```
                                ;***COPYRIGHT 1969, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.***


                                ;THIS SUB-PROGRAM ASSEMBLED WITH SYSTEM PARAMETER FILE - S,MAC(V414)
                                        XLIST
                                        LIST
                                ;THIS SUB-PROGRAM ASSEMBLED WITH CONFIGURATION DEPENDENT FEATURE SWITCHES - FT5ØSB,MAC(
                                VØØ3)
                                        XLIST
                                        LIST
                                IFE FTDISK+FTRC1Ø+2,<
                                TITLE   SCHEDB - SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(1Ø/5Ø)(BURROUGHS DISK)
                                >
                                IFE FTDISK+FTRC1Ø+1,<
                                TITLE   SCHEDD - SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(1Ø/5Ø)(DATA PRODUCTS DISK)
                                >
                                SUBTTL  CLKCSW R.KRASIN/AF/TH/RCC  TS  Ø2 JUNE 69  V421
                                XP VSCHED,421+
                                                ;PUT VERSION NUMBER IN GLOB LISTING AND LOADER STORAGE MAP


                                INTERNAL        FTRC1Ø  ;THIS SOURCE FILE MAY BE ASSEMBLED TO USE EITHER THE
                                                ; NEW PDP-1Ø DISK (MODEL RC-1Ø) OR THE OLD PDP-6 DISK (DATA
                                                ; PRODUCTS DISK FILE) FOR SWAPPING.

                                IFN     FTRC1Ø, <
                                ENTRY   RCXSKD  ;THIS SYMBOL IS SOLELY TO PERMIT SYSTEM
 ØØØØØØ                         RCXSKD:         ; BUILDER TO RETRIEVE THE CORRECT BINARY FILE.
                                INTERNAL        XCKCSW
                                >
                                IFE     FTRC1Ø, <
                                ENTRY   XCKCSW
                                >


                                EXTERNAL JOB,JBTSTS
                                EXTERNAL JBTQ,PJBSTS
                                EXTERNAL PJBSTS,TIMEF,MJOBN

                                INTERNAL NXTJOB,FTSWAP

                                ;ACCUMULATOR DEFINITIONS ---
                        ØØØØØ6  QJ=DEVDAT       ;QJOB WORD
                        ØØØØØ2  SW=TAC1         ;STATUS WORD
                        ØØØØØ4  J=ITEM          ;JOB NO.

                                ;INITIALIZE SCHEDULER  (CALLED FROM SYSINI BEFORE ALL OTHER
                                ;       DEVICES ARE INITIALIZED)

                                INTERNAL NXTINI

 ØØØØØØ                         XCKCSW:
 ØØØØØØ   2Ø1Ø4Ø   ØØØØ11       NXTINI: MOVEI TAC,MAXQ          ;MAX. NO. OF QUEUES
 ØØØØØ1   4Ø2ØØ1   ØØØ244'              SETZM AVALTB(TAC)       ;CLEAR SHARABLE DEVICE AVAIL. FLAGS
 ØØØØØ2   476ØØ1   ØØØ275'              SETOM REQTAB(TAC)       ;SET SHARABLE DEVICE REQUEST COUNT
                                                                ; TO -1,  I.E. NO JOB WAITING OR
```

```
                                                 ; USING DEVICE OTHER THAN INITIALIZATION
ØØØØ73  365Ø4Ø  ØØØØØ1'         SOJGE TAC,,-2
ØØØØ74  4Ø2ØØØ  ØØØ322'         SETZM QJOB       ;CLEAR NO. OF JOBS NEEDING REQUEING
ØØØØ75  4Ø2ØØØ  ØØØ323'         SETZM XJOB       ;CLEAR NO. OF JOBS NEEDING EXPANDING
ØØØØ76  4Ø2ØØØ  ØØØ321'         SETZM JOBQUE     ;CLEAR JOB NO. TO BE REQUEUED
```

```
    62  000027  263140  000230          POPJ PDP,
    63
```

```
                                 ;NXTJOB DECREMENTS CURRENT JOB'S QUANT. AND PROTECT
                                 ;TIMES AND REQUEUES IT IF QUANT. TIME GONE TO Ø.
                                 ;SERVICES ANY JOB REQUEUING REQUESTED AT OTHER PRIORITY
                                 ;LEVELS THEN CALLS SHUFFLER,SWAPPER AND SCHEDULAR.
                                 ;MAKES NO ASSUMPTIONS RE. ACS
                                 ;RETURNS NEXT JOB TO RUN IN J.


                                 EXTERNAL JBTSWP,POTLST

ØØØØ1Ø  33602Ø  ØØØØØØ  NXTJOB: SKIPN TIMEF               ;CLOCK TIC?
ØØØØ11  2540ØØ  ØØØØ22'         JRST NXTJB1               ;NO
ØØØØ12  2052ØØ  ØØØ2ØØ          MOVSI J,MJOBN             ;YES
ØØØØ13  2051ØØ  777777          MOVSI SW,-1               ;DECREMENT IN CORE PROTECT TIME
ØØØØ14  20524Ø  ØØ2ØØØ          MOVSI DAT,SWP
ØØØØ15  612244  ØØØØØØ          TDNE DAT,JBTSTS(J)
ØØØØ16  254ØØØ  ØØØØ21'         JRST .+3
ØØØØ17  3310Ø4  ØØØØØØ          SKIPL JBTSWP(J)
ØØØØ2Ø  272124  ØØØØ17'         ADDM SW,JBTSWP(J)

ØØØØ21  2532ØØ  ØØØØ15'         AOBJN J,.-4

ØØØØ22  33622Ø  ØØØØØØ  NXTJB1: SKIPN J,JOB               ;CURRENT JOB NO., IS IT NULL JOB?
ØØØØ23  254ØØØ  ØØØØ41'         JRST CKJB1                ;YES,GO SEE IF OTHER JOBS NEED RESCHEDULING
ØØØØ24  2Ø134Ø  ØØØØØØ          MOVEI QJ,Ø                ;GET READY IN CASE CURRENT JOB UNRUNABLE
ØØØØ25  554124  ØØØØ15'         HLRZ SW,JBTSTS(J)         ;GET JOB STATUS BITS AND CODES
ØØØØ26  62Ø1ØØ  3114Ø4          TRZ SW,RUNMSK+CMWB               ;MASK OUT DO NOT CARE BITS
ØØØØ27  3021ØØ  44ØØØØ          CAIE SW,RUNABLE           ;IS CURRENT JOB RUNABLE?
ØØØØ3Ø  254ØØØ  ØØØØ51'         JRST CKJB3                ;NO, REQUE CURRENT JOB
ØØØØ31  336ØØØ  ØØØØ1Ø'         SKIPN TIMEF               ;NO, IS THIS A TIME INTERRUPT?
ØØØØ32  254ØØØ  ØØØØ41'         JRST CKJB1                ;NO.

ØØØØ33  3701Ø4  ØØØØ25'         SOS SW,JBTSTS(J)          ;DECREMENT QUANT. TIME
ØØØØ34  6021ØØ  777777          TRNE SW,-1                ;HAS TIME GONE TO Ø?
ØØØØ35  254ØØØ  ØØØØ41'         JRST CKJB1                ;NO
ØØØØ36  20124Ø  ØØØ4Ø2'         MOVEI DAT,QTIME           ;YES--REQUEUE AND RESET QUANT. TIME
ØØØØ37  2ØØØ4Ø  ØØØ321'         MOVE TAC,JOBQUE
ØØØØ4Ø  26014Ø  ØØØ162'         PUSHJ PDP,QXFER
```

```
000041  337300  000322' CKJB1:  SKIPG   QJ,QJOB         ;SET QJ NON ZERO IF ANY REQUEUEING TO DO
000042  254000  000073'         JRST    CKJB5           ;NO REQUEUEING NECESSARY
000043  201200  000000          MOVEI   J,JOBMAX        ;START WITH HIGHEST JOB NUMBER ASSIGNED
000044  205100  000002  CKJB2:  MOVSI   SW,JRQ          ;JOB NEEDS REQUEUEING BIT
000045  616104  000033'         TDNN    SW,JBTSTS(J)    ;THIS JOB?
000046  367200  000045'         SOJG    J,.-1           ;NO,KEEP LOOKING
000047  323200  000073'         JUMPLE  J,CKJB5         ;YES,LOOKED AT ALL JOBS?
                                                        ; (MAY NOT FIND A KJOBED JOB IF HIGHEST
                                                        ; GO DECR. COUNT QJOB ANYWAY)
000050  412104  000045'         ANDCAM  SW,JBTSTS(J)    ;NO,MARK THIS JOB AS DONE
000051  200104  000050' CKJB3:  MOVE SW,JBTSTS(J)       ;JOB STATUS WORD
000052  201240  000330'         MOVEI DAT,QCMW          ;ASSUME COMMAND WAIT
000053  607100  200000          TLNN SW,CMWB            ;IS JOB IN COMMAND WAIT?
000054  254000  000057'         JRST CKJB9              ;NO.
000055  603100  002001          TLNE SW,SWP+JXPN                ;YES, IS JOB ON DISK, OR TRYING TO EXPAND?
000056  254000  000067'         JRST CKJB4A             ;YES, PUT JOB IN COMMAND WAIT Q
000057  325100  000066' CKJB9:  JUMPGE SW,CKJB4         ;NO,WAIT STATUS CODE DETERMINES NEW Q
000060  135100  000000          LDB SW,PJBSTS           ;YES, GET QUEUE CODE.
000061  306100  000001          CAIN    SW,WSQ          ;*** EXPERIMENTAL ***
000062  254000  000132'         JRST    CKJB10          ;*** EXPERIMENTAL ***
000063  306100  000013          CAIN SW,TIOWQ           ;CURRENT JOB GOING INTO TTY IO WAIT?
000064  563004  000020'         HRROS JBTSWP(J)         ;YES, SET IN CORE PROTECT TIME TO -1,
                                                        ; SO HE CAN BE SWAPPED IMMEDIATELY IF SOMEONE
                                                        ; ELSE WANTS TO BE SWAPPED IN
000065  334242  000256' CKJB4B: SKIPA DAT,QBITS(SW)     ;GET ADDRESS OF TRANSFER TABLE
000066  201240  000326' CKJB4:  MOVEI DAT,QSTOP         ;IF RUN BIT WAS OFF
000067  260140  000162' CKJB4A: PUSHJ PDP,QXFER         ;REQUEUE THE JOB
000070  322300  000113'         JUMPE QJ,SCHFD          ;IF FROM NXTJOB GO DIRECTLY TO SCHED
                                                        ; I.E, CURRENT JOB NO LONGER RUNNABLE(IOW)
                                                        ; BUT JRQ WASN'T SET SO DON'T DECR QJOB
000071  373300  000322'         SOSLE   QJ,QJOB         ;ANY MORE JOBS TO REQUEUE?
000072  367200  000044'         SOJG    J,CKJB2         ;YES,BUT LOOK AT EACH JOB ONLY ONCE PER CLOCK TICK

000073  201300  000011  CKJB5:  MOVEI QJ,AVLNUM         ;CK AVAL FLAGS FOR SHAR. DEVS.
000074  336006  000244' CKJB6:  SKIPN AVALTB(QJ)        ;FLAG=0?
000075  367300  000074'         SOJG QJ,CKJB6           ;YES - TRY NEXT ONE
000076  323300  000112'         JUMPLE QJ,CKJB7         ;NO - OR FINISHED?
000077  544206  000307'         HLR J,AVLQTB(QJ)        ;NO--GET 1ST JOB IN Q
000100  570204  000000  CKJB6A: HRRE J,JBTQ(J)
000101  323200  000111'         JUMPLE J,CKJB8          ;FINISHED Q? WAIT TILL SWAPPER BRINGS IN JOB
000102  200244  000051'         MOVE DAT,JBTSTS(J)      ;IS JOB IN CORE?
000103  603240  002000          TLNE DAT,SWP
000104  254000  000100'         JRST CKJB6A             ;NO, LOOK AT NEXT JOB IN THIS QUEUE
                                                        ; TO SEE IF IN CORE.
000105  550246  000307'         HRRZ DAT,AVLQTB(QJ)     ;NO--GET TRANS. TABLE ADDRESS
000106  402006  000244'         SETZM AVALTB(QJ)        ;CLEAR AVAL FLAG
000107  301300  000003          CAIL QJ,MINQ            ;LESS THAN MIN. SHARABLE DEV. Q?
000110  260140  000162'         PUSHJ PDP,QXFER         ;REQUEUE THE JOB AND PUT IT IN
                                                        ; PROCESSOR Q SO SCHEDULER WILL RUN IT
000111  367300  000074' CKJB8:  SOJG QJ,CKJB6           ;CONTINUE IF ANY MORE FLAGS TO LOOK AT
```

```
ØØØ112                      CKJB7:                          ;NONE--GO SHUFFLE AND SWAP
                            IFE FTSWAP,<    EXTERNAL CHKSHF
                                    PUSHJ PDP,CHKSHF
                            >
                            IFN FTSWAP,<
ØØØ112  26Ø14Ø  ØØØ465'             PUSHJ PDP,SWAP>

                            ;SCHEDULAR--SEARCH THRU QUEUES ACCORDING TO SSCAN TABLE
                            ;FOR 1ST JOB IN CORE--RETURN ITS NO. IN J

ØØØ113  4Ø2ØØØ  ØØØØØØ      SCHED:  SETZM POTLST            ;CLEAR POTENTIALLY LOST TIME FLAG
ØØØ114  2Ø124Ø  ØØØ43Ø'             MOVEI DAT,SSCAN         ;ADDRESS OF SCAN TABLE
ØØØ115  26524Ø  ØØØ225'             JSP TAC,QSCAN           ;BEGIN SCAN
ØØØ116  254ØØØ  ØØØ13Ø'             JRST SCHD1              ;NO MORE JOBS--RETURN NULLJOB
ØØØ117  476ØØØ  ØØØ113'             SETOM POTLST            ;SET POTENTIALLY LOST TIME FLAG FOR CLOCK1
ØØØ12Ø  2ØØ374  ØØØ1Ø2'             MOVE QJ,JBTSTS(J)       ;IS THIS JOB SWAPPED OUT
                                    TLNE QJ,SWP+SHF+JXPN    ;MONITOR WAITING FOR I/O TO STOP,OR JOB EXPANDING CORE?
ØØØ121  60334Ø  ØØ6ØØ1
ØØØ122  254ØØ2  ØØØØØØ              JRST (TAC1)             ;YES--CONTINUE SCAN,JOB CANNOT BE RUN


ØØØ123  5761ØØ  ØØØ321'             HLREM TAC1,JOBQUE       ;YES--SAVE ITS Q
ØØØ124  2053ØØ  ØØØ37Ø              MOVSI QJ,WTMASK         ;CLEAR WAIT CODE
ØØØ125  412374  ØØØ12Ø'             ANDCAM QJ,JBTSTS(J)
ØØØ126  4Ø2ØØØ  ØØØ117'             SETZM POTLST            ;CLEAR POTENTIALLY LOST TIME AS A USER IS TO BE RUN
ØØØ127  26314Ø  ØØØØØØ              POPJ PDP,               ;RETURN
ØØØ13Ø  4Ø22ØØ  ØØØØØØ      SCHD1:  SETZ J,                 ;RETURN NULL JOB
ØØØ131  26314Ø  ØØØØØØ              POPJ PDP,



                            ;TEMPORAPY EXPERIMENTAL SCHEDULING CHANGE TO PERMIT TTY-I/O-WAIT-SATISFIED JOBS ON
                            ; THE DISK TO DISPLACE I/O BOUND JOBS IN CORE,... R.CLEMENTS/D.PLUMER 9 MAY 68
ØØØ132  332ØØØ  ØØØ141'     CKJB1Ø: SKIPE   INFLG
ØØØ133  331ØØ4  ØØØØ64'             SKIPL   JBTSWP(J)
ØØØ134  254ØØØ  ØØØØ65'             JRST    CKJB4B
ØØØ135  2Ø124Ø  ØØØ137'             MOVEI   DAT,CKJBT
ØØØ136  254ØØØ  ØØØØ67'             JRST    CKJB4A
ØØØ137  4ØØØØØ  ØØØ2Ø5'     CKJBT:  EXP     EQFIX
ØØØ14Ø  ØØØØØ6  77776Ø              XWD     QQTTY,-PQ2      ;MAKE JOB(LPT) COMPETE WITH CPU BOUND JOBS
ØØØ141  ØØØØØØ  ØØØØØØ      INFLG:  Ø       ;NON-ZERO MEANS AT LEAST ONE JOB ON DISK WAITING TO COME IN.
```

```
                SUBTTL  QCSS R. KRASIN/AF TS3,17 22 MAR 68 V000

                ,THIS ROUTINE MUST BE ASSEMBLED WITH THE CONFIGURATION
                ,TAPE TO DEFINE NUMBER OF JOBS
                ,THIS SECTION CONTAINS 2 ROUTINES FOR Q MANIPULATION
                ,AND NECESSARY TABLES FOR SPECIFING OPERATIONS PERFORMED
                ,BY THEM.

                EXTERNAL IMGIN,JBTSTS,JBTADR,PJBSTS
                INTERNAL QXFER,QSCAN,FTSWAP,FTDISK

                ,STORAGE:
                ,EACH Q IS A RING STRUCTURED, FOWARD AND BACKWARD
                ,LINKED SRING LIST, THE "FIRST" LINK IN A Q IS
                ,A Q-HEADER POINTING TO THE FIRST AND LAST MEMBERS OF THE Q.
                ,A NULL Q HAS ONE LINK--THE Q-HEADER ITSELF.  THE LINKS MAKING
                ,UP THE QS ARE CONTAINED IN A TABLE (JBTQ) WITH NEGATIVE
                ,INDICIES (ADDRESSES LESS THAN JBTQ) USED FOR Q-HEADERS AND
                ,POSITIVE INDICIES USED FOR MEMBERS (JOBS). THUS ONLY ONE WORD
                ,PER LINK IS NECESSARY--ITS ADDRESS RELATIVE TO JBTQ GIVES THE
                ,JOB NO. (OR Q NO. IF NEGATIVE) WHICH IT REPRESENTS WHILE
                ,ITS CONTENTS CONTAINS THE LINKING POINTERS, THESE
                ,POINTERS ARE ALSO INDICIES RELATIVE TO JBTQ RATHER THAN
                ,ABSOLUTE ADDRESSES--RH(LINK)=FOWARD POINTER;
                ,LH(LINK)=BACKWARD POINTER.
                ,A JOB IS ASSUMED TO BE IN NO MORE THAN ONE Q AT A TIME, AND
                ,THE NULL JOB (JOB 0) DOES NOT APPEAR IN THE QS  (I.E, JBTQ
                ,ITSELF IS THE Q-HEADER FOR Q 0).

                ,ROUTINES:
                ,BOTH ROUTINES ARE "TABLE DRIVEN" IN THE SENSE THAT THE
                ,CALLING ROUTINE PROVIDES THE ADDRESS OF A TABLE WHICH
                ,DEFINES THE SPECIFIC OPERATIONS TO BE PERFORMED.
```

```

                                  ;QUEUE INITIALIZATION
                                  ;PUT ALL JOBS IN NULL QUEUE(JOB NO. NOT ASSIGNED)
                                  ;CALLED ON RESTART AT LOC. 143

                                  INTERNAL QINI
                                  EXTERNAL CPOPJ,JOBMAX,MXQUE,JBTQ
                                  EXTERNAL JBTQP1 ;EQUALS JBTQ+1
                                  EXTERNAL JBTQM1 ;EQUALS JBTQ-1
                                  EXTERNAL JBTQMN ;EQUALS JBTQ-NULQ

000142  211040  000000  QINI:     MOVNI TAC,MXQUE           ;MAKE ALL QUEUE HEADERS POINT TO THEMSELVES
000143  504040  000001            HRL TAC,TAC               ;BACKWARD POINTERS TOO
000144  202041  000100'           MOVEM TAC,JBTQ(TAC)
000145  253040  000144'           AOBJN TAC,.-1
000146  201040  777763            MOVEI TAC,-NULQ           ;PUT JOBS ALL IN NULQ QUEUE
000147  206040  000000            MOVSM TAC,JBTQP1          ;BACK POINTER FOR JOB 1
000150  201200  000043'           MOVEI ITEM,JOBMAX         ;MAX. JOB NO.
000151  202044  000144'           MOVEM TAC,JBTQ(ITEM)      ;FOR. POINTER OF JOBMAX JOB NO.

000152  506200  000000            HRLM ITEM,JBTQMN          ;SET NULQ HEADER TO POINT TO JOB1
000153  201040  000001            MOVEI TAC,1               ;AND JOBMAX
000154  542040  000152'           HRRM TAC,JBTQMN ;FORWARD POINTER
000155  542204  000000  QINI1:    HRRM ITEM,JBTQM1(ITEM)    ;JOB I-1 POINT TO JOB I
000156  402000  000151'           SETZM JBTQ
000157  363200  000000            SOJLE ITEM,CPOPJ          ;FINISHED?
000160  506204  000147'           HRLM ITEM,JBTQP1(ITEM)    ;BACK POINTER JOB I+1 POINTS TO JOB I
000161  254000  000155'           JRST QINI1
```

```

,DELETES A JOB FROM ITS "SOURCE-Q", DETERMINES A "DEST-Q"
,ACCORDING TO ONE OF 3 FUNCTIONS, AND INSERTS THE JOB AT
,THE BEGINNING OR END OF THIS DEST-Q. IN ADDITION IT MAY
,RESET THE JOB'S QUANTUM TIME (RH JBTSTS).
,THE DRIVING TABLES ARE "TRANSFER TABLES":
,
,T.TABLE:         EXP <CODE>
,                 XWD <QUANT-TAB>,<Q-TAB>
,
,DEPENDING ON <CODE>, THE SECOND WORD IS EITHER DATA OR THE
,ADDRESSES OF "CORRESPONDANCE TABLES".
,
,THE PREFIX OF <CODE> SPECIFIES WHETHER THE JOB IS TO BE
,INSERTED AT THE BEGINNING OR END OF THE DEST-Q. THE SUFFIX
,DETERMINES THE FUNCTION USED TO SELECT THE DEST-Q.
,THE FOLLOWING ARE THE SIX CODES AND THEIR TABLE FORMATS:


,DEST-Q AS A FIXED (PREDETERMINED) Q:
,BQFIX: INSERT AT BEG OF DEST-Q
,EQFIX: INSERT AT END
,
, THE JOB IS TRANSFERED TO THE END OR BEG. OF THE Q <Q-TAB>
, IF <QUANT-TAB> = -1, QUANT. TIME IS NOT RESET.
, IF <QUANT-TAB> .G. Ø , QUANT. TIME IS RESET TO <QUANT-TAB>.
, SINCE THIS FUNCTION IS FULLY DEFINED BY THE SECOND WORD
, ALONE, NO CORRES. TABLE IS NECESSARY.
```

```
;DEST-Q AS A FUNCTION OF SOURCE-Q
;BQLINK:        INSRT AT BEG OF DEST-Q
;EQLINK:        INSERT AT END
;
; <Q-TAB>=ADDRES OF A CORRES. TABLE "LINKING" SOURCE-QS TO
; DEST-QS.
; IF <QUANT-TAB> = -1, QUANT. TIME IS NOT RESET.
; OTHERWISE <QUANT-TAB> IS TAKEN AS THE ADDRESS OF A
; TABLE OF QUANT. TIMES CORRESPONDING TO THE Q-LINKING TABLE.
; FORMAT OF THE TABLES ARE:
;
; <Q-TAB>:        XWD <SQ1>,<DQ1> ;1ST SOURCE-Q;DEST-Q PAIR
;                 ...
;                 XWD <SQN>,<DQN> ;NTH ...
;                 Z               ;ZERO TERMINATES TABLE
;
; <QUANT-TAB>:  EXP <QUANT1>     ;CORRES. TO <Q-TAB>+0
;                 ...
;               EXP <QUANTN>     ;CORRES. TO <Q-TAB>+N-1

;                 Z
;
; UPON A CALL TO QXFER FOR THESE 2 CODES, AC T2 CONTAINES
; THE SOURCE-Q (CURRENT Q) OF THE JOB.  THE LH OF THE
; <Q-TAB> ENTRIES ARE SEARCHED FOR A MATCH.,IF FOUND, THE
; RH IS TAKEN AS THE DEST-Q AND THE QUANT. TIME IS RESET
; (IF <QUANT-TAB> NOT -1) TO THE CORRRES. ENTRY IN THE
; <QUANT-TAB> TABLE.
; IF NO MATCH FOUND.,NO TRANSFER TAKES PLACE.


;DEST-Q AS A FUNCTION OF JOB SIZE
;BQJSIZ INSERT AT BEG OF DEST-Q
;EQJSIZ INSERT AT END
;
; <Q-TAB>=ADDRESS OF A TABLE ASSOCIATING JOB SIZE
; (IN 1K BLOCKS) TO DEST-QS.
; <QUANT-TAB> HAS SAME MEANING AS FOR B-EQLINK
;
; <Q-TAB>:        XWD <JSIZ1>,<DQ1>
;                 ...
;                 XWD <JSIZN>,<DQN>
;                 Z
;
; <QUANT-TAB>: SIMILAR TO THAT FOR B-EQLINK
;
; THE <JSIZ>'S MUST BE IN INCREASING ORDER.
; THE TABLE IS SEARCHED UNTIL <JSIZ> IS LESS THAN OR
; EQUAL TO THE JOB SIZE, THEN THE CORRES. <DQ> IS
; TAKEN AS THE DEST-Q. IF THE TABLE IS EXAUSTED, NO
; TRANSFER TAKES PLACE.
; QUANT. TIME IS HANDLED AS IN B-EQLINK.
```

```
                                ,CALLING SEQUENCE:
                                ,       MOVE  J,[JOB NUMBER]
                                ,       MOVE T2,[CURRENT Q]      ;BQLINK AND FQLINK ONLY
                                ,       MOVEI TT,TRANS TABLE ADDRESS
                                ,       PUSHJ PDP,QXFER
                                ,       ...               ;RETURN
                                ,  ON RETURN J IS UNALTERED; LH(Q)=-1  IF QUANT. TIME NOT
                                ,  RESET; =QUANT. TIME IF RESET;RH(Q)=DEST.Q
                                ,
                                ,ACS:
                        000025  TT=DAT  ;POINTER TO TRANSFER TABLE
                        000004  J=ITEM  ;JOB NO.
                        000027  Q=PROG  ;DEST-Q AND QUANT. TIME ON RETURN
                        000022  T1=TAC1 ;TEMP
                        000031  T2=TAC  ;TEMP AND SOURCE-Q ON CALL TO B,FQLINK

                                EXTERNAL ERROR

    000162  200345  000001  QXFER:  MOVE Q,1(TT)               ;GET TRANSFER TABLE ADDRESS
    000163  254025  000000          JRST @(TT)                 ;DISPATCH

                                ,DEST-Q AS FUNCTION OF SOURCE-Q
    000164  336107  000000  QLINK:  SKIPN T1,(Q)               ;END OF TABLE?
    000165  263140  000000          POPJ PDP,                  ;YES
    000166  574100  000022          HLRE T1,T1
    000167  312100  000001          CAME T1,T2                 ;NO--SOURCE-Q=LH(TABLE ENTRY)?
    000170  252340  000164'         AOBJP Q,QLINK              ;NO- CONTINUE SEARCH
    000171  254000  000201'         JRST QX2                   ;YES

                                ,DEST-Q AS FUNCTION OF JOB SIZE
    000172  554044  000000  QJSIZ:  HLRZ T2,JBTADR(J)          ;HIGHEST REL. LOC. OF JOB
    000173  240040  777766          ASH T2,-↑D10               ;CONVERT TO NO. OF 1K BLOCKS - 1.
    000174  205041  000001          MOVSI T2,1(T2)             ;NO. OF 1K BLOCKS TO LH.
    000175  336107  000000  QX1:    SKIPN T1,(Q)               ;END OF TABLE?
    000176  265240  000000          JSP DAT,ERROR
    000177  313040  000002          CAMLE T2,T1                ;JOBSIZE .LE. LH(TABLE ENTRY)?
    000200  252340  000175'         AOBJP Q,QX1                ;NO--CONTINUE SEARCH, JUMP ALWAYS.

    000201  204040  000007  QX2:    MOVS T2,Q                  ;T2 IS ADDR. OF QUANT.TIME(IF REQUESTED)
    000202  560347  000000          HRRO Q,(Q)                 ;RH(Q)=DEST-Q;LH=-1(NO QUANT.TIME REQ.)
    000203  331005  000001          SKIPL 1(TT)                ;WAS QUANT. TIME REQUESTED?
    000204  504341  000000          HRL Q,(T2)                 ;YES--GET IT
```

```

                               ,FIXED DEST-Q
ØØØ2Ø5  2ØØ1Ø4  ØØØ156' QFIX:   MOVE T1,JBTQ(J)        ;DELETE JOB FROM SOURCE-Q
ØØØ2Ø6  2Ø4Ø4Ø  ØØØØØ2          MOVS T2,T1             ;T1=FORW. LINK, T2=BACK LINK
ØØØ2Ø7  5421Ø1  ØØØ2Ø5'         HRRM T1,JBTQ(T2)       ;FORW. LINK PAST JOB
ØØØ21Ø  5Ø6Ø42  ØØØ2Ø7'         HRLM T2,JBTQ(T1)       ;BACK LINK PAST JOB

ØØØ211  335Ø25  ØØØØØØ          SKIPGE (TT)            ;END OR BEG. OF Q?
ØØØ212  544347  ØØØ21Ø'         HLR Q,JBTQ(Q)          ;END--THIS WILL LEAVE Q=IDX OF
                                                       ; CURRENT LAST LINK;T2=IDX OF Q-HEADER
ØØØ213  2ØØØ47  ØØØ212'         MOVE T2,JBTQ(Q)        ;BEG--T2=IDX OF CURRENT 1ST LINK
                                                       ; Q=IDX OF Q-HEADER
ØØØ214  5422Ø7  ØØØ213'         HRRM J,JBTQ(Q)         ;INSERT JOB IN DEST-Q
ØØØ215  5Ø62Ø1  ØØØ214'         HRLM J,JBTQ(T2)
ØØØ216  542Ø44  ØØØ215'         HRRM T2,JBTQ(J)
ØØØ217  5Ø6344  ØØØ216'         HRLM Q,JBTQ(J)

ØØØ22Ø  32134Ø  ØØØ224'         JUMPL Q,QX3            ;RETURN IF QUANT. TIME NOT REQ.
ØØØ221  546344  ØØØ125'         HLRM Q,JBTSTS(J)       ;SET QUANT. TIME

ØØØ222  20124Ø  ØØØØØØ          MOVEI TT,RNQ           ;SET JOB STATUS WAIT
ØØØ223  13724Ø  ØØØØ6Ø'         DPB TT,PJBSTS          ;CODE TO RUN QUEUE (Ø).
ØØØ224  26314Ø  ØØØØØØ  QX3:    POPJ PDP,

                ØØØ2Ø5' BQFIX=QFIX
        4ØØØØØ  ØØØ2Ø5' EQFIX=QFIX+1BØ
                ØØØ164' BQLINK=QLINK
        4ØØØØØ  ØØØ164' EQLINK=QLINK+1BØ
                ØØØ172' BQJSIZ=QJSIZ
        4ØØØØØ  ØØØ172' EQJSIZ=QJSIZ+1BØ
        ,SCANS THE QS RETURNINC THE NUMBFRS OF THE JOBS IN THE QS.
        ,THE ORDER AND MANNER IN WHICH THE QS ARE SEARCHED IS
        ,DETERMINED BY A "SCAN TABLE" ADDRESSED IN THE CALLING SEQ.
        ,THE SCAN TARLE HAS THE FORM:
        ,
        ,SCANTAB:       XWD <Q1>,<CODE1>        ;SCN Q1 ACCRDING TO CODE1
        ,               ...
        ,               XWD <QN>,<CODEN>        ;QN ACCORDING TO CODEN
        ,               Z                ;ZERO TERMINATES TABLE
        ,
        ,EACH Q MAY RE SCANNED IN ONF OF FOUR WAYS SPECIFIEDRY <CODE>
        ,THE CODFS ARE:
        ,
        ,QFOR   SCAN WHOLE Q FOWARD
        ,QFOR1  SCAN FOR ONY THE 1ST MEMBER (IF ANY)
        ,QBAK   SCAN WHOLE Q BACKWARD
        ,QBAK1  SCAN BACKWARD FOR ALL MEMBERS EXCEPT THE 1ST
        ,
        ,CALLING SEQ.
        ,
        ,       MOVEI ST,SCAN TABLE ADDRESS
        ,       JSP PC,QSCAN    ;SET UP PC FOR REPEATED RETURNS
        ,       ...             ;RETURN HERE WHEN NO MORE JOBS
        ,       ...             ;RETURN HERE WITH NEXT JOB IN AC J
        ,                       ; AND ITS Q IN LH(QR)
        ,
        ,       PERFORM ANY NECESSARY TESTING OF THIS JOB
        ,       J,ST,PC,QR MUST BE PRESERVED
        ,
        ,       JRST (QR)       ;RETURN TO QSCAN TO GET NEXT JOB
        ,                       ; IF THIS ONE NOT ACCEPTABLE
        ,
        ,ACS:
ØØØØØ4  J=ITEM  ;JOB NO.
ØØØØØ5  ST=DAT  ;POINTER TO SCAN TABLE
ØØØØØ1  PC=TAC  ;RETURN ADDRESS
ØØØØØ2  QR=TAC1 ;ITERATED RETURN ADDRESS TO QSCAN
```

```
ØØØ225  3361Ø5  ØØØØØØ  QSCAN:  SKIPN QR,(ST)    ;END OF SCAN TABLE?
ØØØ226  254ØØ1  ØØØØØØ          JRST (PC)        ;YES--RETURN TO CALL+1
ØØØ227  5742ØØ  ØØØØØ2          HLRE J,QR        ;NO--GET NO. OF Q
ØØØ23Ø  254ØØ2  ØØØØØØ          JRST (QR)        ;DISPATCH

ØØØ231  2Ø11ØØ  ØØØ234' QFOR1:  MOVEI QR,QFOR2   ;ONLY THE FIRST JOB

ØØØ232  57Ø2Ø4  ØØØ217' QFOR:   HRRE J,JBTQ(J)   ;SCAN FOWARD ALL JOBS
ØØØ233  3272Ø1  ØØØØØ1          JUMPG J,1(PC)    ;RETURN THIS JOB NO. CALL+2 UNLESS--
ØØØ234  34424Ø  ØØØ225' QFOR2:  AOJA ST,QSCAN    ;END OF THIS Q--GET NEXT Q

ØØØ235  5742Ø4  ØØØ232' QBAK1:  HLRE J,JBTQ(J)   ;SCAN BACKWARD ALL JOBS EXCEPT 1ST
ØØØ236  333ØØ4  ØØØ235'         SKIPLE JBTQ(J)   ;IS THIS THE FIRST MEMBER?
ØØØ237  254ØØ1  ØØØØØ1          JRST 1(PC)       ;NO--RETURN CALL+2
ØØØ24Ø  34424Ø  ØØØ225'         AOJA ST,QSCAN    ;YES--GET NEXT Q

ØØØ241  5742Ø4  ØØØ236' QBAK:   HLRE J,JBTQ(J)   ;SCAN BACKWARD ALL JOBS
ØØØ242  3272Ø1  ØØØØØ1          JUMPG J,1(PC)    ;RETURN CALL+2 WITH JOB NO. UNLESS
ØØØ243  34424Ø  ØØØ225'         AOJA ST,QSCAN    ;BEG OF THIS Q--GET NEXT Q
```

```
                        INTERNAL FTCHECK,FTMONP
                        IFN FTCHECK+FTMONP,<
                        EXTERNAL AVALTB
                        DEFINE X(A,B),<
                        EXTERNAL A'AVAL
                        INTERNAL A'Q
                        A'Q=ZZ
                        ZZ=ZZ+1
                        >
                                ZZ=Ø
                                QUEUES
                                LOC=ZZ
                        >
                        IFE FTCHECK+FTMONP,<

                        ;SHARABLE DEVICE JUST BECOME AVAILABLE(EXTENDED TO OTHER QUEUEW TOO)
                        ;APPROPRIATE ENTRY IS SET NON-ZERO WHEN SCHEDULER SHOULD LOOK
                        ;AT THAT QUEUE TO FIND A JOB TO RUN
                        ;WSAVAL CONTAINS THE NO. OF JOBS WITH IO WAIT SATISFIED(Ø=NONE)

                        DEFINE X(A,B)
                        <INTERNAL A'AVAL,A'Q
                        A'Q=.-AVALTB
                        A'AVAL: Ø
                        >

                        INTERNAL AVALTB

ØØØ244                  AVALTB: QUEUES   ;GENERATE THE AVAL FLAGS
ØØØ244  ØØØØØØ  ØØØØØØ  RNAVAL: Ø
ØØØ245  ØØØØØØ  ØØØØØØ  WSAVAL: Ø
ØØØ246  ØØØØØØ  ØØØØØØ  TSAVAL: Ø
ØØØ247  ØØØØØØ  ØØØØØØ  STAVAL: Ø
ØØØ25Ø  ØØØØØØ  ØØØØØØ  AUAVAL: Ø
ØØØ251  ØØØØØØ  ØØØØØØ  MQAVAL: Ø
ØØØ252  ØØØØØØ  ØØØØØØ  DAAVAL: Ø
ØØØ253  ØØØØØØ  ØØØØØØ  DTAVAL: Ø
ØØØ254  ØØØØØØ  ØØØØØØ  DCAVAL: Ø
ØØØ255  ØØØØØØ  ØØØØØØ  MTAVAL: Ø
                ØØØØ12  LOC=.-AVALTB
                        >
                ØØØØ12  NQUEUE=LOC              ;NO. OF QUEUES COUNTING RUN QUEUE
                        XP MAXQ,NQUEUE-1        ;MAX. STATE CODE WHICH HAS AN AVAL FLAG
                        XP MINQ,STQ             ;MINIMUM SHARABLE DEVICE QUEUE
                        XP AVLNUM,MAXQ          ;MAX. STATE CODE WHICH HAS AN AVAL FLAG

                        ;DEFINE STATE CODES WHICH DO NOT HAVE AVAL AND REQ FLAGS


                        DEFINE X(A)
                        <INTERNAL A'Q
                        A'Q=LOC
                        LOC=LOC+1
```

```
                 >
                           CODES↑  X IOW,↑INTERNAL IOWQ

                 XP MXCODE,LOC-1 ;MAX. JOB STATE CODE
         ØØØØ17  PQ1=LOC
         ØØØØ2Ø  LOC=LOC+1
         ØØØØ2Ø  PQ2=LOC
         ØØØØ21  LOC=LOC+1
         ØØØØ21  PQ3=LOC
         ØØØØ22  LOC=LOC+1
         ØØØØ22  CMQ=LOC              ;COMMAND DELAY QUEUE
```

```

                        ;CORRESPONDENCE TABLE BETWEEN JOB STATUS CODES AND QUEUE TRANSFER TABLES
                        ;USED BY SCHEDULER
                        ;RUNCSS SETS JOB STATUS WORD TO NEW STATE CODE.
                        ;SCHEDULER SETS UP QUEUE TRANSFER TABLE ADDRESS FROM
                        ;FOLLOWING TABLE USING NEW STATE CODE AS INDEX

                        DEFINE X(A,B)
                        <       EXP Q'A'W
                        >

                        INTERNAL QBITS

ØØØ256  ØØØØØØ  ØØØ332' QBITS:  QUEUES↑ X RN,7  ↑       EXP QRNW
ØØØ257  ØØØØØØ  ØØØ334'         X WS,6  ↑       EXP QWSW
ØØØ260  ØØØØØØ  ØØØ336'         X TS,6  ↑       EXP QTSW
ØØØ261  ØØØØØØ  ØØØ354'         X ST,6  ↑       EXP QSTW
ØØØ262  ØØØØØØ  ØØØ344'                 X AU,4  ↑       EXP QAUW
ØØØ263  ØØØØØØ  ØØØ346'                 X MQ,4  ↑       EXP QMQW
ØØØ264  ØØØØØØ  ØØØ350'                 X DA,4  ↑       EXP QDAW
ØØØ265  ØØØØØØ  ØØØ356'         X DT,4  ↑       EXP QDTW
ØØØ266  ØØØØØØ  ØØØ352'         X DC,4  ↑       EXP QDCW
ØØØ267  ØØØØØØ  ØØØ360'         X MT,4  ↑       EXP QMTW
ØØØ270  ØØØØØØ  ØØØ340'         CODES↑  X IOW,  ↑       EXP QIOWW
ØØØ271  ØØØØØØ  ØØØ342'         X TIOW, ↑       EXP QTIOWW
ØØØ272  ØØØØØØ  ØØØ362'         X SLP,  ↑       EXP QSLPW
ØØØ273  ØØØØØØ  ØØØ324'         X NUL,  ↑       EXP QNULW
ØØØ274  ØØØØØØ  ØØØ326'         X STOP, ↑       EXP QSTOPW
```

```
                                IFN FTCHECK+FTMONP,<
                                DEFINE X(A,B),<
                                EXTERNAL A'REQ
                                >
                                        QUEUES
                                EXTERNAL REQTAB
                                >
                                IFE FTCHECK+FTMONP,<

                                ;SHARABLE DEVICE REQUEST TABLE(GENERALIZED FOR OTHER QUEUES TOO)
                                ;CONTAINS THE NUMBER OF JOB WAITING TO USE SHARBLE DEVICE
                                ;WSREQ AND RNREQ ARE UNUSED

                                DEFINE X(A,B)
                                <A'REQ: Ø
                                INTERNAL A'REQ
                                >

                                INTERNAL REQTAB


ØØØ275                          REQTAB: QUEUES  ;GENERATE REQ TABLE
ØØØ275  ØØØØØØ  ØØØØØØ                  X RN,7  +RNREQ: Ø
ØØØ276  ØØØØØØ  ØØØØØØ                  X WS,6  +WSREQ: Ø
ØØØ277  ØØØØØØ  ØØØØØØ                  X TS,6  +TSREQ: Ø
ØØØ3ØØ  ØØØØØØ  ØØØØØØ                  X ST,6  +STREQ: Ø
ØØØ3Ø1  ØØØØØØ  ØØØØØØ                          X AU,4  +AUREQ: Ø
ØØØ3Ø2  ØØØØØØ  ØØØØØØ                          X MQ,4  +MQREQ: Ø
ØØØ3Ø3  ØØØØØØ  ØØØØØØ                          X DA,4  +DAREQ: Ø
ØØØ3Ø4  ØØØØØØ  ØØØØØØ                  X DT,4  +DTREQ: Ø
ØØØ3Ø5  ØØØØØØ  ØØØØØØ                  X DC,4  +DCREQ: Ø
ØØØ3Ø6  ØØØØØØ  ØØØØØØ                  X MT,4  +MTREQ: Ø
                                >
```

```

                                 ;CORRESPONDENCE TABLE LH=QUEUE CODE, RH=QUEUE TRANSFER TABLE ADR.
                                 ;INDEX INTO TABLE ALSO = QUEUE CODE
                                 ;FOR SHARABLE DEVICES ONLY
                                 ;SCHEDULER TAKES ONE JOB WAITING FOR A SHARABLE DEVICE AND
                                 ;PUTS IT IN THE APPROPRIATE RUN QUEUE ACCORDING TO
                                 ;QUEUE TRANSFER TABLE AS SPECIFIED BELOW BY THE JOB WAIT
                                 ;STATE CODE.

                                 DEFINE X(A,B)
                                 <       XWD -A'Q,Q'A'S
                                 >

                         ØØØØØØ  QRNS=Ø  ;NO CORRESPONDENCE TABLES FO THESE QUEUES
                         ØØØØØØ  QWSS=Ø
                         ØØØØØØ  QTSS=Ø

                                 INTERNAL AVLQTB

ØØØ3Ø7   ØØØØØØ   ØØØØØØ  AVLQTB: QUEUES+ X RN,7  +        XWD -RNQ,QRNS
ØØØ31Ø   777777   ØØØØØØ          X WS,6  +        XWD -WSQ,QWSS
ØØØ311   777776   ØØØØØØ          X TS,6  +        XWD -TSQ,QTSS
ØØØ312   777775   ØØØ372'         X ST,6  +        XWD -STQ,QSTS
ØØØ313   777774   ØØØ4ØØ'                 X AU,4  +        XWD -AUQ,QAUS
ØØØ314   777773   ØØØ364'                 X MQ,4  +        XWD -MQQ,QMQS
ØØØ315   777772   ØØØ366'                 X DA,4  +        XWD -DAQ,QDAS
ØØØ316   777771   ØØØ374'         X DT,4  +        XWD -DTQ,QDTS
ØØØ317   77777Ø   ØØØ37Ø'         X DC,4  +        XWD -DCQ,QDCS
ØØØ32Ø   777767   ØØØ376'         X MT,4  +        XWD -MTQ,QMTS
```

```
                        IFN FTCHECK+FTMONP,<
                        EXTERNAL QJOB,JOBQUE    ;JOBQUE WILL CAUSE LOAD OF PROPER SCHDAT
                                                ; DEPENDING ON FTRC1Ø IN SCHDAT
                        IFN FTSWAP,<
                        EXTERNAL XJOB
                        >>
                        IFE FTCHECK+FTMONP,<


                        INTERNAL JOBQUE
ØØØ321  ØØØØØØ  ØØØØØØ  JOBQUE: Ø       ;JOBS TO BE REQUEUED ON CLOCK INTERRUPT

                        INTERNAL QJOB
ØØØ322  ØØØØØØ  ØØØØØØ  QJOB:   Ø       ;NUMBER OF JOBS NEEDING Q TRANSFERS AT OTHER THAN CLOCK LEVEL

                        IFN FTSWAP,<
ØØØ323  ØØØØØØ  ØØØØØØ  XJOB:   Ø       ;NUMBER OF JOBS NEEDING CORE EXPANSION BY SWAPOUT-IN
                        INTERNAL XJOB
                        >

                        >
                        INTERNAL QSTOP,QTIME,SSCAN,QCMW

                ØØØ2Ø5' BQFIX=QFIX       ;BEGINNING OF QUEUES FIXED QUEUE DISCIPLINE
        4ØØØØØ  ØØØ2Ø5' EQFIX=QFIX+1BØ   ;END OF QUEUFS " " "
                ØØØ164' BQLINK=QLINK
        4ØØØØØ  ØØØ164' EQLINK=QLINK+1BØ
                ØØØ172' BQJSIZ=QJSIZ
        4ØØØØØ  ØØØ172' EQJSIZ=QJSIZ+1BØ
                        DEFINE TTAB(FCTN,QUEUE,QUANT)
                        <               EXP FCTN
                                        XWD QUANT,-QUEUE
                        >
                        DEFINE PTTAB(FCTN,QUEUE,QUANT)
                        <               EXP FCTN
                                        XWD QUANT,QUEUE
                        >

ØØØ324                  QNULW:  TTAB EQFIX,NULQ,-1      ;NULL QUEUE JOB NO. NOT ASSIGNED
ØØØ324  4ØØØØØ  ØØØ2Ø5'                 EXP EQFIX
ØØØ325  777777  777763                  XWD -1  ,-NULQ
ØØØ326                  QSTOP:QSTOPW:   TTAB EQFIX,STOPQ,-1     ;UNRUNABLE JOBS TO END OF STOPQ
ØØØ326  4ØØØØØ  ØØØ2Ø5'                 EXP EQFIX
ØØØ327  777777  777762                  XWD -1  ,-STOPQ
ØØØ33Ø                  QCMW:   TTAB EQFIX,CMQ,-1               ;COMMAND WAIT TILL JOB IN CORE
ØØØ33Ø  4ØØØØØ  ØØØ2Ø5'                 EXP EQFIX
ØØØ331  777777  777756                  XWD -1          ,-CMQ
ØØØ332                  QRNW:   PTTAB EQJSIZ,QSTAB,QQSTAB       ;JUST RUNABLE JOBS
ØØØ332  4ØØØØØ  ØØØ172'                 EXP EQJSIZ
ØØØ333  ØØØ414' ØØØ4Ø4'                 XWD QQSTAB      ,QSTAB
                                ;WHICH ARE NOT IN SOME WAIT STATE BELOW,ENTER PROCESSOR
                                ;QS AT END AND GET QUANT. TIME ACCORDING TO THEIR SIZE

ØØØ334                  QWSW:   TTAB BQFIX,PQ1,QQTTY    ;IO WAIT SAT.(EXCEPT TTY)
ØØØ334  ØØØØØØ  ØØØ2Ø5'                 EXP BQFIX
ØØØ335  ØØØØ16  777761                  XWD QQTTY       ,-PQ1
                                ;ENTER FRONT OF PROCFSSOR QS AND GET QUANT. TIME
                                ;ACCORDING TO JOB SIZE
ØØØ336                  QTSW:   TTAB BQFIX,PQ1,QQTTY    ;TTY IO WAIT SATISFIED(ENTER FRONT OF PQ1)
ØØØ336  ØØØØØØ  ØØØ2Ø5'                 EXP BQFIX
ØØØ337  ØØØØ16  777761                  XWD QQTTY       ,-PQ1

ØØØ34Ø                  QIOWW:  TTAB EQFIX,IOWQ,-1      ;IOW(EXDEPT TTY) HELD IN IOWQ
ØØØ34Ø  4ØØØØØ  ØØØ2Ø5'                 EXP EQFIX
ØØØ341  777777  777766                  XWD -1  ,-IOWQ
ØØØ342                  QTIOWW: TTAB EQFIX,TIOWQ,-1     ;TTY IOW HELD IN TIOWQ
ØØØ342  4ØØØØØ  ØØØ2Ø5'                 EXP EQFIX
ØØØ343  777777  777765                  XWD -1  ,-TIOWQ
ØØØ344  4ØØØØØ  ØØØ2Ø5' QAUW:   TTAB EQFIX,AUQ,-1,+             EXP EQFIX
ØØØ345  777777  777774                  XWD -1,-AUQ
ØØØ346                  QMQW:   TTAB EQFIX,MQQ,-1       ;MON. Q(DISK) WAIT
ØØØ346  4ØØØØØ  ØØØ2Ø5'                 EXP EQFIX
ØØØ347  777777  777773                  XWD -1  ,-MQQ
ØØØ35Ø                  QDAW:   TTAB EQFIX,DAQ,-1       ;DEV. ALLOC.(DISK)
```

```
000350  400000  000205'                  EXP EQFIX
000351  777777  777772                   XWD -1  ,-DAQ
000352                  QDCW:   TTAB EQFIX,DCQ,-1        ;DATA CONTROL WAIT
000352  400000  000205'                  EXP EQFIX
000353  777777  777770                   XWD -1  ,-DCQ
000354                  QSTW:   TTAB EQFIX,STQ,-1        ;SYST TAPE
000354  400000  000205'                  EXP EQFIX
000355  777777  777775                   XWD -1  ,-STQ
000356                  QDTW:   TTAB EQFIX,DTQ,-1        ;DEC TAPE
000356  400000  000205'                  EXP EQFIX
000357  777777  777771                   XWD -1  ,-DTQ
000360                  QMTW:   TTAB EQFIX,MTQ,-1        ;MAG TAPE
000360  400000  000205'                  EXP EQFIX
000361  777777  777767                   XWD -1  ,-MTQ
000362                  QSLPW:  TTAB EQFIX,SLPQ,-1       ;SLEEP UUO
000362  400000  000205'                  EXP EQFIX
000363  777777  777764                   XWD -1  ,-SLPQ
```

```
                                    ;TRANSLATION TABLE FROM WAIT STATE TO SATISFIED STATE
                                    ;DO NOT RESET QUANTUM RUN TIME

                 777777  777777     QQSD=-1

        ØØØ364                      QMQS:    TTAB BQFIX,PQ1,QQSD       ;START MON. Q(DISK) AT PQ1
        ØØØ364   ØØØØØØ  ØØØ2Ø5'                     EXP BQFIX
        ØØØ365   777777  777761                      XWD QQSD          ,-PQ1
        ØØØ366                      QDAS:    TTAB BQFIX,PQ1,QQSD       ;DEV. ALLOC.(DISK)...
        ØØØ366   ØØØØØØ  ØØØ2Ø5'                     EXP BQFIX
        ØØØ367   777777  777761                      XWD QQSD          ,-PQ1
        ØØØ37Ø                      QDCS:    TTAB BQFIX,PQ1,QQSD       ;DATA CONTROL...
        ØØØ37Ø   ØØØØØØ  ØØØ2Ø5'                     EXP BQFIX
        ØØØ371   777777  777761                      XWD QQSD          ,-PQ1
        ØØØ372                      QSTS:    TTAB BQFIX,PQ1,QQSD       ;SYST TAPE
        ØØØ372   ØØØØØØ  ØØØ2Ø5'                     EXP BQFIX
        ØØØ373   777777  777761                      XWD QQSD          ,-PQ1
        ØØØ374                      QDTS:    TTAB BQFIX.PQ1,QQSD       ;DEC TAPE
        ØØØ374   ØØØØØØ  ØØØ2Ø5'                     FXP BQFIX
        ØØØ375   777777  777761                      XWD QQSD          ,-PQ1
        ØØØ376                      QMTS:    TTAB BQFIX,PQ1,QQSD       ;MAG TAPE
        ØØØ376   ØØØØØØ  ØØØ2Ø5'                     EXP BQFIX
        ØØØ377   777777  777761                      XWD QQSD          ,-PQ1
        ØØØ4ØØ                      QAUS:    TTAB BQFIX,PQ1,QQSD       ;ALTER UFD
        ØØØ4ØØ   ØØØØØØ  ØØØ2Ø5'                     EXP BQFIX
        ØØØ4Ø1   777777  777761                      XWD QQSD          ,-PQ1
        ØØØ4Ø2                      QTIME:   PTTAB EQLINK,QTTAB,QQSTAB        ;MOVE JOB TO LOWER Q
        ØØØ4Ø2   4ØØØØØ  ØØØ164'                     EXP EQLINK
        ØØØ4Ø3   ØØØ414' ØØØ41Ø'                     XWD QQSTAB      ,QTTAB
                                             ;WHEN QUANT. TIME EXCEEDED AND RESET QUANT. TIME
```

```

                                ;ENTER PROCESSOR QS ACCORDING TO JOB SIZE
ØØØ4Ø4  ØØØØØ4  777761  QSTAB:  XWD 4,-PQ1       ;PQ1 IF     SIZE .LE. 4K
ØØØ4Ø5  ØØØØ2Ø  77776Ø          XWD ↑D16,-PQ2    ;PQ2 IF  4K .L. SIZE .LE. 16K
ØØØ4Ø6  ØØØ4ØØ  777757          XWD ↑D256,-PQ3   ;PQ3 IF 16 .L. SIZE
ØØØ4Ø7  ØØØØØØ  ØØØØØØ          Z

                                ;PUT JOB DOWN A Q IF EXCEEDS QUANT. TIME
ØØØ41Ø  777761  77776Ø  QTTAB:  XWD -PQ1,-PQ2
ØØØ411  77776Ø  777757          XWD -PQ2,-PQ3
ØØØ412  777757  77776Ø          XWD -PQ3,-PQ2            ;BACK TO PQ2 TO COMPETE WITH IOWS JOBS
ØØØ413  ØØØØØØ  ØØØØØØ          Z

                                ;QUANTUM TABLES

                        ØØØØØ6  QQSD=6  ;TENTH SEC. INITIAL QUANT. FOR SHAR. DEV. WAITERS
                        ØØØØØ6  QQTTY=6 ;TENTH SEC. INITIAL QUANT. FOR TTY IOWS

                                ; QUANT. TIMES ACCORDING TO PROCESSOR Q:

                                INTERNAL RNQUNT

ØØØ414                          RNQUNT:
ØØØ414  ØØØØØØ  ØØØ236  QQSTAB: FXP ↑D3Ø         ;PQ1: ONE HALF SECOND
ØØØ415  ØØØØØØ  ØØØ17Ø          FXP 2*↑D6Ø       ;PQ2: TWO SECONDS
ØØØ416  ØØØØØØ  ØØØ17Ø          FXP 2*↑D6Ø       ;PQ3: TWO SECONDS
ØØØ417  ØØØØØØ  ØØØØØØ          Z
```

```
                                 IFN FTSWAP,<
                                 INTERNAL ISCAN,OSCAN
000420                           ISCAN:  ;SCAN FOR INPUT
000420  777756  000232'                  XWD -CMQ,QFOR   ;MONITOR COMMAND WHICH NEEDS CORE IMAGE IN CORE
000421  777773  000231'                  XWD -MQQ,QFOR1  ;LOOK FOR 1ST JOBS IN SHAR. DEV QUEUES
000422  777772  000231'                  XWD -DAQ,QFOR1
000423  777774  000231'                  XWD -AUQ,QFOR1
000424  777770  000231'                  XWD -DCQ,QFOR1
000425  777767  000231'                  XWD -MTQ,QFOR1
000426  777775  000231'                  XWD -STQ,QFOR1
000427  777771  000231'                  XWD -DTQ,QFOR1
000430  777761  000232' SSCAN:   XWD -PQ1,QFOR   ;SCAN PROCESSOR AS SCHEDULER DOES
000431  777760  000232'                  XWD -PQ2,QFOR
000432  777757  000232'                  XWD -PQ3,QFOR
000433  000000  000000                   Z               ;PATCH SPACE
000434  000000  000000                   Z
000435  000000  000000                   Z               ;FINAL ZERO TO FLAG END

000436                           OSCAN:  ;SCAN FOR OUTPUT
000436  777762  000232'                  XWD -STOPQ,QFOR ;UNRUNABLE JOBS FIRST
000437  777764  000232'                  XWD     -SLPQ,QFOR
000440  777771  000235'                  XWD -DTQ,QBAK1  ;ANY SHAR. DEV.WAITERS MORE THAN 1 DEEP
000441  777775  000235'                  XWD -STQ,QBAK1
000442  777767  000235'                  XWD -MTQ,QBAK1
000443  777770  000235'                  XWD -DCQ,QBAK1
000444  777774  000235'                  XWD -AUQ,QBAK1
000445  777772  000235'                  XWD -DAQ,QBAK1
000446  777773  000235'                  XWD -MQQ,QBAK1
000447  777765  000232'                  XWD -TIOWQ,QFOR         ;TTY IOW
000450  777757  000241'                  XWD -PQ3,QBAK
000451  777771  000231'                  XWD -DTQ,QFOR1
000452  777775  000231'                  XWD -STQ,QFOR1
000453  777767  000231'                  XWD -MTQ,QFOR1          ;NOW SCAN FIRST JOB IN QUEUES
000454  777770  000231'                  XWD -DCQ,QFOR1
000455  777774  000231'                  XWD -AUQ,QFOR1
000456  777772  000231'                  XWD -DAQ,QFOR1
000457  777773  000231'                  XWD -MQQ,QFOR1
000460  777760  000241'                  XWD -PQ2,QBAK
000461  777761  000241'                  XWD -PQ1,QBAK
000462  000000  000000                   Z                       ;PATCH SPACE
000463  000000  000000                   Z
000464  000000  000000                   Z                       ;FINAL ZERO TO FLAG END
                                 >
```

```
                SUBTTL  SWAP R. KRASIN/AF TS4.34  Ø3 FEB 69  V4Ø6

                ,SWAPPER CALLED EVERY CLOCK TIC.
                ,SINCE MOST OPERATIONS STARTED BY THE SWAPPER REQUIRE SEVERAL
                ,TICS TO RUN TO COMPLETION, SEVERAL FLAGS(FINISH,FIT,FORCE
                ;ARE USED TO "REMEMBER" PREVIOUS STATES.
                ,THE BASIC ALGORITHM:
                ;IS CORE SHUFFLER WAITING FOR IO TO FINISH FOR SOME JOB?
                ;  YES--TRY AGAIN TO SHUFFLE(WHEN IO STOPS)
                ;IS CORE SHUFFLER STILL WAITING FOR IO TO FINISH?
                ;  YES--RETURN AND DO NOTHING
                ;IS SWAPPER STILL BUSY?
                ;  YES--RETURN AND DO NOTHING
                ,SCAN QS FOR 1ST JOB OUT OF CORE.
                , IF NONE--RETURN
                ;A:
                , IF ONE--WILL LOW(HIGH) SEG FIT IN LARGEST HOLE IN CORE?
                ,  YES--START INPUT AND RETURN
                ,  NO--IS TOTAL FREE CORE(CORTAL) ENOUGH TO ACCOMMODATE LOW(HIGH) SEG?

                ;     YES--CALL CORE SHUFFLER
                ;       IS SHUFFLER WAITING FOR IO TO STOP?
                ;         YES--RETURN AND DO NOTHING
                ;         NO--GO TO A:
                ,   NO--"REMBER" THIS JOB FOR INPUT AND LOOK FOR OUTPUT:
                ,ANY JOBS WAITING TO XPAND CORE BY SWAP OUT/IN?
                , YES--OUTPUT ONE AND RETURN
                , NO--SCAN QS BACKWARD FOR JOB IN CORE WHOSE PROTECT TIME
                ,                (SET ON INPUT) HAS GONE TO Ø.
                ,  IF NONE--RETURN
                ,  IF ONE--IS IT SWAPPABLE(NO ACTIVE IO AND NOT CURRENT JOB)?
                ,   YES--OUTPUT HIGH SEG(IF ANY AND NOT ON DISK) THEN LOW SEGMENT
                ,   NO--SET SWP BIT(SO SCHEDULER WILL NOT RUN), IO WILL CONTINUE
                ,       IN LOW SEGMENT AS LONG AS IT CAN
                ,       IO ROUTINES NO LONGER STOP IF SWP SET, JUST SHF)

                EXTERNAL JBTSTS
                EXTERNAL BIGHOL,CORTAL,ANYDEV,JBTADR,JBTSWP,KCORE1,TRYSWP
                EXTERNAL IMGOUT,IMGIN,FINISH,FIT,FORCE
                EXTERNAL OERROR,CORGET,JBTDAT,JOBDPG,JOBDPD,JOBPC
                EXTERNAL JBTDAT,SHFWAT,CHKSHF
                EXTERNAL FULCNT,ERRPNT,EXCALP,PCSTOP,PCORSZ,VIRTAL

                INTERNAL SWAP
                INTERNAL XPAND,FT2REL

        ØØØØØ6  T=DEVDAT
        ØØØØØ2  T1=TAC1
        ØØØØØ1  T2=TAC
        ØØØØØ4  J=ITEM

                ,ALL DEVICE DEPENDENT CODE MARKED WITH A "*"
```

```
ØØØ465  332ØØØ  ØØØØØØ  SWAP:   SKIPE SHFWAT            ;IS CORE SHUFFLER WAITING FOR IO TO STOP
                                                        ; FOR SOME JOB?
ØØØ466  26Ø14Ø  ØØØØØØ          PUSHJ PDP,CHKSHF        ;YES, CALL CORE SHUFFLER TO SEE IF
                                                        ; IO STOPPED YET
ØØØ467  336ØØØ  ØØØ465'         SKIPN SHFWAT            ;IS SHUFFLER STILL WAITING?
ØØØ47Ø  332ØØØ  ØØ134Ø'         SKIPE SQREQ             ;*NO--IS SWAP SERV. ROUT. STILL BUSY WITH LAST JOB?
ØØØ471  26314Ø  ØØØØØØ          POPJ PDP,               ;*YES--RETURN
ØØØ472  402ØØØ  ØØØ141'         SETZM   INFLG           ;*** EXPERIMENTAL ***
ØØØ473  3362ØØ  ØØØØØØ          SKIPN J,FINISH          ;NO--ANY IN/OUTPUT TO FINISH?
ØØØ474  254ØØØ  ØØØ547'         JRST SWP2               ;NO-
ØØØ475  3212ØØ  ØØØ537'         JUMPL J,FINOUT          ;YES--INPUT OR OUTPUT?
ØØØ476  332ØØØ  ØØ1342'         SKIPE SERA              ;INPUT, ANY INPUT ERRORS?
ØØØ477  254ØØØ  ØØØ526'         JRST INERR              ;YES
ØØØ5ØØ                  FININØ:         ;HERE IF NOTHING TO SWAP IN(HIGH OR LOW SEG EXPANDING FROM Ø)
                        IFN FT2REL,<
                                EXTERN FININ
ØØØ5ØØ  26Ø14Ø  ØØØØØØ          PUSHJ PDP,FININ ;IS THERE A HIGH SEG WHICH MUST BE SWAPPED IN?
ØØØ5Ø1  254ØØØ  ØØØ561'         JRST FIT1       ;YES, GO SWAP IT IN(J SET TO HIGH SEG NO.JOB # IN INPJOB)
                                                ; NO, EITHER HIGH SEG ALREADY IN FOR ANOTHER USER
                                                ; OR THERE IS NONE, J STILL JOB NO.(IE LOW SEG)
                                                : OR J IS HIGH SEG WHICH EXPANDED FROM NOTHING(XPANDH)
                                                ; IN WHICH CASE IT HAS NO DISK SPACE AND DIDLING ACS
                                                : AND SETTING PROTECT TIME WON'T MATTER EITHER.
                        >
ØØØ5Ø2  1353ØØ  ØØØØØØ          LDB T,IMGIN             ;NEW CORE SIZE
ØØØ5Ø3  1351ØØ  ØØØØØØ          LDB T1,IMGOUT           ;OLD SIZE WHEN ON DISK
ØØØ5Ø4  2741ØØ  ØØØØØ6          SUB T1,T                ;OLD-NEW=DECREASE
                                                        ; HAS USER DECREASED VIRTUAL MEMORY FROM M TO N(N GR Ø)

                                                        ; WHILE OUT ON DISK(R,RUN,GET,KJOB) TO 14Ø WORDS?
                                                        ; CORE COMMAND ALWAYS FORCES SWAP IN BEFORE
                                                        ; CORE REASSIGNMENT SO NOT IN THIS CATEGORY
                                                        ; FRAGMENTED USER TOO HARD TO PARTIALLY RECLAIM DISK SP
                        ACE
                                                        ; ON REDUCTION WHICH DOES NOT GO TO Ø
ØØØ5Ø5  333ØØØ  ØØØØØ2          SKIPLE T1               ;DECREASED?
ØØØ5Ø6  2721ØØ  ØØØØØØ          ADDM T1,VIRTAL          ;YES, NOW INCREASE VIRTUAL MEMORY AVAILABLE BY
                                                        ; AMOUNT OF DECREASE IN HIGH OR LOW SEG
ØØØ5Ø7  26Ø14Ø  ØØ1Ø34'         PUSHJ PDP,ZERSWP        ;RETURN LOW SEG DISK SPACE, SET IMGOUT,IMGIN
                                                        ; AND SWP!SHF(JBTSTS) TO Ø
ØØØ51Ø  1353ØØ  ØØØØØØ          LDB T,PCORSZ            ;COMPUTE AND SET IN CORE IN CORE PROTECT TIME FROM
                                                        ; SIZE OF JOB(1K BLOCKS-1)
ØØØ511  2203ØØ  ØØØØØØ          IMUL T,PROT             ;ADD VARIABLE AMOUNT DEPENDING ON CORE SIZE
ØØØ512  2703ØØ  ØØØØØØ          ADD T,PROTØ             ;ADD FIXED AMOUNT INDEPENDENT OF CORE SIZE
ØØØ513  5Ø63Ø4  ØØØ133'         HRLM T,JBTSWP(J)
ØØØ514  2ØØ344  ØØØØØØ          MOVE JDAT,JBTDAT(J)     ;SETUP LOW SEG PROTECTION,RELOCATION
```

```
                             IFN JDAT-PROG,<
                                     MOVE PROG,JBTADR(J)
                             >
000515  200307  000000               MOVE T,JOBPC(JDAT)        ;JOB STOPPED IN EXEC MODE?
000516  603300  012000               TLNE T,USRMOD             ;TEST PD FLAG
000517  254000  000546'              JRST SWP1                 ;NO
000520  550307  000000               HRRZ T,JOBDPG(JDAT)       ;YES, ADJUST PROG AND PDP IN DUMP AC AREA
000521  275307  000000               SUBI T,(PROG)             ;OLD RELOC-NEW RELOC
000522  213000  000006               MOVNS T                   ;NEW RELOC-OLD RELOC
000523  272307  000000               ADDM T,JOBDPD(JDAT)       ;ADJUST DUMP PDP
000524  202347  000520'              MOVEM PROG,JOBDPG(JDAT)   ;STORE NEW AC PROG
000525  254000  000546'              JRST SWP1

000526  402000  000473'  INERR:      SETZM FINISH              ;CLEAR FINISH FLAG SO SWAPPING CAN CONTINUE
000527  200344  000172'              MOVE PROG,JBTADR(J)       ;SETUP RELOC,PROTECTION FOR HIGH OR LOW SEG
                                     IFN     PROG-JDAT,<MOVE JDAT,JBTDAT(J)>
000530  260140  000000               PUSHJ PDP,KCORE1          ;RETURN CORE
000531  265040  000000               JSP TAC,ERRPNT            ;PRINT ON USER CONSOLE
000532  516570  150100               ASCIZ /SWAP READ ERROR/

000533  512130  142100
000534  426452  247644
000535  000000  000000
000536  254000  000000               JRST     PCSTOP           ;STOP JOB AND FORCE RESCHEDULING
000537  213000  000004  FINOUT: MOVNS J                        ;FINISH OUTPUT, -FINISH=JOB NO.
000540  332000  001342'              SKIPE SERA                ;ANY ERRORS
000541  254000  000575'              JRST SWPREC               ;YES, RECORD ERROR AND TRY AGAIN,
                                                               ; IN A DIFFERENT PLACE ON DISK
000542  200344  000527'              MOVE PROG,JBTADR(J)       ;XWD PROTECT,,RELOC, FOR LOW SEG
                        IFN PROG-JDAT,<
                                     MOVE JDAT,JBTDAT(J)       ;JOB DATA AREA
                        >
000543  260140  000530'              PUSHJ PDP,KCORE1          ;RETURN CORE FOR LOW OR HIGH SEG JUST SWAPPED OUT
                                                               ; EVEN IF
                                                               ; ANOTHER JOB STARTED TO SHARE HIGH SEG DURING
                                                               ; SWAP OUT (GET) SINCE JOB IS MARKED WITH
                                                               ; SWP BIT ON AND CANNOT RUN UNTIL HIGH SEG IS SWAPPED B
                        ACK IN
                        IFN FT2REL,<
                                     EXTERN FINOT
000544  260140  000000               PUSHJ PDP,FINOT           ;IS THIS A HIGH SEG WHICH WAS JUST SWAPPED OUT?
000545  254000  000662'              JRST FORCEL               ;YES, J SET TO LOW SEG NO, GO TRY SWAP IT OUT
                                                               ; NO, THIS WAS A LOW SEG, ALL SWAPPING FOR THIS USER
                                                               ; IS FINISHED.
                        >
000546  402000  000526' SWP1:        SETZM FINISH              ;CLEAR FINISH FLAG
000547  332200  000000  SWP2:        SKIPE J,FORCE             ;WAITING FOR JOB TO BECOME SWAPPABLE?
000550  254000  000663'              JRST FORCE1               ;YES
000551  332200  000000  FIT0:        SKIPE J,FIT               ;NO-- WAITING TO FIT JOB IN CORE?
000552  254000  000561'              JRST FIT1                 ;YES
```

```
                             ;SCAN FOR INPUT
ØØØ553  20124Ø  ØØØ42Ø'             MOVEI DAT,ISCAN
ØØØ554  265Ø4Ø  ØØØ225'             JSP TAC,QSCAN
ØØØ555  254ØØØ  ØØØ6Ø6'             JRST CHKXPN            ;NO INPUT TO DO--CK FOR EXPANDING JOBS
ØØØ556  2ØØ3Ø4  ØØØ221'             MOVE T,JBTSTS(J)       ;THIS JOB OUT OF CORE?
ØØØ557  6Ø73ØØ  ØØ2ØØØ              TLNN T,SWP             ;SWP ON IF HIGH SEG SWAPPED OUT FOR THIS USER
                                                           ; OR BOTH SEGS SWAPPED OUT

ØØØ56Ø  254ØØ2  ØØØØØØ              JRST (TAC1)            ;NO--CONTINUE SCAN

ØØØ561  2022ØØ  ØØØ551' FIT1:       MOVEM J,FIT            ;REMEMBER JOB(OR HIGH SEG) TRYING TO FIT IN
ØØØ562  13564Ø  ØØØ5Ø2'             LDB AC1,IMGIN          ;CORE SIZE NEEDED FOR THIS SEG(Ø IF LOW SEG
                                                           ; OR HIGH SEG WITH UWP OFF ALREADY IN CORE)
                        IFE FT2REL,<
                                    CAMLE AC1,CORTAL       ;WILL LOW SEG FIT IN FREE+DORMANT CORE?
                        >
                        IFN FT2REL,<
                                    EXTERN FITSIZ
ØØØ563  26Ø14Ø  ØØØØØØ              PUSHJ PDP,FITSIZ       ;COMPUTE AMOUNT OF CORE NEEDED TO BRING IN
                                                           ; 1. THIS JOBS LOW SEG AND HIGH SEG
                                                           ; 2. THIS JOBS LOW SEG(HIGH ALREADY IN OR NONE)
                                                           ; 3. THIS HIGH SEG BECAUSE LOW SEG ALREADY IN
                                                           ;WILL LOW SEG FIT IN FREE+DORMANT+IDLE CORE?
                        >
ØØØ564  254ØØØ  ØØØ61Ø'             JRST SCNOUT            ;NO,WILL NOT FIT EVEN IF ALL DORMANT SEGS DELETED
                                                           ; AC1=TOTAL CORE NEEDED(IN K)
ØØØ565  31764Ø  ØØØØØØ              CAMG AC1,BIGHOL        ;YES, WILL THIS SEG FIT IN BIGGEST HOLE OF FREE CORE
                                                           ; WITHOUT DELETEING ANY DORMANT OR IDLE SEGS?
                                                           ; (AC1 RESTORED TO SIZE FOR JUST THIS LOW OR HIGH SEG)

ØØØ566  254ØØØ  ØØØ772'             JRST SWAPI             ;YES, GO SWAP IN THIS LOW OR HIGH SEG
                        IFN FT2REL,<
                                    EXTERN FRECR1,HOLEF
ØØØ567  336ØØØ  ØØØØØØ              SKIPN HOLEF            ;NO, ARE THERE ANY HOLES IN CORE WHICH THE SHUFFLER
                                                           ; COULD ELIMINATE(NOT COUNTING ONE AT TOP)?
ØØØ57Ø  26Ø14Ø  ØØØØØØ              PUSHJ PDP,FRECR1       ;NO, GO DELETE ONE DORMANT SEG IN CORE
                                                           ; AND ALWAYS SKIP RETURN(THERE MUST BE AT LEAST
                                                           ; ONE, OTHERWISE CORTAL=BIGHOL)MONITOR ERROR IF NONE
                        >
ØØØ571  26Ø14Ø  ØØØ466'             PUSHJ PDP,CHKSHF       ;YES, CALL CORE SHUFFLER TO MOVE ONE SEG DOWN
ØØØ572  336ØØØ  ØØØ467'             SKIPN SHFWAT           ;SHUFFLER WAITING FOR IO TO STOP?
ØØØ573  254ØØØ  ØØØ551'             JRST FITØ              ;NO, SEE IF JOB WILL FIT NOW.
ØØØ574  26314Ø  ØØØØØØ              POPJ PDP,              ;YES, RETURN AND WAIT TILL IO STOPS
```

```
                                EXTERN  VIRTAL,SWPERC

000575  200040  001342' SWPREC: MOVE     TAC,SERA        ;ERROR FLAGS
000576  436040  000000          IORM     TAC,SWPERC      ;SAVE FOR POSTERITY
000577  135040  000503'         LDB      TAC,IMGOUT      ;DECREASE TOTAL AMOUNT
000600  213000  000001          MOVNS    TAC             ;OF VIRTUAL CORE IN THE MACHINE
000601  272040  000506'         ADDM     TAC,VIRTAL      ;BY THE AMOUNT BEING GIVEN UP
000602  135040  000577'         LDB      TAC,IMGOUT
000603  661040  000001          TLO      TAC,1
000604  272040  000576'         ADDM     TAC,SWPERC
000605  254000  000670'         JRST SWAPO               ;GO TRY AGAIN


                                ;NO INPUT TO DO, CHECK FOR EXPANDING JOBS
000606  337000  000323' CHKXPN: SKIPG XJOB               ;ANY JOBS TO EXPAND?
000607  263140  000000          POPJ PDP,                ;NO, RETURN FROM SWAPPER, NOTHING TO INPUT OR OUTPUT
                                                         ; YES, FALL INTO SCNOUT WHICH WILL SWAP OUT EXPANDING
                                                         ; JOB SINCE THERE IS ONE
                                ;INPUT TO DO, CHECK TO SEF IF ANY JOBS JUST HAPPEN TO WANT TO EXPAND

                                EXTERN HIGHJB,JBTSTS,ERROR,MAXSIZ,MAXJBN,SUMCOR
000610  337000  000323' SCNOUT: SKIPG XJOB               ;ANY JOBS WAITING TO EXPAND?
000611  254000  000624'         JRST SCNJOB              ;NO, SCAN ALL JOBS IN PRIORITY ORDER LOOKING
                                                         ; FOR ONE TO SWAP OUT
000612  200200  000000          MOVE J,HIGHJB            ;YES, START WITH HIGHEST JOB NUMBER ASSIGNED
000613  205300  000001          MOVSI T,JXPN             ;SETUP JOB EXPANDED BIT
000614  616304  000556'         TDNN T,JBTSTS(J)         ;IS THIS JOB EXPANDING?
000615  367200  000614'         SOJG J,.-1               ;NO, KEEP LOOKING
                                IFN FTRCHK,<
000616  327200  000621'         JUMPG J,SCNOK
000617  402000  000323'         SETZM XJOB               ;CLEAR XJOB SO MESSAGE WILL PRINT
000620  265240  000176'         JSP DAT,ERROR            ;ERROR IF NONE FOUND
                        >
000621  370000  000323' SCNOK:  SOS XJOB                 ;DECREMENT COUNT OF EXPANDING JOBS
000622  412304  000614'         ANDCAM T,JBTSTS(J)       ;CLEAR EXPAND BIT IN JOB STATUS WORD
000623  254000  000655'         JRST FORCE0              ;GO TRY TO SWAP JOB OUT
```

```
                                ;SCAN FOR JOB TO OUTPUT IN ORDER TO MAKE ROOM FOR JOB TO COME IN
                                ;SIZE(IN K) NEEDED TO GET THIS USER IN CORE IS IN AC1(FITSIZ)
                                ;JUST LOW SEG SIZE IF NO HIGH OR HIGH ALREADY IN, JUST HIGH IF LOW ALREADY IN,
                                ;OR SUM IF BOTH MUST BE SWAPPED IN

ØØØ624  2ØØ3ØØ  ØØØØØØ  SCNJOB: MOVE T,CORTAL          ;INITIALIZE FREE CORE COUNTER
ØØØ625  2Ø23ØØ  ØØØØØØ          MOVEM T,SUMCOR
ØØØ626  4Ø2ØØØ  ØØØØØØ          SETZM MAXSIZ           ;CLEAR SIZE OF LARGEST JOB
ØØØ627  2Ø124Ø  ØØØ436'         MOVEI DAT,OSCAN        ;SCAN ALL JOBS RANKED IN PRIORITY TO BE SWAPPED OUT
ØØØ63Ø  265Ø4Ø  ØØØ225'         JSP TAC,QSCAN
ØØØ631  254ØØØ  ØØØ714'         JRST NOFIT             ;NO MORE JOBS LEFT, CANNOT FIT JOB IN CORE
ØØØ632  3162ØØ  ØØØ561'         CAMN J,FIT             ;IS THIS JOB WE ARE TRYING TO FIT IN?
ØØØ633  254ØØ2  ØØØØØØ          JRST(TAC1)             ;YES, GO FIND NEXT JOB TO OUTPUT
ØØØ634  3353Ø4  ØØØ622'         SKIPGE T,JBTSTS(J)     ;JOB RUN BIT STILL ON(JOB STILL WANT TO RUN)?
ØØØ635  335ØØ4  ØØØ513'         SKIPGE JBTSWP(J)       ;YES, IS PROTECT TIME STILL LEFT?
                                                       ; PROTECT TIME IS DECREMENTED ONLY WHEN
                                                       ; A JOB IS RUNABLE, SO LOOK AT IT
                                                       ; ONLY IF RUN BIT STILL ON
                                TLNE T,NSWP+SWP        ;NO, IS THIS JOB NOT TO BE SWAPPED OR ALREADY SWAPPED?
ØØØ636  6Ø33ØØ  Ø12ØØØ
                                                       ; (DISPLAY, REAL TIME)?
ØØØ637  254ØØ2  ØØØØØØ          JRST (TAC1)            ;YES,CONTINUE SCAN TO FIND ANOTHER
ØØØ64Ø  5543Ø4  ØØØ542'         HLRZ T,JBTADR(J)       ;PICK UP SIZE OF JOB
                                JUMPE T,(TAC1)         ;CONTINUE SCAN IF NOT IN CORE (HIGH SEG ALREADY SWAPPED
ØØØ641  3223Ø2  ØØØØØØ
                                                       ; OUT FOR THIS USER IF NO LOW SEG)
ØØØ642  2403ØØ  777766          ASH T,-12              ;CONVERT TO 1K BLOCKS
ØØØ643  2713ØØ  ØØØØØ1          ADDI T,1
                        IFN FT2REL,<
                                EXTERN FORSIZ
ØØØ644  26Ø14Ø  ØØØØØØ          PUSHJ PDP,FORSIZ       ;INCREASE SIZE(T) BY HIGH SEG IF THIS JOB
                                                       ; IS ONLY ONE IN CORE USING HIGH SEG(J= JOB # STILL)
                        >
ØØØ645  3173ØØ  ØØØ626'         CAMG T,MAXSIZ          ;LARGEST SO FAR?
ØØØ646  254ØØØ  ØØØ651'         JRST FORCE2            ;NO
ØØØ647  2023ØØ  ØØØ645'         MOVEM T,MAXSIZ         ;YES, SAVE SIZE
ØØØ65Ø  2022ØØ  ØØØØØØ          MOVEM J,MAXJBN         ;AND JOB NUMBER
ØØØ651  2723ØØ  ØØØ625' FORCE2: ADDM T,SUMCOR          ;ADD TO TOTAL
ØØØ652  31364Ø  ØØØ651'         CAMLE AC1,SUMCOR       ;FOUND ENOUGH CORE FOR JOB TO BE FIT IN?
ØØØ653  254ØØ2  ØØØØØØ          JRST (TAC1)            ;NO, LOOK FOR MORE
ØØØ654  2ØØ2ØØ  ØØØ65Ø'         MOVE J,MAXJBN          ;YES, SWAP OUT LARGEST
```

```
000655  260140  000000   FORCE0: PUSHJ PDP,TRYSWP         ;CAN THIS JOB BE STOPPED IN ORDER TO DO SWAP?
000656  254002  000000           JRST (TAC1)              ;NO, NSWP OR NSHF SET(DISPLAY,REAL TIME) OR
                                                          ; SAVE OR GET IN PROGRESS WITH DEVICE STILL ACTIVE
                                                          ; LOOK FOR AN OTHER JOB TO SWAP

                         IFN FT2REL,<
                                 EXTERN FORHGH
000657  260140  000000           PUSHJ PDP,FORHGH         ;IS THERE A HIGH SEG TO BE WRITTEN BEFORE
                                                          ; TRYING TO SWAP OUT LOW SEGMENT?
                                                          ; WRITE HIGH SEG IF ALL OF THE FOLLOWING ARE TRUE;
                                                          ; 1. JOB HAS A HIGH SEG AND
                                                          ; 2. IT HAS NOT BEEN SWAPPED FOR THIS USER
                                                          ;    (SWP=0 FOR JOB)
                                                          ; 3. IT IS IN CORE(NOT XPANDH)
                                                          ; 4. IF IN-CORE COUNT IS EXACTLY 1 MEANING
                                                          ;    THIS ONLY USER USING IN CORE
                                                          ; 5. HIGH SEG NOT ON DISK YET
                                                          ; 6. THIS HIGH SEG IS NOT THE SAME ONE AS JOB
                                                          ;    BEING FITTED IN IS GOING TO WANT

                                                          ; RETURN HIGH SEG NO. IN J IF YES, OTHERWISE
                                                          ; RETURN LOW SEG NO.
                                                          ; IF JOB JUST HAS LOW SEG, SHF BIT IS SET IN JBTSTS
                                                          ;    FOR JOB SO IO WILL STOP NEXT BUFFER
                         >
000660  205300  002000           MOVSI T,SWP!IFE FT2REL,<SHF> ;SET SWAPPED OUT BIT FOR LOW OR HIGH SEG
000661  436304  000634'          IORM T,JBTSTS(J)         ;SCHEDULER WILL NO LONGER RUN THIS JOB
                                                          ; SET SHF BIT IF ONE SEG SOFTWARE, SO IO WILL
                                                          ; STOP AFTER NEXT BUFFERFUL.

000662  202200  000547'  FORCEL: MOVEM J,FORCE            ;ASSUME NOT SWAPPABLE--IS IT?

000663                   FORCE1:
                         IFN JDAT-PROG,<
                                 MOVE JDAT,JBTDAT(J)
                         >
000663  336344  000640'          SKIPN PROG,JBTADR(J)     ;LOC. IN PHYSICAL CORE, IS CORE
                                                          ; ASSIGNED IN MEMORY?
000664  254000  000670'          JRST SWAPO               ;NO, CANNOT HAVE ACTIVE DEVICES
000665  312200  000022'          CAME J,JOB               ;IF THIS IS CURRENT JOB, WAIT UNTIL
                                                          ; PROTECTED AREA IS MOVED BACK TO JOB DATA AREA
000666  260140  000000           PUSHJ PDP,ANYDEV         ;ANY ACTIVE DEVICES?(2ND HALF OF ANYACT ROUT.)
000667  263140  000000           POPJ PDP,                ;YES--RETURN AND WAIT FOR I/O TO STOP.
```

```
                        ;SWAP OUT LOW OR HIGH SEGMENT

                        INTERNAL FTTRACK

000670                  SWAPO:
                        IFN FTTRACK,<EXTERN LASOUT
                                 MOVEM J,LASOUT        ;SAVE LAST SWAP OUT FOR DEBUGGING ONLY
                        >
000670  402000  000662'          SETZM FORCE           ;CLEAR FORCE FLAG
000671  554304  000663'          HLRZ T,JBTADR(J)      ;COMPUTE CORE IMAGE
000672  322300  000546'          JUMPE T,SWP1          ;DONT OUTPUT IF 0 CORE(IMGOUT ALREADY SET TO 0
                                                       ; WHEN CORE WAS RETURNED

000673  550104  000671'          HRRZ T1,JBTADR(J)
000674  212300  000001           MOVNM T,T2            ;*SAVE COUNT FOR CALL TO SQOUT
000675  240300  777766           ASH T,-↑D10           ;CONVERT TO 1K BLOCKS
000676  271300  000001           ADDI T,1
000677  137300  000602'          DPB T,IMGOUT          ;RECORD AS OUT IMAGE
000700  505101  777777           HRLI T1,-1(T2)        ;*BUILD AND SAVE IOWD FOR SQOUT
000701  261140  000002           PUSH PDP,T1           ;*
000702  135240  000562'          LDB DAT,IMGIN         ;HAS SIZE OF CORE NEEDED WHEN NEXT SWAPPED IN
000703  336000  000005           SKIPN DAT             ;ALREADY BEEN SET(XPAND)
000704  137300  000702'          DPB T,IMGIN           ;NO, SO SET TO # 1K BLOCKS OF CORE NEEDED
000705  200240  000006           MOVE DAT,T            ;*CONVERT CORE IMAGE TO 128 WD BLOCKS
000706  260140  001206'          PUSHJ PDP,GXSAT       ;*GET DEVICE STORAGE
000707  254000  000717'          JRST FULL             ;*NONE AVAILABLE
000710  506044  000635'          HRLM TAC,JBTSWP(J)    ;*SAVE DEVICE ADDRESS
000711  212200  000546' OUTP2:   MOVNM J,FINISH        ;DISK SWAP SPACE ASSIGNED, NOW SET FINISH FLAG
                                                       ; SO THAT SWAPPER WILL KNOW WHICH SEG FINISHED
                                                       ; WHEN IO COMPLETED(SQREQ BECOMES ZERO)
000712  262140  000002           POP PDP,TAC1          ;*GET IOWD
000713  254000  001066'          JRST SQOUT            ;*START OUTPUT AND RETURN

000714  402000  000632' NOFIT:   SETZM FIT             ;FORGET ABOUT FITTING IN A JOB ON DISK
000715  476000  000141'          SETOM   INFLG         ;*** EXPERIMENTAL *** MARK DFSIRE TO INPUT
000716  263140  000000           POPJ PDP,             ;ALL JOBS IN CORE ARE HIGHER PRIORITY.
```

```
                               ;COME HERE WHEN THE AMOUNT OF SPACE NEEDED ON THE DISK
                               ;IS NOT AVAILABLE IN ONE CONTIGUOUS BLOCK

                                       EXTERN GETFCR

ØØØ717  5Ø624Ø  ØØØØ17  FULL:   HRLM    DAT,AC3         ;SAVE DAT (LARGEST AVAILABLE HOLE)
ØØØ72Ø  26Ø14Ø  ØØØ755'         PUSHJ   PDP,FULCOR      ;GET 4 FREE CORE LOCS
ØØØ721  55424Ø  ØØØØ17          HLRZ    DAT,AC3         ;RESTORE DAT
ØØØ722  2ØØ74Ø  ØØØØØ2          MOVE    AC3,TAC1        ;LOC OF 1ST FREE CELL
ØØØ723  5Ø574Ø  777774          HRLI    AC3,-4          ;4 LOCS
ØØØ724  66Ø1ØØ  4ØØØØØ          TRO     TAC1,FRGSEG     ;LIGHT FRAGMENTED BIT
ØØØ725  5Ø61Ø4  ØØØ71Ø'         HRLM    TAC1,JBTSWP(ITEM) ;SAVE LOC OF TABLE IN JBTSWP
ØØØ726  26114Ø  ØØØØØ5  FULL1:  PUSH    PDP,DAT         ;SAVE AMOUNT OF SPACE BEING REQUESTED
ØØØ727  26Ø14Ø  ØØ12Ø6' FULL1A: PUSHJ   PDP,GXSAT       ;GET SOME SWAPPING SPACE
ØØØ73Ø  254ØØØ  ØØØ742'         JRST    FULL2           ;CANT HAVE THAT MUCH
ØØØ731  542Ø57  ØØØØØØ          HRRM    TAC,(AC3)       ;SAVE LOC OF THE DISK SPACE
ØØØ732  26214Ø  ØØØØØ5          POP     PDP,DAT         ;RESTORE AMT GOTTEN
ØØØ733  5Ø6257  ØØØØØØ          HRLM    DAT,(AC3)       ;SAVE AMOUNT IN TABLE
ØØØ734  2743ØØ  ØØØØØ5          SUB     T,DAT           ;AMOUNT STILL NEEDED

ØØØ735  3223ØØ  ØØØ751'         JUMPE   T,FULSET        ;THROUGH IF NEED Ø K NOW
ØØØ736  26Ø14Ø  ØØØ76Ø'         PUSHJ   PDP,BMPAC3      ;STEP TO NEXT TABLE LOCATION
ØØØ737  2ØØ24Ø  ØØØØØ6          MOVE    DAT,T           ;TRY TO GET ALL WE NEED NOW IN 1 CHUNK
ØØØ74Ø  254ØØØ  ØØØ726'         JRST    FULL1
ØØØ741  2ØØ24Ø  ØØØØØ6  FULL1B: MOVE DAT,T      ;RESET AMOUNT OF SPACE NEEDED

                        ;COME HERE WHEN CANT GET THE CHUNK REQUESTED
ØØØ742  2Ø2243  ØØØØØØ  FULL2:  MOVEM   DAT,(PDP)       ;DAT HAS LARGEST CHUNK AVAILABLE
ØØØ743  32724Ø  ØØØ727'         JUMPG   DAT,FULL1A      ;GO GET THAT AMOUNT
                        IFN FT2REL,<                    ; -1=Ø MEANS NO MORE LEFT
                                EXTERN FRESWP
ØØØ744  26Ø14Ø  ØØØØØØ          PUSHJ PDP,FRESWP        ;TRY TO DELETE AN UNUSED HIGH SEG FROM DISK
ØØØ745  254ØØØ  ØØØ741'         JRST FULL1B             ;FOUND ONE, TRY AGAIN, J PRESERVED
                                                        ; NONE FOUND, PRINT MONITOR ERROR
                        >
```

```
                                          EXTERN CERROR

000746  262140  000001                    POP     PDP,TAC           ;WHAT? NONE LEFT?
000747  262140  000001                    POP     PDP,TAC           ;SET PDP TO RIGHT VALUE
000750  265240  000000                    JSP     DAT,CERROR        ;ERROR IN MONITOR AT .....

                                  ;HERE WHEN THE TOTAL AMOUNT OF SPACE NEEDED HAS BEEN OBTAINED
000751  260140  000760'   FULSET: PUSHJ   PDP,RMPAC3        ;STEP TO NEXT (LAST) TABLE LOCATION
000752  402017  000000            SETZM   (AC3)             ;ZERO MEANS END OF TABLE
000753  554044  000725'           HLRZ    TAC,JBTSWP(ITEM) ;LOC OF TABLE OF FRAGMENTS
000754  254000  000711'           JRST    OUTP2             ;GO START OUTPUT

                                  ;HERE TO GET 4 LOCS OF FREE CORE
000755  261140  000004    FULCOR: PUSH    PDP,ITEM          ;GETFCR USES ITEM
000756  260140  000000            PUSHJ   PDP,GETFCR        ;GET 4 CELLS
000757  254000  000000            JRST IPOPJ                ;RETORE ITEM AND RETURN

                                  ;STEP AC3 TO NEXT LOC OF TABLE BEING BUILT
000760  253740  000157'   RMPAC3: AOBJN   AC3,CPOPJ         ;OK IF MORE LOCS OF TABLE
000761  260140  000755'           PUSHJ   PDP,FULCOR        ;GET 4 MORE LOCS
000762  505740  777774            HRLI    AC3,-4
000763  306117  000000            CAIN    TAC1,(AC3)        ;ARE THEY CONTIGUOUS?
000764  263140  000000            POPJ    PDP,              ;YES. RETURN
000765  200057  777777            MOVE    TAC,-1(AC3)       ;NO. CONVERT LAST GOOD LOC
000766  562117  777777            HRROM   TAC1,-1(AC3)      ;TO A POINTER TO NEXT PART OF TABLE
000767  202042  000000            MOVEM   TAC,(TAC1)        ;STORE GOOD DATA IN 1ST WD OF NEW PART

000770  540740  000002            HRR     AC3,TAC1          ;NEW TABLE LOC
000771  253740  000760'           AOBJN   AC3,CPOPJ         ;COUNT WORD AND RETURN
```

```

                                  ;SWAP IN A JOB OR HIGH SEGMENT

ØØØ772                            SWAPI:

                                  IFN FTTRACK,<EXTERN LASIN
                                          MOVEM J,LASIN          ;SAVE LAST SWAP IN FOR DEBUGGING ONLY
                                  >

ØØØ772  2Ø22ØØ  ØØØ711'                   MOVEM J,FINISH         ;SET FINISH FLAG TO INPUT
ØØØ773  4Ø2ØØØ  ØØØ714'                   SETZM FIT              ;CLEAR FIT FLAG
ØØØ774  135Ø4Ø  ØØØ7Ø4'                   LDB TAC,IMGIN          ;SIZE OF CORE TO BE ASSIGNED WHEN SWAPPED IN (INK)
ØØØ775  242Ø4Ø  ØØØØ12                    LSH TAC,↑D1Ø           ;CONVERT TO HIGHEST ADR
ØØØ776  275Ø4Ø  ØØØØØ1                    SUBI TAC,1             ;-1 FOR CALL TO CORGET
ØØØ777  332344  ØØØ673'                   SKIPE PROG,JBTADR(J)   ;IS (LOW)SEG ALREADY IN CORE?
ØØ1ØØØ  254ØØØ  ØØØ5ØØ'                   JRST FININØ            ;YES, POSSIBLE IF THIS IS LOW SEG AND ONLY
                                                                 ; HIGH SEG WAS SWAPPED OUT.
ØØ1ØØ1  26Ø14Ø  ØØØØØØ                    PUSHJ PDP,CORGET       ;NO, GET CORE FOR LOW OR HIGH SEG
ØØ1ØØ2  26524Ø  ØØØØØØ                    JSP DAT,OERROR         ;NOT AVAILABLE-SHOULD NEVER HAPPEN(TELL OPER)


                                  IFN FT2REL,<EXTERN FITHGH
ØØ1ØØ3  26Ø14Ø  ØØØØØØ                    PUSHJ PDP,FITHGH       ;INCREASE INCORE COUNT FOR THIS JOB'S HIGH SEG.
                                  >
ØØ1ØØ4  1353ØØ  ØØØ677'                   LDB T,IMGOUT           ;GET OUTPUT IMAGE
ØØ1ØØ5  3223ØØ  ØØØ5ØØ'                   JUMPE T,FININØ         ;DONT INPUT IF OUT IMAGE IS Ø
ØØ1ØØ6  1351ØØ  ØØØ774'                   LDB TAC1,IMGIN  ;IS SIZE OF CORE SMALLER THAN DISK SPACE?
ØØ1ØØ7  3151ØØ  ØØØØØ6                    CAMGE TAC1,T           ;WELL?
ØØ1Ø1Ø  2ØØ3ØØ  ØØØØØ2                    MOVE T,TAC1            ;YES, ONLY INPUT SMALLER AMOUNT(R,RUN,GET,KJOB)
ØØ1Ø11  2423ØØ  ØØØØ34                    LSH T,↑D18+↑D1Ø        ;*BUILD IOWD FOR SQIN
ØØ1Ø12  21Ø1ØØ  ØØØØØ6                    MOVN TAC1,T            ;*
ØØ1Ø13  54Ø1Ø4  ØØØ777'                   HRR TAC1,JBTADR(J)     ;*
ØØ1Ø14  554Ø44  ØØØ753'                   HLRZ TAC,JBTSWP(J)     ;*GET DEVICE ADDRESS
ØØ1Ø15  254ØØØ  ØØ1Ø65'                   JRST SQIN              ;*START INPUT
```

```
                                  ;ROUTINE TO CHANGE DISK SWAPPING SPACE ALLOCATION(OR SET TO Ø)
                                  ;DIFFERS FROM ZERSWP IN THAT VIRTUAL TALLY FOR SYSTEM IS ALSO CHANGED
                                  ;CALLED FROM COREØ
                                  ;CALL:  MOVE ITEM,JOB OR HIGH SEG NO.
                                  ;       MOVE TAC,#1K BLOCKS TO BE NEW ASSIGNMENT
                                  ;       PUSHJ PDP,CHGSWP
                                  ;       ALWAYS RETURN
                                  ;CALLED ONLY FROM VIRTUAL+PHYSICAL CORE ROUTINE COREØ

                                  INTERN CHGSWP
                                  EXTERN JBTSTS,IMGIN,IMGOUT,JBTSWP,VIRTAL,IPOPJ

ØØ1Ø16  1351ØØ  ØØ1ØØ6' CHGSWP: LDB TAC1,IMGIN            ;SIZE WHEN SEG NEXT SWAPPED IN
ØØ1Ø17  322Ø4Ø  ØØ1Ø35'         JUMPE TAC,CHG1            ;IS ZERO BEING ASKED FOR?
ØØ1Ø2Ø  242Ø4Ø  777766          LSH TAC,-12               ;NO, CONVERT TO 1K BLOCKS
ØØ1Ø21  271Ø4Ø  ØØØØØ1          ADDI TAC,1                ;BUT DO NOT ATTEMPT TO RETURN DISK SPACE
                                                          ; SINCE IT MIGHT BE FRAGMENTED(SWAPPER WILL
                                                          ; RETURN ALL OF DISK SPACE ON NEXT SWAPIN)
                                                          ; HAPPENS ONLY ON R,RUN,GET,KJOB
ØØ1Ø22  137Ø4Ø  ØØ1Ø16'         DPB TAC,IMGIN             ;STORE NEW SIZE WHEN NEXT SWAPPED IN
ØØ1Ø23  26114Ø  ØØØØØ4          PUSH PDP,ITEM             ;SAVE AN AC
ØØ1Ø24  1352ØØ  ØØ1Ø34'         LDB ITEM,IMGOUT           ;GET OLD DISK SIZE OF THIS USER(USES ITEM)
ØØ1Ø25  3151ØØ  ØØØØØ4          CAMGE TAC1,ITEM           ;IS OLD IN-CORE SIZE BIGGER?
ØØ1Ø26  2ØØ12Ø  ØØØØØ4          MOVE TAC1,ITEM            ;NO, USE DISK SIZE AS USER'S OLD VIRTUAL CORE
ØØ1Ø27  315Ø4Ø  ØØØØØ4          CAMGE TAC,ITEM            ;IS NEW IN-CORE SIZE BIGGER?
ØØ1Ø3Ø  2ØØØ4Ø  ØØØØØ4          MOVE TAC,ITEM             ;NO, USE DISK SIZE AS USER NEW
                                                          ; VIRTUAL CORE
ØØ1Ø31  2741ØØ  ØØØØØ1          SUB TAC1,TAC              ;DECREASE OF USER VIRT. CORE=OLD-NEW
ØØ1Ø32  2721ØØ  ØØØ6Ø1'         ADDM TAC1,VIRTAL          ;USER'S DECREASE=SYSTEM'S INCREASE OF VIRTUAL
                                                          ; CORE
ØØ1Ø33  254ØØØ  ØØØ757'         JRST IPOPJ                ;RESTORE ITEM AND RETURN
```

```
                                ;ROUTINE TO RETURN ALL OF DISK SPACE FOR A LOW OR HIGH SEG
                                ;THIS IS A PHYSICAL DEALLOCATION ONLY AND HAS NO EFFECT ON A SEGMENTS
                                ;VIRTUAL CORE ASSIGNMENT
                                ;CALL:  MOVE ITEM,JOB NO. OR HIGH SEG NO.
                                ;       PUSHJ PDP,ZERSWP
                                ;CALLED FROM SEGCON IN MANY PLACES(5)
                                ;AND FININØ HERE IN SWAP

                                        INTERN ZERSWP

ØØ1Ø34  634Ø4Ø  ØØØØØ1  ZERSWP: TDZA TAC,TAC             ;REQUEST O SPACE ON DISK AND ALWAYS SKIP
ØØ1Ø35  2721ØØ  ØØ1Ø32' CHG1:   ADDM TAC1,VIRTAL         ;INCREASE SIZE OF VIRTUAL CORE AVAILABLE IN SYSTEM
                                                         ; AND THEN RETURN ALL OF DISK SPACE(CHGSWP)
ØØ1Ø36  2051ØØ  ØØ6ØØØ          MOVSI TAC1,SWP!SHF       ;CLEAR SWAPPED OUT BIT IN JOB OR SEG
ØØ1Ø37  4121Ø4  ØØØ661'         ANDCAM TAC1,JBTSTS(ITEM);STATUS WORD(SHF SET IF IO WAS TO BE STOPPED
                                                         ; FOR SWAP OR CORE SHUFFLE

ØØ1Ø4Ø  26114Ø  ØØØØØ5          PUSH PDP,DAT             ;SAVE TTY OUTPUT BYTE POINTER(COMMAND DECODER)
ØØ1Ø41  13524Ø  ØØ1Ø24'         LDB DAT,IMGOUT           ;*SIZE ON DISK(1K BLOCKS)

ØØ1Ø42  32224Ø  ØØ1Ø45'         JUMPE DAT,CHG3           ;DID SEG HAVE ANY DISK SPACE?
ØØ1Ø43  554Ø44  ØØ1Ø14'         HLRZ TAC,JBTSWP(ITEM)    ;*YES, LOGICAL DISK BLOCK+FRGSEG BIT
ØØ1Ø44  26Ø14Ø  ØØ1224'         PUSHJ PDP,FXSAT          ;*FREE THE DISK BLOCKS NO LONGER NEEDED
ØØ1Ø45  26214Ø  ØØØØØ5  CHG3:   POP PDP,DAT              ;RESTORE TTY OUTPUT BYTE POINTER
ØØ1Ø46  2Ø1Ø4Ø  ØØØØØØ          MOVEI TAC,Ø              ;Ø IS NEW DISK ASSIGNMENT
ØØ1Ø47  137Ø4Ø  ØØ1Ø41'         DPB TAC,IMGOUT           ;SET DISK ASSIGNMENT TO Ø
ØØ1Ø5Ø  137Ø4Ø  ØØ1Ø22'         DPB TAC,IMGIN            ;SET NEW CORE IMAGE BLOCK SIZE WHEN NEXT SWAPPED IN
                                                         ; HERE FROM CHGSWP IF NOT ASKING FOR Ø
ØØ1Ø51  26314Ø  ØØØØØØ          POPJ PDP,                ;RETURN
```

```
        EXTERN PROTØ,PROT               ;PROT AND PROTØ OCCUR IN COMMON

IFE     FTRC1Ø, <
XP ICPRT1,3+1*3                         ;PROTECT TIME IN CLOCK TICS=
XP ICPROT,↑D1Ø                          ;((JOBSIZE/1K)+PROTØ)*PROT
                                        ; PROTØ=3,PROT=4 PRODUCE PROTECT TIMES ROUGHLY
                                        ; EQUAL TO 27Ø DISK SWAP(1-WAY) TIMES.
>
IFN     FTRC1Ø, <
;SIMILAR IN-CORE PROTECT TIME PARAMETERS FOR FASTER RD-1Ø DISK......
XP ICPRT1,3+1*3                         ;ZERO CORE PLUS K MULTIPLIER
XP ICPROT,3                             ;MULTIPLY BY K-1 OF LOW SEG
>
```

```
                                ;XPAND SETS CONDITIONS TO GET MORE CORE FOR A JOB BY SWAPPING IN OUT
                                ,THEN BACK IN TO DESIRED AMOUNT.
                                ,JOBS POSITION IN QS NOT AFFECTED.
                                ;CALLED ONLY FROM CORE COMMAND
                                ;ASSUMES CALL FOR CURRENT JOB IF EXPANDING HIGH SEG,IE ASSUME AT UUO LEVEL
                                ;THIS IS TRUE SINCE THERE IS NO CORE COMMAND WHICH CAN EXPAND HIGH SEG
                                ,CALL:  MOVE ITEM,[JOB NO.]
                                ,       MOVE TAC,[HIGHEST LEGAL ADDRESS DESIRED]
                                ,       PUSHJ PDP,XPAND
                                ;       RETURN, TAC DESTROYED

001052  242040  777766          XPAND:  LSH TAC,-12                     ;CONVERT HIGHEST DESIRED ADDRESS
001053  271040  000001                  ADDI TAC,1                      ;TO 1K BLOCKS
001054  137040  001050'                 DPB TAC,IMGIN                   ;STORE, SO SWAPPER WILL KNOW HOW MUCH CORE
                                                                        ; TO REQUEST WHEN NEXT SWAPPED IN

                                ;ROUTINE TO FLAG JOB TO BE STOPPED AND SWAPPED OUT
                                ;BECAUSE IT HAS JUST BEEN CONNECTED TO A HIGH SHARABLE SEG WHICH IS ON DISK
                                ;OR ON ITW WAY IN OR OUT.  THE SIZE OF THE HIGH SEG IS UNCHANGED

                                ;THE JOB MUST BE STOPPED UNTIL HIGH SEG SWAPPED IN JUST AS IF JOB HAS
                                ;EXPANDED HIGH SEG(MUST BE CALLED FROM UUO LEVEL FOR CURRENT JOB IF HIGH SEG)
                                ;CALL:  MOVE ITEM,HIGH SEG NUMBER
                                ;       PUSHJ PDP,XPANDH

                                        INTERN XPANDH
                                        EXTERN IPOPJ

001055                          XPANDH:
                                IFN FT2REL,<
001055  261140  000004                  PUSH PDP,ITEM                   ;SAVE JOB NUMBER
001056  303200  000150'                 CAILE ITEM,JOBMAX               ;IS THIS A LOW OR HIGH SEG?
001057  200200  000665'                 MOVE ITEM,JOB                   ;HIGH,SO GET JOB NO.(MUST BE CURRENT JOB)
                                >
001060  205100  000001                  MOVSI TAC1,JXPN                 ;SET THIS JOB EXPANDING BIT SO IT WILL NOT BE RUN
001061  616104  001037'                 TDNN TAC1,JBTSTS(ITEM)          ;IS IT ALREADY SET FOR THIS JOB?(UNLIKELY)
001062  350000  000323'                 AOS XJOB                        ;NO, INCREMENT COUNT ONLY ONCE FOR EACH JOB EXPANDING
001063  436104  001061'                 IORM TAC1,JBTSTS(ITEM)          ;AND SET JOB EXPANDING BIT
                                IFE FT2REL,<
                                        POPJ PDP,                       ;RETURN
                                >
                                IFN FT2REL,<
001064  254000  201033'                 JRST IPOPJ                      ;RESTORE JOB OR HIGH SEG NUMBER (ITEM) AND RETURN
                                >
```

```
                        SUBTTL  SWPSER R,KRASIN/AF TS4,34  Ø3 FEB 69  V4Ø6

                        INTERNAL SQIN,SQOUT,SQGO,SQGO1
                        INTERNAL FTSWAP
                        EXTERNAL DFBUSY,DFRED,DFWRT,CPOPJ,JOBDAC,MJOBCK,CHECK,JBTCHK

                        ;PUT A REQUEST IN THE SWAPPING QUEUE, ENTER AT SQIN FOR
                        ;       INPUT, SWOUT FOR OUTPUT
                        ;CALL:  MOVE TAC1,XWD -NO, OF WORDS,FIRST CORE LOC,(IE IOWD+1)
                        ;       HRRZ TAC,DISK BLOCK NO,
                        ;       PUSHJ PDP,SQIN/SQOUT
                        ;       RETURN HERE ALWAYS
                        ;       CONTENTS OF TAC,TAC1 LOST

ØØ1Ø65  661Ø4Ø  4ØØØØØ  SQIN:   TLO TAC,4ØØØØØ          ;SET READ INDICATOR
ØØ1Ø66  2Ø2Ø4Ø  ØØ1342' SQOUT:  MOVEM TAC,SERA          ;STORE THE BLOCK NUMBER
ØØ1Ø67  2Ø21ØØ  ØØ134Ø'         MOVEM TAC1,SQREQ        ;STORE THE IOWD
ØØ1Ø7Ø  2Ø21ØØ  ØØ1341'         MOVEM TAC1,ESQREQ       ;SAVE IN CASE OF DISK ERROR ON FRAGMENTED JOB
ØØ1Ø71  211Ø4Ø  ØØØØØ1          MOVNI TAC,1             ;IS THE DEVICE BUSY?
ØØ1Ø72  25Ø Ø4Ø  ØØØØØØ          EXCH TAC,DFBUSY
ØØ1Ø73  326Ø4Ø  ØØØ771'         JUMPN TAC,CPOPJ         ;YES IF JUMP


                ØØØØØ3  ERATRY=3        ;NO, OF TIMES TO READ AND WRITE ON ERRORS

                        ;START UP DEVICE WITH SWAPPING REQUEST, THIS ROUTINE
                        ;IS CALLED FROM DISK INTERRUPT SERVICE, AS WELL AS FROM ABOVE,
                        ;IF A SWAPPER REQUEST IS WAITING(SQREQ WILL BE NON-ZERO)

ØØ1Ø74  2Ø11ØØ  ØØØØØ3  SQGO:   MOVEI TAC1,ERATRY
ØØ1Ø75  2Ø21ØØ  ØØ1343'         MOVEM TAC1,SERACT
ØØ1Ø76  2Ø5Ø4Ø  4ØØØØØ          MOVSI TAC,4ØØØØØ
ØØ1Ø77  612Ø4Ø  ØØ1342'         TDNE TAC,SERA           ;WRITE?
ØØ11ØØ  254ØØØ  ØØ11Ø7'         JRST SQGO1              ;NO
ØØ11Ø1  55Ø Ø4Ø  ØØ134Ø'         HRRZ TAC,SQREQ
ØØ11Ø2  271Ø4Ø  ØØØØØØ          ADDI TAC,JOBDAC
ØØ11Ø3  5Ø5Ø4Ø  ØØØØØØ          HRLI TAC,MJOBCK
ØØ11Ø4  26Ø14Ø  ØØØØØØ          PUSHJ PDP,CHECK
ØØ11Ø5  214Ø4Ø  ØØØ772'         MOVM    TAC,FINISH
ØØ11Ø6  2Ø21Ø1  ØØØØØØ          MOVEM   TAC1,JBTCHK(TAC)
```

```
001107  402000  001344' SQG01:  SETZM   SQLEN           ;ZERO AMOUNT TRANSFERRED SO FAR
001110  200100  001340'         MOVE TAC1,SQREQ         ;*PUT IOWD INTO TAC1
001111  205040  200000          MOVSI TAC,200000        ;*SET "SWAPPER I/O GOING" FLAG ON
001112  437040  001342'         ORB TAC,SERA     ;*
001113  626040  400000          TRZN    TAC,FRGSEG      ;*FRAGMENTED?
001114  254000  001142'         JRST    SQG02           ;*NO, READ IN ENTIRE (OR PART) OF SEG

                                EXTERN CLCOR1

001115  261140  000005  FRAGIO: PUSH    PDP,DAT
001116  210240  001344'         MOVN    DAT,SQLEN       ;AMOUNT PREVIOUSLY TRANSFERRED
001117  274100  000005          SUB     TAC1,DAT        ;INCREASE CORE ADDRESS BY WORDCOUNT PREVIOUS
001120  574241  000000  FRGI01: HLRE    DAT,(TAC)       ;NO OF K IN THIS  DISK CHUNK
001121  540041  000000          HRR     TAC,(TAC)       ;SWAPPING ADDRESS FOR THIS DISK CHUNK
001122  325240  001127'         JUMPGE  DAT,FRGI02      ;POINTER TO NEW CORE LIST IF NEG.
001123  201240  077777          MOVEI DAT,77777         ;CLEAR OUT ADR(15 BITS)
001124  412240  001342'         ANDCAM DAT,SERA         ;
001125  436040  001342'         ORM TAC,SERA            ;INSERT NEW ADDRESS
001126  254000  001120'         JRST FRGI01


001127  242240  000012  FRGI02: LSH     DAT,12          ;CONVERT FROM K TO WORDS
001130  272240  001344'         ADDM DAT,SQLEN          ;ADD TO PREVIOUSLY TRANSFERRED AMOUNT
001131  213000  000005          MOVNS DAT               ;-N WORDS
001132  506240  000002          HRLM    DAT,TAC1        ;IOWD IN TAC1
001133  524240  000005          HRLO DAT,DAT            ;-NO. OF WRDS FOR THIS DISK TRANSFER TO LH
001134  317240  001340'         CAMG DAT,SQREQ          ;COMPARE WITH - NO. WORDS FOR REST OF SEG
001135  500100  001340'         HLL TAC1,SQREQ          ;SWAPPER ONLY WANTS TO READ A PORTION OF SEG
                                                        ; NOT ALL OF IT(R,RUN,GET,KJOB COMMAND)
001136  213000  000005          MOVNS DAT               ;+NO. OF WORDS FOR THIS NEXT TRANSFER
001137  271240  777777          ADDI DAT,777777
001140  272240  001340'         ADDM DAT,SQREQ          ;UPDATE LH OF IOWD FOR ENTIRE SEG, SO IT HAS
                                                        ; -NO. OF WORDS LEFT AFTER THIS TRANSFER IS DONE
001141  262140  000005          POP     PDP,DAT
001142  621040  377777  SQG02:  TLZ TAC,377777          ;*CLEAR POSSIBLE TRASH IN LH.
001143  241040  000003          ROT TAC,BLKSPK          ;*RE-POSITION DISK LOGICAL BLOCK NUMBER.
001144  622040  000004          TRZE TAC,4              ;*TEST AND CLEAR READ/WRITE BIT.
001145  364100  000000          SOJA TAC1,DFRED         ;*YES
001146  364100  000000          SOJA TAC1,DFWRT         ;*NO, WRITE.
```

```
                              ;SERVICE A SWAPPING INTERRUPT
                              EXTERNAL DINT4B,CKSMCT
                              INTERNAL SWPINT

ØØ1147  6Ø2ØØØ  7ØØØØØ  SWPINT: TRNE IOS,IODTER!IODERR!IOIMPM
ØØ115Ø  254ØØØ  ØØ1176'         JRST SWPERR             ;ERRORS
ØØ1151  62204Ø  4ØØØØØ          TRZE TAC,FRGSEG ;*FRAGMENTED?
ØØ1152  3311ØØ  ØØ134Ø'         SKIPL TAC1,SQREQ        ;*YES, MORE IOWD TO GO?
ØØ1153  254ØØØ  ØØ1157'         JRST DINT8B             ;NO, ALL DONE SWAP IN OR OUT
ØØ1154  35ØØØØ  ØØ1342'         AOS SERA                ;YES, FRAGMENTED AND MORE TO GO
ØØ1155  332ØØ1  ØØØØØ1          SKIPE 1(TAC)            ;IS THIS THE END OF SWAP COMMAND LIST?
ØØ1156  34404Ø  ØØ1115'         AOJA TAC,FRAGIO         ;NO, BO DO NEXT PIECE OF FRAGMENTED SEG

ØØ1157  60704Ø  4ØØØØØ  DINT8B: TLNN TAC,4ØØØØØ         ;*INPUT?
ØØ116Ø  254ØØØ  ØØ117Ø'         JRST DINT8A             ;*NO
ØØ1161  55ØØ4Ø  ØØ134Ø'         HRRZ TAC,SQREQ
ØØ1162  27104Ø  ØØ11Ø2'         ADDI TAC,JOBDAC
ØØ1163  5Ø5Ø4Ø  ØØ11Ø3'         HRLI TAC,MJOBCK
ØØ1164  26Ø14Ø  ØØ11Ø4'         PUSHJ PDP,CHECK
ØØ1165  21404Ø  ØØ11Ø5'         MOVM TAC,FINISH
ØØ1166  3121Ø1  ØØ11Ø6'         CAME TAC1,JBTCHK(TAC)
ØØ1167  254ØØØ  ØØ1173'         JRST SWPER1
ØØ117Ø  552ØØØ  ØØ1342' DINT8A: HRRZM IOS,SERA
ØØ1171  4Ø2ØØØ  ØØ134Ø'         SETZM SQREQ
ØØ1172  254ØØØ  ØØØØØØ          JRST DINT4B

ØØ1173  2Ø5Ø4Ø  ØØØ1ØØ  SWPER1: MOVSI TAC,1ØØ
ØØ1174  27204Ø  ØØØØØØ          ADDM TAC,CKSMCT
ØØ1175  66ØØØØ  1ØØØØØ          TRO IOS,IODTER
ØØ1176  21404Ø  ØØ1165' SWPERR: MOVM TAC, FINISH        ;*RESET SERA IN CASE OF FRAGMENTED JOB
ØØ1177  554Ø41  ØØ1Ø43'         HLRZ TAC, JBTSWP(TAC)   ;*SWAP LOC (DISK ADR OR TABLE ADR)
ØØ12ØØ  54204Ø  ØØ1342'         HRRM TAC, SERA          ;*RESTORE SERA
ØØ12Ø1  2ØØØ4Ø  ØØ1341'         MOVE TAC,ESQREQ         ;RESTORE ESQREQ IN CASE OF FRAGMENTED JOB
ØØ12Ø2  2Ø2Ø4Ø  ØØ134Ø'         MOVEM TAC,SQREQ
ØØ12Ø3  373ØØØ  ØØ1343'         SOSLE SERACT            ;*TRIED ENOUGH?
ØØ12Ø4  254ØØØ  ØØ11Ø7'         JRST SQGO1              ;*NO, TRY AGAIN
ØØ12Ø5  254ØØØ  ØØ117Ø'         JRST DINT8A             ;*YES, TOUGH.
```

```
        IFE     FTRC1Ø, <
        ;SWPSER LOGIC FOR THE OLD PDP-6 (DATA PRODUCTS) DISK FILE ---

        ;FIND A SERIES OF BLOCKS ON THE DISK TO SWAP ONTO. CALLED
        ;AT CLOCK LEVEL.
        ;CALL:  MOVEI DAT,NO. OF 1K BLOCKS DESIRED
        ;       PUSHJ PDP,GXSAT
        ;       ERROR EXIT      (DISK IS FULL)
        ;       NORMAL EXIT     ;C(TAC) = BLOCK NO.

        ;CONTENTS OF ACS TAC,TAC1,DAT WILL BE LOST.

        INTERNAL GXSAT
        EXTERNAL GETBIT,IPOPJ1

GXSAT:  MOVE AC1,XSAT1          ;SAVE AC1, SET IT TO TABLE LOC.
        MOVE AC2,XSAT2  ;
        LSH DAT,CONVMD          ;CONVERT TO 128 WORD DISK BLOCKS
        PUSH PDP,ITEM           ;SAVE C(ITEM)

        MOVE ITEM,DAT           ;GETBIT EXPECTS PARAMETER IN ITEM
        PUSHJ PDP,GETBIT        ;FIND A HOLE BIG ENOUGH
        JRST IPOPJ              ;NONE, RESTORE ITEM AND ERROR RETURN
        MOVEI TAC,-1(TAC1)
        CAIL TAC,BLOCKS         ;IS IT ON DISK 17?
        ADDI TAC,DIFF           ;YES
        LSH TAC,-CONVMD         ;CARRY DISK ADDRESS SHIFTED TO FIT IN JBTSWP.
        JRST IPOPJ1             ;SKIP RETURN AND RESTORE JOB NUMBER(ITEM)

        ;FREE UP A SERIES OF BLOCKS ON THE SWAPPING DEVICE. CALLED
        ;AT CLOCK LEVEL.
        ;CALL:  MOVEI DAT,NO. OF 1K CORE BLOCKS TO FREE
        ;       MOVE TAC,BLOCK NO. OF FIRST DISK BLOCK TO FREE
        ;       PUSHJ PDP,FXSAT
        ;       ALWAYS RETURN HERE

        ;CONTENTS OF ACS TAC,TAC1 WILL BE LOST.

        INTERNAL FXSAT
        EXTERNAL CLRBIT,IPOPJ

FXSAT:  LSH TAC,CONVMD          ;RESTORE SHIFTED DISK ADDRESS.
        CAIL TAC,HISWAP         ;ON DISK 17?
        SUBI TAC,DIFF           ;YES
        LSH DAT,CONVMD          ;CONVERT TO 128 WORD DISK BLOCKS
        MOVE AC1,XSAT1          ;SET UP AC1 WITH POINTER TO SAT BLOCK
        MOVE AC2,XSAT2          ;AC2 TOO
        PUSH PDP,ITEM           ;SAVE C(ITEM)
        MOVE ITEM,DAT           ;CLRBIT EXPECTS PARAMETER IN DAT
        PUSHJ PDP,CLRBIT
        JRST IPOPJ              ;RETURN, AND RESTORE ITEM

        ;INITIALIZE SWAPPER DISK STORAGE TABLE
```

```
INTERNAL SWPINI

SWPINI: MOVE TAC,XSAT2
        MOVEM TAC,XSAT31
        MOVSI TAC,1B18
        MOVEM TAC,XSAT3
        SETZM XSAT4
        MOVE TAC,XSAT4P
        BLT TAC,XSAT61
        MOVE TAC,XSAT7
        MOVEM TAC,XSAT5
        MOVE TAC,XSAT8
        MOVEM TAC,XSAT6
        POPJ PDP,

IFN FTCHECK+FTMONP,<
EXTERNAL SQREQ,SERA,SERACT,XSAT1,XSAT2,XSAT3,XSAT4,XSAT5,XSAT6,XSAT7
EXTERNAL XSAT8,SWPSIZ,HISWAP,DIFF,CONVMD,BLOCKS,XSAT31,XSAT4P,XSAT61
>
IFE FTCHECK+FTMONP,<

;DATA AND STORAGE AREA FOR SWAPPING, ON THE 27Ø DISK, DISKS
;       Ø & 17 ARE USED FOR SWAPPING, EACH DISK CONTAINING
;       54ØØ (OCTAL) RECORDS.

INTERNAL        SQREQ,SERA,SERACT
SQREQ:  Z                       ;C(LH)=NEG. OF SIZE OF READ/WRITE
                                ; C(RH)=LOC. OF FIRST WORD TO READ/WRITE
ESQREQ: Z                       ;COPY OF SQREQ, IN CASE OF
                                ; ERROR IN FRAGMENTED JOB
SERA:   Z                       ;SIGN IS 1 IF A READ
                                ; C(RH)=BLOCK NUMBER BEFORE READING,
                                ;  ERROR BITS AFTER READING.
SERACT: Ø                       ;COUNTER FOR ERRORS
SQLEN:  Ø                       ;AMOUNT TRANSFERRED SO FAR - FRAG SEG

XSAT1:  EXP XSAT3-1             ;POINTER USED BY GETBIT,CLRBIT
XSAT2:  XWD -SWPSIZ,XSAT4       ;POINTER TO BIT TABLE
XSAT3:  XWD 4ØØØØØ,             ;2-WORD TABLE ALTERED BY GETBIT,CLRBIT
        XWD -SWPSIZ,XSAT4

                                ; BIT TABLE FOR SWAPPING ALLOCATION
XP BLOCKS,54ØØ                  ;NUMBER OF RECORDS/DISK PLATE
        W1=BLOCKS/↑D36          ;NUMBER OF ZERO WORDS AT TOP OF TABLE
        E1=BLOCKS-W1*↑D36       ;LEADING ZEROES IN W1+1ST WORD
        EX=BLOCKS+E1-↑D35
        W2=EX/↑D36              ;ZERO WORDS AT BOTTOM OF TABLE
        E2=EX-W2*↑D36           ;LEADING ZEROES IN W1+W2+2ND WORD

XP XSAT31,XSAT3+1
XSAT4:  BLOCK W1
XSAT5:  BLOCK W2+1
XSAT6:  BLOCK 1
```

```

                        REPEAT 1,<
                                IFE E1,<X=1BØ>
                                IFN E1,<X=1
                                REPEAT ↑D35-E1,<X=X*2>>
                                IFE E2,<Z=-1>
                                IFN E2,<Z=Ø
                                Y=1
                                REPEAT ↑D36-E2,<Z=Z+Y
                                Y=Y*2>
                                >>
                XP XSAT61,XSAT6-1
                XSAT7:  EXP X
                XSAT8:  EXP Z

                SWPSIZ=XSAT6-XSAT4+1                    ;SIZE OF TABLE
                HISWAP=17*BLOCKS                        ;LOGICAL BLOCK NUMBER OF FIRST
                                                        ; BLOCK ON DISK
                DIFF=HISWAP-BLOCKS-1

                XP CONVMD,3     ;CONVERSION FROM 1K CORE BLOCKS TO 128 WORD
                                ;DISC BLOCKS(SHIFT COUNT)

                XP BLKSPK,CONVMD                        ;NO. OF BLOCKS PER K, SAME AS CONVMD
                XSAT4P: XWD XSAT4,XSAT4+1
                >
                >       ;END OF SWPSER LOGIC FOR THE OLD PDP-6 DISK.
```

```
                                    IFN     FTRC1Ø, <
                                    ;SWPSER LOGIC FOR THE NEW PDP-1Ø (MODEL RC-1Ø) DISK ---

                                    INTERNAL        GXSAT,FXSAT,SWPINI
                                    EXTERNAL        GETBIT,CLRBIT
                                    EXTERNAL        LBHIGH,IPOPJ,IPOPJ1


                                    ;SUBROUTINE "GXSAT" IS CALLED TO FIND A SERIES OF CONSECUTIVE FREE BLOCKS ON
                                    ; THE DISK TO SWAP SOME JOB OUT ONTO.  IT IS CALLED AT CLOCK LEVEL.

                                    ;CALLING SEQUENCE ---
                                    ;       PUSHJ   PDP,GXSAT
                                    ;       ERROR EXIT -- THE DISK IS FULL, NO SWAPPING SPACE AVAILABLE.
                                    ;       NORMAL EXIT
                                    ;ENTRY CONDITIONS ---
                                    ;       C(DAT) = NUMBER OF 1K BLOCKS OF DISK STORAGE NEEDED.
                                    ;EXIT CONDITIONS ---
                                    ;       C(TAC) = LOGICAL BLOCK NUMBER (DIVIDED BY 8) OF THE FIRST DISK BLOCK IN
                                    ;                THE SERIES OF CONSECUTIVE BLOCKS WHICH SATISFY THIS REQUEST.
                                    ;ACCUMULATORS DAT, TAC, TAC1, AC1, AND AC2 ARE DESTROYED BY THIS SUBROUTINE.

ØØ12Ø6  26114Ø  ØØØØ24      GXSAT:  PUSH    PDP,ITEM        ;THIS ROUTINE SAVES AND RESTORES ACCUMULATOR "ITEM".
ØØ12Ø7  2ØØ2ØØ  ØØØØØ5              MOVE    ITEM,DAT
ØØ121Ø  20164Ø  ØØ13Ø1'             MOVEI   AC1,SWPENT      ;SET UP ENTRY CONDITIONS FOR THE "GETBIT" SUBROUTINE.
ØØ1211  2ØØ7ØØ  ØØ13Ø1'             MOVE    AC2,SWPENT
ØØ1212  26Ø14Ø  ØØØØØØ              PUSHJ   PDP,GETBIT
ØØ1213  254ØØØ  ØØ1Ø64'             JRST    IPOPJ           ;NO ROOM IN THE SWAPPING PART OF THE DISK.
ØØ1214  2711Ø4  777777              ADDI    TAC1,-1(ITEM)   ;ROOM FOUND--COMPUTE LOGICAL BLOCK NUMBER OF THIS
ØØ1215  3131ØØ  ØØ13ØØ'             CAMLE   TAC1,MAXSWP     ;FIRST BLOCK IN A SERIES OF CONSECUTIVE FREE DISK
ØØ1216  2021ØØ  ØØ13ØØ'             MOVEM   TAC1,MAXSWP     ;BLOCKS.  THE "SWPTAB" TABLE SEARCHED HAS ONE BIT
ØØ1217  2421ØØ  ØØØØØ3              LSH     TAC1,BLKSPK     ;FOR EACH 1K OF SWAPPING DISK SPACE, BUT IS STORED
ØØ122Ø  2741ØØ  ØØØØØØ              SUB     TAC1,LBHIGH     ;BACKWARDS (1ST BIT OF SWPTAB = LAST  1K OF DISK),
ØØ1221  21404Ø  ØØØØØ2              MOVM    TAC,TAC1        ;ALSO UPDATE "MAXSWP" IF PREVIOUS MAXIMUM EXCEEDED.
ØØ1222  24204Ø  777775              LSH TAC,-BLKSPK         ;COMPRESS DISK ADDRESS TO 17 BITS FOR LH OF JBTSW
ØØ1223  34404Ø  ØØØØØØ              AOJA TAC,IPOPJ1         ;ADD 1 TO TAC AND RESTORE ITEM AND SKIP RETURN
```

```
                                        ;SUBROUTINE "FXSAT" IS CALLED TO RETURN A SERIES OF CONSECUTIVE DISK BLOCKS TO
                                        ; THE FREE STORAGE POOL, THUS MAKING THEM AVAILABLE FOR RE-USE IN HANDLING
                                        ; FUTURE SWAPPING REQUESTS.  IT IS CALLED AT CLOCK LEVEL.

                                        ;CALLING SEQUENCE ---
                                        ;       PUSHJ   PDP,FXSAT
                                        ;       NORMAL EXIT
                                        ;ENTRY CONDITIONS ---
                                        ;       C(TAC) = LOGICAL BLOCK NUMBER OF THE FIRST DISK BLOCK IN THE
                                        ;                SERIES WHICH IS TO BE MADE AVAILABLE.
                                        ;       C(DAT) = NUMBER OF CONSECUTIVE 1K BLOCKS OF DISK SPACE WHICH
                                        ;                ARE TO BE MADE AVAILABLE.
                                        ;EXIT CONDITIONS ---
                                        ;       THE REQUESTED BITS IN THE SWAPPING SPACE AVAILABILITY TABLE
                                        ;                (NAMELY, SWPTAB) HAVE BEEN CLEARED TO ZERO.
                                        ;ACCUMULATORS DAT, TAC, TAC1, AC1, AND AC2 ARE DESTROYED BY THIS SUBROUTINE.

ØØ1224  626240  4ØØØØØ  FXSAT:  TRZN    TAC,FRGSEG      ;FRAGMENTED?
ØØ1225  254ØØØ  ØØ1247'         JRST    FXSAT1          ;NO. DO IN REGULAR WAY

ØØ1226  55Ø74Ø  ØØØØØ1  FRAGBK: HRRZ    AC3,TAC         ;YES. LOC OF TABLE IN AC3
ØØ1227  55ØØ57  ØØØØØØ  FRGBK1: HRRZ    TAC,(AC3)       ;LOC OF NEXT DISK ADDRESS
ØØ123Ø  574257  ØØØØØØ          HLRE    DAT,(AC3)       ;NUMBER OF K
ØØ1231  32324Ø  ØØ1234'         JUMPLE  DAT,FRGBK2      ;GIVE UP FREE CORE IF NOT REAL ADDRESS
ØØ1232  26Ø14Ø  ØØ1247'         PUSHJ   PDP,FXSAT1      ;GIVE UP THE DISK SPACE FOR THIS PART
ØØ1233  25274Ø  ØØ1227'         AOBJP   AC3,FRGBK1      ;COUNT WORD OF TABLE. GET NEXT
ØØ1234  55Ø04Ø  ØØØØ17  FRGBK2: HRRZ    TAC,AC3         ;LOC OF TABLE
ØØ1235  26114Ø  ØØØØØ4          PUSH    PDP,ITEM
ØØ1236  5542ØØ  ØØØØ17          HLRZ    ITEM,AC3        ;NUMBER OF WDS IN TABLE
ØØ1237  27404Ø  ØØØØØ4          SUB     TAC,ITEM        ;POINT TAC TO 1ST WORD OF BLOCK
ØØ124Ø  2422ØØ  777776          LSH     ITEM,-2         ;4 WDS PER BIT
ØØ1241  35ØØØØ  ØØØØØ4          AOS     ITEM
ØØ1242  26Ø14Ø  ØØØØØØ          PUSHJ   PDP,CLCOR1      ;GIVE UP FREE CORE
ØØ1243  26214Ø  ØØØØØ4          POP     PDP,ITEM
ØØ1244  332Ø57  ØØØØØØ          SKIPE   TAC,(AC3)       ;END OF TABLE?
ØØ1245  254ØØØ  ØØ1226'         JRST    FRAGBK          ;NO. GO CHASE NEXT PART
ØØ1246  26314Ø  ØØØØØØ          POPJ    PDP,            ;YES. DONE

ØØ1247  26114Ø  ØØØØØ4  FXSAT1: PUSH    PDP,ITEM        ;THIS ROUTINE SAVES AND RESTORES ACCUMULATOR "ITEM".
ØØ125Ø  2ØØ2ØØ  ØØØØØ5          MOVE    ITEM,DAT
ØØ1251  24204Ø  ØØØØØ3          LSH TAC,BLKSPK          ;REGENERATE THREE LOW ORDER Ø BITS IN DISK ADDRESS.
ØØ1252  27404Ø  ØØ122Ø'         SUB     TAC,LBHIGH      ;PREPARE TO CLEAR BITS IN THE SWAPPING
ØØ1253  217ØØØ  ØØØØØ1          MOVMS   TAC             ; AVAILABILITY TABLE BY TRANSFORMING THE LOGICAL
ØØ1254  27104Ø  ØØØØØ1          ADDI    TAC,1           ; BLOCK NUMBER TO AN EQUIVALENT BIT NUMBER IN THIS
ØØ1255  21164Ø  ØØØØØ3          MOVNI   AC1,BLKSPK      ; TABLE (WHICH IS STORED IN REVERSE ORDER AND HAS
ØØ1256  242Ø55  ØØØØØØ          LSH     TAC,(AC1)       ; ONE BIT PER 1K OF SWAPPING SPACE ON THE DISK).
ØØ1257  27404Ø  ØØØØØ5          SUB     TAC,DAT
ØØ126Ø  20164Ø  ØØ13Ø1'         MOVEI   AC1,SWPENT
ØØ1261  20072Ø  ØØ13Ø1'         MOVE    AC2,SWPENT      ;SET UP ENTRY CONDITIONS FOR THE "CLRBIT" SUBROUTINE.
ØØ1262  26Ø14Ø  ØØØØØØ  FXSATC: PUSHJ   PDP,CLRBIT      ;THE "CLRBIT" SUBROUTINE IN "DSKSER" ACTUALLY DOES
                                                        ; THE WORK OF CLEARING THE BITS.
ØØ1263  254ØØØ  ØØ1213'         JRST IPOPJ              ;RESTORE ITEM AND RETURN
```

```

                                 ;ROUTINE TO RE-INITIALIZE THE SWAPPING AVAILABILITY TABLE.  CALLED AT
                                 ; SYSTEM INITIALIZATION TIME.

                                         EXTERN VIRTAL,SWPHGH

001264  200040  001252' SWPINI: MOVE TAC,LBHIGH          ;SET HIGHEST LOGICAL BLOCK
001265  202040  000000          MOVEM TAC,SWPHGH         ;FOR SWAPPING IN COMMON
001266  211040  000001          MOVNI   TAC,1            ;SET ENTIRE TABLE TO ONES, I.E., MARK EVERYTHING
001267  200100  001301'         MOVE    TAC1,SWPENT      ; AS BEING UNAVAILABLE FOR SWAPPING.
001270  202042  000000          MOVEM   TAC,(TAC1)
001271  253100  001270'         AOBJN   TAC1,.-1
001272  201640  001301'         MOVEI   AC1,SWPENT       ;THEN USE THE "CLRBIT" ROUTINE TO CLEAR OUT AS MANY
001273  200700  001301'         MOVE    AC2,SWPENT       ; BITS AT THE FRONT OF THE TABLE AS THERE ARE 1K DISK
001274  200200  000000          MOVE    ITEM,K4SWAP      ; AREAS ALLOCATED FOR SWAPPING USE.  THE PARAMETER
001275  202200  001035'         MOVEM ITEM,VIRTAL        ;TOTAL AMOUNT OF VIRTUAL CORE ALLOWED
                                                         ; IS JUS EQUAL TO AMOUNT OF SWAPPING SPACE
                                                         ; EVEN IF USER CAN NOT HAVE ALL OF PHYSICAL USER CORE
```

```
Ø01276  2Ø1Ø4Ø  ØØØØØØ          MOVEI   TAC,Ø           ; "K4SWAP" IS SET UP DURING DISK REFRESHING, AND
                                                        ; RECALLED FROM THE DISK AT EACH SYSTEM-INITIALIZE.
Ø01277  254ØØØ  ØØ1262'         JRST    CLRBIT          ;IN LIEU OF "PUSHJ CLRBIT" FOLLOWED BY "POPJ".
```

```
                               IFN     FTCHECK+FTMONP, <
                               EXTERNAL        SCYSDT  ;DUMMY GLOBAL SYMBOL TO PERMIT THE SYSTEM BUILDER TO
                                                       ; RETRIEVE THE CORRECT BINARY COPY OF "SCHDAT".
                               EXTERNAL        SQREQ,SERA,SERACT
                               EXTERNAL        LBHIGH,BLKSPK,MXK2SWP,SWPENT,SWPTAB,MAXSWP
                               >

                               IFE     FTCHECK+FTMONP, <
                               ;DATA ASSOCIATED WITH THE SWPSER LOGIC FOR THE NEW PDP-1Ø DISK ---

                                        INTERN  MXK2SWP,CONVMD,BLKSPK,SWPTAB,MAXSWP
                                        EXTERN K4SWAP
                               ;THE ABOVE ARE REFERENCED OR INITIALIZED BY THE "ONCE" ROUTINE.

                               ;C(LOWSWP)=LOWEST LOGICAL BLOCK NUMBER TO BE USED FOR SWAPPING
                               ;DETERMINED BY ONCE ONLY REFRESH DIALOG
                               ;C(K4SWAP)=MAX. NO. OF K DISK WORDS ALLOCATED FRO SWAPPING, ALSO
                               ;DETERMINED BY ONCE ONLY REFRESH DIALOG
ØØ13ØØ  ØØØØØØ  ØØØØØØ  MAXSWP: Ø           ;MAX. NO. OF K FROM TOP TO LOWEST JOB
                                                       ; USED SINCE SYSTEM BEGAN
                               ;C(MAXSWP)=MAX. NO. OF K EVER ASSIGNED COUNTING FROM TOP DOWN TO
                               ;LOWEST EXCURSION, INCLUDING HOLES. WHEN = C(K4SWAP), THEN FRAGMENTATION HAD
                               ;HAPPENED SOMETIME DURING THE PAST
                ØØØØØ3  BLKSPK=3            ;SHIFT FACTOR TO CONVERT K OF CORE TO DISK BLOCKS, I.E.,
                                            ; EIGHT DISK BLOCKS PER K OF CORE.
                ØØØØØ3  CONVMD=BLKSPK       ;ONLY FOR CONSISTENCY WITH OLDER PDP-6 ROUTINE.
                ØØ175Ø  MXK2SWP=↑D1ØØØ      ;MAXIMUM NUMBER OF 1K DISK BLOCKS WHICH MIGHT BE ALLOCATED
                                            ; FOR SWAPPING (UPPER BOUND ON THE VALUE OF K4SWAP  WHICH
                                            ; MAY BE REQUESTED AT DISK REFRESH TIME). (ONE MILLION WORDS
                                            ; FOR SWAPPING SEEMS LIKE A NON-RESTRICTIVE ARBITRARY LIMIT.)
                ØØØØ34  SWPSIZ=MXK2SWP/↑D36+1   ;SIZE OF SWPTAB ALLOCATION TABLE.

ØØ13Ø1  777744  ØØ13Ø4' SWPENT: XWD     -SWPSIZ,SWPTAB  ;THREE WORD POINTER TABLE
ØØ13Ø2  4ØØØØØ  ØØØØØØ          XWD     4ØØØØØ,Ø        ; REQUIRED BY THE "GETBIT" AND
ØØ13Ø3  777744  ØØ13Ø4'         XWD     -SWPSIZ,SWPTAB  ; "CLRBIT" SUBROUTINES.

ØØ13Ø4                  SWPTAB: BLOCK   SWPSIZ  ;SWAPPING SPACE AVAILABILITY TABLE.


                        ;DATA CARRIED OVER FROM THE COMMON PART OF OLD AND NEW SWPSER ROUTINES ---
                        INTERNAL        SQREQ,SERA,SERACT,ESQREQ
ØØ134Ø  ØØØØØØ  ØØØØØØ  SQREQ:  Ø                       ;C(LH)=NEG. OF SIZE OF READ/WRITE
                                                        ; C(RH)=LOC. OF FIRST WORD TO READ/WRITE
                        ESQREQ: Ø                       ;COPY OF SQREQ IN CASE OF SWAP ERROR ON FRAGMENTED JOB
ØØ1341  ØØØØØØ  ØØØØØØ
ØØ1342  ØØØØØØ  ØØØØØØ  SERA:   Ø                       ;SIGN IS 1 IF A READ
                                                        ; C(RH)=BLOCK NUMBER BEFORE READING,
                                                        ;  ERROR BITS AFTER READING.
ØØ1343  ØØØØØØ  ØØØØØØ  SERACT: Ø                       ;COUNTER FOR ERRORS
ØØ1344  ØØØØØØ  ØØØØØØ  SQLEN:  Ø                       ;AMOUNT TRANSFERRED SO FAR FOR FRAGMENTED JOB
                        >
                        >       ;END OF SWPSER LOGIC FOR THE NEW PDP-1Ø DISK.
```

```
  1762  ØØ1345                    SCHEND: END

NO ERRORS DETECTED

PROGRAM BREAK IS ØØ1345
```

```
SCHEDB - SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(1Ø/5Ø)(BURROUGHS DISK)       MACRO.V36 19:1Ø  4-JUN-69 PAGE 64
        SYMBOL TABLE

AC1              ØØØØ15   INT       AC2              ØØØØ16   INT       AC3              ØØØØ17   INT
ANYDEV           ØØØ666'  EXT       AUAVAL           ØØØ25Ø'  INT       AJQ              ØØØØØ4   INT
AUREQ            ØØØ3Ø1'  INT       AVALTB           ØØØ244'  INT       AVLNUM           ØØØØ11   INT
AVLQTB           ØØØ3Ø7'  INT       BIGHOL           ØØØ565'  EXT       BLKSPK           ØØØØØ3   INT
BMPAC3           ØØØ76Ø'            BQFIX            ØØØ2Ø5'            BQJSIZ           ØØØ172'
BQLINK           ØØØ164'            CERROR           ØØØ75Ø'  EXT       CHECK            ØØ1164'  FXT
CHG1             ØØ1Ø35'            CHG3             ØØ1Ø45'            CHGSWP           ØØ1Ø16'  INT
CHKSHF           ØØØ571'  EXT       CHKXPN           ØØØ6Ø6'            CKJB1            ØØØØ41'
CKJB1Ø           ØØØ132'            CKJB2            ØØØØ44'            CKJB3            ØØØØ51'
CKJB4            ØØØØ66'            CKJB4A           ØØØØ67'            CKJB4B           ØØØØ65'
CKJB5            ØØØØ73'            CKJB6            ØØØØ74'            CKJB6A           ØØØ1ØØ'
CKJB7            ØØØ112'            CKJB8            ØØØ111'            CKJB9            ØØØØ57'
CKJBT            ØØØ137'            CKSMCT           ØØ1174'  EXT       CLCOR1           ØØ1242'  FXT
CLRBIT           ØØ1277'  EXT       CMQ              ØØØØ22             CMWB             2ØØØØØ   INT
CONVMD           ØØØØØ3   INT       CORGET           ØØ1ØØ1'  EXT       CORTAL           ØØØ624'  FXT
CPOPJ            ØØ1Ø73'  EXT       DAAVAL           ØØØ252'  INT       DAQ              ØØØØØ6   INT
DAREQ            ØØØ3Ø3'  INT       DAT              ØØØØØ5   INT       DCAVAL           ØØØ254'  INT
DCQ              ØØØØ1Ø   INT       DCREQ            ØØØ3Ø5'  INT       DEVDAT           ØØØØØ6   INT
DFBUSY           ØØ1Ø72'  EXT       DFREQ            ØØ1145'  EXT       DFWRT            ØØ1146'  FXT
DINT4B           ØØ1172'  EXT       DINT8A           ØØ117Ø'            DINT8B           ØØ1157'
DTAVAL           ØØØ253'  INT       DTQ              ØØØØØ7   INT       DTREQ            ØØØ3Ø4'  INT
EQFIX    4ØØØØØ  ØØØ2Ø5'            EQJSIZ   4ØØØØØ  ØØØ172'            EQLINK   4ØØØØØ  ØØØ164'
ERATRY           ØØØØØ3             ERROR            ØØØ62Ø'  EXT       ERRPNT           ØØØ531'  FXT
ESQREQ           ØØ1341'  INT       EXCALP           ØØØØØØ   EXT       FININ            ØØØ5ØØ'  FXT
FININØ           ØØØ52Ø'            FINISH           ØØ1176'  EXT       FINOT            ØØØ544'  FXT
FINOUT           ØØØ537'            FIT              ØØØ773'  EXT       FITØ             ØØØ551'
FIT1             ØØØ561'            FITHGH           ØØ1ØØ3'  EXT       FITSIZ           ØØØ563'  FXT
FORCE            ØØØ67Ø'  EXT       FORCFØ           ØØØ655'            FORCE1           ØØØ663'
FORCE2           ØØØ651'            FORCEL           ØØØ662'            FORHGH           ØØØ657'  FXT
FORSIZ           ØØØ644'  EXT       FRAGBK           ØØ1226'            FRAGIO           ØØ1115'
FRECR1           ØØØ57Ø'  EXT       FRESWP           ØØØ744'  EXT       FRGBK1           ØØ1227'
FRGBK2           ØØ1234'            FRGIO1           ØØ112Ø'            FRGIO2           ØØ1127'
FRGSFG           4ØØØ2Ø   INT       FT2REL   777777  777777   INT       FTCCL    777777  777777
FTCHEC           ØØØØØØ   INT       FTDISK   777777  777777   INT       FTLOGI   777777  777777
FTMONP           ØØØØØØ   INT       FTRC1Ø   777777  777777   INT       FTRCHK   777777  777777
FTSWAP   777777  777777   INT       FTTRAC           ØØØØØØ   INT       FULCNT           ØØØ2ØØ   FXT
FULCOR           ØØØ755'            FULL             ØØØ717'            FULL1            ØØØ726'
FULL1A           ØØØ727'            FULL1B           ØØØ741'            FULL2            ØØØ742'
FULSET           ØØØ751'            FXSAT            ØØ1224'  INT       FXSAT1           ØØ1247'
FXSATC           ØØ1262'            GETBIT           ØØ1212'  EXT       GETFCR           ØØØ756'  FXT
GXSAT            ØØ1276'  INT       HIGHJB           ØØØ612'  EXT       HOLEF            ØØØ567'  FXT
ICPROT           ØØØØØ3   INT       ICPRT1           ØØØØØ6   INT       IMGIN            ØØ1Ø54'  FXT
IMGOUT           ØØ1Ø47'  EXT       INERR            ØØØ526'            INFLG            ØØØ141'
IODERR           2ØØØØØ   INT       IODTER           1ØØØØØ   INT       IOIMPM           4ØØØØØ   INT
IOS              ØØØØØØ   INT       IOWQ             ØØØØ12   INT       IPOPJ            ØØ1263'  FXT
IPOPJ1           ØØ1223'  EXT       ISCAN            ØØØ42Ø'  INT       ITEM             ØØØØØ4   INT
J                ØØØØØ4             JBTADR           ØØ1Ø13'  EXT       JBTCHK           ØØ1166'  FXT
JBTDAT           ØØØ514'  EXT       JBTQ             ØØØ241'  EXT       JBTQM1           ØØØ155'  FXT
JBTQMN           ØØØ154'  EXT       JBTQP1           ØØØ16Ø'  EXT       JBTSTS           ØØ1Ø63'  FXT
JBTSWP           ØØ1177'  EXT       JDAT             ØØØØØ7   INT       JOB              ØØ1Ø57'  FXT
JOBDAC           ØØ1162'  EXT       JOBDPD           ØØØ523'  EXT       JOBDPG           ØØØ524'  FXT
JOBMAX           ØØ1Ø56'  EXT       JOBPC            ØØØ515'  EXT       JOBQUE           ØØØ321'  INT
JRQ              ØØØØØ2   INT       JXPN             ØØØØØ1   INT       K4SWAP           ØØ1274'  FXT
```

```
SCHEDB - SCHEDULING ALGORITHM FOR SWAPPING SYSTEM(1Ø/52)(BURROUGHS DISK)        MACRO.V36 19:12  4-JUN-69 PAGE 64-1
        SYMBOL TABLE

KCORE1          ØØØ543' EXT         LBHIGH          ØØ1264' FXT         LOC             ØØØØ22
MAXJBN          ØØØ654' EXT         MAXQ            ØØØØ11  INT         MAXSIZ          ØØØ647' FXT
MAXSWP          ØØ137Ø' INT         MINQ            ØØØØØ3  INT         MJOBCK          ØØ1163' FXT
MJOBN           ØØØØ12' EXT         MQAVAL          ØØØ251' INT         MQQ             ØØØØØ5  INT
MQREQ           ØØØ3Ø2' INT         MTAVAL          ØØØ255' INT         MTQ             ØØØØ11  INT
MTREQ           ØØØ3Ø6' INT         MXCODE          ØØØØ16  INT         MXK2SW          ØØ175Ø  INT
MXQUE           ØØØ142' EXT         NOFIT           ØØØ714'             NQUEUE          ØØØØ12
NSWP            Ø1ØØØØ  INT         NULQ            ØØØØ15  INT         NXTINI          ØØØØØØ' INT
NXTJB1          ØØØØ22'             NXTJOB          ØØØØ1Ø' INT         OERROR          ØØ1ØØ2' FXT
OSCAN           ØØØ436' INT         OUTP2           ØØØ711'             PC              ØØØØØ1
PCORSZ          ØØØ51Ø' EXT         PCSTOP          ØØØ536' EXT         PDP             ØØØØØ3  INT
PJBSTS          ØØØ223' EXT         POTLST          ØØØ126' FXT         PQ1             ØØØØ17
PQ2             ØØØØ2Ø              PQ3             ØØØØ21              PROG            ØØØØØ7  INT
PROT            ØØØ511' EXT         PROT2           ØØØ512' EXT         Q               ØØØØØ7
QAUS            ØØØ4ØØ'             QAUW            ØØØ344'             QBAK            ØØØ241'
QBAK1           ØØØ235'             QBITS           ØØØ256' INT         QCMW            ØØØ33Ø' INT
QDAS            ØØØ366'             QDAW            ØØØ35Ø'             QDCS            ØØØ37Ø'
QDCW            ØØØ352'             QDTS            ØØØ374'             QDTW            ØØØ356'
QFIX            ØØØ275'             QFOR            ØØØ232'             QFOR1           ØØØ231'
QFOR2           ØØØ234'             QINI            ØØØ142' INT         QINI1           ØØØ155'
QIOWW           ØØØ34Ø'             QJ              ØØØØØ6              QJOB            ØØØ322' INT
QJSIZ           ØØØ172'             QLINK           ØØØ164'             QMQS            ØØØ364'
QMQW            ØØØ346'             QMTS            ØØØ376'             QMTW            ØØØ36Ø'
QNULW           ØØØ324'             QQSD            ØØØØØ6              QQSTAB          ØØØ414'
QQTTY           ØØØØØ6              QR              ØØØØØ2              QRNS            ØØØØØØ
QRNW            ØØØ332'             QSCAN           ØØØ225' INT         QSLPW           ØØØ362'
QSTAR           ØØØ4Ø4'             QSTOP           ØØØ326' INT         QSTOPW          ØØØ326'
QSTS            ØØØ372'             QSTW            ØØØ354'             QTIME           ØØØ4Ø2' INT
QTIOWW          ØØØ342'             QTSS            ØØØØØØ              QTSW            ØØØ336'
QTTAB           ØØØ41Ø'             QWSS            ØØØØØØ              QWSW            ØØØ334'
QX1             ØØØ175'             QX2             ØØØ2Ø1'             QX3             ØØØ224'
QXFER           ØØØ162' INT         RCXSKD          ØØØØØØ' INT         REQTAB          ØØØ275' INT
RNAVAL          ØØØ244' INT         RNQ             ØØØØØØ  INT         RNQUNT          ØØØ414' INT
RNREQ           ØØØ275' INT         RUNABL          44ØØØØ  INT         RUNMSK          1114Ø4  INT
SCHD1           ØØØ13Ø'             SCHED           ØØØ113'             SCHEND          ØØ1345'
SCNJOB          ØØØ624'             SCNOK           ØØØ621'             SCNOUT          ØØØ61Ø'
SERA            ØØ1342' INT         SERACT          ØØ1343' INT         SHF             ØØ4ØØØ  INT
SHFWAT          ØØØ572' EXT         SLPQ            ØØØØ14  INT         SQGO            ØØ1Ø74' INT
SQGO1           ØØ11Ø7' INT         SQGO2           ØØ1142'             SQIN            ØØ1Ø65' INT
SQLEN           ØØ1344'             SQOUT           ØØ1Ø66' INT         SQREQ           ØØ134Ø' INT
SSCAN           ØØØ43Ø' INT         ST              ØØØØØ5              STAVAL          ØØØ247' INT
STOPQ           ØØØØ16  INT         STQ             ØØØØØ3  INT         STREQ           ØØØ3ØØ' INT
SUMCOR          ØØØ652' EXT         SW              ØØØØØ2              SWAP            ØØØ465' INT
SWAPI           ØØØ772'             SWAPO           ØØØ67Ø'             SWP             ØØ2ØØØ  INT
SWP1            ØØØ546'             SWP2            ØØØ547'             SWPENT          ØØ13Ø1'
SWPER1          ØØ1173'             SWPERC          ØØØ6Ø4' EXT         SWPERR          ØØ1176'
SWPHGH          ØØ1265' EXT         SWPINI          ØØ1264' INT         SWPINT          ØØ1147' INT
SWPREC          ØØØ575'             SWPSIZ          ØØØØ34              SWPTAB          ØØ13Ø4' INT
T               ØØØØØ6              T1              ØØØØØ2              T2              ØØØØØ1
TAC             ØØØØØ1  INT         TAC1            ØØØØØ2  INT         TIMEF           ØØØØ31' FXT
TIOWQ           ØØØØ13  INT         TRYSWP          ØØØ655' EXT         TSAVAL          ØØØ246' INT
TSQ             ØØØØØ2  INT         TSREQ           ØØØ277' INT         TT              ØØØØØ5
USRMOD          Ø1ØØØØ  INT         VIRTAL          ØØ1275' EXT         VSCHED          ØØØ421  INT
```

```
        SYMBOL TABLE

WSAVAL          ØØØ245' INT        WSC             ØØØØ31  INT        WSREQ           ØØØ276' INT
WTMASK          ØØØ37Ø  INT        XCKCSW          ØØØØØØ' INT        XJOB            ØØØ323' INT
XPAND           ØØ1Ø52' INT        XPANDH          ØØ1Ø55' INT        ZERSWP          ØØ1Ø34' INT
```

```
A            6#      6
AC1          6#      6     955     97Ø    125Ø    1624    1679    168Ø    1682    17Ø1
AC2          6#      6    1625    1683    17Ø2
AC3          6#   1147    1149    115Ø    1151    1157    1159    1184    1194    1196    1197    1199    12ØØ    12Ø3
          12Ø4    1655    1656    1657    166Ø    1661    1663    1669
AEFERR       6#      6
AL           6#      6
ANYDEV     837    1143
ASSCON       6#      6
ASSPRG       6#      6
AUAVAL     49Ø     49Ø#
AUQ        49Ø     49Ø#    624     678     763     782     791
AUREQ      574#    575
AVALTB      54     135     146     483     485#    486     487     488     489     49Ø     491     492     493     494
           495     496
AVLNUM     134     5Ø2#    5Ø2
AVLQTB     138     145     598     6ØØ#
B            6#      6
BIGHOL     837     97Ø
BLKSPK    1427    1631    1634    1675    1679    172Ø    1733#   1735
BMPAC3    1162    1183    1194#
BQFIX      395#    632#    663     668     7Ø6     7Ø9     712     715     718     721     724
BQJSIZ     399#    636#
BQLINK     397#    634#
BUFPNT       6#      6
BUFWRD       6#      6
CERROR    1176    118Ø
CHECK     1357    139Ø    1449
CHG1      1251    128Ø#
CHG3      1288    1291#
CHGSWP    1247    125Ø#
CHKSHF     84Ø     854     983
CHKXPN     947    1ØØ1#
CKJB1       86      93      97     1Ø1#
CKJB1Ø     12Ø     183#
CKJB2      1Ø4#    132
CKJB3       91     111#
CKJB4      117     126#
CKJB4A     116     127#    187
CKJB4B     125#    185
CKJB5      1Ø2     127     134#
CKJB6      135#    136     15Ø
CKJB6A     139#    143
CKJB7      137     151#
CKJB8      14Ø     15Ø#
CKJB9      114     117#
CKJBT      186     188#
CKSMCT    1432    1458
CLCOR1    14ØØ    1667
CLKR         6#      6
CLRBIT    16Ø4    1684    17Ø9
CLSIN        6#      6
CLSOUT       6#      6
```

```
CMQ         520#     655      76Ø
CMWB          6#       6       89      113
CONVMD     172Ø     1735#
CORCNT        6#       6
CORGET      839     1222
CORTAL      837     1Ø27
CPOPJ       23Ø      248     1194     12Ø4     1357     1373
D             6#       6
DAAVAL      492      492#
DAQ         492      492#     6Ø6      684      762      783      792
DAREQ       576#     577
DAT           6#       6       78       79       98      112      125      126      141      142      145      162      186      34Ø
            364      435      945     1Ø17     1Ø3Ø     1125     1126     1128     1147     1149     1154     1158     1159     116Ø
           1163     1165     1168     1169     118Ø     1223     1286     1287     1288     1291     14Ø2     14Ø3     14Ø4     14Ø5
           14Ø7     14Ø8     14Ø9     1413     1414     1415     1416     1417     1418     1421     1422     1423     1425     1623
           1657     1658     1674     1681
DCAVAL      494      494#
DCL           6#       6
DCLI          6#       6
DCLO          6#       6
DCLR          6#       6
DCQ         494      494#     6Ø8      687      764      781      79Ø
DCREQ       578#     579
DDI           6#       6
DDO           6#       6
DEN           6#       6
DEVADR        6#       6
DEVBUF        6#       6
DEVCHR        6#       6
DEVCTR        6#       6
DEVDAT        6#       6       43      846
DEVEXT        6#       6
DEVFIL        6#       6
DEVIAD        6#       6
DEVIOS        6#       6
DEVLOG        6#       6
DEVMOD        6#       6
DEVNAM        6#       6
DEVOAD        6#       6
DEVPPN        6#       6
DEVPTR        6#       6
DEVSER        6#       6
DFBUSY     1357     1372
DFRED      1357     1429
DFWRT      1357     143Ø
DGF           6#       6
DHNG          6#       6
DIN           6#       6
DINI          6#       6
DINT4B     1432     1455
DINT8A     1445     1453#    1467
DINT8B     1439     1444#
DLK           6#       6
```

```
DMT          6#       6
DNAERR       6#       6
DOU          6#       6
DR           6#       6
DRL          6#       6
DRN          6#       6
DSER         6#       6
DSI          6#       6
DSKRLB       6#       6
DSO          6#       6
DTAVAL     493      493#
DTQ        493      493#     607      693      767      778      787
DTREQ      577#     578
DVAVAL       6#       6
DVCDR        6#       6
DVDIR        6#       6
DVDIRI       6#       6
DVDIS        6#       6
DVDSK        6#       6
DVDTA        6#       6
DVIN         6#       6
DVLNG        6#       6
DVLPT        6#       6
DVMTA        6#       6
DVOUT        6#       6
DVPTP        6#       6
DVPTR        6#       6
DVTTY        6#       6
ENTRS        6#       6
EQFIX      188      396#     633#     648      651      654      672      675      677      68Ø      683      686      689      692
           695      698
EQJSIZ     4ØØ#     637#     657
EQLINK     398#     635#     727
ERATRY    1376#    1382
ERROR      346      364     1ØØ6     1Ø17
ERRPNT     841      915
ESQREQ    137Ø     1463     175Ø     1753#
EXCALP     841
FBMERR       6#       6
FININ      867      868
FININØ     865#    122Ø     1229
FINISH     838      86Ø      911      939     1132     1214     1391     145Ø     146Ø
FINOT      933      934
FINOUT     862      918#
FIT        838      942      954     1Ø33     1138     1215
FITØ       942#     985
FIT1       869      943      954#
FITHGH    1225     1226
FITSIZ     961      962
FNFERR       6#       6
FORCE      838      94Ø     1Ø92     1113
FORCEØ    1Ø21     1Ø63#
FORCE1     941     1Ø94#
```

```
DMT          6#      6
DNAERR       6#      6
DOU          6#      6
DR           6#      6
DRL          6#      6
DRN          6#      6
DSER         6#      6
DSI          6#      6
DSKRLB       6#      6
DSO          6#      6
DTAVAL     493     493#
DTQ        493     493#    607     693     767     778     787
DTREQ      577#    578
DVAVAL       6#      6
DVCDR        6#      6
DVDIR        6#      6
DVDIRI       6#      6
DVDIS        6#      6
DVDSK        6#      6
DVDTA        6#      6
DVIN         6#      6
DVLNG        6#      6
DVLPT        6#      6
DVMTA        6#      6
DVOUT        6#      6
DVPTP        6#      6
DVPTR        6#      6
DVTTY        6#      6
ENTRS        6#      6
EQFIX      188     396#    633#    648     651     654     672     675     677     68Ø     683     686     689     692
           695     698
EQJSIZ     4ØØ#    637#    657
EQLINK     398#    635#    727
ERATRY    1376#   1382
ERROR      346     364    1ØØ6    1Ø17
ERRPNT     841     915
ESQREQ    137Ø    1463    175Ø    1753#
EXCALP     841
FBMERR       6#      6
FININ      867     868
FININØ     865#   122Ø    1229
FINISH     838     86Ø     911     939    1132    1214    1391    145Ø    146Ø
FINOT      933     934
FINOUT     862     918#
FIT        838     942     954    1Ø33    1138    1215
FITØ       942#    985
FIT1       869     943     954#
FITHGH    1225    1226
FITSIZ     961     962
FNFERR       6#      6
FORCE      838     94Ø    1Ø92    1113
FORCEØ    1Ø21    1Ø63#
FORCE1     941    1Ø94#
```

```
FORCE2    1Ø56      1259#
FORCEL     935      1Ø92#
FORHGH    1Ø69      1Ø7Ø
FORSIZ    1Ø51      1Ø52
FRAGBK    1655#     167Ø
FRAGIO    14Ø2#     1442
FRECR1     976       979
FRESWP    1171      1172
FRGBK1    1656#     166Ø
FRGBK2    1658      1661#
FRGIO1    14Ø5#     1411
FRGIO2    14Ø7      1413#
FRGSEG       6#        6      1152      1397      1437      1653
FT2REL       6#      844       866       932       957       96Ø       975      1Ø5Ø      1Ø68      1Ø87      117Ø      1225      1338      1347
          135Ø
FTATTA       6#
FTCCL       1Ø#
FTCHEC       6#      457       458       47Ø       549       556       61Ø       616      171Ø      1717
FTDISK      1Ø#       11        14       2ØØ       49Ø       538       574       6Ø4
FTEXAM       6#
FTFINI       6#
FTGETT       6#
FTHALT       6#
FTKCT        6#
FTLOGI      1Ø#
FTMONP       6#      457       458       47Ø       549       556       61Ø       616      171Ø      1717
FTPRV        6#
FTRA1Ø       6#
FTRC1Ø      1Ø#       11        14        22        26        31      1299      13Ø5      1468      16ØØ
FTRCHK       6#     1Ø14
FTREAS       6#
FTSLEE       6#
FTSWAP      1Ø#       4Ø       152       155       2ØØ       625       757      1356
FTTALK       6#
FTTIME       6#
FTTRAC       6#     11Ø7      111Ø      121Ø
FTTRPS       6#
FTTTYS       6#
FULCNT     841
FULCOR    1148      1189#     1195
FULL      113Ø      1147#
FULL1     1154#     1164
FULL1A    1155#     1169
FULL1B    1165#     1173
FULL2     1156      1168#
FULSET    1161      1183#
FXSAT     129Ø      16Ø3      1653#
FXSAT1    1654      1659      1673#
FXSATC    1684#
GETBIT    16Ø4      1626
GETFCR    1145      119Ø
GXSAT     1129      1155      16Ø3      1622#
HIGHJB    1ØØ6      1Ø1Ø
```

```
HOLEF      976       977
HSAMSK       6#        6
HSAPOS       6#        6
HSASIZ       6#        6
HUNGCT       6#        6
HUNGST       6#        6
I            6#        6
IB           6#        6
IBUFR        6#        6
ICLOSB       6#        6
ICPROT    1379#     1379
ICPRT1    1378#     1378
ILM          6#
ILUERR       6#        6
IMGIN      199       838       676       955      1125      1127      1216      1230      1248      1250      1257      1294      1323
IMGOUT     838       877       991       994      1122      1228      1248      1259      1287      1293
INBFR        6#        6
INERR      864       911#
INFLG      183       19Ø#      859      1139
INITR        6#        6
INPB         6#        6
IO           6#        6

IOACT        6#        6
IOBEG        6#        6
IOBKTL       6#        6
IOBOT        6#        6
IOCON        6#        6
IODEND       6#        6
IODERR       6#        6      1435
IODTER       6#        6      1435      1459
IOEND        6#        6
IOFST        6#        6
IOIMPM       6#        6      1435
IONRCK       6#        6
IOPAR        6#        6
IOS          6#        6      1435      1453      1459
IOTEND       6#        6
IOUSE        6#        6
IOW          6#        6
IOWC         6#        6
IOWQ       511       512#      673
IPOPJ     1191      1248      1268      1335      1351      16Ø5      1627      1686
IPOPJ1    16Ø5      1635
IPPERR       6#        6
ISCAN      758       759#      945
ITEM         6#        6        45       241       242       243       246       248       249       341       434       849      1153      1185
          1189      1258      1259      126Ø      1261      1262      1263      1283      1289      1339      134Ø      1341      1344      1346
          1622      1623      1628      1662      1663      1664      1665      1666      1668      1673      1674      17Ø3      17Ø4
J           45#       76        79        81        82        83        85        88        95       1Ø3       1Ø5       1Ø6       1Ø7       11Ø
           111       122       132       138       139       14Ø       141       166       173       176       184       341#      36Ø       374
           384       385       386       387       39Ø       434#      44Ø       445       446       449       45Ø       454       455       849#
           86Ø       862       896       897       912       918       922       94Ø       942       948       954      1Ø1Ø      1Ø12      1Ø13
          1Ø15      1Ø2Ø      1Ø33      1Ø35      1Ø36      1Ø44      1Ø58      1Ø62      1Ø88      1Ø92      1Ø98      11Ø1      1114      1118
```

```
            1131      1132      1214      1219      1235      1236
  JACCT        6#        6
  JBFADR       6#        6
  JBFCTR       6#        6
  JBFPTR       6#        6
  JBTADR     199       36Ø       837       912       922      1Ø44      1Ø98      1114      1118      1219      1235
  JBTCHK    1357      1392      1451
  JBTDAT     839       84Ø       897
  JBTQ        37       139       23Ø       237       242       247       374       376       377       38Ø       382       384       385       386
             387       445       449       45Ø       454
  JBTQM1     232       246
  JBTQMN     233       243       245
  JBTQP1     231       24Ø       249
  JBTSTS      36        79        88        95       1Ø5       11Ø       111       141       166       173       199       39Ø       836       948
            1ØØ6      1Ø12      1Ø2Ø      1Ø35      1Ø88      1248      1283      1344      1346
  JBTSWP      72        81        82       122       184       837       896      1Ø36      1131      1153      1185      1236      1248      1289
            1461
  JBUF         6#        6
  JDAT         6#        6       897       898       9Ø1       9Ø4       9Ø7       9Ø8       913       923      1Ø95
  JERR         6#        6
  JLOG         6#        6
  JNA          6#        6
  JOB         36        85      11Ø1      1341
  JOBDAC    1357      1388      1447
  JOBDPD     839       9Ø7
  JOBDPG     839       9Ø4       9Ø8
  JOBMAX     1Ø3       23Ø       241      134Ø
  JOBPC      839       9Ø1
  JOBQUE      61        99       171       619       62Ø#
  JRQ          6#        6       1Ø4
  JWPOS        6#        6
  JWSIZ        6#        6
  JXPN         6#        6       115       167      1Ø11      1343
  K4SWAP    17Ø3      1721
  KCORE1     837       914       926
  LBHIGH    16Ø5      1632      1676      1695
  LISTSN       6
  LOC        496#      498       512       512#      514       515       515#      516       517       517#      518       519       519#      52Ø
  LOOKB        6#        6
  MAXJBN    1ØØ6      1Ø58      1Ø62
  MAXQ        53       5ØØ#      5ØØ       5Ø2
  MAXSIZ    1ØØ6      1Ø29      1Ø55      1Ø57
  MAXSWP    1629      163Ø      172Ø      1728#
  MEDDLE       6#        6
  MINQ       147       5Ø1#      5Ø1
  MJOBCK    1357      1389      1448
  MJOBN       38        76
  MQAVAL     491       491#
  MQQ        491       491#      6Ø5       681       761       784       793
  MQREQ      575#      576
  MTAVAL     495       495#
  MTQ        495       495#      6Ø9       696       765       78Ø       789
  MTREQ      579#      58Ø
```

```
MXCODE      514#    514
MXK2SW     172Ø    1736#   174Ø
MXQUE       23Ø     235
NECERR        6#      6
NLFERR        6#      6
NOFIT      1Ø32    1138#
NQUEUE      498#    52Ø
NSDERR        6#      6
NSFERR        6#      6
NSHF          6#      6
NSRBIT        6#      6
NSWP          6#      6    1Ø4Ø
NULQ        239     512#    649
NXM           6#
NXTINI       5Ø      53#
NXTJB1       75      85#
NXTJOB       4Ø      74#
OBUFR         6#      6
OCLOSB        6#      6
OERROR      839    1223
OSCAN       758     775#   1Ø3Ø
OUTBFB        6#      6

OUTP2      1132#   1186
OUTPR         6#      6
PC          436#    439     446     451     455
PCORSZ      841     892
PCSTOP      841     917
PDP           6#      6      62     1ØØ     127     148     156     175     177     353     393     854     858     868
            89Ø     914     926     934     962     979     983     986    1ØØ2    1Ø52    1Ø63    1Ø7Ø    11Ø3    1124
           1124    1129    1135    114Ø    1148    1154    1155    1158    1162    1168    1172    1178    1179    1183
           1189    119Ø    1195    1198    1222    1226    1258    1286    129Ø    1291    1296    1339    139Ø    14Ø2
           1425    1449    1622    1626    1659    1662    1667    1668    1671    1673    1684
PJBSTS       37      38     118     199     392
POTLST       72     161     165     174
POV           6#
PQ1         514#    664     669     7Ø7     71Ø     713     716     719     722     725     732     738     768     795
PQ2         189     516#    733     738     739     74Ø     769     794
PQ3         518#    734     739     74Ø     77Ø     786
PROC          6#      6     342     898     9Ø5     9Ø8     912     913     922     923    1Ø95    1Ø98    1219
PROT        894    1297
PROTØ       895    1297
PRTERR        6#      6
PVSPYA        6#      6
PVSPYM        6#      6
PVTRPS        6#      6
Q           342#    348     352     356     363     366     368     369     371     38Ø     382     384     387     389
            39Ø
QAUS        6Ø4     723#
QAUW        538     677#
QBAK        454#    786     794     795
QBAK1       449#    778     779     78Ø     781     782     783     784
QBITS       125     532     534#
QCMW        112     63Ø     653#
```

```
QDAS        6Ø6       7Ø8#
QDAW        54Ø       682#
QDCS        6Ø8       711#
QDCW        542       685#
QDTS        6Ø7       717#
QDTW        541       691#
QFIX        374#      395       396       632       633
QFOR        445#      76Ø       768       769       77Ø       776       777       785
QFOR1       443#      761       762       763       764       765       766       767       787       788       789       79Ø       791       792
            793
QFOR2       443       447#
QINI        229       235#
QINI1       246#      25Ø
QIOWW       544       671#
QJ           43#       87       1Ø1       128       131       134       135       136       137       138       145       146       147       15Ø
            166       167       172       173
QJOB         59       1Ø1       131       622       623#
QJSIZ       36Ø#      399       4ØØ       636       637
QLINK       352#      356       397       398       634       635
QMQS        6Ø5       7Ø5#
QMQW        539       679#
QMTS        6Ø9       72Ø#
QMTW        543       694#
QNULW       547       647#
QQSD        7Ø3#      7Ø7       71Ø       713       716       719       722       725       745#
QQSTAB      658       728       753#
QQTTY       189       664       669       746#
QR          437#      438       44Ø       441       443
QRNS        594#      67Ø
QRNW        534       656#
QSCAN       163       2ØØ       438#      447       452       456       946       1Ø31
QSLPW       546       697#
QSTAB       658       732#
QSTOP       126       63Ø       65Ø#
QSTOPW      548       65Ø#
QSTS        6Ø3       714#
QSTW        537       688#
QTIME        98       63Ø       726#
QTIOWW      545       674#
QTSS        596#      672
QTSW        536       667#
QTTAB       728       738#
QWSS        595#      671
QWSW        535       662#
QX1         363#      366
QX2         357       368#
QX3         389       393#
QXFER       1ØØ       127       148       2ØØ       348#
RCXSKD       27        28#
RENMB         6#        6
REQTAB       55       567       569#
RNAVAL      486       486#
RNQ         391       486       486#      6ØØ
```

```
RNQUNT    75Ø      752#
RNREQ     572#     571
RUN         6#       6
RUNABL      6#       6       9Ø
RUNMSK      6#       6       89
SCHD1     164      176#
SCHED     128      161#
SCHEND   1762#
SCNJOB   1ØØ8     1Ø27#
SCNOK    1Ø15     1Ø19#
SCNOUT    968     1ØØ7#
SD          6#       6
SERA      863      919      989      1368     1395     1396     14Ø9     141Ø     144Ø     1453     1462     175Ø     1755#
SERACT   1383     1465     175Ø     1758#
SHF         6#       6      167      1282
SHFWAT    84Ø      852      856       984
SHRSEG      6#       6
SLEVEL      6#       6
SLICE       6#       6
SLPQ      512      512#     699       777
SNA         6#       6
SPYSEG      6#       6

SQGO     1355     1382#
SQGO1    1355     1386     1393#     1466
SQGO2    1398     1426#
SQIN     1237     1355     1367#
SQLEN    1393     14Ø3     1414      1759#
SQOUT    1136     1355     1368#
SQREQ     857     1369     1387      1394     1418     1419     1423     1438     1446     1454     1464     175Ø     1751#
SSCAN     162      63Ø      768#
ST        435#     438      447       452      456
STAVAL    489      489#
STOPIO      6#       6
STOPQ     512#     652      776
STQ       489      489#     5Ø1       6Ø3      69Ø      766      779      788
STREQ     573#     574
STTYR1      6#       6
STTYBF      6#       6
SUMCOR   1ØØ6     1028     1Ø59      1Ø6Ø
SW         44#      77       82        88       89       9Ø       95       96      1Ø4      1Ø5      11Ø      111      113      115
          117      118      119       121      125
SWAP      156      843      852#
SWAPI     974     12Ø8#
SWAPO     997     117Ø     1139#
SWP         6#       6       78       115      142      167      949     1Ø4Ø     1287     1282
SWP1      9Ø3      909      939#     1115
SWP2      861      94Ø#
SWPCLR      6#       6
SWPENT   1624     1625     1682      1683     1698     17Ø1     17Ø2     1742#
SWPER1   1452     1457#
SWPERC    987      99Ø      996
SWPERR   1436     146Ø#
SWPHGH   1693     1696
```

```
SWPINI       1603      1695#
SWPINT       1433      1435#
SWPRFC        920       989#
SWPSIZ       1740#     1742      1744      1746
SWPTAB       1720      1742      1744      1746#
SYSDEV          6#        6
T             846#      876       878       892       894       895       896       901       902       904       905       906       907       948
              949      1011      1012      1020      1027      1028      1035      1040      1044      1045      1048      1049      1055      1057
             1059      1087      1088      1114      1115      1119      1120      1121      1122      1127      1128      1160      1161      1163
             1165      1228      1229      1231      1232      1233      1234
T1            343#      352       354       355       363       365       374       375       376       377       847#      877       878       887
              888      1118      1123      1124
T2            344#      355       360       361       362       365       368       371       375       376       377       382       385       386
              848#     1119      1123
TAC             6#        6        53        54        55        58        99       163       235       236       237       238       239       240
              242       244       245       344       436       848       915       946       989       990       991       992       993       994
              995       996      1031      1131      1157      1178      1179      1185      1199      1201      1216      1217      1218      1236
             1251      1252      1253      1257      1262      1263      1265      1279      1289      1292      1293      1294      1321      1322
             1323      1367      1368      1371      1372      1373      1384      1385      1387      1388      1389      1391      1392      1395
             1396      1397      1405      1406      1410      1426      1427      1428      1437      1441      1442      1444      1446      1447
             1448      1450      1451      1457      1458      1460      1461      1462      1463      1464      1633      1634      1635      1653
             1655      1656      1661      1664      1669      1675      1676      1677      1678      1680      1681      1695      1696      1697
             1699      1707
TAC1            6#        6        44       169       171       343       437       847       952      1034      1043      1045      1061      1064
             1135      1150      1152      1153      1197      1200      1201      1203      1230      1231      1232      1234      1235      1250
             1260      1261      1265      1266      1280      1282      1283      1343      1344      1346      1369      1370      1382      1383
             1392      1394      1404      1416      1419      1429      1430      1438      1451      1628      1629      1630      1631      1632
             1633      1698      1699      1700
TEM             6#        6
TIMEF          38        74        92
TIOWQ         121       512       512#      676       785
TRNERR          6#        6
TRYSWP        837      1063
TSAVAL        488       488#
TSQ           488       488#      602
TSREQ         572#      573
TT            340#      348       349       370       379       391       392
TTYATC          6#        6
TTYBIU          6#        6
TTYUSE          6#        6
UCHN            6#        6
UIOMOD          6#        6
USRMOD          6#        6       902
UUO             6#        6
UWP             6#        6
UWPOFF          6#        6
VIRTAL        841       888       987       993      1248      1266      1280      1693      1704
VSCHED         19#       19
WSAVAL        487       487#
WSQ           119       487       487#      601
WSREQ         571#      572
WTMASK          6#        6       172
XCKCSW         29        52#
```

```
XJOB        6Ø      626#     627     1ØØ1     1ØØ7     1Ø16     1Ø19     1345
XPAND      844     1321#
XPANDH    1334     1337#
ZERSWP     89Ø     1277     1279#
```

```
CODES        6#     511     544
DISABL       6#
ENABLE       6#
NOSCHE       6#
NOSHUF       6#
PTTAB      643#     656     726
QUEUES       6#     485     534     569     6ØØ
SCHEDU       6#
SHUFFL       6#
STARTD       6#
TTAB       639#     647     65Ø     653     662     667     671     674     677     679     682     685     688     691
           694      697     7Ø5     7Ø8     711     714     717     72Ø     723
X          478#     486     487     488     489     49Ø     491     492     493     494     495     5Ø7#    511     512
           512#     529#    534     535     536     537     538     539     54Ø     541     542     543     544     545
           546      547     548     563#    57Ø     571     572     573     574     575     576     577     578     579
           591#     6ØØ     6Ø1     6Ø2     6Ø3     6Ø4     6Ø5     6Ø6     6Ø7     6Ø8     6Ø9
XP           6#       6      18     499     5ØØ     5Ø1     513#   13Ø7    13Ø8
```