

```

1          ;***COPYRIGHT 1969, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.***
2
3
4          ;THIS SUB-PROGRAM ASSEMBLED WITH SYSTEM PARAMETER FILE - S,MAC(V414)
5              XLIST
6              LIST
7          ;THIS SUB-PROGRAM ASSEMBLED WITH CONFIGURATION DEPENDENT FEATURE SWITCHES - FT50SB,MAC(
8          V003)
9              XLIST
10             LIST
11          TITLE CORE1 - LOGICAL AND PHYSICAL CORE ALLOCATION ROUTINES - V414
12          SUBTTL I, HASTINGS/TH/RCC IS 04 JUN 69
13          XP VCORE1,414*
14
15             ;THIS MACRO PUTS VERSION NO. IN STORAGE MAP AND GLOB
16
17             ENTRY CORE1      ;ALWAYS LOAD CORE1(FOR LIB SEARCH)
18
19
20          ;CORE ALLOCATION IS DONE ON A 1K BLOCK BASIS
21
22          ;USING A USE BIT TABLE(CORTAB) WHICH HAS A 1
23          ;FOR EVERY BLOCK WHICH IS NOT AVAILABLE BECAUSE:
24          ; 1.IN USE BY MONITOR
25          ; 2.IN USE BY USER
26          ; 3.NON-EXISTANT MEMORY
27          ;0 MEANS BLOCK IS NOT IN USE
28
29          ;WHEN THE SYSTEM IS STARTED, SYSINI SETS THE CORTAB TABLE
30          ;IT ALSO SETS A BYTE POINTER(CORLST) WHICH POINTS TO THE
31          ;LOWEST NON-EXISTANT BLOCK IMMEDIATELY ABOVE THE HIGHEST
32          ;EXISTANT BLOCK. IT ALSO SETS CORTAL TO THE NO. OF
33          ;FREE BLOCKS AVAILABLE.
34          ;THE CORE1 ROUTINE ASSIGNS CORE IF POSSIBLE, SETS THE USE BITS,
35          ;AND MOVES THE JOB IF NEW ASSIGNMENT IS A DIFFERENT PLACE THAN OLD
36          ;THE JBTADR TABLE IS ALSO UPDATED BY THE CORE ROUTINES
37          ;LH=PROTECTION,RH=RELOCATION
38          ;JOBADR IS MODIFIED IF CORE FOR CURRENT JOB
39          ;HARDWARE RELOC. AND PROTEC. ARE RESET IF CURRENT JOB
40          ;FINALLY JOBRELOC(PROTECTION) IN JOB DATA AREA IS ALWAYS UPDATED
41
42          ;LIST OF GLOBALS AFFECTED:
43          ;JBTADR,CORTAL,CORTAR,HOLEF,SHEWAT,JOBADR
44
45          ;ACS USED(RESIDES TAC,TAC1,JPAT,IQS,DEVDAT,AND POP)
46          000012 BLK=BUFPT      ;HIGHEST REL. ADR. IN USER AREA
47          000013 LOC=BUFWRD     ;ABS. LOC. OF FIRST BLOCK IN USER AREA
48          000015 T=AC1         ;TEMPORARY
49          000016 T1=AC2        ;"

```

```

48          ;CORE UUC
49          ;CALL: MOVEI AC,HIGHEST REL, ADR, DESIRED IN LOW SEG
50          ;       WRLI AC,HIGHEST REL, ADR, DESIRED IN HIGH SEG
51          ;       CALL AC,(SIXBIT /CORE/)
52          ;       FRROR RETURN
53          ;       OK RETURN TO USER, JOB MOVED IF NECESSARY
54          ;RETURN NO. OF FREE 1K BLOCKS IN AC(OR MAX. NO. BLOCKS ALLOWED IF SWAPPING SYS)
55          ;BOTH HALVES 0 MEANS ERROR RETURN NO. OF FREE 1K BLOCKS(OR MAX. NO. OF BLOCKS
56          ; ALLOWED IF SWAPPING SYSTEM) IMMEDIATELY WITHOUT AFFECTING CORE
57          ; OR WAITING FOR IO DEVICES
58          ;LH=0 MEANS DO NOT CHANGE HIGH SEG ASSIGNMENT
59          ;RH=0 MEANS DO NOT CHANGE LOW SEG ASSIGNMENT
60
61          INTERNAL CORUUC
62          EXTERNAL USRREL,JOB,CORTAL
63          EXTERNAL ESTOPL,IOWAIT,OERROR,SETREL,STOTAC,WSCHED
64
65 000000 322040 000016' CORUUC: JUMPE TAC,ZERCOR          ;IS HE ASKING FOR ZERO CORE?
66 000001 261140 000001          PUSH PDP,TAC          ;NO, SAVE HIGHEST DESIRED ADDRESS IN BOTH SEGS
67 000002 262140 000000          PUSHJ PDP,IOWAIT       ;WAIT FOR ALL DEVICE INACTIVE
68 000003 550043 000000          HRRZ TAC,(PDP)        ;HIGHEST REL. LOC. DESIRED FOR LOW SEG(RH)
69 000004 322040 000010'          JUMPE TAC,CORU1       ;IS RH 0(IF YES DO NOT CHANGE LOW SEG)?
70 000005 435040 001777          IORI TAC,1777         ;NO, MAKE IT AN EVEN MULTIPLE OF 1K-1
71 000006 260140 000126'          PUSHJ PDP,CORE1       ;TRY TO ASSIGN CORE
72 000007 254000 000014'          JRST COREERR        ;NOT AVAILABLE, ERROR RETURN
73
74 000010 554043 000000 CORU1: HLRZ TAC,(PDP)          ;SUM OF NEW LOW AND OLD HIGH SEG TOO BIG
75          IFB FT2REL,<          ;CHECK TO SEE IF USER IS REQUESTING HIGH CORE
76          CAMG TAC,USRREL      ;0 MEANS NO CHANGE, 1 THRU TOP OF LOW SEG MEANS
77          ; RETURN HIGH CORE (IF ANY), ALWAYS LEGAL
78          ; EVEN IF NO HIGH SEG SHARP WARE OR SOFTWARE
79
80          >
81          IFB FT2REL,<
82          EXTERN UCORHI
83 000011 260140 000000          PUSHJ PDP,UCORHI       ;TRY TO ASSIGN CORE FOR HIGH SEG,
84          ; UCORHI EXPECTS ARG ON PD LIST
85 000012 254000 000014'          JRST COREERR        ;ERROR-ACTIVE IO(SAVE IN PROGRESS FOR SOME USER)
86          ; OR SUM OF NEW LOW SEG AND NEW HIGH SEG TOO BIG
87
88          >
89          AOS -1(PDP)          ;SET FOR OK(SKIP) RETURN
90          CORERR: POP PDP,TAC   ;REMOVE ARG FROM LIST
91          IFB FTSWAP,<
92          PUSHJ PDP,WSCHED     ;CALL SCHEDULER TO STOP JOB
93          ; IN CASE LOW SEG MUST BE SWAPPED OUT TO EXPAND
94          ; OR HIGH SEG LOGICAL CORE ASSIGNED ON DISK
95          ; SAVE ALL ACS EXCEPT AC1,AC2,AC3.
96
97 000016          FERCOR:
98          IFB FTSWAP,<
99          MOVE TAC,CORTAL     ;RETURN NO. OF FREE 1K BLOCKS (COUNTING
100          ; DORMANT AND IDLE SEGMENTS AS FREE)
101          >
102          IFB FTSWAP,<

```

CORE1 - LOGICAL AND PHYSICAL CORE ALLOCATION ROUTINES - V414
T. HASTINGS/TH/RCC TS 04 JUN 69

MACRO,V36 19:03 4-JUN-69 PAGE 14-1

101	000016	200040	000000	MOVE TAC,CORMAX	;RETURN MAX, NO, 1K BLOCKS ALLOWED FOR 1 USER TO HAVE
102	000017	242040	.777766	LSH TAC,-12	
103				>	
104	000020	254000	000000	JRST STOTAC	;STORE IN USER AC AND RETURN TO USER

```
175 ;ROUTINE TO CHECK JOBS TO SEE IF ANY JOB CAN BE SHUFFLED
176 ;IT IS CALLED EVERY 60TH OF A SECOND BY CLOCK ROUTINE
177 ;PROVIDING CURRENT JOB IS IN USEF MODE OR JUST ENTERING
178 ;IO WAIT OR SHARABLE DEVICE WAIT OR RETURNING ON UO CALLS
179 ;IE CURRENT JOB AND ALL OTHER JOB ARE SHUFFABLE WITH RESPECT
110 ;TO RELOCATION INFO IN MONITOR
111 ;SINCE RESCHEDULING IS NEVER DONE WHEN CURRENT JOB
112 ;IS NOT SHUFFABLE, ALL JOBS ARE SHUFFABLE WHEN
113 ;CHKSHF IS CALLED,
114 ;THE NOSHUFFLE MACRO SHOULD STILL BE USED WHEN EVER
115 ;THE MONITOR MOVES RELOCATION OUT OF ACS PDP, PROG, OR JDAT
116 ;IN CASE IT SHOULD PROVE DESIRABLE IN THE FUTURE TO SHUFFLE
117 ;MORE FREQUENTLY THAN AT THE ABOVE TIMES
118 ;FINALLY A JOB MUST HAVE ALL DEVICES INACTIVE(SINCE SOME
119 ;OF THEM USE ABSOLUTE ADDRESSES)BEFORE IT CAN BE MOVED
120 ;SO CORE CANNOT BE REASSIGNED WHILE DEVICES ARE ACTIVE
121 ;IF DEVICES ARE ACTIVE, JOB WILL BE STOPPED SO THAT IO WILL
122 ;CEASE SOON SO JOB CAN BE SHUFFLED
123 ;ALL DEVICES LOOK AT SHF BIT IN JBTSTS(ADVBEF OR ADVBFE)
124 ;TO SEE IF MONITOR IS WAITING TO SHUFFLE JOB
125 ;THE NSHF BIT IN JBTSTS SHOULD BE SET FOR JOBS USING DISPLAYS
126 ;SINCE DISSER CONTINUALLY REFERENCES USER AREA EVEN THOUGH
127 ;IOACT IS OFF,
128
```

```

129
130
131
132
133
134
135
136
137
138
139
140
141 000021 332200 000000 CHKSHF: SKIPE ITEM,SHFWAT ;HAS CHKSHF STOPPED A JOB AND IS WAITING FOR
142 ; DEVICES TO BECOME INACTIVE?
143 000022 254000 000034' JRST SHFLOP ;YES, SEE IF IO HAS STOPPED YET
144 000023 336040 000000 SKIPN TAC,HOLEF ;NO, DOES CORE HAVE A HOLE IN IT?
145 000024 263140 000000 POPJ PDP, ;NO
146 000025 201200 000000 MOVEI ITEM,JBTMAX ;SEARCH FOR JOB OR HIGH SEG JUST ABOVE HOLE
147 000026 550104 000000 HOLL0P: HRRZ TAC1,JBTADR(ITEM) ;ADR. OF JOB

148 000027 312100 000001 CAME TAC1,TAC
149 000030 367200 000026' SOJG ITEM,HOLL0P ;JOB ABOVE HOLE KEEP LOOKG
150 000031 327200 000034' JUMPG ITEM,SHFLOP ;FOUND ONE?
151 000032 402000 000023' SETZM HOLEF ;NO, CLEAR HOLEF OR ELSE ERROR WILL BE PRINTED EVERY
152 ; CLOCK TICK AND ERROR MESSAGE WILL NEVER PRINT
153 000033 265240 000000 JSP DAT,0ERROR ;SYSTEM ERROR(TELL OPERATOR)
154
155 000034 332344 000026' SHFLOP: SKIPE PR0G,JBTADR(ITEM) ;JOB ABOVE HOLE STILL IN CORE ?
156 000035 260140 000057' PUSHJ PDP,ANYACT ;NO ALL DEVICES FINISHED ?
157 000036 254000 000045' JRST NOTSHF ;YES, GO SEE IF HOLE STILL THERE
158 IFN JDAT-PR0G,<
159 MOVE JDAT,JBTDAT(ITEM) ;JOB DATA AREA
160 >
161 000037 554040 000007 HLRZ TAC,PR0G ;YES, REASSIGN SAME AMOUNT OF CORE,
162 000040 260140 000111' PUSHJ PDP,SCORE1 ;IN A LOWER POSITION IN CORE
163
164 000041 402000 000021' NOTSH1: SETZM SHFWAT ;JOB SHUFFLED, CLEAR FLAG
165 000042 205040 004000 MOVSI TAC,SHF ;CLEAR SHUFFLE WAIT BIT IN CASE IT WAS ON
166 000043 412044 000000 ANDCAM TAC,JRTSTS(ITEM)
167 000044 263140 000000 POPJ PDP,
168
169 ;JOB CANNOT BE MOVED BECAUSE IT HAS ACTIVE DEVICES OR NSHF BIT SET(DISPLAY,REALTIME)
170 ; SO JOB CANNOT BY SHUFFLED AT THIS TIME
171
172
173 000045 336000 000032' NOTSHF: SKIPN HOLEF ;IS HOLE STILL THERE?
174 000046 254000 000041' JRST NOTSH1 ;NO
175 IFN FTSWAP,<
176 EXTERNAL FIT,F0RCE
177 000047 214040 000000 MOVN TAC,F0RCE
178 000050 312200 000000 CAME ITEM,FIT
179 000051 316200 000001 CAMN ITEM,TAC
180 000052 254000 000041' JRST NOTSH1
181 >

```

CORE1 - LOGICAL AND PHYSICAL CORE ALLOCATION ROUTINES - V414
T. HASTINGS/TH/RCC TS 24 JUN 69

MACRO,V36 19:03 4-JUN-69 PAGE 16-1

1A2	000053	202200	000041'	MOVEM ITEM,SHFWAY	;SET SHUFFLE WAIT FLAG WITH JOB NO.
1A3	000054	205040	004000	MOVSI TAC,SHF	;SET SHF WAIT BIT IN JOB STATUS WORD
1A4	000055	436044	000043'	IORM TAC,JRTSTS(ITEM)	;SO JOB WILL NOT BE RUN
1A5	000056	263140	000000	POPJ PDP,	IAND IO WILL STOP SOON

```

186                                     ;ROUTINE TO TEST FOR ANY ACTIVE DEVICES
187
188                                     ;CALL:  MOVE ITEM, JOB NUMBER OR HIGH SEG NUMBER
189                                     ;        MOVE JDAT, ADDRESS OF JOB DATA AREA
190                                     ;        PUSHJ PDP, ANYACT
191                                     ;        DEVICES ACTIVE
192                                     ;        DEVICES NOT ACTIVE EXCEPT POSSIBLY TTY
193                                     ;IN SWAPPING SYSTEMS ANYACT IS BROKEN INTO 2 CALLS, TRYSWP AND ANYDEV
194                                     ;TRYSWP TESTS WHTETHER SWAPPER SHOULD EVEN CONSIDER SWAPPING JOB
195
196                                     INTERN ANYACT, ANYDEV
197                                     EXTERN JOBMAX
198
199 000057                               ANYACT:
200                                     IFN FTSWAP, <
201                                     INTERN ANYDEV
202 000057 260140 000104'                PUSHJ PDP, TRYSWP                ;SHOULD SWAPPER MAKE JOB UNRUNNABLE
203                                     ; IN ORDER TO SWAP OUT?
204 000060 263140 000000                POPJ PDP,                        ;NO, ACTIVE SAVE IN PROGRESS OR NSHF, NSWP SET
205                                     ; (REAL TIME OR DISPLAY)
206                                     >
207 000061 200644 000055'                ANYDEV: MOVE T, JRTSTS(ITEM)      ;IS JOB(OR HIGH SEG) NOT SHUFFABLE?
208 000062 603640 001000                TLNE T, NSHF                    ;DISPLAY AND REAL TIME SET NSHF
209 000063 263140 000000                POPJ PDP,                        ;CANNOT BE SHUFFLED
210                                     IFN FT2REL, <
211 000064 303200 000000                CAILE ITEM, JOBMAX              ;YES, IS THIS A HIGH SEG?
212 000065 254000 000000                JRST CPOPJ1                     ;YES, OK TO SHUFFLE OR SWAP SINCE NSHF, NSWP NOT SET
213                                     ; AND NO SAVE IN PROGRESS
214                                     >
215 000066 201647 000000                MOVEI T, JOBJDA(JDAT)           ;ASSUME JOB IS NOT CURRENT JOB
216 000067 316200 000000                CAMN ITEM, JOR                  ;IS IT?
217 000070 201640 000000                MOVEI T, USRJDA                 ;IT IS CURRENT JOB
218                                     ; DEVICE ASSIGNMENT TABLE IN MONITOR
219 000071 201000 010000                MOVEI IOS, IOACT                ;IO DEVICE ACTIVE BIT
220 000072 550715 000000                HRRZ T1, JOBJMH(T)             ;GET NO. OF USER IO CHANNELS IN USE
221                                     ; FOR JOR(EITHER FROM JOB DATA AREA
222                                     ; OR FROM MONITOR (IGNORE LH WHICH MAY BE-1
223                                     ; IF SAVGET IN PROGRESS)
224 000073 661640 000016                TLO T, T1                       ;SET TO ADD T TO T1
225 000074 205740 000010                MOVSI AC3, DVTTY ;DEVICE        ;IS A TTY BIT
226 000075 332320 000015                ANY: SKIPF DEVDAT, @T           ;IS A DEVICE ASSIGNED TO THIS CHANNEL?
227 000076 616006 000002                TDNN IOS, DEVIOS(DEVDAT)       ;YES, IS IT ACTIVE?
228 000077 365700 000075'                ANY2: SOJGE T1, ANY             ;NO, KEEP LOOKING
229 000100 321700 000065'                JUMPL T1, CPOPJ1               ;YES, FINISHED YET?
230 000101 616746 000004                TDNN AC3, DEVMOD(DEVDAT)       ;NOT FINISHED, IS DEVICE TTY?
231 000102 263140 000000                POPJ PDP,                       ;NO, ERROR RETURN, CANNOT ASSIGN CORE
232 000103 254000 000077'                JRST ANY2                       ;YES, KEEP LOOKING FOR AN ACTIVE DEVICE
    
```

```

233 ;ROUTINE TO TEST TO SEE IF JOB OR HIGH SEG CAN BE SWAPPED
234 ; OR WHETHER IT SHOULD BE ALLOWED TO CONTINUE RUNNING
235 ; UNTIL A MORE FAVORABLE TIME
236 ;CALL: MOVE ITEM,HIGH OR LOW SEG NUMBER
237 ; PUSHJ PDP,TRYSWP
238 ;
239 ; RETURN1 - JOB MUST REMAIN RUNABLE(NSHF,NSWP SET OR SAVE,GET IN PROGRESS)
240 ; RETURN2 - OK TO SWAP HIGH OR LOW SEG
241
242 IFN FTSWAP,<
243 INTERN TRYSWP
244 EXTERN JRTSTS,CPOPJ1
245 000104 200644 000061' TRYSWP: MOVE T,JRTSTS(ITEM) ;IS JOB OF HIGH SEG NOT SWAPPABLE?
246 000105 603640 011000 TLNE T,NSHF!NSWP ;OR SHUFFABLE(DISPLAY,REAL TIME)?
247 000106 263140 000000 POPJ PDP, ;YES, ERROR RETURN
248
249 IFN FT2REL,<
250 000107 254000 000000 EXTERN ANYSAV
251 JRST ANYSAV ;NO, SEE IF THIS JOB IS INVLOVED IN A SAV,GET
252 > ; WHICH IS STILL ACTIVE
253
254 IFE FT2REL,<
255 JRST CPOPJ1 ;NO, GIVE OK RETURN
256 >
    
```



```

257 ;ROUTINE TO FLUSH PHYSICAL CORE ASSIGNED IN MEMORY
258 ;NOTE: THIS ROUTINE DIFERS SIGNIFICANTLY FROM CORE0 AND CORE1 IN THAT
259 ;IT IS ONLY A PHYSICAL REMOVAL OF CORE(VIRTUAL IS NOT AFFECTED)
260 ;SEE COMMENTS FOR CORE1
261 ;CALL: MOVE ITEM,HIGH OR LOW SEGMENT NUMBER
262 ;      PUSHJ PDP,KCORE1
263 ;      ALWAYS RETURN HERE
264 ;SCORE1 IS CALLED FROM SHUFFLER WITH TAC SET TO SEG SIZE
265
266         INTERN KCORE1
267
268 000110 201040 000000 KCORE1: MOVEI TAC,0 ;SETUP DESIRED HIGHEST ADR
269 000111 370003 000000 SCOREF1: SOS (PDP) ;CORE1 WILL ALWAYS SKIP RETURN
270 000112 254000 000141' JRST CORE1A ;BYPASS LOGICAL CORE ASSIGNMENT PART
271 ; AND FLUSH PHYSICAL CORE(LOGICAL CORE UNEFFECTED)
272
273
274
275 ;CORE0 IS CALLED BY THE CORE MONITOR COMMAND AND THE CORE SHUFFLER
276
277 ;AND RUN COMMAND
278 ;BOTH LOGICAL AND PHYSICAL CORE ASSIGNMENT ARE AFFECTED
279
280 ;CALL: MOVE TAC,HIGHEST LEGAL ASSESSABLE LOC. DESIRED
281 ;      MOVE ITEM,JOB NUMBER
282 ;      MOVE PROG,CXWD PROT.,RELOC.]=JBTADR(ITEM)
283 ;      PUSHJ PDP,CORE0
284 ;      ERROR ;EITHER JOB HAS ACTIVE IO
285 ;           ; OR NOT ENOUGH CORE
286 ;      OK RETURN
287 ;JOB IS MOVED IF NECESSARY TO SATISFY REQUEST
288 ;PROG AND JDAT ARE SET TO NEW CORE ASSIGNMENT ON EITHER RETURN
289 ;0 MEANS NONE ASSIGNED IN MEMORY, ASSIGNED ON DISK
290 INTERNAL CORE0
291 INTERNAL CORE1,FTTIME,FTTRPSET,CORGET,FTSWAP
292 EXTERNAL JOBJDA,JOB,USRJDA,JOBADR,JBTADR
293 EXTERNAL JOBREL,JOBADR,JOBDAC,JORPC,JORDAT,JBTDAT
294 EXTERNAL CORTAL,CORLST,CORTAR,HOLEF,CLRWRD
295 EXTERNAL USRREL,CPOPJ1,JOBJMH,JOENB,JOBDPD,JOBDPG
296 EXTERNAL JOBPR1,CPOPJ1,JOBPRT,USRPC,CORMAX
297 ;ENTER HERE FROM CORE CONSOLE COMMAND OR INITIAL CORE
298 ;ASSIGNMENT OF JUST A JOB DATA AREA FOR RUN COMMAND
299 ;IE ENTER WHEN DEVICES MAY BE ACTIVE OR JOB MAY HAVE NO PREVIOUS CORE
300 ;JOB CAN HAVE CORE IN MEMORY, CORE ON DISK, OR NONE EITHER PLACE
301 ;JOB CANNOT BE IN PROCESS OF SWAP OUT OR SWAP IN(CALLER'S RESPONSIBILITY)
302 ;CORE0 NO LONGER REASSIGN CORE ON DISK IF OLD CORE ON DISK
303 ;BECAUSE OF FRAGMENTED SWAPPING(TOO HARD) UNLESS 0 BEING ASKED FOR
304 ;THEREFORE THE CORE COMMAND CAUSES JOB TO BE SWAPPED INTO CORE FIRST(INCORE=1)
305 ;HOWEVER, THE R,RUN,GET,KJOB COMMANDS DO NOT REQUIRE THE PREVIOUS CORE IMAGE TO
306 ;BE SWAPPED IN(CAS THIS IS SLOW), THEY ASK FOR 140 WORDS, AND LARGER DISK SPACE IS RELIN
307 QUTSHED
308 ;UPON SWAPIN BY THE SWAPPER, VIRTUAL IS INCREASED THEN RATHER THAN
309 ;ON THE CALL TO CORE0.
310 ;IT WILL TRY TO REASSIGN CORE IN MEMORY IF OLD CORE IN MEMORY
    
```

```
310                   ;IF THIS FAILS, IT WILL REASSIGN NEW CORE ON DISK AND ASK SWAPPER TO EXPAND  
311                   ;IF JOB DID NOT HAVE OLD CORE, AN ATTEMPT WILL BE MADE TO ASSIGN CORE IN MEMORY  
312                   ;IF THIS FAILS, TI WILL BE ASSIGNED ON THE DISK AND ASK SWAPPER TO EXPAND  
313                   ;THE OTHER PLACES IN THE MONITOR WHERE THE IN-CORE COUNT IS TOUCHED IS  
314                   ;IN GET WHERE IT INCREMENTS TO SHARE COPY ALREADY IN CORE.  
315                   ;AND END OF SWAPIN OF LOW SEG AND HIGH SEG IS ALREADY IN CORE FOR OTHER USER  
316                   ;THE CORE ROUTINES DO NOT ALTER THE HIGH SEG IN CORE COUNT. IT IS UP TO THE CALLER  
317                   ;(IN SEGCON) TO CALL THE CORE ROUTINES ONLY IF IN CORE COUNT IS 0.  
318                   ;AND END OF SWAPIN OF LOW SEG AND HIGH SEG IS ALREADY IN CORE FOR OTHER USER
```

```

319 000113          CORE0:
320                IFE FTSWAP,<
321                JUMPE PROG,CORGET          ;IS JOB WITHOUT CORE IN MEMORY?
322                >
323                IFN FTSWAP,<
324                EXTERN IMGOUT,CPOPJ1,CHGSWP
325 000113 326340 000123'          JUMPN PROG,CORE0A          ;DOES JOB HAVE CORE IN MEMORY?
326                ; (ALWAYS TRUE BOTH SEGS IF CORE UUD)
327                IFE FT2REL,<
328                EXTERN CORMAX
329                CAML TAC,CORMAX          ;NO, WILL REQUEST FIT IN PHYSICAL CORE?
330                ; COMPARE WITH LARGEST PERMITTED ADR+1(BUILD AND
331                ; ONCE CAN RESTART CORMAX)
332                >
333                IFN FT2REL,<
334                EXTERN SUMSEG
335                PUSHJ PDP,SUMSEG          ;NO, WILL SUM OF BOTH SEGMENTS FIT IN PHYSICAL CORE?
336 000114 260140 000000          ; LARGEST PERMITTED CORE, COMPARE SUM WITH CORMAX
337                >
338                POPJ PDP,
339 000115 263140 000000          ;NO, GIVE ERROR RETURN
340 000116 205100 002000          ;IS JOB SWAPPED OUT?
341 000117 616104 000104'          ;(MAY HAVE 0 DISK SPACE ALTHOUGH SWAPPED OUT)
342 000120 254000 000126'          ;NO, TRY TO ASSIGN CORE IN MEMORY
343 000121 260140 000000          ;YES, CHANGE ASSIGNMENT OF SWAPPING SPACE ON DISK
344                ; INCREASE VIRTUAL(COUNT OF FREE 1K BLOCKS OF SWAPPING
345                ; (SHOULD NEVER NEED TO DECREASE VIRTUAL SINCE
346                ; CORE COMMAND ALWAYS SWAPS JOB IN FIRST),
347                ; (FRAGMENTATION POSTPONES RETURNING SPACE AND
348                ; INCREASING VIRTUAL UNTIL SWAP IN IF NOT ASKING
349                ; FOR 0 VIRTUAL CORE)
350 000122 254000 000100'          ;GIVE OK RETURN TO CALLER
351                JRST CPOPJ1
352 000123          CORE0A:
353                EXTERNAL JBTSWP
354                >
355                IFN FTTRPSET,<
356                EXTERNAL STOPTS
357 000123 336000 000000          SKIPN STOPTS          ;NO, IS TIME SHARING STOPPED BY
358                ; TRPSET UUD DONE FOR JOB 1?
359                >
360                PUSHJ PDP,ANYACT
361 000124 260140 000057'          ;NO, ANY ACTIVE DEVICE?
362 000125 263140 000000          POPJ PDP,          ;YES, CANNOT ASSIGN CORE
363                ; NO, FALL INTO CORE1
    
```

```

364 ;ROUTINE TO TRY TO ASSIGN CORE IN CORE
365 ;LOW OR HIGH SEG MUST NOT BE SWAPPED OUT(CALLER'S RESPONSIBILITY)
366 ;AND MUST NOT HAVE ANY ACTIVE DEVICES(IT MAY HAVE 0 CORE IN CORE THOUGH)
367 ;IN OTHER WORDS HIGH OR LOW SEG MAY OR MAY NOT HAVE VIRTUAL CORE
368 ;BUT IF IT HAS VIRTUAL CORE IT MUST BE IN PHYSICAL CORE
369 ;THIS IS BOTH A LOGICAL AND A PHYSICAL CORE ASSIGNMENT
370 ;FIRST OLD CORE IS RETURNED TO SYSTEM
371 ;THEN NEW REQUEST IS ATTEMPTED TO BE SATISFIED IN LOWEST
372 ;POSITION POSSIBLE, THUS CORE TENDS TO BE PACKED
373 ;IF NEW REQUEST CANNOT BE GRANTED, OLD AMOUNT IS RETAINED, IF NON-SWAPPING SYS
374 ;OTHERWISE SWAPPER IS CALLED(XPAND) TO EXPAND CORE BY SWAPPING OUT
375
376
377             EXTERN SEGSIZ
378
379 000126      CORE1: NOSCHEDULE      ++      ;PREVENT SCHEDULING
380
381             IFN FTSWAP,<
382             EXTERN VIRTAL
383 000126 260140 000000      PUSHJ PDP,SEGSIZ      ;TAC1=OLD SEG SIZE
384 000127 336540 000001      SKIPN LOC,TAC        ;IS 0 BEING REQUESTED?
385 000130 211540 000001      MOVNI LOC,1          ;YES, PREFND -1(DEPEND ON ASH BUG WHICH KEEPS -1
386                                     ; ON RT. SHIFT)
387 000131 240540 777766      ASH LOC,-12         ;CONVERT TO NO. OF K-1(0,1,2,...)
388 000132 274540 000002      SUB LOC,TAC1        ;NO. OF K-1 INCREASE=NEW-OLD-1
389 000133 315540 000000      CAMGF LOC,VIRTAL    ;IS THERE ENOUGH FREE VIRTUAL CORE IN SYSTEM?
390 >
391             IFE FT2REL,<
392             CAML TAC,CORMAX      ; YES, IS REQUEST LESS THAN MAX, ALLOWED COR+1?
393 >
394             IFN FT2REL,<
395             EXTERN SUMSEG
396 000134 260140 000114'    PUSHJ PDP,SUMSEG     ;YES, IS SUM OF SEGS LESS THAN MAX, ALLOWED CORE+1?
397 >
398 000135 263140 000000      POPJ PDP,            ;NO, ERROR RETURN
399             IFN FTSWAP,<
400 000136 271540 000001      ADDI LOC,1          ;YES, GET NO. OF K OF INCREASE
401 000137 213000 000013      MOVNS LOC          ;MAKE MINUS FOR UPDATE
402 000140 272540 000133'    ADDM LOC,VIRTAL    ;AND UPDATE TOTAL VIRTUAL CORE IN SYSTEM
403                                     ; SINCE THIS REQUEST CAN BE SATISFIED
404 >
405             CORE1A: NOSCHEDULE    ++      ;PREVENT JOB SCHEDULING
406 000141 322340 000146'    JUMPE PROG,CORSET  ;OLD ASSIGNMENT 0?
407                                     ; IF YES, DO NOT ATTEMPT TO RETURN OLD CORE
408 000142 550540 000007      HRRZ LOC,PROG      ;NO, ABS. LOC. OF OLD CORE
409 000143 554500 000007      HLRZ BLK,PROG      ;HIGHEST LEGAL RFL. ADR.
410 000144 201640 000000      MOVEI T,0          ;CLEAR FOR CORSTG CALL
411 000145 260140 000272'    PUSHJ PDP,CORSTG    ;RETURN OLD CORE TO FREE STORAGE
    
```

```

412                                     ;CORGET IS CALLED BY SWAPPER WHEN JOB IS ON DISC AND IS
413                                     ;WANTED IN CORE.
414
415 000146 403540 000007 CORGET: SETZR LOC,PROG          ;SET NEW ASSIGNMENT TO 0 AND DIST. MOVED
416 000147 322040 000227' JUMPF TAC,0IDLE1          ;IS ZERO CORE BEING REQUESTED?
417 000150 260140 000252' PUSHJ PDP,HOLSRC          ;NO, SEARCH FOR HOLE BIG ENOUGH
418 000151 254000 000165' JRST BAKOLD              ;NONE, GIVE BACK OLD AMOUNT
419                                     INTERN FTTRACK
420                                     IFN FTTRACK,<
421                                     EXTERN LASCOR
422                                     MOVEM ITEM,LASCOR          ;LEAVE TRACKS FOR LAST JOB USING
423                                     ; PHYSICAL CORE ALLOCATION
424                                     ; (FOR DEBUGGING ONLY)
425
426 000152 202540 000007 > MOVEM LOC,PROG          ;SETUP NEW RELOC
427 000153 506040 000007 HRLM TAC,PROG          ;AND NEW PROTECT.
428 000154 201501 000000 MOVET BLK,(TAC)          ;HIGHEST REL ADR. BEING REQUESTED
429 000155 201640 000001 MOVET T,1              ;SET USE BITS IN CORE TABLE
430 000156 260140 000272' PUSHJ PDP,CORSTG          ;
431 000157 200504 000034' MOVE BLK,JRTADR(ITEM)      ;OLD CORE ASSIGNMENT
432 000160 326500 000170' JUMPW BLK,MOVCOR          ;WAS THERE OLD MEMORY ASSIGNED?
433                                     ; NO,
434                                     IFN FTSWAP,<
435 000161 135100 000000 LDR TAC1,IMGOUT          ;SIZE(IN K) OF SEG ON DISK TO BE SWAPPED IN
436 000162 242100 000012 LSH TAC1,12          ;CONVERT TO WORDS
437                                     >
438                                     IFE FTSWAP,<
439                                     MOVEI TAC1,0          ;JOB HAS NO PREVIOUS VIRT. CORE(NON-SWAP SYS)
440                                     >
441 000163 200500 000007 MOVE BLK,PROG          ;MAKE OLD ASSIGNMENT(BLK) APPEAR TO START
442                                     ; AT SAME PLACF AS NEW ASSIGNMENT(FOR CLEARING)
443 000164 364100 000204' SOJA TAC1,CLPCR1          ;IF NEW CORE SIZE IS BIGGER THAN
444                                     ;OLD, CLEAR OUT INCREASE SO SECURITY WILL
445                                     ; BE MAINTAINED, TAC1 IS SIZE-1 OF OLD
446                                     ; ASSIGNMENT, -1 OF NO OLD ASSIGNMENT
    
```



```

468                                     ;MOVE OLD CORE TO NEW AREA
469
470 000170 306552 000000 MOVCOR:  CAIN LOC,(BLK)      ;IS NEW CORE IN SAME PLACE AS OLD?
471 000171 254070 000203' JRST CLRCOR      ;YES, DO NOT MOVE IT,CLEAR IF INCREASE
472 000172 554100 000012 HLRZ TAC1,BLK    ;LENGTH OF OLD CORE
473 000173 303101 000000 CAILF TAC1,(TAC)      ;IS OLD CORE LESS THAN NEW?
474 000174 550100 000001 HRRZ TAC1,TAC    ;NO, MOVF THE SHORTENED NEW CORE
475
476 IFN FTIME,<
477 000175 272100 000000 EXTERNAL SHFWRD
478                                     ADDM TAC1,SHFWRD      ;INCREMENT TOTAL NO. WORDS SHUFFLED
479                                     >
480 000176 270100 000013 ADD TAC1,LOC      ;ADD IN NEW RELOC.
481 000177 200640 000013 MOVE AC1,LOC      ;DEST.=NEW RELOC.
482 000201 402004 000166' HRL AC1,BLK      ;SOURCE=OLD RELOC.
483 000202 251642 000000 SETZM JBTADR(ITEM) ;FLAG THAT CORE IS IN TRANSIT(TTY ROUTINES)
484                                     BLT AC1,(TAC1)      ;MOVE CORE TO NEW ASSIGNMENT
485
486                                     ;CLEAR INCREASE IF NEW CORE IS BIGGER THAN OLD
487
487 000203 554100 000012 CLRCOR:  HLRZ TAC1,BLK    ;OLD CORE SIZE-1
488 000204 317040 000002 CLRCR1:  CAMG TAC,TAC1      ;IS NEW CORE SIZE-1 GREATER THAN OLD CORE SIZE-1
489 000205 254070 000216' JRST DIDL      ;NO, DO NOT CLEAR ANY CORE
490
491 IFN FTIME,<
492 000206 200640 000071 MOVE AC1,TAC      ;NEW CORE SIZE
493 000207 274640 000002 SUR AC1,TAC1      ;IFESS OLD CORE SIZE
494 000210 272640 000000 ADDM AC1,CLRWRD   ;ACCUMULATE NO. OF WORDS CLEARED
495                                     >
496 000211 271107 000002 ADDI TAC1,2(PROG) ;YES, OLD SIZE-1+NEW RELOC+2=2ND LOC TO CLEAR
497 000212 271047 000000 ADDI TAC,(PROG)   ;NEW SIZE-1+NEW RELOC=LAST LOC TO CLEAR
498 000213 402002 777777 SETZM -1(TAC1)    ;CLEAR FIRST WORD
499 000214 505102 777777 HRLI TAC1,-1(TAC1) ;SET LH TO FIRST ADR. TO CLEAR
500 000215 251101 000000 BLT TAC1,(TAC)    ;CLEAR THE INCREASE PORTION

```

```

520
521
522      ;IF THE SHUFFLED JOB IS IN EXEC MODE, ITS DUMP ACS
523      ;(PDP,PROG,JOAT SAVED IN JOB DATA AREA) MUST BE
524      ;ALTERED BY DISTANCE CORE WAS MOVED
525
526      ;IF THE SHUFFLED JOB IS CURRENT JOB, THE SOFTWARE STATE OF
527      ;THE MONITOR(IE SOFTWARE INFO OF JOB) MUST BE ALTERED BY AMOUNT
528      ;CORE WAS MOVED
529
530      EXTERNAL SYSSIZ
531
532      000216 275552 000000  DITLE:  SUR1 LOC,(BLK)      ;DISTANCE JOB WAS MOVED(DEST.-SOURCE)
533      000217 312200 000067'  CAME ITEM,JOB      ;IS THIS CURRENT JOB?
534      000220 334047 000000  SKIPA TAC,JORPC(JOAT) ;NO, GET PC IN JOB DATA AREA
535      000221 200040 000000  MOVE TAC,USRPC     ;YES, PC IN PROTECTED SYSTEM AREA
536
537      IFN FT2REL,<
538      EXTERN JOBMAX
539      CAIG ITEM,JORMAX      ;IS THIS A HIGH SEGMENT?
540
541      >
542      TLNE TAC,USRMON     ;NO, IS JOB IN USER MODE?
543      JRST DITLE1        ;YES, DO NOT ALTER DUMP ACS
544
545      ADDM LOC,JOBOPD(JOAT) ;BECAUSE THEY ARE THE USERS(OR HIGH SEG)
546      ADDM LOC,JOBOPG(JOAT) ;AND ALTER PDP BY DIST, OF MOVE
547      MOVEM PROG,JRTADR(ITEM) ;STORE NFW CORE ASSIGNMENT(LOW OR HIGH SFG)
548      CAME ITEM,JOB      ;IS THIS CURRENT JOB?
549      JRST DITLE3        ;NO, DO NOT ALTER STATE OF MONITOR
550      MOVE TAC,SYSSIZ     ;DONT CHANGE PDP IF LIST
551      CAIG TAC,(PDP)     ;IS IN SYSTEM
552      ADD PDP,LOC        ;YES, ALTER PUSH DOWN POINTER BY AMOUNT OF MOVE
553      PUSHJ PDP,SETREL   ;GO SETUP NEW HARDWARE AND SOFTWARE RELOCATION
554
555      DITLE3:
556      IFN FT2REL,<
557      EXTERN CURHIGH
558      PUSHJ PDP,CURHIGH   ;CHECK TO SEE IF THIS CORE ASSIGNMENT IS FOR
559
560      000236 260140 000000  ; HIGH SEG WHICH CURRENT USER MAY ALSO BE USING
561
562      ; IF YES, RESET HARDWARE AND SOFTWARE RELOC INFO,
563      ; RETURN WITH ITEM PRESERVED,PROG SET TO RFLOC
564      ; OF SEG WHICH HAS JUST HAD CORE REASSIGNED
565
566      >
567      SETZR TAC,HOLEF
568      PUSHJ PDP,HOLSRC    ;CLEAR HOLE FLAG
569      JRST COROK         ;IS THERE A NON-ZERO HOLE?
570
571      ADDI LOC,1(BLK)    ;NO
572      CAME T,CORLST     ;YES, FORM ADR, OF JOB JUST ABOVE HOLE
573      MOVEM LOC,HOLEF   ;IS HOLE AT TOP OF MEMORY
574
575      COROK:
576      IFN FTSWAP,<
577      EXTERN BIGHOLE
578      MOVEI TAC,-1
579      PUSHJ PDP,HOLSRC   ;NO, FLAG WITH ADDRESS OF JOB ABOVE HOLE
580
581      000245 201040 777777  ;FIND BIGGEST HOLE
582      000246 260140 000252' ;ALWAYS GET ERROR RETURN
    
```


CORE1 - LOGICAL AND PHYSICAL CORE ALLOCATION ROUTINES - V414
T. HASTINGS/TH/RCC TS 04 JUN 69

MACRO,V36 19:03 4-JUN-69 PAGE 25-1

553	000247	240700	777766	ASH T1,-+D10	;CONVERT TO 1K BLOCKS
554	000250	202700	000000	MOVEM T1,BIGHOLE	
555				>	
556				SCHEDULE++	
557	000251	254000	000122	JRST CPOPJ1	;SKIP RETURN(UNLESS ERROR)

```

558
559 ;ROUTINE TO FIND HOLE BIG ENOUGH FOR REQUEST
560 ;CALL: MOVE TAC,HIGHEST REL. ADP. ASKING FOR
561 ; PUSHJ PDP,HOLSRC
562 ; RETURN1 ;NO HOLES BIG ENOUGH
563 ; RETURN2 ;T BYTE SET TO LAST BLOCK+1 IN HOLE
564 ; ;BLK SFT TO HIGHEST REL. LOC. IN THAT HOLE
565 ; ;LOC SFT TO ADDRESS OF FIRST BLOCK IN HOLE
566 ; ;T1=LARGEST HOLE SEEN
567 ;USES TAC1
568
569 000252 200640 000313' HOLSRC: MOVE T,CORE2P ;BYTE POINTER TO FIRST BIT-1
570 000253 403540 000016 SETZR LOC,T1 ;START AT BOTTOM OF MEMORY
571 ; ;LARGEST HOLE SO FAR=0
572 000254 634570 000012 CORHOL: TDZA BLK,BLK ;START BLK AT 0 AND SKIP
573
574 000255 271570 002070 CORH00: ADDI BLK,2000 ;INCREMENT HIGHEST REL LOC.
575 000256 316640 000243' CORH01: CAMN T,CORLST ;BYTE POINTER TO 1ST NON-EXISTANT BLOCK
576 000257 263140 000000 POPJ PDP, ;NO MORE CORE TO SEARCH
577 000260 134170 000015 ILDR TAC1,T ;GET NEXT CORE USE BIT
578 000261 271540 002000 ADDI LOC,2000 ;INCREMENT ADDRESS OF BLOCK
579 000262 322100 000255' JUMPE TAC1,CORH0? ;IS THIS BLOCK IN USE?
580 000263 322570 000256' JUMPE BLK,CORH01 ;YES, HAVE ANY FREE BLOCKS BEEN SEEN YET?
581 ;IFN FTSWAP,K
582 000264 313520 000016 CAMLE BLK,T1 ;YES, BIGGEST SO FAR?
583 000265 202570 000016 MOVEV BLK,T1 ;YFS, SAVE IN T1.
584 >
585 000266 317570 000071 CAMG BLK,TAC ;YES, IS THIS HOLE EQUAL OR GREATER THAN REQUEST?
586 000267 254070 000254' JRST CORHOL ;NO, KEEP LOOKING FOR HOLES
587 000270 275552 002070 SURJ LOC,2000(BLK) ;YES, SET LOC TO FIRST BLOCK IN HOLE
588 000271 364570 000251' SOJA BLK,CPOPJ1 ;SFT BLK TO HIGHEST REL. LOC.
589 ; AND RETURN
    
```

```

590          ;ROUTINE TO SET AND CLEAR CORE USE TABLE
591          ;CALL: MOVEI T,1          ;TO SET TABLE
592          ;          MOVEI T,0          ;TO CLEAR TABLE
593          ;          MOVE BLK,HIGHEST REL, LOC. IN USER AREA
594          ;          MOVE LOC,ADDRESS OF FIRST BLOCK TO SET CLEAR
595
596          INTERN CORE2P
597          EXTERN TPOPJ
598
599 000272 261140 000001 CORSTG: PUSH PDP,TAC          ;SAVE HIGHEST LOC, BEING REQUESTED
600 000273 240500 777766          ASH BLK,-12          ;CONVERT TO NO. OF BLOCKS-1
601 000274 271500 000001          ADDI BLK,1          ;NO. OF BLOCKS
602 000275 332000 000015          SKIPE T          ;UPDATE NO OF FREE BLOCKS
603 000276 213000 000012          MOVNS RLK          ;DECREASE IF SETTING BITS
604 000277 272500 000000          ADDM BLK,CORTAL          ;INCREASE IF CLEARING,DECREASE IF SSETTING BITS
605 000300 200040 000013          MOVE TAC,LOC          ;ADDRESS OF FIRST BLOCK
606 000301 240040 777766          ASH TAC,-12          ;FORM BYTE POINTER TO BIT-1
607 000302 231040 000044          IDIVI TAC,*D36          ;TAC=WORD,TAC1=BIT
608 000303 270040 000313          ADD TAC,CORE2P          ;FORM BYTE POINTER

609 000304 213000 000002          MOVNS TAC1
610 000305 271100 000044          ADDI TAC1,*D36
611 000306 137100 000325          DPR TAC1,[POINT 6,TAC,5]
612 000307 217000 000012          MOVMS RLK          ;GET MAG. OF NO. OF BLOCKS INVOLVED
613 000310 136640 000001          IDPB T,TAC          ;SET OR CLEAR EACH USE BIT
614 000311 367500 000310          SOJG BLK,,-1
615 000312 254000 000000          JRST TPOPJ          ;RESTORE TAC, AND POPJ
616
617 000313 440100 000000 CORE2P: POINT 1,CORTAB          ;BYTE POINTER TO FIRST BIT-1
    
```

```

618
619
620      ;ROUTINE TO CLEAR PART OF JOB DAT AREA(PART PROTECTED FROM USER IO)
621      ;CALLED WHEN NEW CORE ASSIGNED AND AT SYSTEM RESTART(140)
622      ;      MOVE ITEM,JOB NO.
623      ;CALL:  MOVE JDAT,ADR. OF JOB DATA AREA
624      ;      PUSHJ PDP,CLRJOB
625
626      INTERNAL CLRJOB
627      EXTERNAL JOBPR1,JOBPR1,JOBPF1,JORENB
628      EXTERNAL JOBPD1,JORDDT
629      000314 402007 000000 CLRJOB: SETZM JOBPR1(JDAT)      ;FIRST LOC. PROTECTED FROM USER
630      000315 205047 000314'  MOVSI TAC,JOBPR1(JDAT)
631      000316 541047 000000      MRR1 TAC,JOBPR1(JDAT)
632      000317 200107 000000      MOVE TAC1,JORDDT(JDAT)      ;SAVE DDT STARTING ADDRESS
633      000320 251047 000000      RLT TAC,JOBPF1(JDAT)
634      000321 202107 000317'  MOVEM TAC1,JORDDT(JDAT)
635      000322 402007 000000      SETZM JORENB(JDAT)      ;ALSO CLEAR APR ENABLE WORD
636      000323 402007 000000      SETZM JOBPD1(JDAT)      ;AND UO0 PC FLAGS(USED WHEN JOB STARTS)
637      000324 254000 000000      JRST ESTCP1           ;GO SET JOB STATUS, SO CONT WILL
638                                     ; NOT WORK,(DO NOT CLEAR JACCT BIT)
639
640
641      000325
642      000325 360600 000271      COREND: FND
    
```

NO ERRORS DETECTED
 PROGRAM BREAK IS 000326

AC1		000015	INT	AC2		000016	INT	AC3		000017	INT
ANY		000075'		ANY2		000077'		ANYACT		000057'	INT
ANYDEV		000061'	INT	ANYSAV		000107'	EXT	BAKOLD		000165'	INT
RIGHOL		000250'	EXT	BKOLD1		000166'		BLK		000012	
RUFFNT		000012	INT	RUFWRD		000013	INT	CHGSWP		000121'	FXT
CHKSHF		000021'	INT	CLKCHL		000000	EXT	CLRCOR		000203'	
CLRCR1		000204'		CLRJOB		000314'	INT	CLRWRD		000210'	FXT
CORE0		000113'	INT	CORE0A		000123'		CORE1		000126'	INT
CORE1A		000141'		CORE2P		000313'	INT	COREND		000325'	
CORERR		000014'		CORGET		000146'	INT	CORH00		000255'	
CORH01		000256'		CORHOL		000254'		CORLST		000256'	FXT
CORMAX		000016'	EXT	COROK		000245'		CORSTG		000272'	
CORTAB		000313'	EXT	CORT4L		000277'	EXT	CORU1		000010'	
CORU00		000000'	INT	CPOPJ1		000271'	FXT	CURHGH		000236'	FXT
DAT		000005	INT	DEVDA1		000006	INT	DEVIOS		000002	INT
DEVMOD		000004	INT	IDLE		000216'		IDLE1		000227'	
IDLE3		000236'		DVTTY		000010	INT	ESTOP1		000324'	FXT
FIT		000050'	EXT	FORCE		000047'	EXT	FTREL	777777	777777	
FTCCL	777777	777777		FTDISK	777777	777777		FTLOG1	777777	777777	
FTRC10	777777	777777		FTSW4P	777777	777777	INT	FITIME	777777	777777	INT
FTTRAC		000000	INT	FTTRPS	777777	777777	INT	HOLEF		000244'	FXT
HOLLOP		000026'		HOLSRC		000252'		IMGOUT		000161'	FXT
IOACT		010000	INT	IOS		000000	INT	IOWAIT		000002	FXT
ITEM		000004	INT	JBTAOR		000227'	EXT	JBTDAT		000000	FXT
JBTMAX		000025'	EXT	JBTSTS		000117'	FXT	JBTSWP		000000	FXT
JDAT		000007	INT	JOB		000230'	FXT	JOBADR		000000	FXT
JORDAC		000000	EXT	JORDAT		000000	FXT	JOBDDT		000321'	FXT
JORDPD		000225'	EXT	JORDPG		000226'	EXT	JORENB		000322'	FXT
JORJDA		000066'	EXT	JORJMH		000072'	FXT	JORMAX		000222'	FXT
JORPC		000220'	EXT	JORPD1		000323'	FXT	JORPFI		000300'	FXT
JORPR1		000316'	EXT	JORPRT		000315'	FXT	JORREL		000000	FXT
KCORE1		000110'	INT	LOC		000013		MOVCOR		000170'	
NOTSH1		000041'		NOTSHF		000045'		NSHF		001000	INT
NSWP		010000	INT	ORRROR		000033'	EXT	PDP		000003	INT
PROG		000007	INT	SCORE1		000111'		SEGSIZ		000126'	FXT
SETREL		000235'	EXT	SHF		004000	INT	SHFLOP		000034'	
SHFWAT		000053'	EXT	SHFWRD		000175'	EXT	STOPTS		000123'	FXT
STOTAC		000020'	EXT	SUMSEG		000134'	FXT	SWP		002000	INT
SYSSIZ		000232'	EXT	T		000015		T1		000016	
TAC		000001	INT	TAC1		000002	INT	TPOPJ		000312'	FXT
TRYSWP		000104'	INT	UCORHI		000011'	EXT	USRJDA		000070'	FXT
USRM0D		010000	INT	USRPC		000221'	FXT	USRREL		000000	FXT
VCORE1		000414	INT	VIRTAL		000140'	FXT	WSCHED		000015'	FXT
XPAND		000165'	EXT	ZERCOR		000016'					

CURHGH	534	535			
D	6#	6			
DAT	6#	6	153		
DCL	6#	6			
DCLI	6#	6			
DCLQ	6#	6			
DCLR	6#	6			
DDI	6#	6			
DDO	6#	6			
DEN	6#	6			
DEVADR	6#	6			
DEVBUF	6#	6			
DEVCHR	6#	6			
DEVCTR	6#	6			
DEVDAT	6#	6	226	227	230
DEVEXT	6#	6			
DEVFIL	6#	6			
DEVIAD	6#	6			
DEVIOS	6#	6	227		
DEVLOG	6#	6			
DEVMOD	6#	6	230		
DEVNAM	6#	6			
DEVOAD	6#	6			
DEVPPN	6#	6			
DEVPTR	6#	6			
DEVSER	6#	6			
DGF	6#	6			
DHNG	6#	6			
DIDLE	489	511#			
DIDLE1	416	520	524#		
DIDLE3	526	532#			
DIN	6#	6			
DINI	6#	6			
DLK	6#	6			
DMT	6#	6			
DNAERR	6#	6			
DOU	6#	6			
DR	6#	6			
DRL	6#	6			
DRN	6#	6			
DSEB	6#	6			
DSI	6#	6			
DSKRLB	6#	6			
DSD	6#	6			
DVAVAL	6#	6			
DVCDR	6#	6			
DVDIR	6#	6			
DVDIRI	6#	6			
DVDIS	6#	6			
DVDSK	6#	6			
DVDTA	6#	6			
DVIN	6#	6			
DVLNG	6#	6			

CODES	6#		
DISARL	6#		
ENABLE	6#		
NOSCHE	6#	379	425
NOSHUF	6#		
QUEUES	6#		
SCHEDU	6#	556	
SHUFFL	6#		
STARTO	6#		
XP	6#	6	13