

LENGTH OF PRG 01403

00001

DEBUG EQU

1

01352 P  
 00001 P  
 00020 P  
 00021 P  
 00006 P  
 00007 P  
 01164 P  
 01062 P  
 00665 P  
 00741 P  
 00431 P  
 01123 P  
 00641 P  
 00631 P  
 00630 P  
 00661 P  
 00407 P  
 01347 P  
 00326 P  
 00136 P  
 00001 P  
 00005 P  
 00006 P  
 00003 P  
 00007 P  
 00002 P  
 00004 P  
 01350 P  
 00031 P  
 01253 P  
 01313 P  
 01324 P  
 00545 P  
 01353 P  
 00033 P  
 00540 P

1  
2  
3  
3+001  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77

IDENT DISKOP  
 INCLUDE ↑SYSMAC  
 COSY/ 03 V4.1 08/17/74 0453  
 MACRO NAME DINTC  
 IF DEBUG EQ 0, EXIT  
 CRA 0  
 ASE \*  
 HLT \*  
 END \*  
 MACRO NAME LEVELC  
 IF DEBUG EQ 0, EXIT  
 INA 0036B  
 EXT INTPDL  
 ASE INTPDL  
 HLT \*  
 END \*  
 ENTRY FCLIST  
 ENTRY FILE  
 ENTRY FINK  
 ENTRY FINKP1  
 ENTRY FIRE  
 ENTRY FIREP1  
 ENTRY FIX  
 ENTRY FLOAT  
 ENTRY FREEFILE  
 ENTRY PREPANCIO  
 ENTRY GETBUFF  
 ENTRY GETCORE  
 ENTRY GIVBUFF  
 ENTRY GIVBUFFA  
 ENTRY GIVBUFFP  
 ENTRY IMPURE05  
 ENTRY MSC  
 ENTRY MSCPTR  
 ENTRY MXBT  
 ENTRY MXNE  
 ENTRY MXQADD  
 ENTRY MXQCOM  
 ENTRY MXQCSIAI  
 ENTRY MXQERR  
 ENTRY MXQISTAT  
 ENTRY MXQQ  
 ENTRY MXQWC  
 ENTRY MXSLIST  
 ENTRY PURE05  
 ENTRY RESERVE  
 ENTRY REWRITE  
 ENTRY REWRITEX  
 ENTRY RTN  
 ENTRY RTNCOUNT  
 ENTRY SWAP  
 ENTRY VALUE  
 EXT A  
 EXT ARRAYTBL  
 EXT BIT15  
 EXT BIT16  
 EXT BIT17  
 EXT BIT23  
 EXT BLOCKTBL  
 EXT BT201918  
 EXT D10  
 EXT DKCONTAB  
 EXT DKINT  
 EXT DKSTATAB  
 EXT DKSTATBL

BAD TRACK  
 NORMAL END

LENGTH OF DKSTATAB

```

78 EXT F7
79 EXT FBPP
80 EXT FLAGS
81 EXT FREEBLK
82 EXT FREEMEM
83 EXT GETBLK
84 EXT GETMEM
85 EXT HOUR
86 EXT I1
87 EXT I2
88 EXT IOBOUND
89 EXT IOBUSY
90 EXT MACHERR
91 EXT MSBIT
92 EXT MSQTOTAL
93 EXT MSUNITS
94 EXT MSWAIT
95 EXT MXBTRET
96 EXT MXIRERR
97 EXT MXWAITQ
98 EXT NBIT23
99 EXT NMSWAIT
100 EXT NUMDKCON
101 EXT OPMSG
102 EXT OPMSGX
103 EXT PAGETABL
104 EXT PC
105 EXT PF1
106 EXT READRTN
107 EXT READ
108 EXT RETURN
109 EXT RMDONE
110 EXT SELECT
111 EXT SWAPEXIT
112 EXT SWBIT
113 EXT SYSERR
114 EXT WRITE
115 EXT XFLAG

```

```

07773
07774
00000
00000
00000
00000

```

```

117 DINT EQU 7773B
118 EINT EQU 7774B
119 IMPURE EQU 00000B
120 PFR EQU 000B
121 PFW EQU 000B
122 JMP EQU 000B

```

00776

124 WPFB EQU 510

FILE BLOCK SIZE (IN WORDS)

FCBDEF

```

00000
00001
00002
00003
00004

```

```

71 ACCWORD EQU 0
72 LP EQU 1
73 COREP EQU 2
76 CBP EQU COREP+1
77 CPP EQU 4

```

```

*****
FILE CONTROL BLOCK DEFINITIONS
*****
ACCOUNTING WORD (MUST BE 0)
LOAD POINT BLOCK
CORE POINTER IF NON-ZERO
IF BIT23 = 1, CORE BLOCK HAS
BEEN WRITTEN INTO
BLOCK NUMBER OF THE CURRENT BLOC
CURRENT POSITION POINTER
(REL. POSIT. WITHIN BLOCK CBP)
BIT23 SEZ READ-ONLY
BIT22 SEZ AT LOAD POINT
BIT21 SEZ END OF DATA
BIT20 SEZ FILE MARK JUST READ
BIT18 SEZ BINARY RECORD PROCESSE
BIT17 SEZ ABNORMAL/UNAVAILABLE
BIT16 SEZ ADDRESS ERROR
BIT15 SEZ SAVED FILE
NUMBER OF BLOCKS BEYOND
THE CURRENT BLOC
END POSITION POINTER
BIT22 SEZ THE FILE HAS CHANGED
BIT21 SEZ POSITIONER READY
BIT20 SEZ DESTRUCTIVE READ

```

```

00005
00006

```

```

87 BLKR EQU 5
89 EPP EQU 6

```

93

FILE DIRECTORY \*  
 BITS 15-18 CONTAIN THE HT \*  
 BITS 00-14 CONTAIN END POSITION \*  
 TOTAL LENGTH IN BLOCKS \*

```

    94      *
    95      *
    96      *
    00007   97      TFL      EQU      7
    128
    129
    00001   130     X1       EQU      1
    00002   131     X2       EQU      2
    00003   132     X3       EQU      3
    00001   133     CNBLK    EQU      X1
    00002   134     CPPX     EQU      X2
    00003   135     PSA      EQU      X3
    136
    X       137     AUB      EQU      BIT17
    138
    139
    00022   140     CLOCK    EQU      228
    00001   141     PFLOC    EQU      0018
    04000   142     CORE     EQU      PFLOC*2+11
    143
    
```

\*\*\*\*\*  
 \* THE FOLLOWING ARE FOR FOLIST ELEMENTS \*  
 \* \*\*\*\*\*

```

    149
    00000   150     FCS      EQU      0
    X       151     IOP      EQU      BIT23
    X       152     OCC      EQU      BIT16
    X       153     ALT      EQU      BIT15
    X       154     NIOP     EQU      NBIT23
    00001   155     FCT      EQU      1
    156
    00002   157     FCA      EQU      2
    158
    00003   159     FCB      EQU      FCA+1
    160
    161
    162
    
```

STATUS IN THE UPPER 9 BITS  
 I/O IN PROGRESS  
 OCCUPIED  
 ALTERED  
 NOT I/O IN PROGRESS  
 LAST REFERENCE TIME FOR A  
 FILE CORE BLOCK  
 VFD A9/QUARTER PAGE NUMBER  
 A15/ PCINTER TO THE PROPER  
 CONTROL BLOCK  
 DISK BLOCK NUMBER FOR THIS  
 FILE CORE BLOCK

\*\*\*\*\*  
 \* MXQ DEFINITIONS \*  
 \* \*\*\*\*\*

```

    168
    00001   169     MXQADD   EQU      1
    00002   170     MXQQ     EQU      MXQADD+1
    00003   171     MXQERR   EQU      3
    172
    00004   173     MXQWC    EQU      4
    174
    00005   175     MXQCOM   EQU      MXQWC+1
    176
    177
    00006   178     MXQCSTAT EQU      6
    00007   179     MXQISTAT EQU      7
    00007   180     MXQPR    EQU      MXQISTAT
    181
    
```

DISK ADDRESS OR FILE BLOCK NUMBER  
 18 BIT CORE ADDRESS  
 ERROR COUNT IN UPPER 12 BITS  
 ALLOWABLE COUNT IN LOWER 12 BITS  
 I/O REQUEST IN UPPER 9 BITS  
 WORD COUNT IN LOWER 15 BITS  
 LOGICAL DEVICE NUMBER IN UPPER 6  
 BITS COMPLETION ADDRESS IN  
 LOWER 15 BITS  
 CHANNEL STATUS  
 STATUS RETURNED BY INTSORT  
 PRIORITY LEVEL OF A TRANSFER

\*\*\*\*\*  
 \* MASS STORAGE TRANSFER PRIORITIES \*  
 \* \*\*\*\*\*

```

    187
    00144   188     SUBPR    EQU      100
    00012   189     FINKPR   EQU      10
    00010   190     FIREPR   EQU      8
    00006   191     SWAPPR   EQU      6
    00004   192     FEARPR   EQU      4
    00003   193     FREEPR   EQU      3
    00002   194     FILEPR   EQU      2
    
```

THIS MUST BE THE HIGHEST LEVEL

```

*****
197 *
198 * MXSHARE *
199 *
200 * THIS ROUTINE IS CALLED TO REQUEST A DISK TRANSFER *
201 *
202 * ENI WORD COUNT,X1 *
203 * ENI REQUEST,X2 READ WRITE GAMBLE ETC *
204 * ENI COMPLETION ADDRESS,X3 *
205 * FOR SYSTEM PACK *
206 * LDA DISK BLOCK NUMBER *
207 * LDQ 18 BIT CORE ADDRESS *
208 * FOR USER PACK *
209 * LDA SECTOR NUMBER *
210 * LDQ =VFD A6/UNIT NO,A18/18 BIT CORE ADDRESS *
211 *
212 * AND THEN *
213 * RTJ FINK OR FILE OR FIRE OR FEAR OR FREE OR SWAP *
214 *
215 *
216 * IT USES THE FOLLOWING TEMPS *
217 *
218 * TEMP1 PRIORITY OF THE TRANSFER *
219 * TEMP2 IMMEDIATE RETURN ADDRESS *
220 * TEMP8 ASSUMED TO HAVE UNIT NUMBER AFTER CALL TO MXADDGEN *
221 * TEMP9 COMPLETION ADDRESS *
222 *
223 *
224 * IT USES THE FOLLOWING ROUTINES *
225 *
226 * GETMEM CONVERT MXADDGEN DKHNDLR *
227 *
*****

```

```

229
230
00000 00000002
00001 01000000
00002 45001345 P
00003 25000000 P
00004 01000036 P
231
232 FILE 00 FILEPR
233 UJP IMPURE
234 STAQ MXAQ
235 LDAQ FILE-1
236 UJP MXSHARE
237
00005 00000010
00006 01000000
00007 45001345 P
00010 25000005 P
00011 01000036 P
238
239 FIRE 00 FIREPR
240 UJP IMPURE
241 STAQ MXAQ
242 LDAQ FIRE-1
243 UJP MXSHARE
244
00012 00000004
00013 01000000
00014 45001345 P
00015 25000012 P
00016 01000036 P
245
246 FEAR 00 FEARPR
247 UJP IMPURE
248 STAQ MXAQ
249 LDAQ FEAR-1
250 UJP MXSHARE
251
00017 00000012
00020 01000000
00021 45001345 P
00022 25000017 P
00023 01000036 P
252
253 FINK 00 FINKPR
254 UJP IMPURE
255 STAQ MXAQ
256 LDAQ FINK-1
257 UJP MXSHARE
258
00024 00000003
00025 01000000
00026 45001345 P
00027 25000024 P
00030 01000036 P
259
260 FREEPRWD 00 FREEPR+IMPURE
261 UJP IMPURE
262 STAQ MXAQ
263 LDAQ FREE-1
264 UJP MXSHARE
265
00031 P
266 PURE05 EQU *
267
00031 00000006
00032 01077777 X
00033 45001345 P
00034 25000031 P
00035 14104000
268
269 SWAPX 00 SWAPPR
270 UJP SWAPEXIT IMMEDIATE RETURN
271 STAQ MXAQ
272 LDAQ SWAPX
273 ENI 4000B,X1
274
00036 00036 P
00037 45001357 P
00037 77630000
270
271 MXSHARE EQU *
272 STAQ TEMP1 SAVE PRIORITY AND RETURN ADDRESS
273 JINTC
*****
274 * KLUDGE TO COUNT DISK TRANSFERS *

```

00042 53020020 276  
00043 15410000 277  
00044 04104000 278  
00045 15470000 279  
00046 53420020 280

TMA 203  
INA,S 100008  
ISE 40008,X1  
INA,S -100008+1  
TAM 203

\*\*\*\*\*  
\* END OF KLUDGE \*  
\*\*\*\*\*

00047 47301366 P 284  
00050 77550000 285  
00051 42005714 P 01363 0 286  
00052 14300003 287  
00053 00777777 X 288  
00054 53100000 289  
00055 13077760 290  
00056 53200000 291  
00057 13000017 292  
00060 21001366 P 293  
00061 45300004 294  
00062 20001357 P 295  
00063 40300007 296  
00064 25001345 P 297  
00065 45300001 298  
00066 13000030 299  
00067 54101352 P 300  
00070 53200000 301  
00071 17600002 302  
00072 03100160 P 303

STI TEMP9,X3 SAVE THE COMPLETION ADDRESS  
CIA SAVE THE CHANNEL INDEX REGISTER  
SACH CIRTEMP  
ENI 3,X3 GET AN EIGHT WORD BLOCK OF  
RTJ GETMEM FREE STORAGE  
TIA X1 WORD COUNT TO A  
SHAQ -15 SAVE IN Q  
TIA X2 I/O REQUEST TO A  
SHAQ 15 MERGE TOGETHER  
LDQ TEMP9 LOAD THE COMPLETION ADDRESS  
STAQ MXQWC,X3 SAVE THE GOOD STUFF IN THE BLOCK  
LDA TEMP1 SAVE THE PRIORITY LEVEL IN THE  
STA MXQPR,X3 BLOCK  
LDAQ MXAQ GET THE FILE BLOCK NUMBER AND THE  
STAQ MXQADD,X3 CORE ADDRESS AND SAVE THEM  
SHAQ 24 PUT THE FILE BLOCK NUMBER INTO IQ  
LDI FCLIST,X1 POINT TO THE BEGINNING OF THE Q  
TIA X2  
ANA 00002B CHECK FOR A USER DISK PACK  
AZJ,NE MXSR06 JUMP IF A SYSTEM PACK  
AZJ,NE MXSR04 JUMP IF A SYSTEM PACK  
ENI MXSR08,X2 ENTER THE RETURN ADDRESS  
STI TEMP9,X2  
LDA MXQQ,X3 GET THE UNIT NUMBER  
SHA -18  
TAI X2  
UJP MXADD01

00073 14200163 P 305  
00074 47201366 P 306  
00075 20300002 307  
00076 12077755 308  
00077 53600000 309  
00100 01000234 P 310

\*  
ENI MXSR08,X2 ENTER THE RETURN ADDRESS  
STI TEMP9,X2  
LDA MXQQ,X3 GET THE UNIT NUMBER  
SHA -18  
TAI X2  
UJP MXADD01

00101 20100004 311  
00102 03500155 P 312  
00103 20100001 313  
00104 04277777 X 314  
00105 01000146 P 315  
00106 03300160 P 316  
00107 37077777 X 317  
00110 03000115 P 318  
00111 20100003 319  
00112 53600000 320  
00113 14600000 321  
00114 40200002 322

MXSR01 LDA FCB+1,X1 GET THE FILE BLOCK NUMBER  
AQJ,NE MXSR03 JUMP IF NOT THE SAME ONE  
LDA FCS+1,X1 GET THE STATUS WORD  
ISE READ,X2 SKIP IF A READ REQUEST  
UJP MXSR05 JUMP IF NOT  
AZJ,LT MXSR06 JUMP IF THIS IS A READ AND IOP  
LPA OCC IS THE BLOCK OCCUPIED  
AZJ,EQ MXSR02 JUMP IF NOT  
LDA FCA+1,X1 GET THE PCINTER TO THE CONTROL  
TAI X2 BLOCK  
ENA 0 TELL WHO EVER IT IS THAT THINKS  
STA COREP,X2 HE HAS THIS BLOCK THAT HE  
DOES NOT

00115 21300002 324  
00116 00701036 P 325  
00117 14477777 326  
00120 40200003 327  
00121 20200002 328  
00122 44100003 329  
00123 20100001 330  
00124 35077777 X 331  
00125 40100001 332  
00126 20200000 333  
00127 17677777 334  
00130 40200000 335  
00131 11700000 60000 0 336  
00132 37100003 337  
00133 12000022 338  
00134 40300002 339  
00135 14100211 P 340  
00136 00136 P 341

MXSR02 LDQ MXQQ,X3 GET THE CORE ADDRESS  
RTJ CONVERT GET THE PCINTER TO THE FCLIST  
ENA,S 77777B ELEMENT  
STA FCB,X2 SET THE BLOCK NUMBER TO -0  
LDA FCA,X2 MOVE ALL INFORMATION ABOUT  
SWA FCA+1,X1 THIS BLOCK TO THE ELEMENT  
LDA FCS+1,X1 ADD THE IOP BIT  
SSA IOP ON THE REPLACEMENT FCB  
STA FCS+1,X1  
LDA FCS,X2 REMOVE IOP BIT ON THE...  
ANA 77777B ...ORIGINAL FILE CORE BLOCK  
STA FCS,X2

00131 11700000 60000 0 337  
00132 37100003 338  
00133 12000022 339  
00134 40300002 340  
00135 14100211 P 341  
00136 00136 P 342  
00137 14677777 X 343  
00138 35077777 X 344  
00140 40000137 X 345  
00141 53300000 346  
00142 40300000 347  
00143 35077777 X 348  
00144 40401347 P 349  
00145 01100000 350

ECHA,S 300000B GET THE 1/4 PAGE NUMBER  
LPA FCA+1,X1 SAVE THE 18 BIT CORE ADDRESS  
SHA 18  
STA MXQQ,X3  
ENI MXXIT,X1 SET THE RETURN  
EQU \*  
ENA MSBIT TELL INTSORT THAT MASS STORAGE  
SSA FLAGS HAS FINISHED  
STA FLAGS  
TIA X3 MXQ BLOCK ADDRESS TO A  
STA 0,X3  
SSA BIT17 LINK THE BLOCK INTO THE  
STA,I MSCPTR COMPLETION QUEUE  
UJP 0,X1 EXIT

00146 00146 P 351  
00146 11700000 60000 0 352  
00146 11700000 60000 0 353

MXSR05 EQU \*  
ECHA,S 300000B FORGET WHAT IS IN THE FILE CORE

00147	37100003		354	LPA	FCA+1,X1	BLOCK IF IT IS NOT THE ONE WE ARE
00150	12000022		355	SHA	18	CURRENTLY WORKING WITH
00151	36300002		356	SCA	MXQQ,X3	
00152	03000155	P	357	AZJ,EQ	MXSR03	ITS THE SAME ONE
00153	14477777		358	ENA,S	77777B	
00154	40100004		359	STA	FCB+1,X1	
00155	20100001		360	LJA	FCS+1,X1	GET THE POINTER TO THE NEXT ELEME
00156	53500000		361	TAI	X1	PUT IT INTO X1
00157	02500101	P	362	MXSR04	MXSR01,X1	AND LCCP TILL DCNE
00160	21001345	P	363	MXSR06	MXAQ	LOAD THE FILE BLOCK NUMBER
00161	14200163	P	364	ENI	*+2,X2	ENTER THE RETURN
00162	01000214	P	365	UJP	MXADDGEN	GO LINK THE BLOCK INTO THE
			366			PROPER QUEUE
	00163	P	367	* MXSR08	EQU *	
00163	53300000		368	TIA	X3	CURRENT BLOCK ADDRESS TO (A)
00164	54101365	P	369	LOI	TEMP8,X1	LOAD THE LOGICAL DEVICE NUMBER
00165	36177777	X	370	SCA	MXWAITQ,X1	IS THIS BLOCK ON THE FRONT OF
00166	04600000		371	ASE	0	IT'S QUEUE
00167	01000211	P	372	UJP	MXXIT	JUMP IF NOT
00170	53100000		373	TIA	X1	DEVICE NUMBER TO A
00171	14700477		374	ENQ	477B	WHICH CONTROLLER IS THE DEVICE
00172	13000017		375	SHAQ	15	USING
00173	14177777	X	376	ENI	DKSTATBL,X1	
00174	06177777	X	377	MEQ	DKSTATAB,1	
00175	00777777	X	378	RTJ	SYSERR	
00176	14777000		379	ENQ	77000B	MASK FOR CHANNEL AND EQUIPMENT
00177	27100174	X	380	LDL	DKSTATAB,X1	LOAD CHANNEL AND EQUIPMENT
00200	14177777	X	381	ENI	NUMDKCON,X1	NUMBER OF DISK CONTROLLERS
00201	06177777	X	382	MEQ	DKCONTAB,1	
00202	00700175	X	383	RTJ	SYSERR	
00203	20100201	X	384	LDA	DKCONTAB,X1	IS THE CONTROLLER BUSY
00204	03300211	P	385	AZJ,LT	MXXIT	JUMP IF IT IS
00205	12077760		386	SHA	-15	FAKE AN EQUIPMENT INTERRUPT
00206	53500000		387	TAI	X1	
00207	14300211	P	388	ENI	*+2,X3	
00210	01077777	X	389	UJP	DKINT	
			390			
	00211	P	391	* MXXIT	EQU *	
00211	22005714	P	392	LACH	CIRTEMP	RESTORE THE CHANNEL INDEX
00212	77540000		393	ACI		REGISTER
00213	01401360	P	394	UJP,I	TEMP2	EXIT

```

397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *

```

\*\*\*\*\*

```

417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *

```

```

00214 47201366 P
00215 20001350 P
00216 03000226 P
00217 53500000
00220 20100001
00221 03500224 P
00222 21100002
00223 01000215 P
00224 20100000
00225 03100217 P
00226 13000030
00227 14577777
00230 14277777 X
00231 07177777 X
00232 01000315 P
00233 13000030
00234 41300001 P
00235 20300005
00236 12000006
00237 17477700
00240 53240000
00241 12000022
00242 40300005
00243 20277777 X
00244 03200250 P
00245 14477776 P
00246 34300005
00247 01000135 P
00250 47201365 P
00251 20200165 X
00252 14200000
00253 17677777 P
00254 03000264 P
00255 53500000
00256 20100001
00257 03500262 P
00260 53100000
00261 53600000
00262 20100000
00263 01000253 P
00264 40300006 P
00265 04200000
00266 01000301 P
00267 54201365 P
00270 15200251 X
00271 21300007
00272 00272 P

```

```

MXADDGEN EQU *
MXSCK1 LDA MXSLIST
AZJ, EQ MXSCK4
MXSCK2 TAI X1
LDA 1, X1
AQJ, NE MXSCK3
LDQ 2, X1
UJP MXSCK1
MXSCK3 LDA 0, X1
AZJ, NE MXSCK2
MXSCK4 EQU *
SHAQ 24
ENQ, S 77777B
ENI MSUNITS, X2
MTH BLOCKTBL, 1
IF DEBUG, UJP MXADD12
IF DEBUG EQ 0, UJP MXADD01X
MXADD01 EQU *
STQ MXQADD, X3
LDA MXQCOM, X3
SHA 6
ANA, S 77700B
AIA X2
SHA 18
STA MXQCOM, X3
LDA ARRAYTBL, X2
AZJ, GE *+4
MXADD01X EQU *
ENA, S -1
RAD MXQCOM, X3
UJP MXNEX
MXADD02 EQU *
LDA MXWAITQ, X2
ENI 0, X2
MXADD02 EQU *
ANA 77777B
AZJ, EQ MXADD04
TAI X1
LDA MXQADD, X1
AQJ, NE *+3
TIA X1
TAI X2
LDA 0, X1
UJP MXADD02
MXADD04 EQU *
STA MXQCSTAT, X3
ISE 0, X2
UJP MXADD08
LDA TEMP8, X2
INI MXWAITQ, X2
LDQ MXQPR, X3
MXADD06 EQU *

```

```

SAVE THE RETURN ADDRESS
JUMP IF NO SUBSTITUTION
JUMP IF NO SUBSTITUTION
LOAD THE NEW BLOCK NUMBER
CHECK FOR MULTIPLE SUBSTITUTIONS
LOAD THE NEXT POINTER
JUMP IF LIST IS NOT EMPTY
FILE BLOCK NUMBER IS NOW IN Q
LOCK AT ALL THE BITS
SEARCH THE MASS STORAGE TABLES
JUMP IF ILLEGAL ADDRESS
JUMP IF ILLEGAL ADDRESS
DISK ADDRESS TO Q
SAVE THE DISK ADDRESS
PUT THE UNIT NUMBER INTO THE
MXQ BLOCK
PUT THE WORD BACK TOGETHER
IS THE DEVICE ON LINE
TAKE THE IR RETURN
SAVE THE DEVICE NUMBER
GET THE POINTER TO THE FIRST ELEM
WE HAVE NOT FOUND A MATCH
JUMP IF AT THE END OF THE QUEUE
POINTER TO X1
GET THE ADDRESS ON THIS ELEMENT
JUMP IF DIFFERENT
PUT THE POINTER INTO X2
GET THE ADDRESS OF THE NEXT BLOCK
AND LCCP BACK
CLEAR THE CHANNEL STATUS
SKIP IF NO MATCH
JUMP IF WE FOUND A MATCH
GET THE LOGICAIL DEVICE NUMBER AG
POINT TO THE PROPER PLACE IN THE
GET PRIORITY

```

00272	20200000		475	LDA	0,X2	GET PCINTER
00273	17677777		476	ANA	77777B	MASK TO BOTTOM 15 BITS
00274	03000304	P	477	AZJ, EQ	MXADD10	JUMP IF END OF QUEUE
00275	53600000		478	TAI	X2	GET POINTER
00276	20200007		479	LDA	MXQPR, X2	GET THIS ONE'S PRIORITY
00277	03700303	P	480	AQJ, LT	MXADD09	JUMP IF LOWER PRIORITY
00300	01000272	P	481	UJP	MXADD06	CONTINUE SEARCHING FOR HOLE
			482			
	00301	P	483	MXADD08	EQU	*
00301	20200007		484	LDA	MXQPR, X2	CHANGE THE PRIORITY ON THIS
00302	40300007		485	STA	MXQPR, X3	TRANSFER
			486			
00303	20200000		487	MXADD09	LDA	0, X2
00304	44300000		488	MXADD10	SWA	0, X3
00305	53300000		489	TIA	X3	
00306	44200000		490	SWA	0, X2	MOVE THE POINTERS AROUND
00307	20300004		491	LDA	MXQWC, X3	LOAD THE I/C REQUEST
00310	14700002		492	ENQ	2	ALLOWABLE ERRORS FOR A WRITE
00311	05400000		493	ASG, S	0	SKIP IF A WRITE
00312	14700143		494	ENQ	99	ALLOWABLE ERRORS FOR A READ
00313	41300003		495	STQ	MXQERR, X3	SAVE THE ERROR WORD
00314	01401366	P	496	UJP, I	TEMP9	RETURN TO THE CALLER
			497			
			498			
			499	IF	DEBUG EQ 0, GOTO DEBUG01	
	00315	P	500	MXADD12	EQU	*
00315	14177770		501	ENI	-7, X1	PRINT 8 CHARACTERS
00316	13000003		502	SHAQ	3	
00317	17700007		503	ANQ	7B	
00320	43405772	P	504	SQCH	BADBLOCK+7, X1	SAVE A CHARACTER
00321	02100316	P	505	IJI	*-3, X1	
00322	11005763	P	506	ECHA	BADBLOCK	MESSAGE ADDRESS
00323	14700026		507	ENQ	BADBLKL	MESSAGE LENGTH
00324	14200245	P	508	ENI	MXADD01X, X2	RETURN ADDRESS
00325	01077777	X	509	UJP	OPMSGX	
			510			

.DEBUG01



```

*****
513 *
514 *          MXBT
515 *
516 *          THIS ROUTINE IS ENTERED FROM DKHNDLR AFTER A FATAL DISK
517 *          ERROR IF THE REQUEST WAS A READ OR WRITENS THE CALLER IS
518 *          INFORMED THAT THE REQUEST WAS BAD OTHERWISE A SUBSTUTION IS
519 *          MADE FOR THE BAD ADDRESS IF THE WORD COUNT OF THE TRANSFER
520 *          IS ONE FILE BLOCK A FILE BLOCK IS USED FOR THE SUBSTUTION
521 *          OTHERWISE A PAGE IS ALWAYS USED
522 *
523 *
524 *          IT USES THE FOLLOWING TEMPS
525 *
526 *          TEMP1      NEW BLOCK NUMBER
527 *          TEMP2      ORIGINAL BLOCK NUMBER
528 *          TEMP5      MASS STORAGE UNIT THE ORIGINAL BLOCK WAS ON
529 *
530 *
531 *          IT USES THE FOLLOWING ROUTINES
532 *
533 *          GETMEM     MXADDGEN  GETBLK
534 *
*****
    
```

00326	20300004	P	536			
00327	37077777	X	537			
00330	12000003		538	MXBT	EQU	*
00331	35300004		539		LOA	MXQWC,X3
00332	40300004		540		LPA	BT201918
00333	12000006		541		SHA	3
00334	03377777	X	542		SSA	MXQWC,X3
00335	53300000		543		STA	MXQWC,X3
00336	53600000		544		SHA	6
00337	14300002		545		AZJ,LT	MXIRERR
00340	00700053	X	546		TIA	X3
00341	21200001		547		TAI	X2
00342	41001360	P	548		ENI	2,X3
00343	20001350	P	549		RTJ	GETMEM
00344	47301350	P	550		LDQ	MXQADD,X2
00345	45300000		551		STQ	TEMP2
00346	20200005		552		LOA	MXSLIST
00347	12077755		553		STI	MXSLIST,X3
00350	44001363	P	554		STAQ	0,X3
00351	20200004		555		LOA	MXQCOM,X2
00352	14200001		556		SHA	-18
00353	04601000		557		SWA	TEMP5
00354	14277777	X	558		LOA	MXQWC,X2
00355	14100000		559		ENI	1,X2
00356	00777777	X	560		ASE	WPFB+2
00357	40300002		561		ENI	F0PP,X2
00360	40001357	P	562		ENI	0,X1
00361	54301363	P	563		RTJ	GETBLK
			564		STA	2,X3
			565		STA	TEMP1
			566		LOI	TEMP5,X3
			567			
00362	24000124	X	568		LCA	BIT23
00363	37300270	X	569		LPA	MXWAITQ,X3
00364	40300363	X	570		STA	MXWAITQ,X3
00365	15300364	X	571		INI	MXWAITQ,X3
	00366	P	572	MXBT06	EQU	*
00366	47301363	P	573		STI	TEMP5,X3
	00367	P	574	MXBT08	EQU	*
00367	54201363	P	575		LOI	TEMP5,X2
00370	20200000		576		LOA	0,X2
00371	17677777		577		ANA	77777B
00372	03077777	X	578		AZJ,EQ	MXBIRET
00373	53700000		579		TAI	X3
00374	20300001		580		LOA	MXQADD,X3
00375	21001360	P	581		LDQ	TEMP2
00376	03500366	P	582		AQJ,NE	MXBT06
00377	20300000		583		LOA	0,X3
00400	44200000		584		SWA	0,X2
00401	14600144		585		ENA	SUBPR
00402	40300007		586		STA	MXQPR,X3
00403	21001357	P	587		LDQ	TEMP1
00404	47201363	P	588		STI	TEMP5,X2
00405	14200367	P	589		ENI	MXBT08,X2
00406	01000214	P	590		UJP	MXADDGEN

```

LOAD THE REQUEST WORD
RESTORE THE ORIGINAL REQUEST
CHECK FOR NO SUBSTUTIONS
OH WELL
MXQ POINTER TO X2
GET 4 WORDS FOR A MXSLIST ELEMENT
GET THE ORIGINAL BLOCK NUMBER
SAVE THE BLOCK NUMBER
LINK THE BLOCK INTO THE MXSLIST
LOAD THE UNIT NUMBER
SAVE FOR LATER
PRETEND WE WANT 1 BLOCK
NUMBER OF FILEBLOCKS/PAGE
SAY WE PREFER MS00
FINISH THE MXSLIST ELEMENT
KEEP THE BLOCK NUMBER HANDY
LOAD THE UNIT NUMBER
CLEAR BIT23 ON THE MXQ POINTER
POINT TO THE PROPER ELEMENT
PUT THE PCINTER INTO X2
GET THE NEXT BLOCK ADDRESS
JUMP IF THE END OF THE QUEUE
IS THIS THE SAME BLOCK
LOAD THE ORIGINAL BLOCK NUMBER
JUMP IF NOT
REMOVE THIS BLOCK FROM THIS
QUEUE
SET THE PRIORITY WORD
LOAD THE NEW BLOCK NUMBER
SAVE THE QUEUE POSITION
ENTER THE RETURN
GO LINK THE BLOCK INTO THE NEW
    
```

```

593 *
594 *      MASS STORAGE COMPLETION ROUTINE
595 *
596 *      THIS ROUTINE IS ENTERED IF THE MSBIT IS SET IN FLAGS
597 *
598 *      THIS WILL RETURN TO A ROUTINE AFTER A REQUESTED DISK
599 *      TRANSFER IS FINISHED THE RETURN ADDRESS IS IN X3 AND THE
600 *      Q REGISTER HAS THE ADDRESS OF THE FILE CORE BLOCK THAT WAS
601 *      USED
602 *
*****

```

```

604
605
606 MSC EQU *
607 VFD A12/EINT
00407 77740000 P
00410 20001347 P LDA MSCPTR GET THE QUEUE PCINTER
00411 53500000 TAI X1 PUT IT INTO X1
00412 120000006 SHA 6
00413 03277777 X AZJ,GE RETURN EXIT IF THE QUEUE IS EMPTY
00414 77730000 VFD A12/DINT
00415 21100000 LDQ 0,X1 LOAD THE FORWARD POINTER
00416 27000143 X LDL BIT17 CHECK THE INDIRECT BIT
00417 03100421 P AZJ,NE *+2 JUMP IF NOT EMPTY
00420 14701347 P ENQ MSCPTR SET THE POINTER PROPERLY IF EMPTY
00421 41001347 P STQ MSCPTR
00422 53100000 TIA X1 FREE THE TRANSFER BLOCK
00423 143000003 ENI 3,X3
00424 00777777 X RTJ FREEMEM
00425 211000002 LDQ MXQQ,X1 LOAD THE CORE ADDRESS
00426 201000007 LDA MXQISTAT,X1 LOAD THE LEFT OVER WORD COUNT
00427 14300407 P ENI MSC,X3 ENTER THE RETURN ADDRESS
00430 015000005 UJP,I MXQCOM,X1 AND CALL THE COMPLETION ROUTINE
624

```

```

627 *
628 * GETBUFF *
629 *
630 * THIS ROUTINE IS CALLED TO GET A FILE CORE BLOCK *
631 *
632 * DINT *
633 * ENQ INTERRUPT RETURN *
634 * ENA IMMEDIATE RETURN *
635 * UJP GETBUFF *
636 *
637 * INDEXES 1 AND 3 ARE SAVED *
638 * THE ROUTINE RETURNS THE 1/4 PAGE NUMBER OF A FILE CORE BLOCK *
639 * IN THE A REGISTER *
640 *
641 * IF NO BLOCKS ARE AVAILABLE THE CALL IS QUEUED IN WAITQ *
642 * AND A = -1 ON THE IMMEDIATE RETURN AFTER BLOCKS BECOME *
643 * AVAILABLE RTN WILL CALL THE ROUTINE BACK *
644 *
645 *
646 * IT USES THE FOLLOWING TEMPS *
647 *
648 * TEMP3 INTERRUPT RETURN ADDRESS *
649 * TEMP4 CALLERS X3 *
650 * TEMP5 IMMEDIATE RETURN ADDRESS *
651 * TEMP7 CALLERS X1 *
652 *
*****
    
```

```

654
655
00431 44001363 P 656 GETBUFF SWA TEMP5
00432 77630000 657 DINTC
00435 41001361 P 658 STQ TEMP3 SAVE THE INTERRUPT ADDRESS
00436 47301362 P 659 STI TEMP4,X3 SAVE INDEX 3
00437 47101364 P 660 STI TEMP7,X1 SAVE INDEX 1
661
00440 21077777 X 662 LDQ NBIT23 INITIAL AGE VALUE
00441 54201352 P 663 LDI FCVLIST,X2 GET START OF FC LIST
00442 01000464 P 664 UJP FCV04
665
00443 20200000 666 FCV01 LDA FCS,X2 GET STATUS
00444 03300463 P 667 AZJ,LT FCV03 JUMP IF I/O IN PROGRESS
00445 12000001 668 SHA 1 GET RESERVED BIT
00446 03300462 P 669 AZJ,LT FCV02 JUMP IF RESERVED
00447 12077757 670 SHA -16 GET ALTERED AND OCCUPIED BITS
00450 53500000 671 TAI X1 PUT A INTO X1
00451 17100003 672 ANI 3B,X1 LEAVE THE IMPORTANT BITS
00452 20200003 673 LDA FC08,X2 GET BLOCK NUMBER
00453 05100002 674 ISG 2,X1 SKIP IF OCCUPIED
00454 03300473 P 675 AZJ,LT FCV06 JUMP IF A FREE BLOCK FOUND
00455 20200001 676 LDA FC1,X2 GET THE LAST REFERENCE TIME
00456 30100540 P 677 ADA VALUE,X1 BIAS THE TIME WITH ALTERED AND...
00457 03600462 P 678 AQJ,GE FCV02 ...OCCUPIED. JUMP IF NOT OLDER
00460 13000030 679 SHAQ 24 GET THE OLDEST AGE INTO Q
00461 47201354 P 680 STI LOWVALFC,X2 SAVE POINTER TO OLDER FC BLOCK
00462 20200000 681 FCV02 LDA FCS,X2 GET ADDRESS OF NEXT FC LIST EL.
00463 53600000 682 FCV03 TAI X2
00464 04200000 683 FCV04 ISE 0,X2 SKIP IF LAST ELEMENT
00465 01000443 P 684 UJP FCV01
685
00466 53020022 686 TMA CLOCK GET THE TIME RIGHT NOW
00467 30077777 X 687 ADA HOUR
00470 15470137 688 INA,S -4000 ADJUST TIME FOR 4 SEC. AGO
00471 03700473 P 689 AQJ,LT *+2 JUMP IF NOT 4 SEC. OLD
00472 54201354 P 690 LDI LOWVALFC,X2
00473 47201354 P 691 FCV06 STI LOWVALFC,X2 SAVE POINTER TO AVAILABLE ELEMENT
692
00474 04200000 693 ISE 0,X2 SKIP IF NO AVAILABLE FC BLOCK
00475 01000511 P 694 UJP GTC07 JUMP IF A USEABLE BLOCK FOUND
00476 22005704 P 01361 0 695 GTC05 LACH TEMP3 SHOULD WE QUEUE THIS CALL
00477 03100507 P 696 AZJ,NE GTC06 JUMP IF NOT
00500 14300002 697 ENI 2,X3 GET 4 WORDS FOR A WAITQ ELEMENT
00501 00700340 X 698 RTJ GETMEM
00502 40300000 699 STA 0,X3 SET THE POINTER WORD
00503 35000416 X 700 SSA BIT17 SET THE INDIRECT BIT
00504 40401351 P 701 STA,I WAITQ PUT THE ELEMENT INTO THE QUEUE
00505 25001361 P 702 LDAQ TEMP3 GET THE INTERRUPT ADDRESS
00506 45300001 703 STAQ 1,X3 AND THE CALLER'S INDEX 3
00507 14477776 704 GTC06 ENA,S -1 SAY THE CALL WAS QUEUED
    
```

00510	01000524	P	705		UJP	GTC09	
	00511	P	706	GTC07	EQU	*	
00511	21200000		707		LDQ	FCS,X2	GET THE STATUS WORD
00512	27000107	X	708		LDL	OCC	IS THE BLOCK OCCUPIED
00513	03100527	P	709		AZJ,NE	GTC10	JUMP IF IT IS
00514	20000362	X	710	GTC08	LDA	IOP	
00515	17777777		711		ANQ	77777B	CLOBBER ANY STATUS BITS
00516	41200000		712		STQ	FCS,X2	STORE THE STATUS BACK AND
00517	34200000		713		RAD	FCS,X2	SAY THAT I/O IS HAPPENING
00520	21200002		714		LDQ	FCA,X2	GET THE 1/4 PAGE NUMBER
00521	13000011		715		SHAQ	9	
00522	14577777		716		ENQ,S	77777B	SET THE BLOCK NUMBER TO
00523	41200003		717		STQ	FCB,X2	-0
00524	54301362	P	718	GTC09	LDI	TEMP4,X3	RESTORE INDEX X3
00525	54101364	P	719		LDI	TEMP7,X1	RESTORE INDEX 1
00526	01401363	P	720		UJP,I	TEMP5	RETURN
			721				
	00527	P	722	GTC10	EQU	*	
00527	20200002		723		LDA	FCA,X2	GET THE CONTROL BLOCK ADDRESS
00530	53500000		724		TAI	CNBLK	PUT IT INTO X1
00531	27077777	X	725		LDL	ALT	HAS THE BLOCK BEEN ALTERED
00532	03100535	P	726		AZJ,NE	GTC11	JUMP IF IT HAS
00533	40100002		727		STA	COREP,CNBLK	STEAL THE BLOCK IF IT HAS NOT
00534	01000514	P	728		UJP	GTC08	
			729				
00535	25100002		730	GTC11	LDAQ	COREP,X1	GET THE CORE ADDRESS AND FILE
00536	00701313	P	731		RTJ	REWRITE	BLOCK NUMBER AND REWRITE THE
00537	01000476	P	732		UJP	GTC05	BLOCK
			733				
00540	77742547		734	VALUE	VFD	A24/0-15000	NOT OCCUPIED, -15 SECONDS
00541	77742547		735		VFD	A24/0-15000	NOT OCCUPIED, -15 SECONDS
00542	00001750		736		VFD	A24/1000	NOT ALTERED, +1 SEC.
00543	00000000		737		VFD	A24/0	ALTERED, USE REAL AGE

```

740 *****
741 * THIS ROUTINE RETURNS A BLOCK OF CORE TO THE FCLIST *
742 * OR GIVES IT TO A QUEUED REQUEST IN CWAITQ *
743 *****

```

			745			
			746			
00544	00777777	X	747	RTJ	MACHERR	MEMORY PARITY ERROR
00545	14477776		748	ENA,S	-1	
00546	34001353	P	749	RAD	RTNCOUNT	FIX THE CCOUNTER
	00547	P	750	EQU	*	
00547	77630000		751	DINTC		
00552	00701036	P	752	RTJ	CONVERT	GET THE PCINTER TO THE FCLIST
00553	13000030		753	SHAQ	24	
00554	20200000		754	LDA	FCS,X2	
00555	37000627	P	755	LPA	RESMASK	
00556	40200000		756	STA	FCS,X2	
00557	12000001		757	SHA	1	
00560	03300564	P	758	AZJ,LT	RTN03	
00561	54101351	P	759	LDI	CWAITQ,X1	DOES ANYBODY WANT A FILE CORE
00562	04101351	P	760	ISE	CWAITQ,X1	BLOCK SKIP IF NOT
00563	01000570	P	761	UJP	RTN04	
	00564	P	762			
00564	53020022		763	RTN03	EQU	*
00565	30000467	X	764	TMA	CLOCK	GET THE CURRENT TIME
00566	40200001		765	ADA	HOOR	
00567	01300000		766	STA	FCT,X2	STORE IT INTO THE FCLIST ELEMENT
			767	RX3	UJP	EXIT
			768			
00570	41100003		769	RTN04	STQ	3,X1
			770	*		
00571	21100000		771	LDQ	0,X1	SAVE THE FILE CORE BLOCK 1/4 PAGE
00572	27000503	X	772	L0L	BIT17	ADDRESS IN THE CWAITQ ELEMENT.
00573	05400001		773	ASG,S	1	GET THE PCINTER WORD OF THE NEXT
00574	14701351	P	774	ENQ	CWAITQ	CWAITQ ELEMENT REMOVE THIS BLOCK
00575	41001351	P	775	STQ	CWAITQ	SKIP IF THE QUEUE IS NOT EMPTY
00576	53100000		776	TIA	X1	
00577	14300002		777	ENI	2,X3	
00600	00700424	X	778	RTJ	FREEMEM	
00601	25100002		779	LDAQ	2,X1	
00602	53700000		780	TAI	X3	RESTORE INDEX 3 FOR THE INTERRUPT
00603	20000514	X	781	LDA	IOP	
00604	34200000		782	RAD	FCS,X2	SET IOP INTO THE STATUS
00605	13000011		783	SHAQ	9	
00606	14577777		784	ENQ,S	77777B	SAY NO FILE BLOCK ASSOCIATED WITH
00607	41200003		785	STQ	FCS,X2	THIS FCLIST ELEMENT
00610	14200407	P	786	ENI	MSC,X2	RETURN TO MSC
00611	21001353	P	787	LDQ	RTNCOUNT	WILL BLOCKS BECOME AVAILABLE
00612	04500000		788	QSE,S	0	SKIP IF NOT
00613	01000617	P	789	UJP	RTN02	EXIT IF THEY WILL
00614	21001351	P	790	LDQ	CWAITQ	IF NO REWRITTING IS QUEUED AND
00615	04701351	P	791	QSE	CWAITQ	CWAITQ IS NOT EMPTY
00616	14200620	P	792	ENI	*+2,X2	RETURN TO HERE
00617	01500001		793	RTN02	UJP,I	1,X1
00620	14300407	P	794	ENI	MSC,X3	CALL THE ROUTINE THAT NEEDED THE
00621	14577777		795	ENQ,S	77777B	BLOCK ENTER THE RETURN ADDRESS
00622	14600624	P	796	ENA	*+2	SAY NOT TO QUEUE THE CALL
00623	01000431	P	797	UJP	GETBUFF	SET THE RETURN
00624	03300567	P	798	AZJ,LT	RX3	GO ASK FOR A BLOCK
00625	13077760		799	SHAQ	-15	DONT PANIC IF WE DIDNOT GET ONE
00626	01000547	P	800	UJP	RTN01	FORM CORE ADDRESS IN Q
			801			
00627	20077777		802	RESMASK	OCT	20077777

```

805 *
806 * GIVBUFF *
807 *
808 * CALL TO GIVE UP A FILE CORE BLOCK *
809 *
810 * IF WE SHOULD REMEMBER THE INFORMATION IN THE BLOCK *
811 *
812 * LDQ 18 BIT CORE ADDRESS *
813 * ENI RETURN,X3 *
814 * UJP GIVBUFF *
815 *
816 *
817 * IF WE WANT TO FORGET THE INFORMATION IN THE BLOCK *
818 *
819 * IF 1/4 PAGE NUMBER IS IN A CALL GIVBUFFA *
820 * IF 1/4 PAGE NUMBER IS IN PFLOC CALL GIVBUFFP *
821 * IF 18 BIT CORE ADDRESS IS IN Q CALL GIVBUFFX *
822 *
*****
    
```

```

824
825
826
00630 77650001 827 GIVBUFFP PFA PFLOC+PFR GET GLOBAL PAGE NUMBER FROM PFLOC
00631 13077760 828 GIVBUFFA SHAQ -15 CONVERT TO A CORE ADDRESS
      00632 P 829 GIVBUFFX EQU *
00632 77630000 830 DINTC
00635 00701036 P 831 RTJ CONVERT GO LOOK FOR THE FCLIST ELEMENT
00636 14477777 832 ENA,S 77778 FORGET THE INFORMATION IN THE
00637 40200003 833 STA FCB,X2 BLOCK
00640 01000642 P 834 UJP GIVBUFO1
      835
00641 00701036 P 836 GIVBUFF EQU *
      00642 P 837 RTJ CONVERT FIND THE FCLIST ELEMENT
00642 53300000 838 GIVBUFO1 EQU *
00643 53500000 839 TIA X3 PUT THE RETURN ADDRESS IN X1
00644 41001357 P 840 TAI X1
00645 14300003 841 STQ TEMP1 SAVE THE CORE ADDRESS
00646 00700501 X 842 ENI 3,X3 GET 8 WORDS FOR A FAKE MASS
00647 14600545 P 843 RTJ GETMEM STORAGE COMPLETION BLOCK
00650 40300005 844 ENA RTN SAY TO RETURN TO RTN
00651 20200000 845 STA MXQCOM,X3
00652 35000603 X 846 LDA FCS,X2 LOAD THE STATUS WORD
00653 40200000 847 SSA IOP MAKE THE CORE BLOCK UNAVAILABLE
00654 14600001 848 STA FCS,X2
00655 34001353 P 849 ENA 1 SAY ONE MORE BLOCK IS COMING
00656 21001357 P 850 RAD RTNCOUNT AVAILABLE
00657 41300002 851 LDQ TEMP1 LOAD THE CORE ADDRESS AGAIN
00660 01000136 P 852 STQ MXQQ,X3 SAVE THE ADDRESS IN THE DISK
      853 UJP BLOCK
      854
      855
      856
00661 00000000 857
      858 IMPURE05 VFD A24/IMPURE
    
```

```

861 *
862 * THIS ROUTINE IS TO FREE A FILE OR PART OF A FILE TO THE FREE *
863 * FILE BLOCK TABLE *
864 * ENTER WITH THE NUMBER OF BLOCKS TO BE FREED IN Q AND THE *
865 * NUMBER OF THE FIRST BLOCK IN THE A REGISTER *
866 * RTJ FREEFILE *
867 * THIS ROUTINE RESTORES ALL OF THE INDEX REGISTERS *
868 *
*****

```

```

870
871
00662 14200001 872 FREEZIP ENI 1,X2 SPECIFY A SINGLE BLOCK
00663 00777777 X 873 RTJ FREEBLK FREE THE FILE BLOCK
00664 14200000 874 FREEX2 ENI IMPURE,X2
00665 01000000 875 FREEFILE UJP IMPURE
00666 77730000 876 VFD A12/DINT
00667 05500001 877 QSG,S 1
00670 01000665 P 878 UJP FREEFILE THERE ARE NO BLOCKS IN THE FILE
00671 47200664 P 879 STI FREEX2,X2 SAVE INDEX 2
00672 05500002 880 QSG,S 2
00673 01000662 P 881 UJP FREEZIP NO QUEUEING TO FREE 1 FILE BLOCK.
00674 47100712 P 882 STI FREEX1,X1 SAVE INDEX 1
00675 47300713 P 883 STI FREEX3,X3 SAVE INDEX 3
00676 13000030 884 SHAQ 24
00677 34001355 P 885 RAD COUNT COUNT UP THE TOTAL NUMBER OF
886 * BLOCKS THAT NEED TO BE FREED
887 STAQ TEMP1 SAVE A AND Q
00700 45001357 P 888 ENI 2,X3 GET FOUR WORDS OF FREE MEMORY
00701 14300002 889 RTJ GETMEM
00702 00700646 X 890 LDAQ TEMP1 RESTORE A AND Q
00703 25001357 P 891 STAQ 1,X3 PUT THEM INTO THE QUEUE ELEMENT
00704 45300001 892 STI,I FREEFLAG,X3 LINK THE ELEMENT INTO THE LIST
00705 47701356 P 893 STI FREEFLAG,X3 FIX UP THE END POINTER
00706 47301356 P 894 LDA IOBUSY LOAD THE BUSY FLAG
00707 20077777 X 895 ENI *+2,X3 ENTER THE RETURN ADDRESS
00710 14300712 P 896 AZJ,GE FREESTRT JUMP IF THE QUEUE IS NOT BUSY
00711 03201005 P 897 ENI IMPURE,X1
00712 14100000 898 ENI IMPURE,X3
00713 14300000 899 UJP FREEX2
00714 01000664 P 900
00715 47000722 P 901 STI FREEIR,0 IRRECOVERABLE I/O ERROR
00716 47300025 P 902 STI FREE,X3 SAVE THE RETURN ADDRESS
00717 14300000 903 FREEQB ENI IMPURE,X3 GET POINTER TO QUEUE ELEMENT
00720 13000017 904 SHAQ 15 FILE CORE BLOCK 1/4 PAGE NO. TO A
00721 77640001 905 APF PFLOC+PFW FIX THE PAGE FILE
00722 14600077 906 FREEIR ENA 778+IMPURE ANY NON ZERO VALUE SEZZ NO ERROR
00723 03001020 P 907 AZJ,EQ FORGET JUMP IF IR I/O ERROR
00724 20300002 908 LDA 2,X3 LOAD THE BLOCK NUMBER
00725 14200001 909 ENI 1,X2 SPECIFY A SINGLE FILE BLOCK
00726 00700663 X 910 RTJ FREEBLK FREE THE BLOCK
00727 20300001 911 LDA 1,X3 LOAD THE COUNT OF BLOCKS IN THE
00730 15477776 912 INA,S -1 FILE AND DECREASE IT BY ONE
00731 05400002 913 ASG,S 2 SKIP IF TWO OR MORE LEFT
00732 01000760 P 914 UJP FREEEXIT ONE BLOCK ONLY CAN BE FREED
915 * WITH NO I/O TRANSFERS
916 STA 1,X3 STORE AWAY THE COUNT
00733 40300001 917 ENA,S -1
00734 14477776 918 ADA COUNT COUNT DOWN THE TOTAL COUNT
00735 30001355 P 919 STA COUNT STORE IT BACK
00736 40001355 P 920 LDQ MSQTOTAL GET THE AMOUNT OF UNUSED SPACE
00737 21077777 X 921 ENI 0,X2 SET FREEING PRIORITY TO ZERO
00740 14200000 922 FREPANIC ASG,S IMPURE SKIP IF 10% OF THE STORAGE NEEDS
00741 05400000 923 AQJ,LT *+2 TO BE FREED
00742 03700744 P 924 ENI FREEPR,X2 SET THE FREEING PRIORITY
00743 14200003 925 STI FREEPRD,X2
00744 47200024 P 926 PFA PFLOC+PFR
00745 77650001 927 * GET THE 1/4 PAGE NUMBER OF THE
928 SHAQ -15 CORE WE ARE USING
00746 13077760 929 LDA CORE FORM A CORE ADDRESS IN (Q)
00747 20004000 930 ASG,S 1 GET THE FORWARD POINTER
00750 05400001 931 HLT *
00751 00000751 P 932 STA 2,X3 PUT IT INTO THE QUEUE ELEMENT
00752 40300002 933 ENI 2,X1 READ IN THE POINTER WORDS
00753 14100002 934 ENI READ,X2
00754 14200104 X 935 ENI DONE,X3 INTERRUPT TO DONE WHEN THE I/O
00755 14300716 P 936 ANQ,S 770008 FORM AN 18 BIT CORE ADDRESS
00756 17577000 937 UJP FREEP1 GO INITIATE THE TRANSFER
00757 01000026 P 938

```

00760	20004000		939	FREEEXIT	LDA	CORE	GET THE ADDRESS OF THE LAST BLOCK OF THE FILE
00761	14200001		940	*		1,X2	SPECIFY A SINGLE FILE BLOCK
00762	00700726	X	941		ENI	FREEBLK	FREE THE LAST BLOCK
00763	14477775		942		RTJ	-2	
00764	34001355	P	943	FORGETX	ENA,S	COUNT	
00765	20300000		944		RAD	0,X3	GET ADDRESS OF NEXT QUEUE ELEMENT
00766	44000717	P	945		LDA	FREEQB	ADVANCE THE QUEUE POINTER
00767	53300000		946		SWA	X3	
00770	14300002		947		TIA	2,X3	
00771	00700600	X	948		ENI	FREEEM	FREE THE QUEUE ELEMENT
00772	77650001		949		RTJ	PFLOC+PFR	1/4 PAGE NUMBER TO A
00773	13077760		950		PFA	-15	FORM A CORE ADDRESS IN Q
00774	20001355	P	951		SHAQ	COUNT	
00775	54300025	P	952		LDA	FREE,X3	LOAD THE RETURN ADDRESS
00776	03101015	P	953		LDI	FREELOAD	JUMP IF MORE BLOCKS
00777	42077777	X	954	17777 3	AZJ,NE	IOBUSY	CLEAR THE BUSY FLAG
01000	14600717	P	955		SACH	FREEQB	
01001	44001356	P	956		ENA	FREEFLAG	ADJUST THE END POINTER
01002	01000632	P	957		SWA	GIVBUFFX	GIVE UP THE FILE CORE BLOCK
			958		UJP		
01003	47200025	P	959			FREE,X2	SAVE THE RETURN ADDRESS
01004	01001014	P	960		FREECI	FREECI	
01005	14600070		961		UJP	70B	SAY WE ARE FREEING
01006	42000777	X	962	00177 3	FREESTR	IOBUSY	
01007	14701003	P	963		ENA	FREECI	SET THE INTERRUPT ADDRESS
01010	14601012	P	964		SACH	*+2	ENTER THE IMMEDIATE RETURN
01011	01000431	P	965		ENQ	GETBUFF	GO GET A FILE CORE BLOCK
01012	03300567	P	966		ENA	RX3	EXIT IF THE CALL WAS QUEUED
01013	47300025	P	967		UJP	FREE,X3	SAVE THE RETURN ADDRESS
01014	13077760		968		AZJ,LT	-15	FORM AN 18 BIT CORE ADDRESS IN Q
01015	54300717	P	969		STI	FREEQB,X3	GET THE POINTER TO THE QUEUE ELEM
01016	20300002		970		SHAQ	2,X3	
01017	01000753	P	971		FREELOAD	FREEINIT	
			972		LDA		
			973		UJP		
01020	14100004		974		FORGET	4,X1	PRINT 5 DIGITS
01021	47100722	P	975		ENI	FREEIR,X1	CLEAR THE ERROR FLAG
01022	20300001		976		STI	1,X3	
01023	13077747		977		LDA	-24	
01024	51077777	X	978		SHAQ	D10	
01025	43405734	P	979	01367 0	DVA	LOSTMSG,X1	
01026	02501023	P	980		SQCH	*-3,X1	
01027	11005734	P	981	01367 0	IJD	LOSTMSG	
01030	14700027		982		ECHA	23	ENTER CHARACTER COUNT
01031	14201033	P	983		ENQ	*+2,X2	
01032	01077777	X	984		ENI	OPMSG	
01033	24300001		985		UJP	1,X3	
01034	01000764	P	986		LCA	FORGETX	
			987		UJP		



```

989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *

```

CONVERT  
 DINT  
 LDQ 18 BIT CORE ADDRESS  
 RTJ CONVERT  
 THE ROUTINE RETURNS WITH X2 POINTING TO THE FCLIST ELEMENT  
 THAT CORRESPONDS TO THE CORE ADDRESS AND THE CORE ADDRESS  
 IS MASKED TO JUST THE 1/4 PAGE BITS

```

1001
1002
1003 CONVERTD SHQ 18 FORM 18 CORE ADDRESS IN (Q)
1004 CONVERT UJP IMPURE
1005 DINTC
1006 ENA 0
1007 ANQ,S 77000B GET JUST THE 1/4 PAGE ADDRESS
1008 SHAQ 24+6
1009 SHAQ -17 CONVERT TO A PAGE NUMBER
1010 TAI X2
1011 SHAQ -7 1/4 PAGE NUMBER TO HIGH (Q)
1012 LDA PAGETABL,X2 LOAD FCLIST PCINTER
1013 CONVERT X EQU *
1014 TAI X2 POINTER TO X2
1015 ECHA,S 300000B CHECK FOR THE PROPER 1/4 PAGE
1016 LPA FCA,X2
1017 AQJ,EQ CONVERTD JUMP IF IT IS
1018 LDA 0,X2 POINT TO THE NEXT FCLIST ELEMENT
1019 UJP CONVERTX AND LOOP BACK

```

```

01035 12400022
01036 01000000
01037 77630000
01042 14600000
01043 17577000
01044 13000036
01045 13077756
01046 53600000
01047 13077770
01050 20277777 X
          01051 P
01051 53600000
01052 11700000 60000 0
01053 37200002
01054 03401035 P
01055 20200000
01056 01001051 P

```



```

1071 *
1072 * THIS ROUTINE GETS A BLOCK OF CORE
1073 *
1074 * CALL IT BY DISABLING THE INTERRUPTS AND EXECUTING AN
1075 *
1076 * RTJ GETCORE GET A CORE BLOCK
1077 *
*****
    
```

```

1079
1080
01123 00000000 1081 GETCORE VFD A9/JMP,A15/IMPURE
01124 77630000 1082 DINTC
01127 53020036 1083 LEVELC
01132 53100000 1084 TIA X1 CONTROL BLOCK ADDRESS TO A
01133 40377777 X 1085 STA I1,PSA SAVE IT IN THE USER'S INDEX 1
01134 53300000 1086 TIA PSA PSA POINTER TO A
01135 15677777 X 1087 INA F7 POINT TO THE F7 WORD
01136 40377777 X 1088 STA PC,PSA SAVE IT AS THE USERS PROGRAM
01137 14701146 P 1089 ENQ GETCOREX
01140 14601142 P 1090 ENA *+2
01141 01000431 P 1091 UJP GETBUFF
01142 03301157 P 1092 AZJ,LT SWEAT
01143 77640001 1093 APF PFLOC+PFW
01144 44377777 X 1094 SWA PF1,PSA
01145 01401123 P 1095 UJP,I GETCORE
1096
01146 44301144 X 1097 GETCOREX SWA PF1,PSA SAVE THE 1/4 PAGE NUMBER IN THE
01147 13077760 1098 SHAQ -15 PSA FORM THE CORE ADDRESS IN Q
01150 14477777 X 1099 ENA,S NMSWAIT
01151 34377777 X 1100 RAD IOBOUND,PSA
01152 47377777 X 1101 STI XFLAG,PSA
01153 14677777 X 1102 ENA SWBIT
01154 35000140 X 1103 SSA FLAGS
01155 40001154 X 1104 STA FLAGS
01156 01200000 1105 UJP Q,X2 RETURN
1106
01157 14677777 X 1107 SWEAT EQU *
01160 34301151 X 1108 ENA MNSWAIT
01161 20001123 P 1109 RAD IOBOUND,PSA
01162 40301135 X 1110 LDA GETCORE LOAD THE RETURN ADDRESS
01163 01077777 X 1111 STA F7,PSA STORE THE RETURN ADDRESS INTO F7
1112 UJP RMDONE GET THE HELL OUT OF HERE
    
```

```

1115 *
1116 * THIS ROUTINE IS TO READ A FILE BLOCK SPECIFIED BY THE C&P
1117 * WORD OF THE FILE CONTROL BLOCK POINTED TO BY THE CNBLK INDEX.
1118 * THE BLOCK IS READ INTO THE CORE BLOCK SPECIFIED BY PFLOC
1119 * CALL THE ROUTINE BY DISABLING THE INTERRUPTS AND EXECUTE AN
1120 *
1121 * RTJ FIX FIX THE CURRENT FILE BLOCK
1122 *

```

\*\*\*\*\*

```

1124
1125
01164 00000000 1126 FIX VFD A9/JMP,A15/IMPURE
01165 77630000 1127 DINTC
01170 53020036 1128 LEVELC
01173 20001164 P 1129 LDA FIX SAVE THE RETURN IN CASE THE CALL
01174 44377777 X 1130 SWA I2,PSA IS QUEUED IN GETCORE
01175 00701123 P 1131 RTJ GETCORE
01176 20301174 X 1132 LDA I2,PSA GET THE RETURN ADDRESS
01177 40301162 X 1133 STA F7,PSA SAVE IT IN THE F7 WROD IN THE PSA
01200 77650001 1134 PFA PFLOC+PFR READ THE FILE CORE BLOCK NUMBER
01201 13077760 1135 SHAQ -15 SAVE THE 18-BIT ADDRESS IN Q
01202 00701036 P 1136 RTJ CONVERT FCLIST ELEMENT ADDRESS TO INDEX 2
01203 53300000 1137 TIA PSA GET THE ADDRESS OF THE PSA
01204 44200002 1138 SWA FCA,X2 SAVE IN THE FCLIST ELEMENT
01205 14601157 X 1139 ENA MSWAIT
01206 34301160 X 1140 RAD IOBOUND,PSA
01207 20100003 1141 LDA CBP,CNBLK LOAD THE FILE BLOCK NUMBER
01210 40377777 X 1142 STA A,PSA SAVE IT IN THE PSA ALSO
01211 14200754 X 1143 ENI READ,X2
01212 14301217 P 1144 ENI FIXED,X3 ENTER THE RETURN ADDRESS
01213 14101000 1145 ENI WPF3+2,X1 ENTER THE WORD COUNT
01214 00700013 P 1146 RTJ FEAR CALL THE HIGHER PRIORITY ROUTINE
01215 01001163 X 1147 UJP RMDONE GO TERMINATE THE USER TEMPORARILY
1148
01216 47001235 P 1149 STI IRFLAG,0 IRRECOVERABLE I/O ERROR
01217 00701036 P 1150 RTJ CONVERT FCLIST ELEMENT ADDRESS TO INDEX 2
01220 20200002 1151 LDA FCA,X2 GET THE PSA ADDRESS OF THE PROPER
01221 53500000 1152 TAI X1 USER PUT IN INDEX 1
01222 12000011 1153 SHA 9 GET THE 1/4 PAGE NUMBER OF THE
01223 17600777 1154 ANA 777B FILE CORE BLOCK
01224 44101146 X 1155 SWA PF1,X1 AND SAVE IT IN THE PSA
01225 20101210 X 1156 LDA A,X1 LOAD THE FILE BLOCK NUMBER
01226 40200003 X 1157 STA FCB,X2 AND SAVE IT IN THE FCLIST ELEMENT
01227 14401150 X 1158 ENA,S NMSWAIT
01230 34101206 X 1159 RAD IOBOUND,X1 CLEAR MSWAIT
01231 14601153 X 1160 ENA SWBIT INITIATE SWITCHING
01232 35001155 X 1161 SSA FLAGS
01233 40001232 X 1162 STA FLAGS
01234 47101152 X 1163 STI XFLAG,X1 INDICATE SWITCHING PREFERENCE
01235 04000077 1164 IRFLAG ISE 77B+IMPURE,0
01236 01300000 1165 UJP 0,X3 AND RETURN
01237 14677777 X 1166 ENA READRTN CHANGE THE COMPLETION ADDRESS
01240 40101177 X 1167 STA F7,X1
01241 20177777 X 1168 LDA SELECT,X1 GET THE CONTROL BLOCK POINTER
01242 53500000 1169 TAI CNBLK
01243 20000572 X 1170 LDA AOB LOAD THE ABNORMAL/UNAVAILABLE BIT
01244 35100004 1171 SSA CPP,CNBLK SET IT INTO THE STATUS
01245 40100004 1172 STA CPP,CNBLK
01246 14600077 1173 ENA 77B
01247 44001235 P 1174 SWA IRFLAG SAY NO IRRECOVERABLE ERROR
01250 14600000 1175 ENA 0
01251 40100002 1176 STA COREP,CNBLK CLEAR THE USERS CORE POINTER
01252 01000632 P 1177 UJP GIVBUFFX GIVE UP THE FILE CORE BLOCK

```

```

1180 *
1181 *
1182 * THIS ROUTINE IS TO RESERVE A BLOCK OF CCRE WHICH WAS
1183 * PREVIOUSLY FLOATED BUT HAS NOT BEEN REWRITTEN
1184 * ENTER WITH THE INTERRUPTS DISABLED AND THE CCREP WORD WHICH
1185 * POINTS TO THE BLOCK IN THE A REGISTER BY EXECUTING AN
1186 * RTJ RESERVE RESERVE THE CCRE BLOCK
1187 *
1188 * INDEX 2 IS SET EQUAL TO THE ADDRESS OF THE CCRE BLOCK
1189 *
*****
    
```

```

1191
1192
01253 01000000
01254 12077766
01255 17600777
01256 77640001
01257 13077760
U1260 53020036
01263 00701036 P
01264 20000652 X
01265 35200000
01266 40200000
01267 01001253 P
1193 RESERVE UJP IMPURE
1194 SHA -9 QUARTER PAGE NUMBER TO RIGHT OF A
1195 ANA 7778 MASK OFF ANY GARBAGE
1196 APF PFLOC+PFW LOAD THE PAGE FILE WORD
1197 SHAQ -15 BUFFER ADDRESS TO (Q)
1198 LEVELC ( )
1199 RTJ CONVERT FCLIST ELEMENT ADDRESS TO X2
1200 LDA TOP
1201 SSA FCS,X2 TELL THE WORLD WE ARE USING THIS
1202 STA FCS,X2 BLOCK AGAIN
1203 UJP RESERVE RETURN
    
```

```

1206 *
1207 * THIS ROUTINE REWRITES THE BLOCK OF CORE WHOSE CORE POINTER
1208 * IS IN THE A REGISTER INTO THE FILE BLOCK WHOSE BLOCK NUMBER
1209 * IS IN THE Q REGISTER.
1210 * ALL INDEX REGISTERS ARE SAVED AND RESTORED BY THIS ROUTINE
1211 * TO CALL IT, DISABLE THE INTERRUPTS AND EXECUTE AN
1212 *
1213 * RTJ REWRITE REWRITE AND RELEASE THE BLOCK
1214 *
*****
    
```

```

1216
1217
01270 41001357 P 1218 RWRTX STQ TEMP1 SAVE THE BLOCK NUMBER
01271 13000030 1219 SHAQ 24 CORE ADDRESS TO Q
01272 14600001 1220 ENA 1 COUNT UP THE NUMBER OF BLOCKS
01273 34001353 P 1221 RAD RTNCOUNT BEING REWRITTEN
01274 00701036 P 1222 RTJ CONVERT GET THE FCLIST PCINTER
01275 20001264 X 1223 LOA IOP TELL THE WORLD THAT THIS BLOCK
01276 35200000 1224 SSA FCS,X2 IS BEING REWRITTEN
01277 40200000 1225 STA FCS,X2
01300 20001357 P 1226 LDA TEMP1 GET THE BLOCK NUMBER
01301 40200003 1227 STA FCB,X2 SAVE IT IN THE FCLIST ELEMENT
01302 14101000 1228 ENI WPFB+2,X1
01303 14277777 X 1229 ENI WRITE,X2
01304 14300545 P 1230 ENI RTN,X3 CALL RTN WHEN REWRITTEN
01305 00700001 P 1231 RTJ FILE INITIATE THE TRANSFER
01306 14100000 1232 RWRT1 ENI IMPURE,X1 RESTORE INDEX 1
01307 14200000 1233 RWRT2 ENI IMPURE,X2 RESTORE INDEX 2
01310 14300000 1234 RWRT3 ENI IMPURE,X3 RESTORE INDEX 3
01311 14600000 1235 ENA 0 TELL THE CONTROL BLOCK THAT IT NO
01312 40100002 1236 STA COREP,X1 LONGER HAS THE FILE CORE BLOCK
01313 01000000 1237 REWRITE UJP IMPURE
01314 03001313 P 1238 AZJ,EQ JUMP IF ALREADY REWRITTEN
01315 47101306 P 1239 STI RWRT1,X1 SAVE INDEX 1
01316 47201307 P 1240 STI RWRT2,X2 SAVE INDEX 2
01317 47301310 P 1241 STI RWRT3,X3 SAVE INDEX 3
01320 03301270 P 1242 AZJ,LT JUMP IF IT MUST BE REWRITTEN
01321 13000030 1243 SHAQ 24 CORE ADDRESS TO THE Q REGISTER
01322 14301306 P 1244 ENI RWRT1,X3 FORGET ABOUT WRITING IT OUT
01323 01000641 P 1245 UJP GIVBUFF GIVE UP THE FILE CORE BLOCK
1246
1247
    
```

```

1249 *
1250 * ALTERNATE ENTRY WHICH PERFORMS IMPORTANT BOOKKEEPING
1251 * RTJ REWRTX REWRITE THE RESERVED BLOCK
1252 *
*****
    
```

```

1254
1255
01324 01000000 1256 REWRTX UJP IMPURE
01325 21100003 1257 LDQ CBP,CNBLK GET THE CURRENT BLOCK POINTER
01326 40100003 1258 STA CBP,CNBLK STORE THE NEXT BLOCK POINTER
01327 77630000 1259 DINTC
01332 53020036 1260 LEVELC ( )
01335 20001275 X 1261 LDA BIT23
01336 37100002 1262 LPA COREP,CNBLK GET JUST THE CHANGED BIT
01337 40100002 1263 STA COREP,CNBLK STORE IT BACK
01340 77650001 1264 PFA PFLOC+PFR GET THE ADDRESS FROM THE PAGE FIL
01341 12000011 1265 SHA 9 FORM THE 1/4 PAGE ADDRESS
01342 35100002 1266 SSA COREP,CNBLK GET THE CHANGED BIT
01343 00701313 P 1267 RTJ REWRITE
01344 01001324 P 1268 UJP REWRTX RETURN
    
```

01345	00000000	1270	MXAG	VFD	A24/IMPURE,A24/IMPURE
01347	00001347	1271	MSCPTR	VFD	A9/IMPURE,A15/*+IMPURE
01350	00000000	1272	MXSLIST	00	IMPURE POINTER TO SUBSTITUTION LIST
01351	00001351	1273	CWAITQ	VFD	A9/IMPURE,A15/*+IMPURE
01352	00000000	1274	FCLIST	VFD	A9/000,A15/IMPURE
01353	00000000	1275	RTNCCOUNT	VFD	A24/IMPURE NUMBER OF BLOCKS BEING REWRITTEN
01354	00000000	1276	LCWVALFC	VFD	A9/0,A15/IMPURE
01355	00000000	1277	COUNT	VFD	A24/IMPURE TOTAL NUMBER OF FILE BLOCKS WHICH REMAIN TO BE FREED
		1278	*		
01356	00000717	1279	FREEFLAG	VFD	A6/IMPURE,03/0,A15/FREEQB+IMPURE
01357	00000000	1280	TEMP1	VFD	A24/IMPURE,A24/IMPURE
	01360 P	1281	TEMP2	EQU	TEMP1+1
01361	00000000	1282	TEMP3	VFD	A24/IMPURE,09/000,A15/IMPURE
	01362 P	1283	TEMP4	EQU	TEMP3+1
01363	00000000	1284	TEMP5	VFD	A6/IMPURE,03/0,A15/IMPURE
	05714 P	1285	CIRTEMP	EQU,C	TEMP5 1 CHARACTER OF TEMPORARY STORAGE
01364	00000000	1286	TEMP7	VFD	09/000,A15/IMPURE
01365	00000000	1287	TEMP8	VFD	A24/IMPURE
01366	00000000	1288	TEMP9	VFD	09/000,A15/IMPURE
01367	67676767	1289	LOSTMSG	BCD,C	23,XXXXX FILE BLOCKS LOST^
01374	62637767	1290		IF	DEBUG,BADBLOCK BCD,C 22,XXXXXXXXX BAD BLOCK NO^
	00026	1291		IF	DEBUG,BADBLK EQU,C *-BADBLOCK
		1292			
		1293		END	

NO LINES WITH ERRORS

A	X		65	1142	01210P	1156	01225P				
* ACCWORD		00000	71								
ALT	X		153	725	00531P	1056	01107P				
ARRAY3L	X		66	446	00243P						
AUB	X		137	1170	01243P						
BAD3LKL		00026	1291	507	00323P						
BAD3LOCK		01374P	1290	1291	01402P	504	00320P	506	00322P		
BIT15	X		67	153	00000P						
BIT16	X		68	152	00000P						
BIT17	X		69	137	00000P	348	00143P	614	00416P	700	00503P
BIT23	X		70	151	00000P	568	00362P	1261	01335P	772	00572P
* BLKR		00005	88								
3LOCKTBL	X		71	434	00231P						
BT201918	X		72	540	00327P						
CBP		00003	76	1141	01207P	1257	01325P	1258	01326P		
CIRTEMP		01363P	1285	286	00051P	392	00211P				
CLOCK		00022	140	686	00466P	764	00564P	1045	01075P		
CNBLK		00001	133	724	00530P	727	00533P	1035	01057P	1048	01100P
				1141	01207P	1169	01242P	1171	01244P	1172	01245P
				1258	01326P	1262	01336P	1263	01337P	1266	01342P
CONVERT		01036P	1004	326	00116P	752	00552P	831	00635P	837	00641P
				1150	01217P	1199	01263P	1222	01274P	1044	01074P
CONVERTD		01035P	1003	1017	01054P					1050	01102P
CONVERTX		01051P	1013	1019	01056P					1054	01106P
CORE		04000	142	929	00747P	939	00760P			1050	01057P
COREP		00002	73	76	00000P	323	00114P	727	00533P	730	00535P
				1054	01106P	1176	01251P	1236	01312P	1262	01336P
COUNT		01355P	1277	885	00677P	918	00735P	919	00736P	944	00764P
CPP		00004	77	1171	01244P	1172	01245P			952	00774P
* CPPX		00002	134								
CWAITQ		01351P	1273	701	00504P	759	00561P	760	00562P	774	00574P
				791	00615P	1063	01116P	1065	01120P	775	00575P
D10	X		73	978	01024P					790	00614P
DEBUG		00001	24	7	00037P	435	00232P	436	00233P	499	00315P
				7	00632P	7	01037P	7	01063P	16	01066P
				7	01165P	16	01170P	16	01260P	7	01124P
				1291	01402P					16	01332P
DINT		07773	117	612	00414P	876	00666P			7	00432P
DKONTAB	X		74	382	00201P	384	00203P			7	00547P
DKINT	X		75	389	00210P					16	01127P
DKSTATAB	X		76	377	00174P	380	00177P			16	01374P
DKSTATBL	X		77	376	00173P						
DONE		00716P	902	935	00755P						
EINT		07774	118	607	00407P						
* EPP		00006	90								
F7	X		78	1087	01135P	1111	01162P	1133	01177P	1167	01240P
F3PP	X		79	561	00354P						
FCA		00002	157	160	00000P	320	00111P	329	00121P	330	00122P
				714	00520P	723	00527P	1016	01053P	1049	01101P
FCB		00003	160	312	00101P	328	00120P	359	00154P	673	00452P
				833	00637P	1157	01226P	1227	01301P	1138	01204P
FCLIST	E	01352P	1274	27	00000P	300	00067P	663	00441P	717	00523P
FCS		00000	150	314	00103P	331	00123P	333	00125P	334	00126P
				666	00443P	681	00462P	707	00511P	712	00516P
				756	00556P	782	00604P	846	00651P	848	00653P
				1201	01265P	1202	01266P	1224	01276P	1225	01277P
FCT		00001	155	676	00455P	766	00566P	1047	01077P	1057	01110P
FCV01		00443P	666	684	00465P						
FCV02		00462P	681	669	00446P	678	00457P				
FCV03		00463P	682	667	00444P						
FCV04		00464P	683	664	00442P						
FCV06		00473P	691	675	00454P						
FEAR		00013P	244	246	00015P	1146	01214P				
FEARPR		00004	192	243	00012P						
FILE	E	00001P	232	28	00000P	234	00003P	1231	01305P		
FILEPR		00002	194	231	00000P						
FINK	E	00020P	250	29	00000P	252	00022P				
FINKP1	E	00021P	251	30	00000P						
FINKPR		00012	189	249	00017P						
FIRE	E	00006P	238	31	00000P	240	00010P				
FIREP1	E	00007P	239	32	00000P						
FIREPR		00010	190	237	00005P						
FIX	E	01164P	1126	33	00000P	1129	01173P				
FIXED		01217P	1150	1144	01212P						
FLAGS	X		80	344	00137P	345	00140P	1103	01154P	1104	01155P
FLOAT	E	01062P	1038	34	00000P					1161	01232P
FLOATX		01057P	1035	1062	01115P	1066	01121P			1162	01213P
FLOOR		01061P	1037	1041	01071P	1068	01122P				
FORGET		01020P	974	907	00723P						



FORGETX		00764P	944	986	01034P								
FREE		00025P	256	258	00027P	902	00716P	953	00775P	960	01003P	968	01013P
FREEBLK	X		81	873	00663P	910	00726P	942	00762P				
FREECI		01003P	960	964	01007P								
FREECIC		01014P	969	961	01004P								
FREEFILE	E	00665P	875	35	00000P	878	00670P						
FREEFLAG		01356P	1279	892	00705P	893	00706P	957	01001P				
FREEINIT		00753P	933	972	01017P								
FREEIR		00722P	906	901	00715P	975	01021P						
FREELOAD		01015P	970	954	00776P								
FREEEMEM	X		82	620	00424P	778	00600P	949	00771P				
FREEP1		00026P	257	937	00757P								
FREEPR		00003	193	255	00024P	924	00743P						
FREEPRWD		00024P	255	925	00744P								
FREEQ3		00717P	903	946	00766P	956	01000P	970	01015P	1279	01356P		
FREESTRT		01005P	962	896	00711P								
FREEEX1		00712P	897	882	00674P								
FREEEX2		00664P	874	879	00671P	899	00714P						
FREEEX3		00713P	874	898	00675P								
FREEEXIT		00760P	939	914	00732P								
FREEZIP		00662P	872	881	00673P								
FREPANIC	E	00741P	922	36	00000P								
GET3LK	X		83	563	00356P								
GET3UFF	E	00431P	656	37	00000P	797	00623P	966	01011P	1091	01141P		
GETCORE	E	01123P	1081	38	00000P	1095	01145P	1110	01161P	1131	01175P		
GETCOREX		01146P	1097	1089	01137P								
GETMEM	X		84	288	00053P	549	00340P	698	00501P	843	00646P	889	00702P
GIV3UF01		00642P	838	834	00640P								
GIV3UFF	E	00641P	836	39	00000P	1245	01323P						
GIV3UFFA		00631P	828	40	00000P								
GIV3UFFP		00630P	827	41	00000P								
GIV3UFFX		00632P	829	958	01002P	1177	01252P						
GTC05		00476P	695	732	00537P								
GTC06		00507P	704	696	00477P								
GTC07		00511P	706	694	00475P								
GTC08		00514P	710	728	00534P								
GTC09		00524P	718	705	00510P								
GTC10		00527P	722	709	00513P								
GTC11		00535P	730	726	00532P								
HOUR	X		85	687	00467P	765	00565P	1046	01076P				
I1	X		86	1085	01133P								
I2	X		87	1130	01174P	1132	01176P						
IMPURE		00000	119	232	00001P	238	00006P	244	00013P	250	00020P	255	00024P
				858	00661P	874	00664P	875	00665P	897	00712P	898	00713P
				906	00722P	922	00741P	1004	01036P	1037	01061P	1038	01062P
				1126	01164P	1164	01235P	1193	01253P	1232	01306P	1233	01307P
				1237	01313P	1256	01324P	1270	01345P	1270	01346P	1271	01347P
				1272	01350P	1273	01351P	1273	01351P	1274	01352P	1275	01353P
				1277	01355P	1279	01356P	1279	01356P	1280	01357P	1280	01360P
				1282	01362P	1284	01363P	1284	01363P	1286	01364P	1287	01365P
				42	00000P								
IMPURE05	E	00661P	858										
INTPOL	X		18	18	01130P	18	01171P	18	01261P	18	01333P	19	01067P
				19	01171P	19	01261P	19	01333P				
IOBOUND	X		88	1100	01151P	1109	01160P	1140	01206P	1159	01230P		
IOBUSY	X		89	894	00707P	955	00777P	963	01006P				
IOP	X		151	332	00124P	710	00514P	781	00603P	847	00652P	1200	01264P
IRFLAG		01235P	1164	1149	01216P	1174	01247P					1213	01275P
JMP		00000	122	1081	01123P	1126	01164P						
LOSTMSG		01367P	1289	979	01025P	981	01027P						
LOWVALFC		01354P	1276	680	00461P	690	00472P	691	00473P				
LP		00001	72										
MACHERR	X		90	747	00544P								
MSBIT	X		91	343	00136P								
MSC	E	00407P	606	43	00000P	623	00427P	786	00610P	794	00620P		
MSCPTR	E	01347P	1271	44	00000P	349	00144P	608	00410P	616	00420P	617	00421P
MSQTOIAL	X		92	920	00737P								
MSUNITS	X		93	433	00230P								
MSWAIT	X		94	1108	01157P	1139	01205P						
MXADD01		00234P	438	310	00100P								
MXADD01X		00245P	448	508	00324P								
MXADD02		00253P	456	465	00263P								
MXADD04		00264P	467	458	00254P								
MXADD06		00272P	474	481	00300P								
MXADD08		00301P	483	470	00266P								
MXADD09		00303P	487	480	00277P								
MXADD10		00304P	488	477	00274P								
MXADD12		00315P	500	435	00232P								
MXADDGEN		00214P	419	365	00162P	590	00406P						
MXAQ		01345P	1270	233	00002P	239	00007P	245	00014P	251	00021P	257	00026P
												266	0003P



