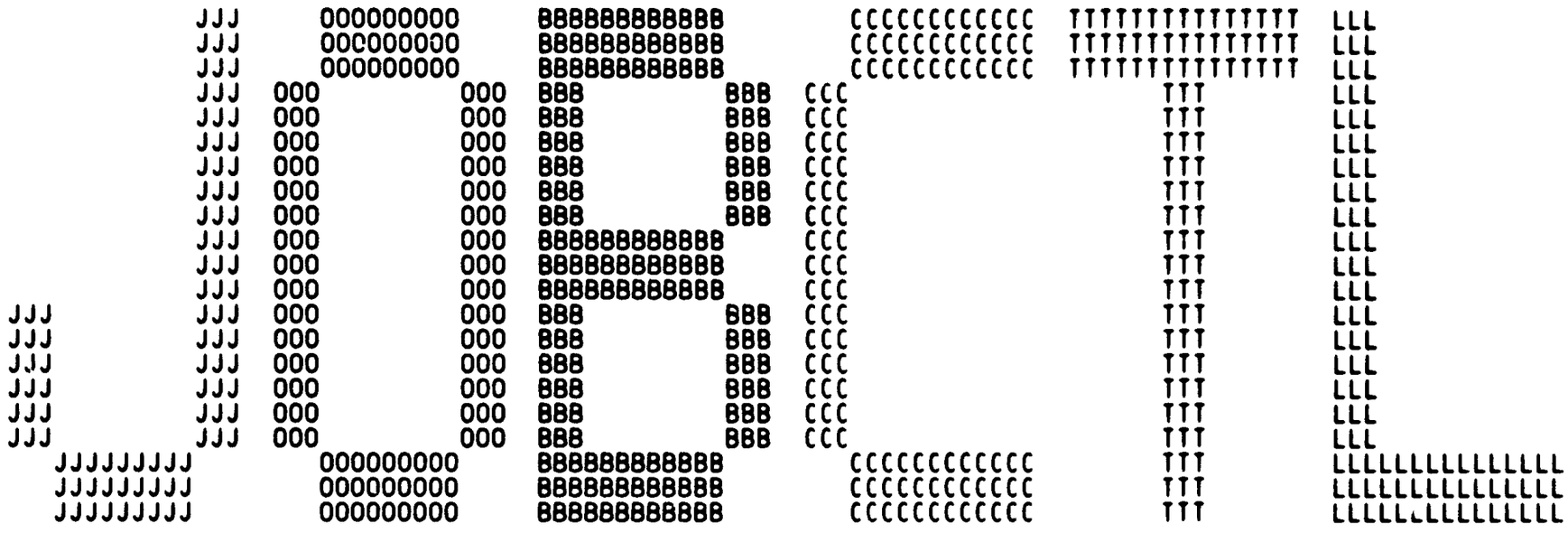
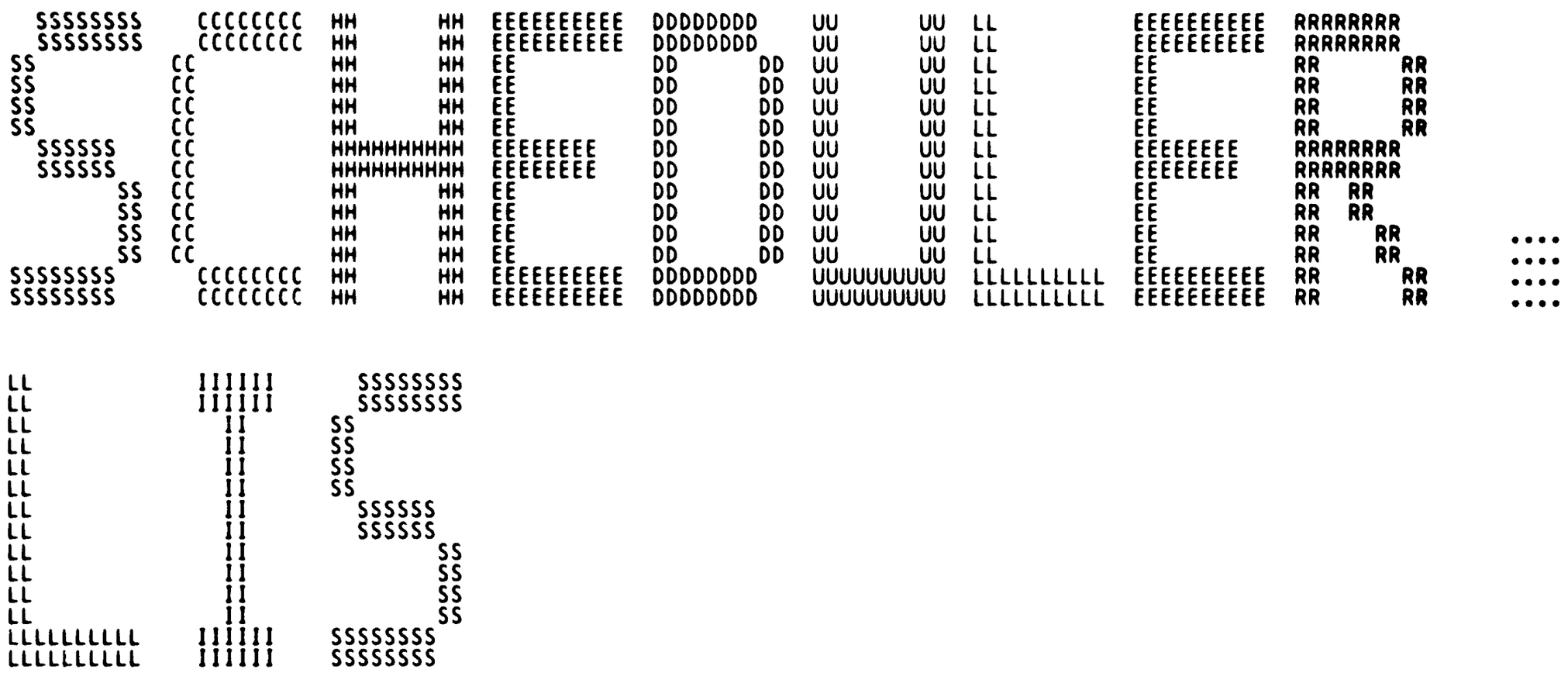


_$2

Val

0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
314
485
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF


SSSSSSSS CCCCCCCC HH HH EEEEEEEEEE DDDDDDDD UU UU LL EEEEEEEEEE RRRRRRRR
SSSSSSSS CCCCCCCC HH HH EEEEEEEEEE DDDDDDDD UU UU LL EEEEEEEEEE RRRRRRRR
SS CC HH HH EE DD DD UU UU LL EE RR RR
SS CC HH HH EE DD DD UU UU LL EE RR RR
SS CC HH HH EE DD DD UU UU LL EE RR RR
SS CC HH HH EE DD DD UU UU LL EE RR RR
SSSSSS CC HHHHHHHHHH EEEEEEEE DD DD UU UU LL EEEEEEEE RRRRRRRR
SSSSSS CC HHHHHHHHHH EEEEEEEE DD DD UU UU LL EEEEEEEE RRRRRRRR
SS CC HH HH EE DD DD UU UU LL EE RR RR
SS CC HH HH EE DD DD UU UU LL EE RR RR
SS CC HH HH EE DD DD UU UU LL EE RR RR
SS CC HH HH EE DD DD UU UU LL EE RR RR
SSSSSSSS CCCCCCCC HH HH EEEEEEEEEE DDDDDDDD UUUUUUUUUU LLLLLLLLLL EEEEEEEEEE RR RR
SSSSSSSS CCCCCCCC HH HH EEEEEEEEEE DDDDDDDD UUUUUUUUUU LLLLLLLLLL EEEEEEEEEE RR RR
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```
;     1          0001  0 MODULE SCHEDULER(%TITLE 'Job scheduler'
;     2          0002  0                    IDENT = 'V04-001'
;     3          0003  0                    ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     6          0006  1 !
;     7          0007  1 !*****************************************************************************
;     8          0008  1 !*                                                                           *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                      *
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    18          0018  1 !*  TRANSFERRED.                                                              *
;    19          0019  1 !*                                                                           *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    22          0022  1 !*  CORPORATION.                                                              *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*****************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:
;    33          0033  1 !       Job controller.
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !       This module contains the job scheduling routines.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:
;    39          0039  1 !       VAX/VMS user and kernel mode.
;    40          0040  1 !--
;    41          0041  1 !
;    42          0042  1 ! AUTHOR: M. Jack, CREATION DATE: 16-Feb-1982
;    43          0043  1 !
;    44          0044  1 ! MODIFIED BY:
;    45          0045  1 !
;    46          0046  1 !       V04-001 JAK0233         J A Krycka      10-Sep-1984
;    47          0047  1 !               In FIND_PENDING_JOBS fix bug that prevented the job header
;    48          0048  1 !               record from being rewritten to disk when scheduling pending
;    49          0049  1 !               jobs from a generic batch queue to various execution queues.
;    50          0050  1 !
;    51          0051  1 !       V03-011 JAK0231         J A Krycka      28-Aug-1984
;    52          0052  1 !               In FIND_PENDING_JOBS protect against failure to place job in
;    53          0053  1 !               execution by START_EXECUTOR_JOB (e.g., $CREPRC service failure).
;    54          0054  1 !               A failure of this nature will result in release of the current
;    55          0055  1 !               job record to the free list, so FIND_PENDING_JOBS should not
;    56          0056  1 !               try to rewrite it.
;    57          0057  1 !
```

```
;   58          0058  1 !       V03-010 KPL0004         P Lieberwirth,   2-Aug-1984
;   59          0059  1 !               Routine FIND_PENDING_JOBS is susceptible to a corrupted
;   60          0060  1 !               PENDING_BATCH_LIST or PENDING_PRINT_LIST.  Protect against
;   61          0061  1 !               that type of corruption.
;   62          0062  1 !
;   63          0063  1 !       V03-009 KPL0003         P Lieberwirth,  31-Jul-1984
;   64          0064  1 !               Add omitted "." in v03-008.
;   65          0065  1 !
;   66          0066  1 !       V03-008 KPL0002         P Lieberwirth,  23-Jul-1984
;   67          0067  1 !               Extend protection described in V03-007 to routine AFTER_NONAST.
;   68          0068  1 !
;   69          0069  1 !       V03-007 KPL0001         P Lieberwirth,   9-Jul-1984
;   70          0070  1 !               Protect routine ENQUEUE_JOB against corrupt SJH.  A corrupt
;   71          0071  1 !               SJH is generally a record that has been deallocatd to the
;   72          0072  1 !               free list but still appears on the SMQ's current_list.  Changes
;   73          0073  1 !               being made at this time in other modules should reduce the
;   74          0074  1 !               frequency of this type of corruption.
;   75          0075  1 !
;   76          0076  1 !       V03-006 MLJ0118         Martin L. Jack, 23-Aug-1983  12:41
;   77          0077  1 !               Change field names.
;   78          0078  1 !
;   79          0079  1 !       V03-005 MLJ0115         Martin L. Jack, 30-Jul-1983  15:03
;   80          0080  1 !               Changes for job controller baselevel.
;   81          0081  1 !
;   82          0082  1 !       V03-004 MLJ0114         Martin L. Jack, 23-Jun-1983  5:03
;   83          0083  1 !               Changes for job controller baselevel.
;   84          0084  1 !
;   85          0085  1 !       V03-003 MLJ0113         Martin L. Jack, 26-May-1983  21:08
;   86          0086  1 !               Changes for job controller baselevel.
;   87          0087  1 !
;   88          0088  1 !       V03-002 MLJ0112         Martin L. Jack, 29-Apr-1983  3:09
;   89          0089  1 !               Changes for joh controller baselevel.
;   90          0090  1 !
;   91          0091  1 !       V03-001 MLJ0109         Martin L. Jack, 14-Apr-1983  12:49
;   92          0092  1 !               Changes for job controller baselevel.
;   93          0093  1 !
;   94          0094  1 !**
```

```
;   96          0095  1 REQUIRE 'SRC$:JOBCTLDEF';
;   97          1136  1
;   98          1137  1
;   99          1138  1 FORWARD ROUTINE
;  100          1139  1         AFTER_NONAST:                   NOVALUE,
;  101          1140  1         AFTER_AST:                      NOVALUE,
;  102          1141  1         REQUEUE_STARTING_JOBS:          NOVALUE,
;  103          1142  1         EXECUTOR_SCHEDULING_POLICY,
;  104          1143  1         EXECUTOR_ACCEPTS_JOB,
;  105          1144  1         FIND_AVAILABLE_EXECUTOR:        L_OUTPUT_1,
;  106          1145  1         FIND_PENDING_JOBS:              NOVALUE,
;  107          1146  1         JOB_SCHEDULING_POLICY,
;  108          1147  1         ENQUEUE_JOB:                    L_OUTPUT_2 NOVALUE;
;  109          1148  1
;  110          1149  1
;  111          1150  1 EXTERNAL ROUTINE
;  112          1151  1         LOCK_QUEUE_FILE:                NOVALUE,
;  113          1152  1         READ_RECORD,
;  114          1153  1         RELEASE_RECORD:                 NOVALUE,
;  115          1154  1         REWRITE_RECORD:                 NOVALUE,
;  116          1155  1         SCAN_INCOMPLETE_SERVICES:       NOVALUE,
;  117          1156  1         SCHEDULE_NONAST:                NOVALUE,
;  118          1157  1         START_EXECUTOR_JOB:             NOVALUE,
;  119          1158  1         UNLOCK_QUEUE_FILE:              NOVALUE,
;  120          1159  1         UPDATE_GETQUI_DATA:             NOVALUE;
;  121          1160  1
;  122          1161  1
;  123          1162  1 BUILTIN
;  124          1163  1         TESTBITSC;
```

```
;  126          1164  1 ROUTINE AFTER_NONAST: NOVALUE=
;  127          1165  1
;  128          1166  1 !++
;  129          1167  1 !
;  130          1168  1 ! FUNCTIONAL DESCRIPTION:
;  131          1169  1 !       This routine is scheduled to execute by the completion AST routine for
;  132          1170  1 !       the timer on the timed job queue.  It requeues all jobs whose expiration
;  133          1171  1 !       time is earlier than or equal to the current time and sets a new timer.
;  134          1172  1 !
;  135          1173  1 ! INPUT PARAMETERS:
;  136          1174  1 !       NONE
;  137          1175  1 !
;  138          1176  1 ! IMPLICIT INPUTS:
;  139          1177  1 !       NONE
;  140          1178  1 !
;  141          1179  1 ! OUTPUT PARAMETERS:
;  142          1180  1 !       NONE
;  143          1181  1 !
;  144          1182  1 ! IMPLICIT OUTPUTS:
;  145          1183  1 !       NONE
;  146          1184  1 !
;  147          1185  1 ! ROUTINE VALUE:
;  148          1186  1 !       NONE
;  149          1187  1 !
;  150          1188  1 ! SIDE EFFECTS:
;  151          1189  1 !       NONE
;  152          1190  1 !
;  153          1191  1 !--
;  154          1192  1
;  155          1193  2 BEGIN
;  156          1194  2 IF .QUEUE_FAB[FAB$W_IFI] NEQ 0
;  157          1195  2 THEN
;  158          1196  3     BEGIN
;  159          1197  3     LOCAL
;  160          1198  3         SQH:            REF BBLOCK,     ! Pointer to SQH
;  161          1199  3         SMQ_N,                          ! Record number of SMQ
;  162          1200  3         SMQ:            REF BBLOCK,     ! Pointer to SMQ
;  163          1201  3         SJH_N,                          ! Record number of SJH
;  164          1202  3         SJH:            REF BBLOCK,     ! Pointer to SJH
;  165          1203  3         STATUS;                         ! Status return
;  166          1204  3
;  167          1205  3
;  168          1206  3     ! Get the current time.
;  169          1207  3     !
;  170          1208  3     $GETTIM(TIMADR=CUR_TIME);
;  171          1209  3
;  172          1210  3
;  173          1211  3     ! Lock the queue file.
;  174          1212  3     !
;  175          1213  3     LOCK_QUEUE_FILE();
;  176          1214  3
;  177          1215  3
;  178          1216  3     ! Requeue all jobs in the timer queue whose expiration times are earlier
;  179          1217  3     ! than or equal to the current time.
;  180          1218  3     !
;  181          1219  3     SQH = READ_RECORD(SQH$K_RECNO);
;  182          1220  3     WHILE .SQH[SQH$L_TIMER_LIST] NEQ 0 DO
```

```
;  183          1221  4          BEGIN
;  184          1222  4          SJH = READ_RECORD(SJH_N = .SQH[SQH$L_TIMER_LIST]);
;  185          1223  5          IF TIME_GEQU(CUR_TIME, SJH[SJH$Q_AFTER_TIME])
;  186          1224  4          THEN
;  187          1225  5              BEGIN
;  188          1226  5              UPDATE_GETQUI_DATA(.SJH_N, .SJH);
;  189          1227  5              SQH[SQH$L_TIMER_LIST] = .SJH[SYM$L_LINK];
;  190          1228  5              IF .SJH[SYM$L_LINK] EQL 0 THEN SQH[SQH$L_TIMER_LIST_END] = 0;
;  191          1229  5              SMQ_N = .SJH[SJH$L_QUEUE_LINK];
;  192          1230  5              IF .SMQ_N NEQ 0
;  193          1231  5              THEN
;  194          1232  5                  ! Queue pointer is OK.
;  195          1233  5                  !
;  196          1234  6                  BEGIN
;  197          1235  6                  SMQ = READ_RECORD(.SMQ_N);
;  198          1236  6                  SMQ[SMQ$W_TIMER_JOB_COUNT] = .SMQ[SMQ$W_TIMER_JOB_COUNT] - 1;
;  199          1237  6                  ENQUEUE_JOB(.SJH_N, .SJH);
;  200          1238  6                  REWRITE_RECORD(.SMQ_N);
;  201          1239  6                  REWRITE_RECORD(.SJH_N);
;  202          1240  6                  END
;  203          1241  5              ELSE
;  204          1242  5                  ! Queue pointer bad, just release SJH.  Next cold or warm start
;  205          1243  5                  ! will fix it.
;  206          1244  5                  !
;  207          1245  5                  RELEASE_RECORD(.SJH_N);
;  208          1246  5              END
;  209          1247  4          ELSE
;  210          1248  5              BEGIN
;  211          1249  5
;  212          1250  5              ! Set a timer on the first remaining job.
;  213          1251  5              !
;  214        P 1252  5              STATUS = $SETIMR(
;  215        P 1253  5                  DAYTIM=SJH[SJH$Q_AFTER_TIME],
;  216        P 1254  5                  ASTADR=AFTER_AST,
;  217          1255  5                  REQIDT=JBC$K_AFTER_IDT);
;  218          1256  5              IF NOT .STATUS
;  219          1257  5              THEN
;  220          1258  5                  SIGNAL(JBC$_SETIMR OR STS$K_ERROR, 0, .STATUS);
;  221          1259  5
;  222          1260  5
;  223          1261  5              EXITLOOP;
;  224          1262  4              END;
;  225          1263  3          END;
;  226          1264  3      REWRITE_RECORD(SQH$K_RECNO);
;  227          1265  3
;  228          1266  3
;  229          1267  3      ! Unlock the queue file.
;  230          1268  3      !
;  231          1269  3      UNLOCK_QUEUE_FILE();
;  232          1270  2      END;
;  233          1271  1  END;


                                                        .TITLE  SCHEDULER Job scheduler
                                                        .IDENT  \V04-001\

                                                        .PSECT  COMMON,NOEXE,  OVR,2
```

```
00000 DIAG_STORAGE_BASE:
                .BLKB   0
00000 DIAG_TRACE:
                .BLKB   96
00060 DIAG_COUNT:
                .BLKB   96
000C0 DIAG_FLAGS:
                .BLKB   4
000C4 WORK_AREA:
                .BLKB   44
000F0 SNDJBC_COUNT:
                .BLKB   132
00174 GETQUI_COUNT:
                .BLKB   40
0019C SNDACC_COUNT:
                .BLKB   28
001B8 SNDSMB_COUNT:
                .BLKB   72
00200 DIAG_STORAGE_END:
                .BLKB   0
00200 FLAGS:    .BLKB   4
00204 IMAGE_DUMP_STSFLG:
                .BLKB   4
00208 THIS_SYSID:
                .BLKB   6
0020E           .BLKB   2
00210 CUR_TIME:
                .BLKB   8
00218 HOURLY_TIME:
                .BLKB   8
00220 HOURLY_PARAMS:
                .BLKB   20
00234 SYMBIONT_COUNT:
                .BLKB   4
00238 QUEUE_REFERENCE_COUNT:
                .BLKB   4
0023C MBX_MESSAGE_COUNT:
                .BLKB   4
00240 MBX:      .BLKB   4
00244 MBX_END:  .BLKB   4
00248 MEMORY_FREE_QUEUES:
                .BLKB   40
00270 NONAST_WORK_QUEUE:
                .BLKB   8
00278 BCB_FREE_LIST:
                .BLKB   4
0027C BCB_ACTIVE_LIST:
                .BLKB   4
00280 GQL_FREE_LIST:
                .BLKB   4
00284 GQL_ACTIVE_LIST:
                .BLKB   4
00288 OPEN_GETQUI_LIST:
                .BLKB   4
0028C PROCESS_DATA_LIST:
                .BLKB   4
```

```
00290 SYMBIONT_CONTROL:
                .BLKB   4
00294 SPARE_AREA:
                .BLKB   12
002A0 REMOTE_REQUEST_LKSB:
                .BLKB   8
002A8 QUEUE_FILE_LKSB:
                .BLKB   8
002B0 QUEUE_LOCK_LKSB:
                .BLKB   8
002B8 RSP:      .BLKB   8
002C0 JBC_PRIORITY:
                .BLKB   4
002C4 JBC_PRIVILEGES:
                .BLKB   8
002CC JBC_QUOTAS:
                .BLKB   66
0030E           .BLKB   2
00310 JBC_UIC:  .BLKB   4
00314 QUEUE_FAB:
                .BLKB   80
00364 QUEUE_RAB:
                .BLKB   68
003A8 QUEUE_NAM:
                .BLKB   96
00408 QUEUE_XAB:
                .BLKB   88
00460 QUEUE_RSA:
                .BLKB   255
0055F           .BLKB   1
00560 QUEUE_ALQ:
                .BLKB   4
00564 QUEUE_MBF:
                .BLKB   1
00565           .BLKB   3
00568 ACCOUNTING_FABS:
                .BLKB   8
00570 ACCOUNTING_RABS:
                .BLKB   8
00578 ACCOUNT_FAB_A:
                .BLKB   80
005C8 ACCOUNT_RAB_A:
                .BLKB   68
0060C ACCOUNT_NAM_A:
                .BLKB   96
0066C ACCOUNT_RSA_A:
                .BLKB   255
0076B           .BLKB   1
0C76C ACCOUNT_FAB_B:
                .BLKB   80
007BC ACCOUNT_RAB_B:
                .BLKB   68
00800 ACCOUNT_NAM_B:
                .BLKB   96
00860 ACCOUNT_RSA_B:
                .BLKB   255
0095F           .BLKB   1
```

```
00960 DIAG_FAB:
                .BLKB   80
009B0 DIAG_RAB:
                .BLKB   68
009F4 MBX_CHAN:
                .BLKB   4
009F8 MBX_IOSB:
                .BLKB   8
00A00 MBX_BUFFER:
                .BLKB   1024
00E00 VALUE_STORAGE_BASE:
                .BLKB   0
00E00 ITEM_PRESENT:
                .BLKB   32
00E20 VALUE_GETQUI_BASE:
                .BLKB   0
00E20 VALUE_ACCOUNTING_MESSAGE:
                .BLKB   6
00E26 VALUE_ACCOUNTING_TYPES:
                .BLKB   4
00E2A VALUE_AFTER_TIME:
                .BLKB   8
00E32 VALUE_ALIGNMENT_PAGES:
                .BLKB   1
00E33 VALUE_BASE_PRIORITY:
                .BLKB   1
00E34 VALUE_BATCH_INPUT:
                .BLKB   6
00E3A VALUE_BATCH_OUTPUT:
                .BLKB   10
00E44 VALUE_BUFFER_COUNT:
                .BLKB   1
00E45 VALUE_CHARACTERISTIC_NAME:
                .BLKB   6
00E4B VALUE_CHARACTERISTIC_NUMBER:
                .BLKB   1
00E4C VALUE_CHARACTERISTICS:
                .BLKB   16
00E5C VALUE_CHECKPOINT_DATA:
                .BLKB   6
00E62 VALUE_CLI:
                .BLKB   6
00E68 VALUE_CPU_DEFAULT:
                .BLKB   4
00E6C VALUE_CPU_LIMIT:
                .BLKB   4
00E70 VALUE_DESTINATION_QUEUE:
                .BLKB   8
00E78 VALUE_DEVICE_NAME:
                .BLKB   6
00E7E VALUE_ENTRY_NUMBER:
                .BLKB   4
00E82 VALUE_ENTRY_NUMBER_OUTPUT:
                .BLKB   10
00E8C VALUE_EXTEND_QUANTITY:
                .BLKB   2
00E8E VALUE_FILE_COPIES:
```

```
                .BLKB   1
00E8F VALUE_FILE_IDENTIFICATION:
                .BLKB   36
00EB3 VALUE_FILE_SETUP_MODULES:
                .BLKB   6
00EB9 VALUE_FILE_SPECIFICATION:
                .BLKB   6
00EBF VALUE_FIRST_PAGE:
                .BLKB   4
00EC3 VALUE_FORM_DESCRIPTION:
                .BLKB   6
00EC9 VALUE_FORM_LENGTH:
                .BLKB   1
00ECA VALUE_FORM_MARGIN_BOTTOM:
                .BLKB   1
00ECB VALUE_FORM_MARGIN_LEFT:
                .BLKB   2
00ECD VALUE_FORM_MARGIN_RIGHT:
                .BLKB   2
00ECF VALUE_FORM_MARGIN_TOP:
                .BLKB   1
00ED0 VALUE_FORM_NAME:
                .BLKB   6
00ED6 VALUE_FORM_NUMBER:
                .BLKB   4
00EDA VALUE_FORM:
                .BLKB   8
00EE2 VALUE_FORM_SETUP_MODULES:
                .BLKB   6
00EE8 VALUE_FORM_STOCK:
                .BLKB   6
00EEE VALUE_FORM_WIDTH:
                .BLKB   2
00EF0 VALUE_GENERIC_TARGET:
                .BLKB   996
012D4 VALUE_JOB_COPIES:
                .BLKB   1
012D5 VALUE_JOB_LIMIT:
                .BLKB   1
012D6 VALUE_JOB_NAME:
                .BLKB   6
012DC VALUE_JOB_RESET_MODULES:
                .BLKB   6
012E2 VALUE_JOB_SIZE_MAXIMUM:
                .BLKB   4
012E6 VALUE_JOB_SIZE_MINIMUM:
                .BLKB   4
012EA VALUE_JOB_STATUS_OUTPUT:
                .BLKB   10
012F4 VALUE_LAST_PAGE:
                .BLKB   4
012F8 VALUE_LIBRARY_SPECIFICATION:
                .BLKB   6
012FE VALUE_LOG_QUEUE:
                .BLKB   8
01306 VALUE_LOG_SPECIFICATION:
                .BLKB   6
```

```
0130C VALUE_NOTE:
                .BLKB   6
01312 VALUE_OPERATOR_REQUEST:
                .BLKB   6
01318 VALUE_OWNER_UIC:
                .BLKB   4
0131C VALUE_PAGE_SETUP_MODULES:
                .BLKB   6
01322 VALUE_PARAMETER_1:
                .BLKB   6
01328 VALUE_PARAMETER_2:
                .BLKB   6
0132E VALUE_PARAMETER_3:
                .BLKB   6
01334 VALUE_PARAMETER_4:
                .BLKB   6
0133A VALUE_PARAMETER_5:
                .BLKB   6
01340 VALUE_PARAMETER_6:
                .BLKB   6
01346 VALUE_PARAMETER_7:
                .BLKB   6
0134C VALUE_PARAMETER_8:
                .BLKB   6
01352 VALUE_PRIORITY:
                .BLKB   1
01353 VALUE_PROCESSOR:
                .BLKB   6
01359 VALUE_PROTECTION:
                .BLKB   4
0135D VALUE_QUEUE:
                .BLKB   6
01363 VALUE_QUEUE_FILE_SPECIFICATION:
                .BLKB   6
01369 VALUE_RELATIVE_PAGE:
                .BLKB   4
0136D VALUE_RESERVED_INPUT_1:
                .BLKB   1
0136E VALUE_RESERVED_INPUT_2:
                .BLKB   2
01370 VALUE_RESERVED_INPUT_3:
                .BLKB   4
01374 VALUE_RESERVED_INPUT_4:
                .BLKB   6
0137A VALUE_RESERVED_OUTPUT_1:
                .BLKB   10
01384 VALUE_RESERVED_OUTPUT_2:
                .BLKB   10
0138E VALUE_SEARCH_STRING:
                .BLKB   6
0139A VALUE_SCSNODE_NAME:
                .BLKB   6
0139A VALUE_WSDEFAULT:
                .BLKB   2
0139C VALUE_WSEXTENT:
                .BLKB   2
0139E VALUE_WSQUOTA:
```

```
                                        .BLKB   2
                                013A0 VALUE_STORAGE_END:
                                        .BLKB   0

                                JBC$_CLOSEOUT=      266328
                                JBC$_NOCMKRNL=      272388
                                JBC$_NOOPER=        272532
                                JBC$_NOSYSNAM=      272404
                                JBC$_OPENIN=        266392
                                JBC$_OPENOUT=       266400
                                JBC$_READERR=       266416
                                JBC$_WRITEERR=      266448
                                        .EXTRN  LOCK_QUEUE_FILE
                                        .EXTRN  READ_RECORD, RELEASE_RECORD
                                        .EXTRN  REWRITE_RECORD, SCAN_INCOMPLETE_SERVICES
                                        .EXTRN  SCHEDULE_NONAST
                                        .EXTRN  START_EXECUTOR_JOB
                                        .EXTRN  UNLOCK_QUEUE_FILE
                                        .EXTRN  UPDATE_GETQUI_DATA
                                        .EXTRN  SYS$GETTIM, SYS$SETIMR

                                        .PSECT  CODE,NOWRT,2

                                OFFC 00000 AFTER_NONAST:
                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 1164
                    59 00000000G   EF   9E 00002        MOVAB   READ_RECORD, R9         ;
                    58 00000000'   EF   9E 00009        MOVAB   CUR_TIME, R8            ;
                          0106     C8   B5 00010        TSTW    QUEUE_FAB+2             ; 1194
                                   01   12 00014        BNEQ    1$                      ;
                                        04 00016        RET                             ;
                                   58   DD 00017 1$:     PUSHL   R8                      ; 1208
                00000000G   00         01   FB 00019        CALLS   #1, SYS$GETTIM          ;
                00000000G   EF         00   FB 00020        CALLS   #0, LOCK_QUEUE_FILE     ; 1213
                                       01   DD 00027        PUSHL   #1                      ; 1219
                            69         01   FB 00029        CALLS   #1, READ_RECORD         ;
                            54         50   D0 0002C        MOVL    R0, SQH                 ;
                                 68    A4   D5 0002F 2$:     TSTL    104(SQH)                ; 1220
                                       03   12 00032        BNEQ    3$                      ;
                                  0096 31   00034        BRW     8$                      ;
                            55         68   A4   D0 00037 3$:     MOVL    104(SQH), SJH_N         ; 1222
                                       55   DD 0003B        PUSHL   SJH_N                   ;
                            69         01   FB 0003D        CALLS   #1, READ_RECORD         ;
                            52         50   D0 00040        MOVL    R0, SJH                 ;
                          009C C2      04   A8   D1 00043        CMPL    CUR_TIME+4, 156(SJH)    ; 1223
                                       09   1A 00049        BGTRU   4$                      ;
                                       56   12 0004B        BNEQ    7$                      ;
                          0098 C2           68   D1 0004D        CMPL    CUR_TIME, 152(SJH)      ;
                                       4F   1F 00052        BLSSU   7$                      ;
                                       52   DD 00054 4$:     PUSHL   SJH                     ; 1226
                                       55   DD 00056        PUSHL   SJH_N                   ;
                00000000G   EF         02   FB 00058        CALLS   #2, UPDATE_GETQUI_DATA  ;
                      68    A4         62   D0 0005F        MOVL    (SJH), 104(SQH)         ; 1227
                                       03   12 00063        BNEQ    5$                      ; 1228
                                 6C    A4   D4 00065        CLRL    108(SQH)                ;
                            56   0134 C2   D0 00068 5$:     MOVL    308(SJH), SMQ_N         ; 1229
                                       29   13 0006D        BEQL    6$                      ; 1230
                                       56   DD 0006F        PUSHL   SMQ_N                   ; 1235
```

```
                                69              01  FB 00071          CALLS   #1, READ_RECORD                          ;
                                53              50  D0 00074          MOVL    R0, SMQ                                 ;
                                          010C  C3  B7 00077          DECW    268(SMQ)                                ; 1236
                                                52  DD 0007B          PUSHL   SJH                                     ; 1237
                                                55  DD 0007D          PUSHL   SJH_N                                   ;
                          0000V CF              02  FB 0007F          CALLS   #2, ENQUEUE_JOB                         ;
                                                56  DD 00084          PUSHL   SMQ_N                                   ; 1238
                00000000G EF                    01  FB 00086          CALLS   #1, REWRITE_RECORD                      ;
                                                55  DD 0008D          PUSHL   SJH_N                                   ; 1239
                00000000G EF                    01  FB 0008F          CALLS   #1, REWRITE_RECORD                      ;
                                                97  11 00096          BRB     2$                                      ; 1230
                                                55  DD 00098  6$:     PUSHL   SJH_N                                   ; 1245
                00000000G EF                    01  FB 0009A          CALLS   #1, RELEASE_RECORD                      ;
                                                8C  11 000A1          BRB     2$                                      ; 1223
                                                01  DD 000A3  7$:     PUSHL   #1                                      ; 1255
                                          0000V CF  9F 000A5          PUSHAB  AFTER_AST                               ;
                                          0098  C2  9F 000A9          PUSHAB  152(SJH)                                ;
                                                7E  D4 000AD          CLRL    -(SP)                                   ;
                00000000G 00                    04  FB 000AF          CALLS   #4, SYS$SETIMR                          ;
                                69              50  D0 000B6          MOVL    R0, STATUS                              ;
                                11              57  E8 000B9          BLBS    STATUS, 8$                              ; 1256
                                                57  DD 000BC          PUSHL   STATUS                                  ; 1258
                                                7E  D4 000BE          CLRL    -(SP)                                   ;
                                    0004845A    8F  DD 000C0          PUSHL   #296026                                 ;
                00000000G 00                    03  FB 000C6          CALLS   #3, LIB$SIGNAL                          ;
                                                01  DD 000CD  8$:     PUSHL   #1                                      ; 1264
                00000000G EF                    01  FB 000CF          CALLS   #1, REWRITE_RECORD                      ;
                00000000G EF                    00  FB 000D6          CALLS   #0, UNLOCK_QUEUE_FILE                   ; 1269
                                                    04 000DD          RET                                             ; 1271

; Routine Size:  222 bytes,    Routine Base:  CODE + 0000
```

```
;  235          1272  1 GLOBAL ROUTINE AFTER_AST: NOVALUE=
;  236          1273  1
;  237          1274  1 !++
;  238          1275  1 !
;  239          1276  1 ! FUNCTIONAL DESCRIPTION:
;  240          1277  1 !       This is the completion AST routine for the timer on the timed job
;  241          1278  1 !       queue.  It requeues all jobs whose expiration time is earlier than or
;  242          1279  1 !       equal to the current time and sets a new timer.
;  243          1280  1 !
;  244          1281  1 ! INPUT PARAMETERS:
;  245          1282  1 !       NONE
;  246          1283  1 !
;  247          1284  1 ! IMPLICIT INPUTS:
;  248          1285  1 !       NONE
;  249          1286  1 !
;  250          1287  1 ! OUTPUT PARAMETERS:
;  251          1288  1 !       NONE
;  252          1289  1 !
;  253          1290  1 ! IMPLICIT OUTPUTS:
;  254          1291  1 !       NONE
;  255          1292  1 !
;  256          1293  1 ! ROUTINE VALUE:
;  257          1294  1 !       NONE
;  258          1295  1 !
;  259          1296  1 ! SIDE EFFECTS:
;  260          1297  1 !       NONE
;  261          1298  1 !
;  262          1299  1 !--
;  263          1300  1
;  264          1301  2 BEGIN
;  265          1302  2 SCHEDULE_NONAST(AFTER_NONAST);
;  266          1303  1 END;


                                        0000 00000      .ENTRY  AFTER_AST, Save nothing                 ; 1272
                                FF1C    CF   9F 00002   PUSHAB  AFTER_NONAST                            ; 1302
                    00000000G   EF              01   FB 00006   CALLS   #1, SCHEDULE_NONAST
                                                04 0000D        RET                                     ; 1303

; Routine Size:  14 bytes,    Routine Base:  CODE + 00DE
```

```
;  268     1304  1 GLOBAL ROUTINE REQUEUE_STARTING_JOBS(SMQ_N,SMQ): NOVALUE=
;  269     1305  1
;  270     1306  1 !++
;  271     1307  1 !
;  272     1308  1 ! FUNCTIONAL DESCRIPTION:
;  273     1309  1 !       This routine requeues starting jobs in a specified queue.
;  274     1310  1 !
;  275     1311  1 ! INPUT PARAMETERS:
;  276     1312  1 !       SMQ_N           - Record number of SMQ.
;  277     1313  1 !       SMQ             - Pointer to SMQ.
;  278     1314  1 !
;  279     1315  1 ! IMPLICIT INPUTS:
;  280     1316  1 !       NONE
;  281     1317  1 !
;  282     1318  1 ! OUTPUT PARAMETERS:
;  283     1319  1 !       NONE
;  284     1320  1 !
;  285     1321  1 ! IMPLICIT OUTPUTS:
;  286     1322  1 !       NONE
;  287     1323  1 !
;  288     1324  1 ! ROUTINE VALUE:
;  289     1325  1 !       NONE
;  290     1326  1 !
;  291     1327  1 ! SIDE EFFECTS:
;  292     1328  1 !       NONE
;  293     1329  1 !
;  294     1330  1 !--
;  295     1331  1
;  296     1332  2 BEGIN
;  297     1333  2 MAP
;  298     1334  2         SMQ:            REF BBLOCK;     ! Pointer to SMQ
;  299     1335  2 LOCAL
;  300     1336  2         PRED_MODIFIED,                  ! True if predecessor modified
;  301     1337  2         SJH_NP,                         ! Record number of predecessor of SJH
;  302     1338  2         SJH_P:          REF BBLOCK,     ! Pointer to predecessor of SJH
;  303     1339  2         SJH_NS,                         ! Record number of successor
;  304     1340  2         SJH_N,                          ! Record number of SJH
;  305     1341  2         SJH:            REF BBLOCK;     ! Pointer to predecessor
;  306     1342  2
;  307     1343  2
;  308     1344  2 PRED_MODIFIED = FALSE;
;  309     1345  2 SJH_NP = 0;
;  310     1346  2 SJH_N = .SMQ[SMQ$L_CURRENT_LIST];
;  311     1347  2 WHILE .SJH_N NEQ 0 DO
;  312     1348  3     BEGIN
;  313     1349  3     SJH = READ_RECORD(.SJH_N);
;  314     1350  3     SJH_NS = .SJH[SYM$L_LINK];
;  315     1351  3     IF .SJH[SJH$V_STARTING]
;  316     1352  3     THEN
;  317     1353  4         BEGIN
;  318     1354  4         UPDATE_GETQUI_DATA(.SJH_N, .SJH);
;  319     1355  4         IF .SJH_NP EQL 0
;  320     1356  4         THEN
;  321     1357  5             BEGIN
;  322     1358  5             SMQ[SMQ$L_CURRENT_LIST] = .SJH_NS;
;  323     1359  5             IF .SJH_NS EQL 0 THEN SMQ[SMQ$L_CURRENT_LIST_END] = 0;
;  324     1360  5             END
```

```
;   325          1361  4          ELSE
;   326          1362  5              BEGIN
;   327          1363  5              SJH_P[SYM$L_LINK] = .SJH_NS;
;   328          1364  5              IF .SJH_NS EQL 0 THEN SMQ[SMQ$L_CURRENT_LIST_END] = .SJH_NP;
;   329          1365  5              PRED_MODIFIED = TRUE;
;   330          1366  4              END;
;   331          1367  4          SMQ[SMQ$B_CURRENT_JOB_COUNT] = .SMQ[SMQ$B_CURRENT_JOB_COUNT] - 1;
;   332          1368  4          ENQUEUE_JOB(.SJH_N, .SJH);
;   333          1369  4          REWRITE_RECORD(.SJH_N);
;   334          1370  4          END
;   335          1371  3      ELSE
;   336          1372  4          BEGIN
;   337          1373  4          IF .SJH_NP NEQ 0
;   338          1374  4          THEN
;   339          1375  4              IF TESTBITSC(PRED_MODIFIED)
;   340          1376  4                  THEN REWRITE_RECORD(.SJH_NP)
;   341          1377  4                  ELSE RELEASE_RECORD(.SJH_NP);
;   342          1378  4          SJH_NP = .SJH_N;
;   343          1379  4          SJH_P = .SJH;
;   344          1380  3          END;
;   345          1381  3      SJH_N = .SJH_NS;
;   346          1382  2      END;
;   347          1383  2 IF .SJH_NP NEQ 0
;   348          1384  2 THEN
;   349          1385  2      IF .PRED_MODIFIED
;   350          1386  2          THEN REWRITE_RECORD(.SJH_NP)
;   351          1387  2          ELSE RELEASE_RECORD(.SJH_NP);
;   352          1388  1 END;
; INFO#250                 L1:1363
; Referenced LOCAL symbol SJH_P is probably not initialized
```

```
                                    0FFC 00000          .ENTRY  REQUEUE_STARTING_JOBS, Save R2,R3,R4,R5,R6,-; 1304
                                                                R7,R8,R9,R10,R11
                    59 00000000G    EF   9E 00002          MOVAB   REWRITE_RECORD, R9
                                    57   D4 00009          CLRL    PRED_MODIFIED                              ; 1344
                                    53   D4 0000B          CLRL    SJH_NP                                     ; 1345
                    52        08    AC   D0 0000D          MOVL    SMQ, R2                                    ; 1346
                    55        48    A2   D0 00011          MOVL    72(R2), SJH_N
                                    71   13 00015 1$:       BEQL    9$                                         ; 1347
                                    55   DD 00017          PUSHL   SJH_N                                      ; 1349
              00000000G     EF              01   FB 00019      CALLS   #1, READ_RECORD
                    54              50   D0 00020          MOVL    R0, SJH
                    56              64   D0 00023          MOVL    (SJH), SJH_NS                              ; 1350
          3A        11    A4              04   E1 00026      BBC     #4, 17(SJH), 5$                            ; 1351
                                    54   DD 0002B          PUSHL   SJH                                        ; 1354
                                    55   DD 0002D          PUSHL   SJH_N
              00000000G     EF              02   FB 0002F      CALLS   #2, UPDATE_GETQUI_DATA
                                    53   D5 00036          TSTL    SJH_NP                                     ; 1355
                                    0B   12 00038          BNEQ    2$
                    48        A2              56   D0 0003A      MOVL    SJH_NS, 72(R2)                         ; 1358
                                    11   12 0003E          BNEQ    4$                                         ; 1359
                                4C  A2   D4 00040          CLRL    76(R2)
                                    0C   11 00043          BRB     4$                                         ; 1355
```

```
                        68              56  DO 00045 2$:    MOVL    SJH_NS, (SJH_P)                 ; 1363
                                        04  12 00048        BNEQ    3$                              ; 1364
                     4C A2              53  DO 0004A        MOVL    SJH_NP, 76(R2)                  ;
                        57              01  DO 0004E 3$:    MOVL    #1, PRED_MODIFIED               ; 1365
                             0115       C2  97 00051 4$:    DECB    277(R2)                         ; 1367
                                        54  DD 00055        PUSHL   SJH                             ; 1368
                                        55  DD 00057        PUSHL   SJH_N                           ;
                   0000V CF             02  FB 00059        CALLS   #2, ENQUEUE_JOB                 ;
                                        55  DD 0005E        PUSHL   SJH_N                           ; 1369
                        69              01  FB 00060        CALLS   #1, REWRITE_RECORD              ;
                                        1E  11 00063        BRB     8$                              ; 1351
                                        53  D5 00065 5$:    TSTL    SJH_NP                          ; 1373
                                        14  13 00067        BEQL    7$                              ;
                07                      57  00 E5 00069    BBCC    #0, PRED_MODIFIED, 6$           ; 1375
                                        53  DD 0006D        PUSHL   SJH_NP                          ; 1376
                        69              01  FB 0006F        CALLS   #1, REWRITE_RECORD              ;
                                        09  11 00072        BRB     7$                              ;
                                        53  DD 00074 6$:    PUSHL   SJH_NP                          ; 1377
                00000000G EF            01  FB 00076        CALLS   #1, RELEASE_RECORD              ;
                        53              55  DO 0007D 7$:    MOVL    SJH_N, SJH_NP                   ; 1378
                        58              54  DO 00080        MOVL    SJH, SJH_P                      ; 1379
                        55              56  DO 00083 8$:    MOVL    SJH_NS, SJH_N                   ; 1381
                                        8D  11 00086        BRB     1$                              ; 1347
                                        53  D5 00088 9$:    TSTL    SJH_NP                          ; 1383
                                        12  13 0008A        BEQL    11$                             ;
                        06              57  E9 0008C        BLBC    PRED_MODIFIED, 10$              ; 1385
                                        53  DD 0008F        PUSHL   SJH_NP                          ; 1386
                        69              01  FB 00091        CALLS   #1, REWRITE_RECORD              ;
                                            04 00094        RET                                     ;
                                        53  DD 00095 10$:   PUSHL   SJH_NP                          ; 1387
                00000000G EF            01  FB 00097        CALLS   #1, RELEASE_RECORD              ;
                                            04 0009E 11$:    RET                                     ; 1388

; Routine Size:  159 bytes,    Routine Base:  CODE + 00EC
```

```
;  354          1389  1 ROUTINE EXECUTOR_SCHEDULING_POLICY(SJH,SMQ_1,SMQ_2)=
;  355          1390  1
;  356          1391  1 !++
;  357          1392  1 !
;  358          1393  1 ! FUNCTIONAL DESCRIPTION:
;  359          1394  1 !       This routine implements the default executor scheduling policy.
;  360          1395  1 !
;  361          1396  1 ! INPUT PARAMETERS:
;  362          1397  1 !       SJH             - Pointer to SJH.
;  363          1398  1 !       SMQ_1           - Pointer to first SMQ.
;  364          1399  1 !       SMQ_2           - Pointer to second SMQ.
;  365          1400  1 !
;  366          1401  1 ! IMPLICIT INPUTS:
;  367          1402  1 !       NONE
;  368          1403  1 !
;  369          1404  1 ! OUTPUT PARAMETERS:
;  370          1405  1 !       NONE
;  371          1406  1 !
;  372          1407  1 ! IMPLICIT OUTPUTS:
;  373          1408  1 !       NONE
;  374          1409  1 !
;  375          1410  1 ! ROUTINE VALUE:
;  376          1411  1 !       True if SMQ_1 is more desirable than SMQ_2; false otherwise.
;  377          1412  1 !
;  378          1413  1 ! SIDE EFFECTS:
;  379          1414  1 !       NONE
;  380          1415  1 !
;  381          1416  1 !--
;  382          1417  1
;  383          1418  2 BEGIN
;  384          1419  2 MAP
;  385          1420  2         SJH:            REF BBLOCK,     ! Pointer to SJH
;  386          1421  2         SMQ_1:          REF BBLOCK,     ! Pointer to SMQ
;  387          1422  2         SMQ_2:          REF BBLOCK;     ! Pointer to SMQ
;  388          1423  2
;  389          1424  2
;  390          1425  2 IF .SMQ_1[SMQ$V_BATCH]
;  391          1426  2 THEN
;  392          1427  3     BEGIN
;  393          1428  3
;  394          1429  3     ! In this implementation, the batch queue that will have the lower
;  395          1430  3     ! percentage utilization is the more desirable.
;  396          1431  3     !
;  397          1432  3     IF
;  398          1433  3         (.SMQ_2[SMQ$B_CURRENT_JOB_COUNT] + 1) * 100 / .SMQ_2[SMQ$B_JOB_LIMIT] LSSU
;  399          1434  3         (.SMQ_1[SMQ$B_CURRENT_JOB_COUNT] + 1) * 100 / .SMQ_1[SMQ$B_JOB_LIMIT]
;  400          1435  3     THEN
;  401          1436  3         RETURN FALSE;
;  402          1437  3     END
;  403          1438  2 ELSE
;  404          1439  3     BEGIN
;  405          1440  3
;  406          1441  3     ! In this implementation, printers on the local node are more desirable
;  407          1442  3     ! than printers on remote nodes, but otherwise printers are equally
;  408          1443  3     ! desirable.
;  409          1444  3     !
;  410          1445  4     IF  SYSID_EQL(SJH[SJH$T_SYSID], SMQ_2[SMQ$T_SYSID])
```

```
;  411          1446  4      AND SYSID_NEQ(SJH[SJH$T_SYSID], SMQ_1[SMQ$T_SYSID])
;  412          1447  3      THEN
;  413          1448  3          RETURN FALSE;
;  414          1449  2      END;
;  415          1450  2
;  416          1451  2
;  417          1452  2 TRUE
;  418          1453  1 END;


                                000C 00000 EXECUTOR_SCHEDULING_POLICY:
                                           .WORD   Save R2,R3                                   ; 1389
                          52        0C   AC  D0 00002          MOVL    SMQ_2, R2               ; 1433
                          50        08   AC  D0 00006          MOVL    SMQ_1, R0               ; 1425
                          37        0C   A0  E9 0000A          BLBC    12(R0), 1$              ;
                          51      0115   C2  9A 0000E          MOVZBL  277(R2), R1             ; 1433
                          51  00000064   8F  C4 00013          MULL2   #100, R1                ;
                          53        64   A1  9E 0001A          MOVAB   100(R1), R3             ;
                          51      0116   C2  9A 0001E          MOVZBL  278(R2), R1             ;
                          53             51  C6 00023          DIVL2   R1, R3                  ;
                          51      0115   C0  9A 00026          MOVZBL  277(R0), R1             ; 1434
                          51  00000064   8F  C4 0002B          MULL2   #100, R1                ;
                          51        64   A1  9E 00032          MOVAB   100(R1), R1             ;
                          52      0116   C0  9A 00036          MOVZBL  278(R0), R2             ;
                          51             52  C6 0003B          DIVL2   R2, R1                  ;
                          51             53  D1 0003E          CMPL    R3, R1                  ;
                                         2D  1E 00041          BGEQU   3$                      ;
                                         28  11 00043          BRB     2$                      ; 1436
                          51        04   AC  D0 00045 1$:      MOVL    SJH, R1                 ; 1445
                    0106  C2      016C  C1  D1 00049          CMPL    364(R1), 262(R2)        ;
                                        1E  12 00050          BNEQ    3$                      ;
                    010A  C2      0170  C1  B1 00052          CMPW    368(R1), 266(R2)        ;
                                        15  12 00059          BNEQ    3$                      ;
                    0106  C0      016C  C1  D1 0005B          CMPL    364(R1), 262(R0)        ; 1446
                                        09  12 00062          BNEQ    2$                      ;
                    010A  C0      0170  C1  B1 00064          CMPW    368(R1), 266(R0)        ;
                                        03  13 0006B          BEQL    3$                      ;
                                        50  D4 0006D 2$:      CLRL    R0                      ; 1448
                                        04  0006F             RET                             ;
                          50              01  D0 00070 3$:      MOVL    #1, R0                  ; 1453
                                        04  00073             RET                             ;

; Routine Size:  116 bytes,    Routine Base:  CODE + 018B
```

```
;  420     1454  1 ROUTINE EXECUTOR_ACCEPTS_JOB(QSMQ_N,QSMQ,ESMQ_N,ESMQ,SJH_N,SJH)=
;  421     1455  1
;  422     1456  1 !++
;  423     1457  1 !
;  424     1458  1 ! FUNCTIONAL DESCRIPTION:
;  425     1459  1 !       This routine determines whether executor queue ESMQ can accept job SJH
;  426     1460  1 !       that was entered in queue QSMQ.
;  427     1461  1 !
;  428     1462  1 ! INPUT PARAMETERS:
;  429     1463  1 !       QSMQ_N          - Record number of job's SMQ.
;  430     1464  1 !       QSMQ            - Pointer to job's SMQ.
;  431     1465  1 !       ESMQ_N          - Record number of executor SMQ.
;  432     1466  1 !       ESMQ            - Pointer to executor SMQ.
;  433     1467  1 !       SJH_N           - Record number of SJH.
;  434     1468  1 !       SJH             - Pointer to SJH.
;  435     1469  1 !
;  436     1470  1 ! IMPLICIT INPUTS:
;  437     1471  1 !       NONE
;  438     1472  1 !
;  439     1473  1 ! OUTPUT PARAMETERS:
;  440     1474  1 !       NONE
;  441     1475  1 !
;  442     1476  1 ! IMPLICIT OUTPUTS:
;  443     1477  1 !       NONE
;  444     1478  1 !
;  445     1479  1 ! ROUTINE VALUE:
;  446     1480  1 !       True if the job can be initiated, false otherwise.
;  447     1481  1 !
;  448     1482  1 ! SIDE EFFECTS:
;  449     1483  1 !       NONE
;  450     1484  1 !
;  451     1485  1 !--
;  452     1486  1
;  453     1487  2 BEGIN
;  454     1488  2 MAP
;  455     1489  2         QSMQ:           REF BBLOCK,     ! Pointer to SMQ
;  456     1490  2         ESMQ:           REF BBLOCK,     ! Pointer to SMQ
;  457     1491  2         SJH:            REF BBLOCK;     ! Pointer to SJH
;  458     1492  2
;  459     1493  2
;  460     1494  2 ! If either the job's queue or the executor queue is not running, fail
;  461     1495  2 ! immediately.
;  462     1496  2 !
;  463     1497  2 IF .ESMQ[SMQ$V_PAUSED]
;  464     1498  2 OR .ESMQ[SMQ$V_PAUSING]
;  465     1499  2 OR .ESMQ[SMQ$V_RESUMING]
;  466     1500  2 OR .ESMQ[SMQ$V_STARTING]
;  467     1501  2 OR .ESMQ[SMQ$V_STOPPED]
;  468     1502  2 OR .ESMQ[SMQ$V_UNAVAILABLE]
;  469     1503  2 OR .QSMQ[SMQ$V_STOPPED]
;  470     1504  2 THEN
;  471     1505  2     RETURN FALSE;
;  472     1506  2
;  473     1507  2
;  474     1508  2 ! If the job's queue is assigned, but not to the executor queue, fail.
;  475     1509  2 !
;  476     1510  2 IF .QSMQ[SMQ$L_ASSIGNED_QUEUE_LINK] NEQ 0
```

```
;  477          1511  2 THEN
;  478          1512  3     BEGIN
;  479          1513  3     IF .QSMQ[SMQ$L_ASSIGNED_QUEUE_LINK] NEQ .ESMQ_N
;  480          1514  3     THEN
;  481          1515  3         RETURN FALSE;
;  482          1516  3     END
;  483          1517  3
;  484          1518  3
;  485          1519  3 ! If the job's queue is generic and the executor queue cannot accept generic
;  486          1520  3 ! selection from this queue, fail.
;  487          1521  3 !
;  488          1522  2 ELSE IF .QSMQ[SMQ$V_GENERIC_QUEUE]
;  489          1523  2 THEN
;  490          1524  3     BEGIN
;  491          1525  3     IF .QSMQ[SMQ$L_GENERIC_TARGET] NEQ 0
;  492          1526  3     THEN
;  493          1527  4         BEGIN
;  494          1528  4         LOCAL
;  495          1529  4             AUX:        REF BBLOCK;     ! Pointer to generic target block
;  496          1530  4
;  497          1531  4         AUX = READ_RECORD(.QSMQ[SMQ$L_GENERIC_TARGET]);
;  498          1532  4         IF
;  499          1533  5             BEGIN
;  500          1534  5             DECR I FROM .VECTOR[AUX[SYM$T_DATA], 0] TO 1 DO
;  501          1535  5                 IF .VECTOR[AUX[SYM$T_DATA], .I] EQL .ESMQ_N
;  502          1536  5                     THEN EXITLOOP FALSE
;  503          1537  5             END
;  504          1538  4         THEN
;  505          1539  5             BEGIN
;  506          1540  5             RELEASE_RECORD(.QSMQ[SMQ$L_GENERIC_TARGET]);
;  507          1541  5             RETURN FALSE;
;  508          1542  5             END
;  509          1543  4         ELSE
;  510          1544  4             RELEASE_RECORD(.QSMQ[SMQ$L_GENERIC_TARGET]);
;  511          1545  4         END
;  512          1546  3     ELSE
;  513          1547  4         BEGIN
;  514          1548  4         IF NOT .ESMQ[SMQ$V_GENERIC_SELECTION]
;  515          1549  4         OR .QSMQ[SMQ$V_TERMINAL] NEQ .ESMQ[SMQ$V_TERMINAL]
;  516          1550  5         OR (.ESMQ[SMQ$V_SERVER]
;  517          1551  5             AND CH$NEQ(
;  518          1552  5                 SMQ$S_PROCESSOR, QSMQ[SMQ$T_PROCESSOR],
;  519          1553  5                 SMQ$S_PROCESSOR, ESMQ[SMQ$T_PROCESSOR]))
;  520          1554  4         THEN
;  521          1555  4             RETURN FALSE;
;  522          1556  3         END;
;  523          1557  3     END
;  524          1558  3
;  525          1559  3
;  526          1560  3 ! Otherwise, the queue must be the same queue.
;  527          1561  3 !
;  528          1562  2 ELSE IF .QSMQ_N NEQ .ESMQ_N
;  529          1563  2 THEN
;  530          1564  2     RETURN FALSE;
;  531          1565  2
;  532          1566  2
;  533          1567  2 ! Ensure that the job count is not exceeded and that the characteristics match.
```

```
;  534          1568  2 !
;  535          1569  2 IF .ESMQ[SMQ$B_CURRENT_JOB_COUNT] GEQU .ESMQ[SMQ$B_JOB_LIMIT]
;  536          1570  2 OR (.(SJH[SJH$T_CHARACTERISTICS]   ) AND NOT .(ESMQ[SMQ$T_CHARACTERISTICS]   )) NEQ 0
;  537          1571  2 OR (.(SJH[SJH$T_CHARACTERISTICS]+ 4) AND NOT .(ESMQ[SMQ$T_CHARACTERISTICS]+ 4)) NEQ 0
;  538          1572  2 OR (.(SJH[SJH$T_CHARACTERISTICS]+ 8) AND NOT .(ESMQ[SMQ$T_CHARACTERISTICS]+ 8)) NEQ 0
;  539          1573  2 OR (.(SJH[SJH$T_CHARACTERISTICS]+12) AND NOT .(ESMQ[SMQ$T_CHARACTERISTICS]+12)) NEQ 0
;  540          1574  2 THEN
;  541          1575  2     RETURN FALSE;
;  542          1576  2
;  543          1577  2
;  544          1578  2 ! Finish with printer queue specific testing.
;  545          1579  2 !
;  546          1580  2 IF NOT .ESMQ[SMQ$V_BATCH]
;  547          1581  2 THEN
;  548          1582  3     BEGIN
;  549          1583  3
;  550          1584  3     ! If the job requires lowercase, ensure that the queue supports it.
;  551          1585  3     !
;  552          1586  4     IF (.SJH[SJH$V_LOWERCASE] AND NOT .ESMQ[SMQ$V_LOWERCASE])
;  553          1587  3     THEN
;  554          1588  3         RETURN FALSE;
;  555          1589  3
;  556          1590  3
;  557          1591  3     ! If the queue has job size limits, ensure that the job is in range.
;  558          1592  3     !
;  559          1593  4     IF (.ESMQ[SMQ$L_JOB_SIZE_MAXIMUM] NEQ 0
;  560          1594  4         AND .SJH[SJH$L_JOB_SIZE] GTRU .ESMQ[SMQ$L_JOB_SIZE_MAXIMUM])
;  561          1595  4     OR (.ESMQ[SMQ$L_JOB_SIZE_MINIMUM] NEQ 0
;  562          1596  4         AND .SJH[SJH$L_JOB_SIZE] LSSU .ESMQ[SMQ$L_JOB_SIZE_MINIMUM])
;  563          1597  3     THEN
;  564          1598  3         RETURN FALSE;
;  565          1599  3
;  566          1600  3
;  567          1601  3     ! Ensure that the form stock names match.
;  568          1602  3     !
;  569          1603  3     IF
;  570          1604  4         BEGIN
;  571          1605  4         IF .SJH[SJH$L_FORM_LINK] EQL .ESMQ[SMQ$L_FORM_LINK]
;  572          1606  4         THEN
;  573          1607  4             FALSE
;  574          1608  4         ELSE
;  575          1609  5             BEGIN
;  576          1610  5             LOCAL
;  577          1611  5                 SFM_1:          REF BBLOCK,     ! Pointer to job's SFM
;  578          1612  5                 SFM_2:          REF BBLOCK;     ! Pointer to queue's SFM
;  579          1613  5
;  580          1614  5             SFM_1 = READ_RECORD(.SJH[SJH$L_FORM_LINK]);
;  581          1615  5             SFM_2 = READ_RECORD(.ESMQ[SMQ$L_FORM_LINK]);
;  582          1616  5             IF CH$EQL(
;  583          1617  5                 SFM$S_STOCK, SFM_1[SFM$T_STOCK],
;  584          1618  5                 SFM$S_STOCK, SFM_2[SFM$T_STOCK])
;  585          1619  5             THEN
;  586          1620  6                 BEGIN
;  587          1621  6                 RELEASE_RECORD(.SJH[SJH$L_FORM_LINK]);
;  588          1622  6                 RELEASE_RECORD(.ESMQ[SMQ$L_FORM_LINK]);
;  589          1623  6                 FALSE
;  590          1624  6                 END
```

```
;  591          1625  5              ELSE
;  592          1626  6                  BEGIN
;  593          1627  6                  RELEASE_RECORD(.SJH[SJH$L_FORM_LINK]);
;  594          1628  6                  RELEASE_RECORD(.ESMQ[SMQ$L_FORM_LINK]);
;  595          1629  6                  TRUE
;  596          1630  6                  END
;  597          1631  5              END
;  598          1632  4          END
;  599          1633  3      THEN
;  600          1634  3          RETURN FALSE;
;  601          1635  2      END;
;  602          1636  2
;  603          1637  2
;  604          1638  2 TRUE
;  605          1639  1 END;


                                00FC 00000 EXECUTOR_ACCEPTS_JOB:
                                           .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1454
                            57 00000000G   EF  9E 00002        MOVAB   READ_RECORD, R7
                            56 00000000G   EF  9E 00009        MOVAB   RELEASE_RECORD, R6
                            54       10    AC  D0 00010        MOVL    ESMQ, R4                ; 1497
                            50       10    A4  9E 00014        MOVAB   16(R4), R0
                    03      60             02  E1 00018        BBC     #2, (R0), 2$
                                      0130 31  00001C 1$:       BRW     18$
                    F9      60             03  E0 0001F 2$:    BBS     #3, (R0), 1$            ; 1498
                    F5      60             06  E0 00023        BBS     #6, (R0), 1$            ; 1499
                            F1       01    A0  E8 00027        BLBS    1(R0), 1$               ; 1500
                    ED      60             09  E0 0002B        BBS     #9, (R0), 1$            ; 1501
                    E9      60             0B  E0 0002F        BBS     #11, (R0), 1$           ; 1502
                            50       08    AC  D0 00033        MOVL    QSMQ, R0                ; 1503
                    E0   11 A0             01  E0 00037        BBS     #1, 17(R0), 1$
                            52       08    AC  D0 0003C        MOVL    QSMQ, R2                ; 1510
                                     2C    A2  D5 00040        TSTL    44(R2)
                                           07  13 00043        BEQL    3$
                         0C AC       2C    A2  D1 00045        CMPL    44(R2), ESMQ_N          ; 1513
                                           51  11 0004A        BRB     9$
                    47   0D A2             02  E1 0004C 3$:    BBC     #2, 13(R2), 8$          ; 1522
                            53       74    A2  D0 00051        MOVL    116(R2), R3             ; 1525
                                           23  13 00055        BEQL    7$
                                           53  DD 00057        PUSHL   R3                      ; 1531
                                     67        01  FB 00059        CALLS   #1, READ_RECORD
                    51   0C A0             01  C1 0005C        ADDL3   #1, 12(A0X), I          ; 1535
                                           08  11 00061        BRB     5$
                         0C AC       0C A041  D1 00063 4$:    CMPL    12(AUX)[I], ESMQ_N
                                           08  13 00069        BEQL    6$
                            F5             51  F5 0006B 5$:    SOBGTR  I, 4$
                                           53  DD 0006E        PUSHL   R3                      ; 1540
                                        00D9 31  00070        BRW     17$
                                           53  DD 00073 6$:    PUSHL   R3                      ; 1544
                                     66        01  FB 00075        CALLS   #1, RELEASE_RECORD
                                           25  11 00078        BRB     10$                     ; 1525
                    9D   0D A4             03  E1 0007A 7$:    BBC     #3, 13(R4), 1$          ; 1548
                    50   0E A4       0E    A2  8D 0007F        XORB3   14(R2), 14(R4), R0      ; 1549
```

```
                93          50          06  E0 00085          BBS     #6, R0, 1$                                          ;
                11       0E A4          04  E1 00089          BBC     #4, 14(R4), 10$                                     ; 1550
             00D4    C4 00D4 C2          28  29 0008E          CMPC3   #40, 212(R2), 212(R4)                               ; 1553
                             05  11 00096          BRB     9$                                                          ;
                          0C AC    04     AC  D1 00098 8$:    CMPL    QSMQ_N, ESMQ_N                                      ; 1562
                             09  12 0009D 9$:     BNEQ    11$                                                         ;
                       0116 C4    0115   C4  91 0009F 10$:    CMPB    277(R4), 278(R4)                                    ; 1569
                             03  1F 000A6          BLSSU   12$                                                         ;
                         00A4    31 000A8 11$:    BRW     18$                                                         ;
                          50 18 AC  D0 000AB 12$:    MOVL    SJH, R0                                                     ; 1570
                          51 30 A4  D2 000AF         MCOML   48(R4), R1                                                  ;
                          51 00A0 C0  D3 000B3       BITL    160(R0), R1                                                 ;
                             EE  12 000B8          BNEQ    11$                                                         ;
                          50 18 AC  D0 000BA         MOVL    SJH, R0                                                     ; 1571
                          51 34 A4  D2 000BE         MCOML   52(R4), R1                                                  ;
                          51 00A4 C0  D3 000C2       BITL    164(R0), R1                                                 ;
                             DF  12 000C7          BNEQ    11$                                                         ;
                          50 18 AC  D0 000C9         MOVL    SJH, R0                                                     ; 1572
                          51 38 A4  D2 000CD         MCOML   56(R4), R1                                                  ;
                          51 00A8 C0  D3 000D1       BITL    168(R0), R1                                                 ;
                             77  12 000D6          BNEQ    18$                                                         ;
                          50 18 AC  D0 000D8         MOVL    SJH, R0                                                     ; 1573
                          51 3C A4  D2 000DC         MCOML   60(R4), R1                                                  ;
                          51 00AC C0  D3 000E0       BITL    172(R0), R1                                                 ;
                             68  12 000E5          BNEQ    18$                                                         ;
                          67 0C A4  E8 000E7         BLBS    12(R4), 19$                                                 ; 1580
                          50 18 AC  D0 000EB         MOVL    SJH, R0                                                     ; 1586
                    04    0D A0  05  E1 000EF      BBC     #5, 13(R0), 13$                                             ;
                          57 0E A4  E9 000F4         BLBC    14(R4), 18$                                                 ;
                          51 0080 C4  D0 000F8 13$:  MOVL    128(R4), R1                                                 ; 1593
                             07  13 000FD          BEQL    14$                                                         ;
                          51 0100 C0  D1 000FF       CMPL    256(R0), R1                                                 ; 1594
                             49  1A 00104          BGTRU   18$                                                         ;
                          51 0084 C4  D0 00106 14$:  MOVL    132(R4), R1                                                 ; 1595
                             07  13 0010B          BEQL    15$                                                         ;
                          51 0100 C0  D1 0010D       CMPL    256(R0), R1                                                 ; 1596
                             3B  1F 00112          BLSSU   18$                                                         ;
                          55 00FC C0  D0 00114 15$:  MOVL    252(R0), R5                                                 ; 1605
                       70 A4          55  D1 00119       CMPL    R5, 112(R4)                                         ;
                                      33  13 0011D       BEQL    19$                                                 ;
                                      55  DD 0011F       PUSHL   R5                                                  ; 1614
                          67          01  FB 00121       CALLS   #1, READ_RECORD                                     ;
                          52          50  D0 00124       MOVL    R0, SFM_T                                           ;
                             70 A4    DD 00127           PUSHL   112(R4)                                             ; 1615
                          67          01  FB 0012A       CALLS   #1, READ_RECORD                                     ;
0134    C0       0134    C2          20  29 0012D       CMPC3   #32, 308(SFM_1), 308(SFM_2)                         ; 1618
                             0D  12 00135          BNEQ    16$                                                         ;
                                      55  DD 00137       PUSHL   R5                                                  ; 1621
                          66          01  FB 00139       CALLS   #1, RELEASE_RECORD                                  ;
                             70 A4    DD 0013C           PUSHL   112(R4)                                             ; 1622
                          66          01  FB 0013F       CALLS   #1, RELEASE_RECORD                                  ;
                                      0E  11 00142       BRB     19$                                                 ;
                                      55  DD 00144 16$:  PUSHL   R5                                                  ; 1627
                          66          01  FB 00146       CALLS   #1, RELEASE_RECORD                                  ;
                             70 A4    DD 00149           PUSHL   112(R4)                                             ; 1628
                          66          01  FB 0014C 17$:  CALLS   #1, RELEASE_RECORD                                  ;
                                      50  D4 0014F 18$:  CLRL    R0                                                  ; 1634
```

```
                                        04 00151                RET
                          50        01  D0 00152 19$:           MOVL    #1, R0                                  ;  1639
                                        04 00155                RET                                             ;
```

; Routine Size: 342 bytes,   Routine Base: CODE + 01FF

; R

```
;  607     1640  1 ROUTINE FIND_AVAILABLE_EXECUTOR(SMQ_N,SMQ,SJH_N,SJH; ESMQ): L_OUTPUT_1=
;  608     1641  1
;  609     1642  1 !++
;  610     1643  1 !
;  611     1644  1 ! FUNCTIONAL DESCRIPTION:
;  612     1645  1 !       This routine searches for an executor that can accept a job, and if one
;  613     1646  1 !       is found, initiates the job.
;  614     1647  1 !
;  615     1648  1 ! INPUT PARAMETERS:
;  616     1649  1 !       SMQ_N           - Record number of SMQ.
;  617     1650  1 !       SMQ             - Pointer to SMQ.
;  618     1651  1 !       SJH_N           - Record number of SJH.
;  619     1652  1 !       SJH             - Pointer to SJH.
;  620     1653  1 !
;  621     1654  1 ! IMPLICIT INPUTS:
;  622     1655  1 !       NONE
;  623     1656  1 !
;  624     1657  1 ! OUTPUT PARAMETERS:
;  625     1658  1 !       ESMQ            - Pointer to executor SMQ.
;  626     1659  1 !
;  627     1660  1 ! IMPLICIT OUTPUTS:
;  628     1661  1 !       NONE
;  629     1662  1 !
;  630     1663  1 ! ROUTINE VALUE:
;  631     1664  1 !       True if the job was initiated, false otherwise.
;  632     1665  1 !
;  633     1666  1 ! SIDE EFFECTS:
;  634     1667  1 !       NONE
;  635     1668  1 !
;  636     1669  1 !--
;  637     1670  1
;  638     1671  2 BEGIN
;  639     1672  2 MAP
;  640     1673  2         SMQ:            REF BBLOCK,     ! Pointer to SMQ
;  641     1674  2         SJH:            REF BBLOCK,     ! Pointer to SJH
;  642     1675  2         ESMQ:           REF BBLOCK;     ! Pointer to ESMQ
;  643     1676  2 LOCAL
;  644     1677  2         ESMQ_N,                         ! Record number of ESMQ
;  645     1678  2         CSMQ_N,                         ! Record number of CSMQ
;  646     1679  2         CSMQ:           REF BBLOCK;     ! Pointer to CSMQ
;  647     1680  2
;  648     1681  2
;  649     1682  2 ! If the queue to which the job was submitted is not running, fail immediately.
;  650     1683  2 !
;  651     1684  2 IF .SMQ[SMQ$V_PAUSED]
;  652     1685  2 OR .SMQ[SMQ$V_PAUSING]
;  653     1686  2 OR .SMQ[SMQ$V_RESUMING]
;  654     1687  2 OR .SMQ[SMQ$V_STARTING]
;  655     1688  2 OR .SMQ[SMQ$V_STOPPED]
;  656     1689  2 OR .SMQ[SMQ$V_UNAVAILABLE]
;  657     1690  2 THEN
;  658     1691  2     RETURN FALSE;
;  659     1692  2
;  660     1693  2
;  661     1694  2 ! If the queue to which the job was submitted is a logical queue, then the
;  662     1695  2 ! job must execute on the queue assigned to the logical queue.  Determine if
;  663     1696  2 ! that queue can accept the job.  Otherwise, fail immediately.
```

```
;  664     1697  2 !
;  665     1698  2 IF .SMQ[SMQ$L_ASSIGNED_QUEUE_LINK] NEQ 0
;  666     1699  2 THEN
;  667     1700  3     BEGIN
;  668     1701  3     ESMQ = READ_RECORD(ESMQ_N = .SMQ[SMQ$L_ASSIGNED_QUEUE_LINK]);
;  669     1702  3     IF NOT EXECUTOR_ACCEPTS_JOB(.SMQ_N, .SMQ, .ESMQ_N, .ESMQ, .SJH_N, .SJH)
;  670     1703  3     THEN
;  671     1704  4         BEGIN
;  672     1705  4         RELEASE_RECORD(.ESMQ_N);
;  673     1706  4         RETURN FALSE;
;  674     1707  3         END;
;  675     1708  3     END
;  676     1709  3
;  677     1710  3
;  678     1711  3 ! If the queue to which the job was submitted is an executor queue, then the
;  679     1712  3 ! job must execute on that queue.  Determine if that queue can accept the job.
;  680     1713  3 ! Otherwise, fail immediately.
;  681     1714  3 !
;  682     1715  2 ELSE IF NOT .SMQ[SMQ$V_GENERIC_QUEUE]
;  683     1716  2 THEN
;  684     1717  3     BEGIN
;  685     1718  3     IF NOT EXECUTOR_ACCEPTS_JOB(.SMQ_N, .SMQ, .SMQ_N, .SMQ, .SJH_N, .SJH)
;  686     1719  3     THEN
;  687     1720  3         RETURN FALSE;
;  688     1721  3
;  689     1722  3     ESMQ_N = .SMQ_N;
;  690     1723  3     ESMQ = .SMQ;
;  691     1724  3     END
;  692     1725  3
;  693     1726  3
;  694     1727  3 ! If the queue to which the job was submitted is a generic queue that has
;  695     1728  3 ! specific target queues, search the target queues to locate a queue on
;  696     1729  3 ! which the job can execute.
;  697     1730  3 !
;  698     1731  2 ELSE IF .SMQ[SMQ$L_GENERIC_TARGET] NEQ 0
;  699     1732  2 THEN
;  700     1733  3     BEGIN
;  701     1734  3     LOCAL
;  702     1735  3         AUX:            REF BBLOCK;     ' Pointer to generic list
;  703     1736  3                                                                          ; R
;  704     1737  3
;  705     1738  3     AUX = READ_RECORD(.SMQ[SMQ$L_GENERIC_TARGET]);
;  706     1739  3     ESMQ_N = 0;
;  707     1740  3     INCR I FROM 1 TO .VECTOR[AUX[SYM$T_DATA], 0] DO
;  708     1741  4         BEGIN
;  709     1742  4         CSMQ_N = .VECTOR[AUX[SYM$T_DATA], .I];
;  710     1743  4         CSMQ = READ_RECORD(.CSMQ_N);
;  711     1744  4         IF .CSMQ[SMQ$V_BATCH] EQL .SMQ[SMQ$V_BATCH]
;  712     1745  4         AND NOT .CSMQ[SMQ$V_GENERIC_QUEUE]
;  713     1746  4         AND .CSMQ[SMQ$L_ASSIGNED_QUEUE_LINK] EQL 0
;  714     1747  4         THEN
;  715     1748  4             IF EXECUTOR_ACCEPTS_JOB(.SMQ_N, .SMQ, .CSMQ_N, .CSMQ, .SJH_N, .SJH)
;  716     1749  4             THEN
;  717     1750  5                 BEGIN
;  718     1751  5                 IF .ESMQ_N EQL 0
;  719     1752  5                 THEN
;  720     1753  6                     BEGIN
```

```
;  721     1754  6                   ESMQ_N = .CSMQ_N;
;  722     1755  6                   ESMQ = .CSMQ;
;  723     1756  6                   END
;  724     1757  5               ELSE IF NOT EXECUTOR_SCHEDULING_POLICY(.SJH, .ESMQ, .CSMQ)
;  725     1758  5               THEN
;  726     1759  6                   BEGIN
;  727     1760  6                   RELEASE_RECORD(.ESMQ_N);
;  728     1761  6                   ESMQ_N = .CSMQ_N;
;  729     1762  6                   ESMQ = .CSMQ;
;  730     1763  5                   END;
;  731     1764  4               END;
;  732     1765  4           IF .CSMQ_N NEQ .ESMQ_N THEN RELEASE_RECORD(.CSMQ_N);
;  733     1766  3           END;
;  734     1767  3       RELEASE_RECORD(.SMQ[SMQ$L_GENERIC_TARGET]);
;  735     1768  3       IF .ESMQ_N EQL 0 THEN RETURN FALSE;
;  736     1769  3       END
;  737     1770  3
;  738     1771  3
;  739     1772  3   ! If the queue to which the job was submitted is a generic queue, execute a
;  740     1773  3   ! full search of the queues to locate a queue on which the job can execute.
;  741     1774  3   ! CSMQ is the candidate queue, and ESMQ is the most desirable of the queues
;  742     1775  3   ! on which the job can execute.
;  743     1776  3   !
;  744     1777  2   ELSE
;  745     1778  3       BEGIN
;  746     1779  3       LOCAL
;  747     1780  3           SQX:            REF BBLOCK,     ! Pointer to SQX
;  748     1781  3           SQX_N,                          ! Record number of SQX
;  749     1782  3           SQX_NS,                         ! Record number of successor of SQX
;  750     1783  3           SQE:            REF BBLOCK,     ! Pointer to SQX entry
;  751     1784  3
;  752     1785  3
;  753     1786  3       ! Scan the queue index for an available queue.
;  754     1787  3       !
;  755     1788  3       ESMQ_N = 0;
;  756     1789  3       SQX = READ_RECORD(SQH$K_RECNO);
;  757     1790  3       SQX_N = .SQX[SQH$L_QUEUE_INDEX_LIST];
;  758     1791  3       RELEASE_RECORD(SQH$K_RECNO);
;  759     1792  3       WHILE .SQX_N NEQ 0 DO
;  760     1793  4           BEGIN
;  761     1794  4
;  762     1795  4           ! Read the queue index record.
;  763     1796  4           !
;  764     1797  4           SQX = READ_RECORD(.SQX_N);
;  765     1798  4
;  766     1799  4
;  767     1800  4           ! Scan the queue index record.
;  768     1801  4           !
;  769     1802  4           SQE = SQX[SYM$T_DATA];
;  770     1803  4           INCR SQE_N FROM 0 TO SQX$K_ENTRIES-1 DO
;  771     1804  5               BEGIN
;  772     1805  5               IF CH$RCHAR(SQE[SQX$T_NAME]) EQL 0
;  773     1806  5               THEN
;  774     1807  5                   EXITLOOP
;  775     1808  5               ELSE
;  776     1809  6                   BEGIN
;  777     1810  6                   IF .SQE[SQX$V_BATCH] EQL .SMQ[SMQ$V_BATCH]
```

```
;  778          1811  6                AND .SQE[SQX$V_EXECUTOR]
;  779          1812  6                AND .SQE[SQX$V_GENERIC_SELECTION]
;  780          1813  6                THEN
;  781          1814  7                    BEGIN
;  782          1815  7                    CSMQ = READ_RECORD(CSMQ_N = .SQE[SQX$L_QUEUE_LINK]);
;  783          1816  7                    IF EXECUTOR_ACCEPTS_JOB(
;  784          1817  7                        .SMQ_N, .SMQ, .CSMQ_N, .CSMQ, .SJH_N, .SJH)
;  785          1818  7                    THEN
;  786          1819  8                        BEGIN
;  787          1820  8                        IF .ESMQ_N EQL 0
;  788          1821  8                        THEN
;  789          1822  9                            BEGIN
;  790          1823  9                            ESMQ_N = .CSMQ_N;
;  791          1824  9                            ESMQ = .CSMQ;
;  792          1825  9                            END
;  793          1826  8                        ELSE IF NOT EXECUTOR_SCHEDULING_POLICY(
;  794          1827  8                            .SJH, .ESMQ, .CSMQ)
;  795          1828  8                        THEN
;  796          1829  9                            BEGIN
;  797          1830  9                            RELEASE_RECORD(.ESMQ_N);
;  798          1831  9                            ESMQ_N = .CSMQ_N;
;  799          1832  9                            ESMQ = .CSMQ;
;  800          1833  8                            END;
;  801          1834  7                        END;
;  802          1835  7                    IF .CSMQ_N NEQ .ESMQ_N THEN RELEASE_RECORD(.CSMQ_N);
;  803          1836  6                    END;
;  804          1837  5                END;
;  805          1838  5            SQE = .SQE + SQX$S_SQX;
;  806          1839  4            END;
;  807          1840  4
;  808          1841  4
;  809          1842  4        ! Advance to the next index block.
;  810          1843  4        !
;  811          1844  4        SQX_NS = .SQX[SYM$L_LINK];
;  812          1845  4        RELEASE_RECORD(.SQX_N);
;  813          1846  4        SQX_N = .SQX_NS;
;  814          1847  3        END;
;  815          1848  3
;  816          1849  3
;  817          1850  3    IF .ESMQ_N EQL 0 THEN RETURN FALSE;
;  818          1851  2    END;
;  819          1852  2
;  820          1853  2
;  821          1854  2 ! ESMQ has been determined to be available and acceptable for the job.
;  822          1855  2 ! Initiate the job.
;  823          1856  2 !
;  824          1857  2 UPDATE_GETQUI_DATA(.SJH_N, .SJH);
;  825          1858  2 START_EXECUTOR_JOB(.ESMQ_N, .ESMQ, .SJH_N, .SJH);
;  826          1859  2 IF .ESMQ_N NEQ .SMQ_N
;  827          1860  2 THEN
;  828          1861  3     BEGIN
;  829          1862  3     READ_RECORD(.ESMQ_N);
;  830          1863  3     REWRITE_RECORD(.ESMQ_N);
;  831          1864  2     END;
;  832          1865  2 TRUE
;  833          1866  1 END;
```

```
                                07FC 00000 FIND_AVAILABLE_EXECUTOR:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                ; 1640
                          52    08   AC  D0 00002        MOVL    SMQ, R2                           ; 1684
                          50    10   A2  9E 00006        MOVAB   16(R2), R0
                47        60         02  E0 0000A        BBS     #2, (R0), 1$
                43        60         03  E0 0000E        BBS     #3, (R0), 1$                      ; 1685
                3F        60         06  E0 00012        BBS     #6, (R0), 1$                      ; 1686
                          3B    01   A0  E8 00016        BLBS    1(R0), 1$                         ; 1687
                37        60         09  E0 0001A        BBS     #9, (R0), 1$                      ; 1688
                33        60         0B  E0 0001E        BBS     #11, (R0), 1$                     ; 1689
                                2C   A2  D5 00022        TSTL    44(R2)                            ; 1698
                                     31  13 00025        BEQL    2$
                          5B    2C   A2  D0 00027        MOVL    44(R2), ESMQ_N                    ; 1701
                                     5B  DD 0002B        PUSHL   ESMQ_N
                00000000G         EF   01  FB 0002D        CALLS   #1, READ_RECORD
                          57         50  D0 00034        MOVL    R0, ESMQ
                          7E    0C   AC  7D 00037        MOVQ    SJH_N, -(SP)                      ; 1702
                                     57  DD 0003B        PUSHL   ESMQ
                                0804 8F  BB 0003D        PUSHR   #^M<R2,R11>
                                   04   AC  DD 00041        PUSHL   SMQ_N
                     FE61       CF        06  FB 00044        CALLS   #6, EXECUTOR_ACCEPTS_JOB
                          2E         50  E8 00049        BLBS    R0, 3$
                                     5B  DD 0004C        PUSHL   ESMQ_N                            ; 1705
                00000000G         EF   01  FB 0004E        CALLS   #1, RELEASE_RECORD
                                 0173 31  00055 1$:      BRW     19$                               ; 1706
                20        0D     A2        02  E0 00058 2$:      BBS     #2, 13(R2), 4$            ; 1715
                          7E    0C   AC  7D 0005D        MOVQ    SJH_N, -(SP)                      ; 1718
                                     52  DD 00061        PUSHL   R2
                                   04   AC  DD 00063        PUSHL   SMQ_N
                                     52  DD 00066        PUSHL   R2
                                   04   AC  DD 00068        PUSHL   SMQ_N
                     FE3A       CF        06  FB 0006B        CALLS   #6, EXECUTOR_ACCEPTS_JOB
                          E2         50  E9 00070        BLBC    R0, 1$
                          5B    04   AC  D0 00073        MOVL    SMQ_N, ESMQ_N                     ; 1722
                          57         52  D0 00077        MOVL    R2, ESMQ                          ; 1723
                                 0152 31  0007A 3$:      BRW     20$                               ; 1715
                                     5B  D4 0007D 4$:      CLRL    ESMQ_N                            ; 1739
                                   74   A2  D5 0007F        TSTL    116(R2)                           ; 1731
                                     03  12 00082        BNEQ    5$
                                 008C 31  00084        BRW     11$
                                   74   A2  DD 00087 5$:      PUSHL   116(R2)                       ; 1738
                00000000G         EF   01  FB 0008A        CALLS   #1, READ_RECORD
                          55         50  D0 00091        MOVL    R0, AUX
                                     53  D4 00094        CLRL    I                                 ; 1740
                                     69  11 00096        BRB     9$
                          56    0C A543  D0 00098 6$:      MOVL    12(AUX)[I], CSMQ_N                ; 1742
                                     56  DD 0009D        PUSHL   CSMQ_N                            ; 1743
                00000000G         EF   01  FB 0009F        CALLS   #1, READ_RECORD
                          54         50  D0 000A6        MOVL    R0, CSMQ
                50        0C     A2   0C  A4  8D 000A9        XORB3   12(CSMQ), 12(R2), R0          ; 1744
                          41         50  E8 000AF        BLBS    R0, 8$
                3C        0D     A4        02  E0 000B2        BBS     #2, 13(CSMQ), 8$              ; 1745
                                     2C  A4  D5 000B7        TSTL    44(CSMQ)                          ; 1746
```

```
                                 37  12 000BA           BNEQ    8$                                  ;
                    7E       0C  AC  7D 000BC           MOVQ    SJH_N, -(SP)                        ; 1748
                                 54  DD 000C0           PUSHL   CSMQ                                ;
                           0044  8F  BB 000C2           PUSHR   #^M<R2,R6>                          ;
                             04  AC  DD 000C6           PUSHL   SMQ_N                               ;
              FDDC  CF           06  FB 000C9           CALLS   #6, EXECUTOR_ACCEPTS_JOB            ;
                    22           50  E9 000CE           BLBC    R0, 8$                              ;
                                 5B  D5 000D1           TSTL    ESMQ_N                              ; 1751
                                 18  13 000D3           BEQL    7$                                  ;
                                 54  DD 000D5           PUSHL   CSMQ                                ; 1757
                                 57  DD 000D7           PUSHL   ESMQ                                ;
                             10  AC  DD 000D9           PUSHL   SJH                                 ;
              FD55  CF           03  FB 000DC           CALLS   #3, EXECUTOR_SCHEDULING_POLICY      ;
                    0F           50  E8 000E1           BLBS    R0, 8$                              ;
                                 5B  DD 000E4           PUSHL   ESMQ_N                              ; 1760
         00000000G  EF           01  FB 000E6           CALLS   #1, RELEASE_RECORD                  ;
                    5B           56  D0 000ED 7$:       MOVL    CSMQ_N, ESMQ_N                      ; 1761
                    57           54  D0 000F0           MOVL    CSMQ, ESMQ                          ; 1762
                    5B           56  D1 000F3 8$:       CMPL    CSMQ_N, ESMQ_N                      ; 1765
                                 09  13 000F6           BEQL    9$                                  ;
                                 56  DD 000F8           PUSHL   CSMQ_N                              ;
         00000000G  EF           01  FB 000FA           CALLS   #1, RELEASE_RECORD                  ;
    92              53       0C  A5  F3 00101 9$:       AOBLEQ  12(AUX), I, 6$                      ; 1740
                             74  A2  DD 00106           PUSHL   116(R2)                             ; 1767
         00000000G  EF           01  FB 00109           CALLS   #1, RELEASE_RECORD                  ;
                               00B4  31 00110 10$:      BRW     18$                                 ; 1768
                                 01  DD 00113 11$:      PUSHL   #1                                  ; 1789
         00000000G  EF           01  FB 00115           CALLS   #1, READ_RECORD                     ;
                    58           50  D0 0011C           MOVL    R0, SQX                             ;
                    59       64  A8  D0 0011F           MOVL    100(SQX), SQX_N                     ; 1790
                                 01  DD 00123           PUSHL   #1                                  ; 1791
         00000000G  EF           01  FB 00125           CALLS   #1, RELEASE_RECORD                  ;
                                 59  D5 0012C 12$:      TSTL    SQX_N                               ; 1792
                                 E0  13 0012E           BEQL    10$                                 ;
                                 59  DD 00130           PUSHL   SQX_N                               ; 1797
         00000000G  EF           01  FB 00132           CALLS   #1, READ_RECORD                     ;
                    58           50  D0 00139           MOVL    R0, SQX                             ;
                    53       0C  A8  9E 0013C           MOVAB   12(R8), SQE                         ; 1802
                                 55  D4 00140           CLRL    SQE_N                               ; 1803
                                 63  95 00142 13$:      TSTB    (SQE)                               ; 1805
                                 6F  13 00144           BEQL    17$                                 ;
    50        0C    A2       20  A3  8D 00146           XORB3   32(SQE), 12(R2), R0                 ; 1810
                    5F           50  E8 0014C           BLBS    R0, 16$                             ;
    5A        20    A3           01  E1 0014F           BBC     #1, 32(SQE), 16$                    ; 1811
    55        20    A3           02  E1 00154           BBC     #2, 32(SQE), 16$                    ; 1812
                    56       24  A3  D0 00159           MOVL    36(SQE), CSMQ_N                     ; 1815
                                 56  DD 0015D           PUSHL   CSMQ_N                              ;
         00000000G  EF           01  FB 0015F           CALLS   #1, READ_RECORD                     ;
                    54           50  D0 00166           MOVL    R0, CSMQ                            ;
                    7E       0C  AC  7D 00169           MOVQ    SJH_N, -(SP)                        ; 1817
                                 54  DD 0016D           PUSHL   CSMQ                                ;
                           0044  8F  BB 0016F           PUSHR   #^M<R2,R6>                          ;
                             04  AC  DD 00173           PUSHL   SMQ_N                               ;
              FD2F  CF           06  FB 00176           CALLS   #6, EXECUTOR_ACCEPTS_JOB            ;
                    22           50  E9 0017B           BLBC    R0, 15$                             ;
                                 5B  D5 0017E           TSTL    ESMQ_N                              ; 1820
                                 18  13 00180           BEQL    14$                                 ;
```

```
                        54  DD 00182          PUSHL   CSMQ                                    ; 1827
                        57  DD 00184          PUSHL   ESMQ
                    10  AC  DD 00186          PUSHL   SJH
                FCA8    CF  03  FB 00189      CALLS   #3, EXECUTOR_SCHEDULING_POLICY
                        0F  50  E8 0018E      BLBS    R0, 15$
                            5B  DD 00191      PUSHL   ESMQ_N                                  ; 1830
            00000000G   EF      01  FB 00193  CALLS   #1, RELEASE_RECORD
                        5B      56  D0 0019A 14$:   MOVL    CSMQ_N, ESMQ_N                  ; 1831
                        57      54  D0 0019D    MOVL    CSMQ, ESMQ                          ; 1832
                        5B      56  D1 001A0 15$:   CMPL    CSMQ_N, ESMQ_N                  ; 1835
                                09  13 001A3      BEQL    16$
                                56  DD 001A5      PUSHL   CSMQ_N
            00000000G   EF      01  FB 001A7      CALLS   #1, RELEASE_RECORD
                        53      28  C0 001AE 16$:   ADDL2   #40, SQE                        ; 1838
                8D      55      0B  F3 001B1    AOBLEQ  #11, SQE_N, 13$                     ; 1803
                        5A      68  D0 001B5 17$:   MOVL    (SQX), SQX_NS                   ; 1844
                                59  DD 001B8      PUSHL   SQX_N                           ; 1845
            00000000G   EF      01  FB 001BA      CALLS   #1, RELEASE_RECORD
                        59      5A  D0 001C1    MOVL    SQX_NS, SQX_N                       ; 1846
                            FF65  31 001C4      BRW     12$                                 ; 1792
                                5B  D5 001C7 18$:   TSTL    ESMQ_N                          ; 1850
                                04  12 001C9      BNEQ    20$
                                50  D4 001CB 19$:   CLRL    R0
                                35  11 001CD      BRB     22$
                    7E  0C  AC  7D 001CF 20$:   MOVQ    SJH_N, -(SP)                        ; 1857
            00000000G   EF      02  FB 001D3    CALLS   #2, UPDATE_GETQUI_DATA
                    7E  0C  AC  7D 001DA        MOVQ    SJH_N, -(SP)                        ; 1858
                                57  DD 001DE      PUSHL   ESMQ
                                5B  DD 001E0      PUSHL   ESMQ_N
            00000000G   EF      04  FB 001E2      CALLS   #4, START_EXECUTOR_JOB
                    04  AC      5B  D1 001E9    CMPL    ESMQ_N, SMQ_N                       ; 1859
                                12  13 001ED      BEQL    21$
                                5B  DD 001EF      PUSHL   ESMQ_N                          ; 1862
                    00000000G   EF      01  FB 001F1  CALLS   #1, READ_RECORD
                                5B  DD 001F8      PUSHL   ESMQ_N                          ; 1863
                    00000000G   EF      01  FB 001FA  CALLS   #1, REWRITE_RECORD
                                50      01  D0 00201 21$:   MOVL    #1, R0                  ; 1866
                                5B      57  D0 00204 22$:   MOVL    R7, R11
                                            04 00207    RET

; Routine Size:  520 bytes,    Routine Base:  CODE + 0355
```

```
;  835     1867  1 GLOBAL ROUTINE FIND_PENDING_JOBS(SMQ_N,SMQ): NOVALUE=
;  836     1868  1
;  837     1869  1 !++
;  838     1870  1 !
;  839     1871  1 ! FUNCTIONAL DESCRIPTION:
;  840     1872  1 !       This routine searches for pending jobs from a specified generic or
;  841     1873  1 !       logical queue, or that can be executed on a specified executor queue,
;  842     1874  1 !       and starts as many of these jobs as possible.
;  843     1875  1 !
;  844     1876  1 ! INPUT PARAMETERS:
;  845     1877  1 !       SMQ_N           - Record number of SMQ.
;  846     1878  1 !       SMQ             - Pointer to SMQ.
;  847     1879  1 !
;  848     1880  1 ! IMPLICIT INPUTS:
;  849     1881  1 !       NONE
;  850     1882  1 !
;  851     1883  1 ! OUTPUT PARAMETERS:
;  852     1884  1 !       NONE
;  853     1885  1 !
;  854     1886  1 ! IMPLICIT OUTPUTS:
;  855     1887  1 !       NONE
;  856     1888  1 !
;  857     1889  1 ! ROUTINE VALUE:
;  858     1890  1 !       NONE
;  859     1891  1 !
;  860     1892  1 ! SIDE EFFECTS:
;  861     1893  1 !       NONE
;  862     1894  1 !
;  863     1895  1 !--
;  864     1896  1
;  865     1897  2 BEGIN
;  866     1898  2 MAP
;  867     1899  2         SMQ:            REF BBLOCK;     ! Pointer to SMQ
;  868     1900  2 LOCAL
;  869     1901  2         CONTINUE,                       ! True if loop to be continued
;  870     1902  2         SQH_MODIFIED,                   ! True if SQH modified
;  871     1903  2         PRED_MODIFIED,                  ! True if predecessor modified
;  872     1904  2         LIST_HEAD:      REF VECTOR,     ! Pointer to pending list head
;  873     1905  2         SQH:            REF BBLOCK,     ! Pointer to SQH
;  874     1906  2         SJH_NP,                         ! Record number of predecessor of SJH
;  875     1907  2         SJH_P:          REF BBLOCK,     ! Pointer to predecessor of SJH
;  876     1908  2         SJH_N,                          ! Record number of SJH
;  877     1909  2         SJH_NS,                         ! Record number of successor of SJH
;  878     1910  2         SJH:            REF BBLOCK,     ! Pointer to SJH
;  879     1911  2         QSMQ_N,                         ! Record number of job's SMQ
;  880     1912  2         QSMQ:           REF BBLOCK;     ! Pointer to job's SMQ
;  881     1913  2
;  882     1914  2
;  883     1915  2 ! If the queue is not available, fail immediately.
;  884     1916  2 !
;  885     1917  2 IF .SMQ[SMQ$V_PAUSED]
;  886     1918  2 OR .SMQ[SMQ$V_PAUSING]
;  887     1919  2 OR .SMQ[SMQ$V_RESUMING]
;  888     1920  2 OR .SMQ[SMQ$V_STARTING]
;  889     1921  2 OR .SMQ[SMQ$V_STOPPED]
;  890     1922  2 OR .SMQ[SMQ$V_UNAVAILABLE]
;  891     1923  2 OR .SMQ[SMQ$B_CURRENT_JOB_COUNT] GEQU .SMQ[SMQ$B_JOB_LIMIT]
```

```
;  892          1924  2 THEN
;  893          1925  2     RETURN;
;  894          1926  2
;  895          1927  2
;  896          1928  2 PRED_MODIFIED = FALSE;
;  897          1929  2 SQH_MODIFIED = FALSE;
;  898          1930  2 CONTINUE = TRUE;
;  899          1931  2 SQH = READ_RECORD(SJH_NP = SQH$K_RECNO);
;  900          1932  2 IF .SMQ[SMQ$V_BATCH]
;  901          1933  2     THEN LIST_HEAD = SQH[SQH$L_PENDING_BATCH_LIST]
;  902          1934  2     ELSE LIST_HEAD = SQH[SQH$L_PENDING_PRINT_LIST];
;  903          1935  2 SJH_N = .LIST_HEAD[0];
;  904          1936  2
;  905          1937  2
;  906          1938  2 ! Note that invoking START_EXECUTOR_JOB and FIND_AVAILABLE_EXECUTOR (which calls
;  907          1939  2 ! START_EXECUTOR_JOB) can result in the job being completed and its job record
;  908          1940  2 ! being put on the free list instead of the current list for the executor queue.
;  909          1941  2 ! This happens if the $CREPRC system service fails (for example when no PCB
;  910          1942  2 ! slots are available). When this happens the job record already has been
;  911          1943  2 ! released and it should not be rewritten.
;  912          1944  2 !
;  913          1945  2 WHILE .SJH_N NEQ 0 AND .CONTINUE DO
;  914          1946  3     BEGIN
;  915          1947  3     SJH = READ_RECORD(.SJH_N);
;  916          1948  3     IF .SJH[SJH$L_QUEUE_LINK] EQL 0 THEN EXITLOOP;
;  917          1949  3     SJH_NS = .SJH[SYM$L_LINK];
;  918          1950  3     IF
;  919          1951  4         BEGIN
;  920          1952  4         IF .SMQ[SMQ$V_GENERIC_QUEUE] OR .SMQ[SMQ$L_ASSIGNED_QUEUE_LINK] NEQ 0
;  921          1953  4         THEN
;  922          1954  5             BEGIN
;  923          1955  5             IF .SJH[SJH$L_QUEUE_LINK] EQL .SMQ_N
;  924          1956  5             THEN
;  925          1957  5                 IF FIND_AVAILABLE_EXECUTOR(.SMQ_N, .SMQ, .SJH_N, .SJH)
;  926          1958  5                 THEN
;  927          1959  6                     BEGIN
;  928          1960  6                     SMQ[SMQ$W_PENDING_JOB_COUNT] = .SMQ[SMQ$W_PENDING_JOB_COUNT] - 1;
;  929          1961  6                     TRUE
;  930          1962  6                     END
;  931          1963  5                 ELSE
;  932          1964  5                     FALSE
;  933          1965  5             ELSE
;  934          1966  5                 FALSE
;  935          1967  5             END
;  936          1968  4         ELSE
;  937          1969  5             BEGIN
;  938          1970  5             QSMQ = .SMQ;
;  939          1971  5             QSMQ_N = .SJH[SJH$L_QUEUE_LINK];
;  940          1972  5             IF .QSMQ_N NEQ .SMQ_N THEN QSMQ = READ_RECORD(.QSMQ_N);
;  941          1973  5             IF EXECUTOR_ACCEPTS_JOB(.QSMQ_N, .QSMQ, .SMQ_N, .SMQ, .SJH_N, .SJH)
;  942          1974  5             THEN
;  943          1975  6                 BEGIN
;  944          1976  6                 QSMQ[SMQ$W_PENDING_JOB_COUNT] = .QSMQ[SMQ$W_PENDING_JOB_COUNT] - 1;
;  945          1977  6                 IF .QSMQ_N NEQ .SMQ_N THEN REWRITE_RECORD(.QSMQ_N);
;  946          1978  6                 START_EXECUTOR_JOB(.SMQ_N, .SMQ, .SJH_N, .SJH);
;  947          1979  6                 IF .SMQ[SMQ$B_CURRENT_JOB_COUNT] GEQU .SMQ[SMQ$B_JOB_LIMIT]
;  948          1980  6                     THEN CONTINUE = FALSE;
```

```
;  949          1981  6                  TRUE
;  950          1982  6                  END
;  951          1983  5              ELSE
;  952          1984  6                  BEGIN
;  953          1985  6                  IF .QSMQ_N NEQ .SMQ_N THEN RELEASE_RECORD(.QSMQ_N);
;  954          1986  6                  FALSE
;  955          1987  6                  END
;  956          1988  5              END
;  957          1989  4          END
;  958          1990  3      THEN
;  959          1991  4          BEGIN
;  960          1992  4          IF .SJH_NP EQL SQH$K_RECNO
;  961          1993  4          THEN
;  962          1994  5              BEGIN
;  963          1995  5              LIST_HEAD[0] = .SJH_NS;
;  964          1996  5              IF .SJH_NS EQL 0 THEN LIST_HEAD[1] = 0;
;  965          1997  5              SQH_MODIFIED = TRUE;
;  966          1998  5              END
;  967          1999  4          ELSE
;  968          2000  5              BEGIN
;  969          2001  5              SJH_P[SYM$L_LINK] = .SJH_NS;
;  970          2002  5              IF .SJH_NS EQL 0
;  971          2003  5              THEN
;  972          2004  6                  BEGIN
;  973          2005  6                  LIST_HEAD[1] = .SJH_NP;
;  974          2006  6                  SQH_MODIFIED = TRUE;
;  975          2007  5                  END;
;  976          2008  5              PRED_MODIFIED = TRUE;
;  977          2009  4              END;
;  978          2010  4
;  979          2011  4          ! Guard against the possibility of having the job record already
;  980          2012  4          ! released and put on the queue header free list instead of it being
;  981          2013  4          ! placed at the end of the current list for the executor queue.
;  982          2014  4          !
;  983          2015  4          IF NOT .FLAGS[FLAGS_V_INVALID_SJH] THEN REWRITE_RECORD(.SJH_N);
;  984          2016  4          END
;  985          2017  3      ELSE
;  986          2018  4          BEGIN
;  987          2019  4          IF .SJH_NP NEQ SQH$K_RECNO
;  988          2020  4          THEN
;  989          2021  4              IF TESTBITSC(PRED_MODIFIED)
;  990          2022  4                  THEN REWRITE_RECORD(.SJH_NP)
;  991          2023  4                  ELSE RELEASE_RECORD(.SJH_NP);
;  992          2024  4          SJH_NP = .SJH_N;
;  993          2025  4          SJH_P = .SJH;
;  994          2026  3          END;
;  995          2027  3      SJH_N = .SJH_NS;
;  996          2028  2      END;
;  997          2029  2
;  998          2030  2
;  999          2031  2 IF .SJH_NP NEQ SQH$K_RECNO
; 1000          2032  2 THEN
; 1001          2033  2     IF .PRED_MODIFIED
; 1002          2034  2         THEN REWRITE_RECORD(.SJH_NP)
; 1003          2035  2         ELSE RELEASE_RECORD(.SJH_NP);
; 1004          2036  2
; 1005          2037  2
```

```
; 1006          2038  2 IF .SQH_MODIFIED
; 1007          2039  2     THEN REWRITE_RECORD(SQH$K_RECNO)
; 1008          2040  2     ELSE RELEASE_RECORD(SQH$K_RECNO);
; 1009          2041  1 END;
; INFO#250                L1:2001
; Referenced LOCAL symbol SJH_P is probably not initialized
```

```
                                        0FFC 00000          .ENTRY  FIND_PENDING_JOBS, Save R2,R3,R4,R5,R6,R7,- ; 1867
                                                                    R8,R9,R10,R11
                                5E          10  C2 00002          SUBL2   #16, SP
                                52      08  AC  D0 00005          MOVL    SMQ, R2                                 ; 1917
                                50      10  A2  9E 00009          MOVAB   16(R2), R0
                    01          60          02  E1 0000D          BBC     #2, (R0), 1$
                                                04 00011          RET
                    01          60          03  E1 00012 1$:      BBC     #3, (R0), 2$                            ; 1918
                                                04 00016          RET
                    01          60          06  E1 00017 2$:      BBC     #6, (R0), 3$                            ; 1919
                                                04 0001B          RET
                                01      01  A0  E9 0001C 3$:      BLBC    1(R0), 4$                               ; 1920
                                                04 00020          RET
                    01          60          09  E1 00021 4$:      BBC     #9, (R0), 5$                            ; 1921
                                                04 00025          RET
                    01          60          0B  E1 00026 5$:      BBC     #11, (R0), 6$                           ; 1922
                                                04 0002A          RET
            0116    C2      0115    C2  91  0002B 6$:      CMPB    277(R2), 278(R2)                        ; 1923
                                            01  1F 00032          BLSSU   7$
                                                04 00034          RET
                                            6E  7C 00035 7$:      CLRQ    PRED_MODIFIED                           ; 1928
                        08  AE              01  D0 00037          MOVL    #1, CONTINUE                            ; 1930
                                55          01  D0 0003B          MOVL    #1, SJH_NP                              ; 1931
                                            01  DD 0003E          PUSHL   #1
                00000000G   EF              01  FB 00040          CALLS   #1, READ_RECORD
                                06      0C  A2  E9 00047          BLBC    12(R2), 8$                              ; 1932
                                54      54  A0  9E 0004B          MOVAB   84(R0), LIST_HEAD                       ; 1933
                                            04  11 0004F          BRB     9$
                                54      5C  A0  9E 00051 8$:      MOVAB   92(R0), LIST_HEAD                       ; 1934
                                57          64  D0 00055 9$:      MOVL    (LIST_HEAD), SJH_N                      ; 1935
                                58      04  AC  D0 00058          MOVL    SMQ_N, R8                               ; 1955
                                            57  D5 0005C 10$:     TSTL    SJH_N                                   ; 1945
                                            04  13 0005E          BEQL    11$
                                03      08  AE  E8 00060          BLBS    CONTINUE, 12$
                                        0106 31 00064 11$:      BRW     28$
                                            57  DD 00067 12$:     PUSHL   SJH_N                                   ; 1947
                00000000G   EF              01  FB 00069          CALLS   #1, READ_RECORD
                                56          50  D0 00070          MOVL    R0, SJH
                                50      0134 C6 D0 00073          MOVL    308(SJH), R0                            ; 1948
                                            EA  13 00078          BEQL    11$
                                5A          66  D0 0007A          MOVL    (SJH), SJH_NS                           ; 1949
        05      0D  A2              02  E0 0007D          BBS     #2, 13(R2), 13$                         ; 1952
                                    2C  A2  D5 00082          TSTL    44(R2)
                                            1B  13 00085          BEQL    14$
                                58          50  D1 00087 13$:     CMPL    R0, R8                                  ; 1955
                                            7E  12 0008A          BNEQ    18$
```

```
                                56  DD 0008C          PUSHL   SJH                                     ; 1957
                          0084  8F  BB 0008E          PUSHR   #^M<R2,R7>
                                58  DD 00092          PUSHL   R8
                        FD5F    CF  04  FB 00094          CALLS   #4, FIND_AVAILABLE_EXECUTOR
                                6E  50  E9 00099          BLBC    R0, 18$
                          0102  C2  B7 0009C          DECW    258(R2)                                 ; 1960
                                6A  11 000A0          BRB     19$
                                53  52  D0 000A2 14$:    MOVL    R2, QSMQ                            ; 1970
                                59  50  D0 000A5          MOVL    R0, QSMQ_N                          ; 1971
                                5B  D4 000A8          CLRL    R11                                     ; 1972
                                58  59  D1 000AA          CMPL    QSMQ_N, R8
                                0E  13 000AD          BEQL    15$
                                5B  D6 000AF          INCL    R11
                                59  DD 000B1          PUSHL   QSMQ_N
                    00000000G   EF  01  FB 000B3          CALLS   #1, READ_RECORD
                                53  50  D0 000BA          MOVL    R0, QSMQ
                                56  DD 000BD 15$:    PUSHL   SJH                                     ; 1973
                          0084  8F  BB 000BF          PUSHR   #^M<R2,R7>
                          0108  8F  BB 000C3          PUSHR   #^M<R3,R8>
                                59  DD 000C7          PUSHL   QSMQ_N
                        FBD4    CF  06  FB 000C9          CALLS   #6, EXECUTOR_ACCEPTS_JOB
                                2D  50  E9 000CE          BLBC    R0, 17$
                          0102  C3  B7 000D1          DECW    258(QSMQ)                               ; 1976
                                09  5B  E9 000D5          BLBC    R11, 16$                            ; 1977
                                59  DD 000D8          PUSHL   QSMQ_N
                    00000000G   EF  01  FB 000DA          CALLS   #1, REWRITE_RECORD
                                56  DD 000E1 16$:    PUSHL   SJH                                     ; 1978
                          0084  8F  BB 000E3          PUSHR   #^M<R2,R7>
                                58  DD 000E7          PUSHL   R8
                    00000000G   EF  04  FB 000E9          CALLS   #4, START_EXECUTOR_JOB
                        0116    C2  0115  C2 91 000F0          CMPB    277(R2), 278(R2)                ; 1979
                                    13 1F 000F7          BLSSU   19$
                                08  AE  D4 000F9          CLRL    CONTINUE                            ; 1980
                                0E  11 000FC          BRB     19$                                     ; 1979
                                42  5B  E9 000FE 17$:    BLBC    R11, 24$                            ; 1985
                                59  DD 00101          PUSHL   QSMQ_N
                    00000000G   EF  01  FB 00103          CALLS   #1, RELEASE_RECORD
                                37  11 0010A 18$:    BRB     24$
                                01  55  D1 0010C 19$:    CMPL    SJH_NP, #1                          ; 1992
                                0E  12 0010F          BNEQ    21$
                                64  5A  D0 00111          MOVL    SJH_NS, (LIST_HEAD)                 ; 1995
                                03  12 00114          BNEQ    20$                                     ; 1996
                                04  A4  D4 00116          CLRL    4(LIST_HEAD)
                          04    AE  01  D0 00119 20$:    MOVL    #1, SQH_MODIFIED                    ; 1997
                                11  11 0011D          BRB     23$                                     ; 1992
                          0C    BE  5A  D0 0011F 21$:    MOVL    SJH_NS, @SJH_P                      ; 2001
                                08  12 00123          BNEQ    22$                                     ; 2002
                          04    A4  55  D0 00125          MOVL    SJH_NP, 4(LIST_HEAD)                ; 2005
                          04    AE  01  D0 00129          MOVL    #1, SQH_MODIFIED                    ; 2006
                                6E  01  D0 0012D 22$:    MOVL    #1, PRED_MODIFIED                   ; 2008
                2F  00000000'   EF  05  E0 00130 23$:    BBS     #5, FLAGS, 27$                      ; 2015
                                57  DD 00138          PUSHL   SJH_N
                    00000000G   EF  01  FB 0013A          CALLS   #1, REWRITE_RECORD
                                24  11 00141          BRB     27$                                     ; 1950
                                01  55  D1 00143 24$:    CMPL    SJH_NP, #1                          ; 2019
                                18  13 00146          BEQL    26$
                0B              6E  00  E5 00148          BBCC    #0, PRED_MODIFIED, 25$              ; 2021
```

```
                                    55  DD 0014C          PUSHL   SJH_NP                                  ; 2022
                00000000G   EF      01  FB 0014E          CALLS   #1, REWRITE_RECORD                      ;
                                    09  11 00155          BRB     26$                                     ;
                                    55  DD 00157 25$:     PUSHL   SJH_NP                                  ; 2023
                00000000G   EF      01  FB 00159          CALLS   #1, RELEASE_RECORD                      ;
                            55      57  D0 00160 26$:     MOVL    SJH_N, SJH_NP                           ; 2024
                      0C    AE      56  D0 00163          MOVL    SJH, SJH_P                              ; 2025
                            57      5A  D0 00167 27$:     MOVL    SJH_NS, SJH_N                           ; 2027
                                  FEEF  31 0016A          BRW     10$                                     ; 1945
                            01      55  D1 0016D 28$:     CMPL    SJH_NP, #1                              ; 2031
                                    17  13 00170          BEQL    30$                                     ;
                            0B      6E  E9 00172          BLBC    PRED_MODIFIED, 29$                      ; 2033
                                    55  DD 00175          PUSHL   SJH_NP                                  ; 2034
                00000000G   EF      01  FB 00177          CALLS   #1, REWRITE_RECORD                      ;
                                    09  11 0017E          BRB     30$                                     ;
                                    55  DD 00180 29$:     PUSHL   SJH_NP                                  ; 2035
                00000000G   EF      01  FB 00182          CALLS   #1, RELEASE_RECORD                      ;
                            0A  04  AE  E9 00189 30$:     BLBC    SQH_MODIFIED, 31$                       ; 2038
                                    01  DD 0018D          PUSHL   #1                                      ; 2039
                00000000G   EF      01  FB 0018F          CALLS   #1, REWRITE_RECORD                      ;
                                        04 00196          RET                                             ;
                                    01  DD 00197 31$:     PUSHL   #1                                      ; 2040
                00000000G   EF      01  FB 00199          CALLS   #1, RELEASE_RECORD                      ;
                                        04 001A0          RET                                             ; 2041

; Routine Size:  417 bytes,    Routine Base:  CODE + 055D
```

```
;  1011     2042  1 GLOBAL ROUTINE JOB_SCHEDULING_POLICY(SMQ,SJH_1,SJH_2)=
;  1012     2043  1
;  1013     2044  1 !++
;  1014     2045  1 !
;  1015     2046  1 ! FUNCTIONAL DESCRIPTION:
;  1016     2047  1 !       This routine implements the default job scheduling policy.
;  1017     2048  1 !
;  1018     2049  1 ! INPUT PARAMETERS:
;  1019     2050  1 !       SMQ             - Pointer to SMQ.
;  1020     2051  1 !       SJH_1           - Pointer to first SJH.
;  1021     2052  1 !       SJH_2           - Pointer to second SJH.
;  1022     2053  1 !
;  1023     2054  1 ! IMPLICIT INPUTS:
;  1024     2055  1 !       NONE
;  1025     2056  1 !
;  1026     2057  1 ! OUTPUT PARAMETERS:
;  1027     2058  1 !       NONE
;  1028     2059  1 !
;  1029     2060  1 ! IMPLICIT OUTPUTS:
;  1030     2061  1 !       NONE
;  1031     2062  1 !
;  1032     2063  1 ! ROUTINE VALUE:
;  1033     2064  1 !       True if SJH_1 should execute before SJH_2; false otherwise.
;  1034     2065  1 !
;  1035     2066  1 ! SIDE EFFECTS:
;  1036     2067  1 !       NONE
;  1037     2068  1 !
;  1038     2069  1 !--
;  1039     2070  1
;  1040     2071  2 BEGIN
;  1041     2072  2 MAP
;  1042     2073  2         SMQ:            REF BBLOCK,     ! Pointer to SMQ
;  1043     2074  2         SJH_1:          REF BBLOCK,     ! Pointer to SJH
;  1044     2075  2         SJH_2:          REF BBLOCK;     ! Pointer to SJH
;  1045     2076  2
;  1046     2077  2
;  1047     2078  2 ! In this implementation the batch queue is ordered by increasing submission
;  1048     2079  2 ! time within decreasing priority; the print queue is ordered by increasing
;  1049     2080  2 ! submission time within decreasing file size within decreasing priority; but
;  1050     2081  2 ! if /SCHEDULE=NOSIZE is specified, file size is not considered.
;  1051     2082  2 !
;  1052     2083  2 IF .SJH_1[SJH$B_PRIORITY] GTRU .SJH_2[SJH$B_PRIORITY]
;  1053     2084  2 THEN
;  1054     2085  2     TRUE
;  1055     2086  2 ELSE IF .SJH_1[SJH$B_PRIORITY] EQL .SJH_2[SJH$B_PRIORITY]
;  1056     2087  2 THEN
;  1057     2088  2     IF .SMQ[SMQ$V_BATCH] OR NOT .SMQ[SMQ$V_JOB_SIZE_SCHEDULING]
;  1058     2089  2     THEN
;  1059     2090  3         IF TIME_GTRU(SJH_2[SJH$Q_TIME], SJH_1[SJH$Q_TIME])
;  1060     2091  2         THEN
;  1061     2092  2             TRUE
;  1062     2093  2         ELSE
;  1063     2094  2             FALSE
;  1064     2095  2     ELSE
;  1065     2096  2         IF .SJH_1[SJH$L_JOB_SIZE] LSSU .SJH_2[SJH$L_JOB_SIZE]
;  1066     2097  2         THEN
;  1067     2098  2             TRUE
```

```
;  1068          2099  2          ELSE IF .SJH_1[SJH$L_JOB_SIZE] EQLU .SJH_2[SJH$L_JOB_SIZE]
;  1069          2100  2          THEN
;  1070          2101  3              IF TIME_GTRU(SJH_2[SJH$Q_TIME], SJH_1[SJH$Q_TIME])
;  1071          2102  2              THEN
;  1072          2103  2                  TRUE
;  1073          2104  2              ELSE
;  1074          2105  2                  FALSE
;  1075          2106  2          ELSE
;  1076          2107  2              FALSE
;  1077          2108  2 ELSE
;  1078          2109  2     FALSE
;  1079          2110  1 END;


                                                    0004 00000          .ENTRY  JOB_SCHEDULING_POLICY, Save R2                  ; 2042
                                          52    08  AC  D0 00002          MOVL    SJH_1, R2                                       ; 2083
                                          51    0C  AC  D0 00006          MOVL    SJH_2, R1                                       ;
                                    017D  C1  017D  C2  91 0000A          CMPB    381(R2), 381(R1)                                ;
                                                    2E  1A 00011          BGTRU   2$                                              ;
                                                    30  12 00013          BNEQ    3$                                              ; 2086
                                          50    04  AC  D0 00015          MOVL    SMQ, R0                                         ; 2088
                                          10    0C  A0  E8 00019          BLBS    12(R0), 1$                                      ;
                                OB        0D  A0      06  E1 0001D          BBC     #6, 13(R0), 1$                                  ;
                                    0100  C1  0100  C2  D1 00022          CMPL    256(R2), 256(R1)                                ; 2096
                                                    16  1F 00029          BLSSU   2$                                              ;
                                                    18  12 0002B          BNEQ    3$                                              ; 2099
                                    0140  C2  0140  C1  D1 0002D 1$:      CMPL    320(R1), 320(R2)                                ; 2101
                                                    0B  1A 00034          BGTRU   2$                                              ;
                                                    0D  12 00036          BNEQ    3$                                              ;
                                    013C  C2  013C  C1  D1 00038          CMPL    316(R1), 316(R2)                                ;
                                                    04  1B 0003F          BLEQU   3$                                              ;
                                          50            01  D0 00041 2$:      MOVL    #1, R0                                      ;
                                                            04 00044          RET                                                 ;
                                                    50  D4 00045 3$:      CLRL    R0                                              ; 2086
                                                        04 00047          RET                                                     ; 2110

; Routine Size:  72 bytes,    Routine Base:  CODE + 06FE
```

```
;  1081     2111  1 GLOBAL ROUTINE ENQUEUE_JOB(SJH_N,SJH; RESULT,ESMQ): L_OUTPUT_2 NOVALUE=
;  1082     21 2  1
;  1083     2113  1 !++
;  1084     2114  1 !
;  1085     2115  1 ! FUNCTIONAL DESCRIPTION:
;  1086     2116  1 !       This routine enqueues a newly available job on an appropriate queue.
;  1087     2117  1 !
;  1088     2118  1 ! INPUT PARAMETERS:
;  1089     2119  1 !       SJH_N           - Record number of SJH.
;  1090     2120  1 !       SJH             - Pointer to SJH.
;  1091     2121  1 !
;  1092     2122  1 ! IMPLICIT INPUTS:
;  1093     2123  1 !       NONE
;  1094     2124  1 !
;  1095     2125  1 ! OUTPUT PARAMETERS:
;  1096     2126  1 !       RESULT          - Result of the enqueue.
;  1097     2127  1 !       ESMQ            - Pointer to executor SMQ, if job is executing.
;  1098     2128  1 !
;  1099     2129  1 ! IMPLICIT OUTPUTS:
;  1100     2130  1 !       NONE
;  1101     2131  1 !
;  1102     2132  1 ! ROUTINE VALUE:
;  1103     2133  1 !       NONE
;  1104     2134  1 !
;  1105     2135  1 ! SIDE EFFECTS:
;  1106     2136  1 !       NONE
;  1107     2137  1 !
;  1108     2138  1 !--
;  1109     2139  1
;  1110     2140  2 BEGIN
;  1111     2141  2 MAP
;  1112     2142  2         SJH:            REF BBLOCK;     ! Pointer to SJH
;  1113     2143  2 LOCAL
;  1114     2144  2         SQH:            REF BBLOCK,     ! Pointer to SQH
;  1115     2145  2         SMQ_N,                          ! Record number of SMQ
;  1116     2146  2         SMQ:            REF BBLOCK,     ! Pointer to SMQ
;  1117     2147  2         SJH_NP,                         ! Record number of predecessor SJH
;  1118     2148  2         SJH_P:          REF BBLOCK,     ! Pointer to predecessor SJH
;  1119     2149  2         SJH_NS,                         ! Record number of successor SJH
;  1120     2150  2         SJH_S:          REF BBLOCK,     ! Pointer to successor SJH
;  1121     2151  2         STATUS;                         ! Status return
;  1122     2152  2
;  1123     2153  2
;  1124     2154  2 ! If this SJH is invalid (can only occur due to file corruption) then
;  1125     2155  2 ! return a spurious COMPLETE as result, thereby evaporating the enqueue
;  1126     2156  2 ! request.  Note that file is still corrupt.
;  1127     2157  2 !
;  1128     2158  2 IF .SJH[SJH$L_QUEUE_LINK] EQL 0
;  1129     2159  2 THEN
;  1130     2160  3     BEGIN
;  1131     2161  3     RESULT = ENQ_K_COMPLETE;
;  1132     2162  3     RETURN;
;  1133     2163  2     END;
;  1134     2164  2
;  1135     2165  2 ! If there are services outstanding for this job, cancel them.
;  1136     2166  2 !
;  1137     2167  2 IF .SJH[SJH$V_ABORTING]
```

```
;  1138          2168  2 OR .SJH[SJH$V_STARTING]
;  1139          2169  2 THEN
;  1140          2170  2     SCAN_INCOMPLETE_SERVICES(ISRV_K_PURGE_SJH, .SJH_N);
;  1141          2171  2
;  1142          2172  2
;  1143          2173  2 ! If there is a pending requeue for this job, execute it.
;  1144          2174  2 !
;  1145          2175  2 IF .SJH[SJH$V_REQUEUE]
;  1146          2176  2 THEN
;  1147          2177  3     BEGIN
;  1148          2178  3     IF .SJH[SJH$V_REQUEUE_HOLD] THEN SJH[SJH$V_HOLDING] = TRUE;
;  1149          2179  3     SJH[SJH$B_PRIORITY] = .SJH[SJH$B_REQUEUE_PRIORITY];
;  1150          2180  3     IF .SJH[SJH$L_REQUEUE_QUEUE_LINK] NEQ 0
;  1151          2181  3         THEN SJH[SJH$L_QUEUE_LINK] = .SJH[SJH$L_REQUEUE_QUEUE_LINK];
;  1152          2182  2     END;
;  1153          2183  2
;  1154          2184  2
;  1155          2185  2 ! Finish initializing the completed job.
;  1156          2186  2 !
;  1157          2187  2 IF .SJH[SJH$V_EXECUTING] THEN SJH[SJH$V_RESTARTING] = TRUE;
;  1158          2188  2 SJH[SYM$L_LINK] = 0;
;  1159          2189  2 SJH[SJH$V_ABORTED] = FALSE;
;  1160          2190  2 SJH[SJH$V_ABORTING] = FALSE;
;  1161          2191  2 SJH[SJH$V_EXECUTING] = FALSE;
;  1162          2192  2 SJH[SJH$V_FILE_STARTING] = FALSE;
;  1163          2193  2 SJH[SJH$V_OPEN] = FALSE;
;  1164          2194  2 SJH[SJH$V_REQUEUE] = FALSE;
;  1165          2195  2 SJH[SJH$V_REQUEUE_HOLD] = FALSE;
;  1166          2196  2 SJH[SJH$V_STARTING] = FALSE;
;  1167          2197  2 SJH[SJH$V_SYSTEM_FAILURE] = FALSE;
;  1168          2198  2 SJH[SJH$B_REQUEUE_PRIORITY] = 0;
;  1169          2199  2 SJH[SJH$L_REQUEUE_QUEUE_LINK] = 0;
;  1170          2200  2 SJH[SJH$L_EXECUTOR_PID] = 0;
;  1171          2201  2
;  1172          2202  2
;  1173          2203  2 ! Read the SMQ record.
;  1174          2204  2 !
;  1175          2205  2 SMQ = READ_RECORD(SMQ_N = .SJH[SJH$L_QUEUE_LINK]);
;  1176          2206  2
;  1177          2207  2
;  1178          2208  2 IF .SJH[SJH$V_HOLDING]
;  1179          2209  2 OR .SJH[SJH$V_REFUSED]
;  1180          2210  2 OR .SJH[SJH$V_RETAINED]
;  1181          2211  2 THEN
;  1182          2212  3     BEGIN
;  1183          2213  3
;  1184          2214  3     ! Job specified with /HOLD, was retained after execution, or was refused.
;  1185          2215  3     ! Enqueue to the hold queue for the specified queue.
;  1186          2216  3     !
;  1187          2217  3     IF .SMQ[SMQ$L_HOLD_LIST] EQL 0
;  1188          2218  3     THEN
;  1189          2219  3         SMQ[SMQ$L_HOLD_LIST] = .SJH_N
;  1190          2220  3     ELSE
;  1191          2221  4         BEGIN
;  1192          2222  4         SJH_P = READ_RECORD(SJH_NP = .SMQ[SMQ$L_HOLD_LIST_END]);
;  1193          2223  4         SJH_P[SYM$L_LINK] = .SJH_N;
;  1194          2224  4         REWRITE_RECORD(.SJH_NP);
```

```
;1195    2225  3          END;
;1196    2226  3      SMQ[SMQ$L_HOLD_LIST_END] = .SJH_N;
;1197    2227  3      REWRITE_RECORD(.SMQ_N);
;1198    2228  3      RESULT = ENQ_K_HOLD;
;1199    2229  3      END
;1200    2230  3
;1201    2231  3
;1202    2232  3  ELSE IF TIME_GTRU(SJH[SJH$Q_AFTER_TIME], CUR_TIME)
;1203    2233  2  THEN
;1204    2234  3      BEGIN
;1205    2235  3
;1206    2236  3      ! Job specified with /AFTER.  Enqueue to the timer queue.
;1207    2237  3      !
;1208    2238  3      SQH = READ_RECORD(SJH_NP = SQH$K_RECNO);
;1209    2239  3      SMQ[SMQ$W_TIMER_JOB_COUNT] = .SMQ[SMQ$W_TIMER_JOB_COUNT] + 1;
;1210    2240  3
;1211    2241  3
;1212    2242  3      ! Check for the special case that the job goes at the end.
;1213    2243  3      !
;1214    2244  3      IF .SQH[SQH$L_TIMER_LIST_END] NEQ 0
;1215    2245  3      THEN
;1216    2246  4          BEGIN
;1217    2247  4          SJH_S = READ_RECORD(SJH_NS = .SQH[SQH$L_TIMER_LIST_END]);
;1218    2248  5          IF TIME_GEQU(SJH[SJH$Q_AFTER_TIME], SJH_S[SJH$Q_AFTER_TIME])
;1219    2249  4          THEN
;1220    2250  5              BEGIN
;1221    2251  5              SJH_S[SYM$L_LINK] = .SJH_N;
;1222    2252  5              SQH[SQH$L_TIMER_LIST_END] = .SJH_N;
;1223    2253  5              REWRITE_RECORD(.SJH_NS);
;1224    2254  5              REWRITE_RECORD(.SMQ_N);
;1225    2255  5              REWRITE_RECORD(SQH$K_RECNO);
;1226    2256  5              RESULT = ENQ_K_TIMER;
;1227    2257  5              RETURN;
;1228    2258  4              END;
;1229    2259  3          END;
;1230    2260  3
;1231    2261  3
;1232    2262  3      ! Search down the timer list looking for the insertion point.
;1233    2263  3      !
;1234    2264  3      SJH_NS = .SQH[SQH$L_TIMER_LIST];
;1235    2265  3      WHILE .SJH_NS NEQ 0 DO
;1236    2266  4          BEGIN
;1237    2267  4          SJH_S = READ_RECORD(.SJH_NS);
;1238    2268  5          IF TIME_GTRU(SJH_S[SJH$Q_AFTER_TIME], SJH[SJH$Q_AFTER_TIME])
;1239    2269  4          THEN
;1240    2270  5              BEGIN
;1241    2271  5              RELEASE_RECORD(.SJH_NS);
;1242    2272  5              EXITLOOP;
;1243    2273  4              END;
;1244    2274  4          IF .SJH_NP NEQ SQH$K_RECNO THEN RELEASE_RECORD(.SJH_NP);
;1245    2275  4          SJH_NP = .SJH_NS;
;1246    2276  4          SJH_P = .SJH_S;
;1247    2277  4          SJH_NS = .SJH_S[SYM$L_LINK];
;1248    2278  3          END;
;1249    2279  3
;1250    2280  3
;1251    2281  3      ! Enqueue the job.
```

```
; 1252          2282  3      !
; 1253          2283  3      IF .SJH_NP EQL SQH$K_RECNO
; 1254          2284  3      THEN
; 1255          2285  4          BEGIN
; 1256          2286  4          SQH[SQH$L_TIMER_L'   ] = .SJH_N;
; 1257          2287  4          $CANTIM(REQIDT=JBL   _AFTER_IDT);
; 1258        P 2288  4          STATUS = $SETIMR(
; 1259        P 2289  4              DAYTIM=SJH[SJH$Q_AFTER_TIME],
; 1260        P 2290  4              ASTADR=AFTER_AST,
; 1261          2291  4              REQIDT=JBC$K_AFTER_IDT);
; 1262          2292  4          IF NOT .STATUS
; 1263          2293  4          THEN
; 1264          2294  4              SIGNAL(JBC$_SETIMR OR STS$K_ERROR, 0, .STATUS);
; 1265          2295  4          END
; 1266          2296  3      ELSE
; 1267          2297  4          BEGIN
; 1268          2298  4          SJH_P[SYM$L_LINK] = .SJH_N;
; 1269          2299  4          REWRITE_RECORD(.SJH_NP);
; 1270          2300  3          END;
; 1271          2301  3
; 1272          2302  3
; 1273          2303  3      SJH[SYM$L_LINK] = .SJH_NS;
; 1274          2304  3      IF .SJH_NS EQL 0 THEN SQH[SQH$L_TIMER_LIST_END] = .SJH_N;
; 1275          2305  3      REWRITE_RECORD(.SMQ_N);
; 1276          2306  3      REWRITE_RECORD(SQH$K_RECNO);
; 1277          2307  3      RESULT = ENQ_K_TIMER;
; 1278          2308  3      END
; 1279          2309  3
; 1280          2310  3
; 1281          2311  2 ELSE
; 1282          2312  3      BEGIN
; 1283          2313  3      IF FIND_AVAILABLE_EXECUTOR(.SMQ_N, .SMQ, .SJH_N, .SJH; ESMQ)
; 1284          2314  3      THEN
; 1285          2315  4          BEGIN
; 1286          2316  4          REWRITE_RECORD(.SMQ_N);
; 1287          2317  4          RESULT = ENQ_K_CURRENT;
; 1288          2318  4          END
; 1289          2319  3      ELSE
; 1290          2320  4          BEGIN
; 1291          2321  4          LOCAL
; 1292          2322  4              LIST_HEAD:  REF VECTOR;      ! Pointer to pending list head
; 1293          2323  4
; 1294          2324  4
; 1295          2325  4          ! Job cannot be started now because no executor is available.  Enqueue
; 1296          2326  4          ! the job to the pending queue according to the established policy.
; 1297          2327  4          !
; 1298          2328  4          SMQ[SMQ$W_PENDING_JOB_COUNT] = .SMQ[SMQ$W_PENDING_JOB_COUNT] + 1;
; 1299          2329  4          SQH = READ_RECORD(SJH_NP = SQH$K_RECNO);
; 1300          2330  4          IF .SMQ[SMQ$V_BATCH]
; 1301          2331  4              THEN LIST_HEAD = SQH[SQH$L_PENDING_BATCH_LIST]
; 1302          2332  4              ELSE LIST_HEAD = SQH[SQH$L_PENDING_PRINT_LIST];
; 1303          2333  4
; 1304          2334  4
; 1305          2335  4          ! Check for the special case that the job goes at the end.
; 1306          2336  4          !
; 1307          233'  4          IF .LIST_HEAD[1] NEQ 0
; 1308          2338  4          THEN
```

```
;  1309          2339  5              BEGIN
;  1310          2340  5              SJH_S = READ_RECORD(SJH_NS = .LIST_HEAD[1]);
;  1311          2341  5              IF JOB_SCHEDULING_POLICY(.SMQ, .SJA_S, .SJH)
;  1312          2342  5              THEN
;  1313          2343  6                  BEGIN
;  1314          2344  6                  SJH_S[SYM$L_LINK] = .SJH_N;
;  1315          2345  6                  LIST_HEAD[1] = .SJH_N;
;  1316          2346  6                  REWRITE_RECORD(.SJH_NS);
;  1317          2347  6                  REWRITE_RECORD(.SMQ_N);
;  1318          2348  6                  REWRITE_RECORD(SQH$K_RECNO);
;  1319          2349  6                  RESULT = ENQ_K_PENDING;
;  1320          2350  6                  RETURN;
;  1321          2351  5                  END;
;  1322          2352  4              END;
;  1323          2353  4
;  1324          2354  4
;  1325          2355  4          ! Search down the pending list looking for the insertion point.
;  1326          2356  4          !
;  1327          2357  4          SJH_NS = .LIST_HEAD[0];
;  1328          2358  4          WHILE .SJH_NS NEQ 0 DO
;  1329          2359  5              BEGIN
;  1330          2360  5              SJH_S = READ_RECORD(.SJH_NS);
;  1331          2361  5              IF JOB_SCHEDULING_POLICY(.SMQ, .SJH, .SJH_S)
;  1332          2362  5              THEN
;  1333          2363  6                  BEGIN
;  1334          2364  6                  RELEASE_RECORD(.SJH_NS);
;  1335          2365  6                  EXITLOOP;
;  1336          2366  5                  END;
;  1337          2367  5              IF .SJH_NP NEQ SQH$K_RECNO THEN RELEASE_RECORD(.SJH_NP);
;  1338          2368  5              SJH_NP = .SJH_NS;
;  1339          2369  5              SJH_P = .SJH_S;
;  1340          2370  5              SJH_NS = .SJA_S[SYM$L_LINK];
;  1341          2371  4              END;
;  1342          2372  4
;  1343          2373  4
;  1344          2374  4          ! Enqueue the job.
;  1345          2375  4          !
;  1346          2376  4          IF .SJH_NP EQL SQH$K_RECNO
;  1347          2377  4          THEN
;  1348          2378  4              LIST_HEAD[0] = .SJH_N
;  1349          2379  4          ELSE
;  1350          2380  5              BEGIN
;  1351          2381  5              SJH_P[SYM$L_LINK] = .SJH_N;
;  1352          2382  5              REWRITE_RECORD(.SJH_NP);
;  1353          2383  4              END;
;  1354          2384  4
;  1355          2385  4
;  1356          2386  4          SJH[SYM$L_LINK] = .SJH_NS;
;  1357          2387  4          IF .SJH_NS EQL 0 THEN LIST_HEAD[1] = .SJH_N;
;  1358          2388  4          REWRITE_RECORD(.SMQ_N);
;  1359          2389  4          REWRITE_RECORD(SQH$K_RECNO);
;  1360          2390  4          RESULT = ENQ_K_PENDING;
;  1361          2391  3          END;
;  1362          2392  2      END;
;  1363          2393  1  END;
```

```
                                                        .EXTRN  SYS$CANTIM

                                                 03FC 00000     .ENTRY  ENQUEUE_JOB, Save R2,R3,R4,R5,R6,R7,R8,R9  ; 2111
                                             SE  08   C2 00002  SUBL2   #8, SP                                     ;
                                         08  54  AC   D0 00005  MOVL    SJH, R4                                    ; 2158
                                       0134  52  C4   9E 00009  MOVAB   308(R4), R2                               ;
                                             62  D5   0000E  TSTL    (R2)                                       ;
                                             04  12   00010  BNEQ    1$                                         ;
                                             5A  04   D0 00012  MOVL    #4, RESULT                                 ; 2161
                                                 04   00015  RET                                                  ; 2160
                                             53  10   A4 9E 00016 1$:  MOVAB   16(R4), R3                        ; 2167
                                         04  63      01   E0 0001A  BBS     #1, (R3), 2$                           ;
                                         0C  63      0C   E1 0001E  BBC     #12, (R3), 3$                          ; 2168
                                             04  AC   DD 00022 2$:  PUSHL   SJH_N                                  ; 2170
                                             05  DD   00025  PUSHL   #5                                           ;
                                 00000000G   EF      02   FB 00027  CALLS   #2, SCAN_INCOMPLETE_SERVICES          ;
                                             18  01   A3   E9 0002E 3$:  BLBC    1(R3), 5$                      ; 2175
                                         03  63      09   E1 00032  BBC     #9, (R3), 4$                           ; 2178
                                             63      20   88 00036  BISB2   #32, (R3)                             ;
                                       017D  C4  017E C4   90 00039 4$:  MOVB    382(R4), 381(R4)                   ; 2179
                                             50  0138 C4   D0 00040  MOVL    312(R4), R0                            ; 2180
                                                      03   13 00045  BEQL    5$                                     ;
                                             62       50   D0 00047  MOVL    R0, (R2)                               ; 2181
                                         04  63      03   E1 0004A 5$:  BBC     #3, (R3), 6$                           ; 2187
                                           01  A3      04   88 0004E  BISB2   #4, 1(R3)                            ;
                                                      64   D4 00052 6$:  CLRL    (R4)                               ; 2188
                                             63  535B  8F   AA 00054  BICW2   #21339, (R3)                        ; 2197
                                                 017E  C4   94 00059  CLRB    382(R4)                            ; 2198
                                                 0138  C4   D4 0005D  CLRL    312(R4)                            ; 2199
                                                 0168  C4   D4 00061  CLRL    360(R4)                            ; 2200
                                           04  AE          62   D0 00065  MOVL    (R2), SMQ_N                        ; 2205
                                                 04  AE   DD 00069  PUSHL   SMQ_N                                 ;
                                 00000000G   EF      01   FB 0006C  CALLS   #1, READ_RECORD                      ;
                                             52      50   D0 00073  MOVL    R0, SMQ                              ;
                                             59  04   AC   D0 00076  MOVL    SJH_N, R9                          ; 2219
                                         08  63      05   E0 0007A  BBS     #5, (R3), 7$                           ; 2208
                                             63      95   7E 0007E  TSTB    (R3)                                   ; 2209
                                                      04   19 00080  BLSS    7$                                     ;
                                         3A  63      0B   E1 00082  BBC     #11, (R3), 10$                         ; 2210
                                             78  A2   D5 00086 7$:  TSTL    120(SMQ)                               ; 2217
                                                      06   12 00089  BNEQ    8$                                     ;
                                           78  A2          59   D0 0008B  MOVL    R9, 120(SMQ)                       ; 2219
                                                      1D   11 0008F  BRB     9$                                     ;
                                             58  7C   A2   D0 00091 8$:  MOVL    124(SMQ), SJH_NP                   ; 2222
                                                      58   DD 00095  PUSHL   SJH_NP                                ;
                                 00000000G   EF      01   FB 00097  CALLS   #1, READ_RECORD                      ;
                                             6E      50   D0 0009E  MOVL    P, SJH_P                             ;
                                           00  BE          59   D0 000A1  MOVL    R9, @SJH_P                         ; 2223
                                                      58   DD 000A5  PUSHL   SJH_NP                                ; 2224
                                 00000000G   EF      01   FB 000A7  CALLS   #1, REWRITE_RECORD                    ;
                                           7C  A2          59   D0 000AE 9$:  MOVL    R9, 124(SMQ)                       ; 2226
                                             04  AE   DD 000B2  PUSHL   SMQ_N                                    ; 2227
                                 00000000G   EF      01   FB 000B5  CALLS   #1, REWRITE_RECORD                    ;
                                             5A      01   D0 000BC  MOVL    #1, RESULT                           ; 2228
                                                      04   00 0BF  RET                                           ; 2208
                                             57  009C   C4   D0 000C0 10$:  MOVL    156(R4), R7                        ; 2232
```

```
                00000000'  EF          57  D1 000C5          CMPL    R7, CUR_TIME+4                          ;
                                        10  1A 000CC          BGTRU   13$                                     ;
                                        03  13 000CE          BEQL    12$                                     ;
                                    010B  31 000D0 11$:       BRW     24$                                     ;
                00000000'  EF      0098  C4  D1 000D3 12$:    CMPL    152(R4), CUR_TIME                       ;
                                        F2  1B 000DC          BLEQU   11$                                     ;
                           58           01  D0 000DE 13$:    MOVL    #1, SJH_NP                              ; 2238
                                        01  DD 000E1          PUSHL   #1                                      ;
                00000000G  EF           01  FB 000E3          CALLS   #1, READ_RECORD                         ;
                           55           50  D0 000EA          MOVL    R0, SQH                                 ;
                                    010C  C2  B6 000ED          INCW    268(SMQ)                                ; 2239
                                      6C  A5  D5 000F1          TSTL    108(SQH)                                ; 2244
                                          37  13 000F4          BEQL    15$                                     ;
                           56           6C  A5  D0 000F6          MOVL    108(SQH), SJH_NS                        ; 2247
                                        56  DD 000FA          PUSHL   SJH_NS                                  ;
                00000000G  EF           01  FB 000FC          CALLS   #1, READ_RECORD                         ;
                           53           50  D0 00103          MOVL    R0, SJH_S                               ;
                               009C     C3  57  D1 00106          CMPL    R7, 156(SJH_S)                          ; 2248
                                        0B  1A 0010B          BGTRU   14$                                     ;
                                        1E  12 0010D          BNEQ    15$                                     ;
                               0098     C3  0098  C4  D1 0010F    CMPL    152(R4), 152(SJH_S)                     ;
                                        15  1F 00116          BLSSU   15$                                     ;
                           63           04  AC  D0 00118 14$:    MOVL    SJH_N, (SJH_S)                          ; 2251
                             6C     A5   04  AC  D0 0011C          MOVL    SJH_N, 108(SQH)                         ; 2252
                                        56  DD 00121          PUSHL   SJH_NS                                  ; 2253
                00000000G  EF           01  FB 00123          CALLS   #1, REWRITE_RECORD                      ;
                                    009A  31  0012A          BRW     23$                                     ; 2254
                           56           68  A5  D0 0012D 15$:    MOVL    104(SQH), SJH_NS                        ; 2264
                                        42  13 00131 16$:    BEQL    20$                                     ; 2265
                                        56  DD 00133          PUSHL   SJH_NS                                  ; 2267
                00000000G  EF           01  FB 00135          CALLS   #1, READ_RECORD                         ;
                           53           50  D0 0013C          MOVL    R0, SJH_S                               ;
                           57      009C C3  D1 0013F          CMPL    156(SJH_S), R7                          ; 2268
                                        0B  1A 00144          BGTRU   17$                                     ;
                                        14  12 00146          BNEQ    18$                                     ;
                               0098     C4  0098  C3  D1 00148    CMPL    152(SJH_S), 152(R4)                     ;
                                        0B  1B 0014F          BLEQU   18$                                     ;
                                        56  DD 00151 17$:    PUSHL   SJH_NS                                  ; 2271
                00000000G  EF           01  FB 00153          CALLS   #1, RELEASE_RECORD                      ;
                                        19  11 0015A          BRB     20$                                     ; 2270
                           01           58  D1 0015C 18$:    CMPL    SJH_NP, #1                              ; 2274
                                        09  13 0015F          BEQL    19$                                     ;
                                        58  DD 00161          PUSHL   SJH_NP                                  ;
                00000000G  EF           01  FB 00163          CALLS   #1, RELEASE_RECORD                      ;
                           58           56  D0 0016A 19$:    MOVL    SJH_NS, SJH_NP                          ; 2275
                           6E           53  D0 0016D          MOVL    SJH_S, SJH_P                            ; 2276
                           56           63  D0 00170          MOVL    (SJH_S), SJH_NS                         ; 2277
                                        BC  11 00173          BRB     16$                                     ; 2265
                           01           58  D1 00175 20$:    CMPL    SJH_NP, #1                              ; 2283
                                        37  12 00178          BNEQ    21$                                     ;
                             68     A5   59  D0 0017A          MOVL    R9, 104(SQH)                            ; 2286
                                    7E   01  7D 0017E          MOVQ    #1, -(SP)                               ; 2287
                00000000G  00           02  FB 00181          CALLS   #2, SYS$CANTIM                          ;
                                        01  DD 00188          PUSHL   #1                                      ; 2291
                                    F80A CF  9F 0018A          PUSHAB  AFTER_AST                               ;
                                    0098 C4  9F 0018E          PUSHAB  152(R4)                                 ;
                                        7E  D4 00192          CLRL    -(SP)                                   ;
```

```
                00000000G  00              04  FB 00194          CALLS   #4, SYS$SETIMR                          ;
                           20              50  E8 00019B         BLBS    STATUS, 22$                            ; 2292
                                           50  DD 0019E          PUSHL   STATUS                                 ; 2294
                                           7E  D4 001A0          CLRL    -(SP)                                  ;
                               0004845A    8F  DD 001A2          PUSHL   #296026                                ;
                00000000G  00              03  FB 001A8          CALLS   #3, LIB$SIGNAL                         ;
                                           0D  11 001AF          BRB     22$                                    ; 2283
                      00   BE              59  D0 001B1 21$:     MOVL    R9, @SJH_P                             ; 2298
                                           58  DD 001B5          PUSHL   SJH_NP                                 ; 2299
                00000000G  EF              01  FB 001B7          CALLS   #1, REWRITE_RECORD                     ;
                           64              56  D0 001BE 22$:     MOVL    SJH_NS, (R4)                           ; 2303
                                           04  12 001C1          BNEQ    23$                                    ; 2304
                      6C   A5              59  D0 001C3          MOVL    R9, 108(SQH)                           ;
                                C4     AE   DD 001C7 23$:     PUSHL   SMQ_N                                  ; 2305
                00000000G  EF              01  FB 001CA          CALLS   #1, REWRITE_RECORD                     ;
                                           01  DD 001D1          PUSHL   #1                                     ; 2306
                00000000G  EF              01  FB 001D3          CALLS   #1, REWRITE_RECORD                     ;
                           5A              03  D0 001DA          MOVL    #3, RESULT                             ; 2307
                                           04  001DD             RET                                            ; 2232
                                           54  DD 001DE 24$:     PUSHL   R4                                     ; 2313
                                     0204  8F  BB 001E0          PUSHR   #^M<R2,R9>                             ;
                                       10  AE  DD 001E4          PUSHL   SMQ_N                                  ;
                          FA23     CF        04  FB 001E7          CALLS   #4, FIND_AVAILABLE_EXECUTOR            ;
                                   0D        50  E9 001EC          BLBC    R0, 25$                                ;
                                       04  AE  DD 001EF          PUSHL   SMQ_N                                  ; 2316
                00000000G  EF              01  FB 001F2          CALLS   #1, REWRITE_RECORD                     ;
                                           5A  D4 001F9          CLRL    RESULT                                 ; 2317
                                           04  001FB             RET                                            ; 2313
                                     0102  C2  B6 001FC 25$:     INCW    258(SMQ)                               ; 2328
                                   58          01  D0 00200          MOVL    #1, SJH_NP                             ; 2329
                                               01  DD 00203          PUSHL   #1                                     ;
                00000000G  EF                  01  FB 00205          CALLS   #1, READ_RECORD                        ;
                           55                  50  D0 0020C          MOVL    R0, SQH                                ;
                           06          0C      A2  E9 0020F          BLBC    12(SMQ), 26$                           ; 2330
                           57          54      A5  9E 00213          MOVAB   84(R5), LIST_HEAD                      ; 2331
                                               04  11 00217          BRB     27$                                    ;
                           57          5C      A5  9E 00219 26$:     MOVAB   92(R5), LIST_HEAD                      ; 2332
                                       04      A7  D5 0021D 27$:     TSTL    4(LIST_HEAD)                           ; 2337
                                               2C  13 00220          BEQL    28$                                    ;
                           56          04      A7  D0 00222          MOVL    4(LIST_HEAD), SJH_NS                   ; 2340
                                               56  DD 00226          PUSHL   SJH_NS                                 ;
                00000000G  EF                  01  FB 00228          CALLS   #1, READ_RECORD                        ;
                           53                  50  D0 0022F          MOVL    R0, SJH_S                              ;
                                               1C  BB 00232          PUSHR   #^M<R2,R3,R4>                          ; 2341
                          FD7F     CF        03  FB 00234          CALLS   #3, JOB_SCHEDULING_POLICY              ;
                                   12        50  E9 00239          BLBC    R0, 28$                                ;
                                   63        59  D0 0023C          MOVL    R9, (SJH_S)                            ; 2344
                                       04  A7  59  D0 0023F          MOVL    R9, 4(LIST_HEAD)                       ; 2345
                                               56  DD 00243          PUSHL   SJH_NS                                 ; 2346
                00000000G  EF                  01  FB 00245          CALLS   #1, REWRITE_RECORD                     ;
                                               61  11 0024C          BRB     35$                                    ; 2347
                           56                  67  D0 0024E 28$:     MOVL    (LIST_HEAD), SJH_NS                    ; 2357
                                               3C  13 00251 29$:     BEQL    32$                                    ; 2358
                                               56  DD 00253          PUSHL   SJH_NS                                 ; 2360
                00000000G  EF                  01  FB 00255          CALLS   #1, READ_RECORD                        ;
                           53                  50  D0 0025C          MOVL    R0, SJH_S                              ;
                                               53  DD 0025F          PUSHL   SJH_S                                  ; 2361
```

```
                                 14  BB 00261          PUSHR   #^M<R2,R4>                                ;
                        FD50     CF         03  FB 00263          CALLS   #3, JOB_SCHEDULING_POLICY           ;
                                 0B         50  E9 00268          BLBC    R0, 30$                             ;
                                            56  DD 0026B          PUSHL   SJH_NS                              ; 2364
                    00000000G    EF         01  FB 0026D          CALLS   #1, RELEASE_RECORD                  ;
                                            19  11 00274          BRB     32$                                 ; 2363
                                 01         58  D1 00276 30$:     CMPL    SJH_NP, #1                          ; 2367
                                            09  13 00279          BEQL    31$                                 ;
                                            58  DD 0027B          PUSHL   SJH_NP                              ;
                    00000000G    EF         01  FB 0027D          CALLS   #1, RELEASE_RECORD                  ;
                                 58         56  D0 00284 31$:     MOVL    SJH_NS, SJH_NP                      ; 2368
                                 6E         53  D0 00287          MOVL    SJH_S, SJH_P                        ; 2369
                                 56         63  D0 0028A          MOVL    (SJH_S), SJH_NS                     ; 2370
                                            C2  11 0028D          BRB     29$                                 ; 2358
                                 01         58  D1 0028F 32$:     CMPL    SJH_NP, #1                          ; 2376
                                            05  12 00292          BNEQ    33$                                 ;
                                 67         59  D0 00294          MOVL    R9, (LIST_HEAD)                     ; 2378
                                            0D  11 00297          BRB     34$                                 ;
                        00       BE         59  D0 00299 33$:     MOVL    R9, @SJH_P                          ; 2381
                                            58  DD 0029D          PUSHL   SJH_NP                              ; 2382
                    00000000G    EF         01  FB 0029F          CALLS   #1, REWRITE_RECORD                  ;
                                 64         56  D0 002A6 34$:     MOVL    SJH_NS, (R4)                        ; 2386
                                            04  12 002A9          BNEQ    35$                                 ; 2387
                        04       A7         59  D0 002AB          MOVL    R9, 4(LIST_HEAD)                    ;
                                      04    AE  DD 002AF 35$:     PUSHL   SMQ_N                               ; 2388
                    00000000G    EF         01  FB 002B2          CALLS   #1, REWRITE_RECORD                  ;
                                            01  DD 002B9          PUSHL   #1                                  ; 2389
                    00000000G    EF         01  FB 002BB          CALLS   #1, REWRITE_RECORD                  ;
                                 5A         02  D0 002C2          MOVL    #2, RESULT                          ; 2390
                                            04  002C5          RET                                         ; 2393

; Routine Size:  710 bytes,    Routine Base:  CODE + 0746
```

```
; 1365          2394  1 END
; 1366          2395  0 ELUDOM


                                                        .EXTRN  LIB$SIGNAL

;                               PSECT SUMMARY
;
;       Name                    Bytes                   Attributes
;
;  COMMON                         5024  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                           2572  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                             Library Statistics
;
;                                      -------- Symbols --------      Pages       Processing
;       File                           Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1        18619        40         0      1000          00:01.4


; Information:  2
; Warnings:     0
; Errors:       0


;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:SCHEDULER/OBJ=OBJ$:SCHEDULER MSRC$:SCHEDULER/UPDATE=(ENH$:SCHEDULER)

; Size:         2572 code + 5024 data bytes
; Run Time:       00:42.1
; Elapsed Time:   02:36.2
; Lines/CPU Min:   3414
; Lexemes/CPU-Min: 32333
; Memory Used:  396 pages
; Compilation Complete
```

RESTRICT
LIS

RESPONSE
LIS

SND:BC
LIS

SCHEDULER
LIS