOS,IO          ;SET OUTPUT INDICATION
2100 003076 202006 000002      MOVEM IOS,DEVIOS(DEVDAT)
2101
2102 003077 550246 000014 DFOUT1: HRRZ DAT,DEVACC(DEVDAT) ;*GET POINTER LOC
2103
2104          IFG CHKCNT,<
2105 003100 550046 000010      HRRZ TAC,DEVOAD(DEVDAT) ;*
2106 003101 270040 004613'    ADD TAC,[XWD NBLKSZ,2] ;*
2107 003102 260140 004473'    PUSHJ PDP,CHKSUM      ;*
2108 003103 506120 000005      WRLM TAC1,@DAT      ;*
2109 003104 205100 200000      MOVSI TAC1,PNTDIF     ;*
2110 003105 436106 000010      ORM TAC1,DEVOAD(DEVDAT) ;*
2111          >

2112 003106 550060 000005      HRRZ TAC,@DAT          ;*ALREADY HAVE A POINTER?
2113 003107 326040 003124'    JUMPN TAC,DFOT3A        ;*YES IF JUMP
2114 003110 302246 000037      CAIE DAT,PTRN(DEVDAT)   ;*NO, CLEAR NEXT ONE?
2115 003111 402005 000001      SETZM 1(DAT)           ;*YES
2116 003112 661000 000100      TLO IOS,NMP           ;*SET "NO MORE POINTERS"
2117 003113 202006 000002      MOVEM IOS,DEVIOS(DEVDAT) ;*
2118 003114 200106 000010      MOVE TAC1,DEVOAD(DEVDAT) ;*POINT TO OUTPUT BUFFER
2119 003115 271100 000001      ADDI TAC1,1            ;*NOW TO WORD COUNT
2120 003116 550120 000002      HRRZ TAC1,@TAC1        ;*RETRIEVE WORD COUNT
2121 003117 260140 004534'    PUSHJ PDP,UPDEV        ;*UPDATE DEVCNT
2122 003120 261140 000005      PUSH PDP,DAT           ;*
2123 003121 260140 003371'    PUSHJ PDP,DFGETF        ;*GET A FREE BLOCK
2124 003122 262140 000005      POP PDP,DAT           ;*
2125
2126          IFG CHKCNT,<
2127 003123 542060 000005      HRRM TAC,@DAT>
2128          IFLE CHKCNT,<
2129          MOVSI TAC1,PNTDIF
2130          ORM TAC1,DEVOAD(DEVDAT)
2131          MOVEM TAC,@DAT>
2132
2133 003124 302246 000037 DFOT3A: CAIE DAT,PTRN(DEVDAT)   ;*LIST FULL?
2134 003125 254000 004067'    JRST QOUT              ;*NO, WRITE AND LEAVE

```

```

2135          ;OUTPUT UO CONTINUED.
2136          ;WRITE OUT POINTER LIST.
2137
2138 003126 260140 003634'      PUSHJ PDP,SETBFO      ;CHOOSE AN OUTPUT BUFFER TO READ RIB INTO
2139 003127 260140 004067'      PUSHJ PDP,ROUT      ;PUT REQUEST IN QUEUE
2140 003130 260140 000616'      PUSHJ PDP,WSYNC      ;WAIT TIL DATA WRITTEN
2141
2142 003131 200046 000017 DF04A: MOVE TAC,DEVBLK(DEV DAT) ;RESET DEVBLK TO REFER TO RETRIEVAL POINTERS
2143 003132 202046 000016      MOVEM TAC,DEVBLK(DEV DAT) ;NOW IN DDB
2144
2145 003133 200046 000010      MOVE TAC,DEVOD(DEV DAT) ;ANY POINTERS BEEN WRITTEN?
2146 003134 623040 100000      TLZE TAC,VRGPTR
2147 003135 260140 003172'      PUSHJ PDP,DF07A      ;NO, CREATE NEW BLOCK, SKIP NEXT INSTRUCTION.
2148 003136 260140 003262'      PUSHJ PDP,RR1        ;READ RETRIEVAL BLOCK
2149 003137 201746 000023      MOVEI AC3,PTR1(DEV DAT)
2150
2151 003140 260140 003314' DF04B: PUSHJ PDP,SET000      ;TAC1 POINTS TO RIB
2152 003141 550706 000016      HRRZ AC2,DEVBLK(DEV DAT) ;INDEX INTO RIB
2153 003142 270700 000002      ADD AC2,TAC1          ;ABSOLUTE POINTER INTO RIB
2154 003143 271100 000176      ADDI TAC1,BLKSIZ-2      ;END OF RIB DATA
2155 003144 200660 000017 DFOUT5: MOVE AC1,@AC3        ;GET RETRIEVAL POINTER FROM DDB
2156 003145 322640 003154'      JUMPE AC1,DFOUT6      ;0 SIGNALS END
2157 003146 311700 000002      CAML AC2,TAC1          ;BLOCK FULL?
2158 003147 254000 003200'      JRST DFOUT7          ;YES, WRITE IT OUT
2159 003150 202660 000016      MOVEM AC1,@AC2        ;STASH POINTER IN RIB
2160 003151 271700 000001      ADDI AC2,1
2161 003152 302746 000037      CAIE AC3,PTRN(DEV DAT) ;DONE?
2162 003153 344740 003144'      AOJA AC3,DFOUT5
2163
2164 003154 200106 000020 DFOUT6: MOVE TAC1,DSKBUF(DEV DAT) ;POINT TO RIB
2165 003155 275702 000000      SUBI AC2,(TAC1)        ;FORM INDEX
2166 003156 542706 000016      HRRM AC2,DEVBLK(DEV DAT) ;STORE INDEX TO RETRIEVAL POINTERS
2167 003157 205700 200000      MOVS1 AC2,PNTDIF      ;CLEAR "POINTERS DIFFER"
2168 003160 412706 000010      ANDCAM AC2,DEVOD(DEV DAT)
2169 003161 260140 003274'      PUSHJ PDP,WRI        ;WRITE THE BLOCK
2170 003162 201046 000023 DF06A: MOVEI TAC,PTR1(DEV DAT)
2171 003163 542046 000014      HRRM TAC,DEVACC(DEV DAT) ;DEVACC NOW POINTS TO FIRST POINTER IN DDB
2172 003164 402020 000001      SETZM @TAC          ;CLEAR FIRST POINTER
2173 003165 607000 000100      TLNN IOS,NMP        ;MORE POINTERS?
2174 003166 254000 003016'      JRST DFIN4          ;YES, BRING THEM IN AND LEAVE
2175 003167 200046 000016      MOVE TAC,DEVBLK(DEV DAT)
2176 003170 202046 000017      MOVEM TAC,DEVBLK(DEV DAT) ;SAVE BLOCK#,WORD INDEX OF CURRENT POINTER PACKET
2177 003171 263140 000000      POPJ PDP,

```

2178	003172	202046	000010	DF07A:	MOVEM TAC,DEV0AD(DEV0AT)	;SAVE VRGPTR BIT
2179	003173	260140	003304		PUSHJ PDP,SET177	;SET TAC1 TO WORD 128
2180	003174	402020	000002		SETZM @TAC1	;CLEAR BLOCK# WORD
2181	003175	350003	000000		AOS (PDP)	;ALWAYS SKIP RETURN
2182	003176	261140	000017		PUSH PDP,AC3	;BECAUSE ENTRY FROM NEWRIB PUSHES AC3
2183	003177	254000	003215		JRST DFOUT8	
2184						
2185	003200	260140	003202	DFOUT7:	PUSHJ PDP,NEWRIB	
2186	003201	254000	003140		JRST DF04B	
2187						
2188	003202	261140	000017	NEWRIB:	PUSH PDP,AC3	;CREATE A NEW RETRIEVAL BLOCK
2189	003203	260140	003311		PUSHJ PDP,SET176	;SET TAC1 TO LINK WORD
2190	003204	510060	000002		HLLZ TAC,@TAC1	;LINK TO NEXT POINTER BLOCK ALREADY PRESENT?
2191	003205	326040	003233		JUMPN TAC,DFOUT9	;YES, WRITE CURRENT ONE AND READ NEXT ONE
2192	003206	260140	003371		PUSHJ PDP,DFGETF	;NO, GET A BLOCK FOR POINTERS IN TAC
2193	003207	554746	000016		HLRZ AC3,DEVBLK(DEV0AT)	;BLOCK# OF CURRENT RIB
2194	003210	206046	000016		MOVSM TAC,DEVBLK(DEV0AT)	;STORE BLOCK#, INDEX FOR NEW BLOCK
2195	003211	260140	003311		PUSHJ PDP,SET176	;SET TAC1 TO WORD 127 AGAIN
2196	003212	506060	000002		HRLM TAC,@TAC1	;LH LINKS TO NEW BLOCK
2197	003213	200040	000017		MOVE TAC,AC3	;BLOCK# OF CURRENT BLOCK
2198	003214	260140	003275		PUSHJ PDP,WRIA	;WRITE THE BLOCK OUT
2199						
2200	003215	200106	000020	DFOUT8:	MOVE TAC1,DSKBUF(DEV0AT)	;POINT TO BUFFER CONTAINING RIB
2201					NOSCHEDULE	
2202	003216	700600	002024		CONO PI,DSKOFF	
2203	003217	607000	000200		TLNN IOS,UBFS	;DUMP MODE?
2204	003220	270100	000007		ADD TAC1,PROG	;NO, RELOCATE
2205	003221	201042	000001		MOVEI TAC,1(TAC1)	;DESTINATION
2206	003222	504040	000002		HRL TAC,TAC1	;SOURCE
2207	003223	402002	000000		SETZM (TAC1)	
2208	003224	251042	000175		BLT TAC,BLKSI2-3(TAC1)	;ZERO OUT FIRST 126 WORDS
2209	003225	200042	000177		MOVE TAC,BLKSI2-1(TAC1)	;BLOCK# OF CURRENT RIB
2210	003226	552042	000176		HRRZM TAC,BLKSI2-2(TAC1)	;SET "PREVIOUS" POINTER,,CLEAR LINK TO NEXT
2211	003227	402002	000177		SETZM BLKSI2-1(TAC1)	;CLEAR WORD 128,
2212					SCHEDULE	
2213	003230	700600	002250		CONO PI,DSKON	
2214	003231	262140	000017	DF08A:	POP PDP,AC3	
2215	003232	263140	000000		POPJ PDP,	
2216						
2217	003233	250046	000016	DFOUT9:	EXCH TAC,DEVBLK(DEV0AT)	;SAVE NEW BLOCK# INDEX. GET OLD BLOCK# INDEX
2218	003234	557000	000001		HLRZS TAC	;BLOCK# IN RH
2219	003235	260140	003275		PUSHJ PDP,WRIA	;WRITE THIS BLOCK OF POINTERS OUT
2220	003236	260140	003262		PUSHJ PDP,RI	;READ NEXT BLOCK OF POINTERS IN
2221	003237	254000	003231		JRST DF08A	

```

2222
2223          ENTER HERE AT INTERRUPT LEVEL TO WRITE ANOTHER
2224
2225          EXTERN MJOBN
2226 003240 260140 000000 DFOUTX: PUSHJ PDP,ADVBFE      ;*ANY MORE TO DO?
2227 003241 263140 000000      POPJ PDP,              ;*
2228 003242 331000 000000      SKIPL DAREQ            ;*DO NOT CONTINUE THIS OUTPUT AT INTERRUPT
2229 003243 263140 000000      POPJ PDP,              ;* LEVEL IF THE DAWAIT RTN WOULD BE CALLED.
2230 003244 550106 000014      HRRZ TAC1,DEVACC(DEVDAT) ;*NEED TO WRITE POINTERS?
2231 003245 301106 000037      CAIL TAC1,PTRN(DEVDAT)  ;*
2232 003246 263140 000000      POPJ PDP,              ;*
2233 003247 550060 000015      HRRZ TAC,@SATPTR        ;*CURRENT SAT BLOCK FULL?
2234 003250 606040 400000      TRNN TAC,WLBIT         ;*OR WRITE-LOCKED?
2235 003251 301040 011000      CAIL TAC,NUMBIT+MJOBN   ;*BE SURE TO LEAVE ATLEAST JOBN FREE BLOCKS
2236                                     ; SO THAT EACH JOB CAN HAVE ONE MORE
2237 003252 263140 000000      POPJ PDP,              ;*
2238 003253 261140 000015      PUSH PDP,AC1            ;*
2239 003254 261140 000016      PUSH PDP,AC2            ;*
2240 003255 260140 003077      PUSHJ PDP,DFOUT1        ;*
2241 003256 262140 000016      POP PDP,AC2            ;*
2242 003257 262140 000015      POP PDP,AC1            ;*
2243 003260 263140 000000      POPJ PDP,              ;*

```

```

2244          SUBTTL READ/WRITE RETRIEVAL POINTERS
2245          ;INPUT A RETRIEVAL BLOCK.
2246
2247 003261 334046 000012 RRIB:  SKIP A TAC,DEVEXT(DEV DAT) ;READ FIRST RETRIEVAL BLOCK
2248 003262 554046 000016 RRI:  HLRZ TAC,DEVBLK(DEV DAT) ;READ NEXT RETRIEVAL BLOCK
2249 003263 260140 004110' RRIA:  PUSHJ PDP,MQIN ;READ RETRIEVAL BLOCK SPECIFIED BY TAC ON ENTRY
2250 003264 254000 004457' JRST RERA ;ERRORS, YOU LOSE
2251 003265 260140 003304' PUSHJ PDP,SET177 ;CHECK RH OF WORD 128
2252 003266 554046 000021 HLRZ TAC,DSKCNT(DEV DAT) ;BLOCK# EXPECTED
2253 003267 200660 000002 MOVE AC1,@TAC1 ;GET BLOCK# STORED IN RIB
2254 003270 302055 000000 CAIE TAC,(AC1) ;SAME?
2255 003271 254000 004457' JRST RERA ;IT WAS WRONG, YOU LOSE,
2256 003272 263140 000000 POPJ PDP, ;EVERYTHING OK.
2257
2258          ;OUTPUT A RETRIEVAL BLOCK.
2259
2260 003273 334046 000012 WRIB:  SKIP A TAC,DEVEXT(DEV DAT) ;WRITE FIRST RETRIEVAL BLOCK
2261 003274 554046 000016 WRI:  HLRZ TAC,DEVBLK(DEV DAT) ;WRITE NEXT RETRIEVAL BLOCK
2262 003275 260140 003304' WRIA:  PUSHJ PDP,SET177 ;WRITE RETRIEVAL BLOCK SPECIFIED BY TAC ON ENTRY
2263 003276 542060 000002 HRRM TAC,@TAC1 ;STORE THIS BLOCK# IN WORD 128
2264 IFN FIRCHK,<
2265 003277 336000 000001 SKIPN TAC ;ATTEMPTING TO WRITE RETRIEVAL INFORMATION
2266 ; IN BLOCK NR 0 IS AN ERROR
2267 003300 254200 004443' HALT WERA ;CONTINUE WILL POINT ERROR MESSAGE
2268 >
2269 003301 260140 004112' PUSHJ PDP,MQOUT ;WRITE IT.
2270 003302 254000 004443' JRST WERA ;ERRORS
2271 003303 263140 000000 POPJ PDP, ;NO, RETURN.
2272
2273 003304 550106 000020 SET177: HRRZ TAC1,DSKBUF(DEV DAT)
2274 003305 271100 000177 ADDI TAC1,BLKSI2-1
2275 003306 607000 000200 S177A: TLNN IOS,UBFS
2276 003307 661100 000007 TLO TAC1,PROG
2277 003310 263140 000000 POPJ PDP,
2278
2279 003311 550106 000020 SET176: HRRZ TAC1,DSKBUF(DEV DAT)
2280 003312 271100 000176 ADDI TAC1,BLKSI2-2
2281 003313 254000 003306' JRST S177A
2282
2283 003314 550106 000020 SET000: HRRZ TAC1,DSKBUF(DEV DAT)
2284 003315 254000 003306' JRST S177A

```

```

2285
2286          ;PUT POINTER INFO IN DOB
2287          ;ENTER WITH TAC CONTAINING THE BLOCK NUMBER OF THE POINTERS.
2288
2289 003316 516046 000016 SETPTR: HRLZM TAC,DEVBLK(DEV DAT) ;XWD RIB#, 0...0 INDEX
2290 003317 420000 004614' ANDCM IOS,[XWD IOEND+NMP,IOEND]
2291 003320 202006 000002 MOVEM IOS,DEVIOS(DEV DAT)
2292 003321 260140 003620' PUSHJ PDP,SETBUF          ;ACQUIRE BUFFER IN MONITOR STORAGE
2293 003322 260140 003262' PUSHJ PDP,RR1          ;GET RETRIEVAL BLOCK
2294
2295 003323 260140 003016' PUSHJ PDP,DFIN4          ;COPY POINTERS TO DOB
2296 003324 201746 000027 MOVEI AC3,FPNTR(DEV DAT) ;SET DEVACC TO SKIP FIRST 4 WORDS
2297 003325 542746 000014 HRRM AC3,DEVACC(DEV DAT)
2298 003326 263140 000000 POPJ PDP,
2299
2300          ;RECLAIM STORAGE ON DISK.
2301          ;ENTER WITH TAC POINTING TO FIRST RETRIEVAL BLOCK OF THE FILE
2302          ; TO BE DELETED.
2303
2304
2305 003327 260140 003620' RECLAM: PUSHJ PDP,SETBUF          ;FIND SPACE IN USER AREA
2306
2307 003330 260140 003344' PUSHJ PDP,RECLM5          ;GET FIRST POINTER BLOCK
2308 003331 271240 000004 ADDI DAT,DIRSI2          ;SKIP OVER FIRST WORDS.
2309 003332 311240 000002 RECLM2: CAML DAT,TAC1          ;MORE POINTERS IN BLOCK?
2310 003333 260140 003342' PUSHJ PDP,RECLM4          ;NO, GET MORE
2311 003334 550060 000005 HRRZ TAC,@DAT          ;FREE A DATA BLOCK
2312 003335 322040 003654' JUMPE TAC,CLRBUF          ;EXIT WHEN A ZERO POINTER IS ENCOUNTERED
2313 003336 261140 000002 PUSH PDP,TAC1
2314 003337 260140 003464' PUSHJ PDP,SETFRE          ;FREE UP SPECIFIED BLOCK
2315 003340 262140 000002 POP PDP,TAC1
2316 003341 344240 003332' AOJA DAT,RECLM2          ;GO BACK FOR MORE
2317
2318
2319 003342 554060 000002 RECLM4: HLRZ TAC,@TAC1          ;PICK UP POINTER TO NEXT BLOCK
2320 003343 322040 003353' JUMPE TAC,RECLM6          ;NO MORE, GO HOME.
2321 003344 260140 003263' RECLM5: PUSHJ PDP,RR1A          ;READ IN A R.I. BLOCK.
2322
2323 003345 554046 000021 HLRZ TAC,DSKCNT(DEV DAT) ;FREE THAT BLOCK
2324 003346 260140 003464' PUSHJ PDP,SETFRE
2325
2326 003347 260140 003314' PUSHJ PDP,SET000          ;SET TAC1 TO FIRST WORD
2327 003350 200240 000002 MOVE DAT,TAC1
2328 003351 271100 000176 ADDI TAC1,BLKSIZ-2          ;SET TAC1 TO WORD 127
2329 003352 263140 000000 POPJ PDP,
2330
2331 003353 262140 000001 RECLM6: POP PDP,TAC
2332 003354 254000 003654' JRST CLRBUF          ;CLEAR ANY DUMP BUFFER AND LEAVE

```

```

2333      ;RELEASE UUO.
2334
2335      EXTERNAL DAREQ,DAVAL
2336
2337 003355 661300 000020 DFREL: TLO      DEVDAT,DSKRLB ;MARK RELEASE (VIA RESET UUO) IN PROGRESS,
2338 003356 260140 001601' PUSHJ PDP,DFCLSI ;CLOSE INPUT FIRST
2339 003357 260140 001453' PUSHJ PDP,DFCLSO ;THEN CLOSE OUTPUT
2340
2341 003360 336000 000017'      SKIPN SATCHG ; SAME ORDER AS CLOSE AND RELEAS IN UUOCON
2342 003361 254000 003367'      JRST DFREL1 ;HAS SAT BLOCK BEEN MODIFIED?
2343 003362 260140 003414'      PUSHJ PDP,UPDA ;NO. SUPPRESS WRITING
2344 003363 261146 000020      PUSH PDP,DSKBUF(DEVDAT) ;SET DAREQ INTERLOCK WHILE WRITING SAT BLOCK
2345 003364 260140 003536'      PUSHJ PDP,SATWRT ;WRITE CURRENT SAT BLOCK
2346 003365 262146 000020      POP PDP,DSKBUF(DEVDAT)
2347 003366 260140 003404'      PUSHJ PDP,DOWNDA ;RESET DAREQ INTERLOCK
2348 003367 621300 000020 DFREL1: TLZ      DEVDAT,DSKRLB ;CLEAR RELEASE MARKER,
2349 003370 254000 002614'      JRST WAIT1
  
```

```

2350          SURTTL DISK SPACE ALLOCATION
2351          ;DFGETF: GET A FREE DISK BLOCK
2352          ;CALL:  PUSHJ PDP,DFGETF
2353          ;      ... RETURNS WITH TAC:=LOGICAL BLOCK#
2354          ;      MAY NOT RETURN IF DISK IS FULL OR NEARLY FULL
2355
2356          EXTERNAL DAWAIT,DAVAL,DAREQ,JOBN
2357          INTERNAL DFGETF
2358
2359 003371 260140 003414' DFGETF: PUSHJ PDP,UPDA          ;SET DISK ALLOCATION INTERLOCK
2360 003372 260140 003421' DFGETF1: PUSHJ PDP,SATGET      ;FOCUS ON A NON-FULL SAT BLOCK
2361 003373 550640 000015'          HRRZ AC1,SATPTR      ;POINT TO 3 WORD SAT ENTRY
2362 003374 200700 000236'          MOVE AC2,SATBK2      ;XWD -L, POINTER TO SAT BLOCK
2363 003375 201200 000001'          MOVEI ITEM,1         ;REQUEST 1 BLOCK
2364 003376 260140 003744'          PUSHJ PDP,GETBIT     ;RETURNS BIT# IN TAC1 (IN RANGE 1 TO NUMBLK)
2365 003377 254000 003411'          JRST DFGETF2         ;FULL AFTER ALL
2366 003400 476000 000017'          SETOM SATCHG         ;SHOW SAT BLOCK IN CORE NOT = DISK COPY
2367 003401 350060 000015'          AOS TAC,@SATPTR      ;COUNT ANOTHER BLOCK USED
2368 003402 557000 000001'          HLRZS TAC           ;FIRST LOGICAL BLOCK# REPRESENTED BY SAT BLOCK
2369 003403 271042 777777'          ADDI TAC, -1(TAC1)   ;FORM LOGICAL BLOCK#
2370 003404 371000 003242' DOWNDA: SOSL DAREQ           ;THIS EXITING ROUTINE TURNS OFF DAREQ INTERLOCK
2371                                     ; THUS MAKING NON-SHARABLE SECTIONS OF CODE
2372                                     ; AVAILABLE TOTHE NEXT USER IN THE QUEUE
2373 003405 476000 000000          SETOM DAAVAL
2374 003406 621000 004000          TLZ IOS,DAFLG
2375 003407 202006 000002          MOVEM IOS,DEVIOS(DEVNAT)
2376 003410 263140 000000          POPJ PDP,
2377
2378 003411 201040 011000  DFGETF2: MOVEI TAC,NUMBIT      ;BITS/SAT BLOCK
2379 003412 542060 000015'          HRRM TAC,@SATPTR      ;MARK SAT ENTRY FULL
2380 003413 254000 003372'          JRST DFGETF1
2381
2382
2383          ;UPDA - TURN ON THE DISK ALLOCATION INTERLOCK (DAREQ)
2384          ;      - A CALL TO UPDA MUST PRECEDE SAT BLOCK MANIPULATIONS
2385
2386 003414 352000 003404' UPDA:  AOSE DAREQ              ;INCREMENT COUNT OF USERS REQUESTING DISK ALLOCATION
2387 003415 260140 000000          PUSHJ PDP,DAWAIT      ;WAIT UNTIL JOB REACHES TOP OF QUEUE
2388 003416 661000 004000          TLO IOS,DAFLG
2389 003417 202006 000002          MOVEM IOS,DEVIOS(DEVNAT) ;MARK THIS JOB AS USING SAT BLOCK
2390 003420 263140 000000          POPJ PDP,

```

```

2391          ;SATGET - FIND A NON-FULL SAT BLOCK FOR DFGFTF OR DUMP MODE OUTPUT.
2392          ;      CONSIDER DISK FULL WHEN FEW BLOCKS REMAIN FREE UNLESS THE USER
2393          ;      IS LOGGING IN OR USING A NON-SHARABLE RESOURCE.
2394
2395          EXTERNAL JBTSTS,HNGSTP
2396
2397 003421 400640 000000 SATGET: SETZ AC1,          ;CLEAR COUNT
2398 003422 550060 000015' HRRZ TAC,@SATPTR      ;BLOCKS USED IN CURRENT SAT BLOCK
2399 003423 606040 400000 TRNN TAC,WLBIT        ;WRITE LOCKED?
2400 003424 260140 003457' PUSHJ PDP,SATCNT     ;NO. COUNT # FREE
2401 003425 301640 000000 CAIL AC1,JOBN        ;IS THERE ENOUGH SPACE?
2402 003426 263140 000000 POPJ PDP,          ;YES - USE IT,
2403 003427 400640 000000 SETZ AC1,          ;CLEAR COUNT
2404 003430 201100 000022' MOVEI TAC1,SATENT    ;START WITH FIRST SAT ENTRY
2405 003431 550042 000000 SATGT1: HRRZ TAC, 0(TAC1) ;GET WLBIT + BLOCKS USED
2406 003432 606040 400000 TRNN TAC,WLBIT        ;WRITE LOCKED?
2407 003433 260140 003457' PUSHJ PDP,SATCNT     ;NO. COUNT # FREE
2408 003434 271100 000003 ADDI TAC1,SENTSZ    ;NEXT SAT ENTRY
2409 003435 301640 003425' CAIL AC1,JOBN        ;ENOUGH SPACE ACCUMULATED?

2410 003436 254000 003455' JRST SATGT2          ;YES - READ A SAT BLOCK
2411 003437 305100 000033' CAIGE TAC1,SATTOP    ;END OF ENTRIES?
2412 003440 254000 003431' JRST SATGT1          ;NO. CONTINUE
2413 003441 322640 003447' JUMPE AC1,DSKFUL     ;IF NONE FOUND DISK IS REALLY FULL
2414
2415
2416          ;THERE ARE JOBN OR FEWER DISK BLOCKS AVAILABLE.
2417          ;GIVE THEM OUT ONLY IN HARSHIP CASES
2418 003442 135040 000000 LDB TAC,PJOBN
2419 003443 200041 000000 MOVE TAC,JBTSTS(TAC)
2420 003444 607040 100000 TLNN TAC,JACCT
2421 003445 603000 010200 TLNE IOS,UBFS+AUFLG    ;IS THIS USER LOGGING IN?
2422 003446 254000 003455' JRST SATGT2          ;DOES USER HAVE MONITOR BUFFER OR IS HE ALTERING UFD?
2423 003447 603000 010200 DSKFUL: TLNE IOS,UBFS+AUFLG ;YES TO EITHER, GIVE HIM A BLOCK ANYWAY.
2424          JRST DFERR1                          ;NON-RECOVERABLE ERROR IF USING BUFFER OR ALTERING UFD
2425 003450 254000 004342'
2426 003451 260140 003404' PUSHJ PDP,DOWNDA    ;RESET ALLOCATION INTERLOCK
2427 003452 260140 000000 PUSHJ PDP,HNGSTP    ;PRINT "DSK OK?"
2428 003453 260140 003414' PUSHJ PDP,UPDA      ;IF USER TYPES "CONT" WE TRY AGAIN
2429 003454 254000 003421' JRST SATGET
2430
2431 003455 200100 000016' SATGT2: MOVE TAC1,SATPIK ;POINT TO ENTRY OF NON-FULL SAT BLOCK
2432 003456 254000 003520' JRST SATRD          ;READ IN SAT BLOCK
2433
2434 003457 302040 011000 SATCNT: CAIE TAC,NUMBIT ;IS ENTRY FULL?
2435 003460 202100 000016' MOVEM TAC1,SATPIK    ;NO. SAVE POINTER TO A NON-FULL ENTRY
2436 003461 271640 011000 ADDI AC1,NUMBIT      ;MAXIMUM NUMBER
2437 003462 274640 000001 SUB AC1,TAC        ;LESS NUMBER USED
2438 003463 263140 000000 POPJ PDP,

```

```

2439          ;SETFRE - RETURN A DISK BLOCK TO AVAILABLE STORAGE
2440          ;CALL      TAC:=LOGICAL BLOCK L
2441          ;          PUSHJ PDP,SETFRE
2442          ;          ... RETURN
2443
2444          INTERNAL SETFRE,FTRCHK
2445
2446 003464 260140 003414' SETFRE: PUSHJ PDP,UPDA          ;SET DISK ALLOCATION INTERLOCK
2447          IFN FTRCHK,<
2448          EXTERNAL LBHIGH
2449
2450 003465 332000 000001      SKIPE TAC          ;BLOCK 0 SHOULD NEVER BE CLEARED
2451 003466 313040 000000      CAMLE TAC,LBHIGH      ;BLOCK# TOO LARGE
2452 003467 254200 003404'      HALT DOWNDA          ;YES, HALT, CONTINUE WILL COUNT DOWN DAREQ,
2453
2454 003470 554660 000015'      >
2455 003471 311040 000015'      HLRZ AC1,@SATPTR      ;GET LOGICAL BLOCK# FOR B0 OF CURRENT SAT BLOCK
2456 003472 301055 007640      CAML TAC,AC1          ;IS L LESS THAN FIRST BLOCK REPRESENTED?
2457 003473 334000 000000      CAIL TAC,NUMBLK(AC1)   ;IS L LESS THAN LAST BLOCK REPRESENTED
2458 003474 254000 003510'      SKIPA              ;OUTSIDE RANGE
2459 003475 201100 000022'      JRST SETFR2          ;IN RANGE
2460 003476 334000 000000      MOVEI TAC1,SATENT      ;INITIALIZE POINTER
2461 003477 271100 000003      SETFR1: ADDI TAC1,SENTSZ ;INCREMENT POINTER
2462 003500 554660 000002      HLRZ AC1,@TAC1        ;GET LOGICAL BLOCK FOR B0 OF SAT BLOCK
2463 003501 311040 000015'      CAML TAC,AC1          ;IS L LESS THAN FIRST BLOCK?
2464 003502 301055 007640      CAIL TAC,NUMBLK(AC1)   ;IS L LESS THAN LAST BLOCK?
2465 003503 254000 003477'      JRST SETFR1          ;OUT-OF-RANGE, TRY NEXT
2466 003504 261140 000001      PUSH PDP,TAC          ;SAVE BLOCK#
2467 003505 260140 003520'      PUSHJ PDP,SATRD      ;READ SAT BLOCK FOR ENTRY POINTED TO BY TAC1
2468 003506 262140 000001      POP PDP,TAC           ;RESTORE BLOCK#
2469 003507 554660 000015'      HLRZ AC1,@SATPTR
2470 003510 274040 000015'      SETFR2: SUB TAC,AC1   ;FORM BIT# (IN RANGE 0 TO NUMBLK-1)
2471 003511 550640 000015'      HRRZ AC1,SATPTR      ;POINT TO 3 WORD ENTRY
2472 003512 200700 000236'      MOVE AC2,SATBK2      ;XWD -L, POINTER TO SAT BLOCK
2473 003513 201200 000001      MOVEI ITEM,1          ;RETURN 1 BIT
2474 003514 260140 004032'      PUSHJ PDP,CLRBIT
2475 003515 370020 000015'      SOS @SATPTR
2476 003516 476000 000017'      SETOM SATCHG
2477 003517 254000 003404'      JRST DOWNDA          ;DECREMENT BLOCKS USED
                                                    ;SAT BLOCK IN CORE NOT = DISK COPY
                                                    ;REMOVE ALLOCATION INTERLOCK AND EXIT.

```

```

2478                                     ;NOTE: SATRD USES TEMPORARY STORAGE LOCATIONS SATTMP AND SATTEM TO HELP AVOID
2479                                     ; A PDL OVFLW PROBLEM. IT CAN DO SO BECAUSE IT IS NOT RE-ENTRANT, I.E., ONLY
2480                                     ; ONE USER AT A TIME MAY PASS THROUGH HERE BECAUSE OF THE DAREQ INTERLOCK,
2481
2482 003520 202100 000021' SATRD: MOVEM TAC1,SATTMP      ;SAVE POINTER TO DESIRED SAT ENTRY.
2483 003521 200106 000020'      MOVE TAC1,DSKBUF(DEV DAT) ;ALSO SAVE IOWD FROM THE DDB
2484 003522 202100 000020'      MOVEM TAC1,SATTEM
2485 003523 332000 000017'      SKIPE SATCHG          ;HAS COPY IN CORE BEEN MODIFIED?
2486 003524 260140 003536'      PUSHJ PDP,SATWRT      ;YES, WRITE IT.
2487 003525 200040 000021'      MOVE TAC,SATTMP       ;RETRIEVE SAT POINTER FOR SAT BLOCK TO BE READ IN.
2488 003526 202040 000015'      MOVEM TAC,SATPTR
2489 003527 260140 003546'      PUSHJ PDP,SATBLK      ;COMPUTE BLOCK#, IOWD
2490 003530 661000 000400'      TLO IOS,NORELB        ;NO RELOCATION
2491 003531 260140 004110'      PUSHJ PDP,MQIN        ;READ THE BLOCK
2492 003532 254000 004457'      JRST RERA            ;ERROR
2493 003533 200100 000020'      MOVE TAC1,SATTEM      ;RESTORE SAVED IOWD TO DDB
2494 003534 202106 000020'      MOVEM TAC1,DSKBUF(DEV DAT)
2495 003535 254000 003543'      JRST SATW1
2496
2497 003536 661000 000400' SATWRT: TLO IOS,NORELB      ;NO RELOCATION
2498 003537 260140 003546'      PUSHJ PDP,SATBLK      ;COMPUTE BLOCK" IOWD
2499 003540 260140 004112'      PUSHJ PDP,MQOUT      ;WRITE BLOCK
2500 003541 254000 004443'      JRST WERA            ;ERROR
2501 003542 402000 000017'      SETZM SATCHG         ;SHOW COPY IN CORE = BLOCK ON DISK
2502 003543 205000 000400' SATW1: MOVSI IOS,NORELB   ;RESET BIT
2503 003544 413006 000002'      ANDCAB IOS,DEVIOS(DEV DAT)
2504 003545 263140 000000'      POPJ PDP,
2505
2506                                     ;SET UP TAC AND TAC1 FOR READ OR WRITE OF SAT BLOCK
2507
2508
2509 003546 200040 000015' SATBLK: MOVE TAC,SATPTR
2510 003547 275040 000022'      SUBI TAC,SATENT
2511 003550 231040 000003'      IDIVI TAC,SENTSZ
2512 003551 270040 000001'      ADD TAC,SATXWD
2513 003552 200100 000236'      MOVE TAC1,SATBK2
2514 003553 202106 000020'      MOVEM TAC1,DSKBUF(DEV DAT)
2515 003554 263140 000000'      POPJ PDP,

```

```

2516          SUBTTL "FREE" CORE ALLOCATION
2517          ;CLEAR OUT DDB AT RELEASE TIME
2518
2519          INTERNAL CLRDD8
2520
2521 003555 201040 000237' CLRDD8: MOVEI TAC,DSKDD8      ;POINT TO PROTOTYPE DDB
2522 003556 200100 000001' CLDDR1: MOVE TAC1,TAC        ;COPY LINK TO TAC1
2523 003557 554042 000003'      HLRZ TAC,DEVSER(TAC1)    ;GET LINK TO NEXT DDB
2524 003560 322040 002160'      JUMPE TAC,CPOPJ        ;0 MEANS END
2525 003561 302046 000000'      CAIE TAC,(DEVDAT)      ;OWNED BY CURRENT USER?
2526 003562 254000 003556'      JRST CLDD81           ;NO
2527
2528 003563 700600 003216'      NOSCHEDULE
2529 003564 200641 000003'      CONO PI,DSKOFF
2530 003565 502642 000003'      MOVE AC1,DEVSER(TAC)    ;GET LINK FROM THIS DDB
2531 003566 550200 000002'      WLLM AC1,DEVSER(TAC1)  ;STORE IN PREVIOUS DDB TO REMOVE THIS ONE
2532 003567 306201 000000'      WRRZ ITEM,DDBPTR      ;GET INTERRUPT LEVEL POINTER
2533 003570 206640 000002'      CAIN ITEM,0(TAC)       ;POINTING TO DISCARDED DDB?
2534
2535 003571 700600 003230'      MOVSM AC1,DDBPTR      ;YES, BYPASS DISCARDED DDB
2536 003572 201200 000010'      SCHEDULE+
2537 003573 254000 004026'      CONO PI,DSKON
2538          JRST CLCOR1
2539          ;BUILD A DISK DEVICE DATA BLOCK AND ASSIGN IF LOGICAL NAME GIVEN.
2540          ;THIS IS DONE AT INIT TIME.
2541
2542          INTERNAL SETDD8
2543
2544 003574 261140 000004' SETDD8: PUSH PDP,ITEM        ;SAVE AC'S USED BY SETDD8
2545 003575 261140 000002'      PUSH PDP,TAC1
2546 003576 200106 000004'      MOVE TAC1,DEVMOD(DEV DAT)
2547 003577 602100 200000'      TRNE TAC1,ASSPRG
2548 003600 254000 003604'      JRST SETDD0
2549 003601 550100 000006'      HRRZ TAC1,DEV DAT
2550 003602 302100 000237'      CAIE TAC1,DSKDD8
2551 003603 254000 003615'      JRST SETDD1
2552 003604 201200 000010' SETDD0: MOVEI ITEM,DSKCOR
2553          NOSCHEDULE+
2554 003605 700600 003563'      CONO PI,DSKOFF
2555 003606 260140 003732'      PUSHJ PDP,GTCCR1
2556 003607 540300 000002'      HRR DEV DAT,TAC1
2557 003610 505100 000237'      HRLI TAC1,DSKDD8
2558 003611 251106 000023'      BLT TAC1,PTR1(DEV DAT)
2559 003612 201100 000237'      MOVEI TAC1,DSKDD8
2560
2561          ; LINK PROTOTYPE DDB TO COPY
2562          ; COPY ALREADY LINKS WHERE PROTOTYPE DID
2563
2564 003613 506302 000003'      HRLM DEV DAT,DEVSER(TAC1)
2565          SCHEDULE+
2566 003614 700600 003571'      CONO PI,DSKON
2567 003615 262140 000002' SETDD1: POP PDP,TAC1
2568 003616 262140 000004'      POP PDP,ITEM
2569 003617 263140 000000'      POPJ PDP,

```

```

2568      ;SET TAC1 TO SOME AREA FOR BUFFER.
2569      ;GET BUFFER SPACE FROM USER INPUT OR OUTPUT RING IF AVAILABLE
2570      ;ASSIGN FROM MONITOR CORE AS A LAST RESORT.
2571      ;SET DSKBUF(DEVDAT) TO POINT TO BUFFER.
2572
2573      EXTERNAL PIOMOD,UADRCK,JOBF
2574
2575      003620 603000 000200 SETBUF: TLNE IOS,UBFS      ;HAVE ONE IN FREE STORAGE?
2576      003621 263140 000000      POPJ PDP,          ;YES, LEAVE
2577      003622 135100 001463'      LDB TAC1,PIOMOD      ;NO, DUMP MODE?
2578      003623 305100 000015      CAIGE TAC1,SD
2579      003624 603300 000020      TLNE DEVDAT,DSKRLR    ;RESET UO IN PROGRESS ?
2580      003625 254000 003647'      JRST SETBF6          ;YES TO EITHER QUESTION
2581
2582      003626 607300 000400      TLNN DEVDAT,INBF      ;INBUF DONE YET?
2583      003627 254000 003632'      JRST SETBF1          ;NO
2584      003630 550106 000007 SETBF1: HRRZ TAC1,DEVIAD(DEVDAT) ;CALL HERE FROM INPUT - RING KNOWN TO BE AVAILABLE
2585      003631 254000 003635'      JRST SETBF2
2586
2587      003632 607300 000200 SETBF1: TLNN DEVDAT,OUTBF      ;OUTBUF DONE YET?
2588      003633 254000 003647'      JRST SETBF6          ;NO
2589      003634 550106 000010 SETBF0: HRRZ TAC1,DEVOAD(DEVDAT) ;CALL HERE FROM OUTPUT - RING KNOWN TO BE AVAILABLE
2590
2591      003635 201102 000002 SETBF2: MOVEI TAC1,2(TAC1)
2592
2593      003636 550640 000002      HRRZ AC1,TAC1          ;ADDRESS CHECK LOW END
2594      003637 260140 000000      PUSHJ PDP,UADRCK
2595      003640 201642 000177      MOVEI AC1,BLKSI2-1(TAC1)
2596      003641 260140 003637'      PUSHJ PDP,UADRCK
2597      003642 661000 001000      TLO IOS,UBFU          ;BUFFER IN USER AREA
2598      003643 505100 777600 SETBF5: HRLI TAC1,NBLKSZ
2599      003644 202106 000020      MOVEM TAC1,DSKBUF(DEVDAT)
2600      003645 202006 000002      MOVEM IOS,DEVIOS(DEVDAT)
2601      003646 263140 000000      POPJ PDP,
2602      ;DUMP MODE, USE AREA IN FREE STORAGE.
2603
2604      003647 352000 000000 SETBF6: AOSE MOREQ
2605      003650 260140 000000      PUSHJ PDP,MQWAIT
2606      003651 661000 000200      TLO IOS,UBFS          ;BUFFER IN MONITOR CORE
2607      003652 201100 000266'      MOVEI TAC1, MONBUF    ;TAC1 POINTS TO MONITOR BUFFER
2608      003653 254000 003643'      JRST SETBF5
  
```

```

2609          ;CLEAR THE BUFFER IN FREE STORAGE IF THERE.
2610
2611          EXTERNAL BUFCLR,JOBFF,IADRCK
2612          CLRBUF: SCHEDULE+
2613          003654 700600 003614' CONO PI,DSKON
2614          003655 627000 004000 TLZN IOS,DAFLG          ;USING SAT BLOCK?
2615          003656 254000 003661' JRST ,+3
2616          003657 371000 003414' SOSL DAREQ          ;YES, RELEASE
2617          003660 476000 003405' SETOM DAAVAL
2618          003661 627000 010000 TLZN IOS,AUFLG          ;USING UFD?
2619          003662 254000 003665' JRST ,+3
2620          003663 371000 002123' SOSL AUREQ          ;YES, RELEASE UFD
2621          003664 476000 002124' SETOM AUAVL
2622          003665 627000 000200 TLZN IOS,UBFS          ;USING MONIGOR BUFFER?
2623          003666 254000 003671' JRST ,+3
2624          003667 371000 003647' SOSL MOREQ          ;YES, RELEASE MONITOR BUFFER
2625          003670 476000 000000 SETOM MQAVL
2626          003671 621000 002000 TLZ IOS,NCTRLC          ;RENABLE +C
2627          003672 623000 001000 TLZE IOS,UBFU          ;USING BUFFER IN USER AREA
2628          003673 254000 003676' JRST CLRBF1          ;YES, CLEAR BUFFER
2629          003674 202000 000002 MOVEM IOS,DEVIOS(DEVDAT)
2630          003675 263140 000000 POPJ PDP,
2631
2632          ;CLEAR USER BUFFER AFTER USING
2633
2634          003676 202000 000002 CLRBF1: MOVEM IOS,DEVIOS(DEVDAT)
2635          003677 550046 000020 HRRZ TAC,DSKBUF(DEVDAT)          ;GET LOC. OF BUFFER
2636          003700 311047 000000 CAML TAC,JOBFF(JDAT)          ;ABOVE JOBFF?
2637          003701 263140 000000 POPJ PDP,          ;NO, RETURN,
2638          003702 260140 000000 PUSHJ PDP,IADRCK
2639          003703 263140 000000 POPJ PDP,
2640          003704 275040 000002 SUBI TAC,2          ;SET TAC TO 2ND BUFFER WORD
2641          003705 370003 000000 SOS (PDP)          ;BUFCLR WILL SKIP RETURN
2642          003706 254000 000000 JRST BUFCLR          ;CLEAR THE BUFFER
  
```

```

2643      ;FIND THE FIRST ZERO IN A SET OF WORDS,
2644      ;ENTER WITH LH OF TAC SET TO NEGATIVE OF NUMBER OF WORDS,
2645      ;      RH OF TAC CONTAINING ADDRESS OF FIRST WORD,
2646      ;CALL:  PUSHJ PDP,GETZER
2647      ;      EXIT1      NO ZEROES
2648      ;      EXIT2      NORMAL
2649      ;NORMAL EXIT LEAVES TAC POINTING TO WORD CONTAINING THE ZERO,
2650      ;AND TAC1 HAVING A 1-BIT IN THE POSITION OF THE ZERO.
2651      ;
2652      ;ENTER AT GETZER1 IF TAC1 IS ALL SET UP.
2653      ;CALLED AT UUO AND INTERRUPT LEVELS
2654
2655      003707 211240 000001 GETZER: MOVNI DAT,1
2656      003710 254000 003716' JRST GETZ3
2657
2658      003711 241100 777777 GETZ1:  ROT TAC1,-1
2659      003712 325100 003716' JUMPGE TAC1,GETZ3
2660
2661      003713 252040 003560' GETZ2:  AOBJP TAC,CPOPJ
2662      003714 316241 000000      CAMN DAT,(TAC)
2663      003715 254000 003713' JRST GETZ2
2664
2665      003716 612101 000000 GETZ3:  TONE TAC1,(TAC)
2666      003717 254000 003711' JRST GETZ1
2667      003720 254000 002156' JRST CPOPJ1
2668
2669      ;TIMING:      M=NUMBER OF LEADING WORDS CONTAINING NO ZEROES
2670      ;      N=NUMBER OF LEADING ONES IN M+1ST WORD
2671
2672      ;      38+12.8M+17.1N  MICRO-SECS
2673
2674      ;WORST CASE WITH 128 WORDS = 2.3 MILS
2675
2676
2677      ;FIND THE NUMBER OF ZERO BITS TO THE RIGHT OF A WORD
2678      ;CONTAINING A SINGLE 1-BIT
2679      ;ENTER WITH BIT IN TAC1, EXIT WITH RESIDUE IN TAC1.
2680      ;CALLED AT UUO AND INTERRUPT LEVELS
2681
2682      003721 607100 777000 RESIDU: TLNN TAC1,777000
2683      003722 254000 003725' JRST RESID1
2684      003723 242100 777767      LSH TAC1,-11
2685      003724 665100 043000      TLOA TAC1,43000
2686      003725 661100 032000 RESID1: TLO TAC1,32000
2687      003726 140100 004615' FAD TAC1,[01
2688      003727 242100 777745      LSH TAC1,-33
2689      003730 263140 000000      POPJ PDP,

```

```

2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705

```

003731	201200	000001	GETFCR: MOVEI ITEM,1		
003732	201640	000262	GTCCR1: MOVEI AC1,LOCORE		IREQUEST ONE 4 WORD BLOCK
003733	201700	000265	MOVEI AC2,DDRTAB		IPPOINT TO THREE WORD FREE CORE ENTRY
003734	261140	000005	PUSH PDP,DAT		IPPOINT TO BIT TABLE
003735	260140	003744	PUSHJ PDP,GETBIT		
003736	254000	004421	JRST DFERR9		IFIND A HOLE
003737	262140	000005	POP PDP,DAT		IFHERE ARE NONE
003740	242100	000002	LSH TAC1,2		
003741	275100	000004	SURJ TAC1,4		IMULTIPLY BIT# BY 4
003742	270100	000262	ADD TAC1,LOCORE		ISYNCHRNIZE
003743	263140	000000	POPJ PDP,		IFORM CORE ADDRESS

```

2706                                INTERNAL GETBIT
2707
2708
2709                                ;SEARCH THROUGH A TABLE TO FIND A SERIES OF ZERO-BITS.
2710                                ;ENTER WITH AC1 SET UP AS IN CLRRIT ROUTINE, C(ITEM) SET TO THE
2711                                ;   SIZE HOLE DESIRED.
2712                                ;CALLED AT UWO AND INTERRUPT LEVELS
2713                                ;ON UNSUCCESSFUL RETURN DAT CONTAINS THE SIZE
2714                                ;OF TARGETS HOLE ENCOUNTERED
2715
2716 003744 261140 004616' GETBIT: PUSH PDP,-1J                                ;INITIALIZE LARGEST HOLE FOUND SO FAR
2717 003745 261140 000004'                                PUSH PDP,ITEM                                ;SAVE HOLE SIZE
2718 003746 200055 000002'                                MOVE TAC,2(AC1)                                ;TAC:=IOWD LENGTH OF TABLE, TABLE ADDRESS
2719 003747 200115 000001'                                MOVE TAC1,1(AC1)
2720 003750 260140 003707'                                PUSHJ PDP,GETZER                                ;FIND THE FIRST ZERO
2721 003751 254000 004016'                                JRST GTBIT5                                ;NO ZEROES LEFT
2722 003752 202115 000001'                                MOVEM TAC1,1(AC1)                                ;SAVE THE SPOT
2723 003753 202055 000002'                                MOVEM TAC,2(AC1)
2724 003754 261140 000001'                                PUSH PDP,TAC
2725 003755 261140 000002'                                PUSH PDP,TAC1
2726
2727 003756 350000 000005' GTRIT1: AOS DAT                                ;INCREMENT SIZE OF HOLE BY 1 (GETZER LEFT DAT=-1)
2728 003757 363200 003776'                                SOJLE ITEM,GTBIT2                                ;FOUND ENOUGH?
2729 003760 241100 777777'                                ROT TAC1,-1                                ;NO
2730 003761 335000 000002'                                SKIPGE TAC1
2731 003762 252040 003765'                                AOBJP TAC,GTBIT1A                                ;JUMP IF TABLE EXHAUSTED
2732 003763 616101 000000'                                TONN TAC1,(TAC)                                ;IS NEXT BIT ZERO?
2733 003764 254000 003756'                                JRST GTBIT1                                ;YES, CONTINUE
2734
2735 003765 313243 777775' GTRIT1A: CAMLE DAT,-3(PDP)                                ;NO, LARGEST HOLE SO FAR?
2736 003766 202243 777775'                                MOVEM DAT,-3(PDP)                                ;YES, SAVE SIZE
2737 003767 327040 004021'                                JUMPG TAC,GTBIT6                                ;TABLE EXHAUSTED? IF SO, ERROR EXIST,
2738 003770 260140 003707'                                PUSHJ PDP,GETZER                                ;NO, TRY FURTHER ON
2739 003771 254000 004021'                                JRST GTBIT6                                ;NO MORE, ERROR
2740 003772 202103 000000'                                MOVEM TAC1,(PDP)
2741 003773 202043 777777'                                MOVEM TAC,-1(PDP)
2742 003774 200203 777776'                                MOVE ITEM,-2(PDP)
2743 003775 254000 003756'                                JRST GTBIT1
2744
2745 003776 200103 000000' GTRIT2: MOVE TAC1,(PDP)                                ;HOLE FOUND, SET BITS TO ONES
2746 003777 200043 777777'                                MOVE TAC,-1(PDP)
2747 004000 200203 777776'                                MOVE ITEM,-2(PDP)
2748
2749 004001 436101 000000' GTRIT3: ORM TAC1,(TAC)
2750 004002 363200 004006'                                SOJLE ITEM,GTBIT4
2751 004003 241100 777777'                                ROT TAC1,-1
2752 004004 325100 004001'                                JUMPG TAC1,GTRIT3
2753 004005 253040 004001'                                AOBJN TAC,GTRIT3                                ;THIS SHOULD ALWAYS JUMP
2754
2755 004006 262140 000002' GTRIT4: POP PDP,TAC1
2756 004007 262140 000001'                                POP PDP,TAC
2757 004010 350003 777776'                                AOS -2(PDP)
2758

```

2759	004011	553000	000001	HRRZS TAC	
2760	004012	275056	777777	SUBI TAC,-1(AC2)	
2761	004013	260140	003721	PUSHJ PDP,RESIDU	
2762	004014	221040	000044	IMULI TAC,*D36	
2763	004015	276040	000002	SURM TAC,TAC1	
2764	004016	262140	000004	GTBIT5: POP PDP,ITEM	
2765	004017	262140	000005	POP PDP,DAT	;SIZE OF LARGEST HOLE FOUND IF UNSUCCESSFUL
2766	004020	263140	000000	POPJ PDP,	
2767					
2768	004021	262140	000001	GTBIT6: POP PDP,TAC	
2769	004022	262140	000002	POP PDP,TAC1	
2770	004023	350003	777777	AOS -1(PDP)	;SIZE OF LARGEST HOLE MUST BE ADJUSTED.
2771	004024	254000	004016	JRST GTBIT5	

```

2772          ;FREE A FOUR WORD SECTION OF CORE.
2773          ;ENTER WITH ADDRESS IN TAC
2774
2775          INTERNAL CLCOR1
2776
2777 004025 201200 000001 CLRCOR: MOVE! ITEM,1
2778 004026 274040 000262' CLCOR1: SUB TAC,LOCORE
2779 004027 242040 777776      LSH TAC,-2
2780 004030 201640 000262'      MOVE! AC1,LOCORE
2781 004031 200700 000265'      MOVE AC2,CRINIT      ;SEARCH FROM BEGINNING
2782
2783          ;CLEAR A SERIES OF BITS IN SOME TABLE
2784          ;ENTER WITH:      C(TAC) SET TO A BIT NUMBER
2785          ;                  C(AC1) SET TO EITHER A SAT ENTRY OR FREE
2786          ;                  CORE ENTRY
2787          ;                  C(AC2) SET TO BEGINNING OF TABLE (IOWD FORM)
2788          ;                  C(ITEM) SET TO NUMBER OF BITS TO CLEAR
2789          ;EXIT WITH CORRECT BITS SET TO ZERO AND SAT ENTRY OR FREE CORE
2790          ;      ENTRY UPDATED IF NECESSARY.

2791
2792          INTERNAL CLRRIT
2793
2794 004032 231040 000044 CLRBIT: IDIVI TAC,*D36
2795 004033 507000 000001      HRLS TAC      ;BOTH HALVES OF TAC CONTAIN
2796          ;      WORD INCREMENT
2797 004034 270040 000016      ADD TAC,AC2
2798 004035 261140 000005      PUSH PDP,DAT
2799 004036 210240 000002      MOVN DAT,TAC1
2800 004037 205100 400000      MOVS! TAC1,400000
2801 004040 241105 000000      ROT TAC1,(DAT)
2802 004041 262140 000005      POP PDP,DAT
2803 004042 261140 000001      PUSH PDP,TAC
2804 004043 261140 000002      PUSH PDP,TAC1
2805
2806 004044 412101 000000 CLRB4: ANDCAM TAC1,(TAC)      ;CLEAR A BIT
2807 004045 363200 004051'      SOJLE ITEM,CLRB5      ;IS THAT ALL?
2808 004046 241100 777777      ROT TAC1,-1      ;NO
2809 004047 325100 004044'      JUMPGE TAC1,CLRB4
2810 004050 253040 004044'      AOBJN TAC,CLRB4      ;THIS SHOULD ALWAYS JUMP
2811
2812 004051 262140 000002 CLRBS: POP PDP,TAC1
2813 004052 262140 000001      POP PDP,TAC
2814 004053 313055 000002      CAMLE TAC,2(AC1)      ;NEED TO RESET?
2815 004054 263140 000000      POPJ PDP,      ;NO
2816 004055 312055 000002      CAME TAC,2(AC1)
2817 004056 254000 004063'      JRST CLRB2      ;YES
2818 004057 321100 004063'      JUMPL TAC1,CLRB2
2819 004060 335015 000001      SKIPGE 1(AC1)
2820 004061 254000 004064'      JRST CLRB3
2821 004062 311115 000001      CAML TAC1,1(AC1)
2822 004063 202115 000001 CLRBS: MOVEM TAC1,1(AC1)
2823 004064 202055 000002 CLRBS: MOVEM TAC,2(AC1)
2824 004065 263140 000000      POPJ PDP,

```

```

2825          SURTTL QUEUEING AND INTERRUPT PROCESSING
2826          ;PUT A REQUEST IN THE USER QUEUE.
2827          ;ENTER AT QIN FOR INPUT, QOUT FOR OUTPUT.
2828          ;CALLED AT BOTH INTERRUPT AND I/O LEVELS
2829
2830          EXTERNAL SETACT,CPOPJ
2831
2832 004066 334120 000006 QIN:   SKIPA TAC1,DEV DAT      ;*
2833 004067 201106 000001 QOUT:  MOVEI TAC1,DEV DAD-DEVIAD(DEV DAT) ;*
2834
2835 004070 201040 000003          MOVEI TAC,TRIES          ;*INITIALIZE ERROR COUNT
2836 004071 202046 000021          MOVEM TAC,DSK CNT(DEV DAT) ;*
2837 004072 260140 000000          PUSHJ PDP,SETACT        ;*SET IOACT
2838 004073 205040 400000          MOVSI TAC,W8BIT         ;*SET WAIT FLAG
2839 004074 436042 000007          ORM TAC,DEVIAD(TAC1)     ;*
2840          NOSCHEDULE ;*
2841 004075 700600 003605'        CONO PI,DSKOFF
2842 004076 350000 000006'        AOS USRCNT              ;*INCREMENT COUNTER OF TRANSFER REQUESTS
2843
2844 004077 211240 000001          MOVNI DAT,1              ;*
2845 004100 250240 000007'        EXCH DAT,OFBUSY          ;*
2846 004101 326240 004106'        JUMPN DAT,QIOEND        ;*JUMP IF DISK BUSY
2847
2848 004102 412042 000007          ANDCAM TAC,DEVIAD(TAC1) ;*
2849 004103 312100 000006          CAME TAC1,DEV DAT       ;*
2850 004104 201040 000000          MOVEI TAC,0             ;*
2851 004105 260140 004267'        PUSHJ PDP,USRG0         ;*
2852
2853 004106          QIOEND: SCHEDULE ;*
2854 004106 700600 003654'        CONO PI,DSKON
2855 004107 263140 000000          POPJ PDP,                ;*
  
```

```

2856          ;PUT A REQUEST IN THE MONITOR QUEUE.
2857          ;ENTER AT MQIN FOR INPUT, MQOUT FOR OUTPUT.
2858          ;ENTER WITH      C(TAC1 0-17) = -SIZE OF BLOCK
2859          ;                  C(TAC1 18-35) = CORE ADDRESS
2860          ;                  C(TAC 18-35) = LOGICAL BLOCK NUMBER
2861
2862          INTERNAL MQIN,MQOUT
2863          EXTERNAL MOREQ,MQWAIT,SETACT,MQUEUE,WSYNC
2864
2865 004110 514100 000006 MQIN:   HRLZ TAC1,DEV DAT
2866 004111 665100 400000      TLOA TAC1,400000      ;BIT 0=1 IS READ INDICATOR
2867 004112 514100 000006 MQOUT:  HRLZ TAC1,DEV DAT
2868          IFN FTRCHK,<
2869          TRNN TAC,-1      ;WRITING OR READING BLOCK NR 0 SHOULD NEVER OCCUR HERE
2870 004113 606040 777777
2871 004114 254200 004115'      HALT ,+1      ;CONTINUING AFTER HALT WILL YIELD INFORMATION
2872          ; LOSS ON THE DISK
2873
2874 004115 505040 000003      >
2875          HRLI TAC,TRIES
2876
2875 004116 206046 000021      MOVSM TAC,DSKCNT(DEV DAT);STASH AWAY BLOCK #, ERROR COUNT.
2876 004117 260140 004072'      PUSHJ PDP,SETACT      ;SET IOACT
2877          NOSCHEDULE*
2878 004120 700600 004075'      CONO PI,DSKOFF
2879 004121 607000 000600      TLNN IOS,URFS*NORELB      ;USING BUFFER IN FREE STORAGE?
2880 004122 540100 000007      HRR TAC1,PROC      ;NO, RELOCATE
2881 004123 202120 000004'      MOVEM TAC1,MIPTR
2882
2883 004124 350640 000004'      AOS AC1,MIPTR      ;RING INDEX MIPTR
2884 004125 301640 000000      CAIL AC1,MQTOP
2885 004126 201640 000560'      MOVEI AC1,MQUEUE
2886 004127 202640 000004'      MOVEM AC1,MIPTR
2887
2888 004130 211640 000001      MOVNI AC1,1
2889 004131 250640 000007'      EXCH AC1,DFBUSY      ;FORCE DFBUSY TO-1, EXTRACT PREVIOUS CONTENTS.
2890 004132 336000 000015      SKIPN AC1      ;ALREADY BUSY?
2891 004133 260140 004307'      PUSHJ PDP,MONGO      ;NO, START A MONITOR JOB
2892
2893          SCHEDULE*
2894 004134 700600 004106'      CONO PI,DSKON
2895 004135 260140 003130'      PUSHJ PDP,WSYNC
2896 004136 550646 000021      WRRZ AC1,DSKCNT(DEV DAT) ;ANY ERRORS?
2897 004137 606640 700000      TRNN AC1,IODERR:IODTER:IOIMPM
2898 004140 350003 000000      AOS (PDP)      ;NO, SKIP RETURN
2899 004141 263140 000000      POPJ PDP,

```

```

2900                                ;*
2901                                ;* EVERYTHING FROM HERE THRU SETDUN CALLED AT INTERRUPT LEVEL
2902                                ;*
2903                                ;ROUTINE CALLED BY INTERRUPT IN DISK.
2904                                ;CHECK JOB FOR ERRORS, THEN START UP ANY WAITING JOB.
2905                                ;UPON ENTRY, THE ACS HAVE BEEN SAVED, IOS CONTAINS ANY ERROR
2906                                ;FLAGS.
2907
2908                                INTERNAL DFINT,FTSWAP
2909                                EXTERNAL ADVBFE,ADVBFF,PJOBN,SETIOD,PIOMOD
2910                                EXTERNAL MQREQ,MQAVL,DFRED,DFWRT
2911                                EXTERNAL JBTADR
2912                                EXTERNAL MQTOP,MQUEUE
2913
2914
2915                                DFINT:
2916                                IFN FTSWAP,<
2917                                EXTERNAL SQREQ,SERA,SQCO,SWPINT
2918                                MOVE TAC,SERA                                ;*WAS THAT A SWAPPING JOB?
2919                                TLNE TAC,200000                                ;*
2920                                JRST SWPINT                                ;*YES
2921                                >
2922                                SKIPN RUNUSR                                ;*WAS THAT A USER JOB?
2923                                JRST DFINT2                                ;*NO, MUST HAVE BEEN MONITOR
2924
2925                                MOVS DEVDAT,RUNUSR                                ;*YES
2926                                ANDI DEVDAT,377777                                ;*CLEAR READ INDICATOR BIT,
2927                                TRNE IOS,I0IMPM                                ;*WRITE-LOCK?
2928                                PUSHJ PDP,DINT1A                                ;*YES
2929                                TRNE IOS,I0DTER!IODERR                                ;*ERRORS?
2930                                JRST DFINT9                                ;*ERRORS
2931                                AOS DEVACC(DEVDAT)                                ;INCREMENT POINTER TO RETRIEVAL POINTER
2932                                AOS SETCNT(DEVDAT)                                ;INCREMENT RELATIVE BLOCK NUMBER
2933                                SOS USRCNT                                ;*
2934                                PUSHJ PDP,SETDUN                                ;*TURN OFF I/O WAIT
2935                                LDB TAC,PJOBN
2936                                MOVE PROG,JBTADR(TAC)                                ;*SET RELOC. AND PROT, FOR THIS JOB
2937
2938                                SKIPL RUNUSR                                ;*WAS IT A READ OR A WRITE?
2939                                JRST DFINT1                                ;*WRITE
2940
2941                                MOVEI TAC,@DEVIAD(DEVDAT) ;*SET UP WORD COUNT
2942                                ADDI TAC,1                                ;*
2943                                MOVEI TAC1,BLKSI2
2944                                WRRM TAC1,@TAC
2945

```

```

2946      IFG CHKCNT,<
2947      MOVE TAC,DEV0AD(DEV DAT)
2948      TLNE TAC,CKSUPR
2949      JRST DINT0R
2950      HRRZ TAC,DEVIAD(DEV DAT)
2951      ADD TAC,[XWD NBLKSZ,2]
2952      PUSHJ PDP,CHKSUM          ICLEAR IOACT
2953      MOVE TAC,DEVACC(DEV DAT)
2954      HLRZ TAC,-1(TAC)
2955      CAMN TAC,TAC1
2956      JRST .+3
2957      PUSHJ PDP,CKREC3
2958      MOVEM IOS,DEVIOS(DEV DAT)
2959      >
2960      DINT0B:
2961      PUSHJ PDP,DFINX          IENTER INPUT ROUTINE
2962      SKIP
2963      DFINT1: PUSHJ PDP,DFOUTX    ICHECK FOR MORE OUTPUT
2964      SETZM RUNUSR

2965      JRST DFINT4
2966
2967      DINT1A: HRRZ TAC,DEVACC(DEV DAT) IWRITE-LOCK FOUND, SET WLBIT
2968      HRRZ TAC,-1(TAC)
2969      JRST SETWL              IAND RETURN
2970
2971      DFINT2: MOVS DEV DAT,@MOPTR    IIT WAS A MONITOR JOB
2972      ANDI DEV DAT,377777        IREMOVE POSSIBLE READ INDICATOR
2973      TRNN IOS,IODERR:IODTER    IERRORS?
2974      JRST DFINT3              INO,
2975      SOS TAC,DSKCNT(DEV DAT)    IYES,TRIED ENOUGH?
2976      TRNE TAC,-1
2977      JRST DINT4A              INO, TRY AGAIN.
2978
2979      DFINT3: HRRM IOS,DSKCNT(DEV DAT) ISAVE ANY ERRORS
2980      HLRZ TAC,DSKCNT(DEV DAT) IWRITE-LOCK ERROR?
2981      TRNE IOS,IOIMPM
2982      PUSHJ PDP,SETWL          IYES, SET WLBIT
2983      PUSHJ PDP,SETDUN
2984      SETZM @MOPTR
2985      AOS TAC,MOPTR            IREMOVE THIS MONITOR JOB
2986      CAIL TAC,MQTOP          IINCREMENT POINTER,
2987      MOVEI TAC,MQUEUE
2988      MOVEM TAC,MOPTR          IFALL THROUGH TO DFINT4
  
```

```

2999
2990
2991
2992 004236 202006 000002 ;DETERMINE NEXT TRANSFER TO MAKE ACCORDING TO PRIORITY
2993 004237 ; 1) SWAPPING 2) MQ I/O 3) BUFFERED MODE I/O
2994 004237 332000 000524' DFINT4: MOVEM IOS,DEVIOS(DEVSTAT) ;CLEAR IOACT
2995 004240 254000 000000 DINT4A: IFN FTSWAP,<
                SKIPE SQREQ ;SWAPPING JOB WAITING?
                JRST SOGO ;YES,START IT
2996
2997
2998
2999 004241 332020 000003' DINT4B: SKIPE @MOPTR ;MONITOR JOB WAITING?
3000 004242 254000 004307' JRST MONGO ;YES, START IT UP
3001
3002 004243 331000 000006' SKIPL USRCNT ;USER WAITING?
3003 004244 254000 004247' JRST FINDV ;YES.
3004 004245 402000 000007' SETZM OFRUSY ;NO. TURN OFF BUSY FLAG
3005 004246 263140 000000 POPJ PDP, ;EXIT INTERRUPT
  
```

```

3006                ;FIND A WAITING USER
3007
3008 004247 550300 000002' FINDV: HRRZ DEVDAT,DDBPTR
3009 004250 205040 400000      MOVSI TAC,W0BIT      ;PREPARE TO TURN OFF WAIT INDICATOR
3010
3011 004251 554106 000000 DFINT5: HLRZ TAC1,(DEVDAT)      ;END OF DISK DDBS?
3012 004252 332000 000006      SKIPE DEVDAT
3013 004253 302100 446353      CAIE TAC1,(SIXBIT /DSK/)
3014 004254 201300 000237'      MOVEI DEVDAT,DSK0DB      ;YES, RESET TO TOP
3015
3016 004255 335006 000007      SKIPGE DEVIAD(DEVDAT)      ;ANY READS WAITING?
3017 004256 254000 004266'      JRST DFINT7              ;YES
3018 004257 335006 000010      SKIPGE DEVOAD(DEVDAT)      ;ANY WRITES WAITING?
3019 004260 254000 004263'      JRST DFINT6              ;YES
3020 004261 554306 000003      HLRZ DEVDAT,DEVSER(DEVDAT) ;NO, TRY NEXT DDB
3021 004262 254000 004251'      JRST DFINT5
3022
3023                ;START UP A USER WRITE
3024
3025 004263 412046 000010 DFINT6: ANDCAM TAC,DEVOAD(DEVDAT) ;CLEAR OUTPUT WAIT BIT IN DDB
3026 004264 201040 000000      MOVEI TAC,0              ;PREPARE TO SET BIT 0=0 IN RUNUSR
3027 004265 254000 004267'      JRST USRG0
3028
3029                ;START UP A USER READ
3030
3031
3032 004266 412046 000007 DFINT7: ANDCAM TAC,DEVIAD(DEVDAT) ;CLEAR INPUT WAIT BIT IN DDB
3033
3034 004267 206300 000005' USRG0: MOVSM DEVDAT,RUNUSR      ;SAVE DEVDAT FOR USE AT DFINT
3035 004270 436040 000005'      ORM TAC,RUNUSR            ;SET READ (R0=1) WRITE (R0=0) INDICATION
3036 004271 350000 000013'      AOS UXFERS              ;COUNT TOTAL USER TRANSFERS
3037 004272 135340 004161'      LDB PROG,PJOBIN          ;SETUP RELOCATION FOR JOB
3038 004273 200347 004162'      MOVE PROG,JBTADR(PROG)
3039 004274 200046 000003      MOVE TAC,DEVSER(DEVDAT)    ;GET DDB LINK
3040 004275 206040 000002'      MOVSM TAC,DDBPTR          ;START SEARCH WITH NEXT DDB
3041 004276 200046 000014      MOVE TAC,DEVACC(DEVDAT)    ;GET POINTER LOCATION
3042 004277 550041 000000      HRRZ TAC,(TAC)            ;GET RETRIEVAL POINTER (XWD CHECKSUM,BLOCK #)
3043 004300 205100 777600      MOVSI TAC1,NBLKSZ
3044 004301 331000 000005'      SKIPL RUNUSR              ;READ?
3045 004302 254000 004305'      JRST USRG03
3046
3047 004303 541126 000007      HRRI TAC1,@DEVIAD(DEVDAT) ;GET ADDRESS OF RING BUFFER
3048 004304 344100 000000      AOJA TAC1,DFRED            ;FORM IOWD AND READ
3049
3050 004305 541126 000010 USRG03: HRRI TAC1,@DEVOAD(DEVDAT) ;GET ADDRESS OF OUTPUT BUFFER
3051 004306 344100 000000      AOJA TAC1,DFWRT            ;FORM IOWD AND WRITE

```

```

3052
3053          ;START UP A MONITOR READ OR WRITE
3054
3055 004307 204220 000003' MONGO: MOVS ITEM,@MOPTR          ;XWD CORE ADDRESS, SAVED DEVDAT
3056 004310 421200 400000      ANDCM1 ITEM,400000      ;CLEAR READ INDICATOR
3057 004311 200104 000020      MOVE TAC1,DSKBUF(ITEM)    ;GET CONTROLLING IOWD
3058 004312 554040 000004      HLRZ TAC,ITEM            ;GET CORE ADDRESS
3059 004313 270100 000001      ADD TAC1,TAC
3060          HLL TAC1,DSKBUF(ITEM)          ;RESTORE LH IN CASE ADDING REL. CAUSES OVRFL IN
3061 004314 500104 000020 T LH      SUBI TAC1,1          ;FORM IOWD LENGTH ADDRESS
3062 004315 275100 000001      HLRZ TAC,DSKCNT(ITEM)      ;GET LOGICAL BLOCK#
3063 004316 554044 000021      SKIPL @MOPTR
3064 004317 331020 000003'      JRST DFVRT
3065 004320 254000 004306'      JRST DFVRT
3066 004321 254000 004304'
3067
3068          ;ERROR ON USER JOB.
3069
3070 004322 350000 000014' DFINT9: AOS ECOUNT          ;COUNT USER HARDWARE ERRORS
3071 004323 370046 000021      SOS TAC,DSKCNT(DEV DAT)    ;TRIED ENOUGH?
3072 004324 606040 777777      TRNN TAC,-1
3073 004325 254000 004155'      JRST DFINTA              ;YES.
3074 004326 200040 000006      MOVE TAC,DEV DAT          ;NO, RESET FLAG SO IT WILL GO AGAIN.
3075 004327 331000 000005'      SKIPL RUNUSR            ;READ?
3076 004330 271040 000001      ADDI TAC,DEV DAT-DEVIAD    ;NO.
3077 004331 205100 400000      MOVSI TAC1,400000
3078 004332 436101 000007      ORM TAC1,DEVIAD(TAC)      ;RESTORE WAIT BIT TO DEVIAD OR DEV DAT
3079 004333 402000 000005'      SETZM RUNUSR            ;CLEAR USER INDICATOR
3080 004334 254000 004237'      JRST DINT4A              ;DO NEXT TASK.
3081
3082          ;CLEAR I/O DONE FLAG IF IT IS ON.
3083
3084 004335          SETDUN:
3085 004335 434006 000002      OR IOS,DEV IOS(DEV DAT)    ;*
3086 004336 421000 010000      ANDCM1 IOS,IOACT          ;*
3087 004337 623000 000001      TLZE IOS,IOW              ;*
3088 004340 254000 000000      JRST SETIOD              ;*
3089 004341 263140 000000      POPJ PDP,                ;*

```

```

3090          SUBTTL ERRORS, SUBROUTINES,
3091          ;ERROR ROUTINES CALLED FROM UUD LEVEL
3092
3093          EXTERN EXCALP,ERRPTU,TPOPJ
3094
3095          DEFINE ERRORS (X),<
3096          PUSHJ PDP,CLRBUF          ;CLEAR ANY DUMP BUFFER
3097          JSP TAC,ERRPTU           ;OUTPUT A MESSAG
3098          ASCIZ *X,
3099          JRST EXCALP              ;TYPE "UUD FROM" MESSAGE AND QUIT
3100          >
3101
3102          004342          DFERR1: ERRORS          <DISK FULL-- YOUR JOB HAS LOST, >*
3103          004342          260140          003654' PUSHJ PDP,CLRBUF          ;CLEAR ANY DUMP BUFFER
3104          004343          265040          000000' JSP TAC,ERRPTU           ;OUTPUT A MESSAG
3105          004344          045710          444646' ASCIZ * <DISK FULL-- YOUR JOB HAS LOST, >*
3106          004345          455010          652630'
3107          004346          461325          520262'
3108          004347          476532          220224'
3109
3110          004350          476044          044202'
3111          004351          515011          447646'
3112          004352          521344          020174'
3113          004353          000000          000000'
3114          004354          254000          000000'          JRST EXCALP              ;TYPE "UUD FROM" MESSAGE AND QUIT
3115
3116          004355          201000          400000' DFERR2: MOVEI IOS,IOIMPM
3117          004356          437006          000002' ORB IOS,DEVIOS(DEV DAT)
3118          004357          263140          000000' POPJ PDP,
3119
3120          004360          DFERR3: ERRORS <INCORRECT RETRIEVAL INFORMATION>*
3121          004360          260140          003654' PUSHJ PDP,CLRBUF          ;CLEAR ANY DUMP BUFFER
3122          004361          265040          004343' JSP TAC,ERRPTU           ;OUTPUT A MESSAG
3123          004362          446350          347644' ASCIZ *INCORRECT RETRIEVAL INFORMATION*
3124          004363          512130          352100'
3125          004364          512132          451222'
3126          004365          426550          146100'
3127          004366          446350          647644'
3128          004367          466032          444636'
3129          004370          470000          000000'
3130          004371          254000          004354'          JRST EXCALP              ;TYPE "UUD FROM" MESSAGE AND QUIT
3131
3132          ;THESE ERROR ROUTINES ARE GENERALLY CALLED AFTER SETLE HENCE 0(PDP) IS XWD PROJ,PROG
3133
3134          004372          201040          000003' DFERR6: MOVEI TAC,NORITE          ;FILE ALREADY BEING ALTERED
3135          004373          550100          000014' DFERR7: HRRZ TAC1,UUD
3136          004374          270100          004605' ADD TAC1,CXWD PROJ,ERRPTU ;FORM POINTER TO UUD PARAMETER BLOCK
3137          004375          542060          000002' HRRM TAC,*TAC1          ;STORE ERROR CODE
3138          004376          262140          000001' POP PDP,TAC          ;REMOVE XWD PROJ,PROG
3139          004377          254000          003654' JRST CLRBUF          ;RELEASE ANY RESOURCES
3140
3141          004400          201040          000002' DFERR5: MOVEI TAC,PROTF          ;PROTECTION FAILURE
3142          004401          254000          004373' JRST DFERR7
3143
3144          3129
  
```

```

3130 004402 201040 000001 DFERR4: MOVEI TAC,NOTINM ;USER NOT IN MFD
3131 004403 254000 004373' JRST DFERRY
3132
3133 004404 201040 000000 DFERR7: MOVEI TAC,NOTINU ;FILE NOT IN UFD
3134 004405 254000 004373' JRST DFERRY
3135
3136 004406 DFERR8: ERRORS <LOOKUP AND ENTER HAVE DIFFERENT NAMES>+
3137 004406 260140 003654' PUSHJ PDP,CLRBUF ;CLEAR ANY DUMP BUFFER
3138 004407 265040 004361' JSP TAC,ERRPTU ;OUTPUT A MESSAG
3139 004410 462371 745652 ASCII? *LOOKUP AND ENTER HAVE DIFFERENT NAMES+
004411 501010 147210
004412 202131 652212
004413 511011 040654
004414 425010 444614
004415 432132 242634
004416 521011 640632
004417 426460 000000
3140 004420 254000 004371' JRST EXCALP ;TYPE "UUD FROM" MESSAGE AND QUIT
3141
3142 004421 DFERR9: SCHEDULE+
3143 004421 700600 004134' CONO PI,DSKON
3144 ERRORS <NOT ENOUGH FREE CORE IN MONITOR>+
3145 004422 260140 003654' PUSHJ PDP,CLRBUF ;CLEAR ANY DUMP BUFFER
3146 004423 265040 004407' JSP TAC,ERRPTU ;OUTPUT A MESSAG
3147 004424 472372 420212 ASCII? *NOT ENOUGH FREE CORE IN MONITOR+
004425 472372 543620
004426 202152 242612
004427 202071 751212
004430 202231 620232
004431 476351 152236
004432 510000 000000
3148 004433 254000 004420' JRST EXCALP ;TYPE "UUD FROM" MESSAGE AND QUIT
3149
3150 004434 201040 000004 DFERR10: MOVEI TAC,RENFAL
3151 004435 254000 004373' JRST DFERRY
3152
3153 004436 201040 000005 DFERR11: MOVEI TAC,NOFILE
3154 004437 254000 004373' JRST DFERRY
3155
3156 004440 201040 000000 DFERR12: MOVEI TAC,0
3157 004441 261140 000001 PUSH PDP,TAC ;ADD SOMETHING TO BE REMOVED (SINCE SETLE NOT CALLED)
3158 004442 254000 004373' JRST DFERRY

```

```

3159 004443 205000 000400 WERA:  MOVSI IOS,NORELB
3160 004444 413006 000002      ANDCAB IOS,DEVIOS(DEVDAT)
3161                                ERRORS <NON-RECOVERABLE DISC WRITE ERROR>+
3162 004445 260140 003654'      PUSHJ PDP,CLRBUF          ;CLEAR ANY DUMP BUFFER
3163 004446 265040 004423'      JSP TAC,ERRPTU          ;OUTPUT A MESSAG
3164 004447 472371 626644      ASCII? *NON-RECOVERABLE DISC WRITE ERROR+
      004450 426071 753212
      004451 512030 246212
      004452 202111 151606
      004453 202572 244650
      004454 425010 551244
      004455 476440 000000
3165 004456 254000 004433'      JRST EXCALP              ;TYPE "UUO FROM" MESSAGE AND QUIT
3166
3167
3168 004457 205000 000400 RERA:  MOVSI IOS,NORELB
3169 004460 413006 000002      ANDCAB IOS,DEVIOS(DEVDAT)
3170                                ERRORS <NON-RECOVERABLE DISC READ ERROR>+
3171 004461 260140 003654'      PUSHJ PDP,CLRBUF          ;CLEAR ANY DUMP BUFFER
      004462 265040 004446'      JSP TAC,ERRPTU          ;OUTPUT A MESSAG
3172 004463 472371 626644      ASCII? *NON-RECOVERABLE DISC READ ERROR+
3173 004464 426071 753212
      004465 512030 246212
      004466 202111 151606
      004467 202450 540610
      004470 202132 251236
      004471 510000 000000
3174 004472 254000 004456'      JRST EXCALP              ;TYPE "UUO FROM" MESSAGE AND QUIT

```

```

3175                                     ;CALLED AT BOTH INTERRUPT AND UDO LEVELS
3176                                     ;COMPUTE CHECK-SUM.
3177                                     ;ENTER WITH C(TAC 18-35) SET TO BUFFER ADDRESS
3178                                     ;RETURN WITH C(TAC1)SET TO ONES COMPLEMENT SUM IN RH.
3179
3180                                     IFG CHKCNT,<INTERNAL CHKSUM
3181 004473 554100 000001 CHKSUM: HLRZ TAC1,TAC ;IS COUNT GREATER THAN CHKCNT?
3182 004474 305100 777600 CAIGE TAC1,-CHKCNT
3183 004475 201100 777600 MOVEI TAC1,-CHKCNT ;YES, USE CHKCNT
3184 004476 200000 004617' MOVE IOS,EXWD UBFS,IOACT]
3185 004477 420006 000002 ANDCM IOS,DEVIOS(DEV DAT)
3186 004500 436006 000002 IORM IOS,DEVIOS(DEV DAT)
3187 004501 603000 000200 TLNE IOS,UBFS
3188 004502 271047 000000 ADDI TAC,(PROG) ;NO, RELOCATE,
3189 004503 504040 000002 HRL TAC,TAC1 ;RESET COUNT IN LH AFTER POSSIBLE OVERFLOW
3190 004504 260140 004521' PUSHJ PDP,CHECK
3191 004505 413006 000002 ANDCAB IOS,DEVIOS(DEV DAT)
3192 004506 620000 010000 TRZ IOS,IOACT
3193 004507 263140 000000 POPJ PDP,
3194
3195 004510 344100 004525' CHKSM3: AOJA TAC1,CHKSM2 ;BRING CARRY AROUND
3196
3197                                     ;CHECKSUM ERROR COUNTING ROUTINE
3198 004511 261140 000001 CKREC2: PUSH PDP,TAC
3199 004512 436746 000002 IORM AC3,DEVIOS(DEV DAT)
3200 004513 201040 010000 MOVEI TAC,10000
3201 004514 272040 000011' ADDM TAC,CKSMCT
3202 004515 254000 000000 JRST TPOPJ
3203 004516 435000 100000 CKREC3: IORI IOS,IODTER
3204 004517 350000 000011' AOS CKSMCT
3205 004520 263140 000000 POPJ PDP,
3206                                     >
3207
3208                                     ;TIMING OF MAIN LOOP IS 13.8 MICRO-SECS AVERAGE.
3209                                     ;THIS IS 1.8 MILS OR SO FOR 128 WORDS,
3210
3211
3212 IFN <IFG CHKCNT, <CHKCNT+>>FTCHECK+FTMONP+FTSWAP, <
3213
3214                                     INTERNAL CHECK
3215 004521 201100 000000 CHECK: MOVEI TAC1,0
3216 004522 255200 004523' JFCL 4,.*1 ;CLEAR SUM
3217                                     ;CLEAR CARRY0 FLAG
3218 004523 270101 000000 CHKSM1: ADD TAC1,(TAC) ;ADD NEXT WORD
3219 004524 255200 004510' JFCL 4,CHKSM3 ;JUMP IF CARRY0
3220 004525 253040 004523' CHKSM2: AORJN TAC,CHKSM1 ;LOOP UNTIL DONE
3221 004526 554040 000002 HLRZ TAC,TAC1 ;ADD HALVES
3222 004527 553000 000002 HRRZS TAC1
3223 004530 270100 000001 ADD TAC1,TAC
3224 004531 623100 000001 TLZE TAC1,1
3225 004532 271100 000001 ADDI TAC1,1 ;CARRY INTO LH?
3226 004533 263140 000000 POPJ PDP, ;YES, BRING IT AROUND
3227                                     >

```

```

3228          ;UPDATE DEVCNT IF NECESSARY, DEVCNT HAS SIZE OF FILE IN R.H. (IN WORDS IF POSSIBLE,
3229          ; OTHERWISE NEGATIVE BLOCK COUNT),
3230          ; ENTER WITH SIZE OF CURRENT BLOCK IN TAC1
3231 004534 550046 000022 UPDVC1: HRRZ TAC,SETCNT(DEVDAT) ;*CONVERT CURRENT RELATIVE BLOCK NUMBER TO WORDS
3232 004535 275040 000001 SUBI TAC,1 ;*
3233 004536 242040 000007 LSH TAC,BLKP2 ;*
3234 004537 270040 000002 ADD TAC,TAC1 ;*ADD IN THE ADDITIONAL WORDS IN THIS NEXT
3235          ; OUTPUT COMMAND.
3236 004540 550106 000015 HRRZ TAC1,DEVCNT(DEVDAT) ;*PICK UP PREVIOUS MAXIMUM FILE SIZE OF COMPARISON,
3237 004541 602100 400000 TRNE TAC1,400000 ;*PREVIOUS MAXIMUM SIZE GREATER THAN 2 EXP 17 WORDS?
3238 004542 254000 004552' JRST UPDVC2 ;*YES, THEN COMPARE AND SAVE NEGATIVE BLOCK COUNTS,
3239          ;*NO, DOES THE NEWLY COMPUTED CURRENT FILE SIZE EXCEED
3240 004543 607040 777777 TLNN TAC,-1 ;*
3241          ;* 2EXP17 WORDS?
3242 004544 602040 400000 TRNE TAC,400000 ;*
3243 004545 254000 004551' JRST UPDVC1 ;*YES, GO SAVE NEGATIVE BLOCK COUNT,
3244 004546 313040 000002 CAMLE TAC,TAC1 ;*NO, COMPARE POSITIVE WORD COUNTS
3245 004547 542046 000015 HRRM TAC,DEVCNT(DEVDAT) ;*UPDATE MAXIMUM FILE SIZE ONLY IF PREVIOUS
3246          ; MAXIMUM EXCEEDED

3247 004550 263140 000000 POPJ PDP, ;*
3248 004551 660100 777777 UPDVC1: TRO TAC1,-1 ;*SET PREVIOUS MAXIMUM AS ONLY ONE BLOCK
3249          ;* (THUS FORCING CURRENT BLOCK COUNT TO BE STORED),
3250 004552 271040 000177 UPDVC2: ADDI TAC,BLKSIZE-1 ;*CONVERT POSITIVE WORD COUNT TO NEGATIVE
3251          ; BLOCK COUNT,
3252 004553 242040 777771 LSH TAC,-BLKP2 ;*
3253 004554 213000 000001 MOVNS TAC ;*
3254 004555 666040 400000 TRON TAC,400000 ;*MORE THAN 2EXP17 BLOCKS (16,777,216 WORDS)
3255          ; IN THIS FILE?
3256 004556 664000 040000 TROA IOS,IOBKTL ;*YES, (PREPOSTEROUS) SET ERROR BIT AND STORE
3257          ; BLOCK COUNT MODULO 2EXP17
3258 004557 303101 000000 CAILE TAC1,(TAC) ;*NO, COMPARE PREVIOUS MAXIMUM VERSUS PRESENT
3259          ; FILE SIZE,
3260 004560 542046 000015 HRRM TAC,DEVCNT(DEVDAT) ;*STORE NEW MAXIMUM FILE SIZE ONLY IF PREVIOUS
3261          ; MAXIMUM EXCEEDED,
3262 004561 263140 000000 POPJ PDP, ;*

```

```

3263                                     ;OUTPUT WRITE-LOCK ERROR, FREE THE BLOCK AND GET NEW ONE.
3264
3265
3266 004562 554046 000021 WLER: HLRZ TAC,DSKCN(DEVDA) ;PICK UP BLOCK NUMBER.
3267 004563 260140 003464' PUSHJ PDP,SETFRE ;FREE THE BLOCK.
3268 004564 254000 003371' JRST DFGETF ;GET ANOTHER.
3269
3270                                     ;SET WRITE-LOCK INDICATION IN A SAT ENTRY.
3271                                     ;ENTER WITH A BLOCK NUMBER IN TAC.
3272
3273
3274 004565 201100 000022' SETWL: MOVEI TAC1,SATENT ;*
3275 004566 254000 004570' JRST ,+2 ;*
3276 004567 271100 000003 SETWL1: ADDI TAC1,SENTSZ ;*
3277 004570 554220 000002 HLRZ ITEM,@TAC1 ;*IS THE BLOCK IN THIS SAT ENTRY?
3278 004571 311040 000004 CAML TAC,ITEM ;*
3279 004572 301044 007640 CAIL TAC,NUMBLK(ITEM) ;*
3280 004573 254000 004567' JRST SETWL1 ;*NO, LOOP.
3281 004574 201200 400000 MOVEI ITEM,WLBIT ;*
3282 004575 436220 000002 ORM ITEM,@TAC1 ;*
3283 004576 263140 000000 POPJ PDP, ;*

```

DSKSPB - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (BURROUGHS)
ERRORS. SUBROUTINES.

MACRO,V36 19:05 4-JUN-69 PAGE 96

```
3284 004577          DSKSR:  END
3285 004577 141300 000002
3286 004600 270400 000002
3287 004601 270405 777776
3288 004602 000406 000002
3289 004603 331120 000014
3290 004604 331120 000002
3291 004605 000007 000001
3292 004606 002120 000002
3293 004607 000200 000200
3294 004610 000002 000002
3295 004611 000003 000003
3296 004612 000040 020000
3297 004613 777600 000002
3298 004614 000140 020000
3299 004615 000000 000000
3300 004616 777777 777777
3301 004617 000200 010000
```

NO ERRORS DETECTED

PROGRAM BREAK IS 004620

AC1	000015	INT	AC2	000016	INT	AC3	000017	INT
ACCIN1	000511		ACCIN2	000515		ACCIN1	000527	INT
ADRERR	002440	EXT	ADVBE	003240	EXT	ADVBEF	003065	EXT
ALLOK	001044		ALLXIT	001061		ALTMFD	001071	
ASSCON	400000	INT	ASSPRG	000000	INT	ATBITS	500000	
ATBLOK	000003		ATCLO	000000		ATEXT	000002	
ATIND	040000		ATLINK	000003		ATNAME	000001	
ATPP	000000		AUAV4L	003664	EXT	AUFLG	010000	
AUREQ	003663	EXT	AUWAIT	001757	EXT	BLKP2	000007	
BLKSIZ	000200		BUFCLE	003706	EXT	RUFPT	000012	INT
BUFWPD	000013	INT	CHECK	004521	INT	CHKCNT	000200	INT
CHKSM1	004523		CHKSM2	004525		CHKSM3	004510	
CHKSUM	004473	INT	CKREC2	004511		CKREC3	004516	
CKSMCT	000011	INT	CKSUPR	010000		CLCOR1	004026	INT
CLDDR1	003556		CLRAT	002160		CLRAT1	002170	
CLRB2	004063		CLRB3	004064		CLRB4	004044	
CLRB5	004051		CLRBF1	003676		CLRBIT	004032	INT
CLRBUF	003654		CLRCOR	004025		CLRDDB	003555	INT
CLSI	000002	INT	COMCHK	002437	EXT	CORBIT	000263	
CORIWD	000264		CPOPJ	003713	EXT	CPOPJ1	003720	EXT
CPOPJ2	001410	EXT	CRINIT	000265	INT	DAVAL	003660	EXT
DAFLG	004000		DAREQ	003657	EXT	DAT	000005	INT
DAKAIT	003415	EXT	DCBBIT	000570	EXT	DBPTR	000002	
DBTAB	003733	EXT	DEVACC	000014	INT	DEVBK0	000017	INT
DEVBLK	000016	INT	DEVBUF	000006	INT	DEVCT	000015	INT
DEVDAT	000006	INT	DEVEXT	000012	INT	DEVFIL	000011	INT
DEVIAD	000007	INT	DEVLOS	000002	INT	DEVMOD	000004	INT
DEVNAM	000000	INT	DEVOAD	000010	INT	DEVPPN	000013	INT
DEVSER	000003	INT	DEBUSY	000007	INT	DFC16A	001556	
DFCL16	001551		DFCL17	001557		DFCL2	001506	
DFCL20	001562		DFCL21	001603		DFCL23	001617	
DFCLSI	001621		DFCLSO	001453		DFCLU1	001567	
DFCLU2	001577		DFDI0	002441		DFDI0A	002447	
DFDI1	002467		DFDI10	002603		DFDI1A	002471	
DFDI1B	002500		DFDI1C	002501		DFDI2	002504	
DFDI3	002514		DFDI4	002525		DFDI5	002531	
DFDI6	002533		DFDI7	002557		DFDI8	002571	
DFDI9	002574		DFDMP1	002431		DFDMP0	002215	
DFD00	002223		DFD00A	002231		DFD01	002233	
DFD01A	002234		DFD01B	002243		DFD02A	002241	
DFD02B	002336		DFD02C	002303		DFD02D	002312	
DFD02E	002316		DFD02G	002327		DFD03A	002340	
DFD03B	002350		DFD03C	002362		DFD04	002373	
DFD06	002402		DFD07	002407		DFD08	002414	
DFD09	002422		DFEN2A	000656		DFENT1	000640	
DFENT2	000651		DFENT3	000664		DFENT4	000720	
DFENT5	000736		DFENT7	000763		DFENTR	000625	
DFER10	004434		DFER11	004436		DFER12	004440	
DFERR1	004342		DFERR2	004355		DFERR3	004360	
DFERR4	004402		DFERR5	004400		DFERR6	004372	
DFERR7	004404		DFERR8	004406		DFERR9	004421	
DFERRY	004373		DFGETF	003371	INT	DFGT1	003372	
DFGT2	003411		DFIN	002757		DFIN1	002762	

DFIN1A	002775'	DFIN1B	003002'	DFIN1D	002773'
DFIN1E	002777'	DFIN2	003005'	DFIN4	003016'
DFIN5	003030'	DFIN6A	003044'	DFIN7	003053'
DFIN8	003062'	DFIN8A	003063'	DFINT	004142' INT
DFINT1	004207'	DFINT2	004215'	DFINT3	004224'
DFINT4	004236'	DFINT5	004251'	DFINT6	004263'
DFINT7	004266'	DFINT9	004322'	DFINTA	004155'
DFINX	003065'	DFLOOK	000766'	DFLUK2	001001'
DFLUK3	001004'	DFLUK4	001025'	DFLUK7	001064'
DFLUK8	001066'	DF04A	003131'	DF04R	003140'
DF06A	003162'	DF07A	003172'	DF08A	003231'
DFOUT3A	003124'	DFOUT	003073'	DFOUT1	003077'
DFOUT5	003144'	DFOUT6	003154'	DFOUT7	003200'
DFOUT8	003215'	DFOUT9	003233'	DFOUTX	003240'
DFRED	004321' EXT	DFREL	003355'	DFREL1	003367'
DFREN	001103'	DFREN1	001124'	DFREN2	001151'
DFREN3	001213'	DFREN5	001242'	DFREN7	001315'
DFREN8	001310'	DFREN9	001312'	DFRENX	001106'
DFREXT	001207'	DFRN1A	001154'	DFRN1B	001162'
DFRN2A	001144'	DFRN8A	001306'	DFSET	002607'
DFSET1	002645'	DFSET3	002640'	DFSET5	002670'
DFSET6	002676'	DFSET7	002737'	DFSET8	002747'
DFSET8	002707'	DFSETC	002711'	DFSETD	002723'
DFSETX	002660'	DFWRT	004320' EXT	DFWUNS	000565' EXT
DINT08	004205'	DINT1A	004212'	DINT4A	004237'
DINT4B	004241' INT	DIRSZ	000004' INT	DIRSRC	001634'
DISKUP	000562' EXT	DOWNDA	003404'	OR	000016' INT
DRSRC3A	001655'	DRSRC0	001637'	DRSRC4	001662'
DRSRC6	001701'	DRSRC7	001667'	DRSRC8	001673'
DSKBIT	000567' EXT	DSKBUF	000020' INT	DSKCNT	000021' INT
DSKCON	000526' EXT	DSKCOR	000010'	DSKDDB	000237' INT
DSKDSP	000470' INT	DSKFGS	016200' INT	DSKFUL	003447'
DSKIN0	000534'	DSKIN1	000553'	DSKINI	000524' INT
DSKOFF	004120' EXT	DSKON	004421' EXT	DSKRLB	000020' INT
DSKSIZ	000040'	DSKSR	004577'	DSKSRB	000000' INT
DSKSTP	000567' INT	DSKXDB	037200'	DSRC10	001704'
DTRIT	200000'	DUMPPP	001450' EXT	DVDIR	000004' INT
DVDSK	200000' INT	DVIN	000002' INT	DVLNG	001000' INT
DVOUT	000001' INT	ECOUNT	000014'	ENTRR	020000' INT
ERRBIT	000001'	ERRPTU	004462' EXT	EXCALP	004472' EXT
FAT	000010'	FINDE	001720'	FINDE1	001726'
FINDE2	001752'	FINDV	004247'	FPNTR	000027' INT
FPNTR1	000026' INT	FTCCL	777777 777777	FTCHEC	000000' INT
FTDISK	777777 777777	FTLOGI	777777 777777	FTMONP	000000' INT
FTRA10	000000' INT	FTRC10	777777 777777 INT	FTRCWK	777777 777777 INT
FTSWAP	777777 777777 INT	GETBIT	003744' INT	GETFCR	003731' INT
GETPTR	003012'	GETZ1	003711'	GETZ2	003713'
GETZ3	003716'	GETZER	003707'	GTBIT1	003756'
GTBIT2	003776'	GTBIT3	004001'	GTBIT4	004006'
GTBIT5	004016'	GTBIT6	004021'	GTBITA	003765'
GTCOR1	003732'	GTPTR1	003022'	HNGSTP	003452' EXT
HUNGST	000001' INT	IADRC	003702' EXT	ICLOS8	002000' INT
INRFR	000400' INT	INPB	010000' INT	INSD1	001765'

INSD10	002073'	INSD11	002100'	INSD2	001772'
INSD3	002074'	INSD3A	002013'	INSD3B	002005'
INSD5	002023'	INSD5A	002043'	INSD6	002046'
INSD7	002053'	INSD7A	002060'	INSDR	002070'
INSDIR	001756'	IO	000020' INT	IOACT	010000' INT
IOPKTL	040020' INT	IOEND	020000' INT	IOERR	200000' INT
IONTER	100000' INT	IOEND	000040' INT	IOIMPM	400000' INT
IOS	000020' INT	IOW	000041' INT	IOWC	000020' INT
ITEM	000024' INT	JACCT	100000' INT	JBTADR	004273' FXT
JBTSTS	003443' EXT	JDAT	000007' INT	JIFMIN	000715' FXT
JOB	001435' EXT	JORFF	003700' FXT	JOBN	003435' FXT
JSAT06	000624' INT	LBHIGH	003466' EXT	LIR	000040'
LOCORE	000262' INT	LOOKR	040000' INT	MFOBLK	000000' INT
MIPTR	000024'	MJOBN	000000' EXT	MONBUF	000266'
MONGO	004327'	MOPTR	000003'	MQAVAL	003670' FXT
MOIN	004110' INT	MOOUT	004112' INT	MQREQ	003667' FXT
MOTOP	004233' EXT	MQUEUE	004234' EXT	MQWAIT	003650' FXT
NBLKSZ	777777	NCTRLC	002000'	NEWRI8	003202'
NMP	000100'	NOFILE	000005'	NORELB	000400'
NORITE	000003'	NOTINM	000001'	NOTINU	000000'
NUMBIT	011000' INT	NUMBLK	007640' INT	NUMSAT	000004' INT
OCLOSE	001000' INT	OUT	001504' EXT	OUTBFB	000200' INT
PDP	000003' INT	PIOMOD	003622' EXT	PJOBN	004272' FXT
PNTDIF	200000'	PRCHG	020000'	PRJPRG	001447' FXT
PROG	000007' INT	PROT1	001444'	PROTEK	001423'
PROTF	000002'	PROTKX	001434'	PROTKY	001435'
PTR1	000023' INT	PTRN	000037' INT	QIN	004066'
QIOEND	004106'	QOUT	004067'	RCOUNT	007777'
RECLAM	003327'	RECLM2	003332'	RECLM4	003342'
RECLM5	003344'	RECLM6	003353'	REFLAG	000012' INT
RENBIT	020000'	RENFAL	000004'	RERA	004457'
RESID1	003725'	RESIDU	003721'	RRI	003262'
RRIA	003263'	RRI8	003261'	RT8IT	100000'
RUNUSR	000005'	RWTBIT	010000'	S177A	003306'
SAT	000036' INT	SATBK2	000236' INT	SATBLK	003546'
SATCHG	000017'	SATCNT	003457'	SATENT	000022' INT
SATGET	003421'	SATGT1	003431'	SATGT2	003455'
SATM2	000034' INT	SAT01	000577'	SAT02	000606'
SAT04	000615'	SAT05	000605' INT	SAT06	000620'
SATPIK	000016'	SATPTR	000015' INT	SATRD	003520'
SATTEM	000020'	SATTMP	000021'	SATTOP	000033' INT
SATW1	003543'	SATWRT	003536'	SATXWD	000001' INT
SAVDDL	002355' EXT	SCANAT	002143'	SCNAT0	002144'
SCNAT1	002146'	SCANAT2	002157'	SCNHLT	002145'
SD	000015' INT	SENTSZ	000003' INT	SERA	004142' FXT
SET000	003314'	SET176	003311'	SET177	003304'
SETACT	004117' EXT	SETAT	002221'	SETBF1	003632'
SETBF2	003635'	SETBF5	003643'	SETBF6	003647'
SETBFI	003630'	SETBFO	003634'	SETBUF	003620'
SETCNT	000022' INT	SETOD0	003604'	SETD01	003615'
SETDDB	003574' INT	SETDUN	004335'	SETFR1	003477'
SETFR2	003510'	SETFRE	003464' INT	SETHLT	002214'
SETI00	004340' EXT	SETLE	001340'	SETLE0	001342'

DSKSRB - ALMOST DEVICE INDEPENDENT DISK SERVICE ROUTINES (BURROUGHS)
SYMBOL TABLE

MACRO,V36 19:05 4-JUN-69 PAGE 97-3

SETLE9	001352'	SETPTR	003316'	SETSAT	000563' INT
SETWL	004565'	SETWL1	004567'	SQGO	004240' EXT
SQREQ	004237' EXT	SRCU1	001742'	STARTS	000566'
STOINS	000000' EXT	SWPINT	004144' EXT	SYSDEV	000100' INT
SYSPP	001356' EXT	TA	000012	TAC	000001' INT
TAC1	000002' INT	TB	000013	TBITS	000002
THSDAT	001630' EXT	TIME	000714' EXT	TPOPJ	004515' EXT
TRIES	000003	UADCK	003641' EXT	UBFS	000200
UBFU	001000	UDLKC	000772' EXT	UPDA	003414'
UPDEV	004534'	UPDVC1	004551'	UPDVC2	004552'
USRCNT	000006'	USRC0	004267'	USRC03	004305'
UUC	000014' INT	UXFERS	000013'	VDSKSR	000424' INT
VRGPTR	100000	WBBIT	400000	WAIT1	003370' EXT
WERA	004443'	WLBIT	400000' INT	WLER	004562'
WPRO	040000	WRI	003274'	WRIA	003275'
WRIB	003273'	WSYNC	004135' EXT	WTBIT	400000
WUFD	002105'	WUFD1	002114'	WUFD3	002130'

A	AC1	6#	6	718	720	721	723	748	749	766	771	912	913	915	916
		6#	6	975	976	979	980	984	986	987	991	997	998	1031	1032
		931	943												
		1034	1036	1038	1047	1048	1049	1053	1054	1084	1085	1142	1143	1145	1146
		1159	1176	1195	1205	1209	1215	1243	1250	1293	1294	1482	1483	1494	1547
		1586	1588	1590	1591	1592	1593	1594	1595	1598	1626	1628	1956	1957	1959
		1971	2049	2057	2069	2070	2155	2156	2159	2238	2242	2253	2254	2361	2397
		2401	2403	2409	2413	2436	2437	2454	2455	2456	2462	2463	2464	2469	2470
		2471	2529	2530	2533	2593	2595	2696	2718	2719	2722	2723	2780	2814	2816
		2819	2821	2822	2823	2883	2884	2885	2886	2888	2889	2890	2896	2897	
AC2		6#	6	876	877	879	881	884	885	951	978	980	1080	1443	1448
		1479	1480	1482	1483	1485	1490	1493	1494	1548	1592	1594	1595	1596	1624
		1626	1627	1631	1632	1636	1644	1647	1654	1665	1694	1695	1697	1727	1728
		1730	1732	1769	1770	1772	1774	1776	1781	1782	1783	1789	1793	1795	1798
		1812	1821	1826	1829	1835	1849	1852	1853	1854	1958	1959	1960	1962	1963
		2034	2035	2039	2040	2054	2057	2058	2075	2077	2079	2152	2153	2157	2159
		2160	2165	2166	2167	2168	2239	2241	2362	2472	2697	2760	2781	2797	
		6#	948	949	1177	1180	1184	1185	1194	1195	1456	1457	1783	1785	1830
		1833	1834	2045	2046	2050	2053	2055	2060	2061	2062	2067	2068	2149	2155
		2161	2162	2182	2188	2193	2197	2214	2296	2297	3199				
ACCIN1	356#	359													
ACCIN2	357	360#													
ACCINI	352	354#													
ADRERR	1519	1527	1702	1711											
ADVBFE	2095	2226	2909												
ADVBFF	1994	2085	2909												
AEFERR	6#	6													
AL	6#	6													
ALLOK	544	640#													
ALLXIT	652	658#													
ALTMFD	467	675#	783												
ASSCON	6#	6	401												
ASSPRG	6#	6	2547												
ATBITS	116#	741													
ATBLOK	110#														
ATCLO	118#	485	487	1016	1123										
ATFXT	108#	1455	1505												
ATIND	117#	485	1067												
ATLINK	111#	1463	1480	1											

CHECK	3190	3214	3215#											
CHKCNT	54	71#	647	1334	1394	1399	1404	1640	1813	1820	2104	2126	2128	2946
	3180	3182	3183	3212										
CHKSM1	3218#	3220												
CHKSM2	3195	3220#												
CHKSM3	3195#	3219												
CHKSUM	1336	1396	1643	1828	2107	2952	3180	3181#						
CKREC2	1832	3198#												
CKREC3	2957	3203#												
CKSMCT	163	171#	3201	3204										
CKSUPR	100#	654	2948											
CLCOR1	2537	2775	2778#											
CLDDR1	2522#	2526												
CLKR	6#	6												
CLRAT	396	1079	1089	1101	1110	1134	1473#							
CLRAT1	1482#	1491												
CLRB2	2817	2818	2822#											
CLRB3	2820	2823#												
CLRB4	2806#	2809	2810											
CLRB5	2807	2812#												
CLRBF1	2628	2634#												
CLRBIT	2474	2792	2794#											
CLRBUF	660	777	1087	1102	1112	1138	1144	1148	1605	1664	1848	1926	1978	1989
	2312	2332	2612#	3103	3113	3125	3137	3145	3162	3171				
CLRCOR	1495	2777#												
CLRDOB	402	2519	2521#											
CLSIN	6#	6	588											
CLSOUT	6#	6												
CMWB	6#	6												
COMCHK	1519	1526	1702	1710										
CORBIT	307#	363												
CORCNT	6#	6												
CORIWD	308#	365												
CPOPJ	1119	1124	1441	1464	2524	2661	2830							
CPOPJ1	331	462	689	693	905	960	972	996	1441	1458	2667			
CPOPJ2	905	955												
CRINIT	302	309#	364	2781										
D	6#	6												
DAAV4L	2335	2356	2373	2617										
DAFLG	82#	84	2374	2388	2614									
DAREQ	2228	2335	2356	2370	2386	2616								
DAT	6#	6	481	493	494	512	515	516	518	523	524	533	536	537
	598	611	617	640	641	650	707	725	729	740	741	808	919	920
	922	923	924	926	927	930	956	957	959	990	992	994	995	1062
	1063	1067												

DCLI	6#	6												
DCLR	6#	6												
DDPRT	6#	6												
DDPRT	158#	396	2531	2533	3008	3040								
DDPRT	303	329	2697											
DDI	6#	6												
DDO	6#	6												
DDN	6#	6												
DEVACC	273#	273	394	514	550	603	641	712	747	787	815	891	1009	1061
	1075	1098	1111	1118	1162	1167	1269	1271	1378	1398	1433	1478	1624	1654
	1727	1736	1769	1794	1798	1849	1853	1871	1924	2004	2030	2040	2088	2102
	2171	2230	2297	2931	2953	2967	3041							
DEVADR	6#	6												
DEVAKO	284#	284	508	2042	2142	2176								
DEVBLK	280#	280	507	818	1047	1232	1237	1901	1965	1974	1985	2041	2045	2068
	2074	2143	2152	2166	2175	2193	2194	2217	2248	2261	2289			
DEVBJF	6#	6	1023											
DEVCHR	6#	6												
DEVCMT	276#	276	541	636	791	812	942	1053	1168	1200	1229	1266	1273	1286
	1330	1881	1913	3236	3245	3260								
DEVCTR	6#	6												
DEVJAT	6#	6	355	356	357	358	361	388	389	390	392	394	398	399
	400	433	434	446	454	468	479	488	507	508	509	514	515	539
	540	541	543	547	550	561	565	589	603	606	616	621	623	631
	636	637	639	640	641	643	645	653	656	695	699	712	716	719
	747	753	757	767	787	788	790	791	792	795	810	811	812	815
	816	818	827	876	891	908	910	913	916	921	924	941	942	943
	949	961	974	976	978	1007	1009	1021	1023	1045	1047	1048	1053	1060
	1061	1065	1069	1071	1073	1075	1094	1098	1104	1111	1116	1118	1137	1162
	1163	1167	1168	1172	1200	1214	1219	1228	1229	1232	1237	1239	1243	1247
	1266	1267	1269	1271	1273	1274	1286	1293	1303	1306	1330	1332	1335	1337
	1338	1358	1364	1378	1386	1395	1398	1402	1414	1416	1422	1425	1429	1431
	1433	1451	1456	1478	1504	1506	1510	1521	1525	1562	1567	1568	1571	1572
	1578	1580	1606	1608	1612	1615	1624	1631	1634	1638	1647	1654	1665	1677
	1681	1704	1709	1727	1728	1736	1737	1742	1764	1769	1770	1782	1794	1797
	1798	1805	1841	1849	1852	1853	1863	1869	1871	1872	1876	1881	1886	1887
	1891	1896	1901	1904	1913	1919	1922	1924	1925	1930	1937	1940	1943	1945
	1956	1963												

NAME	6#	6	398	2584	2833	2839	2848	2941	2950	3016	3032	3047	3076	3078
DEV10S	6#	6	434	446	454	539	653	910	1045	1069	1364	1416	1422	1429
	1525	1634	1709	1764	1841	1869	1891	1896	1919	1987	2015	2019	2027	2044
	2065	2130	2117	2291	2375	2389	2503	2600	2629	2634	2958	2992	3085	3109
	3160	3169	3185	3186	3191	3199								
DEVLOG	6#	6												
DEVMOD	6#	6	400	2546										
DEVNAM	6#	6	358	392										
DEVODD	6#	6	399	543	547	616	621	639	645	656	699	795	908	961
	1402	1414	1431	1638	1887	1937	2034	2070	2105	2110	2118	2145	2168	2178
	2589	2833	2947	3018	3025	3050	3076							
DEVPPN	6#	6	270#	270	924									
DEVPTB	6#	6												
DEVSER	6#	6	356	361	389	2523	2529	2530	2562	3020	3039			
DEVUSY	163	169#	383	2845	2889	3004								
DFC16A	1066	1089#	1109	1133										
DFCL16	1084#	1088												
DFCL17	1082	1092#												
DFCL2	1022	1027	1039	1043#										
DFCL20	1068	1097#												
DFCL21	662	778	804	857	864	894	1118#							
DFCL23	1130	1135#												
DFCLSI	344	1116#	2338											
DFCLSO	333	1007#	2339											
DFCLU1	1064	1103#												
DFCLU2	1107	1111#												
DFDI0	1712#	1717	1760											
DFDI0A	1715	1719#												
DFDI1	1738#	1838												
DFDI10	1845	1852#												
DFDI11A	1741#	1842												
DFDI1B	1731	1762#												
DFDI1C	1707	1763#												
DFDI2	1739	1769#												
DFDI3	1778#	1786												
DFDI4	1784	1788#												
DFDI5	1780	1792#												
DFDI6	1791	1794#												
DFDI7	1827#	1835												
DFDI8	1773	1840#												
DFDI9	1729	1771	1844#											
DFDMP1	339	1704#												
DFDMP0	338	1521#												
DFD07	1529#	1534	1658											
DFD07A	1532	1536#												

DFD02G	1574	1612#							
DFD03A	1626#	1652							
DFD03B	1635#	1699							
DFD03C	1650#	1667							
DFD04	1648	1661#							
DFD06	1550	1669#							
DFD07	1619	1677#							
DFD08	1684#	1687							
DFD09	1629	1693#							
DFEN2A	493#	498							
DFENT1	477#	579							
DFENT2	473	487#							
DFENT3	492	497	500#						
DFENT4	526	531#							
DFENT5	469	547#							
DFENT7	475	578#							
DFENTR	336	465#							
DFER10	805	3150#							
DFER11	696	3153#							
DFER12	466	587	3156#						
DFERR1	2424	3102#							
DFERR2	1522	1705	1865	1998	2098	3108#			
DFERR3	483	613	731	3112#					
DFERR4	472	593	702	797	3130#				
DFERR5	484	501	549	663	686	858	3127#		
DFERR6	474	495	555	580	704	865	965	3120#	
DFERR7	594	703	3133#						
DFERR8	562	568	3136#						
DFERR9	2700	3142#							
DFERRY	3121#	3128	3131	3134	3151	3154	3158		
DFGETF	505	1331	2123	2192	2357	2359#	3268		
DFGTF1	2360#	2380							
DFGTF2	2365	2378#							
DFIN	335	1997#							
DFIN1	2001#	2092							
DFIN1A	2015#	2025							
DFIN1B	2011	2023#							
DFIN1D	2013#	2031							
DFIN1E	2008	2018#							
DFIN2	2006	2027#							
DFIN4	1238	1977	2039#	2174	2295				
DFIN5	2050#	2058							
DFIN6A	2065#	2081							
DFIN7	2051	2072#							
DFIN8	2073	2079#							
DFIN8A	2056	2080#							
DFINT	2908	2915#							
DFINT1	2939	2963#							
DFINT2	2923	2971#							
DFINT3	2974	2979#							
DFINT4	2965	2992#							
DFINT5	3011#	3021							
DFINT6	3019	3025#							

[illegible]

DFINT7	3017	3032#					
DFINT9	2930	3070#					
DFINTA	2931#	3073					
DFINX	2085#	2961					
DFLOOK	337	586#					
DFLUK2	598#	668					
DFLUK3	596	602#	669				
DFLUK4	571	623#					
DFLUK7	610	614	662#				
DFLUK8	595	667#					
DF04A	1046	1418	1663	1892	1939	2036	2142#
DF04B	2151#	2186					
DF06A	1988	2170#					
DF07A	2147	2178#					
DF08A	2214#	2221					
DF0T3A	2113	2133#					
DFOUT	334	2097#					
DFOUT1	2102#	2240					
DFOUT5	2155#	2162					
DFOUT6	2156	2164#					
DFOUT7	2158	2185#					
DFOUT8	2183	2200#					
DFOUT9	2191	2217#					
DFOUTX	2226#	2963					
DFREN	2910	3048	3066				
DFREL	332	2337#					
DFREL1	2342	2348#					
DFREN	343	691#					
DFREN1	705	711#					
DFREN2	735#	856					
DFREN3	754	759	780#				
DFREN5	798	808#					
DFREN7	785	868#					
DFREN8	727	781	857#				
DFREN9	742	862#					
DFRENX	691	695#					
DFREXT	775#	851					
DFRN1A	740#	745					
DFRN1B	738	744	747#				
DFRN2A	724	729#					
DFRN8A	732	855#					
DFSET	340	341	1861#				
DFSET1	1889	1895#					
DFSET3	1879	1911#					
DFSET5	1917	1921#	1933				
DFSET6	1874	1930#					
DFSET7	1961	1970#					
DFSET8	1964	1982#					
DFSETB	1900	1942#					
DFSETC	1905	1945#	1967				
DFSETD	1952	1956#					
DFSETX	1885	1913#					
DFWRT	2910	3051	3065				

	199#	214	218	218#	220	224	224#	226	230	230#	232	236	236#
DSKXDB	199#	214	218	218#	220	224	224#	226	230	230#	232	236	236#
DSO	6#	6											
DSRC10	944	1209#											
DTBIT	114#	892	1084	1108	1132								
DUMPPP	972	998											
DVAVAL	6#	6											
DVCDR	6#	6											
DVDIR	6#	6	260										
DVDIRI	6#	6											
DVOIS	6#	6											
DVDSK	6#	6	260										
DVDTA	6#	6											
DVIN	6#	6	260										
DVLNG	6#	6	260										
DVLPT	6#	6											
DVMTA	6#	6											
DVOUT	6#	6	260										
DVPTP	6#	6											
DVPTR	6#	6											
DVTTY	6#	6											
ECQUANT	174#	3070											
ENTRB	6#	6	589	1007	1060	1521	1863	2097					
ERRBIT	323#	3122											
ERRPTU	3093	3104	3114	3138	3146	3163	3172						
EXCALP	3093	3106	3116	3140	3148	3165	3174						
FAT	170#	405	1444	1479	1511	1513							
FBMERR	6#	6											
FINDE	840	868	1070	1225#									
FINDE1	1232#	1258											
FINDE2	1235	1255#											
FINDV	3003	3008#											
FNFERR	6#	6											
FPNTR	298#	298	299	640	2296								
FPNTR1	299#	299	631										
FRGSEG	6#	6											
FT2REL	6#												
FTATTA	6#												
FTCCL	10#												
FTCHEC	6#	147	148	155									
FTDISK	10#												
FTEXAM	6#												
FTFINI	6#												
FTGETT	6#												
FTHALT	6#												
FTKCT	6#												
FTLOGI	10#												
FTMONP	6#	147	148	155									
FTPRV	6#												
FTRA10	6#	11	14	26	38	42	59	126	183	186	379		
FTRC10	10#	15	18	34	43	47	61	66	127	130	189		
FTRCHK	6#	1010	1015	1122	1307	1473	1484	2264					

FTSWAP	10#	375	2908	2916	2993					
FTTALK	6#									
FTTIME	6#									
FTTRAC	6#									
FTTRPS	6#									
FTTYS	6#									
GETBIT	1549	2364	2699	2706	2716#					
GETFCR	1503	2693	2695#							
GETPTR	1191	1377	1847	2029	2034#					
GETZ1	2658#	2666								
GETZ2	2661#	2663								
GETZ3	2656	2659	2665#							
GETZER	2655#	2720	2738							
GTBIT1	2727#	2733	2743							
GTBIT2	2728	2745#								
GTBIT3	2749#	2752	2753							
GTBIT4	2750	2755#								
GTBIT5	2721	2764#	2771							
GTBIT6	2737	2739	2768#							
GTRT1A	2731	2735#								
GTCOR1	2555	2696#								
GTPT1	2043#	2078								
HNGSTP	2395	2427								
HSAMSK	6#	6								
HSAPOS	6#	6								
HSASIZ	6#	6								
HUNGCT	6#	6								
HUNGST	6#	6	257							
I	6#	6								
IADROK	2611	2638								
IB	6#	6								
IBUFR	6#	6								
ICLOSB	6#	6	637							
ILM	6#									
ILUERR	6#	6								
INRFR	6#	6	2582							
INITP	6#	6								
INPB	6#	6	1137							
INS01	1271#	1284								
INS010	1275	1375#								
INS011	1342	1383#								
INS02	1277#	1379								
INS03	1287	1293#								
INS03A	1295	1303#	1329							
INS03B	1294#	1300								
INS05	1278	1318#	1376							
INS05A	1341#	1387								
INS06	1345#	1350								
INS07	1347	1352#								
INS07A	1358#	1371								
INS08	1354	1369#	1370							
INS01R	1097	1263#								
IO	6#	6	470	591	697	1212	1524	1708	1999	2099

IOACT	6#	6	453	3086	3184	3192										
IOBEG	6#	6														
IORCTL	6#	6	3256													
IOROT	6#	6														
IOCON	6#	6														
IONEND	6#	6	1763	1840	1868	2290										
IONERR	6#	6	2897	2929	2973											
IODTER	6#	6	1830	2897	2929	2973	3203									
IOEND	6#	6	86	1706	1763	1840	1848	1918	1986	2290						
IOFST	6#	6														
IOIMPM	6#	6	1383	1426	1428	1678	2897	2927	2981	3108						
IONRCK	6#	6														
IOPAR	6#	6														
IOS	6#	6	434	470	486	500	532	538	539	591	609	697	722	728		
	733	764	769	780	855	878	910	1035	1044	1045	1069	1174	1189	1212		
	1216	1218	1227	1265	1363	1364	1375	1383	1415	1416	1421	1422	1428	1429		
	1524	1525	1633	1634	1706	1708	1709	1763	1764	1840	1841	1844	1858	1869		
	1890	1891	1893	1896	1918	1919	1986	1987	1999	2001	2014	2015	2018	2019		
	2023	2024	2027	2044	2063	2065	2080	2099	2100	2116	2117	2173	2203	2275		
	2290	2291	2374	2375	2388	2389	2421	2423	2490	2497	2502	2503	2575	2597		
	2600	2606	2614	2618	2622	2626	2627	2629	2634	2879	2927	2929	2958	2973		
	2979	2981	2992	3085	3086	3087	3108	3109	3159	3160	3168	3169	3184	3185		
	3186	3187	3191	3192	3203	3256										
IoTEND	6#	6														
IOUSE	6#	6														
IOW	6#	6	3087													
IOWC	6#	6	1035													
IPPERR	6#	6														
ITEM	6#	6	439	449	534	535	681	682	918	920	985	986	997	1543		
	1544	1545	1546	1552	1560	1563	1571	1620	1652	1661	1666	1669	1670	1693		
	1698	2363	2473	2531	2532	2536	2544	2552	2566	2695	2717	2728	2742	2747		
	2750	2764	2777	2807	3055	3056	3057	3058	3060	3063	3277	3278	3279	3281		
	3282															
JACCT	6#	6	2420													
JBFA DR	6#	6														
JBFC TR	6#	6														
JBFP TR	6#	6														
JBTADR	2911	2936	3038													
JBTSTS	2395	2419														

LBHIGH	2448	2451																	
LIR	86#	2001	2014	2018	2024														
LISTSN	6																		
LOCORE	302	305#	2696	2704	2778	2780													
LOOKS	6#	6	468	1104	1116	1704	1863	1997											
MEDDL	6#	6																	
MEDBLK	60	62#	932																
MIPTR	165#	407	2881	2883	2896														
MJOBH	2224	2235																	
MONBUF	311#	2607																	
MONGO	2891	3000	3055#																
MOPTR	164#	408	2971	2984	2985	2988	2999	3055	3064										
MQAVAL	2625	2910																	
MQIN	1169	1240	1279	1744	1806	2249	2491	2862	2865#										
MQOUT	441	1341	1412	1601	1618	2269	2499	2862	2867#										
MQREQ	2604	2624	2863	2910															
MQTOP	411	2804	2912	2986															
MQUEUE	406	409	410	2863	2885	2912	2907												
MQWAIT	2605	2863																	
NBLKSZ	91#	1723	1724	2106	2598	2951	3043												
NCTRLC	80#	84	1044	2626															
NECERR	6#	6																	
NEWRTB	1356	2185	2188#																
NLEERR	6#	6																	
NMP	76#	538	1044	1189	1375	1415	1633	1844	1890	1893	1986	2023	2063	2080					
	2116	2173	2290																
NOFILE	322#	3153																	
NORELB	78#	2490	2497	2502	2879	3159	3168												
NORITE	319#	3120																	
NOTINM	317#	3130																	
NOTINU	316#	3133																	
NSDERR	6#	6																	
NSFERR	6#	6																	
NSHF	6#	6																	
NSRBIT	6#	6																	
NSWP	6#	6																	
NUMBIT	125#	125	1671	2235	2378	2434	2436												
NUMBLK	191#	191	218	224	230	236	2456	2464	3279										
NUMSAT	63	129#	129	207	239														
NXM	6#																		
OBUFFR	6#	6																	
OCLOSB	6#	6	540																
OUT	1005	1040																	
OUTBFB	6#	6	2587																
OUTPR	6#	6																	
PDP	6#	6	342	345	346	366	396	402	412	429	431	436	440	441					
	445	447	450	456	467	471	480	482	491	496	504	505	506	513					
	549	555	570	571	578	590	592	599	607	612	618	658	659	662					
	667	679	684	691	692	701	708	717	726	730	737	743	773	775					
	776	777	782	783	789	796	801	804	808	814	819	820	837	840					
	850	857	864	868	889	890	926	927	930	933	934	938	944	945					
	950	951	954	958	963	964	977	981	983	984	1000	1001	1008	1040					
	1041	1043	1046	1050	1051	1055	1070	1074	1078	1079	1080	1081	1086	1089					

[illegible]

[illegible]

[illegible]

TAC	6#	6	354	358	360	361	362	363	364	365	391	392	394	395
	397	398	399	400	401	406	407	408	409	411	424	425	426	427
	437	438	453	454	479	493	507	508	509	516	517	518	527	528
	530	535	536	542	543	547	548	550	553	556	560	561	564	565
	567	588	600	602	603	606	623	626	627	628	631	633	635	636
	638	639	642	643	644	645	654	656	658	675	676	677	678	682
	683	698	699	709	711	712	716	719	721	740	747	749	752	753
	756	757	758	776	782	787	788	789	794	795	801	814	815	816
	817	818	822	823	826	827	828	829	830	831	834	835	843	844
	847	848	891	893	907	908	932	939	940	941	947	948	956	961
	962	963	964	974	975	979	995	1009	1011	1014	1016	1019	1020	1023
	1025	1026	1028	1031	1061	1062	1071	1072	1073	1075	1078	1085	1094	1098
	1103	1118	1119	1121	1128	1162	1163	1165	1166	1173	1175	1177	1179	1183
	1184	1194	1197	1214	1228	1232	1239	1245	1247	1249	1250	1256	1267	1271
	1272	1274	1277	1278	1303	1304	1306	1308	1312	1321	1322	1323	1324	1325
	1332	1335	1338	1346	1347	1358	1359	1378	1386	1395	1398	1400	1410	1414
	1417	1434	1444	1448	1452	1455	1463	1465	1474	1490	1493	1529	1531	1533
	1536	1537	1563	1564	1566	1567	1582	1586	1587	1589	1612	1615	1616	1617
	1641	1657	1671	1672	1677	1678	1681	1684	1695	1712	1714	1716	1719	1720
	1722	1724	1725	1734	1735	1743	1775	1778	1779	1788	1789	1790	1792	1799
	1800	1802	1804	1827	1829	1831	1872	1873	1877	1878	1884	1886	1887	1888
	1898	1899	1901	1902	1903	1904	1913	1914	1915	1916	1920	1921	1923	1925
	1931	1932	1937	1938	1940	1942	1943	1945	1949	1957	1970	1971	1973	1974
	1983	1984	2041	2042	2053	2054	2056	2072	2073	2074	2079	2105	2106	2112
	2113	2127	2142	2143	2145	2146	2170	2171	2172	2175	2176	2178	2190	2191
	2194	2196	2197	2205	2206	2208	2209	2210	2217	2218	2233	2234	2235	2247
	2248	2252	2254	2260	2261	2263	2265	2289	2311	2312	2319	2320	2323	2331
	2367	2368	2369	2378	2379	2398	2399	2405	2406	2418	2419	2420	2434	2437
	2450	2451	2455	2456	2463	2464	2466	2468	2470	2487	2488	2509	2510	2511
	2512	2521	2522	2523	2524	2525	2529	2532	2635	2636	2640	2661	2662	2665
	2718	2723	2724	2731	2732	2737	2741	2746	2749	2753	2756	2759	2760	2762
	2763	2768	2778	2779	2794	2795	2797	2803	2806	2810	2813	2814	2816	2823
	2835	2836	2838	2839	2848	2850	2869	2874	2875	2918	2919	2935	2936	2941
	2942	2944	2947	2948	2950	2951	2953	2954	2955	2967	2968	2975	2976	2980
	2985	2986	2987	2988	3009	3025	3026	3032	3035	3039	3040	3041	3042	3058
	3059	3063	3071	3072	3074	3076	3078	3104	3114	3120	3123	3124	3127	3130
	3133	3138	3146	3150	3153	3156	3157	3163	3172	3181	3188	3189	3198	3200
	3201	3218	3220	3221	3223	3231	3232	3233	3234	3239	3242	3244	3245	3250
	3252	3253	3254	3258	3260	3266	3278	3279						
TAC1	6#	6	485	487	504	506	512	514	521	522	523	524	525	526
	529	530	531	532	533	553	554	556	600	616	619	620	621	627
	628	650	651	653	655	709	767	768	770	771	823	824	826	831
	832	835	844	846	848	872	873	874	881	892	893	1026	1029	1034
	1036	1052	1054	1072	1121	1123	1128	1129	1132	1141	1142	1146	1172	1173
	1185	1245	1246	1248	1249	1251	1256	1283	1294	1296	1297	1302	1304	1305
	1312	1321	1337	1346	1352	1353	1354	1359	1369	1370	1371	1400	1401	1402
	1425	1426	1430	1431	1433	1434	1505	1507	1509	1512	1513	1558	1561	1562
	1565	1566	1568	1569	1572	1573	1576	1580	1596	1597	1599	1600	1607	1608
	1610	1644	1650	1651	1722	1723	1741	1742	1776	1785	1786	1794	1795	1796
	1797	1799	1801	1805	1821	1822	1823	1824	1825	1826	1831	1871	1876	1877
	1881	1882	1883	1884	1898	1922	1923	1924	1930	1931	1947	1950	1951	1953
	1954	1958	1972	1973	1982	1984	1985	2004	2005	2007	2010	2013	2030	2046
	2047	2050	2060	2066	2067	2072	2088	2089	2108	2109	2110	2118	2119	2120

	2153	2154	2157	2164	2165	2190	2190	2196	2200	2204	2205	2206	2207	2208
	2209	2210	2211	2230	2231	2253	2263	2273	2274	2276	2279	2280	2283	2309
	2313	2315	2319	2327	2328	2369	2404	2405	2408	2411	2431	2435	2459	2461
	2462	2482	2483	2484	2493	2494	2513	2514	2522	2523	2530	2545	2546	2547
	2549	2550	2556	2557	2558	2559	2562	2565	2577	2578	2584	2589	2591	2593
	2595	2598	2599	2607	2658	2659	2665	2682	2684	2685	2686	2687	2688	2702
	2703	2704	2719	2722	2725	2729	2730	2732	2740	2745	2749	2751	2752	2755
	2763	2769	2799	2800	2801	2804	2806	2808	2809	2812	2818	2821	2822	2832
	2833	2839	2848	2849	2865	2866	2867	2880	2881	2943	2944	2955	3011	3013
	3043	3047	3048	3050	3051	3057	3059	3060	3062	3077	3078	3121	3122	3123
	3181	3182	3183	3189	3195	3215	3218	3221	3222	3223	3224	3225	3234	3236
	3237	3244	3248	3258	3274	3276	3277	3282						
TB	57#	792	810											
TBITS	109#	493	553	556	602	711	740	749	893	956	1014	1062	1085	1103
	1121	1128												
TEM	6#	6												
THSDAT	462	521	529	690	829	1143	1145							
TIME	462	527												
TPOPJ	584	3093	3202											
TRIES	89#	439	2835	2874										
TRNERR	6#	6												
TTYATC	6#	6												
TTYBIU	6#	6												
TTYUSE	6#	6												
UADRCR	2573	2594	2596											
UBFS	77#	84	769	878	1174	2203	2275	2421	2423	2575	2606	2622	2879	3184
	3187													
UBFU	79#	2597	2627											
UCHN	6#	6												
UDLKC	584	590												
UIONOD	6#	6												
UPDA	1539	2343	2359	2386#	2428	2446								
UPDEVIC	1570	2121	3231#											
UPDVC1	3243	3248#												
UPDVC2	3238	3250#												
USRCNT	168#	385	2842	2933	3002									
USRGO	2851	3027	3034#											
USRG03	3045	3050#												
USRM00	6#	6												
UUD	6#	6	465	517	560	563	564	569	586	626	632	633	634	675
	700	718	752	755	756	759	765	766	784	822	825	828	833	834
	842	843	845	847	911	912	914	915	917	919	923	925	1529	1533
	1658	1712	1716	1760	1864	1873	1878	1916	1921	1932	1960	1970	1976	3121
UWP	6#	6												
UWPOFF	6#	6												
UXFERS	173#	3036												
VDSKSR	23#	23												
VRGPTR	97#	542	638	644	2146									
WBRIT	72#	397	2838	3009										
WAIT1	1025	1041	1859	1860	2349									
WERA	1384	1427	2267	2270	2500	3159#								
WLRIT	54	135#	135	2234	2309	2406	3281							
WLERA	1385	1432	3266#											

WPRO	99#	548	619	620	644				
WRI	773	837	1360	2169	2261#				
WRIA	2198	2219	2262#						
WRIB	1055	1147	2260#						
WSYNC	422	436	438	445	450	2095	2140	2863	2895
WTRIT	113#	485	487	494	957				
WTMASK	6#	6							
WUFD	850	889	1074	1289	1394#				
WUFD1	1412#	1435							
WUFD3	1413	1425#							

CODES	6#														
DISARL	6#														
FNABLE	6#														
ERRORS	3095#	3102	3112	3136	3144	3161	3170								
NOSCHE	6#	138#	489	510	551	624	735	870	952	1076	1099	1126	1319	2221	
	2527	2553	2840	2877											
NOSHUF	6#														
QUEUES	6#														
SCHEOU	6#	144#	477	502	519	557	604	629	713	750	802	862	886	1092	
	1135	1326	1555	2212	2534	2563	2612	2853	2893	3142					
SHUFFL	6#														
STARTD	6#														
XP	6#	6	22	124	128	134	190	197	238	245	248	265	267	269	
	272	275	279	283	286	289	292	295	297	298	299				