

```

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 COMCON - COMMAND DECODER AND SAVEGET ROUTINES - V433
12     SUBTTL /RCC 03 JUN 69
13     XP VCOMCN,433+
14
15         ;THIS MACRO PUTS VERSION NO, IN STORAGE MAP AND GLOB
16
17     ENTRY COMCON      ;ALWAYS LOAD COMCON IF LIBRARY SEARCH
18     COMCON:
19
20     ;CALLED FROM CLOCK ROUTINE WHEN 'COMCNT' IS GREATER THAN 0
21
22     ;AS SET BY TTY SERVICE ROUTINE
23     ;ALL AC'S HAVE BEEN SAVED BY CLOCK CHANNEL
24     ;THE COMMAND DECODER CALLS TTYCOM WHICH SCANS FOR TTY WHICH TYPED
25     ;THE COMMAND AND THEN DISPATCHES(PUSHJ) TO APPROPRIATE
26     ;COMMAND SETUP ROUTINE OF THE SAME NAME WITH AC'S SET AS:
27     ;ITEM = JOB NUMBER
28     ;TAC = BYTE POINTER TO COMMAND STRING(SPACE OR CR AFTER COMMAND NAME)
29     ;DAT = BYTE POINTER TO LAST OUTPUT CHARACTER
30     ;DEV DAT = ADDRESS OF DEVICE DATA BLOCK TYPING COMMAND
31     ;JDAT = ADDRESS OF JOB AREA, = 0 IF NO JOB AREA
32     ;PROG = ADDRESS OF PROGRAM AREA, 0 MEANS NOT IN CORE OR NO CORE
33     ;IOS IS SET TO 0, USED FOR ADDRESS OF SUBSEQUENT CALLS
34     ;DEV DAT IS ALSO PUSHED ON END OF PD LIST SO IT MAY BE DESTROYED.
35     ;UPON RETURN FROM COMMAND SETUP ROUTINE, A CR-LF IS ADDED TO
36     ;MESSAGE AND TTY IS STARTED ON TTY SPECIFIED BY -1(PDP)
37     ;SEVERAL COMMAND FLAGS ARE CHECKED BEFORE DISPATCHING TO
38     ;COMMAND SETUP ROUTINES TO SEE IF COMMAND IS LEGAL AT THIS TIME
39     ;SEVERAL MORE FLAGS ARE CHECKED UPON RETURN FROM COMMAND SETUP
40     ;ROUTINES(UNLESS AN ERROR HAS OCCURRED) FOR STANDARD
41     ;COMMAND RESPONSE
42     ;IF AN ERROR OCCURS, THE JOB NO.(ITEM) SHOULD BE 0 ON RETURN
43     ;SO THAT JOB WILL NOT BE INITIALIZED IF FIRST COMMAND,
44     ;ERRMES ROUTINE WILL SET ITEM TO 0
45
46     ;SINCE THE COMMAND DECODER IS CALLED FROM THE CLOCK ROUTINE
47     ;COMMAND SETUP ROUTINE MUST RUN TO COMPLETION QUICKLY
48     ;IF A COMMAND FUNCTION CANNOT DO THIS, IT MUST JUST SET
49     ;THE JOB TO RUNABLE STATUS AND RETURN IMMEDIATELY
50     ;OR DELAY THE COMMAND FOR LATER EXECUTION

```

```

49                                   INTERNAL COMMAND
50                                   INTERNAL FTLOGIN,FTSWAP,FTTIME
51                                   EXTERNAL COMCNT,TTYCOM,JRTSTS,JOBMAX,JBTDAT,JBTADR,JOBHCU
52                                   EXTERNAL CONFIG,HIGHJB
53                                   EXTERNAL TTYSTR,TTYUSR,TTYATT
54                                   EXTERNAL TTYTAB,PUNIT,TTYATI
55                                   EXTERNAL ADRERR,ANYACT,ASSASG,CORE0,DAMESS,DEVLG,DEVPHY,DEVSRG
56                                   EXTERNAL DLYCOM,ERNAM,ESTOP,GETWD1,GETWDU,KSTOP,CORE1,MONJOB,DECP1
57                                   EXTERNAL MONSTR,MSTART,OCTPNT,PHOLD,PRNAME,RADX10,RELEA5
58                                   EXTERNAL RELEA6,RESET,SETRUN,STOP1
59                                   EXTERNAL STOTAC,UPTIME,USRXIT,USTART,UUOERR
60
61
62                                   000012 T=BUFPNT                   ;TEMPORARY AC'IS
63                                   000013 T1=BUFWRD
64                                   000014 T2=UUO
65                                   000015 T3=AC1
66                                   000016 T4=AC2
67 000000 260140 000000 COMMAND:PUSHJ PDP,TTYCOM           ;SETUP DEVDAT,DAT,TAC,AND ITEM
68                                                                           ;FOR ANY TTY WHICH HAS TYPED A COMMAND
69 000001 263140 000000 COM0:   POPJ PDP,                           ;NONE FOUND
70 000002 261140 000006           PUSH PDP,DEV DAT           ;SAVE TTY DEVICE DATA BLOCK ADDRESS
71 000003 260140 001721'           PUSHJ PDP,CTEXT           ;SCAN COMMAND NAME, RETURN IT IN TAC1
72 000004 200500 000002           MOVE T,TAC1               ;COPY COMMAND,
73 000005 211540 000001           MOVNI T1,1               ;SET MASK ALL ONES
74 000006 242540 777772           LSH T1,-6               ;CLEAR OUT ONE MORE CHAR.
75 000007 242500 000006           LSH T,6                 ;SHIFT 1 COMMAND CHAR OFF
76 000010 326500 000006'           JUMPN T,-2               ;IF NOT END OF COMMAND, GO AROUND
77 000011 201700 000000           MOVEI T4,0               ;CLEAR FLAG REGISTER
78 000012 205500 777721           MOVSI T,-DISPL           ;SEARCH COMMAND TABLE FOR MATCH
79 000013 200612 000232' COMPL:   MOVE T2,COMTAB(T)           ;GET NEXT ENTRY FROM COMMAND TABLE
80 000014 630600 000013           TDZ T2,T1               ;MASK OUT CHAR 5
81 000015 316112 000232'           CAMN TAC1,COMTAB(T)       ;EXACT MATCH?
82 000016 254000 000027'           JRST COMFND              ;YES, THIS IS IT.
83 000017 312100 000014           CAME TAC1,T2             ;MATCH?
84 000020 254000 000024'           JRST COMNEQ              ;NOT EVEN GOOD MASK
85 000021 662700 000001           TROE T4,1               ;MATCHES MASKED, IS THIS FIRST ONE
86 000022 664700 000002           TROA T4,2               ;NO, SET 2ND OCCUR FLAG
87 000023 200640 000012           MOVE T3,T               ;YES, COPY CURRENT INDFX
88 000024 253500 000013' COMNEQ:   AORJN T,COMPL           ;NO, KEEP LOOKING
89 000025 306700 000001           CAIN T4,1               ;DID ONE AND ONLY ONE COMMAND MATCH?
90 000026 200500 000015           MOVE T,T3               ;YES, GET ITS INDEX

```

```

91 000027 200112 000311' COMFND: MOVE TAC1,DISP(T)      ;GET DISPATCH TABLE ENTRY,
92 000030 261140 000002      PUSH PDP,TAC1      ;SAVE RH(DISPATCH ADR,+BITS)
93 000031 200514 000000      MOVE T,JRTSTS(ITEM) ;JOB STATUS WORD FOR THIS JOB
94                                IFN FTLOGIN,<
95 000032 607500 000004      TLNN T,JLOG          ;IS JOB LOGGED IN?
96 000033 603100 100000      TLNE TAC1,NOLOGIN   ;NO, CAN COMMAND PROCEED WITH NO LOGIN?
97 000034 254000 000041'      JRST CHKNO          ;YES
98 000035 265040 000134'      JSP TAC,COMER      ;NO, TYPE "LOGIN PLEASE"
99 000036 462370 744634      ASCIIZ /LOGIN PLEASE
000037 202411 442602
100 000040 516121 505000 /
101 >
102 000041 326200 000070' CHKNO: JUMPN ITEM,CHKRUN      ;JOB NUMBER ALREADY ASSIGNED?
103 000042 603100 200000      TLNE TAC1,NOJOB     ;NO, DOES THIS COMMAND NEED A JOB NUMBER?
104 000043 254000 000137'      JRST COMGO          ;NO
105 000044 201200 000001      MOVEI ITEM,1        ;YES, SCAN FOR ONE STARTING WITH 1
106 000045 200504 000031' NUMLOP: MOVE T,JRTSTS(ITEM) ;SCAN FOR FREE JOB NO.
107 000046 607500 240000      TLNN T,JNA+CMWR     ;THIS NUMBER ASSIGNED?
108 000047 254000 000060'      JRST NEWJOB        ;NO, SO USE THIS NUMBER

109 000050 305200 000000      CAIGE ITEM,JOBMAX   ;YES, IS THE MAX. JOB NO.?
110 000051 344200 000045'      AOJA ITEM,NUMLOP    ;NO, KEEP LOOKING
111 000052 265040 000134'      JSP TAC,COMER      ;YES, NONE LEFT, PRINT "JOB CAPACITY EXCEEDED"
112 000053 452370 220206      ASCIIZ /JOB CAPACITY EXCEEDED
000054 406410 141622
000055 522624 042660
000056 416130 542212
113 000057 420321 200000 /
114
115
116                                ; EVEN THOUGH THIS IS A NEW JOB NUMBER
117                                ; IT MAY HAVE CORE ASSIGNED NOW BECAUSE IT WAS DELAYED
118
119 000060 201540 400000 NEWJOB: MOVEI T1,ASSCON      ; UNTIL IT COULD BE SWAPPED IN(LOGIN WITH CORE FULL)
120 000061 436546 000004      IORM T1,DEVMOD(DEVDAT) ;SET ASSIGNED BY CONSOLE BIT FOR TTY
121 000062 402006 000005      SETZM DEVLOG(DEVDAT)   ;SO OTHER JOBS CAN NOT USE
122                                ;SET LOGICAL NAME TO ZERO
123                                ; "TTY" IS PUBLIC LOGICAL NAME
124                                IFN FTTIME,<
125 000063 402004 000000      SETZM RTIME(ITEM)    ;CLEAR INCREMENTAL JOB RUNNING TIME
126 000064 402004 000000      SETZM TTIME(ITEM)    ;CLEAR TOTAL JOB RUNNING TIME
127 >
128                                IFN FTKCT,<
129 000065 402004 000000      EXTERN JRTKCT
130                                SETZM JBTKCT(ITEM) ;CLEAR CORE-RUNNING TIME PRODUCT CODE
131 >
132 000066 313200 000000      CAMLE ITEM,HIGHJB   ;HIGHEST JOB NUMBER ASSIGNED?
133 000067 202200 000066'      MOVEM ITFM,HIGHJR   ;YES,SAVE IT FOR SCHEDULER SCAN OF JOBS
  
```

```

134 000070 603500 400000 CHKRUN: TLNE T,RUN ;RUN BIT ON IN JOB STATUS?
135 000071 607100 010000 TLNN TAC1,NORUN ;YES, DOES THIS COMMAND REQUIRE A JOB?
136 000072 254000 000101' JRST CHKACT ;NO
137 000073 265040 000134' JSP TAC,COMER ;YES,
138 000074 502310 540646 ASCIIZ /PLEASE TYPE *C FIRST
000075 425012 454640
000076 425013 641500
000077 432232 251650
139 000100 064240 000000 /
140
141 000101 CHKACT:
142
143 IFN PROG-JDAT,<
144 MOVE JDAT,JBTDAT(ITEM) ;ADDRESS OF JOB DATA AREA
145 >
146 000101 200344 000000 MOVE PROG,JBTADR(ITEM) ;XWD PROTECTION,RELOCATION
147 IFN FTSWAP,<
148 000102 603100 024000 TLNE TAC1,INCORINOTRAN ;MUST JOB NOT BE SWAPPING OR
149 ; IF JOB HAS CORE ASSIGNED, MUST IT BE
150 ; IN PHYSICAL CORE (RATHER THAN DISK OR ON ITS WAY)
151 000103 607500 002000 TLNN T,SWP ;YES, IS JOB ON DISK OR ON ITS WAY?
152 000104 254000 000113' JRST CHKC02 ;NO
153 000105 607100 004000 TLNN TAC1,INCOR ;YES, MUST JOB BE IN CORE?
154 000106 322340 000113' JUMPF PROG,CHKC02 ;NO, IS A SWAP FOR THIS JOB IN PROGRESS?
155 ; NO, SO COMMAND MUST BE DELAYED
156 ; (EITHER BECAUSE SWAP IN PROGRESS OR
157 ; JOB ON DISK AND MUST BE IN PHY CORE)
158 000107 541100 001676' HRR1 TAC1,DLYCM ;ASSUME JOB MUST BE IN PHY CORE
159 ; SO SET TO SWAP JOB IN
160 000110 607100 004000 TLNN TAC1,INCOR ;IS THIS TRUE?
161 CHKDLY: HRR1 TAC1,DLYCM1 ;NO, JUST DELAY COMMAND UNTIL SWAP OUT OR IN IS FINISHE
162 000111 541100 001677' D
163 000112 254000 000143' JRST COMDIS ;AND DISPATCH TO DELAY COMMAND
164 >
165 000113 603100 040000 CHKC02: TLNE TAC1,NOACT ;CAN COMMAND BE PERFORMED WITH ACTIVE DEVICES?
166 000114 260140 001665' PUSHJ PDP,RUNCHK ;NO, RETURN IF JOB STOPPED AND NO ACTIVE DEVICES
167 000115 603100 400000 CHKC01: TLNE TAC1,NOCORE ;DOES THIS COMMAND NEED CORE?
168 000116 254000 000137' JRST COMGO ;NO. GO DISPATCH
169 000117 326340 000125' JUMPN PROG,CHKXPN ;YES, IS CORE IN MEMORY?
170 000120 265040 000134' JSP TAC,COMER ;NO, PRINT "NO CORE ASSIGNED"
171 000121 472364 041636 ASCIIZ /NO CORE ASSIGNED
000122 512124 040646
000123 516230 747212
172 000124 420321 200000 /
  
```

```

173 000125 607100 000004 CHKXPN: T,NA TAC1,P,EXPN      ;DOES THIS COMMAND NEED CORE TO BE EXPANDED?
174 000126 254000 000137' JRST COMGO          ;NO
175 000127 574007 000000 HLRE IOS,JOBCU(JDAT)    ;YES, IS CORE STILL COMPRESSED(SAVE DID NOT GO
176                                     ; TO COMPLETION)
177 000130 345000 000137' AOJGE IOS,COMGO      ;LH=-2 DURING SAVE, WHEN CORE COMPRESSED
178                                     ;LH=-1 DURING SAVE OF HIGH SEG, OR GET OF LOW
179                                     ; OR HIGH SEG
180 000131 260140 002602' PUSHJ PDP,EXPAND    ;YES, EXPAND CORE FIRST
181 000132 255000 000000 JFCL              ;IGNORE ADDRESS CHECK ERROR, WE TRIED
182 IFN FTSWAP,<
183 000133 254000 000111' JRST CHKDLY        ;DELAY COMMAND BECAUSE COMMAND DECODER ACS ARE
184                                     ; ALL GONE, NEXT TIME JOBCU WILL BE 0
185 >
186
187 IFE FTSWAP,<
188 HRRR TAC1,OLYCM1
189 JRST COMDIS                          ;DELAY COMMAND TILL NEXT CLOCK TICK
190 >
191
192 000134 205100 000400 COMER: MOVSI TAC1,NOINCK
193 000135 202103 000000 MOVEM TAC1,(PDP)
194 000136 201100 000433' MOVEI TAC1,CERR      ;CALL ERROR MESSAGE ROUTINE
195
196 000137 205000 002000 COMGO: MOVSI IOS,CMWRQ
197 000140 627500 200000 TLZN T,CMWB
198 000141 412003 000000 ANDCAM IOS,(PDP)
199 000142 202504 000045' MOVEM T,JBSTSTS(ITEM) ;CLEAR CMWB; WAS JOB IN COMM WAIT
200                                     ;NO, CLEAR REQUEU BIT IN DISP. FLAGS
201
202 000143 201000 000000 COMDIS: MOVEI IOS,0
203 000144 260142 000000 PUSHJ PDP,(TAC1)    ;CLEAR IOS FOR SETTING DISPATCH ADDRESSES
                                         ;DISPATCH TO COMMAND SETUP ROUTINE.
    
```

```

204                                     ;RETURN FROM COMMAND SETUP ROUTINE
205
206 000145 262140 000013 COMRET: POP PDP,T1           ;RESTORE COMMAND FLAGS
207 000146 262140 000006      POP PDP,DEV DAT       ;RESTORE TTY DDB ADDRESS.
208                                     IFN FTTYSER,<
209                                     EXTERN TITAKR,TISYNC
210 000147 135406 000000      LDB TEM,TITAKR(DEV DAT)   ;GET BREAK CHARACTER
211 000150 260140 000222'    PUSHJ PDP,SKPBRK        ;SKIP TO BREAK CHAR
212 000151 370006 000000      SOS TISYNC(DEV DAT)     ;REDUCE LINE COUNT
213                                     >
214 000152 205500 400000      MOVSI T,400000
215 000153 135040 000000      LDR TAC,PUNIT
216 000154 412501 000000      ANDCAM T,TTYTAB(TAC)    ;YES, TURN OFF SIGN BIT
217 000155 370000 000000      SOS COMCNT
218 000156 326200 000161'    JUMPN ITEM,COMRT1      ;DID AN ERROR OCCUR?
219 000157 607540 200000      TLNN T1,NOJOBN        ;I.E., ITEM=0 AND NOJOBN=0?
220 000160 205540 002401      MOVSI T1,NOINCK+ERRFLG+CMWRQ ;YES, PRINT ERROR MESSAGE ONLY, AND
221                                     ; REQUEUE JOB IF NECESSARY.
222
223 000161 200504 000142' COMRT1: MOVE T,JRTSTS(ITEM)   ;JOB STATUS WORD
224 000162 607540 000400      TLNN T1,NOINCK        ;SUPPRESS JOB INIT, CHECK?
225 000163 663500 040000      TLOE T,JNA           ;NO, IS JOB INIT BIT ALREADY SET?
226 000164 254000 000200'    JRST PCRLF           ;YES.
227 000165 202504 000161'    MOVEM T,JBTSTS(ITEM)  ;NO, SO SET IT THIS COMMAND
228 000166 260140 000000      PUSHJ PDP,TTYATI      ;ATTACH TTY TO JOB
229 000167 255000 000000      JFCL                 ;IGNORE IF CAN NOT(SHOULD NEVER HAPPEN)
230 000170 260140 002025'    PUSHJ PDP,INLMES      ;AND PRINT "JOB "
231 000171 452370 220000      ASCII7 /JOB /
232 000172 200040 000004      MOVE TAC,ITEM        ;PRINT JOB NUMBER
233 000173 260140 000000      PUSHJ PDP,RADX10
234 000174 260140 002032'    PUSHJ PDP,PRSPC
235 000175 201040 000000      MOVEI TAC,CONFIG     ;PRINT SYSTEM CONFIGURATION NAME
236 000176 260140 002017'    PUSHJ PDP,CONMES
237                                     IFN FTLOGIN, <
238 000177 260140 002016'    PUSHJ PDP,CRLF
239                                     >
240                                     IFE FTLOGIN, <
241                                     PUSHJ PDP,PRSPC
242                                     PUSHJ PDP,DAYTM1
243                                     >

```

```

244 000200 603540 000001 PCRLF: TLNE T1,ERRFLG      ;DID AN ERROR OCCUR?
245 000201 260140 002030' PUSHJ PDP,PROM      ;YES, APPEND ?
246 000202 607540 000200 TLNN T1,NOCRLF     ;SUPPRESS CRLF?
247 000203 260140 002016' PUSHJ PDP,CRLF      ;NO
248 000204 607540 000100 TLNN T1,NOPER     ;SUPPRESS PRINTING PERIOD?
249 000205 260140 002041' PUSHJ PDP,PRPER     ;NO
250 000206 322200 000213' JUMPE ITEM,PCRLF1 ;JOB DOES NOT RUN IF ERROR OR NO JOB NO, ASSIGNED
251 000207 603540 000040 TLNE T1,TTYRNU     ;JOB TO RUN WHEN TTY FINISHED TYPING?
252                                     ; COMMAND RESPONSE (TTY TO USER MODE)?
253 000210 260140 000000 PUSHJ PDP,TTYUSR    ;YES, CALL SCANNER ROUTINE
254 000211 603540 000020 TLNE T1,TTYRNC     ;NO, JOB TO RUN AND REMAIN IN MONITOR MODE?
255 000212 260140 000000 PUSHJ PDP,SETRUN   ;YES, CALL SCANNER ROUTINE
256 000213 607540 000010 PCRLF1: TLNN T1,NOMESS ;IS THERE A MESSAGE?
257 000214 260140 000000 PUSHJ PDP,TTYSTR    ;YES, START TTY TYPING IT OUT
258                                     ;
259                                     IFN FTSWAP, <
260 000215 135200 000000 LDB ITEM,PJOBN    ;GET JOB NUMBER FROM TTY DEVICE DATA BLOCK
261 000216 322200 000000 JUMPE ITEM,CPOPJ  ;REQUEUE JOB AFTER COMMAND WAIT OR ERROR?
262 000217 603540 002000 TLNE T1,CMWRQ
263 000220 254000 000000 JRST REQUE      ;YES
264 >
265 000221 263140 000000 POPJ PDP,
266
267                                     IFN FITTYSER,< EXTERN SPCHEK,TITAKR,GETCHR,TPOPJ
268
269 000222 261140 000001 SKPBRK: PUSH PDP,TAC
270 000223 260140 000000 SKPBR2: PUSHJ PDP,SPCHEK
271 000224 254000 000227' JRST SKPBR1
272 000225 603040 000000 TLNE TAC,BREAKB   ;IS THIS CHAR THE BREAK?
273 000226 254000 000000 JRST TPOPJ       ;YES, RESTORE TAC, RETURN
274 000227 260140 000000 SKPBR1: PUSHJ PDP,GETCHR
275 000230 326400 000223' JUMPN TEM,SKPBR2 ;LOOK FURTHER FOR BREAK
276 000231 254000 000226' JRST TPOPJ     ;UNLESS NO MORE CHARS
277 >

```

```

278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321

```

TABLE OF CONSOLE COMMANDS

INTERNAL FTATTACH,FTTIME,FTTALK,FTEXAMINE,FTLOGIN,FTREASSIGN
 INTERNAL FTFINISH,FTCCL

BITS CHECKED BEFORE DISPATCHING TO COMMAND SETUP ROUTINE

```

400000 NOCORE=400000 ;NO CORE NEEDED FOR COMMAND
200000 NOJOBN=200000 ;NO JOB NUMBER NEEDED FOR COMMAND
100000 NOLOGIN=100000 ;JOB DOES NOT NEED TO BE ALREADY LOGGED IN.
040000 NOACT=400000 ;COMMAND MUST BE DELAYED IF JOB HAS ACTIVE DEVICES.
020000 NOTRAN=200000 ;COMMAND MUST BE DELAYED IF SWAPPING TRANSFER IN PROGRESS
010000 NORUN=100000 ;AN EXPLICIT *C MUST BE TYPED BY USER BEFORE COMMAND
; IS EXECUTED IF JOB IS RUNNING
; JOB MUST BE SWAPPED IN IF IT HAS LOGICAL CORE
004000 INCOR=4000 ;USED ONLY BY COMMAND DISPATCH
; DOES NOT APPEAR IN COMMAND TABLE
024000 INCORE=4000;NOTRAN ;COMMAND MUST BE DELAYED IF JOB HAS LOGICAL CORE
; WHICH IS ON DISK OR ON ITS WAY IN OR OUT.
; JOB WILL BE SWAPPED IN (SEE DLYCOM)
; AND IS NOT SITTING QUIETLY IN CORE(IE NO SWAPPING TRANSFER)
000004 PLSXPN=4 ;CORE MUST BE EXPANDED IF STILL COMPRESSED BY SAVE NOT GOING
; TO COMPLETION(CONTROL C OR DEVICE OK?)
; MUST BE USED IN CONJUNCTION WITH COMMANDS WHICH
; NEED CORE AND NEED IT IN CORE

```

BITS CHECKED AFTER RETURN FROM COMMAND SETUP ROUTINE

```

002000 CMWRQ=2000 ;REQUEUE JOB AFTER COMMAND WAIT
001000 NODATE=1000 ;DON'T PRINT DATE DURING JOB INTERROGATION
000400 NOINCK=400 ;NO CHECK FOR JOB INITIALIZATION (JNA=0)
000200 NOCRLF=200 ;NO PRINTING OF CRLF
000100 NOPER=100 ;NO PRINTING OF PERIOD
000040 TTYRNU=40 ;SET TTY TO USER MODE AND START JOB
; WHEN COMMAND RESPONSE STOPS TYPING
000020 TTYRNC=20 ;KEEP TTY IN COMMAND MODE AND START JOB
; WHEN COMMAND RESPONSE STOPS TYPING
000010 NOMESS=10 ;NO COMMAND RESPONSE EVER, DO NOT CALL TTYSTR
;4 TAKEN ABOVE BY 'PLSXPN' BIT
000001 ERRFLG=1 ;COMMAND ERROR

```

```

322            DEFINE NAMES<
323            C START,START,NOPER!TTYRNU!INCORE!NOACT!NORUN
324            C HALT,STOP,NOCORE!NOJOB!NOLOGIN!NOINCK!CMWRQ
325            C KJOB,KJOB,NOCORE!NOJOB!NOINCK!NOLOGIN!NOACT!NORUN!NOPER!NOCRLF!NOMESS!NOTRAN
326
327            C <>,CBLANK,NOCORE!NOJOB!NOLOGIN!NOINCK!NOCRLF
328            C R,RCOM,NOCORE!NOPER!TTYRNU!NOCRLF!NOACT!NORUN!NOTRAN
329            C RUN,RUNCOM,NOCORE!NOPER!TTYRNU!NOCRLF!NOACT!NORUN!NOTRAN
330            C CORE,CORE,NOCORE!NORUN!CMWRQ!INCORE!NOACT!NOTRAN
331            C GET,GET,NOCRLF!NOPER!TTYRNC!NOCORE!NOACT!NORUN!NOTRAN
332            C SAVE,SAVE,NOCRLF!NOPER!TTYRNC!INCORE!NOACT!NORUN!PLSXP
333            IFN FT2REL,<
334            C SSAVE,SSAVE,NOCRLF!NOPER!TTYRNC!INCORE!NOACT!NORUN!PLSXP
335            >
336            C CONTINUE,CONT,NOPER!TTYRNU!INCORE!NORUN
337            IFN FTXAMINE,<
338            C D,DCOM,CMWRQ!INCORE
339            C E,ECOM,NOCRLF!CMWRQ!INCORE
340            >
341            C PJOB,PJOB,NOCORE
342            C ASSIGN,ASSIGN,NOCORE
343            C DEASSI,DEASSIGN,NOCORE!NOJOB!NOINCK
344            C DOT,DOTGO,NOPER!TTYRNU!INCORE!NORUN!PLSXP
345            IFN FTFINISH,<
346            C FINISH,CFINI,TTYRNC!NOACT!INCORE!NORUN
347            >
348            C REENTER,REENTER,TTYRNU!NOPER!INCORE!NORUN!PLSXP
349            C CSTART,STARTC,TTYRNC!INCORE!NOACT!NORUN!PLSXP
350            C CCONTINUE,CONTC,TTYRNC!INCORE!NORUN
351            IFN FTATTACH,<
352            C DETACH,DETACH,NOPER+NOCRLF+NOCORE!NOJOB!NOINCK
353            C ATTACH,ATTACH,NOCORE!NOJOB!NOLOGIN!NOINCK
354            >
355            C DAYTIME,DAYTIM,NOCORE!NOJOB!NOLOGIN!NOINCK
356            IFN FTTIME,<
357            C TIME,RUNTIM,NOCORE!NOLOGIN!NOINCK
358            >
359            C RESOURCES,FREDEV,NOCORE!NOJOB!NOLOGIN!NOINCK
360            IFN FTLOGIN,<
361            C SCHEDULE,SKEDUL,NOCORE!NOJOB!NOLOGIN!NOINCK
362            >
363            IFN FTTALK,<
364            C TALK,TALK,NOCRLF+NOPER+NOCORE!NOJOB!NOLOGIN!NOMESS!NOINCK
365            >
366            IFN FTLOGIN,<
367            C LOGIN,CLOGIN,NOCORE!NOLOGIN!NOPER!TTYRNU!NOCRLF!NOACT!NORUN!NOTRAN
368            >
369            IFE FTLOGIN,<
370            C LOGIN,CPOPJ,NOCORE!NOACT!NORUN!NOTRAN
371            >
372            IFN FTREASSIGN,<
373            C REASSI,REASS,NOCORE!NOACT!INCORE!CMWRQ!NORUN
374            >

```

```
375 C HELP,HELP,NOCORE!NOLOGIN!NOPER!TTYRNU!NOCRLF!NOACT!NORUN!NOTRAN
376 C SYSTAT,SYSTAT,NOCORE!NOLOGIN!NOACT!NORUN!NOCRLF!NOPER!TTYRNU!NOTRAN
377 IFN FTCCCL,<
378 C COMPILE,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
379 C CREATE,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
380 C CREFF,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
381 C DEBUG,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
382 C DELETE,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
383 C DIRECT,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
384 C EDIT,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
385 C EXECUTE,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
386 C LIST,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
387 C LOAD,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
388 C MAKE,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
389 C RENAME,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
390 C TECO,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
391 C TYPE,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
392 C CTEST,CCLRUN,NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN
393 >
394 >
```

```

395          ;GENERATE TABLE OF SIXBIT COMMAND NAMES
396
397          DEFINE C(A,B,D) <
398              <SIXBIT /A/>
399          >
400
401          COMTAB: NAMES+
402          000232      636441  626400    <SIXBIT /START/>
403          000233      504154  640000    <SIXBIT /HALT/>
404          000234      535257  420000    <SIXBIT /KJOB/>
405          000235      000000  000000    <SIXBIT //>
406          000236      620000  000000    <SIXBIT /R/>
407          000237      626556  000000    <SIXBIT /RUN/>
408          000240      435762  450000    <SIXBIT /CORE/>
409          000241      474564  000000    <SIXBIT /GET/>
410          000242      634166  450000    <SIXBIT /SAVE/>
411          000243      636341  664500    <SIXBIT /SSAVE/>
412          000244      435756  645156    <SIXBIT /CONTINUE/>
413          000245      440000  000000    <SIXBIT /D/>
414          000246      450000  000000    <SIXBIT /E/>
415          000247      605257  420000    <SIXBIT /PJOB/>
416          000250      416363  514756    <SIXBIT /ASSIGN/>
417          000251      444541  636351    <SIXBIT /DEASSI/>
418          000252      444464  000000    <SIXBIT /DDT/>
419          000253      465156  516350    <SIXBIT /FINISH/>
420          000254      624545  566445    <SIXBIT /REENTER/>
421          000255      436364  416264    <SIXBIT /CSTART/>
422          000256      434357  566451    <SIXBIT /CCONTINUE/>
423          000257      444564  414350    <SIXBIT /DETACH/>
424          000260      416464  414350    <SIXBIT /ATTACH/>
425          000261      444171  645155    <SIXBIT /DAYTIME/>
426          000262      645155  450000    <SIXBIT /TIME/>
427          000263      624563  576562    <SIXBIT /RESOURCES/>
428          000264      634350  454465    <SIXBIT /SCHEDULF/>
429          000265      644154  530000    <SIXBIT /TALK/>
430          000266      545747  515600    <SIXBIT /LOGIN/>
431          000267      624541  636351    <SIXBIT /REASSI/>
432          000270      504554  600000    <SIXBIT /HFLP/>
433          000271      637163  644164    <SIXBIT /SYSTAT/>
434          000272      435755  605154    <SIXBIT /COMPILE/>
435          000273      436245  416445    <SIXBIT /CREATE/>
436          000274      436245  460000    <SIXBIT /CREF/>
437          000275      444542  654700    <SIXBIT /DEBUG/>
438          000276      444554  456445    <SIXBIT /DELETE/>
439          000277      445162  454364    <SIXBIT /DIRFCT/>
440          000300      454451  640000    <SIXBIT /EDIT/>
441          000301      457045  436564    <SIXBIT /EXECUTE/>
442          000302      545163  640000    <SIXBIT /LIST/>
443          000303      545741  440000    <SIXBIT /LOAD/>
444          000304      554153  450000    <SIXBIT /MAKE/>
445          000305      624556  415545    <SIXBIT /RENAME/>
446          000306      644543  570000    <SIXBIT /TFCC/>
447          000307      647160  450000    <SIXBIT /TYPE/>

```

COMCON - COMMAND DECODER AND SAVEGET ROUTINES - V433 MACRO,V36 19102 4-JUN-69 PAGE 22-1
/RCC 03 JUN 69

448 000310 436445 636400 <SIXBIT /CTEST/>
449 000057 DISPL=-,COMTAB ILENGTH OF TABLE

```

450                                     ;GENERATE THE DISPATCH TABLE PLUS SPECIAL BITS
451
452                                     DEFINE C(A,B,D) <
453                                     ?      R      (D)
454                                     >
455                                     DISP:  NAMES+
456 000311      074140  000436'  Z      START      (NOPER!TTYRNU!INCORE!NOACT!NORUN)
457 000312      702400  000454'  Z      STOP      (NOCORE!NOJORN!NOLOGIN!NOINCK!CMWRQ)
458 000313      770710  000410'  Z      KJOB      (NOCORE!NOJORN!NOINCK!NOLOGIN!NOACT!NORUN!NOPER!NOCRLF!NOMESS!NOTRAN)
459 000314      700600  001570'  Z      CBLANK    (NOCORE!NOJORN!NOLOGIN!NOINCK!NOCRLF)
460 000315      470340  000617'  Z      RCOM      (NOCORE!NOPER!TTYRNU!NOCRLF!NOACT!NORUN!NOTRAN)
461 000316      470340  000625'  Z      RUNCOM    (NOCORE!NOPER!TTYRNU!NOCRLF!NOACT!NORUN!NOTRAN)
462 000317      476000  000473'  Z      CORE      (NOCORE!NORUN!CMWRQ!INCORE!NOACT!NOTRAN)
463 000320      470320  000536'  Z      GET       (NOCRLF!NOPER!TTYRNC!NOCORE!NOACT!NORUN!NOTRAN)
464 000321      074324  000531'  Z      SAVE      (NOCRLF!NOPER!TTYRNC!INCORE!NOACT!NORUN!PLSXPN)
465 000322      074324  000532'  Z      SSAVE     (NOCRLF!NOPER!TTYRNC!INCORE!NOACT!NORUN!PLSXPN)
466 000323      034140  000463'  Z      CONT      (NOPER!TTYRNU!INCORE!NORUN)
467 000324      026000  001504'  Z      DCOM      (CMWRQ!INCORE)
468 000325      026200  001453'  Z      ECOM      (NOCRLF!CMWRQ!INCORE)

469 000326      400000  000405'  Z      PJOB      (NOCORE)
470 000327      400000  000633'  Z      ASSIGN    (NOCORE)
471 000330      600400  000757'  Z      DEASSIGN   (NOCORE!NOJOB!NOINCK)
472 000331      034144  000446'  Z      DDTGN     (NOPER!TTYRNU!INCORE!NORUN!PLSXPN)
473 000332      074020  001574'  Z      CFINI     (TTYRNC!NOACT!INCORE!NORUN)
474 000333      034144  000444'  Z      REENTER   (TTYRNU!NOPER!INCORE!NORUN!PLSXPN)
475 000334      074024  000436'  Z      STARTC   (TTYRNC!INCORE!NOACT!NORUN!PLSXPN)
476 000335      034020  000463'  Z      CONTC    (TTYRNC!INCORE!NORUN)
477 000336      600700  001247'  Z      DETACH    (NOPER+NOCRLF+NOCORE!NOJOB!NOINCK)
478 000337      700400  001136'  Z      ATTACH    (NOCORE!NOJORN!NOLOGIN!NOINCK)
479 000340      700400  001312'  Z      DAYTIM    (NOCORE!NOJORN!NOLOGIN!NOINCK)
480 000341      500400  001350'  Z      RUNTIM    (NOCORE!NOLOGIN!NOINCK)
481 000342      700400  001624'  Z      FREDEFV   (NOCORE!NOJORN!NOLOGIN!NOINCK)
482 000343      700400  001555'  Z      SKEDUL    (NOCORE!NOJORN!NOLOGIN!NOINCK)
483 000344      700710  001432'  Z      TALK      (NOCRLF+NOPER+NOCORE!NOJOB!NOLOGIN!NOMESS!NOINCK)
484 000345      570340  000600'  Z      CLOGIN    (NOCORE!NOLOGIN!NOPER!TTYRNU!NOCRLF!NOACT!NORUN!NOTRAN)
485 000346      476000  001015'  Z      PEASS     (NOCORE!NOACT!INCORE!CMWRQ!NORUN)
486 000347      570340  000554'  Z      HELP      (NOCORE!NOLOGIN!NOPER!TTYRNU!NOCRLF!NOACT!NORUN!NOTRAN)
487 000350      570340  000557'  Z      SYSTAT    (NOCORE!NOLOGIN!NOACT!NORUN!NOCRLF!NOPER!TTYRNU!NOTRAN)
488 000351      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
489 000352      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
490 000353      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
491 000354      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
492 000355      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
493 000356      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
494 000357      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
495 000360      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
496 000361      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
497 000362      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
498 000363      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
499 000364      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
500 000365      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
501 000366      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)
502 000367      474340  000570'  Z      CCLRUN    (NOCORE!NOPER!TTYRNU!NOCRLF!INCORE!NOACT!NORUN)

```

COMCON - COMMAND DECODER AND SAVEGET ROUTINES - V433 MACRO,V36 19102 4-JUN-69 PAGE 23-1
/RCC 03 JUN 69

503
504 000370 700400 001573' XWD NOCORE+NOJOBN+NOLOGIN+NOINCK,NOCOM
505

```

506
507      ;CALLED FROM COMMAND DECODER WHICH IS CALLED FROM CLOCK ROUTINE
508      ;WITH FOLLOWING AC'S SET:
509      ;TAC= BYTE POINTER TO COMMAND STRING(SPACE OR CR AFTER COMMAND NAME)
510      ;ITEM = JOB NUMBER
511      ;DAT = BYTE POINTER TO LAST OUTPUT CHARACTER
512      ;DEV DAT= ADDRESS OF DEVICE DATA BLOCK INITIATING COMMAND
513      ;JDAT = ADDRESS OF JOB DATA AREA, = 0 IF NO JOB AREA
514      ;PROG=XWD PROTECTION,RELOCATION; 0 IF NO CORE
515      ;IOS=0
516
517      ;UPON RETURN FROM COMMAND SETUP ROUTINE, A CR-LF IS ADDED IF
518      ;NEEDED AND TTY IS STARTED
519
520      ;SINCE THE COMMAND DECODER IS CALLED FROM THE CLOCK ROUTINE
521      ;COMMAND SETUP ROUTINES MUST RUN TO COMPLETION QUICKLY
522      ;IF A COMMAND FUNCTION CANNOT DO THIS, IT MUST JUST SET
523      ;THE JOB TO A RUNABLE STATUS AND RETURN IMMEDIATELY
524
525
526      EXTERN TTYFNU
527      000371 322340 000216' COR0:  JUMPE PROG,CPOPJ      ;RETURN IF JOB DOES NOT HAVE CORE
528
529      000372 265100 000000      JSP TAC1,MONJOB      ;SET TO RUN MONITOR JOB(PC IN EXEC MODE)
530      ; RETURN HERE AT UUD LEVEL(NO ACS SET UP)
531      000373 201100 000000      MOVEI TAC1,ESTOP      ;PUT ERROR STOP ON END OF PDL
532      000374 265040 000000      JSP TAC,MONSTR      ;START MONITOR JOB BY SETTING UP ACS AND
533      000375 260140 000000      JOR1:  PUSHJ PDP,RESET  ;RELEASE ALL IO DEVICES ASSIGNED TO THIS JOB
534      000376 260140 000000      PUSHJ PDP,TTYFNU      ;FIND TTY FOR THIS JOB(SETUP ITEM WITH JOB NO,)
535      IFN FTSWAP,<
536      EXTERN ZERSWP,NOCOR0
537      000377 260140 000000      PUSHJ PDP,NOCOR0      ;PUT JOB IN NO CORE 0
538      000400 260140 000000      PUSHJ PDP,ZERSWP      ;CLEAR SWAP SPACE (IF ANY) FOR LOW SEG
539      ; AND CLEAR SWP,SHF,IMGIN,IMGOUT FOR LOW SEG
540      >
541      IFN FT2REL,<
542      FXTERN KILHGH
543      000401 260140 000000      PUSHJ PDP,KILHGH      ;HIGH SEG FOR THIS USER, RETURN CORE
544      ; REMOVE HIGH SEG FROM THIS USER LOGICAL ADDRESSING SPA
545      CE
546      ; IF NO OTHER USERS IN CORE ARE USING IT
547      ; RETURN DISK SPACE IF NO LONGER SHARABLE HIGH SEG
548      >
549      000402 201040 000000      MOVEI TAC,0
550      000403 370003 000000      SOS (PDP)
551      000404 254000 000000      JRST CORE1
552      ;RETURN BOTH PHYSICAL AND LOGICAL CORE
553      ;NEGATE SKIP RETURN WHICH CORE1 WILL DO
554      ;GO DO IT AND RETURN TO CALLER

```

```

552
553           ; "PJOB" PRINT JOB NUMBER OF JOB TTY IS ATTACHED TO
554
555           INTERNAL PJOB,DECLF
556
557 000405 200040 000004 PJOB:  MOVE TAC,ITEM           ;JOB NO.
558 000406 260140 000173' DECLF:  PUSHJ PDP,RADX10       ;PRINT TAC AS DEC, THEN CRLF
559 000407 254000 002016'          JRST CRLF
560
561
562           ; "KJOB" KILL ATTACHED JOB
563
564           EXTERNAL JBTSTS,TTYKIL,CPOPJ
565           INTERNAL FTTIME,JOBKL
566
567 000410 322200 000434' KJOB:  JUMPE ITEM,JOBKB         ;WAS JOB INITIALIZED?
568                                IFN  FTLOGIN, <
569 000411 627500 100004          TLZN  T,JLOG+JACCT       ;TEST JACCT ALSO IN CASE COMMAND WAS DELAYED
570 000412 254000 000422'          JRST  KJOB2           ;IF JOB NOT LOGGED IN
571 000413 661500 100000          TLO   T,JACCT         ;DISABLE +C
572 000414 202504 000165'          MOVEM T,JBTSTS(ITEM)
573                                IFN  FTSWAP, <
574 000415 260140 000565'          PUSHJ PDP,ALOGN1       ;ATTACH TTY
575                                >
576 000416 205040 000050          MOVSI  TAC,TTYRNU+NOMESS
577 000417 432043 777777          XORM  TAC,-1(PDP)       ;SET TTYRNU AND CLEAR NOMESS FOR COMRET
578 000420 200040 003053'          MOVE  TAC,[SIXBIT /LOGOUT/] ;NAME OF CUSP
579 000421 254000 000621'          JRST  ARCOM
580                                >
581 000422                                KJOB2:
582                                IFE  FTSWAP+FTLOGIN,<
583                                JUMPE PROG,JOBKA         ;YES, DOES JOB HAVE CORE IN MEMORY?
584                                >
585 000422 260140 002221'          PUSHJ PDP,GETMIN       ;GET MINIMAL JOB AREA ON DISK OR CORE
586 000423 322340 001677'          JUMPE PROG,DLYCM1     ;DELAY COMMAND IF CORE ASSIGNED ON DISK
587 000424 265100 000372'          JSP  TAC1,MONJOB      ;YES, SCHEDULE MONITOR JOB(PC IN EXEC MODE)
588                                ; RETURN HERE AT UVO LEVEL WHEN SCHEDULED

```

```

589 000425 201100 000373' JOBKL: MOVEI TAC1,ESTOP      IPUT ESTOP ON END OF PDL
590 000426 265040 000374'   JSP TAC,MONSTR      IGO SETUP ACS AND PD LIST
591 000427 260140 000375'   PUSHJ PDP,JOBI     IFLUSH CORE AFTER RELEASING DEVICES
592 000430
593
594
595
596
597
598 000430 260140 002060'   JOBKA:
599 000431 260140 000000'     IFN FTTIME,<
600 000432 254000 000000'     IFE FTLOGIN,<
601
602
603
604
605
606 000433 254000 002015'   >>
607
608 000434 402003 777777'   PUSHJ PDP,DEASTY   IDEASSIGN ALL BUT TTY
609 000435 254000 001231'   PUSHJ PDP,TTYKIL  IRETURN TTY TO VIRGIN STATE
                                JRST KSTOP      ICLEAR JOB STATUS WORD AND STOP JOB
                                IERROR IN COMMAND
606 000433 254000 002015' CERR: JRST ERRMES
608 000434 402003 777777' JOBKB: SETZM -1(PDP)   ICLEAR NOJOBN SO COMRET WILL PRINT ERROR MSG.
609 000435 254000 001231'   JRST ATT4         I"NOT A JOB"
  
```

```

610                                   ; "START L" OR "START" - START AT LOC, L OR STARTING ADDRESS
611                                   INTERNAL FT2REL
612                                   EXTERNAL JOBSA
613
614 000436                            START:                                   ;SAME AS CSTART, DIFF BY COMTAB BITS
615                                                                           ; WHICH PUT TTY IN MONITOR OR USER MODE
616
617
618                                   ; "CSTART L" OR "CSTART" - START AT LOC, L(TTY IN COMMAND MODE)
619
620
621 000436 260140 002073' STARTC: PUSHJ PDP,OCTIN                   ;CONVERT POSSIBLE OCTAL NO ARG.
622                                   IFE FT2REL,<
623                                                                            SKIPA TAC1,JOBSA(JDAT)   ;NO START ADR, SPECIFIED RETURN
624                                   >
625                                   IFN FT2REL,<
626 000437 254000 000443'                                   JRST SNOARG           ;NO ARG SPECIFIED RETURN
627                                   >
628 000440 254000 002000'                                   JRST COMERA           ;ILLEGAL CHARACTER
629                                   IFN FT2REL,<
630                                                                            EXTERN CHKMED
631 000441 260140 000000                                    PUSHJ PDP,CHKMED       ;CHECK TO SEE IF HIGH SEG WHICH IS SHARABLE
632                                                                                   ; IF YES, TURN ON USER MODE WRITE PROTECT
633                                                                                   ; FOR THIS USER, AND SET MEDDLE BIT SO HE CANNOT
634                                                                                   ; TURN UWP OFF.>
635 000442 254000 000447'                                   JRST CHKSTR           ;START JOB WITH PC IN USER MODE,
636                                                                                   ; IF START ADR IS NON-ZERO
637                                   IFN FT2REL,<
638 000443 334107 000000                                    SNOARG: SKIPA TAC1,JOBSA(JDAT)   ;NO, ARG SPECIFIED, USE START ADR SUPPLIFD
639                                                                                   ; BY PROGRAM END STATEMENT AND STORED BY LINKING LOADER
640                                                                                   ; SKIP INTO REENTER, START WITH PC IN USER MODE
641                                   >

```

```

642                               ; "REENTER" - REENTER USER PROGRAM
643
644                               EXTERNAL JOBREN
645
646   000444   200107   000000   REENTER: MOVE TAC1, JOBREN(JDAT)   ; GET REENTER ADDRESS FROM JOB DATA AREA
647   000445   254000   000447                    JRST CHKSTR                   ; START JOB WITH PC IN USER MODE
648                                                                               ; IF START ADR IS NON-ZERO
649
650                               ; "DDT" - START EXECUTION AT DDT IN USER AREA
651
652                               EXTERN JOBDT
653
654   000446   200107   000000   DDTGO:  MOVE TAC1, JOBDT(JDAT)   ; DDT STARTING ADR, IN JOB DATA AREA
655                                                                               ; FALL INTO CHSTR
656
657                               ; ROUTINE TO CHECK TO SEE IF STARTING ADDRESS IS NON-ZERO, AND START USER IF OK
658
659   000447   326100   000000   CHKSTR: JUMPN TAC1, USTART           ; IS IT NON-ZERO?, IS YES
660                                                                               ; STORE OLD PC IN JOBOPC IN JOB DATA AREA
661
662   000450   265040   002015                    JSP TAC, ERRMES                   ; THEN START WITH PC IN USER MODE
663   000451   472364   051650                    ASCII /NO START ADR           ; NO, PRINT "NO START ADR"
664   000452   406452   420202
664   000453   422441   505000   /
  
```

```

665
666           ; "STOP" OR "<CONTROL>C"
667           ;SCANNER ROUTINES DUMMY UP STOP WHEN CONTROL C TYPED IN
668           ;STOP MUST BE DELAYED IF THIS JOB IS SYSTEM TAPE USER
669           ;AND SYSTEM TAPE IS ACTIVE, OTHERWISE, THE JOB WILL NOT BE
670           ;STOPPED WHEN DONE USING THE SYSTEM TAPE,
671
672           INTERNAL FTDISK
673           EXTERNAL STUSER,JOB,PJBSTS,IOWQ,JOBDAC
674
675 000454 322200 000462' STOP:  JUMPE  ITEM,STOPB
676 000455 316200 000000'          CAMN ITEM,STUSER          ;IS THIS SYSTEM TAPE USER?
677 000456 254000 001577'          JRST CFINS             ;YES, DO FINISH SYS COMMAND
678           IFN FTDISK,<
679
680 000457 260140 001705'          PUSHJ PDP,STOPCK          ;CAN THE JOB STOP?
681 000460 254000 001677'          JRST DLYCM1             ;NO. DELAY IT.
682           >
683 000461 260140 000000'          PUSHJ PDP,STOP1           ;STOP JOB
684           STOPB:  IFE FTTTYSER,<
685           JSP TAC,CONMES          ;PRINT "*C CRLF CRLF PERIODS"
686           ASCIIZ /*C
687           /
688           >
689 000462 263140 000000' IFN FTTTYSER,< POPJ  PDP,0>
690
691           ; "CONTC" - CONTINUE EXECUTION(TTY REMAINS IN COMMAND MODE)
692
693 000463           CONTC:          ;SAME AS CONT
694
695           ; "CONT" - CONTINUE EXECUTION FROM WHERE LEFT OFF
696           EXTERNAL JBTSTS
697
698 000463 205100 020000' CONT:  MOVSI TAC1,JERR
699 000464 616104 000414'          TDNN TAC1, JRTSTS(ITEM) ;IS JOB ERROR BIT SET?
700 000465 263140 000000'          POPJ PDP,             ;COMMAND DECODER WILL DO REST
701 000466 265040 002015'          JSP TAC,ERRMES          ;YES, PRINT CANT CONTINUE
702
703 000467 416031 623650          ASCIIZ /CAN'T CONTINUE
704 000470 202071 747250
705 000471 446352 542432
706 000472 050000 000000' /

```



```

736 000507          COR2:  IFN FTTYSER,<
737 000507 200303 777776      MOVE DEVDAT,-2(PDP)      ;RESTORE TTY DDB ADR
738                                     >
739 000510 260140 002025'    PUSHJ PDP,INLMES      ;PRINT ? FOR PATCH
740                                     ASCIZ /?
741 000511 374321 200000 /
742 000512 260140 002210'    COR3:  PUSHJ PDP,PRTSEG      ;PRINT SIZE OF LOW SEG
743                                     IFN FT2REL,<
744                                     EXTERN PRTHGH
745 000513 260140 000000      PUSHJ PDP,PRTHGH      ;PRINT SIZE OF HIGH SEG (+0 IF NONE)
746                                     >
747                                     ; NOTHING IF NOT 2 REG, MACHINE OR SOFTWARE
748 000514 260140 002025'    PUSHJ PDP,INLMES
749 000515 274000 000000      ASCIZ "/"
750
751                                     IFN FTSWAP,<
752                                     EXTERN CORTAL
753                                     MOVE TAC,CORTAL      ;NO. OF FREE 1K BLOCKS
754                                     >
755                                     IFN FTSWAP,<
756                                     EXTERNAL CORMAX
757 000516 200040 000000      MOVE TAC,CORMAX      ;PRINT MAX, NO. OF BLOCKS IN PHYSICAL CORE
758 000517 242040 777766      LSH TAC,-12      ;AVAILABLE TO A SINGLE USER(AS SET
759                                     >
760                                     ; BY BUILD AND/OR ONCE)
761 000520 260140 000406'    PUSHJ PDP,RADX10      ;PRINT NO. OF 1K BLOCKS AVAILABLE
762                                     IFN FTSWAP,<
763                                     JSP TAC,CONMES
764                                     ASCIZ /K CORE
765                                     /
766                                     >
767                                     IFN FTSWAP,<
768                                     EXTERN VIRTUAL
769 000521 260140 002025'    PUSHJ PDP,INLMES
770 000522 455010 347644      ASCIZ /K CORE
771 000523 424321 253222    VIR. CORE LEFT=/
772 000524 511344 041636
773 000525 512124 046212
774 000526 432507 500000
772 000527 200040 000000      MOVE TAC,VIRTUAL      ;PRINT AMOUNT OF FREE SWAP SPACE LEFT
773 000530 254000 000520'    JRST RADX10      ;IN DECIMAL
774                                     >

```

```

775           ; "SSAVE FILE-NAME[PROJ,,PROG,] CORE"
776           ;WORKS LIKE SAVE, EXCEPT THAT HIGH SEG IS SAVED AS SHARABLE(EXT=SHR)
777           ;INSTEAD OF NON-SHARABLE(EXT=HGHI)
778
779
780           ; "SAVE FILE-NAME[PROJ,,PROG,] CORE" - SAVES JOB AREA ON RETRIEVABLE DEVICE
781           ;ONLY A SAVE OR A GET IN PROGRESS FOR EACH JOB
782           ;NO ATTEMPT IS MADE TO SAVE DEVICE ASSIGNMENTS, AC'S, OR PC
783
784 000531      SAVE:IFN FT2REL,<
785 000531 661000 400000      TLO IOS,NSRBIT           ;SET FLAG FOR NON-SHARABLE EXT(HGH)
786 >
787 000532 541000 002314'  SSAVE:  HRR1 IOS,SAVJOB           ;SETUP TO RUN SAVJOB (IOS CLEARED BEFORE
788 000533 260140 001723'  PUSHJ PDP,CTEXT1       ;DISPATCH
789 000534 402004 000000      SETZM  JBTPRG(ITEM)       ;CLEAR JOB'S PROGRAM NAME FOR RCOM
790 000535 254000 002241'  JRST SGSET
791
792
793           ; "GET DEVICE:FILE-NAME[PROJ,,PROG,] CORE" - SETS UP JOB AREA FROM RETREIVABLE
794           ;DEVICE AND ASSIGNS CORE.
795
796
797 000536 201000 002406'  GET:   MOVEI IOS,GETJOB           ;SETUP TO RUN GETJOB
798 000537 260140 001723'  PUSHJ PDP,CTEXT1       ;GET DEVICE NAME
799 000540 402004 000534'  SETZM  JBTPRG(ITEM)       ;CLEAR NAME FOR RCOM
800 000541 254000 000630'  JRST RUNCM
801
802           ; JOB DATA AREA FOR MONITOR JOB
803
804           IFN FTLOGIN,<   ;AUTOMATIC LOGIN AFTER DSK REFRESH OR WHEN REFLAG SET
805 EXTERNAL      REFLAG
806
807 000542 202044 000000      LOGREF: MOVEM  TAC,PRJPRG(ITEM)       ;PRJ,PRG NRS. IN REFLAG
808 000543 205040 000004      MOVSI  TAC,JLOG
809 000544 436044 000464'    IORM   TAC,JBTPRG(ITEM)
810 000545 402003 777777      SETZM  -1(PDP)
811 000546 402000 000000      SETZM  REFLAG           ;CLEAR ALL FLAGS FOR COMRET
812 000547 263140 000000      POPJ   PDP,
813 >
  
```

```

814          ;"HELP" -HELP COMMAND
815
816          INTERNAL FTLOGIN
817
818          IFN FTLOGIN,<  EXTERNAL HELPPP
819 000550      HELP1:
820 000550      200040  000000      MOVE TAC,HELPPP
821 000551      607500  000004      TLNN T,JLOG
822 000552      254000  000562'    JRST ALOGIN
823 000553      263140  000000      POPJ  PDP,
824
825          >
826          HELP:  IFN FTLOGIN,<
827 000554      260140  000550'    PUSHJ  PDP,HELP1
828
829          >
829 000555      200040  003054'    MOVE   TAC,[SIXBIT /HELP/]    ;CUSP NAME
830 000556      254000  000621'    JRST   ARCOM  ;RUN IT
831
832          SYSTAT: IFN FTLOGIN,<
833 000557      260140  000550'    PUSHJ  PDP,HELP1
834
835          >
835 000560      200040  003055'    MOVE   TAC,[SIXBIT /SYSTAT/]  ;CUSP NAME
836 000561      254000  000621'    JRST   ARCOM  ;RUN IT
837
838          EXTERNAL JBTSTS,TTYATT
839
840          ALOGIN:
841 000562      INTERNAL FTLOGIN
842          IFN FTLOGIN,<  EXTERNAL PRJPRG
843 000562      202044  000542'    MOVEM TAC,PRJPRG(ITEM)
844 000563      205040  000004      MOVSI TAC,JLOG
845 000564      412044  000544'    ANDCAM TAC,JBTSTS(ITEM)
846
847          >
848          IFN FTSWAP,<EXTERNAL  TTYATT
849 000565      260140  000000      ALOGN1: PUSHJ PDP,TTYATT      ;MAKE SURE TTY IS ATTACHED IN CASE
850
851          >
851          JFCL
852 000566      255000  000000
853
854          >
854          POPJ PDP,
  
```

```

855          INTERN FTCL
856
857          IFN FTCL,<
858
859 000570 200040 003056' CCLRUN: MOVE TAC,[SIXBIT /COMPIL/] ICUSP NAME
860 000571 254000 000621' JRST ARCOM IRUN IT
861
862          >
863          I"LOGIN" - LOGIN COMMAND
864
865          INTERNAL FTLOGIN
866          IFN FTLOGIN,<
867
868          EXTERNAL SYSPP
869
870 000572 265040 002015' LOGDET: JSP TAC,ERRMES
871 000573 502310 540646 ASCIZ /PLEASE KJOB OR DETACH
      000574 425011 345236
      000575 411011 751100

      000576 422132 440606
872 000577 440321 200000 /
873 000600 603500 000004 CLOGIN: TLNE T,JLOG IFORCE USER TO LOGOUT BEFORE
874 000601 254000 000572' JRST LOGDET I LOGGING IN ANEW,
875          IFN FTSWAP,<
876          EXTERN VIRTAL,LOGSIZ
877 000602 200040 000527' MOVE TAC,VIRTAL IAMOUNT OF FREE VIRTUAL CORE LEFT
878 000603 301040 000000 CAIL TAC,LOGSIZ IIS THERE AT LEAST 2K FOR LOGIN?
879          I CUSTOMER CAN REDEFINE TO BE BIGGER THAN 2
880          I WITH MONGEN DIALOG
881 000604 254000 000607' JRST CLOG2 IYES,
882 000605 201200 000000 MOVEI ITEM,0 INO, SET JOB NUMBER TO 0 FOR COMMAND DECODER
883 000606 254000 000507' JRST COR2 IAND GO TELL USER AMOUNT OF CORE LEFT
884          >
885 000607 332040 000546' CLOG2: SKIPE TAC,REFLAG IREFLAG SET NON-ZERO FOR AUTOMATIC LOGIN?
886 000610 254000 000542' JRST LOGREF IYES, LOG USER IN WITHOUT RUNNING CUSP
887          I SO HE CAN RELOAD THE DISK AFTER REFRESH
888 000611 200040 000000 MOVE TAC,SYSPP ISET PRJPRG TO SYSPP
889 000612 260140 000562' PUSHJ PDP,ALOGIN IAUTOMATIC LOGIN
890 000613 205040 100000 MOVSI TAC,JACCT IFLAG LOGIN OR LOGOUT RUNNING AND
891 000614 436044 000564' IORM TAC,JBTSTS(ITEM);DISABLE *C,
892 000615 200040 003057' MOVE TAC,[SIXBIT /LOGIN/] ICUSP NAME
893 000616 254000 000621' JRST ARCOM IRUN IT
894          >
  
```

```

895          ; "R CUSPNAME CORE" - DOES "RUN SYS:CUSPNAME"
896
897 000617 201040 000000 RCOM: MOVEI TAC,0 ;NO FILE NAME, TTY WILL SUPPLY IT
898 000620 254000 000622' JRST ARCOM1
899
900 000621 260140 000222' ARCOM: IFN FTTYSER,< PUSHJ PDP,SKPBRK>
901 000622 202044 000540' ARCOM1: MOVEM TAC,JBTPRG(ITEM) ;STORE FILE NAME
902 000623 205100 637163 MOVSI TAC1,(SIXBIT /SYS/) ;READ FROM SYS DEVICE
903 000624 254000 000627' JRST RUNCO2
904          ; "RUN DEVICE:FILECPROJ,,PROG,J (CORE)"
905          ;DOES A CORE,GET,START ALL IN ONF
906          ;IF CORE ARG IS MISSING, SIZE IN DIRECTORY IS USED
907          ;JOB ON DISK OR IN CORE OR NO CORE, BUT NOT IN MIDDLE OF SWAP
908
909          EXTERNAL JOBDA
910
911 000625 260140 001723' RUNCOM: PUSHJ PDP,CTEXT1 ;GET DEVICE NAME FROM COMMAND STRING
912 000626 402004 000622' SETZM JBTPRG(ITEM) ;CLEAR FILE NAME
913 000627 201000 002413' RUNCO2:RUNC1: MOVEI IOS,RUNJOB
914 000630 260140 002221' RUNCM: PUSHJ PDP,GETMIN ;GET MINIMAL JOB AREA IN CORE OR DISK
915 000631 326340 002241' JUMPN PROG,SGSET ;WAS CORE ASSIGNED IN MEMORY? IF YES, GO SCANARGS
916          IFE FTSWAP,<
917          JRST COR2 ;NO, PRINT "OK CORE LEFT"
918          >
919          IFN FTSWAP,<
920 000632 254000 001676' JRST DLYCM ;NO, DELAY COMMAND UNTIL IN CORE
921          >
  
```

```

922           ; "ASSIGN DEVNAME" - ASSIGN DEVICE TO JOB AND GIVE IT LOGICAL NAME
923
924           EXTERNAL DEVLST,PJOB,SYSTAP
925
926 000633 260140 001723' ASSIGN: PUSHJ PDP, CTEXT1      ;GET FIRST ARGUMENT
927 000634 322100 002034'      JUMPE TAC1, NOTENF      ;NO ARGUMENT TYPED IF 0
928           IFN FTLOGIN,<EXTERN PRJPRG
929 000635 312100 003060'      CAME TAC1,[SIXBIT ,SYS,]
930 000636 254000 000642'      JRST ASSG6
931 000637 554544 000562'      HLRZ T1,PRJPRG(ITEM)
932 000640 306540 000001'      CAIN T1,1          ;PROJECT NR, 1?
933 000641 254000 000701'      JRST ASSG7          ;YES
934 000642           ASSG6:
935 >
936 000642 200540 000002           MOVE T1, TAC1          ;SAVE DEVICE NAME
937           IFN FTTYSER,<
938 000643 261140 000006           PUSH PDP,DEVDAT      ;SAVE TTY DDR
939 >
940           IFE FTTYSER,<
941           PUSH PDP, TAC          ;SAVE INPUT BYTE POINTER
942 >
943 000644 200040 000002           MOVE TAC,TAC1
944 000645 260140 000000           PUSHJ PDP,DEVPHY
945 000646 254000 000651'           JRST ASSG3
946 000647 201100 400000           MOVEI TAC1,ASSCON
947 000650 254000 000656'           JRST ASSG4
948 000651 201100 400000           ASSG3: MOVEI TAC1,ASSCON      ;SETUP ASSIGNED BY CONSOLE BIT
949 000652 554300 000000           HLRZ DEVDAT, DEVLST
950 000653 200046 000000           ASSG0: MOVE TAC,DEVNAM(DEVDAT)
951 000654 312546 000000           CAME T1,DEVNAM(DEVDAT)  ;DOES PHYSICAL NAME MATCH?
952 000655 254000 000665'           JRST ASSG1          ;NO
953 000656 316540 003061'           ASSG4: CAMN T1,[SIXBIT /DSK/]
954 000657 254000 000662'           JRST ASSG5          ;IS THIS A DISC?(IF YES,
955 000660 316040 000000           CAMN TAC,SYSTAP        ;BYPASS CHECK FOR SYSTEM TAPE AND ASSIGN BELOW
956 000661 254000 000723'           JRST ASSFIN         ;IS NEXT DEVICE IN LIST THE SYSTEM TAPE?
957 000662 260140 000000           ASSG5: PUSHJ PDP,ASSAG  ;YES, DO NOT ASSIGN, CLEAR DIRECTORY IN CORE BIT
958 000663 254000 000707'           JRST ASSER1         ;NO, TRY TO ASSIGN DEVICE
959 000664 254000 000723'           JRST ASSFIN         ;ALREADY ASSIGNED TO ANOTHER JOB
959 000664 254000 000723'           ;ASSIGNED
  
```

```

960                                     ;SYSTEM TAPE OR NOT MATCH OF ARG AND PHYSICAL NAME.
961
962 000665 316040 000660' ASSG1:  CAMN TAC,SYSTAP           ;IS THIS SYSTEM TAPE?
963 000666 254000 000675'          JRST ASSG2              ;YES
964 000667 135500 000215'          LDB T, PJOBN          ;SAVE OLD JOB NUMBER FOR THIS DEVICE
965 000670 620040 777777'          TRZ TAC,-1             ;COMAPRE LEFT HALF ONLY
966 000671 316040 000013'          CAMN TAC, T1
967 000672 260140 000662'          PUSHJ PDP, ASSASG        ;MATCH, TRY TO ASSIGN IT
968 000673 254000 000675'          JRST ASSG2              ;KEEP LOOKING
969 000674 322500 000723'          JUMPE T,ASSFIN          ;IF OLD JOB NUMBER 0, DEVICE PREVIOUSLY UNASSIGNED
970
971 000675 554306 000003' ASSG2:  HLRZ DEVDAT, DEVSER(DEV DAT)
972 000676 325300 000653'          JUMPN DEVDAT, ASSG0
973                                     IFN FTTTYSER,<
974 000677 262140 000006'          POP      PDP,DEV DAT      ;RESTORE TTY DDB
975                                     >
976                                     IFE FTTTYSER,<
977                                     POP PDP, TAC
978                                     >
979 000700 254000 000770'          JRST NOTDEV              ;PRINT NO SUCH DEVICE
980
981                                     IFN FTLOGIN,<EXTERN STREQ,SYSTAP
982 000701 260140 001723' ASSG7:  PUSHJ  PDP,CTEXT1
983 000702 322100 002034'          JUMPE  TAC1,NOTENF
984 000703 331000 000000'          SKIPL  STREQ
985 000704 254000 001677'          JRST   QLYCM1
986 000705 202100 000665'          MOVEM  TAC1,SYSTAP
987 000706 263140 000000'          POPJ   PDP,
988                                     >

```

```

989
990 000707      ;ALREADY ASSIGNED TO ANOTHER JOB
991              ASSR1:
992              IFE FTTYSER,<
993                POP PDP, TAC
994                MOVEI TAC, ASSMS2
995                PUSHJ PDP,ERRMES
996                LDB TAC, PJOBN
997                JRST DECLF              ;PRINT JOB NUMBER CRLF
998              >
999 000707 135040 000667' IFN FTTYSER,<
1000 000710 262140 000006'   LDB TAC,PJOBN      ;GET JOB NUMBER FOR DEVICE
1001 000711 261140 000001'   POP PDP,DEV DAT    ;GET DDB FOR TTY
1002 000712 201040 000716'   PUSH PDP,TAC ;SAVE JOB NO.
1003 000713 260140 002015'   MOVEI TAC,ASSMS2   ;TYPE ERROR MESSAGE
1004 000714 262140 000001'   PUSHJ PDP,ERRMES
1005 000715 254000 000406'   POP PDP,TAC ;GET JOB NO. BACK
1006              JRST DECLF              ;AND TYPE IT
1007              >

1008 000716 406312 242602 ASSMS2: ASCIZ /ALREADY ASSIGNED TO JOB /
      000717 422624 040646
      000720 516230 747212
      000721 421012 447500
      000722 452370 220000
  
```

```

1009
1010 000723 402006 000005 ;DEVICE ASSIGNED, GIVE IT A LOGICAL NAME
1011                                ASSFIN: SETZM  DEVLOG(DEVDAT)  ;CLEAR LOGICAL NAME
1012                                IFE FTTYSER,<
1013                                POP PDP,TAC                                ;RESTORE INPUT BYTE POINTER
1014                                PUSH PDP,DEVDAT                            ;SAVE ODB ADDRESS
1015                                >
1016 000724 250303 000000 IFN FTTYSER,<
1017                                EXCH  DEVDAT,0(PDP)  ;GET TTYDOB, SAVE DEVICE DOB
1018                                >
1019 000725 260140 001723' PUSHJ PDP,CTEXT1  ;GET SECOND ARG, LOGICAL DEVICE NAME
1020 000726 332040 000002 SKIPE TAC,TAC1   ;IS THERE A LOGICAL NAME SPECIFIED?
1021 000727 260140 000000 PUSHJ PDP, DEVLG   ;YES, SEE IF IT IS ALREADY IN USE
1022                                ; BY THIS USER
1023 000730 254000 000735' JRST ASSF1
1024 000731 201040 000751' MOVEI TAC,LOGERR  ;NO
1025 000732 200303 777775 MOVE  DEVDAT,-3(PDP) ;YES, PRINT ERROR
1026 000733 260140 002015' PUSHJ PDP,ERRMES  ;RESTORE TTY DOB
1027 000734 201040 000000 MOVEI TAC,0
1028 000735 262140 000006 ASSF1: POP PDP,DEVDAT ;CLEAR LOGICAL NAME FOR THIS DEVICE
1029 000736 202046 000005 MOVEM TAC,DEVLOG(DEVDAT);STORE IN DEVICE DATA BLOCK
1030 000737 205100 400000 MOVI TAC1,DVOIRIN  ;CLEAR DIRECTORY IN CORE BIT
1031 000740 413106 000004 ANDCAB TAC1,DEVMOD(DEVDAT) ;SETUP TAC1 WITH DEVICE CHARACTERISTICS FOR ASG
1032                                HGH
1033                                IFN FT2REL,<
1034                                EXTERN ASGHGH
1035 000741 260140 000000 MENTS  ;GO CHECK IF THIS DEVICE HAS INITIALIZED ANY SHARED SEG
1036                                ; IF YES, CLEAR SEG NAMES SO NO NEW SHARING (OTA,MTA ON
1037                                LY)
1038                                >
1039 000742 200106 000000 MOVE TAC1,DEVNAM(DEVDAT) ;PHYSICAL NAME
1040 000743 200303 777776 MOVE  DEVDAT,-2(PDP) ;RESTORE TTY DOB
1041 000744 260140 000000 PUSHJ PDP,PRNAME  ;PRINT IT
1042 000745 265040 002017' JSP TAC,CONMES
1043
1044 000746 202032 351622 ASCII? / ASSIGNED
1045 000747 436350 542032 /
1046 000750 050000 000000 /
1047 000751 462370 744606 LOGERR: ASCII? /LOGICAL NAME ALREADY IN USE, /
000752 406304 047202
000753 466124 040630
000754 512130 142262
000755 202231 620252
000756 516125 420000
  
```

```

1048
1049
1050           ;"DEASSIGN DEV" - DEASSIGNS DEVICE FROM CONSOLE
1051
1052           INTERNAL NOTDEV
1053 000757 322200 000371' DEASSI: JUMPE ITEM,CPOPJ           ;NO-OP IF NO JOB NUMBER
1054 000760 260140 001723'           PUSHJ PDP,CTEXT1       ;GET DEVICE NAME
1055 000761 322100 002060'           JUMPE TAC1,DEASTY      ;NO ARG, IF 0, DEASSIGN ALL BUT TTY
1056 000762 200040 000002'           MOVE TAC, TAC1         ;DEVICE NAME
1057 000763 260140 000000'           PUSHJ PDP, DEVSRC     ;SEARCH FOR DEVICE
1058 000764 254000 000770'           JRST DEAR1           ;NOT FOUND
1059 000765 260140 002043'           PUSHJ PDP, DEASG     ;FOUND, DEASSIGN IT
1060 000766 254000 000776'           JRST DEAR2           ;NOT PREVIOUSLY ASSIGNED
1061 000767 263140 000000'           POPJ PDP,           ;DEVICE DEASSIGNED
1062
1063 000770           NOTDEV:
1064 000770 200303 777776' DEAR1: MOVE   DEVDAT,-2(PDP) ;RESTORE TTY DDB
1065 000771 265040 002015'           JSP TAC,ERRMES     ;PRINT NO SUCH DEVICE
1066
1067 000772 472364 051652           ASCII /NO SUCH DEVICE
1068 000773 416204 042212
1069 000774 532230 342432
1070 000775 050000 000000 /
1071 000776 200106 000000' DEAR2: MOVE TAC1,DEVNAM(DEVDAT) ;PRINT PHYSICAL DEVICE NAME
1072 000777 200303 777776'           MOVE   DEVDAT,-2(PDP) ;RESTORE TTY DDB
1073 001000 260140 000744'           PUSHJ PDP,PRNAME
1074 001001 265040 002015'           JSP TAC,ERRMES
1075 001002 202570 151634           ASCII / WASN'T ASSIGNED
1076 001003 236504 040646
1077 001004 516230 747212
1078 001005 420321 200000 /
  
```

```

1077          INTERNAL FTREASSIGN
1078          IFE FTREASSIGN,<
1079          REASSI=UOERR
1080          >
1081
1082          IFN FTREASSIGN,<
1083
1084          ;REASSIGN UO
1085          ;CALL MOVE AC,JOB NUMBER
1086          ;      MOVE AC+1,SIXBIT /DEVICE/
1087          ;      CALL AC,[SIXBIT /REASSIGN/]
1088          ;
1089          ;IF C(AC)=0, JOB HAS NOT BEEN INITIALIZED
1090          ;IF C(AC+1)=0, DEVICE NOT ASSIGNED TO THIS JOB OR DEVICE IS A TTY
1091
1092          INTERNAL REASSIGN
1093          EXTERNAL PUUOAC,JOB
1094          001006 135600 000000 REASSI: LDB UO,PUUOAC
1095          001007 505600 000007          HRLI UO,PROG
1096          001010 261160 000014          PUSH PDP,@UO          ;STACK JOB NUMBER TO BE REASSIGNED TO
1097          001011 350000 000014          AOS UO
1098          001012 200120 000014          MOVE TAC1,@UO          ;DEVICE NAME
1099          001013 200200 000000          MOVE ITEM,JOB          ;THIS JOB NUMBER
1100          001014 364600 001026          SOJA UO,REASS1
1101
1102          ;"REASSIGN DEV:JOB" - REASSIGN DEVICE "DEV" TO JOB "JOB"
1103
1104
1105          001015          REASS:
1106          001015 260140 001723'          PUSHJ PDP,CTEXT1          ;GET DEVICE NAME
1107          001016 322100 002034'          JUMPE TAC1,NOTENF          ;NULL NAME?
1108          001017 200600 000002          MOVE UO,TAC1          ;SAVE IT IN UO
1109          001020 260140 002071'          PUSHJ PDP,DECIN1          ;GET NEW JOB NUMBER
1110          001021 254000 002034'          JRST NOTENF          ;NONE SPECIFIED, DOESN'T RETURN IF ERROR
1111          001022 254000 002000'          JRST COMERA          ;ILLEGAL CHARACTER
1112          001023 261140 000002          PUSH PDP,TAC1          ;PUT JOB NUMBER ON STACK, DEVICE
1113          001024 200100 000014          MOVE TAC1,UO          ;NAME IN TAC1
1114          001025 402000 000014          SETZM UO          ;SET COMMAND SWITCH

```

```

1115
1116
1117
1118
1119
1120 001026 250203 000000 REASS1: EXCH ITEM,(PDP)
1121 001027 303200 000000 CALE ITEM,JOBN
1122 001030 254000 001115' JRST REASE2
1123 001031 200044 000614' MOVE TAC,JBSTSTS(ITEM)
1124 001032 250203 000000 EXCH ITEM,(PDP)
1125 001033 607040 040000 TLNN TAC,JNA
1126 001034 254000 001111' JRST REASE1
1127 001035 200040 000002 MOVE TAC,TAC1
1128 001036 260140 000763' PUSHJ PDP,DEVSRG
1129 001037 254000 001115' JRST REASE2
1130 001040 135040 000707' LDB TAC,PJOBN
1131 001041 312040 000004 CAME TAC,ITEM
1132 001042 254000 001121' JRST REASE3
1133 001043 200046 000004 MOVE TAC,DEVMOD(DEV DAT)

1134 001044 603040 000010 TLNE TAC,DVTTY
1135 001045 254000 001123' JRST REASE6
1136 001046 606040 200000 TRNN TAC,ASSPRG
1137 001047 254000 001102' JRST REASS3
1138 001050 326670 001066' JUMPN UUD,REASS4
1139 001051 504303 000000 HRL DEV DAT,(PDP)
1140 001052 202307 000000 MOVEM DEV DAT,JOBF DV(J DAT)
1141 001053 262140 000002 POP PDP,TAC1
1142 001054 200303 777776 MOVE DEV DAT,-2(PDP)
1143 001055 205100 000020 MOVSI TAC1,TTYRNC
1144 001056 436103 777777 IORM TAC1,-1(PDP)
1145 001057 265100 000000 JSP TAC1,MSTART
1146 001060 265040 000426' JSP TAC,MONSTR
1147 001061 554047 001052' HLRZ TAC,JOBF DV(J DAT)
1148 001062 261140 000001 PUSH PDP,TAC
1149 001063 550307 001061' HRRZ DEV DAT,JOBF DV(J DAT)
1150 001064 200200 001013' MOVE ITEM,JOB
1151 001065 476000 000014 SETOM UUD
1152 001066 550446 000003 REASS4: HRRZ DSER,DEV SER(DEV DAT)
1153 001067 552307 001063' HRRZM DEV DAT,JOBF DV(J DAT)
1154 001070 200500 000000 MOVE UCHN,USRHCU

;ROUTINE COMMON TO REASSIGN UUD AND COMMAND
EXTERNAL JBSTSTS,PJOBN,JOBF DV
;NEW JOB NO, IN ITEM
;IS JOB NUMBER OUT OF RANGE?
;YES, DO NOT REASSIGN
;NEW JOB STATUS
;RESTORE ITEM AND STACK
;DOES NEW JOB EXIST?
;NO.
;SEARCH FOR DEV
;NOT FOUND
;ASSIGNED TO THIS JOB
;NO
;IS IT A TTY?
;YES, CAN'T BE REASSIGNED.
;IS DEVICE INITED?
;NO,
;YES, COMMAND LEVEL?
;YES, SCHEDULE RELEASE
;SET TTYRNC SO JOB WILL RUN
;SET FLAG TO STOP JOB

```

```

1155 001071 200312 000000 REASS2: MOVE DEVDAT,USRJDA(UCHN)            IGET XWD UO BITS,0DB ADDRESS
1156 001072 550047 001067' HRRZ TAC,JOBFDV(JDAT)        IGET ADDR, OF DDB SAVED BY COMMAND
1157 001073 261140 000012        PUSH PDP,UCHN                ISAVE USER CHANNEL
1158 001074 306046 000000        CAIN TAC,(DEVDAT)            IIS CHOSEN DEVICE ON THHS CHANNEL?
1159 001075 260140 000000        PUSHJ PDP,RELEASE5           IYES, RELEASE DEVICE
1160 001076 262140 000012        POP PDP,UCHN
1161 001077 365500 001071'        SOJGE UCHN,REASS2
1162 001100 200307 001072'        MOVE DEVDAT,JOBFDV(JDAT)
1163 001101 200200 001064'        MOVE ITEM,JOB                ICURRENT JOB NUMBER
1164 001102 201040 400000        REASS3: MOVEI TAC,ASSCON      IASSIGN IT BY CONSOLE
1165 001103 436046 000004        IORM TAC,DEVMON(DEVDAT)
1166 001104 250203 000000        EXCH ITEM,(PDP)
1167 001105 137200 001040'        DPB ITEM,PJOBN               IPUT IN NEW JOB NUMBER
1168 001106 262140 000004        POP PDP,ITEM
1169 001107 321600 000425'        JUMPL UO,ESTOP
1170 001110 263140 000000        POPJ PDP,
1171
1172 001111 262140 000001 REASE1: POP PDP,TAC
1173 001112 322600 001231'        JUMPE UO,ATT4                IJOB NEVER WAS INITIATED
1174 001113 402020 000014 REASE4: SETZM @UO              ICLEAR C(AC)
1175 001114 263140 000000        POPJ PDP,
1176
1177 001115 201040 000770' REASE2: MOVEI TAC,NOTDEV        INO SUCH DEVICE
1178 001116 262140 000002 REASE5: POP PDP,TAC1
1179 001117 322601 000000        JUMPE UO,(TAC)
1180 001120 344600 001113'        AOJA UO,REASE4
1181
1182 001121 201040 000776' REASE3: MOVEI TAC,DEAER2        IWASN'T ASSIGNED
1183 001122 254000 001116'        JRST REASE5
1184
1185 001123 201040 001125' REASE6: MOVEI TAC,REASE7
1186 001124 254000 001116'        JRST REASE5
1187 001125 200176 000000 REASE7: MOVE TAC1,DEVNAM(DEVDAT)
1188 001126 200303 777776        MOVE    DEVDAT,-2(PDP)        IRESTORE TTY DDB
1189 001127 260140 001020'        PUSHJ PDP,PRNAME
1190 001130 265040 002015'        JSP TAC,ERRMES
1191 001131 202070 147116        ASCII? / CAN'T BE REASSIGNED
      001132 521010 242500
      001133 512130 151646
      001134 446171 642610
1192 001135 064240 000000 /
1193                                    >

```

```

1194 INTERNAL FTATTACH
1195 IFN FTATTACH,<
1196
1197 ;"ATTACH DEVNAM" -ATTACHES A PREVIOUSLY PARTITIONED DEVICE
1198 ; NOTE-MUST BE LOGGED IN UNDER [1,1] TO DO THIS
1199 ; "ATTACH N [PROJ,,PROG.]" - ATTACH CONSOLE TO JOB N
1200 ;CHANGES ADDRESS OF TTY DEVICE DATA BLOCK STORED IN -2(PDP)
1201 ;BY THE COMMAND DECODER
1202
1203 INTERNAL ATTACH
1204 EXTERNAL TTYATT,JOBN,TTYFND
1205
1206 ATTACH: IFE FTTTYSER,<
1207 MOVE AC2,TAC ;SAVE BYTE POINTER>
1208 001136 260140 002071' PUSHJ PDP,DECIN ;GET JOB NO,
1209 001137 254000 002034' JRST NOTENF ;NOT A NUMBER OR NONE SPECIFIED
1210 IFN FTLOGIN, <
1211 001140 254000 001164' JRST DEVATT ;WANTS TO ATTACH A DEVICE
1212 >
1213
1214 IFE FTLOGIN, <
1215 JRST ATT1
1216 >
1217 001141 332000 000002 SKIPE TAC1 ;0 IS ILLEGAL
1218 001142 301100 001027' CAIL TAC1, JOBN ;IS JOB NUMBER TOO BIG?
1219 001143 254000 001206' JRST ATT1 ;ILLEGAL JOB NUMBER
1220 001144 205540 040000 MOVSI T1,JNA ;HAS THIS JOB NO BEEN ASSIGNED?
1221 001145 616542 001031' TDNN T1,JBTSTS(TAC1)
1222 001146 254000 001231' JRST ATT4 ;NO, PRINT ERROR
1223 INTERNAL FTLOGIN
1224 IFN FTLOGIN,<
1225 001147 200000 000002 MOVE IOS,TAC1 ;SAVE JOB NO,
1226 001150 260140 002117' PUSHJ PDP,PJPGNO ;GET PROJ,-PROG, NOS, ARG(IF ERROR, POP SUB LEVEL
1227 ; OFF 1, PRINT ERROR, AND DO NOT RETURN HERE)
1228 001151 202000 000002 MOVEM IOS,TAC1 ;RESTORE
1229 001152 336000 000016 SKIPN AC2 ;DID USER TYPE IN A PROJ,PROG # IN [ ]'S?
1230 001153 200704 000637' MOVE AC2,PRJPRG(ITEM) ;NO, ASSUME PROJ,PROG NUMBER FOR CURRENT JOB
1231 ; SO USER CAN AVOID TYPEING PROJ,PROG NO, IF GOING
1232 ; FROM 1 JOB TO ANOTHER UNDER SAME PROJ,PROG
1233 ; IS THIS THE SAME PERSON WHO DETACHED FROM THIS JOB NO?
1234 001154 312702 001153' BERR?
1235 001155 254000 001224' JRST ATT3 ;NO-ERROR
1236 ; YES
1237 >
1238 001156 200200 000002 MCVE ITEM,TAC1 ;JOB NUMBER TO ITEM
1239 001157 260140 000565' PUSHJ PDP,TTYATT ;NO, ATTACH TTY
1240 001160 254000 001214' JRST ATT2 ;ERROR CAN'T ATTACH
1241 001161 202303 777776 MOVEM DEVDAT,-2(PDP) ;CHANGE DEV DATA BLOCK ADDRESS
1242 IFN FTTTYSER,<
1243 EXTERN TSETBF
1244 001162 260140 000000 PUSHJ PDP,TSETBF ;INITIALIZE TTY INPUT BUFFER
1245 >
1246 001163 254000 000000 JRST TTYFND ;ATTACHED, GO SET UP OUTP, BYTE PTR,

```

1247					
1248				INTERNAL FTLOGIN	
1249				IFN FTLOGIN,<	
1250				DEVATT: IFE FTTTYSER,<	
1251				MOVE TAC,AC2	IRESTORE BYTE POINTER>
1252	001164	260140	001723'	PUSHJ PDP,CTEXT1	I GET DEVICE ARGUMENT
1253	001165	255000	000000	JFCL	ISHOULD NEVER RETURN
1254	001166	200504	001154'	MOVE T,PRJPRG(ITEM)	I GET PROJ,-PROG, NOS.
1255	001167	312500	000611'	CAME T,SYSPP	I PRJPRG = [1,1]?
1256	001170	254000	001235'	JRST ATT5	INO - ERROR
1257	001171	200040	000002	MOVE TAC,TAC1	I YES-SET UP DEVICE NAME
1258	001172	261140	000006	PUSH PDP,DEV DAT	ISAVE DDB FOR THIS TTY
1259	001173	260140	001036'	PUSHJ PDP,DEV SRC	ISEARCH FOR DEVICE
1260	001174	254000	001302'	JRST TEMP1	INOT FOUND
1261	001175	135500	001105'	LDB T,PJOBN	I GET JOB NUMBER
1262	001176	326500	001242'	JUMPN T,ATT6	IS IT = 0?
1263	001177	137200	001175'	DPB ITEM,PJOBN	ISSET JOB NUMBER
1264	001200	200046	000004	MOVE TAC,DEV MOD(DEV DAT);	CHECK TO SEE IF THIS IS A TTY
1265	001201	135100	000153'	LDB TAC1,PUNIT	I
1266	001202	603040	000010	TLNE TAC,DVTTY	IS IT A TTY?
1267	001203	542302	000154'	HRRM DEV DAT,TTYTAB(TAC1)	ISSET TRANS TABLE TO POINT TO DDB
1268	001204	262140	000006	POP PDP,DEV DAT	I
1269	001205	263140	000000	POPJ PDP,	IRETURN
1270					

```

1271
1272 001206 265040 002015' ATT1: JSP TAC,ERRMES
1273 001207 446311 442616      ASCIZ /ILLEGAL JOB NUMBER
      001210 406304 045236
      001211 411011 652632
1274 001212 412132 206424 /
      001213 000000 000000
1275
1276 001214 200106 000000 ATT2: MOVE TAC1,DEVNAM(DEVDAT)      ;PRINT PHYSICAL NAME
1277 001215 200303 777776      MOVE DEVDAT,=2(PDP) ;RESTORE TTY DDB
1278 001216 260140 001127'      PUSHJ PDP,PRNAME
1279 001217 265040 002015'      JSP TAC,ERRMES
1280 001220 202031 451212      ASCIZ / ALREADY ATTACHED
      001221 406113 120202
      001222 522510 141620
1281 001223 426101 505000 /
1282
1283 001224 265040 002015' ATT3: JSP TAC,ERRMES
1284 001225 416031 623650      ASCIZ /CAN'T ATT TO JOB
      001226 202032 452100
      001227 522364 045236
1285 001230 410321 200000 /
1286
1287 001231 265040 002015' ATT4: JSP TAC,ERRMES
1288 001232 472372 420202      ASCIZ /NOT A JOB
      001233 202251 741032
1289 001234 050000 000000 /
1290
1291
1292 001235 265040 002015' ATT5: JSP TAC,ERRMES
1293 001236 416031 623650      ASCIZ /CAN'T ATT DEV
      001237 202032 452100
1294 001240 422132 606424 /
      001241 000000 000000
1295
1296 001242 262140 000006 ATT6: POP PDP,DEVDAT
1297 001243 265040 002015'      JSP TAC,ERRMES
1298 001244 536032 347116      ASCIZ /WASN'T DET
      001245 521010 442650
1299 001246 064240 000000 /
1300 >
  
```

```

1301                   INTERNAL FTATTACH
1302                   IFN FTATTACH,<
1303
1304                   ;"DETACH" - DETACH CONSOLE FROM JOB
1305                   ;"DETACH DEVNAM" - DETACHES DEVICE FROM THE SYSTEM SOFTWAREWISE
1306                   ;           NOTE - MUST BE LOGGED IN UNDER [1,1] TO DO THIS
1307
1308                   EXTERNAL TTYDET
1309                   INTERNAL FTLOGIN
1310
1311                   IFE FTLOGIN,<
1312                   DETACH: JRST TTYDET                   ;GO DETACH TTY
1313                   >
1314
1315                   IFN FTLOGIN,<
1316                   EXTERNAL PRJPRG,SYSPP,PUNIT,TTYTAB,PJOBN
1317                   DETACH: PUSHJ PDP,CTEXT1                   ;GET ARGUMENT
1318                   JUMPE TAC1,TTYDET                   ;ONLY "DET" TYPED
1319                   MOVE T,PRJPRG(ITEM)                   ;GET PROJ,-PROG. NUMBER
1320                   CAME T,SYSPP                   ;PRJPRG = [1,1]?
1321                   JRST LOGER1                   ;NO-PRINT ERROR MSG.
1322                   MOVE TAC,TAC1                   ;YES-SET UP DEVICE NAME
1323                   PUSH PDP,DEV DAT                   ;SAVE TTY DDB
1324                   PUSHJ PDP,DEV SRC                   ;SEARCH FOR DEVICE
1325                   JRST TEMP1                   ;DEVICE NOT FOUND
1326                   MOVE TAC,DEV MOD(DEV DAT)                   ;CHECK TO SEE IF THIS IS DSK
1327                   TLNE TAC,DV DSK                   ;IS IT THE DSK?
1328                   JRST TEMP1                   ;YES-PRINT ERROR MSG.
1329                   MOVEI TAC1,ASS CON                   ;FOUND-SET UP ASSIGNED BY CONSOLE
1330                   PUSHJ PDP,ASS SASG                   ;TRY TO ASSIGN
1331                   JRST TEMP2                   ;CAN'T ASSIGN
1332                   TLNE DEV DAT,SYS DEV                   ;IS THIS SYSTEM DEVICE?
1333                   JRST TEMP1                   ;YES-PRINT ERROR MSG.
1334                   XOP T,T                   ;NO-SET TO ZERO
1335                   DPR T,PJOBN                   ;SET JOB NO. TO NULL JOB
1336                   MOVE TAC,DEV MOD(DEV DAT)                   ;CHECK TO SEE IF THIS IS A TTY
1337                   TLNN TAC,DV TTY                   ;IS IT A TTY?
1338                   JRST ,+4                   ;NO-GO AHEAD
1339                   LDB TAC,PUNIT                   ;YES-SET UP FOR SENSER
1340                   MOVEI TAC1,-1                   ;SET LEFT HALVE OF TTYTAB TO -1
1341                   HRRM TAC1,TTYTAB(TAC)                   ;SO THAT SENSER CHECKS FOR THIS
1342                   POP PDP,DEV DAT                   ;RESTORE TTY DDB
1343                   POPJ PDP,                   ;SUCCESSFUL RETURN
  
```

```
1344 001302 262140 000006 TEMP1: POP PDP,DEV DAT            ;RESTORE TTY DDB
1345 001303 254000 000770'            JRST DEAE1            ;PRINT ERROR MSG. AND RETURN
1346 001304            TEMP2:
1347            IFE FTTYSER,< POP PDP,DEV DAT            ;RESTORE TTY DDB
1348            JRST ASSER1+1            ;PRINT ERROR MSG. AND RETURN
1349            >
1350            IFN FTTYSER,<
1351 001304 254000 000707'            JRST    ASSER1
1352            >
1353 001305 265040 002015' LOGER1: JSP TAC,ERRMES
1354 001306 416031 023650            ASCIZ /CAN'T DET DEV
      001307 202110 552100
1355 001310 422132 006424 /
      001311 000000 000000
1356            >>            ;CLOSE BOTH FTLOGIN AND FTATTACH CONDITIONALS.
```

```

1357                                     ;"DAYTIME" - PRINT TIME OF DAY
1358
1359                                     EXTERNAL TIME,THSDAT,MONTAB,JIFMIN
1360
1361 001312                               DAYTIME
1362                                     IFN      FTLOGIN, <
1363 001312 322200 001330'                JUMPE ITEM,DAYTM1
1364 001313 260140 002071'                PUSHJ PDP,DECIN      ;WAS AN ARGUMENT TYPED?
1365 001314 254000 001330'                JRST DAYTM1         ;NO, PRINT TIME
1366 001315 254000 002000'                JRST COMERA        ;ERROR
1367 001316 554044 001251'                HLRZ TAC,PRJPRG(ITEM) ;YES
1368 001317 302040 000001'                CAIE TAC,1         ;IS THIS PROJECT 1?
1369 001320 254000 001330'                JRST DAYTM1         ;NO, PRINT TIME OF DAY ANYWAY(IGNORE ARG)
1370 001321 200040 000002'                MOVE TAC,TAC1      ;YES, RESET TIME ACCORDING TO ARGUMENT
1371 001322 231040 000144'                IDIVI TAC,*D100
1372 001323 221040 000074'                IMULI TAC,*D60
1373 001324 270040 000002'                ADD TAC,TAC1       ;COMPUTE MINUTES
1374 001325 221040 000000'                IMULI TAC,JIFMIN   ;COMPUTE JIFFIES
1375 001326 202040 000000'                MOVEM TAC,TIME     ;AND STORE
1376 001327 263140 000000'                POPJ PDP,
1377                                     >
1378 001330 200040 000000'                DAYTM1: MOVE TAC,THSDAT      ;PRINT TODAY'S DATE
1379 001331 231040 000037'                IDIVI TAC,*D31
1380 001332 250040 000002'                EXCH TAC,TAC1      ;YEAR AND MONTH IN TAC1
1381 001333 260140 000000'                PUSHJ PDP,DECP1    ;ADD 1 TO DAY(IN TAC) AND PRINT DECIMAL
1382 001334 201040 000000'                MOVEI TAC,0
1383 001335 235040 000014'                DIVI TAC,*D12
1384 001336 250040 000002'                EXCH TAC,TAC1
1385 001337 200041 000000'                MOVE TAC,MONTAB(TAC) ;MONTH
1386 001340 137040 003062'                DPB TAC,(POINT 21,DAMESS,27]
1387 001341 201040 000000'                MOVEI TAC,DAMESS
1388 001342 260140 002017'                PUSHJ PDP,CONMES   ;PRINT DAY
1389 001343 201042 000100'                MOVEI TAC,*D64(TAC1)
1390 001344 260140 000530'                PUSHJ PDP,RADX10   ;PRINT YEAR
1391 001345 260140 002032'                PUSHJ PDP,PRSPC
1392 001346 200040 001326'                MOVE TAC,TIME     ;PRINT TIME OF DAY
1393 001347 254000 002174'                JRST PRTIM1
  
```

```

1394                                     INTERNAL FTTIME
1395                                     IFN FTTIME,<
1396                                     ;"TIME (JOB NO.)" - PRINT TOTAL AND INCREMENTAL RUNNING TIME FOR A JOB
1397                                     ;FOLLOWED BY KILO-CORE TICKS
1398                                     ;"TIME 0" IMPLIES RUNTIME FOR NULL JOB AND THE TOTAL TIME SPENT SHUFFLING USERS
1399                                     ;          IF NO JOB NO. GIVEN-TIME WILL BE FOR CURRENTLY LOGGEN IN JOB NO.
1400
1401                                     EXTERNAL RTIME,TTIME,SHFWRD,WDPJIF,JOBN,JBTSTS,CLRWRD,LSTWRD
1402
1403 001350 260140 002071' RUNTIM: PUSHJ PDP,DECIN          ;GETJOB NO, ARG,
1404 001351 254000 001413'          JRST RUN1          ;NO ARG, GIVEN - LOGGED IN?
1405 001352 254000 002000'          JRST COMERA        ;ILLEGAL DECIMAL CHARACTER RETURN
1406 001353 322100 001360'          JUMPE TAC1,RUN2        ;RUNTIME 0 GIVEN
1407 001354 301100 001142'          CAIL TAC1,JOBN        ;JOB NO, TOO BIG?
1408 001355 254000 001206'          JRST ATT1          ;YES,ILLEGAL JOB. NO.
1409 001356 200042 000064'          MOVE TAC,TTIME(TAC1) ;TIME SINCE LAST LOGGIN
1410 001357 254000 002146'          JRST PRTIME        ;PRINT AND RETURN
1411
1412 001360 260140 002025' RUN2:  PUSHJ PDP,INLMES          ;PRINT
1413 001361 516210 646100          ASCIZ /SHFL /          ;"SHUFFLING "
1414 001362 000000 000000
1415 001363 200040 000000          MOVE TAC,SHFWRD        ;NO, WORDS SHUFFLED
1416 001364 231040 000000          IDIVI TAC,WDPJIF       ;NO,WORDS/JIFFY USING BLT
1417 001365 260140 002146'          PUSHJ PDP,PRTIME        ;PRINT SHUFFLE TIME
1418 001366 260140 002025'          PUSHJ PDP,INLMES
1419 001367 552071 751100          ASCIZ /ZCOR /
1420 001370 000000 000000
1421 001371 200040 000000          MOVE TAC,CLRWRD
1422 001372 231040 001364'          IDIVI TAC,WDPJIF
1423 001373 260140 002146'          PUSHJ PDP,PRTIME        ;PRINT TIME SPENT CLEARING CORE
1424 001374 260140 002025'          PUSHJ PDP,INLMES
1425 001375 462372 352100          ASCIZ /LOST /
1426 001376 000000 000000
1427 001377 200040 000000          MOVE TAC,LSTWRD
1428 001400 260140 002146'          PUSHJ PDP,PRTIME        ;PRINT TIME SPENT "LOST" RUNNING NULL JOB
1429 001401 260140 002025'          PUSHJ PDP,INLMES
1430 001402 472531 446100          ASCIZ /NULL /        ;PRINT
1431 001403 000000 000000          ;"NULL TIME "
1432 001404 200040 001356'          MOVE TAC,TTIME        ;PRINT NULL JOB RUNNING TIME
1433 001405 260140 002146'          PUSHJ PDP,PRTIME
1434 001406 260140 002025'          PUSHJ PDP,INLMES
1435 001407 526404 020100          ASCIZ /UP /          ;PRINT TOTAL SYSTEM UP TIME
1436 001410 000000 000000
1437 001411 200040 000000          MOVE TAC,UPTIME
1438 001412 254000 002146'          JRST PRTIME          ;AND RETURN
1439

```



```

1461                                     ;ROUTINE TO LET TTY TALK TO ANY OTHER RING OF TTYS
1462                                     ; "TALK TTYN"
1463                                     ; TTYN NEED NEVER HAVE BEEN TYPED ON BEFORE
1464
1465                                     INTERNAL FTTALK
1466
1467                                     IFN FTTALK,<
1468                                     EXTERNAL TTYTLK
1469
1470 001432 260140 001723' TALK:  PUSHJ PDP,CTEXT1           ;GET ARGUMENT
1471 001433 322100 001442'      JUMPE TAC1,TALK2           ;NONE SPECIFIED
1472 001434 200040 000002'      MOVE TAC,TAC1
1473 001435 260140 001256'      PUSHJ PDP,DEVSRG           ;SEARCH FOR DEVICE
1474 001436 254000 001445'      JRST TALK1                 ;DEVSRG SHOULD FIND A FREE TTY DDB
1475                                     ; EVEN IF TTY NEVER TYPED ON YET
1476 001437 260140 000000'      PUSHJ PDP,TTYTLK
1477 001440 254000 001445'      JRST TALK1
1478 001441 263140 000000'      POPJ PDP,
1479
1480 001442 205200 000400' TALK2:  MOVSI ITEM,NOINCK           ;SET FLAGS SO RESPONSE WILL PRINT
1481 001443 202203 777777'      MOVEM ITEM,-1(PDP)
1482 001444 254000 002034'      JRST NOTENF
1483
1484 001445 200303 777776' TALK1:  MOVE  DEVDAT,-2(PDP)       ;RESTORE TTY DDB
1485 001446 205200 000400'      MOVSI ITEM,NOINCK           ;SET FLAGS TO PRINT RESPONSE
1486 001447 202203 777777'      MOVEM ITEM,-1(PDP)
1487 001450 265040 002015'      JSP TAC,ERRMES
1488 001451 412532 354432'      ASCIZ /BUSY
1489 001452 050000 000000'      /
1490                                     >

```

```

1491                                     INTERNAL FTEXAMINE
1492
1493                                     IFN FTEXAMINE,<
1494
1495                                     ;"EXAMINE LOC" - LOOKS A CONTENTS OF LOC AND PRINTS IN OCTAL
1496                                     ;IF LOC IS MISSING, NEXT LOC IS PRINTED
1497                                     ;IF PREVIOUS WAS E COMMAND, SAME LOC IF PREVIOUS WAS D COMMAND
1498                                     ;TAB. IS PRINTED INSTEAD OF CRLF(LIKE DDT)
1499
1500                                     EXTERNAL TPOPJ,JOBJC
1501
1502 001453 335007 000000 ECOM: SKIPGE JOBEXM(JDAT) ;WAS PREVIOUS D OR E COMMAND, A D COMMAND?
1503 001454 350007 001453' AOS JOBEXM(JDAT) ;NO, IT WAS AN E, INCREMENT IN CASE HE TYPES NO ARG
1504 001455 563007 001454' HRRS JOBEXM(JDAT) ;YES, FLAG THAT E HAPPENED LAST(LH=-1)
1505 001456 260140 002073' PUSHJ PDP,OCTIN ;GET OCTAL LOCATION
1506 001457 334107 001455' SKIPA TAC1,JOBEXM(JDAT) ;NONE SPECIFIED, USE LAST LOC OF D OR NEXT OF E
1507 001460 254000 002000' JRST COMERA ;ILLEGAL CHARACTER
1508 001461 260140 001535' PUSHJ PDP,DEAT ;CHECK FOR AC REFERENCE AND STORE JOBEXM
1509 001462 550600 000002 HRRZ UUO,TAC1 ;IGNORE LH
1510 001463 505600 000007 HRLI UUO,PROG ;SET TO RELOCATE
1511 001464 260140 002234' PUSHJ PDP,GETWRD ;GET WORD FROM LOW OR HIGH SEG
1512 001465 254000 001550' JRST ECOMA ;ERROR, OUT OF BOUNDS
1513 001466 261140 000001 PUSH PDP,TAC ;SAVE CONTENTS OF LOC TO BE PRINTED
1514 001467 550047 001457' HRRZ TAC,JOBEXM(JDAT) ;PRINT LOC BEING EXAMINED
1515 001470 260140 000000 PUSHJ PDP,OCTPNT
1516 001471 260140 002025' PUSHJ PDP,INLMES ;PRINT SLASH TAB
1517 001472 274220 000000 ASCIZ */ *
1518 001473 554043 000000 HLRZ TAC,(PDP) ;PRINT LEFT HALF
1519 001474 260140 001470' PUSHJ PDP,OCTPNT
1520 001475 260140 002025' PUSHJ PDP,INLMES ;PRINT SPACE
1521 001476 200000 000000 ASCIZ / /
1522 001477 550043 000000 HRRZ TAC,(PDP) ;PRINT RIGHT HALF
1523 001500 260140 001474' PUSHJ PDP,OCTPNT
1524 001501 260140 002025' PUSHJ PDP,INLMES ;PRINT FINAL TAB
1525 001502 044000 000000 ASCIZ / /
1526 001503 254000 000231' JRST TPOPJ ;IPOP PDP,TAC,POPJ PDP,

```

```

1527
1528 ;"DEPOSIT LH RH LOC" - DEPOSITS XWD LH,RH IN LOCATION LOC
1529 ;IF LOC IS MISSING, ASSUME NEXT LOC IF PREVIOUS D, SAME LOC IF PREVIOUS E
1530
1531 EXTERNAL JOBEXM,JOBPF I
1532
1533 001504 260140 002073' DCOM: PUSHJ PDP, OCTIN IGET LH
1534 001505 254000 002034' JRST NOTENF INOT ENOUGH ARGUMENTS
1535 001506 254000 002000' JRST COMERA IILLEGAL CHARACTER
1536 001507 506100 000000' HRLM TAC1, IOS ISAVE LH
1537 001510 260140 002073' PUSHJ PDP, OCTIN IGET RH
1538 001511 254000 002034' JRST NOTENF INOT ENOUGH ARGUMENTS
1539 001512 254000 002000' JRST COMERA IILLEGAL CHARACTER
1540 001513 542100 000000' HRRM TAC1, IOS ISAVE RH
1541 001514 331007 001467' SKIPL JOBEXM(JDAT) ;HAS PREVIOUS D OR E AN E COMMAND?
1542 ; LH=-1 IF E, LH=0 IF D
1543 001515 350007 001514' AOS JOBEXM(JDAT) ;NO, INCREMENT IN CASE USER TYPED NOT THIRD ARG
1544 ; FOR SUCCESSIVE D'S
1545 001516 553007 001515' HRRZS JOBEXM(JDAT) ;FLAG THAT A D WAS DONE LAST(LH=0)
1546 001517 260140 002073' PUSHJ PDP, OCTIN IGET LOC
1547 001520 334107 001516' SKIPA TAC1, JOBEXM(JDAT) ;NOT SPECIFIED, USE LAST OF E OR NEXT OF D
1548 001521 254000 002000' JRST COMERA IILLEGAL CHARACTER
1549 IFN FT2REL, <
1550 EXTERN CHKMED
1551 001522 260140 000441' PUSHJ PDP, CHKMED ;CHECK TO SEE IF HIGH SEG IS SHARABLE
1552 ; IF YES, SET USER-MODE WRITE PROTECT (UWP) ON
1553 ; FOR THIS USER, AND SET MEDDLE FOR THIS USER
1554 ; SO HE CANNOT TURN UWP OFF
1555
1556 001523 606100 777760 > TRNN TAC1, 777760 ;IN USER ACS
1557 001524 254000 001531' JRST DCOM1 ;YES
1558 001525 554040 000007' HLRZ TAC, PROG IGET PROTECTION
1559 001526 303100 000000' CAILE TAC1, JOBPF I ;NO, GREATER THAN HIGHEST LOC, PROTECTED
1560 ; FROM IO IN JOB DATA AREA?
1561 001527 313100 000001' CAMLE TAC1, TAC ;IN BOUNDS?
1562 001530 254000 001545' JRST DCOMA ;NO
1563 001531 260140 001535' DCOM1: PUSHJ PDP, DEAT ;CHECK FOR AC REFERENCE
1564 001532 505100 000007' HRLI TAC1, PROG ;SET FOR RELOCATION
1565 001533 202020 000002' MOVEM IOS, @TAC1
1566 001534 263140 000000' POPJ PDP,
1567
1568

```

```

1569 001535 621100 777777 DEAT: TLZ TAC1,-1 ;CLEAR LH IN CASE THIS IS A SUCCESSIVE E WITH NO ARG
1570 001536 542107 001520' HRRM TAC1,JOBEXM(JDAT) ;STORE FOR NEXT TIME, DO NOT TOUCH LH(D OR E LAST)
1571 ; YES, WAS JOB STOPPED IN USER MODE?
1572 001537 301100 000020 CAIL TAC1,20 ;IS IT AN AC?
1573 001540 263140 000000 POPJ PDP, ;NO
1574 001541 200047 000000 MOVE TAC,JOBPC(JDAT)
1575 001542 603040 010000 TLNE TAC,USRMOD ;USER MODE?
1576 001543 271100 000020 ADDI TAC1,20 ;YES USER ACS ARE AT 20 INSTEAD OF 0
1577 001544 263140 000000 POPJ PDP,
1578
1579 001545 DCOMA:
1580 IFN FT2REL,<
1581 EXTERN HGHDEP
1582 001545 260140 000000 PUSHJ PDP,HGHDEP ;IS IT IN BOUNDS AND ALLOWED IN HIGH SEG?
1583 001546 254000 001550' JRST ECOMA ;NO, PRINT "OUT OF BOUNDS"
1584 001547 263140 000000 POPJ PDP, ;YES, IOS DEPOSITED, AND JOBEXM UPDATED
1585 >
1586 001550 265040 002015' ECOMA: JSP TAC,ERRMES ;OUT OF BOUNDS
1587 001551 476532 420236 ASCIZ /OUT OF BOUNDS
1588 001552 431010 247652 /
1588 001553 472112 306424 /
1589 001554 000000 000000 >

```

```

1590                                   IFN FTLOGIN,<
1591                                   ;"SCHEDULE OCTN" - SETS RH OF STATES TO OCTN, IF TYPED FROM
1592                                   ; THE OPERATOR CONSOLE (C(DEVOPR)), OTHERWISE ILLEGAL
1593                                   ;"SCHEDULE" WITH NO ARGUMENT TYPES OUT RH OF STATES, LEGAL FOR ALL,
1594
1595                                   EXTERN DEVOPR,STATES,OCTPNT
1596
1597   001555   260140   002073' SKEDUL: PUSHJ   PDP,OCTIN           ;GET THE ARGUMENT IF ANY
1598   001556   254000   001565'           JRST   SKE01           ;NO ARGUMENT
1599   001557   254000   002000'           JRST   COMERA          ;BAD SYNTAX IN OCTAL NUMBER
1600   001560   200046   000000           MOVE   TAC,DEVNAM(DEV DAT) ;AN ARGUMENT, IS THIS THE OPR?
1601   001561   312040   000000           CAME   TAC,DEVOPR
1602   001562   254000   002001'           JRST   COMERR          ;NO, THIS IS ILLEGAL, THEN.
1603   001563   542100   000000           HRRM   TAC1,STATES       ;YES, STORE ARGUMENT IN RH OF STATES
1604   001564   263140   000000           POPJ   PDP,0           ;RETURN
1605
1606   001565   550040   001563' SKED1: HRRZ   TAC,STATES       ;SCHEDULE WITH NO ARGUMENTS.
1607   001566   260140   001500'           PUSHJ   PDP,OCTPNT       ;PRINT RH OF STATES,
1608   001567   254000   002016'           JRST   CRLF            ;AND RETURN WITH A CRLF
1609                                   >
1610                                   ;"BLANK" OR NO ALPHA NUMERIC BEFORE BREAK CHAR COMMAND
1611                                   ;DISPATCHED TO LIKE ANY OTHER COMMAND(0 IN COMMAND TABLE)
1612
1613
1614   001570                           CBLANK:
1615                                   IFE FTTTYSER,<
1616                                   LDB   TEM,TAC           ;GET BREAK CHARACTER
1617                                   CAIE   TEM,15           ;IS IT A CR
1618                                   >
1619                                   IFN FTTTYSER,<
1620   001570   302400   000012           CAIE   TEM,12           ;WAS BREAK A LF
1621                                   >
1622   001571   306400   000073           CAIN TEM,";"           ;NO, IS IT SEMI COLON(MONITOR COMMENT)
1623   001572   263140   000000           POPJ PDP,            ;YES, IGNORE
1624                                   ; FALL IN UNKNOWN COMMAND
1625
1626                                   ;COMMAND NOT IN COMMAND DIRECTORY
1627
1628   001573   254000   002001' NOCOM: JRST COMERR       ;NO, APPEND ? TO WHAT HE TYPED IN

```

```

1629                           IFN FTFINISH,<
1630                           ; "FINISH DEVICE" - CLOSES,RELEASE AND DEASSIGNS DEVICE
1631                           ;JOB MUST HAVE CORE
1632
1633                           EXTERNAL USRJDA,PUUOAC,JOB,USRHCU,JOBFDV,SYSTAP
1634   001574   260140   001723' CFINI:   PUSHJ PDP,CTEXT1           ;GET DEVICE NAME
1635   001575   322100   002034'           JUMPE TAC1,NOTENF       ;NOT ENOUGH ARG, IF NONE
1636   001576   334040   000002'           SKIPA TAC,TAC1           ;SEARCH FOR SIXBIT DEVICE NAME
1637   001577   200040   000705' CFINS:   MOVE TAC,SYSTAP           ;HERE ON CONTROL C ON SYSTEM TAPE USER
1638                                           ; DO A FINISH SYS COMMAND FOR HIM
1639   001600   260140   001435'           PUSHJ PDP,DEVSRC
1640   001601   254000   000770'           JRST NOTDEV             ;PRINT NOT A DEVICE
1641   001602   552307   001100'           HRRZM DEVDAT,JOBFDV(JDAT)   ;STORE DOB ADR, IN JOB DATA AREA
1642   001603   200303   777776'           MOVE DEVDAT,-2(PDP)
1643   001604   265100   001057'           JSP TAC1,MSTART         ;SETUP MONITOR JOB AND RETURN
1644                                           ; RETURN HERE AT UUO LEVEL WHEN SCHEDULED
1645   001605   265040   001060'           JSP TAC,MONSTR         ;SETUP ACS,PROG,JDAT,POP
1646   001606   200500   001070'           MOVE UCHN,USRHCU         ;HIGHEST USER IO CHANNEL IN USE
1647   001607   550312   001071' FDV1:   HRRZ DEVDAT,USRJDA(UCHN)   ;GET NEXT DEVICE
1648   001610   205600   071000'           MOVSI UUO,071000         ;SETUP RELEASE UUO
1649   001611   137500   001006'           DPB UCHN,PUUOAC         ;WITH CHANNEL NO.
1650   001612   261140   000012'           PUSH PDP,UCHN
1651   001613   316307   001602'           CAMN DEVDAT,JOBFDV(JDAT)   ;IS THIS DEV. THE ONE TO RELEASE?
1652   001614   256000   000014'           XCT UUO                 ;YES, RELEASE IT(AND CLOSE)
1653   001615   262140   000012'           POP PDP,UCHN
1654   001616   365500   001607'           SOJGE UCHN,FDV1         ;FINISHED?
1655   001617   200220   001101'           MOVE ITEM,JOB           ;GET JOB NUMBER
1656   001620   200307   001613'           MOVE DEVDAT,JOBFDV(JDAT)   ;RESET DEVDAT
1657   001621   260140   002043'           PUSHJ PDP,DEASG         ;DEASSIGN DEVICE
1658   001622   255000   000000'           JFCL                    ;IGNORE IF NOT ASSIGNED BY CONSOLE
1659   001623   254000   001107'           JRST ESTOP             ;STOP JOB SO HE CANNOT CONTINUE
1660
  >

```

```

1661                   IFN FTIME,<
1662                   ;"RESOURCES" - PRINT OUT AVAILABLE DEVICES AND FREE BLOCKS ON THE DISK
1663
1664                   EXTERNAL DEVLST
1665
1666                   FREDEV:
1667                   IFN     FTDISK, <
1668                   EXTERNAL     SATENT,SATXWD,WLBIT,SENTSZ,NUMBIT
1669
1670     001624   430040   000001           XOR TAC,TAC           ;INITIALIZE COUNTER
1671     001625   574600   000000           HLRE T2,SATXWD       ;SET INDEX WITH NUMBER OF SAT BLOCKS IN EXISTENCE,
1672     001626   213000   000014           MOVNS T2
1673     001627   541540   000000           HRR1 T1,SATENT       ;GET SATENT ENTRY POINTER
1674     001630   550113   000000     LOP05: HRRZ TAC1,0(T1)       ;GET VALUE
1675     001631   602100   000000           TRNE TAC1,WLBIT       ;IS THIS SPACE WRITE PROTECTED?
1676     001632   254000   001635'           JRST LOP06           ;YES, NO FREE BLOCKS IN THIS SAT BLOCK,
1677     001633   271040   000000           ADDI TAC,NUMBIT       ;NO, ADD THE FOLLOWING QUANTITY INTO THE
1678     001634   274040   000002           SUB TAC,TAC1           ; RUNNING TOTAL: (TOTAL BLOCKS IN THIS SAT
1679                                           ; BLOCK) - (BLOCKS ALREADY IN USE),
1680     001635   271540   000000     LOP06: ADDI T1,SENTSZ       ;BUMP POINTER
1681     001636   366600   001630'           SOJN T2,LOP05       ;DECREMENT INDEX
1682     001637   260140   001430'           PUSHJ PDP,RADX10     ;CONVERT TO DECIMAL
1683     001640   260140   002025'           PUSHJ PDP,INLMES     ;PRINT
1684     001641   271010   246226           ASCII? /. BLKS/
1685                   >
1686
1687     001643   554500   000652'           HLRZ T,DEVLST       ;GET DDB POINTER
1688     001644   201700   000000           MOVEI AC2,0          ;SET DEVICE NAME 0 FOR FIRST COMPARE
1689     001645   200552   000004     LOP01: MOVE T1,DEVMOD(T)     ;DEVICE CHARACTERISTICS
1690     001646   606540   600000           TRNN T1,ASSCON:ASSPRG;DEVICE ASSIGNED BY CONSOLE OR PROGRAM?
1691     001647   603540   200010           TLNE T1,DVTY!DVDSK   ;NO, IS IT A TTY OR DSK?
1692     001650   254000   001662'           JRST LOP02           ;YES DO NOT PRINT
1693                   IFE FTDISK,<
1694                   JUMPE AC2,LOP01B       ;SUPPRESS LEADING COMMA IF NO DISK
1695                   >
1696     001651   260140   002025'           PUSHJ PDP,INLMES     ;PRINT ,(INSTEAD OF CRLF SO WILL FIT IN 1 BUFFER)
1697     001652   260000   000000           ASCII? //
1698     001653   204652   000000     LOP01B: MOVS AC1,DEVNAM(T)   ;GET DEVICE NAME
1699     001654   510100   000015           HLLZ TAC1,AC1       ;ASSUME SAME TYPE AS LAST ONE, PRINT
1700                                           ; ONLY RH OF NAME (UNIT NUMBER)
1701     001655   306715   000000           CAIN AC2,0(AC1)      ;IS IT REALLY THE SAME?
1702     001656   254000   001661'           JRST LOP01A          ;YES, PRINT THE UNIT NUMBER,
1703     001657   204100   000015           MOVS TAC1,AC1       ;NO, MUST PRINT WHOLE NAME,
1704     001660   550700   000015           HRRZ AC2,AC1        ; AND GET THE NEW DEVICE IN FLAG AC.
1705     001661   260140   001216'     LOP01A: PUSHJ PDP,PRNAME   ;AS BEING FREE
1706     001662   554512   000003     LOP02: HLRZ T,DEVSER(T)   ;GET NEXT DEVICE IN CHAIN
1707     001663   326500   001645'           JUMPN T,LOP01       ;IS THERE ONE?
1708     001664   254000   002016'           JRST CRLF           ;NO, DONE, PRINT CR, LF AND THEN POPJ
1709                   >

```

```

1710                                     EXTERNAL CPOPJ,STUSER
1711                                     ;ROUTINE TO CHECK FOR ACTIVE DEVICES
1712                                     ;NEVER GET HERE DURING SWAP IN OR OUT
1713                                     ;SINCE COMMAND TABLE SHOULD HAVE NOTRAN BIT ON
1714
1715 001665 316200 000455' RUNCHK: CAMN ITEM,STUSER           ;SYSTEM TAPE USER?
1716 001666 254000 001677'           JRST DLYCM1           ;YES, DELAY,BUT DO NOT STOP JOB
1717 001667 322340 001672' ACTCHK: JUMPE PROG,ACTCH1       ;DOES JOB HAVE CORE IN MEMORY?
1718 001670 260140 000000           PUSHJ PDP,ANYACT       ;YES, ARE DEVICES ACTIVE?
1719 001671 254000 001676'           JRST DLYCM           ;YES, DELAY COMMAND,
1720 001672 200103 777777 ACTCH1: MOVE TAC1,-1(PDP)       ;RESTORE COMMAND DISPATCH ADDRESS
1721 001673 200303 777776           MOVE DEVDAT,-2(PDP)   ;RESTORE TTY DDB ADDRESS
1722 001674 200504 001414'           MOVE T,JBTSTS(ITEM)   ;RESTORE JOB STATUS
1723 001675 263140 000001' CONTC1: POPJ PDP,COM0
1724
1725                                     ;ROUTINE TO DELAY A COMMAND
1726
1727                                     ;DELAYS COMMAND TO BE EXECUTED WHEN JOB IN CORE MEMORY
1728                                     ;AND CAUSES JOB TO BE SWAPPED IN(COMMAND WAIT BIT IS SET IN JBTSTS)
1729                                     ;AND POPS LEVEL UP ONE,
1730
1731                                     EXTERNAL TTYCM,LINSAV
1732
1733 001676 260140 000000 DLYCM:  PUSHJ  PDP,DLYCOM           ;SET COMMAND WAIT BIT IN JOB STATUS AND PUT
1734                                     ; JOB IN COMMAND WAIT QUEUE
1735                                     ; SO JOB IS NOT RUNNABLE ANY MORE
1736                                     ;ROUTINE TO DELAY A COMMAND IF A SWAP OUT OR IN IS IN PROGRESS
1737                                     ;DIFFERS FROM DLYCM IN THAT JOB IS NOT MADE TO BE SWAPPED IN
1738                                     ;REQUIRED FOR DELAYING COMMAND IF SYSTEM
1739                                     ;TAPE USER TYPES +C (HALT COMMAND)
1740
1741 001677 262140 000001 DLYCM1: POP      PDP,TAC
1742 001700 262140 000002           POP      PDP,TAC1
1743 001701 262140 000002           POP      PDP,TAC1
1744 001702 261140 001675'           PUSH    PDP,CONTC1
1745 001703 200100 000000           MOVE    TAC1,LINSAV
1746 001704 254000 000000           JRST   TTYCM
  
```

```

1747
1748          IFN FTDISK,<
1749          ;SEARCH DEVICE CHAIN FOR DSK WITH COUNTS ON
1750          ;CALL  MOVE ITEM,JOB NO
1751          ;      PUSHJ POP,STOPCK
1752          ;      CAN'T STOP RETURN
1753          ;      OK TO STOP RETURN
1754
1755          EXTERNAL DSKOBB,DSKFGS,CPOPJ1
1756
1757 001705 201300 000000 STOPCK: MOVEI DEVDAT,DSKOBB
1758 001706 205100 000000          MOVSI TAC1,DSKFGS
1759 001707 135040 001271' STOPD:  LDB TAC,PJOBN
1760 001710 302201 000000          CAIE ITEM,(TAC)          ;ASSIGNED TO THIS JOB?
1761 001711 254000 001714'          JRST STOPC          ;NO
1762 001712 612106 000002          TDNE TAC1,DEVIOS(DEVDAT) ;YES, FLAG ON?
1763 001713 254000 000757'          JRST CPOPJ          ;YES, DELAY
1764 001714 554306 000003 STOPC:  HLRZ DEVDAT,DEVSER(DEVDAT)
1765 001715 205040 446353          MOVSI TAC,446353

1766 001716 316046 000000          CAMN TAC,DEVNAM(DEVDAT)
1767 001717 254000 001707'          JRST STOPD
1768 001720 254000 000000          JRST CPOPJ1          ;OK TO STOP
1769
  >
  
```

```

1770                                   SUBTTL  COMCSS - COMMON COMMAND SUBROUTINES
1771
1772                                   ;ROUTINE TO RETURN NEXT ALPHANUMERIC STRING
1773                                   ; IN COMMAND LINE (SIXBIT)
1774                                   ;CALL:  MOVE TAC,BYTE POINTER TO PREVIOUS CHAR.
1775                                   ;        PUSHJ PDP, CTEXT
1776                                   ; SIXBIT STRING RETURN LEFT JUSTIFIED IN AC TAC1
1777
1778                                   INTERNAL CTEXT
1779                                   INTERNAL CTEXT1
1780
1781                                   000012 T=BUF PNT                           ;TEMPORARY AC'S
1782                                   000013 T1=BUFWRD
1783                                   000014 T2=UUO
1784
1785 001721 260140 001756' CTEXT:  PUSHJ  PDP,SKIPS                           ;CALL HERE IF AT START OF LINE
1786 001722 334000 000000                                    SKIPA
1787
1788 001723 260140 001757' CTEXT1: PUSHJ PDP,SKIPS1                           ;SKIP LEAD SPACES,TABS,NULLS AND CLEAR TAC1
1789                                   ; DO NOT RETURN IF CR WAS PREVIOUS BREAK
1790 001724 200500 003063'                                    MOVE T,(POINT 6,TAC1)
1791                                   IFE FTITYSER,<
1792                                    LDR     TEM,TAC
1793                                    JRST CTEX1
1794                                   CTEX0:  ILDB    TEM,TAC
1795                                   >
1796                                   IFN FTITYSER,<
1797                                    EXTERN TAKR,GETCHR
1798 001725 135405 000000                                    LDB     TEM,TAKR(DAT)                   ;GET LAST CHAR.
1799 001726 254000 001730'                                    JRST    CTEX1
1800 001727 260140 000227' CTEX0:  PUSHJ  PDP,GETCHR                       ;ROUTINE IN SCNSER TO PICK UP CHAR.
1801                                   >
1802
1803 001730 260140 001735' CTEX1:  PUSHJ PDP, CTEX                           ;IS IT ALPHANUMERIC
1804 001731 640400 000040                                    TRC TEM,40                           ;CONVERT TO SIXBIT
1805 001732 603500 770000                                    TLNE    T,770000                     ;SIX CHARS YET?
1806 001733 136400 000012                                    IDPB    TEM,T                       ;NO. BUILD WORD
1807 001734 254000 001727'                                    JRST CTEX0                       ;LOOP FOR MORE

```

```

1808                               ;SCAN FOR ALPHANUMERIC CHAR IN TEM
1809 001735 303400 000172 CTEX:  CAILE  TEM,"Z"+40      ;GREATER THAN LC Z?
1810 001736 254000 001747' JRST  CTEXA      ;YES, NOT SIXBIT.
1811 001737 301400 000141 CAIL  TEM,"A"+40      ;LOWER CASE LETTER?
1812 001740 620400 000040 TRZ   TEM,40        ;YES, MAKE UPPER CASE.
1813 001741 301400 000060 CAIL  TEM, "0"
1814 001742 303400 000132 CAILE  TEM, "Z"      ;LETTERS ARE LARGER THAN NOS.
1815 001743 254000 001747' JRST  CTEXA      ;NEITHER
1816 001744 303400 000071 CAILE  TEM, "9"
1817 001745 301400 000101 CAIL  TEM, "A"
1818 001746 263140 000000 POPJ  PDP,          ;LETTER OR NUMBER RETURN
1819                               CTEXA:  IFE FTTYSER,<
1820                               POP   PDP,T1      ;REDUCE PDP BY 1 LEVEL
1821                               POPJ  PDP,          ;AND RETURN
1822                               >
1823                               IFN FTTYSER,<
1824 001747 306400 000072 CAIN  TEM,":"          ;DEVICE NAME?
1825 001750 260140 001727' PUSHJ  PDP,GETCHR    ;YES, SKIP COLON
1826 001751 302400 000003 CAIE   TEM,3          ;CONTROL C?

1827 001752 254000 001503' JRST  TPOPJ      ;NO, RETURN ONE LEVEL UP
1828 001753 205100 504154 CTXCNC: MOVSI  TAC1,(SIXBIT /HAL/) ;MAKE PHONY HALT COMMAND
1829 001754 201400 000012 MOVEI  TEM,12      ;WITH LF FOR TERMINATION
1830 001755 254000 001752' JRST  TPOPJ      ;AND RETURN UP A LEVEL
1831                               >

```

```

1832 ;ROUTINE TO IGNORE LEADING SPACES, TABS, AND NULLS
1833 ;ALSO CLEARS TAC1
1834 ;DOES NOT RETURN IF PREVIOUS CHAR, OR NEXT NON-SPACING
1835 ;CHAR. IS CR(IE POPS SUBROUTINE LEVEL UP 1 ON RETURN)
1836 ;CALL: MOVE TAC,BYTE POINTER TO PREVIOUS BREAK CHAR.
1837 ; PUSHJ PDP, SKIPS1
1838
1839 INTERNAL SKIPS1,SKIPS
1840
1841 IFE FTTYSER,<
1842 SKIPS1:
1843 SKIPS: MOVEI TAC1,0 ;CLEAR TAC1
1844 LDB TEM,TAC ;WAS PRECEDING BREAK A CR?
1845 CAIN TEM,15
1846 JRST SKIPS2 ;YES, POP SUB, LEVEL UP 1
1847
1848 SKIPSA:
1849 ILDB TEM,TAC
1850 JUMPE TEM,SKIPSA ;NULL?
1851 CAIE TEM," " ;SPACE?
1852 CAIN TEM, 11
1853 JRST SKIPSA
1854 CAIN TEM,15 ;CR?
1855 SKIPS2: POP PDP,T ;YES, POP SUB, LEVEL UP ONE
1856 POPJ PDP,
1857 >
1858 IFN FTTYSER,< EXTERNAL GETCHR,SPCHEK,BREAKB,TAKR
1859
1859 001756 260140 001750' SKIPS: PUSHJ PDP,GETCHR ;GET FIRST CHAR ON LINE
1860 001757 201170 000000' SKIPS1: MOVEI TAC1,0 ;FOR CTEXT, DECIN
1861 001760 135425 001725' LDR TEM,TAKR(DAT) ;IN CASE TEM CLOBBERED
1862 001761 306400 000015' CAIN TEM,15 ;SKIP TO LF IF CR
1863 001762 260140 001756' SKIPSA: PUSHJ PDP,GETCHR ;NEXT CHARACTER
1864 001763 322470 001773' JUMPE TEM,SKIPS3 ;POP UP A LEVEL IF NO CHARS
1865 001764 260140 000223' PUSHJ PDP,SPCHEK ;SPECIAL?
1866 001765 254000 001770' JRST SKIPS2 ;NO
1867 001766 603040 000225' TLNE TAC,BREAKB ;BREAK?(END OF LINE)
1868 001767 254070 001773' JRST SKIPS3 ;YES, POP UP RETURN
1869 001770 307420 000040' SKIPS2: CAIG TEM,40 ;SPACE OR CONTROL CHAR?
1870 001771 254070 001762' JRST SKIPSA ;YES,
1871 001772 263140 000000' POPJ PDP,0 ;NO, RETURN
1872 001773 306400 000003' SKIPS3: CAIN TEM,3 ;*C?
1873 001774 254020 001753' JRST CTXCNC ;HANDLE IT
1874 001775 201470 000012' MOVEI TEM,12 ;FOR BREAK CONSISTENCY
1875 001776 254020 001755' JRST TPOPJ ;RETURN ONE LEVEL UP
1876 >

```

```

1877          ;ROUTINE TO APPEND A "?" TO INPUT STRING AND SET AS OUTPUT
1878          ;CALLED FROM OCTIN, RETURNS TO SECOND LEVEL ON PDL
1879          ;CALL: MOVE TAC, BYTE POINTER TO LAST CHAR, IN INPUT STRING
1880          ;      PUSHJ PDP, COMERA
1881
1882          INTERNAL COMERA,COMERP
1883
1884 001777 262140 000012 COMERP: POP PDP,T          ;REMOVE SUB, RETURN BEFORE CALLING COMERA
1885 002000          COMERA: IFE FTITYSER,< IBP TAC>
1886 002000 260140 001762' IFN FTITYSER,< PUSHJ PDP,GETCHR> ;MOVE UP, A CH
1887
1888
1889          ;ROUTINE TO REPLACE LAST CHARACTER IN INPUT STRING BY "?"
1890          ;AND SET AS OUTPUT
1891          ;CALL: MOVE TAC, BYTE POINTER TO LAST CHAR, IN INPUT STRING
1892          ;      PUSHJ PDP, COMERR
1893
1894          INTERNAL COMERR
1895
1896
1897 002001          COMERR:
1898          IFN FTITYSER,<
1899          EXTERN OUTCHS,TRESCN,TITAKR,SETBFI,TISYNC
1900
1901 002001 200506 000147' MOVE T,TITAKR(DEVPAT) ;SAVE POSITION IN SCAN
1902 002002 260140 000000 PUSHJ PDP,TRESCN ;BACK UP TO START OF COMMAND
1903 002003 260140 002000' COMERL: PUSHJ PDP,GETCHR ;GET A CHAR FROM COMMAND
1904 002004 316526 002001' CAMN T,TITAKR(DEVPAT) ;PASS BAD CHAR YET?
1905 002005 254000 002010' JRST COMER1 ;YES
1906 002006 260140 000000 PUSHJ PDP,OUTCHS ;NO. TYPE CHAR,
1907 002007 326400 002003' JUMPN TEM,COMERL ;LOOP IF NOT OUT OF CHARACTERS
1908
1909 002010 260140 000000 COMER1: PUSHJ PDP,SETBFI ;CLEAR ANY MORE TYPEIN
1910 002011 350006 000151' AOS TISYNC(DFVDAT) ;THIS WILL BE SOS-ED AT COMRET
1911 002012 201400 000077 MOVEI TEM,"?" ;APPEND ? TO ERRONEOUS WORD
1912 002013 260140 002006' PUSHJ PDP,OUTCHS
1913
1914          >
1915          IFE FTITYSER,<
1916          MOVE DAT,TAC
1917          MOVEI T,"?" ;STORE ? ON TOP OF BREAK CHAR.
1918          DFR T,DAT
1919          >
1919 002014 634202 000074          TDPA ITEM,ITEM ;CLEAR JOB NO. AND SKIP INTO CRLF ROUT

```

```

1920          ;ROUTINE TO PRINT A COMMAND ERROR MESSAGE
1921          ;SAME CALL AS CONMES
1922
1923          INTERNAL ERRMES
1924
1925          ; COMERR SKIPS THIS ROUT.(SEE ABOVE)
1926 002015 634200 000004 ERRMES: TOZA ITEM,ITEM      ;CLEAR JOB NUMBER TO INDICATE ERROR
1927          ; SKIP INTO CONMES ROUTINE
1928
1929          ;ROUTINE TO PRINT CARRIAGE RETURN-LINE-FEED
1930          ;CALL: MOVE DAT,BYTE POINTER TO OUTPUT
1931          ;      PUSHJ PDP,CRLF
1932
1933          INTERNAL CRLF
1934
1935          ; COMERR SKIPS TO HERE(SEE ABOVE)
1936          CRLF: MOVEI TAC,(ASCII /
1937 002016 201040 003064' /)
1938
1939          ;ROUTINE TO MOVE ASCII CHAR. STRING TO CONSOLE OUTPUT BUFFER
1940          ; CALL: MOVE DAT, BYTE POINTER TO LAST OUTPUT CHARACTER
1941          ;      MOVEI TAC, ADDRESS OF ASCII MESSAGE
1942          ;      PUSHJ PDP, CONMES
1943          ; STRING TERMINATED BY NULL
1944          ; CONMES DOES NOT START TTY
1945          ;CONMS1 - SAME CALLING SEQUENCE AS CONMES,EXCEPT LH IS BYTE POINTER
1946
1947          INTERNAL CONMES
1948          EXTERNAL TPOPJ
1949
1950          ; ERRMES SKIPS TO HERE
1951 002017 505040 440700 CONMES: WRLI TAC,440700      ;FORM ASCII BYTE POINTER
1952 002020 261140 000001 CONMS1: PUSH PDP,TAC      ;SAVE BYTE POINTER
1953 002021 134403 000000 CON0:  ILOB TEM,(PDP)    ;GET NEXT CHAR.
1954 002022 322400 001776' JUMPE TEM,TPOPJ      ;IS IT NULL?(IF YES, DO POP TAC, POPJ)
1955 002023          TYOINS:          ;TAG FOR THE TYPE OUT INSTRUCTION
1956 002023          CONOUT:          ;EXECUTED FROM REST OF COMCSS TO OUTPUT CHAR
1957
1958          IFN FITTYSER,<
1959 002023 260140 002013'      PUSHJ PDP,OUTCHS      ;NO, STORE TTY OUTPUT BUFFER
1960          >
1961          IFE FITTYSER,<
1962          IOPB TEM,DAT
1963          >
1964 002024 254000 002021'      JRST CON0          ;KEEP GOING
  
```

```

1965                               ;ROUTINE TO PRINT INLINE ASCIZ MESSAGE
1966                               ;CALL:  PUSHJ PDP,INLMES
1967                               ;      ASCIZ /THE MESSAGE/
1968                               ;RETURN TO NEXT LOC AFTER MESSAGE
1969
1970                               INTERNAL INLMES
1971
1972   002025  262140  000001  INLMES:  POP PDP,TAC           ;SETUP PRINT ADRESS FOR CONMES
1973   002026  260140  002017'  PUSHJ PDP,CONMES
1974   002027  254001  000001  JRST 1(TAC)           ;RETURN TO NEXT LOC AFTER MESSAGE
1975
1976                               ;ROUTINE TO APPEND ? TO ERROR MESSAGE
1977                               ;CALL  PUSHJ PDP,PRQM
1978                               ;      RETURN
1979
1980                               INTERNAL PRQM
1981
1982   002030  201400  000077  PRQM:  MOVEI TEM,"?"
1983   1983     IFN FTTYSER,<

1984                               EXTERN OUTCHS
1985   002031  254000  002023'  JRST  OUTCHS
1986   1986     >
1987   1987     IFE FTTYSER,<
1988   1988     IOPB  TEM,DAT
1989   1989     POPJ  PDP,
1990   1990     >
1991   002032  201040  003065'  PRSPC:  MOVEI TAC,[ASCIZ /
1992   002033  254000  002017'  JRST CONMES
  
```

```
1993
1994
1995
1996           ;ROUTINE TO PRINT "TOO FEW ARGUMENTS"
1997           ;CALL:  MOVE DAT,BYTE POINTER
1998           ;           PUSHJ PDP,NOTENF
1999
2000           INTERNAL NOTENF
2001 002034 265040 002015' NOTENF: JSP TAC,ERRMES
2002
2003 002035 522371 720214 ASCIZ /TOO FEW ARGUMENTS
      002036 426564 040644
      002037 436531 542634
2004 002040 522461 505000 /
2005
2006           ;ROUTINE TO PRINT A PERIOD
2007           ;CALL:  PUSHJ PDP,PRPER
2008
2009           INTERNAL PRPER
2010
2011 002041 265040 002017' PRPER:  JSP TAC,CONMES
2012 002042 270000 000000         ASCIZ ./.
```

```

2013
2014
2015      ;ROUTINE TO DEASSIGN A DEVICE
2016      ;CALL:  MOVE DEVDAT, DEVICE DATA BLOCK
2017      ;
2018      ;         PUSHJ PDP, DEASG
2019      ;         ERROR NOT PREVIOUSLY ASSIGNED
2020      ;         OK RETURN WITH DEVICE DEASSIGNED
2021
2022      INTERNAL DEASG
2023      EXTERNAL PJOBN,CPPOPJ1,IPOPJ
2024 002043 135500 001707' DEASG:  LDR T, PJOBN           ;WAS DEVICE ASSIGNED TO THIS JOB?
2025 002044 312500 000004      CAME T, ITEM
2026 002045 263140 000000      POPJ PDP,           ;NO, RETURN
2027 002046 261140 000004      PUSH PDP,ITEM      ;SAVE JOB NUMBER
2028 002047 205100 400000      MOVSI TAC1,OVDIRI ;CLEAR DIRECTORY IN CORE BIT
2029 002050 413106 000004      ANDCAB TAC1,DEVMOD(DEVDAT) ;SET DEVICE CHARACTERISTICS FOR TEST
2030
2031 002051 402006 000005      SETZM DEVLOG(DEVDAT) ;CLEAR LOGICAL NAME
2032
2033 002052 602100 400000      TRNE TAC1,ASSCON   ;IS DEVICE ASSIGNED BY CONSOLE?
2034 002053 350003 777777      AOS -1(PDP)       ;YES, DO OK RETURN
2035
2036 002054 260140 000741' IFN FT2REL,<
2037      EXTERN ASGHGH
2038      PUSHJ PDP,ASGHGH ;IF DTA OR MTA, CLEAR ANY HIGH SEGMENT NAMES
2039      ; FROM THIS DEVICE SO NO NEW SHARING
2040      ; DEVMOD SETUP IN TAC1 ON CALL
2041
2042 >
2043 002055 201100 400000      MOVEI TAC1,ASSCON ;SETUP ASSIGNED BE CONSOLE BIT FOR RELEASE
2044 002056 260140 000000      PUSHJ PDP,RELEASE ;CLEAR JOB NO, IN DDB IF DDB NOT NEEDED
2045 002057 254000 000000      JRST IPOPJ       ;RESTORE JOB NUMBER AND RETURN
2046
2047      ;ROUTINE TO DEASSIGN ALL DEVICES EXCEPT LOGICAL TTY
2048      ;CALL:  MOVE ITEM, JOB NUMBER
2049      ;
2050      ;         MOVE DEVDAT,ADR, OF DEVICE NOT TO BE DEASSIGNED
2051      ;         PUSHJ PDP, DEASTY
2052
2053      INTERNAL DEASAL,DEASTY
2054      EXTERNAL DEVLST
2055
2056 002060 261140 000006      DEASTY:
2057 002061 554300 001643' DEASAL:  PUSH PDP,DEV DAT      ;SAVE TTY DDB ADDRESS
2058 002062 302323 000000      HLRZ DEV DAT,DEVLST ;SEARCH ALL DDB'S
2059 002063 260140 002043' DEAI:   CAIE DEV DAT,*(PDP) ;IS THIS DEVICE NOT TO BE DEASSIGNED?
2060 002064 255000 000000      PUSHJ PDP, DEASG    ;NO, TRY TO DEASSIGN IT
2061 002065 554306 000003      JFCL              ;IGNORE IF CAN'T
2062 002066 326300 002062'      HLRZ DEV DAT, DEVSER(DEV DAT)
2063 002067 262140 000006      JUMPN DEV DAT, DEAI
2064 002070 263140 000000      POP PDP,DEV DAT   ;RESTORE TTY DDB ADDRESS
2065      POPJ PDP,

```

```

2063
2064
2065 ;ROUTINE TO READ CONSOLE AND CONVERT ANY RADIX NUMBER
2066 ; CALL: MOVE TAC1, DESIRED RADIX
2067 ; MOVE TAC, BYTE POINTER TO FIRST CHARACTER
2068 ; PUSHJ PDP, ANYRIN
2069 ; NO ARG, TYPED RETURN, TAC1=0
2070 ; ILLEGAL CHARACTER RETURN
2071 ; NORMAL EXIT ;AC TAC1 CONTAINS NUMBER
2072 ;SCAN STOPS ON FIRST CR,DASH,SPACE,OR TAB OR ILLEGAL CHAR.
2073 ;SKIPS LEADING SPACES AND TABS
2074
2075 INTERNAL OCTIN,OCTIN1,DECIN,DECIN1
2076 EXTERNAL CPOPJ1,CPOPJ2
2077
2078 000012 C=BUFPNT ;CHARACTER AC
2079 000013 R=BUFWRD ;RADIX AC
2080
2081 002071 DECIN1:
2082 002071 201540 000012 DECIN: MOVEI R,12 ;DECIMAL INPUT
2083 002072 254000 002074' JRST ANYRIN
2084
2085 002073 OCTIN1:
2086 002073 201540 000010 OCTIN: MOVEI R,10 ;OCTAL INPUT
2087 002074 260140 001757' ANYRIN: PUSHJ PDP,SKIPS1 ;SKIP LEADING SPACES, TABS, NULLS
2088 ; DO NOT RETURN IF CR WAS PREVIOUS BREAK
2089 ; OR THIS BREAK
2090
2091 002075 305400 000175 IFE FTTYSER,< LDR TEM,TAC>
2092 002076 306400 000133 OCT0: CAIGE TEM,175 ;ALTHODES(175 OR 176)?
2093 ;NO, LEFT BRACKET(SO SPACE NOT REQ
2094 ; BEFORE [P,P] IN ATT AND GET COMMANDS.
2095
2096 002077 254000 000000 JRST CPOPJ2 ;YES, SKIP RETURN
2097 002100 302400 000055 CAIE TEM,"-" ;DASH?
2098 002101 307400 000040 CAIG TEM,40 ;SPACE OR CONTROL CHAR?
2099 002102 254000 002077' JRST CPOPJ2 ;YES, ONLY LEGAL TERMINATORS
2100 002103 302400 000054 CAIE TEM,", " ;COMMA?
2101 002104 306400 000135 CAIN TEM,"]" ;RIGHT BRACKET?
2102 002105 254000 002102' JRST CPOPJ2 ;YES.
2103 002106 275400 000060 SUBI TEM,60
2104 002107 321400 001720' JUMPL TEM,CPOPJ1 ;ERROR IF NEG.,REMOVE SUB. RETURN AND PRINT
2105 002110 311400 000013 CAML TEM, R ;OR .GE, RADIX
2106 002111 254000 002107' JRST CPOPJ1 ;ERROR, GREATER OR EQUAL TO RADIX
2107 002112 220100 000013 IMUL TAC1, R
2108 002113 270100 000010 ADD TAC1,TEM
2109
2110 IFE FTTYSER,< ILDB TEM,TAC ;NEXT CHAR>
2111 IFN FTTYSER,< PUSHJ PDP,GETCHR> ;NEXT CHAR
2112 JRST OCT0

```

```

2109                   INTERNAL FTLOGIN
2110                   IFN FTLOGIN,<
2111                   ;GET PROJECT-PROGRAMMER NUMBERS
2112                   ;CALL: MOVE TAC,INPUT BYTE POINTER
2113                   ;            PUSHJ PDP,PJPGNO
2114                   ;
2115                   ;(AC2)LH ← PROJECT NUMBER
2116                   ;(AC2)RH ← PROGRAMMER NUMBER
2117                   ;(AC2) = 0 IF NO [ ]'S TYPED
2118                   ;THE TERMINAL ] IS OPTIONAL
2119
2120
2121                   IFE FTTYSER,<
2122                   PJPGNO: SKIPA    TAC1,TAC           ;SAVE INPUT BYTE POINTER IN CASE NO
2123                   ; [ ]'S WERE TYPED IN,
2124                   PP0:    IBP    TAC           ;GET NEXT CHARACTER (2ND TIME THRU LOOP)
2125                   ;            LDB    TEM,TAC       ;GET CHAR WHICH STOPPED PREVIOUS FIELD SCAN
2126                   ;                               ; (OR NEXT CHAR ON 2ND TIME THRU LOOP)
2127                   >

2128                   IFN FTTYSER,<
2129                   002116 260140 001757' PP0:    PUSHJ    PDP,SKIPS1
2130                   002117                   PJPGNO:
2131                   >
2132                   002117 306400 000133            CAIN    TEM,"["       ;IS IT A "[" ?
2133                   002120 254000 002126'        JRST    PP1           ;YES, GET PROJECT-PROGRAMMER NUMBERS FROM INSIDE.
2134                   002121 302400 000040        CAIE    TEM," "       ;NO, IS IT A SPACE ?
2135                   002122 306400 000011        CAIN    TEM,11       ; OR A TAB ?
2136                   002123 254000 002116'        JRST    PP0           ;YES, KEEP LOOKING FOR "[",
2137                   002124 201700 000000        MOVEI   AC2,0       ;NEITHER SPACE NOR TAB, THUS RETURN A 0 MEANING
2138                   ; NO PROJECT-PROGRAMMER NUMBER ENCOUNTERED.
2139                   IFE FTTYSER,<    MOVEM    TAC1,TAC       ;RESTORE ORIGINAL BYTE POINTER,>
2140                   002125 263140 000000        POPJ     PDP,           ;EXIT,.....
2141
2142                   PP1:
2143                   002126 260140 001756' IFN FTTYSER,< PUSHJ    PDP,SKIPS>
2144                   002127 260140 002073'            PUSHJ PDP,OCTIN       ;GET FIRST ARG.--PROJ. NO.
2145                   002130 254000 001777'            JRST COMERP       ;NO ARG. GIVEN
2146                   002131 254000 001777'            JRST COMERP       ;ILLEGAL DECIMAL CHARACTER GIVEN
2147                   002132 504700 000002            HRL AC2,TAC1       ;ENTER
2148                   002133 302400 000054 IFN FTTYSER,< CAIE    TEM,""
2149                   002134 254000 001777'        JRST    COMERP
2150                   002135 260140 001756'            PUSHJ PDP,SKIPS>
2151                   002136 260140 002073'            PUSHJ PDP,OCTIN       ;GET SECOND ARG.--PROG. NO.
2152                   002137 254000 001777'            JRST COMERP       ;
2153                   002140 254000 001777'            JRST COMERP       ;
2154                   002141 540700 000002            HRR AC2,TAC1       ;ENTER
2155                   002142 260140 001757'            PUSHJ PDP,SKIPS1     ;SKIP BLANKS
2156                   002143 306400 000135            CAIN TEM,"J"       ;IS USUAL ENDING A "J"?
2157                   IFE FTTYSER,<IBP TAC>           ;YES, SKIP IT
2158                   002144 260140 002114' IFN FTTYSER,<PUSHJ PDP,GETCHR>   ;YES, SKIP IT SO FINAL ] IS OPTIONAL
2159                   002145 263140 000000            POPJ PDP,           ;RETURN TO CALL
2160                   >

```

```

2161 ;ROUTINE TO PRINT TIME AS HOURS,MINUTES,SECONDS, AND HUNDRETHS
2162 ;FORMAT IS HHMM:SS.HH
2163 ;CALL: MOVE TAC,TIME IN JIFFIES(60THS,50THS OR MILLISECONDS)
2164 ; MOVE DAT,OUTPUT TEXT BYTE POINTER
2165 ; PUSHJ PDP,PRTIME
2166
2167 ;SCALEING IS DONE USING THE FOLLOWING GLOBAL SYMBOLS DEFINED
2168 ;ON THE CONFIGURATION TAPE (IOINI1)
2169 ;THUS ANY INSTALLATION MAY HAVE ANY RATE CLOCK
2170
2171 EXTERNAL JIFMIN,JIFSEC,JIFSC2
2172
2173 ;JIFMIN=NO. OF JIFFIES(CLOCK TICKS) PER MINUTE
2174 ;JIFSEC=NO. OF JIFFIES PER SECOND
2175 ;JIFSC2=1/2*JIFSEC(USED FOR ROUNDING)
2176
2177 INTERNAL PRTIME
2178
2179 002146 231040 001325' PRTIME: IDIVI TAC,JIFMIN ;FORM MINUTES
2180 002147 261140 000002' PUSH PDP,TAC1 ;SAVE REMAINDER IN JIFFIES
2181 002150 322040 002162' JUMPE TAC,PR1 ;SUPPRESS 0 HOURS IN MINUTES
2182 002151 231040 000074' IDIVI TAC,*D60 ;HOURS, MINUTES IN TAC,TAC1
2183 002152 322040 002156' JUMPE TAC,PR0 ;SUPPRESS 0 HOURS
2184 002153 260140 001637' PUSHJ PDP,RADX10
2185 002154 260140 002025' PUSHJ PDP,INLMES ;PRINT "HH:" OR "H:"
2186 002155 350000 000000' ASCIZ /:/
2187 002156 200040 000002' PR0: MOVE TAC,TAC1 ;GET MINUTES
2188 002157 260140 002274' PUSHJ PDP,PRT2 ;PRINT "MM:"
2189 002160 260140 002025' PUSHJ PDP,INLMES
2190 002161 350000 000000' ASCIZ /:/
2191 002162 262140 000001' PR1: POP PDP,TAC ;RESTORE SECONDS(IN JIFFIES)
2192 002163 231040 001427' IDIVI TAC,JIFSEC ;JIFFIES PER SECOND
2193 002164 260140 002153' PUSHJ PDP,RADX10 ;PRINT SECONDS
2194 002165 260140 002041' PUSHJ PDP,PRPER ;PRINT PERIOD
2195 002166 200040 000002' MOVE TAC,TAC1 ;NO OF JIFFIES(HUNDRETHS)
2196 002167 221040 000144' IMULI TAC,*D100 ;CONVERT TO HUNDRETHS
2197 002170 231040 002163' IDIVI TAC,JIFSEC
2198 002171 301100 000000' CAIL TAC1,JIFSC2 ;ROUND IF GREATER THAN HALF
2199 002172 350000 000001' AOS TAC
2200 002173 254000 002272' JRST PRT2LF ;PRINT
  
```

```
2201 002174 231040 002146' PRTIM1: IDIVI TAC,JIFMIN            ;  
2202 002175 231040 000074            IDIVI TAC,*D60            ;HOURS,MINUTES IN TAC,TAC1  
2203 002176 260140 002204'            PUSHJ PDP,PRT2  
2204 002177 260140 002025'            PUSHJ PDP,INLMES        ;PRINT "HH:"  
2205 002200 350000 000000            ASCIZ /!/  
2206 002201 200040 000002            MOVE TAC,TAC1  
2207 002202 260140 002204' PRT2LF: PUSHJ PDP,PRT2            ;PRINT MM  
2208 002203 254000 002016'            JRST CRLF  
2209  
2210 002204 201400 000060 PRT2:        MOVEI TEM,"0"  
2211 002205 305040 000012            CAIGE TAC,*D10  
2212 002206 256000 002023'            XCT CONOUT            ;PUT LEADING 0 IF LESS THAN 10  
2213 002207 254000 002164'            JRST RADX10            ;PRINT REST OF NUMBER
```

```

2214          ;ROUTINE TO PRINT SIZE OF LOGICAL SEGMENT (LOW OR HIGH)
2215          ;CALL:  MOVE ITEM, HIGH OR LOW SEG NUMBER
2216          ;      PUSHJ PDP,PRT SEG
2217          ;      RETURN
2218          ;CALLED AT CLOCK LEVEL FROM CORE (UO ARG) COMMAND AND SEGCON
2219
2220          INTERN PRTSEG
2221          EXTERN PCORSZ
2222
2223 002210 260140 002213' PRTSEG: PUSHJ PDP,SEGSIZ          ;TAC1=SIZE OF HIGH OR LOW SEG
2224 002211 200040 000002          MOVE TAC,TAC1          ;RADX10 WANT DEC. NO. IN TAC
2225 002212 254000 002207'          JRST RADX10          ;PRINT DECIMAL
2226
2227          ;ROUTINE TO RETURN SIZE OF HIGH OR LOW SEG
2228          ;CALL:  MOVE ITEM,LOW OR HIGH SEG NUMBER
2229          ;      PUSHJ PDP,SEGSIZ
2230          ;      RETURN WITH SIZE IN K IN TAC1
2231
2232          INTERN SEGSIZ
2233          EXTERN CPOPJ
2234
2235 002213          SEGSIZ:
2236          IFN FTSWAP,<
2237          EXTERN IMGIN
2238 002213 135100 000000          LOB TAC1,IMGIN          ;SIZE WHEN NEXT SWAPPED IN(IN K)
2239 002214 326100 001713'          JUMPN TAC1,CPDPJ          ;0 MEANS NOT SWAPPED OUT
2240          >
2241 002215 554104 000101'          HLRZ TAC1,JBTADR(ITEM) ;SIZE-1 LOW LOW OR HIGH SEG IN WORDS IN CORE
2242 002216 322100 002214'          JUMPE TAC1,CPDPJ          ;IS IT IN CORE?
2243 002217 242100 777766          LSH TAC1,-12          ;YES, CONVERT TO #K-1
2244 002220 344100 002216'          AOJA TAC1,CPDPJ          ;ADD 1 AND RETURN NUMBER OF K
  
```

```

2245 ;ROUTINE TO ASSIGN A MINIMAL CORE AREA(140 WORDS)
2246 ;CALLED FROM CORE,KJOB, AND RUN COMMANDS
2247 ;THIS ROUTINE PRESERVES INPUT BYTE POINTER IN TAC
2248 ;CALL: PUSHJ PDP,GETMIN
2249 ;
2250 ; RETURN PROG=0 IF UNSUCCESSFUL OR CORE ASSIGNED ON DISK
2251
2252 EXTERN JOBDA,TPOPJ
2253 002221 GETMIN:
2254 IFE FTTYSER,<
2255 PUSH PDP,TAC ;SAVE INPUT BYTE POINTER TO COMMAND
2256 >
2257 IFN FTTYSER,<
2258 002221 261140 000006 PUSH PDP,DEV DAT ;SAVE TTY DDB ADR
2259 >
2260 002222 261140 000002 PUSH PDP,TAC1 ;SAVE DEVICE NAME(GET)
2261 002223 261140 000000 PUSH PDP,IOS ;SAVE DISPATCH ADDRESS(ANYACT USES IOS)
2262 IFN FT2REL,<
2263 EXTERN KILHGH
2264 002224 260140 000401' PUSHJ PDP,KILHGH ;KILL HIGH SEG
2265 >
2266 002225 201040 000000 MOVEI TAC,JOBDA ;LENGTH OF JOBDA AREA
2267 002226 260140 000501' PUSHJ PDP,CORE0 ;ASSIGN 140 WORDS ON DISK OR MEMORY
2268 002227 255000 000000 JFCL ;IGNORE IF CANT(PROG=0)
2269 002230 262140 000000 POP PDP,IOS
2270 002231 262140 000002 POP PDP,TAC1 ;RESTORE PUSHED ACS
2271 IFN FTTYSER,<
2272 002232 262140 000006 POP PDP,DEV DAT ;RESTORE TTY DDB ADR
2273 002233 263140 000000 POPJ PDP, ;TAC NOT USER BY FULL DUPLEX SCNSER
2274 >
2275 IFE FTTYSER,<
2276 JRST TPOPJ ;RESTORE TAC AND RETURN
2277 >

```

```

2278 ;ROUTINE TO GET 1 WORD FROM USER ARE WHICH CAN BE IN LOW OR HIGH SEG
2279 ;CALL: MOVE PROG,CXWD PROT,RELOC FOR LOW SEG]
2280 ; MOVE ITEM,JOB NUMBER
2281 ; HRLI U00,PROG ;FOR RELOCATION
2282 ; HRR U00,USER ADDRESS(IE BEFORE RELOCATION)
2283 ; PUSHJ PDP,GETWRD
2284 ; ERROR RETURN ADDRESS OUT OF BOUNDS
2285 ; OK RETURN, CONTENTS IN AC TAC
2286 ;CAN BE CALLED AT CLOCK OR U00 LEVEL
2287 ;CALLED FROM E COMMAND,INIT,OPEN AND CALL U00S
2288
2289 INTERN GETWRD
2290 EXTERN CPOPJ1
2291
2292 002234 554040 000007 GETWRD: HLRZ TAC,PROG ;LARGEST REL LOC IN LOW SEG
2293 002235 305054 000000 CAIGE TAC,(U00) ;IS ADR, IN LOW SEG?
2294 IFN FT2REL,<
2295 EXTERN HGHWRD
2296 002236 254000 000000 JRST HGHWRD ;NO, CHECK IF IN HIGH SEG(ERROR RET IF NO)
2297 >
2298 IFE FT2REL,<
2299 POPJ PDP, ;NO, ERROR RETURN
2300 >
2301 002237 200060 000014 MOVE TAC,@U00 ;YES, GET IT FROM LOW SEG
2302 002240 254000 002111 JRST CPOPJ1 ;AND SKIP RETURN
  
```



```

2355 ;ROUTINE TO SCAN COMMAND STRING ARGUMENTS FOR SAVE,GET,RUN AND R
2356 ;COMMANDS AND STORE THEM IN JOB DATA AREA WHICH MUST BE IN CORE
2357 ;WHEN SGSET IS CALLED FROM COMMAND DECODER
2358 ;CALL: MOVE TAC,INPUT BYTE POINTER
2359 ; MOVE TAC1,SIXBIT DEVICE NAME
2360 ; MOVE DAT,OUTPUT BYTE POINTER
2361 ; MOVE IOS,ADR. OF MONITOR JOB(SAVJOB,GETJOB,RUNJOB)
2362 ; MOVE PROG, ADR. OF JOB AREA
2363 ; PUSHJ PDP,SGSET
2364
2365 000012 C=BUFPNT
2366
2367 INTERNAL FTLOGIN,FT2REL,FTDISK
2368 EXTERNAL JOBCOR,JBTPRG
2369
2370 002241 322100 002034' SGSET: JUMPE TAC1,NOTENF ;NOT ENOUGH ARG IF NO DEVICE NAME
2371 002242 202107 000011 MOVEM TAC1,SGADEV(PROG) ;STORE DEVICE NAME
2372 002243 336104 000626' SKIPN TAC1,JBTPRG(ITEM) ;GET AUTOMATIC FILENAME, OR
2373 002244 260140 001723' PUSHJ PDP, CTEXT1 ;GET FILE NAME FROM COMMAND STRING
2374 002245 322100 002034' JUMPE TAC1,NOTENF ;THERE MUST BE A FILE NAME
2375 002246 202107 000000 MOVEM TAC1,SGANAM(PROG) ;STORE FILE NAME
2376 002247 202104 002243' MOVEM TAC1,JBTPRG(ITEM) ;SAVE FILE NAME FOR SYSTAT COMMAND
2377 002250 201100 000000 MOVEI TAC1,0 ;ASSUME USER DID NOT SPECIFY AN EXTENSION
2378 ; 0 WILL BE TURNED INTO SAV OR DMP
2379
2380 002251 306400 000056 IFE FTTTYSER,< LOB TEM,TAC>
2381 002252 260140 001721' CAIN TEM,"" ;IS AN EXTENSION SPECIFIED?
2382 002253 512107 000001 PUSHJ PDP,CTEXT ;YES, GET EXTENSION
2383 ; HLLZM TAC1,SGAEXT(PROG) ;STORE IT FOR LOOKUP
2384
2385 IFN FT2REL,<
2386 002254 260140 000000 EXTERN SETEXT ;ALSO SAVE IT AGAIN IN SGALOW FOR LOW SEG
2387 ; LOOKUP OR ENTER
2388 ; SET HIGH EXTENSION(SGAHGH) TO ,SHR IF SSAVE OR GET
2389 ; ,HGH IF SAVE(LH IOS=NSRBIT),
2390 >
2391 002255 402007 000002 SETZM SGADAT(PROG) ;SET DATE(E+2) TO 0, SO MONITOR WILL USE TODAY'S
2392
2393 IFN FTLOGIN,<
2394 002256 260140 002117' PUSHJ PDP,PJPGNO ;GET PROJ, PROG, NO.
2395 002257 202707 000007 MOVEM AC2,SGAPPN(PROG) ;STORE 0 IF NO []'S TYPED BY USER
2396 >
2397 002260 260140 002071' PUSHJ PDP,DECIN1 ;AMOUNT OF CORE (OPTIONAL THIRD ARG.)
2398 002261 254000 002265' JRST SGSET1 ;DOES NOT RETURN IF ERROR, RETURN HERE IF NO ARG.
2399 002262 254000 002000' JRST COMERA ;ILLEGAL CHARACTER
2400 002263 242100 000012 LSH TAC1,12 ;CONVERT TO HIGHEST REL, LOC.
2401 002264 275100 000001 SUBI TAC1,1
2402 SGSET1: MOVEM TAC1,SGANAM(PROG) ;STORE FOR RUN AND SAVE
2403 002266 550100 000000 HRRZ TAC1,IOS ;SCHEDULE MONITOR JOB
2404 ; GUARRANTEE LH OF PC WORD IS 0, SINCE IT WILL
2405 ; BE ADDED TO STARTING ADDRESS(IF RUN COM)
2406 ; START JOB WITH PC IN MONITOR MODE

```

```

2404 ;ROUTINE TO PICKUP ARGUMENTS FOR RUN AND GETSEG UUOS
2405 ;THIS ROUTINE DOES SAME THING AS SGSET, EXCEPT THAT ARGUMENTS ARE
2406 ;OBTAINED FROM USER UUO ARGUMENTS INSTEAD OF FROM CONSOLE COMMAND
2407 ;THE USERS ARG ARE MOVED TO USER ACS(SGA,...), THEREBY CLOBBERING HIS ACSS
2408 ;USER AC FIELD AND START PC OFFSET(RUN UUO) ARE SAVED ON PD LIST AT JOBPD3
2409 ;THEN LOWER CORE IS SET UP(SG2 CALLED) RESET IS NOT DONE (FOR GETSEGUUO)
2410 ;JBTPRG NOT SET FOR LOW SEG, SINCE GETSEGUUO SHOULD NOT
2411 ;CALL: MOVE TAC,CONTENTS OF USER AC(ADR, OF 3 WORD ARG LIST)
2412 ; MOVE PROG,JOB RELOCATION
2413 ; PUSHJ PDP,GETARG
2414 ; RETURN
2415
2416 INTERN GETARG
2417 EXTERN JBTPRG,JOBCOR,PUUOAC
2418
2419 002270 540600 000001 GETARG: HRR UUO,TAC ;MOVE ADR, OF ARG LIST TO UUO
2420 002271 250043 000000 EXCH TAC,(PDP) ;AND PUT ON PD LIST
2421 002272 261140 000001 PUSH PDP,TAC ;MOVE RETURN PC UP ONE IN PD LIST
2422 002273 135040 001611 LDB TAC,PUUOAC ;USER AC FIELD IN RUN OR GETSEG UUO

2423 002274 540043 777777 HRRM TAC,-1(PDP) ;SAVE IN CASE OF ERROR RETURN
2424 002275 260140 000000 PUSHJ PDP,GETWOU ;GET FIRST ARG FROM USER AREA
2425 002276 202047 000011 MOVEM TAC,SGADEV(PROG) ;STORE DEVICE NAME
2426 002277 260140 000000 PUSHJ PDP,GETWD1 ;GET NEXT ARG FROM USER AREA
2427 002300 202047 000000 MOVEM TAC,SGANAM(PROG) ;STORE FILE NAME FOR LOOKUP (DO NOT STORE FOR LOWSEG)
2428 002301 260140 002277 PUSHJ PDP,GETWD1 ;GET THIRD ARG(EXTENSION WORD E+1)
2429 002302 200100 000001 MOVE TAC1,TAC ;PUT ARG IN TAC1, SO SAME AS SGSET RETURN FROM CTEXT
2430 002303 202107 000001 MOVEM TAC1,SGAEXT(PROG) ;STORE EXTENSION AND RH FROM USER
2431 IFN FT2REL,<
2432 EXTERN SETEX1
2433 002304 260140 000000 PUSHJ PDP,SETEX1 ;SAVE EXT AGAIN IN SGALOW
2434 ; SETUP EXT FOR HIGH SEG(SGAHGH="SHR")
2435 ; SETUP EXTENSION FOR LOW SEG(SGALOW="SAV")
2436 >
2437 002305 260140 002301 PUSHJ PDP,GETWD1 ;GET FOURTH ARG(DATE WORD)
2438 002306 202047 000002 MOVEM TAC,SGADAT(PROG)
2439 002307 260140 002305 PUSHJ PDP,GETWD1 ;GET FIFTH USER ARG FROM USER AREA
2440 002310 202047 000007 MOVEM TAC,SGAPPN(PROG) ;STORE PROJECT,PROGRAMMER NO. OR Ø
2441 002311 260140 002307 PUSHJ PDP,GETWD1 ;SIXTH ARG FROM USER
2442 002312 552047 000015 HRRZM TAC,SGANEW(PROG) ;STORE CORE ARG OR Ø(HIGHEST LOC DESIRED)
2443 ; IGNORE LH
2444 002313 254000 002440 JRST SG2A ;GO SET UP LOWER CORE AND RETURN
2445 ; DO NOT DO A RESET
    
```

```

2446 ;THIS JOB SAVES A JOB AREA ON RETRIEVABLE STORAGE
2447 ;THIS JOB RUNS IN EXEC MODE AND CALLS IO ROUTINES USING REGULAR UUOS
2448 ;NO ATTEMPT IS MADE TO SAVE STATUS OF IO DEVICES, JOBPDP, OR AC'S
2449 ;IN FACT THE ONLY USEFUL THING WHICH MAY BE DONE WITH A JOB AREA
2450 ;AFTER IT HAS BEEN SAVED IS TO START EXECUTION OVER AT THE STARTING
2451 ;ADDRESS
2452
2453 INTERNAL SAVJOB,SAVERR
2454 EXTERNAL JOB41,JOBS41,JORDDT,JOBSDD,JOBSV
2455
2456
2457 002314 265100 002436' SAVJOB: JSP TAC1,SG1 ;SET UP ACS PROG,PDP,JDAT,ITEM,
2458 ; RESET DEVICES
2459 002315 574107 000013 HLRE TAC1,SGADMP(PROG) ;-NO. OF WORDS TO WRITE
2460 002316 260140 002753' PUSHJ PDP,CKIOWD ;CHECK USER'S CORE ARG(IF ANY) WITH AMOUNT TO WRITE
2461 ; RETURN ONLY IF 0 OR NOT SMALLER
2462 002317 542047 000000 HRRM TAC,JOBCOR(PROG) ;STORE MAX OF SIZE OF FILE OR CORE ARG
2463 ; FOR ASSIGNING INITIAL CORE WHEN FILE GOTTEN
2464
2465 IFN FT2REL,<
2466 002320 260140 000000 EXTERN SAVHGH
2467 002321 254000 002374' PUSHJ PDP,SAVHGH ;INIT DEV,SAVE HIGH SEG, IF ANY, RETURN IF OK
2468 JRST SAVFIN ;HIGH SAVED, BUT NO DATA IN LOW SEG, SO NO
2469 ; NOT WRITE LOW FILE
2470 ; SKIP RETURN IF LOW SEG TO BE WRITTEN
2471 >
2472 002322 050000 000010 SAVJB1: OPEN 0,SGAMOD ; SGALEN, AND SGAEXT RESTORED
2473 ;RE INIT DEVICE, SO UGETF WILL SET FIRST FREE
2474 002323 254000 002473' JRST SGERRA ; BLOCK BECAUSE NO LOOKUP OR ENTER DONE
2475 002324 073000 000012 UGETF 0,SGAHED ;DEVICE NOT AVAILABLE
2476 ;GET FIRST FREE BLOCK(MEANINGFUL ONLY IF DTA)
2477 ; CAUSE ENTER TO ASSIGN FIRST LOBCK OF FILE
2478 ; AS LOWEST FREE BLOCK SO TENDMP CAN READ
2479 002325 077000 000000 ENTER 0,SGANAM ; SAVED FILES, SGAHED IS NO LONGER NEEDED(OPEN UUO)
2480 002326 254000 002400' JRST SAVFRR ;ENTER FILE NAME IN DIRECTORY
2481 002327 200047 000000 MOVE TAC,JOB41(JDAT) ;DIRECTORY FULL OR PROTECTION FAILURE
2482 002330 202047 000000 MOVEM TAC,JOBS41(JDAT) ;SAVE USER UUO HANDLING JSR
;IN UPPER PART OF JOB DATA AREA

```

2483	002331	200047	000446'	MOVE TAC,JOBDDT(JDAT)	ISAVE DDT STARTING ADDRESS HIGHER UP IN JOB DATA AREA
2484	002332	202047	000000	MOVEM TAC,JOBSDD(JDAT)	ISO COMPRESS ALWAYS MOVES CODE DOWN
2485	002333	563000	001606'	HRROS USRHCU	I FLAG THAT SAVE GET IS UNDER WAY
2486					I SO THAT JOBHRL WILL NOT BE MODIFIED BY SETHGH ROUTINE
2487					
2488					I TO USER'S HIGH SEG RELOCATION INFO, SINCE
2489					I IT WILL CONTAIN COMPRESSION IOWD,
2490					I COMPRESSION WILL ALWAYS MOVE DOWN
2491	002334	550407	000443'	HRRZ TEM,JOBSA(JDAT)	ISAV START ADDRESS FOR IOWDMP
2492	002335	201047	000000	MOVEI TAC,JOBSV(PROG)	IPOINT TO 1ST DATA WORD
2493	002336	200100	003066'	MOVE TAC1,[XWD PROG,JOBSDD]	IT STARTS AT JOBSDD
2494	002337	574207	000013	HLRE ITEM,SGADMP(PROG)	I IOWD FOR THIS SIZE CORE(-LENGTH TO WRITE)
2495	002340	213000	000004	MOVNS ITEM	I POSITIVE LENGTH
2496	002341	271200	000000	ADDI ITEM,JOBSVM	I ADD IN FIRST LOC-1 TO WRITE=HIGHEST LOC TO WRITE
2497					I TO MAKE END TEST
2498	002342	505200	000007	HRLI ITEM,PROG	I USE PROG FOR RELOCATION
2499	002343	202040	000005	CMPLP1: MOVEM TAC,DAT	ISAVE 1ST LOC FOR IOWD
2500	002344	313100	000004	CAMLE TAC1,ITEM	ISEARCH FOR 1ST NON-0 WORD
2501	002345	344040	002362'	AOJA TAC,CMPTHR	ITHROUGH
2502	002346	336020	000002	SKIPN @TAC1	IThis A DATA WORD?
2503	002347	344100	002344'	AOJA TAC1,-3	INO, KEEP LOOKING
2504	002350	211640	000001	MOVNI AC1,1	IYES, AC1 WILL BE AN IOWD
2505	002351	505642	777777	HRLI AC1,-1(TAC1)	I1ST LOCATION - 1
2506	002352	261060	000002	CMPLP2: PUSH TAC,@TAC1	ISAVE A DATA WORD
2507	002353	350000	000002	AOS TAC1	
2508	002354	315100	000004	CAMGE TAC1,ITEM	IAT TOP?
2509	002355	336020	000002	SKIPN @TAC1	INO, NEXT WORD NON-0?
2510	002356	254000	002360'	JRST .+2	INO, THROUGH THIS BLOCK
2511	002357	364640	002352'	SOJA AC1,CMPLP2	ICOUNT THE WORD AND CHECK NEXT
2512	002360	206645	000000	MOVSM AC1,(DAT)	ISAVE IOWD IN FRONT OF BLOCK
2513	002361	344040	002343'	AOJA TAC,CMPLP1	ILOOK FOR NEXT NON-0 BLOCK
2514	002362	505400	254000	CMPTHR: HRLI TEM,254000	ISET A JRST C(JOBSA)
2515	002363	202401	777777	MOVEM TEM,-1(TAC)	IAT END OF FILE
2516	002364	275047	002335'	SUBI TAC,JOBSV(JDAT)	ICOMPUTE WORD COUNT
2517	002365	213000	000001	MOVNS TAC	IMAKE AN IOWD
2518	002366	504047	000013	HRL TAC,SGADMP(PROG)	ISTART ADDRESS
2519	002367	206047	000003	MOVSM TAC,SGALEN(PROG)	I IOWD FOR THE OUTPUT UUO
2520	002370	201040	777776	MOVEI TAC,-2	I FLAG THAT CORE HAS BEEN COMPRESSED
2521	002371	506040	002333'	HRLM TAC,USRHCU	IKEEP LH NEG, COMMAND DECODER WILL EXPAN
2522					I CORE ON START ,DDT,SAVE, REENTER,SSAVE IN CASE
2523					I THIS SAVE IO DOES NOT GO TO COMPLETION, (CONTROL C
2524					I OR DEVICE FULL, SO THAT CORE DOES NOT GET EXPANDED

COMCON - COMMAND DECODER AND SAVEGET ROUTINES - V433 . MACRO,V36 19102 4-JUN-69 PAGE 90
 SAVGET - SAVE,GET,R,RUN COMMANDS AND RUN,GETSEG UUOS

2525	002372	260140	002560'	PUSHJ PDP,SGDOA	;DO OUTPUT,RELEASE,FIND TTY
2526	002373	067000	000003	OUTPUT 0,SGALEN	;OUTPUT UUO EXECUTED BY SGDO
2527					; RETURN HERE ONLY IF NO ERRORS
2528	002374	260140	002542'	SAVFIN: PUSHJ PDP,SGREL	;RELEASE DEVICE AND FIND TTY
2529	002375	265040	000000	JSP TAC,PHOLD	;PRINT MESSAGE AND STOP JOB
2530	002376	452370	220246	ASCIZ /JOB SAVED/	
	002377	406550	542000		
2531					
2532	002400	201040	000002	SAVERR: MOVEI TAC,PRERR	;ERROR CODE IN CASE RUN UUO(PROTECTION ERROR)
2533					; CHANGE TO DISK ERROR CODE IF DEV IS DSK
2534	002401	260140	003025'	PUSHJ PDP,SGRELL	;CHANGE TO DISK ENTER ERROR CODE IF DSK
2535					; RELEASE DEVICE AND RETURN TO USER(IF RUN UUO)
2536					; OR FIND TTY+PRINT ?CRLF
2537	002402	265040	002375'	JSP TAC,PHOLD	;PRINT MESSAGE AND STOP JOB
2538	002403	422232	242606	ASCIZ /DIRECTORY FULL/	
	002404	522372	254500		
	002405	432531	446000		

```

2539
2540
2541      ;THIS JOB GETS A JOB AREA FROM A RETRIEVABLE DEVICE
2542      ;THIS JOB RUNS IN EXEC, MODE AND CALLS IO ROUTINES DIRECTLY
2543      ;NO ATTEMPT IS MADE TO RESTORE STATUS OF IO DEVICES, PC, OR AC'S
2544      ;JOBPC IS SET TO STARTING ADDRESS OF JOB
2545      ;CORE MUST ALREADY HAVE BEEN ASSIGNED AND THE FOLLOWING LOC, SETUP IN
2546      ;JOB DATA AREA:
2547      ;JOBPOP, JOBRCL
2548
2549      INTERNAL GETJOB
2550      002406 265100 002436' GETJOB: JSP TAC1,SG1      ;SETUP ACS, SETUP LOWER CORE(SGALN,SGADMP)
2551      002407 260140 002512'      PUSHJ PDP,GETJB      ;GET THE JOB
2552      002410 265040 002402'      JSP TAC,PHOLD      ;RETURN ONLY IF EVERYTING OK
2553      002411 452370 220246      ASCIZ /JOB SETUP/
2554      002412 426512 550000
2555
2556      ;THIS JOB GETS A JOB AREA FROM A RETRIEVAL DEVICE AND STARTS IT UP
2557
2558      ;JOB HAS JUST A JOB DATA AREA ASSIGNED WHEN CONTROL GETS HERE
2559      ;THIS MONITOR JOB GETS A JOB AREA FROM A RETREIVABLE DEVICE
2560      ;ASSIGNS CORE AND START
2561
2562      INTERNAL RUNJOB
2563      002413 265100 002436' RUNJOB: JSP TAC1,SG1      ;SETUP ACS, SETUP LOWER CORE(SGALN,SGADMP)
2564      002414 254000 002422'      JRST URUN1      ;LH PF PC WORD=0(MSTART) SO LH OF TAC1=0 AFTER JSP
2565      ; PUT TAC1 AS FIRST ITEM ON PD LIST(JOBPD1)
2566      ; LH USED BY SGRELE ON ERROR TO SEE IF FROM USER
2567      ; AND LH ADDED TO START PC(JOBSA) BY URUN

```

```

2568 ;RUN UUD
2569 ;CALL: MOVE AC,[XWD N,D]
2570 ; CALL AC,[SIXBIT /RUN/]
2571 ; ERROR RETURN ;UNLESS LH=HALT(PRINT CONSOLE MESS, IF YES)
2572 ; IK OK, TRANSFER TO C(JOBSA)+N FOR NEW PROGRAM
2573 ; USERS ACS CLOBBERED SO CANNOT PASS DATA TO NEW PROGRAM
2574
2575 ;WHERE: D/ DEVICE NAME
2576 ; D+1/ FILE NAME
2577 ; D+2/ FILE EXT OR # (LH SIXBIT)
2578 ; D+3/ DATE ETC
2579 ; D+4/ PROJECT,PROGRAMMER NO OR 0(CURRENT UFD OR DTA,MTA)
2580 ; D+5/ HIGHEST LOC DESIRED(OR 0) ANALOGOUS TO RUN COMMAND
2581 ; LH IS IGNORED(RATHER THAN ASSIGNING CORE TO HIGH SEG)
2582
2583 INTERN URUN
2584 EXTERN JOBSA
2585
2586 002415 260140 000375' URUN: PUSHJ PDP,RESET ;RELEASE DEVICES
2587 ; WARNING! THIS GOES VERY DEEP IN
2588 ; PUSHDOWN, SEE MCO 518
2589 ; (AC UUD PRESERVED IN RESET)
2590 002416 200060 000014 MOVE TAC,@UUD ;RESTORE CONTENTS OF USER'S CALLING AC
2591 002417 260140 002270' PUSHJ PDP,GETARG ;GET 6 ARGS FROM USER AND STORE
2592 ; SAVE STARTING ADDRESS INCREMENT(LH OF TAC)
2593 ; AND USER AC NUMBER(IN CASE OF ERROR RETURN)
2594 ; SETUP ACS,DEV DAT TO DDB
2595 002420 200047 000000 MOVE TAC,SGANAM(PROG) ;STORE FILE NAME FOR LOW SEG
2596 002421 202044 002247' MOVEM TAC,JBTPRG(ITEM) ;FOR SYSTAT
2597 002422 260140 002512' URUN1: PUSHJ PDP,GETJB ;GET BOTH LOW AND HIGH SEGMENTS
2598 002423 554123 000000 HLRZ TAC1,(PDP) ;GET STARTING ADDRESS INCREMENT(0 IF RUN COM)
2599
2600 IFN FT2REL,<
2601 EXTERN CHKMED
2602 002424 303120 200001 CAILE TAC1,1 ;IS START INCREMENT 2 OR 1?
2603 002425 260140 001522' PUSHJ PDP,CHKMED ;NO, CHECK TO SEE IF THIS IS SHARABLE SEG
2604 ; AND IF YES, SET MEDDLE BIT FOR THIS USER
2605 ; SO UWP CANNOT BE TURNED OFF AND CORE FOR HIGH SEG
2606 ; CANNOT BE INCREASED OR DECREASED (TAC1 PRESERVED).
2607
2608 002426 273127 002334' > ADDB TAC1,JOBSA(JDAT) ;ADD STARTING ADDRESS TO BOTH
2609 ; SO THAT <CONTROL>C START WILL START
2610 ; PROGRAM AT SAME STARTING ADDRESS
2611 002427 505120 010000 HRLI TAC1,USRMOD ;SET USER MODE BIT IN PC
2612 002430 261140 000002 PUSH PDP,TAC1 ;PUT ON PD LIST
2613 002431 254000 000000 JRST USRXIT ;AND GO TO RETURN TO USER AS IF FROM UUD

```

COMCON - COMMAND DECODER AND SAVEGET ROUTINES - V433 MACRO,V36 19:02 4-JUN-69 PAGE 93
SAVGET - SAVE,GET,R,RUN COMMANDS AND RUN,GETSEG UUOS

```
2614 ;UUO TO GET JUST HIGH SEG AND RETURN TO USER
2615 ;CALL IS THE SAME AS FOR RUN UUO EXCEPT THAT OK RETURN IS SKIP RETURN
2616 ;IF ERROR RETURN HAS HALT IN LH, STANDARD CONSOLE MESSAGE IS PRINTED AND JOB STOPPED
2617
2618 INTERN UGTSEG,UGTERR
2619 002432 UGTSEG:
2620
2621 IFN FT2REL,<
2622 EXTERN UGETHI
2623 002432 254000 000000 JRST UGETHI ;IN SEGCON
2624 >
2625 002433 201040 000013 UGTERR: MOVEI TAC,ILUERR ;ILLEGAL UUO ERROR CODE
2626 002434 260140 003030 PUSHJ PDP,SGRELE ;SEE IF USER WANTS ERROR
2627 002435 254000 000000 JRST UUGERR ;NO, PRINT ILLEGAL UUO
```

```

2628 ;ROUTINE TO SETUP ACS, RESET IO, AND SETUP LOWER CORE LOCATIONS
2629 ;FOR SAVE AND GET(SGALEN SET TO IOWD OR PP IF DTA OR DSK)
2630 ;SGADMP SET TO IOWD FOR THIS SIZE CORE
2631 ;CALL: JSP TAC1,SG1
2632 ;
2633 ; ALWAYS RETURN HERE, UNLESS DEVICE NOT FOUND
2634 ; DEVDAT SETUP TO DEVICE DATA BLOCK(BUT NOT INITED, UNLESS FT2REL=? )
2635 ; SO THAT INIT NOT NECESSARY IF SEG ALREADY KNOWN(NO DTA QUEUEING)
2636 ; DEVICE CHARACTERISTICS WORD(DEVMOD) RETURNED IN AC TAC1
2637 ; IF DISK SYSTEM
2638
2639 INTERN SG2,SG3,SG4 ;CALLED FROM SEGCON
2640 EXTERN USRDDT,USRREL,JOBSAV,JOBBF,CPOPJ1
2641 EXTERN TTYFNU,JOBSVM,IADCK,SAVDMP
2642 002436 265040 001605' SG1: JSP TAC,MONSTR ;SETUP PROG,PDP,JDAT,ITEM=JOB NUMBER
2643 ; PUT TAC1 ON END OF PD LIST(EXEC MODE PC,
2644 ; SO ERROR MESSAGES WILL ALWAYS PRINT(SEE SGRELE)
2645
2646 002437 260140 002415' SG2: PUSHJ PDP,RESET ;RELEASE ALL DEVICES
2647 002440 201040 000016 SG2A: MOVE! TAC,DR ;DUMP MODE 16(DUMP BY RECORDS
2648 ; IN CASE THIS IS MAGTAPE)
2649
2650 002441 202047 000010 MOVEM TAC,SGAMOD(PROG) ;STORE FOR OPEN UO
2651 002442 402007 000004 SETZM SGAEND(PROG) ;END OF DUMPE MODE COMMAND LIST
2652 002443 402007 000005 SETZM SGAREN+1(PROG) ;FOR DELETE
2653 002444 402007 000006 SETZM SGAREN+2(PROG) ;FOR DELETE
2654 002445 513007 002317' HLLZS JOBCOR(JDAT) ;THIRD ARG IN JOBDATA AREA(SO CHKARG WILL
2655 ; WORK ON FIRST CALL(SAVE AND GET)
2656 002446 402007 000012 SETZM SGAHED(PROG) ;CLEAR BUFFER HEADER ARG. FOR OPEN UO
2657
2658 IFE FT2REL,<
2659 OPEN 0,SGAMOD ;TRY TO ASSIGN DEVICE, SINCE IT MUST BE DONE
2660 >
2661 IFN FT2REL,<
2662 002447 200047 000011 MOVE TAC,SGADEV(PROG) ;PHYSICAL OR LOGICAL DEVICE NAME
2663 002450 260140 001600' PUSHJ PDP,DEVSRV ;FIND DEVICE AND SETUP DEVDAT TO DDB
2664 ; DO NOT INIT DEV SINCE IO MAY NOT BE NECESSARY
2665 ; DO NOT WANT TO WAIT IN DTA SYSTEM
2666 ; TAPE QUEUE IN 10/40 SYS
2667 >
2668 002451 254000 002473' JRST SGERRA ;NOT AVAILABLE
;DEVICE INITED(OR FOUND)

```

```

2669 ;COMMON EXIT FROM SAVHGH AND GETHGH ROUTINES(HIGH SEG SAVE AND GET)
2670 ;SO THAT SGA... LOCATIONS ARE RESTORED TO ORIGINAL VALUES FOR LOW SEG
2671
2672 002452 200047 000000 SG3: MOVE TAC,JOBF(JDAT) ;FIRST FREE LOC IN JOB(SET FROM LH OF
2673 ; JORSA WHICH IS SET BY LOADER
2674 002453 201041 777777 MOVEI TAC,-1(TAC) ;MAKE LAST LOC TO SAVE OR GET(MAKE 0=777777)
2675 002454 330000 000000 SKIPN USRDDT ;USER DDT IN USE(IF YES, SAVE ALL OF CORE
2676 ; SO HIS SYMBOLES WILL BE INCLUDED
2677 002455 260140 000000 PUSHJ PDP,IADRCK ;NO, ADDRESS TOO SMALL OR TOO LARGE?
2678 002456 200040 000000 MOVE TAC,USRREL ;YES, DUMP ALL OF CORE RATHER THAN GIVE
2679 ; ADDRESS CHECK MESSAGE-HIGHEST REL.ADR,
2680 002457 213000 000001 MOVNS TAC ; ADDRESS CHECK MESSAGE-HIGHEST REL.ADR,
2681 002460 271040 002341 ADDI TAC,JOBSVM ;-HIGHEST ADR TO SAVE OR GET
2682 002461 505040 002460 HRLI TAC,JOBSVM ;LOWER CORE NOT DUMPED
2683 002462 206047 000013 MOVSM TAC,SGADMP(PROG) ;IE FIRST LOC-1 TO BE DUMPED
2684 002463 ;STORE IOWD WORD OF THIS SIZE CORE
2685 SG4:
2686 002463 200106 000004 IFN FTDISK,<
2687 002464 603100 200000 MOVE TAC1,DEVMOD(DEVSTAT) ;RETURN DEVICE CHARACTERISTICS(IF DISK SYS)
;IS THIS DEVICE A DISK?
2688 002465 204047 000007 MOVNS TAC,SGAPPN(PROG) ;YES, MAKE SURE FOURTH WORD IS PROJ,PROG NO.
2689 >
2690 002466 206047 000003 MOVSM TAC,SGALEN(PROG) ;NO, MAKE SURE FOURTH WORD IS IOWD FOR DECTAPE
2691 ; SINCE DECTAPE USES RH TO COMPUTE LENGTH IN K
2692 ; FOR BOTH SAVE AND GET
2693 002467 336047 000001 SKIPN TAC,SGAEXT(PROG) ;DID USER SPECIFY AN EXTENSION ?
2694 002470 205040 000000 MOVSI TAC,SAVDMP ;NO, USE ,SAV OR ,DMP
2695 002471 202047 000001 MOVEM TAC,SGAEXT(PROG) ;AND STORE FOR LOOK UP OR ENTER
2696 002472 263140 000000 POPJ PDP,
2697
2698 ;ERROR ON INIT OR DEVICE SEARCH
2699
2700 INTERN SGERRA ;CALLED FROM SEGCON
2701
2702
2703 002473 322300 002504 SGERRA: JUMPE DEVDAT,SGERR1 ;WAS DEVICE FOUND, BUT JUST UNAVAILABLE?
2704 002474 202303 000000 MOVEM DEVDAT,(PDP) ;YES, SAVE DDB ADDRESS FOR MESSAGE(ERNAM)
2705 002475 201040 000011 MOVEI TAC,0NAERR ;ERROR CODE IN CASE RUN UUD(DEVICE NOT AVAILABLE)
2706 002476 260140 003030 PUSHJ PDP,SGRELE ;RETURN TO USER IF RUN UUD
2707 ; OR FIND TTY AND PRINT ?CRLF
2708 002477 260140 000000 PUSHJ PDP,ERNAM ;PRINT DEVICE NAME USING (PDP)
2709 002500 265040 002410 JSP TAC,PHOLD ;START TTY AND STOP JOB
2710 002501 202351 750100 ASCIZ /NOT AVAILABLE/
2711 002502 406550 144630
2712 002503 406051 442400
2713
2714 002504 201040 000012 SGERR1: MOVEI TAC,NSDERR ;ERROR CODE IN CASE RUN UUD(NO SUCH DEVICE)
2715 002505 260140 003030 PUSHJ PDP,SGRELE ;RETURN TO USER IF RUN UUD
2716 ; OR FIND TTY AND PRINT ?CRLF
2717 002506 265040 002500 JSP TAC,PHOLD ;START TTY AND STOP JOB
2718 002507 472364 051652
2719 002510 416204 042212
2720 002511 532230 342400

```

```

2717 ;ROUTINE TO GET FILE FROM DEVICE(LOW AND/OR HIGH)
2718 ;CALL: ACS JDAT,PROG,PDP,DEV DAT SETUP
2719 ; MOVE ITEM, JOB NUMBER
2720 ; IFN FTDISK,<MOVE TAC1,DEV MOD(DEV DAT) ;DEVICE CHAR.>
2721 ; PUSHJ PDP,GETJB
2722 ; RETURN ONLY IF EVERYTHING OK
2723
2724
2725 EXTERNAL JOBCOR, JOB, CPOPJ, JOBS41, JOB41
2726
2727 002512 GETJB!
2728 IFN FT2REL,<
2729 EXTERN GETHGH
2730 002512 260140 000000 PUSHJ PDP,GETHGH ;SEE IF HIGH SEG ALREADY EXISTS AND BEING SHARED
2731 ; IF NOT, TRY TO LOOKUP AND READ IN HIGH FILE
2732 ; IF ,SHR DOESN'T EXIST, TRY ,HGH, IF NEITHER-SKIP RETU
2733 RN
2734 ; TAC1=DEV MOD(DEV DAT) IF DISK(DEV CHAR,)
2735 002513 254000 002533' JRST LOWFIN ;HIGH SEG NOW IN CORE AND NO LOW FILE NEEDED
2736 ; LOW FILE NEEDED
2737 ; EITHER BECAUSE NOHIGH SEG
2738 ; ORHIGH SEG ALSO NEEDS LOW FILE
2739 >
2740 002514 076000 000000 LOOKUP 0,SGANAM ;LOOKUP LOW SEG FILE(EXT=SAV,DMP OR LOW(IF HIGH SEG
2741 ; REQUIRES LOW SEG AND USER DID NOT TYPE EXT)
2742 ; MODE=SAV MOD SO DECTAPE SERVICE WILL RETURN
2743 ; IOWD IN E+3(TEMPORARY)
2744 002515 254000 003012' JRST NOFILE ;GO PRINT FILE,EXT NOT FOUND
2745 002516 574107 000003 HLR TAC1,SGALEN(PROG) ;-NO. OF WORDS IN FILE
2746 002517 260140 002753' PUSHJ PDP,CKIOWD ;CHECK USER'S SUPPLIED CORE ARG TO MAKE SURE NOT
2747 ; TOO SMALL, RETURN LARGER OF FILE SIZE OR CORE ARG
2748 002520 260140 002776' PUSHJ PDP,GETCOR ;OK, TRY TO GET CORE
2749 002521 200106 000004 MOVE TAC1,DEV MOD(DEV DAT) ;DEVICE CHARACTERISTICS
2750 002522 213000 000001 MOVNS TAC
2751 002523 271040 002461' ADDI TAC,JOBSVM ;-WORD COUNT FOR ALL USERS CORE
2752 002524 603100 000020 TLN TAC1,DVMTA ;MAG TAPE?
2753 002525 506047 000003 HRLM TAC,SGALEN(PROG) ;YES, USE USER-SPECIFIED CORE ARGUMENT
2754 002526 553007 000000 HRRZS JOBP01(JDAT) ;TURN OFF USER MODE PC FLAG IN CASE THIS
2755 ; IS A RUN UO,SO ERRORS WILL NOT TRY TO RETURN TO USER
2756
2757 002527 260140 002557' PUSHJ PDP,SGDO ;READ IN FILE INTO LOW SEGMENT
2758 002530 066000 002003 INPUT 0,SGALEN ;EXECUTED FROM SGDO
2759 002531 200047 002330' MOVE TAC,JOBS41(JDAT) ;RESTORE USER UO JSR LOC
2760 002532 202047 002327' MOVEM TAC,JOBS41(JDAT) ;SAVED BY SAVE
2761 002533 550047 002445' LOWFIN: HRRZ TAC,JOBCOR(JDAT) ;CORE ARG FROM PREVIOUS SAVE(THIS MONITOR
2762 ; ALWAYS STORES SOMETHING)
2763 002534 336000 000001 SKIPN TAC ;IS THIS AN OLD FORMAT FILE WITH NO CORE ARG TO SAVE?
2764 002535 200040 002456' MOVE TAC,USRREL ;YES, USE ASSIGNMENT MADE WHEN LOW FILE READ IN
2765 002536 260140 002756' PUSHJ PDP,CKSARG ;RETURN ONLY IF USER'S SUPPLIED ARG IS 2 OR NOT
2766 ; SMALLER THAN SAVE CORE ARG, RETURN LARGER
2767 002537 260140 002776' PUSHJ PDP,GETCOR ;TRY TO GET THIS AMOUNT OF CORE
2768 002540 200040 002535' MOVE TAC,USRREL ;HIGHEST LOC ASSIGNED TO LOW SEG
2769 002541 542047 002533' HRRM TAC,JOBCOR(JDAT) ;SET INITIAL CORE ASSIGNMENT IN JOB DATA AREA FOR

```

COMCON - COMMAND DECODER AND SAVEGET ROUTINES - V433 - MACRO,V36 19:02 4-JUN-69 PAGE 96-1
SAVEGET - SAVE,GET,R,RUN COMMANDS AND RUN,GETSEG UOPS

2770
2771
2772

; USER TO USE TO RESET CORE TO INITIAL SETTING WHEN
; PROGRAM IS RESTARTED
; FALL INTO SGREL

2826	002573	201040	000006	MOVEI TAC,TRNERR	IYES, ERROR CODE IN CASE THIS IS RUN UUO
2827					I (TRANSMISSION ERROR)
2828	002574	260140	003030'	PUSHJ PDP,SGRELE	IRELEASE DEVICE AND ERROR RETURN TO USER IF RUN UUO
2829					I OR FIND TTY AND PRINT ?CRLF
2830	002575	265040	002506'	JSP TAC,PHOLD	ISTART TTY AND STOP JOB
2831	002576	522450	147246	ASCIZ /TRANSMISSION ERROR/	
	002577	466232	351622		
	002600	476344	042644		
	002601	512372	200000		

```

2832 ;ROUTINE TO EXPAND CORE AFTER A SAVE(LOW SEG ONLY)
2833 ;CALL: MOVE DEVDAT,DEVICE ADR,
2834 ; MOVE PROG,JOBADR,
2835 ; MOVE ITEM,JOB NUMBER
2836 ; PUSHJ PDP,EXPAND
2837 ; ERROR RETURN, ADR, CHECK, OR NEED CORE BUT NOT CURRENT USER(COMMAND
2838 ; DECODER EXPANDING AFTER USER HAS REDUCE CORE)
2839 ; OK RETURN, CORE EXPANDED
2840 ;CALLED FROM SAVE AND COMMAND DECODER
2841 ;START,CSTART,DDT,REENTER,SAVE,SSAVE COMMANDS IF CORE STILL COMPRESSED
2842
2843 002602 574747 000003 EXPAND: HLRE AC3,SGALEN(PROG) ;I-LENGTH OF FILE
2844 002603 213000 000017 MOVNS AC3 ;I+LENGTH OF FILE
2845 002604 271740 002523' ADDI AC3,JOBSVM ;ADD FIRST LOC-1 TO FORM HIGHEST LEGAL ADR.
2846 002605 505740 000007 HRLI AC3,PROG ;PUT PROG IN LH FOR COMPARE
2847 ; AC3 SET FOR ADDRESS CHECKING BELOW
2848 002606 331047 002364' SKIPL TAC,JOBSV(PROG) ;IF FIRST LOC IS POSITIVE
2849 002607 254000 002741' JRST SGG01 ;OLD FORMAT, SO DONT EXPAND
2850 002610 553000 000001 HRRZS TAC ;LOOK AT 1ST WORD OF FILE
2851 002611 303040 002564' CAILE TAC,JORDDT ;IS IT BELOW JOBJDA?
2852 002612 254000 002633' JRST EXPND1 ;NO, NEW COMPRESSED FORMAT
2853 002613 302040 000000 CAIE TAC,JOBSAV ;IS JOBDOT THE DATA WORD?
2854 002614 254000 002633' JRST EXPND1 ;NO, EXPAND
2855 002615 336007 000000 SKIPM JOBSV3(JDAT) ;IS THE SAVE FILE FROM CONVERT?
2856 ; CONVERT DOES NOT ZERO COMPES
2857 ; IT JUST WRITES ENTIRE FILE WITH 1 IOWD IN FRONT
2858 002616 364740 002741' SOJA AC3,SG001 ;YES, GO BLT DATA DOWN AS IF OLD DISK SAVE FILE
2859 ; (NON-COMPRESSED)
2860 002617 561040 000000 HRR0I TAC,JOBS01 ;YES, CHANGE TO IOWD 1,JOBSDD
2861 002620 250047 002606' EXCH TAC,JOBSV(PROG) ;ZAP, THE IOWD IS FIXED
2862 002621 577000 000001 HLRES TAC ;WORD COUNT OF IOWD
2863 002622 342040 002633' AOJE TAC,EXPND1 ;JUST 1 DATA WORD - THROUGH
2864 002623 205041 000001 MOVSI TAC,1(TAC) ;MAKE IOWD N-2,JOBSAV
2865 002624 322040 002631' JUMPF TAC,EXPZ ;SO NEXT DATA WDS WILL BE SKIPPED
2866 002625 541040 002613' HRR1 TAC,JOBSAV
2867 002626 201107 002620' MOVEI TAC1,JOBSV(PROG) ;STORE IN COMPRESSED DATA
2868 002627 202042 000002 MOVEM TAC,2(TAC1)
2869 002630 254000 002633' JRST EXPND1
2870 002631 205040 777776 EXPZ: MOVSI TAC,-2
2871 002632 502047 002626' HLLM TAC,JOBSV(PROG)
    
```

```

2872                                     ;COME HERE TO DO THE ACTUAL EXPANSION OF A FILE
2873 002633 200040 003067' EXPND1: MOVE TAC,[XWD PROG,JOBSV] ;IT WAS READ INTO JOBSV
2874 002634 200120 000001          MOVE TAC1,@TAC ;FIRST IOWD
2875 002635 550640 000002 EXPLP1: HRRZ AC1,TAC1 ;ADDRESS OF IOWD
2876 002636 305640 002625' CAIGE AC1,JOBSAV ;LEGAL?
2877 002637 344640 002713' AOJA AC1,TOOLW ;NO. DELETE DATA WHICH IS TOO LOW
2878 002640 574640 000002 HLRE AC1,TAC1 ;YES. GET WOROCOUNT
2879 002641 213000 000015 MOVNS AC1 ;+N
2880 002642 506640 000002 HRLM AC1,TAC1 ;CONVERT IOWD TO +N IN LH
2881 002643 271055 000001 ADDI TAC,1(AC1) ;ADDRESS OF NEXT IOWD
2882 002644 313040 000017 CAMLE TAC,AC3 ;IN BOUNDS?
2883 002645 254000 002750' JRST SGDOER ;NO. COMPLAIN
2884 002646 271115 000000 ADDI TAC1,(AC1) ;YES. CHANGE RH OF IOWD
2885 002647 250120 000001 EXCH TAC1,@TAC ;MAKE IT XWD +N,A+N-1 AFTER DATA BLOCK
2886 002650 321100 002635' JUMPL TAC1,EXPLP1 ;CONTINUE IF NEXT THING IS AN IOWD
2887 002651 261140 000001 PUSH PDP,TAC ;SAVE DATA LOCATION
2888 002652 550060 000001 HRRZ TAC,@TAC ;TOP REAL LOCATION NEEDED
2889
2890 002653 660040 001777 TR0 TAC,1777 ;MAKE IT NK-1

2891 002654 554100 000007 HLRZ TAC1,PROG ;PRESENT SIZ OF LOW SEG
2892 002655 317040 000002 CAMG TAC,TAC1 ;IS THERE ENOUGH?
2893 002656 254000 002662' JRST EXPCOR ;YES,
2894 002657 312200 002561' CAME ITEM,JOB ;NO, IS TIS THE CURENT JOB?
2895 002660 254000 002750' JRST SGDOER ;NO, GIVE ERROR RETURN, MUST BE COMMAND DECODER
2896 ; DOING EXPAND AFTER USER HAS REDUCED CORE
2897 002661 260140 002776' PUSHJ PDP,GETCOR ;YES. GET IT
2898 002662 262140 000002 EXPCOR: POP PDP,TAC1
2899 002663 201660 000002 MOVEI AC1,@TAC1 ;TOP DATA LOC
2900 002664 505655 000001 HRLI AC1,1(AC1) ;SET TO ZERO TO TOP OF CORE
2901 002665 541655 000002 HRRI AC1,2(AC1)
2902 002666 402015 777777 SETZM -1(AC1)
2903 002667 505040 000007 HRLI TAC,PROG ;RELOCATE TOP LOC OF JOB
2904 002670 251660 000001 BLT AC1,@TAC ;ZAP, CORE IS 0 FROM HERE UP
2905 002671 561060 000002 HRR0I TAC,@TAC1 ;FROM DATA POINTER
2906 002672 550641 000000 EXPLP2: HRRZ AC1,(TAC) ;TO DATA POINTER
2907 002673 271647 000000 ADDI AC1,(PROG) ;RELOCATE
2908 002674 554701 000000 HLRZ AC2,(TAC) ;WORD COUNT
2909 002675 275116 000001 SUBI TAC1,1(AC2) ;POINT TAC1 TO PREVIOUS IOWD
2910 002676 402001 000000 SETZM (TAC) ;ZERO THIS IOWD
2911 002677 374000 000001 SOSA TAC ;POINT TO DATA
2912 002700 370000 000015 EXPLP3: SOS AC1
2913 002701 305647 002332' CAIGE AC1,JOBSDD(PROG) ;DON'T STORE DATA BELCW JOBSDD
2914 002702 374000 000001 SOSA TAC
2915 002703 262055 000000 POP TAC,(AC1) ;MOVE A DATA WORD
2916 002704 402001 000001 SETZM 1(TAC) ;ZERO WHERE IT CAME FROM
2917 002705 367702 002700' SOJG AC2,EXPLP3 ;LOOP IF MORE DATA
2918 002706 313100 003067' CAMLF TAC1,[XWD PROG,JOBSV] ;THROUGH?
2919 002707 254000 002672' JRST EXPLP2 ;NO. DO NEXT BLOCK
2920 002710 250707 002701' FXCH AC2,JOBSDD(PROG) ;YES, ZERO JOBSDD
2921 002711 202707 002611' MOVEM AC2,JOBSDD(JDAT) ;SET USER DDT STR ADR
2922 002712 254000 002747' JRST SGDO2 ;AND SETUP JSRDDT IN MONITOR PROTECTED
2923 ; FROM THIS USER
    
```

```

2924                                     ;THIS ROUTINE WILL DELETE ALL DATA FROM A COMPRESSED FILE
2925                                     ;WHICH IS BELOW JOBSDD (PROBABLY WRITTEN BY TENDUMP)
2926
2927 002713 574720 000001  TOLOW: HLRE   AC2,@TAC      ;WORDCOUNT OF OFFENDING IOWD
2928 002714 274640 000016          SUB     AC1,AC2      ;ADDRESS+N
2929 002715 307640 002710'        CAIG   AC1,JORSDD   ;IS ANY DATA IN IT LEGAL?
2930 002716 344040 002727'        AOJA   TAC,NXIOWD  ;NO, TEST NEXT IOWD
2931 002717 275640 002715'        SUBI   AC1,JORSDD   ;YES, NUMBER OF WORDS TO KEEP
2932 002720 213000 000016          MOVNS  AC2          ;TOTAL NUMBER OF WORDS
2933 002721 274700 000015          SUB     AC2,AC1    ;NUMBER OF WORDS TO DROP
2934 002722 507000 000016          HRLS   AC2          ;INTO BOTH HALVES
2935 002723 270720 000001          ADD    AC2,@TAC    ;NEW IOWD FOR ONLY GOOD DATA
2936 002724 271056 000000          ADDI   TAC,(AC2)   ;POINT TAC TO LAST BAD DATA LOC
2937 002725 202720 000001          MOVEM  AC2,@TAC    ;STORE UPDATED IOWD OVER IT
2938 002726 254000 002733'        JRST   IOWBLT     ;GO BLT OVER BAD DATA
2939 002727 274040 000016  NXIOWD: SUB     TAC,AC2    ;POINT TAC TO NEXT IOWD
2940 002730 550660 000001          HRRZ   AC1,@TAC    ;GET ADDRESS
2941 002731 305640 002617'        CAIGE  AC1,JORSDD  ;LEGAL?
2942 002732 344640 002713'        AOJA   AC1,TOLOW   ;NO, AT LEAST PART OF THE DATA IS LOW
2943 002733 205120 000001  IOWBLT: MOVSI   TAC1,@TAC  ;YES, KEEP THE ENTIRE IOWD DATA
2944 002734 541107 002632'        HRRI   TAC1,JOBSV(PPOG);TAC1 IS A BLT POINTER
2945 002735 275040 002734'        SUBI   TAC,JOBSV   ;RH OF TAC IS AMOUNT BEING DELETED
2946 002736 275741 000000          SUBI   AC3,(TAC)   ;AC3 POINTS TO TOP OF DATA READ IN-(N)
2947 002737 251120 000017          RLT    TAC1,@AC3  ;MOVE ONLY GOOD DATA DOWN
2948 002740 254000 002633'        JRST   EXPNC1     ;GO EXPAND THE GOOD DATA

```

2949						
2950	002741	201047	002711'	SGD01:	MOVEI TAC,JORDOT(PROG)	;MOVE EVERYTHING DOWN)MUST BE NON-COMPRESSED DSK FILE
2951	002742	505041	000000		HRLI TAC,JOBSVD(TAC)	;FOR CONVERT SAVE FILE
2952	002743	335007	002735'		SKIPGE JOBSV(JDAT)	;IS THIS CONVERT FILE(FIRST WORD IS IOWD)?
2953	002744	505041	000000		HRLI TAC,JOBSDP(TAC)	IYES, ALSO SKIP OVER IOWD
2954	002745	275740	002742'		SUBI AC3,JOBSVD	
2955	002746	251060	000017		RLT TAC,*AC3	
2956	002747	354003	000000'	SGD02:	AOSA (PDP)	ISET FOR OK RETURN
2957	002750	262140	000001	SGD0EK:	POP PDP,TAC	
2958	002751	402007	000127'		SETZM JORHCU(JDAT)	;CLEAR LH AND SET HIGHEST USER CHAN. IN
2959						; USE TO 0(WHERE IT SHOULD BE ANYWAY)
2960	002752	263140	000000		POPJ PDP,	;ERROR RETURN OR OK RETURN

```

2961 ;ROUTINE TO CHECK USER SUPPLIED CORE ARG AND CHECK TO SEE IF 0
2962 ;OR GREATER THAN OR EQUAL TO IOWD USED TO SAVE OR GET FILE
2963 ;CALL: HLRE TAC1,-NO, OF WORDS IN FILE
2964 ; PUSHJ PDP,CKIOWD
2965 ; RETURN WITH LARGER OF 2 POSITIVE NOS, IN TAC(1777 ORED IN)
2966 ; DO NOT RETURN IF CORE ARG SUPPLIED BY USER IS TOO SMALL
2967 ;CALLED FROM SAVE AND GET
2968
2969 EXTERN JOBSVM
2970
2971 002753 201040 002604 CKIOWD: MOVEI TAC,JOBSVM ;FIRST LOC-1 READ OR WRITTEN IN USER AREA
2972 002754 542047 000003 HRRM TAC,SGALEW(PROG) ;RESTORE RH TO JOBSAV AFTER LOOKUP
2973 002755 274040 000002 SUB TAC,TAC1 ;HIGHEST LOC=FIRST LOC-1+LENGTH OF FILE
2974 ; FALL INTO CHSARG
2975
2976 ;ROUTINE TO CHECK USER SUPPLIED CORE ARG AND CHECK IF 0
2977 ;OR GREATER THAN OR EQUAL TO CORE ARG FOR PREVIOUS SAVE
2978 ;CALL: HRRZ TAC,JOBCOR(JDAT) ;WRITTEN WHEN FILE SAVED
2979 ; PUSHJ PDP,CKSARG
2980 ; RETURN WITH LARGER OF 2 IN AC TAC, ONLY IF USER ARG NOT TOO SMALL
2981 ;CALLED ONLY FROM GET AFTER JOB DATA AREA LOADED FROM FILE(JOBCOR) OR HIGH SEG
2982
2983 INTERN NROOM1 ;CALLED FROM SEGCON
2984
2985 002756 435040 001777 CKSARG: IORI TAC,1777 ;MAKE SURE 1K-1
2986 002757 317047 000015 CAMG TAC,SGANEW(PROG) ;IS USER SUPPLIED CORE ARG BIGGER?
2987 002760 334047 000015 SKIPA TAC,SGANEW(PROG) ;YES, RETURN IT
2988 002761 336007 000015 SKIPN SGANEW(PROG) ;NO, DID USER SUPPLY ONE?
2989 002762 263140 000000 POPJ PDP, ;NO, RETURN LARGER OF TWO
2990
2991 ;ROUTINE TO PRINT #K OF CORE NEEDED
2992 ;CALL: MOVE TAC,HIGHEST REL. USER ADR,
2993 ; PUSHJ PDP,NROOM1
2994 ; NEVER RETURN
2995
2995 002763 261140 000001 NROOM1: PUSH PDP,TAC ;YES, ERROR, USER'S SUPPLIED CORE ARG TOO SMALL
2996 002764 201040 000010 NOROOM: MOVEI TAC,NECERR ;ERROR CODE IN CASE THIS IS RUN UJO( NOT ENOUGH CORE)
2997 002765 260140 003030 PUSHJ PDP,SGRELE ;RELEASE DEVICE AND ERROR RETURN TO USER IF RUN UJO
2998 ; OR FIND TTY AND PRINT ?CRLF
2999 002766 550043 000000 HRRZ TAC,(PDP) ;GET AMOUNT OF CORE REQUESTED
3000 002767 242040 777766 LSH TAC,-12 ;CONVERT TO NO, OF 1K BLOCKS-1
3001 002770 260140 001333 PUSHJ PDP,DECP1 ;ADD 1 TO TAC AND PRINT DECIMAL
3002 002771 265040 002575 JSP TAC,PHOLD ;START TTY AND STOP JOB
3003 002772 455011 743100 ASCIZ /K OF CORE NEEDED/
002773 416372 242500
002774 472130 542212
002775 420000 000000

```

```
3004
3005
3006 ;ROUTINE TO ASSIGN CORE FOR LOW AND HIGH SEG
3007 ;CALL: MOVE PROG,LOW SEG RELOCATION
3008 ; HRR TAC,HIGHEST LOC DESIRED
3009 ; PUSHJ PDP,GETCOR
3010 ; RETURN ONLY IF ASSIGNED
3011
3012 INTERN GETCOR
3013
3014 002776 552047 000014 GETCOR: HRRZM TAC,SGACOR(PROG) ;SOTRE CORE ARG FOR CORE UUD IN USER AC
3015 002777 261140 000001 PUSH PDP,TAC ;SAVE IN CASE OF ERROR
3016 003000 047600 000011 CALLI SGACOR,11 ;DO CORE UUD
3017 003001 254000 002764 JRST NOROOM ;NOT AVAILABLE, PRINT ERROR AND AMOUNT TRYING FOR
3018 003002 254000 002022 JRST TPOPJ ;OK, REMOVE TAC FROM PD LIST AND RETURN
```

```

3019          ;ROUTINE TO PRINT NOT A SAVE FILE IF WRONG FORMAT FILE DETECTED
3020
3021          INTERN GETERR
3022
3023 003003 201040 000007 GETERR: MOVEI TAC,NSFERR          ;ERROR CODE IN CASE THIS IS RUN UUO(NOT SAVE FILE)
3024 003004 260140 003030' PUSHJ PDP,SGRELE          ;RELEASE DEVICE AND ERROR RETURN TO USER IF RUN UUO
3025                                     ; OR FIND TTY AND PRINT ?CRLF
3026 003005 265040 002771' JSP TAC,PHOLD          ;START TTY AND STOP JOB
3027 003006 472372 420202          ASCIZ /NOT A SAVE FILE/
      003007 202470 153212
      003010 202151 146212
      003011 000000 000000

3028
3029          ;ROUTINE TO PRINT FILE NOT FOUND OR NEEDS 2 RELOC REG
3030
3031          INTERN NOFILE ;CALLED FROM SEGCON
3032
3033 003012 201040 000000 NOFILE: MOVEI TAC,PNFERR          ;ERROR CODE IN CASE THIS IS RUN UUO(FILE NOT FOUND)
3034                                     ; CHANGE ERROR CODE TO DISK ERROR CODE IF DEV IS DSK
3035 003013 260140 003025' PUSHJ PDP,SGRELL          ;RETURN DISK LOOKUP OR ENTER ERROR CODE IF DSK
3036                                     ; RELEASE DEVICE AND ERROR RETURN TO USER IF HE WANTED
3037
3038
3039 003014 200107 000000          MOVE TAC1,SGANAM(PROG) ; OR FIND TTY AND PRINT ?CRLF
3040 003015 260140 001661' PUSHJ PDP,PRNAME          ;PRINT FILE NAME
3041 003016 260140 002041' PUSHJ PDP,PRPER
3042 003017 510107 000001          HLLZ TAC1,SGAEXT(PROG) ;PRINT PERIOD
3043 003020 260140 003015' PUSHJ PDP,PRNAME          ;PRINT EXTENSION
3044 003021 265040 003005' JSP TAC,PHOLD          ;START TTY AND STOP JOB
3045 003022 202351 752100
      003023 432372 547210
      003024 000000 000000
  
```

```

3046 ;ROUTINE TO RELEASE DEVICE ON AN ERROR AND CHECK TO SEE
3047 ;IF THIS IS A MONITOR COMMAND OR USER UUO
3048 ;IF USER UUO, GIVE ERROR RETURN TO USER UNLESS THERE IS A HALT
3049 ;IN LH OF ERROR RETURN WORD, IN WHICH CASE FIND TTY, PRINT ?CRLF
3050 ;AND RETURN TO CALLER SO CAN ADD MORE INFO TO ERROR MESSAGE AND STOP JOB
3051 ;CALL: MOVEI TAC,ERROR CODE(DEFINED IN S,MAC)
3052 ; PUSHJ PDP,SGRELE
3053 ;DO NOT RETURN TO CALLER IF USER WANTS ERROR RETURN ON RUN AND GETSEG UUOS
3054
3055 EXTERN JOBPD1,JOBPD3,USRJDA
3056
3057 003025 SGRLE1: ;LOOKUP OR ENTER FAILURE
3058 IFN FTDISK,<
3059 003025 200106 000004 MOVE TAC1,DEVMOD(DEVNAT) ;IS THIS DEVICE A DISK?
3060 003026 603100 200000 TLNE TAC1,OVDISK
3061 003027 550047 000001 HRRZ TAC,SGAEXT(PROG) ;YES, RETURN DISK SERVICE ERROR CODE
3062 >
3063 003030 200107 002526 SGRLE: MOVE TAC1,JOBPD1(JDAT) ;GET FIRST PC ON PD LIST
3064 003031 607100 010000 TLNN TAC1,USRMOD ;IS IT IN USER MODE(IE USER UUO)?
3065 003032 254000 003050 JRST SGRLE1 ;NO, MUST BE MONITOR COMMAND OR CALLER OVERLAYED
3066 ; RELEASE DEVICE, FIND TTY, AND RETURN TO CALLER
3067 003033 261140 000001 PUSH PDP,TAC ;SAVE ERROR CORE
3068 003034 540600 000002 HRR UUO,TAC1 ;ADDRESS OF RETURN AFTER RUN OR GETSEG UUO
3069 003035 260140 002275 PUSHJ PDP,GETWDU ;GET ERROR RETURN WORD FROM RUN OR GETSEG UUO
3070 003036 554100 000001 HLRZ TAC1,TAC ;GET OP CODE
3071 003037 262140 000001 POP PDP,TAC ;RESTORE ERROR CODE
3072 003040 306100 254200 CAIN TAC1,(HALT) ;IS LH HALT?
3073 003041 254000 003050 JRST SGRLE1 ;YES, RELEASE DEVICE, FIND TTY, AND RETURN TO CALLER
3074 003042 540607 000000 HRR UUO,JOBPD3(JDAT) ;NO, AC NUMBER OF RUN OR GETSEG UUO
3075 003043 260140 000000 PUSHJ PDP,STOTAC ;STORE ERROR NUMBER IN USER AC
3076 003044 332000 002542 SKIPE USRJDA+0 ;DO NOT RELEASE CHANNEL 0 IF NOT INITED YET
3077 ; UUO HANDLER DOES NOT ALLOW THIS FROM EXEC MODE
3078 003045 071000 000000 RELEAS 0, ;RELEASE DEVICE(IF INITED)
3079 003046 261147 003030 PUSH PDP,JOBPD1(JDAT) ;PUT RETURN ON END OF PDLIST
3080 003047 254000 002431 JRST USRXIT ;AND RETURN TO USER TO HANDLE ERROR
3081
3082
3083 003050 260140 002542 SGRLE1: PUSHJ PDP,SGREL ;RELEASE DEVICE AND FIND TTY
3084 003051 265040 002017 JSP TAC,CONMES ;PRINT ?CRLF AND RETURN TO CALLER
3085 ; WHO WILL PRINT REST OF ERROR MESSAGE AND STOP JOB
3086
3087 003052 374321 200000 / ASCII? /?
3088
3089
3090 003053 COMEND: END ;END OF COMCON
3091 003053 545747 576564
3092 003054 504554 600000
3093 003055 637163 644164
3094 003056 435755 605154
3095 003057 545747 515600
3096 003060 637163 000000
3097 003061 446353 000000
3098 003062 102500 001341

```

COMCON - COMMAND DECODER AND SAVEGET ROUTINES - V433 MACRO,V36 19:02 4-JUN-69 PAGE 105-1
SAVGET - SAVE,GET,R,RUN COMMANDS AND RUN,GETSEG UUOS

3099	003063	440600	000002
3100	003064	064240	000000
3101	003065	201004	020000
3102	003066	000007	002717'
3103	003067	000007	002743'

NO ERRORS DETECTED

PROGRAM BREAK IS 003070

AC1	000015	INT	AC2	000016	INT	AC3	000017	INT
ACTCH1	001672	'	ACTCHK	001667	'	ADRERR	002563	FXT
ALOGIN	000562	'	ALOGN1	000565	'	ANYACT	001670	FXT
ANYRIN	002074	'	ARCOM	000621	'	ARCOM1	000622	'
ASGHHG	002054	EXT	ASSASG	001264	EXT	ASSCON	400000	INT
ASSER1	000707	'	ASSF1	000735	'	ASSFIN	000723	'
ASSG0	000653	'	ASSG1	000665	'	ASSG2	000675	'
ASSG3	000651	'	ASSG4	000656	'	ASSG5	000662	'
ASSG6	000642	'	ASSG7	000701	'	ASSIGN	000633	'
ASSMS2	000716	'	ASSPRG	200000	INT	ATT1	001206	'
ATT2	001214	'	ATT3	001224	'	ATT4	001231	'
ATT5	001235	'	ATT6	001242	'	ATTACH	001136	INT
BREAKB	001766	EXT	BUFNT	000012	INT	BUFWRD	000013	INT
C	000012	'	CBLANK	001570	'	CCLRUN	000570	'
CERR	000433	'	CFINI	001574	'	CFINS	001577	'
CHKACT	000101	'	CHKCO1	000115	'	CHKCO2	000113	'
CHKDLY	000111	'	CHKMED	002425	EXT	CHKNO	000041	'
CHKRUN	000070	'	CHKSTR	000447	'	CHKXPN	000125	'
CKIOWD	002753	'	CKSARG	002756	'	CLOG2	000607	'
CLOGIN	000600	'	CLRWRD	001371	EXT	CLSOUT	000001	INT
CMPLP1	002343	'	CMPLP2	002352	'	CMPTH	002362	'
CMWB	200000	INT	CMWRQ	002000	'	COM0	000001	'
COMCNT	000155	EXT	COMCON	000000	INT	COMDIS	000143	'
COMEND	003053	'	COMER	000134	'	COMER1	002010	'
COMERA	002000	INT	COMERL	002003	'	COMERP	001777	INT
COMERR	002001	INT	COMFND	000027	'	COMGO	000137	'
COMLP	000013	'	COMMAN	000000	INT	COMNEQ	000024	'
COMRET	000145	'	COMRT1	000161	'	COMTAB	000232	'
CON0	002021	'	CONFIG	000175	EXT	CONMES	002017	INT
CONMS1	002020	'	CONOUT	002023	'	CONT	000463	'
CONTC	000463	'	CONTC1	001675	'	COR0	000371	'
COR1	000504	'	COR2	000507	'	COR3	000512	'
CORE	000473	'	CORE0	002226	FXT	CORE1	000404	FXT
CORMAX	000516	EXT	CPOPJ	002220	EXT	CPPOPJ1	002240	FXT
CPOPJ2	002105	EXT	CRLF	002016	INT	CTEX	001735	'
CTEX0	001727	'	CTEX1	001730	'	CTEXA	001747	'
CTEXT	001721	INT	CTEXT1	001723	INT	CTXCNC	001753	'
DAMESS	003062	EXT	DAT	000005	INT	DAYTIM	001312	'
DAYTM1	001330	'	DCOM	001524	'	DCOM1	001531	'
DCOMA	001545	'	DDTGO	000446	'	DEA1	002062	'
DEAER1	000770	'	DEAER2	000776	'	DEASAL	002060	INT
DEASG	002043	INT	DEASSI	000757	'	DEASTY	002060	INT
DEAT	001535	'	DECIN	002071	INT	DECIN1	002071	INT
DECLF	000406	INT	DECP1	002770	EXT	DETACH	001247	'
DEVATT	001164	'	DEV0AT	000006	INT	DEVIOS	000002	INT
DEVLG	000727	EXT	DEVLOG	000005	INT	DEVLST	002061	FXT
DEVMOD	000004	INT	DEVNAM	000000	INT	DEVUPR	001561	FXT
DEVPHY	000645	EXT	DEVSEF	000003	INT	DEVSRC	002450	FXT
DISP	000311	'	DISPL	000057	'	DLYCM	001676	'
DLYCM1	001677	'	DLYCOM	001676	FXT	DNAERR	000011	INT
DR	000016	INT	DSEF	000011	INT	DQKDDB	001705	FXT
DQKFGS	001706	EXT	DVOIRI	400000	INT	DVDSK	200000	INT
DVMTA	000020	INT	DVTY	000010	INT	ECOM	001453	'

ECOMA	001550'		ERNAM	002477'	EXT	ERRFLG	000001	
ERRMES	002015'	INT	ESTOP	001623'	EXT	EXPAND	002602'	
EXPCOR	002662'		EXPLP1	002635'		EXPLP2	002672'	
EXPLP3	002700'		EXPND1	002633'		EXPZ	002631'	
FDV1	001607'		FNFERR	000000	INT	FREDEV	001624'	
FT2REL	777777	INT	FTATTA	777777	INT	FTCCL	777777	INT
FTDISK	777777	INT	FTEXAM	777777	INT	FTFINI	777777	INT
FTKCT	777777		FTLOGI	777777	INT	FTRC10	777777	
FTREAS	777777	INT	FTSWAP	777777	INT	FTTALK	777777	INT
FTTIME	777777	INT	FTTTY5	777777		GET	000536'	
GETARG	002270'	INT	GETCHR	002144'	EXT	GETCOR	002776'	INT
GETERR	003003'	INT	GETHGH	002512'	EXT	GETJB	002512'	
GETJOB	002406'	INT	GETMIN	002221'		GETWD1	002311'	FXT
GETWDU	003035'	EXT	GETWRD	002234'	INT	HELP	000554'	
HELP1	000550'		HELPPP	000550'	EXT	HGHDEP	001545'	FXT
HGHWRD	002236'	EXT	HIGHJB	000067'	EXT	IADRCK	002455'	FXT
ILUERR	000013	INT	IMGIN	002213'	EXT	INCR	004000	
INCORE	024000		INLMES	002025'	INT	INPB	010000	INT
IOBKTL	040000	INT	IODEND	020000	INT	IODERR	200000	INT
IODTER	100000	INT	IOIMPM	400000	INT	IOS	000000	INT
IODEND	002000	INT	IOWBLT	002733'		IOWG	000000	FXT
IPOPJ	002057'	EXT	ITEM	000004	INT	JACCT	100000	INT
JBTAADR	002215'	EXT	JBTDAT	000000	EXT	JBTKCT	001426'	FXT
JBTPRG	002421'	EXT	JBTSTS	001674'	EXT	JDAT	000007	INT
JERR	020000	INT	JIFMIN	002174'	EXT	JIFSC2	002171'	FXT
JIFSEC	002170'	EXT	JLOG	000004	INT	JNA	040000	INT
JOB	002657'	EXT	JOR1	000375'		JOB41	002532'	FXT
JOBCOR	002541'	EXT	JOR0A	002225'	EXT	JOBDAC	000000	FXT
JOBDDT	002741'	EXT	JOBEXM	001536'	EXT	JOBFDV	001620'	FXT
JOBFF	002452'	EXT	JOBHCU	002751'	EXT	JOBKA	000430'	
JOBKR	000434'		JOBKL	000425'	INT	JOBMAX	000050'	FXT
JOBK	001354'	EXT	JOBPC	001541'	EXT	JOBPD1	003046'	FXT
JOBPD3	003042'	EXT	JOBPF1	001526'	EXT	JOBREN	000444'	FXT
JOB541	002531'	EXT	JOB5A	002426'	EXT	JOB5AV	002636'	FXT
JOB5D1	002731'	EXT	JOB5DD	003066'	EXT	JOB5OP	002744'	FXT
JOB5V	003067'	EXT	JOB5V3	002615'	EXT	JOB5VD	002745'	FXT
JOB5VM	002753'	EXT	KILHGH	002224'	EXT	KJOB	000410'	
KJOB2	000422'		KSTOP	000432'	EXT	LINSAV	001703'	FXT
LOGDET	000572'		LOGER1	001305'		LOGERR	000751'	
LOGREF	000542'		LOGS1Z	000603'	EXT	LOP01	001645'	
LOP01A	001661'		LOP01B	001653'		LOP02	001662'	
LOP05	001630'		LOP06	001635'		LOWFIN	002533'	
LSTWRD	001377'	EXT	MONJOB	000424'	EXT	MONSTR	002436'	FXT
MONTAB	001337'	EXT	MSTART	002267'	EXT	NECERR	000010	INT
NEWJOB	000060'		NOACT	040000		NOCOM	001573'	
NOCORE	400000		NOCORQ	000377'	EXT	NOCRLF	000200	
NODATE	001000		NOFILE	003012'	INT	NOINCK	000400	
NOJOBK	200000		NOLOGI	100000		NOMESS	000010	
NOPEP	000100		NOROOM	002764'		NORUN	010000	
NOTDEV	000770'	INT	NOTEMF	002034'	INT	NOTRAN	020000	
NROOM1	002763'	INT	NSDERR	000012	INT	NSFERR	000007	INT
NSRBIT	400000	INT	NUMBIT	001633'	EXT	NUMLOP	000045'	
NXLOWJ	002727'		OCT0	002075'		OCTIN	002073'	INT

OCTIN1	002073'	INT	OCTPNT	001566'	FXT	CUTCHS	002031'	FXT
PCORSZ	000000'	EXT	PCRLF	000200'	FXT	PCRLF1	000213'	
PDP	000003'	INT	PHOLD	003021'	FXT	PJBSTS	000000'	FXT
PJOB	000405'	INT	PJOBN	002043'	FXT	PJPGNO	002117'	
PLSXPN	000004'	DLD	PP0	002116'		PP1	002126'	
PR0	002156'		PR1	002162'		PRJPRG	001316'	FXT
PRNAME	003020'	EXT	PROG	000007'	INT	PRPER	002041'	INT
PRQM	002030'	INT	PRSPC	002032'		PRT2	002204'	
PRT2LF	002202'		PRTERR	000002'	INT	PRTHGH	000513'	FXT
PRTIM1	002174'		PRTIME	002146'	INT	PRTSEG	002210'	INT
PRTTIM	001420'		PUNIT	001275'	EXT	PUUOAC	002273'	FXT
R	000013'		RADX10	002212'	EXT	RCOM	000617'	
REASE1	001111'		REASE2	001115'		REASE3	001121'	
REASE4	001113'		REASE5	001116'		REASE6	001123'	
REASE7	001125'		REASS	001015'		REASS1	001026'	
REASS2	001071'		REASS3	001102'		REASS4	001066'	
REASS3	001006'	INT	REENTE	000444'		REFLAG	000607'	FXT
RELEA5	001075'	EXT	RELEA6	002056'	EXT	REQUE	000220'	FXT
RESET	002437'	EXT	RTIME	001416'	FXT	RUN	000000'	INT
RUN1	001413'		RUN2	001360'		RUNC1	000627'	
RUNCHK	001665'		RUNCM	000630'		RUNC02	000627'	
RUNCOM	000625'		RUNJOB	002413'	INT	RUNTIM	001350'	
SATENT	001627'	EXT	SATXWD	001625'	EXT	SAVDMP	002470'	FXT
SAVE	000531'		SAVERR	002400'	INT	SAVFIN	002374'	
SAVHGH	002320'	EXT	SAVJR1	002322'		SAVJOB	002314'	INT
SEGSIZ	002213'	INT	SENTS2	001635'	EXT	SETBFI	002010'	FXT
SETEX1	002304'	EXT	SETEXT	002254'	EXT	SETRUN	000212'	FXT
SG1	002436'		SG2	002437'	INT	SG2A	002440'	
SG3	002452'	INT	SG4	002463'	INT	SGACOR	000014'	INT
SGADAT	000002'	INT	SGADEV	000011'	INT	SGADMP	000013'	INT
SGAEND	000004'	INT	SGAEXT	000001'	INT	SGAHED	000012'	INT
SGAHGH	000016'	INT	SGALEN	000003'	INT	SGALOW	000017'	INT
SGAMOD	000010'	INT	SGANAM	000000'	INT	SGANEW	000015'	INT
SGAPPN	000007'	INT	SGAREN	000004'	INT	SGDO	002557'	
SGD01	002741'		SGD02	002747'		SGD0A	002560'	
SGDOER	002750'		SGERR1	002504'		SGERRA	002473'	INT
SGIOCK	002570'	INT	SGREL	002542'	INT	SGREL1	002554'	
SGRELE	003030'		SGRELL	003025'		SGRLE1	003050'	
SGSET	002241'		SGSET1	002265'		SHFWRD	001363'	FXT
SKED1	001565'		SKEDUL	001555'		SKIPS	001756'	INT
SKIPS1	001757'	INT	SKIPS2	001770'		SKIPS3	001773'	
SKIPSA	001762'		SKPBR1	000227'		SKPBR2	000223'	
SKPBRK	000222'		SNOARG	000443'		SPCHEK	001764'	FXT
SSAVE	000532'		START	000436'		STARTC	000436'	
STATES	001565'	EXT	STOP	000454'		STOP1	000461'	FXT
STOPR	000462'		STOPC	001714'		STOPCK	001705'	
STOP0	001707'		STOTAC	003043'	FXT	STREQ	000703'	FXT
STUSER	001665'	EXT	SWP	002000'	INT	SYSDEV	000100'	INT
SYSPP	001252'	EXT	SYSTAP	001577'	FXT	SYSTAT	000557'	
T	000012'		T1	000013'		T2	000014'	
T3	000015'		T4	000016'		TAC	000001'	INT
TAC1	000002'	INT	TAKR	001760'	EXT	TALK	001432'	
TALK1	001445'		TALK2	001442'		TEM	000010'	INT

TEMP1	001302'	TEMP2	001304'	THSDAT	001330' FXT
TIME	001346' EXT	TISYNC	002011' EXT	TITAKR	002004' FXT
TOLOW	002713'	TPOPJ	003002' EXT	TRESCN	002002' FXT
TRNERR	000006 INT	TSETBF	001162' EXT	TTIME	001420' FXT
TTYATI	000166' EXT	TTYATT	001157' EXT	TTYCM	001704' FXT
TTYCOM	000000' EXT	TTYDET	001250' EXT	TTYFND	001163' FXT
TTYFNU	000376' EXT	TTYFUW	002556' EXT	TTYKIL	000431' EXT
TTYRNC	000020	TTYRNU	000040	TTYSTR	000214' EXT
TTYTAB	001277' EXT	TTYTLK	001437' EXT	TTYUSR	000210' FXT
TYOINS	002023'	UCHN	000012 INT	UGETHI	002432' FXT
UGTERR	002433' INT	UGTSEG	002432' INT	UPTIME	001411' FXT
URUN	002415' INT	URUN1	002422'	USRDDT	002565' FXT
USRHCU	002566' EXT	USRJDA	003044' EXT	USRMOD	010000 INT
USRREL	002540' EXT	USRXIT	003047' EXT	USTART	000447' FXT
UUC	000014 INT	UUCERR	002435' EXT	VCOMCN	000433 INT
VIRTUAL	000602' EXT	WDPJIF	001372' EXT	WLBIT	001631' FXT
ZERSWP	000400' EXT				

