```
 0001 MUP11 DOMUS MACRO ASSEMBLER REV 02.00
01                ;                                          RCSL:    43-GL10269
02                ;                                          AUTHOR:  PG/MLM/JHA
03                ;                                          EDITED:  80.05.13
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21                ;                  MUP11
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39                ; KEYWORD:         MUS, PAGING SYSTEM, LISTING.
40                ; ABSTRACT:        MUS PAGING SYSTEM.
41                ;                  ASCII PAPER TAPE: RCSL 43-GL10268.
42                ;                  REL. BINARY PAPER TAPE: RCSL 43-GL10270.
43
44
```

```
10002 MUP11
01              ; PG 76-04-07   PAGING SYSTEM                          PAGE..02..
02
03
04              ; DESCRIPTION OF ENTRY POINTS:
05
06              ; PROCEDURE COMON(SUBROUTINE CALL);
07              ; EXECUTES THE SUBROUTINE CALL AND ARRANGES FOR A
08              ; PROPER RETURN. MAY CHANGE THE PAGE MAP.
09              ;       CALL          RETURN          LINK
10              ; AC0    *)            *)             +0: SUBROUTINE CALL
11              ; AC1    *)            *)             +1: RETURN
12              ; AC2    *)            *)
13              ; AC3   LINK           *)             *): AS FOR THE SUBROUTINE CALL
14
15              ; PROCEDURE CALL(POINT);
16              ; EXECUTES A SUBROUTINE JUMP TO THE POINT.
17              ; MAY CHANGE THE PAGE MAP.
18              ;       CALL          CONTINUATION    LINK
19              ; AC0                 UNCHANGED       +0: POINT
20              ; AC1                 UNCHANGED       +1: POSSIBLE RETURN
21              ; AC2                 UNCHANGED
22              ; AC3   LINK          LINK+1
23
24              ; PROCEDURE GOTO(POINT);
25              ; EXECUTES A JUMP TO THE POINT.
26              ; MAY CHANGE THE PAGE MAP.
27              ;       CALL          CONTINUATION    LINK
28              ; AC0                 UNCHANGED       +0: POINT
29              ; AC1                 UNCHANGED
30              ; AC2                 UNCHANGED
31              ; AC3   LINK          DESTROYED
32
33              ; PROCEDURE GETADR(POINT, ADDRESS);
34              ; COMPUTES THE ADDRESS OF THE POINT GIVEN.
35              ; MAY CHANGE THE PAGE MAP
36              ;       CALL          RETURN          LINK
37              ; AC0   POINT         UNCHANGED       +0: RETURN
38              ; AC1                 UNCHANGED
39              ; AC2                 UNCHANGED
40              ; AC3   LINK          ADDRESS
41
42              ; PROCEDURE GETPOINT(ADDRESS, POINT);
43              ; COMPUTES THE POINT CORRESPONDING TO THE ADDRESS GIVEN.
44              ;       CALL          RETURN          LINK
45              ; AC0   ADDRESS       UNCHANGED       +0: RETURN
46              ; AC1                 UNCHANGED
47              ; AC2                 UNCHANGED
48              ; AC3   LINK          POINT
49
50
```

```
10003 MUP11
01                ; PG 76-07-26   PAGING SYSTEM                    PAGE..03..
02
03
                  .TITLE MUP11
05          000012 .RDX  10
06          000001 .TXTM 1
07
08
09                ; TABLE OF PAGE ZERO ENTRY POINTS
10
11          000354 .LOC  COMON-GOS            ; PROCEDURE COMON
12  00354   000000'       PS0                  ;
13
14          000355 .LOC  CALL-GOS             ; PROCEDURE CALL
15  00355   000044'       PS1                  ;
16
17          000356 .LOC  GOTO-GOS             ; PROCEDURE GOTO
18  00356   000056'       PS2                  ;
19
20          000357 .LOC  GETADR-GOS           ; PROCEDURE GETADR
21  00357   000066'       PS3
22
23          000360 .LOC  GETPOINT-GOS         ; PROCEDURE GETPOINT
24  00360   000074'       PS4
25
26
27                .NREL
28
29
```

```
10004 MUP11
01              ; PG 76-06-17   PAGING SYSTEM                          PAGE..04..
02
03
04              ; RELATIVE DISPLACEMENTS IN  PROGRAMS
05
06        000007 PSIZE   = NAME+3            ; PAGE SIZE
07        000010 PMASK   = PSIZE+1           ; PAGE MASK
08        000011 PBLK    = PMASK+1           ; BLOCKING FACTOR
09        000012 PTAB    = PBLK+1            ; ADR. PAGETABLE/ADR. NO OF PAGES
10        000013 PMAP    = PTAB+1            ; ADR. PAGEMAP/ADR. SEMAPHORE
11        000014 PSTAT   = PMAP+1            ; ADR. PAGE STATISTICS PROCEDURE
12        000015 PFOF    = PSTAT+1           ; FIRST OF FRAMES
13        000016 PTOF    = PFOF+1            ; TOP OF FRAMES
14        000017 PVIC    = PTOF+1            ; VICTIM
15        000020 PREAD   = PVIC+1            ; NO OF PAGES READ
16        000021 PWRIT   = PREAD+1           ; NO OF PAGES WRITTEN
17        000022 PAGEIN  = PWRIT+1           ; PAGE TO BE LOADED INTO CORE
18        000023 PAGEOUT= PAGEIN+1           ; PAGE TO BE SAVED ONTO DISK
19        000024 PINADR  = PAGEOUT+1         ; ADR. INPUT MESSAGE
20        000025 PINMES  = PINADR+1          ; INPUT MESSAGE
21        000031 POUTADR= PINMES+4           ; ADR. OUTPUT MESSAGE
22        000032 POUTMES= POUTADR+1          ; OUTPUT MESSAGE
23        000036 PFLAG   = POUTMES+4         ; PAGER FLAG(NOT USED!)
24
25        000037 PAC0    = PFLAG+1           ; SAVED AC0
26        000040 PAC1    = PAC0+1            ; SAVED AC1
27        000041 PAC2    = PAC1+1            ; SAVED AC2
28        000042 PAC3    = PAC2+1            ; SAVED AC3: CURRENT ADDRESS
29        000043 PREF    = PAC3+1            ; VIRTUAL REFERENCE
30        000044 PLINK   = PREF+1            ; LINK IN PAGEFAULT ROUTINE
31
32        000045 PWRK1   = PLINK+1           ; WORKING LOCATIONS
33        000046 PWRK2   = PWRK1+1           ;
34        000047 PWRK3   = PWRK2+1           ;
35        000050 PWRK4   = PWRK3+1           ;
36        000051 PDRIVE  = PWRK4+1           ; ADR. NAME OF DISC DRIVER
37        000052 PDISP   = PDRIVE+1          ; KIT DISPLACEMENT
38
39
40              ; RELATIVE DISPLACEMENTS IN COROUTINES
41
42        000005 PCAC0   = CAC1SAVE+1        ; SAVED AC0
43        000006 PCAC1   = PCAC0+1           ; SAVED AC1
44        000007 PCAC2   = PCAC1+1           ; SAVED AC2
45        000010 PCAC3   = PCAC2+1           ; SAVED AC3: CURRENT POINT
46        000011 PCREF   = PCAC3+1           ; VIRTUAL REFERENCE
47        000012 PCLINK  = PCREF+1           ; LINK IN PAGEFAULT ROUTINE
48
49
```

```
10005 MUP11
01                ; PG 76-04-07   PAGING SYSTEM                        PAGE..05..
02
03
04                ; EXTERNAL PROCEDURE COMON(SUBROUTINE CALL);
05
06 00000'060277 PS0:  INTDS                    ;   DISABLE;
07 00001'040564       STA    0   PSAC0         ;   SAVE(AC0);
08 00002'044564       STA    1   PSAC1         ;   SAVE(AC1);
09 00003'050564       STA    2   PSAC2         ;   SAVE(AC2);
10 00004'021400       LDA    0   0,3           ;   AC0:= SUBROUTINE CALL;
11 00005'040417       STA    0   PS01          ;   STORE(SUBROUTINE CALL);
12 00006'161400       INC    3,0              ;   AC0:=RETURNADR:= CALLADR+1;
13 00007'126400       SUB    1,1              ;
14 00010'101112       MOVL#  0,0 SZC          ;   IF CALL FROM HIGH CORE THEN
15 00011'126520       SUBZL  1,1              ;   PCREF:=1
16 00012'030017       LDA    2   COROUT       ;   ELSE PCREF:=0;
17 00013'045011       STA    1   PCREF,2      ;
18 00014'004517       JSR        PS7          ;   AC3:= COMPUTEPOINT(RETURNADR);
19 00015'161000       MOV    3,0              ;   AC0:= RETURNPOINT;
20 00016'034017       LDA    3   COROUT       ;   AC3:= CURRENT COROUTINE;
21 00017'041412       STA    0   PCLINK,3     ;   CCOROUTINE.PCLINK:= RETURNPOINT;
22 00020'020545       LDA    0   PSAC0         ;   RESTORE(AC0);
23 00021'024545       LDA    1   PSAC1         ;   RESTORE(AC1);
24 00022'030545       LDA    2   PSAC2         ;   RESTORE(AC2);
25 00023'060177       INTEN                   ;   ENABLE AND
26 00024'000000 PS01: 0                       ;   EXECUTE(SUBROUTINE CALL);
27                                            ; RETURN FROM SUBROUTINE:
28 00025'060277       INTDS                   ;   DISABLE;
29 00026'040537       STA    0   PSAC0         ;   SAVE(AC0);
30 00027'044537       STA    1   PSAC1         ;   SAVE(AC1);
31 00030'034017       LDA    3   COROUT       ;
32 00031'054537       STA    3   PSAC3         ;   SAVE(AC3= COROUT);
33 00032'021412       LDA    0   PCLINK,3     ;
34 00033'025411       LDA    1   PCREF,3      ;
35 00034'115000       MOV    0,3              ;   IF PCREF = 0 THEN
36 00035'125005       MOV    1,1 SNR          ;     RETURNADDR:=
37 00036'004451       JSR        PS6          ;     TAKEPOINT(CCOROUTINE.PCLINK);
38 00037'054526       STA    3   PSAC0         ;   SAVE(RETURNADR);
39 00040'024526       LDA    1   PSAC1         ;   RESTORE(AC1);
40 00041'034527       LDA    3   PSAC3         ;   RESTORE(AC3);
41 00042'060177       INTEN                   ;   ENABLE AND
42 00043'002522       JMP@       PSAC0        ;   RETURN;
43
44
```

```
01                ; PG 75-11-27   PAGING SYSTEM                          PAGE..06..
02
03
04                ; EXTERNAL PROCEDURE CALL(POINT);
05
06 00044'060277 PS1:  INTDS                    ;   DISABLE;
07 00045'040520       STA     0   PSAC0       ;   SAVE(AC0);
08 00046'054522       STA     3   PSAC3       ;   SAVE(AC3);
09 00047'021400       LDA     0   0,3         ;   AC0:=POINT:= PARAMETER;
10 00050'004437       JSR         PS6         ;   AC3:= TAKEPOINT(POINT);
11 00051'054514       STA     3   PSAC0       ;   SAVE(CONTINUEADR);
12 00052'034516       LDA     3   PSAC3       ;   RESTORE(AC3);
13 00053'175400       INC     3,3             ;   AC3:=RETURNADR:= CALLADR+1;
14 00054'060177       INTEN                   ;   ENABLE AND
15 00055'002510       JMP@        PSAC0       ;   EXIT TO CONTINUE ADDRESS;
16
17
18                ; EXTERNAL PROCEDURE GOTO(POINT);
19
20 00056'060277 PS2:  INTDS                    ;   DISABLE;
21 00057'040506       STA     0   PSAC0       ;   SAVE(AC0);
22 00060'021400       LDA     0   0,3         ;   AC0:=POINT:= PARAMETER;
23 00061'176400       SUB     3,3             ;   AC3:= 0; I.E. NO RETURNADR;
24 00062'054506       STA     3   PSAC3       ;   SAVE(AC3);
25 00063'004424       JSR         PS6         ;   AC3:= TAKEPOINT(POINT);
26 00064'060177       INTEN                   ;   ENABLE AND
27 00065'001400       JMP         0,3         ;   EXIT TO CONTINUE ADDRESS;
28
29
30                ; EXTERNAL PROCEDURE GETADR(POINT);
31
32 00066'060277 PS3:  INTDS                    ;   DISABLE;
33 00067'040476       STA     0   PSAC0       ;   SAVE(AC0);
34 00070'054500       STA     3   PSAC3       ;   SAVE(AC3);
35 00071'004416       JSR         PS6         ;   AC3:= TAKEPOINT(POINT);
36 00072'060177       INTEN                   ;   ENABLE AND
37 00073'002475       JMP@        PSAC3       ;   RETURN;
38
39
40                ; EXTERNAL PROCEDURE GETPOINT(ADR);
41
42 00074'060277 PS4:  INTDS                    ;   DISABLE;
43 00075'040470       STA     0   PSAC0       ;   SAVE(AC0);
44 00076'044470       STA     1   PSAC1       ;   SAVE(AC1);
45 00077'050470       STA     2   PSAC2       ;   SAVE(AC2);
46 00100'054470       STA     3   PSAC3       ;   SAVE(AC3);
47 00101'004432       JSR         PS7         ;   AC3:= COMPUTEPOINT(ADR);
48 00102'020463       LDA     0   PSAC0       ;   RESTORE(AC0);
49 00103'024463       LDA     1   PSAC1       ;   RESTORE(AC1);
50 00104'030463       LDA     2   PSAC2       ;   RESTORE(AC2);
51 00105'060177       INTEN                   ;   ENABLE AND
52 00106'002462       JMP@        PSAC3       ;   RETURN;
53
54
```

```
10007 MUP11
01
02                ; PG 75-11-27   PAGING SYSTEM                          PAGE..07..
03
04
05                ; PROCEDURE TAKEPOINT(POINT);
06                ; COMPUTES THE ADDRESS OF A POINT
07                ;       CALL          RETURN          LINK
08                ; AC0   POINT         PSAC0           +0: RETURN
09                ; AC1                 UNCHANGED
10                ; AC2                 UNCHANGED
11                ; AC3   LINK          ADDRESS
12
13 00107'054463 PS6:  STA     3   PSLINK     ;   SAVE(LINK);
14 00110'115103       MOVL    0,3 SNC         ;   IF POINTREF(0:0) = 0
15 00111'000416       JMP         PS60        ;   THEN GOTO POINT TAKEN;
16 00112'103240       ADDOR   0,0             ;   AC0:=VIRTUALREF:= POINTREF-1B0;
17 00113'040456       STA     0   PSREF       ;   SAVE(VIRTUALREF);
18 00114'050453       STA     2   PSAC2       ;   SAVE(AC2);
19                PFRETURN:                   ; RETURN FROM PAGEFAULT:
20 00115'030040       LDA     2   CUR         ;   AC2:=PROCESS:= CUR;
21 00116'031012       LDA     2   PROG,2      ;   AC2:=PROGRAM:= PROCESS.PROG;
22 00117'004430       JSR         PS8         ;   AC0:= COMPUTEPAGE(VITUALREF);
23 00120'035013       LDA     3   PMAP,2      ;
24 00121'117000       ADD     0,3             ;
25 00122'035400       LDA     3   0,3         ;   AC3:= PMAP(PAGE);
26 00123'030446       LDA     2   PSREF       ;   AC2:= VIRTUALREF;
27                                            ;   AC3:=BYTEADR:=
28 00124'157102       ADDL    2,3 SZC         ;     2*(VIRTUALREF+PMAP(PAGE));
29 00125'002405       JMP@        .PAGEFAULT; IF BYTEADR>2**16 THEN PAGEFAULT;
30 00126'030441       LDA     2   PSAC2       ;   RESTORE(AC2);
31                PS60:                       ; POINT TAKEN:
32 00127'175220       MOVZR   3,3             ;   AC3:=ABSADR:= BYTEADR/2;
33 00130'020435       LDA     0   PSAC0       ;   RESTORE ORIGINAL(AC0);
34 00131'002441       JMP@        PSLINK      ;   RETURN;
35
36
37 00132'000173'.PAGEFAULT: PAGEFAULT
```

```
10008 MUP11
01                ; PG 75-11-27   PAGING SYSTEM                          PAGE..08..
02
03
04                ; PROCEDURE COMPUTEPOINT(ADR);
05                ; COMPUTES A POINT FROM AN ADDRESS
06                ;       CALL          RETURN          LINK
07                ; AC0   ADR           DESTROYED       +0: RETURN
08                ; AC1                 DESTROYED
09                ; AC2                 PROGRAM
10                ; AC3   LINK          POINT
11
12 00133'054000 PS7:  STA     3   0          ;   SAVE(LINK);
13 00134'030040       LDA     2   CUR        ;   AC2:=PROCESS:= CUR;
14 00135'031012       LDA     2   PROG,2     ;   AC2:=PROGRAM:= PROCESS.PROG;
15 00136'006464       JSR@        COUPPER    ;   COMPUTE FRAME AND REL;
16 00137'000406       JMP         PS70       ;   IF ADR IN FRAMES THEN
17 00140'035400       LDA     3   0,3        ;   BEGIN AC3:= FIRST WORD ON FRAME;
18 00141'025010       LDA     1   PMASK,2     ;     AC1:= PROGRAM.MASK;
19 00142'137400       AND     1,3            ;     AC3:=POINT:=
20 00143'117000       ADD     0,3            ;       PAGEBASE+REL+1B0
21 00144'177241       ADDOR   3,3 SKP        ;   END
22 00145'115000 PS70: MOV     0,3            ;   ELSE AC3:=POINT:= ADR;
23 00146'002000       JMP@        0          ;   RETURN;
24
25
26                ; PROCEDURE COMPUTEPAGE(VIRTUAL ADDRESS);
27                ; COMPUTES THE PAGENUMBER FROM A VIRTUAL ADDRESS
28                ;       CALL          RETURN          LINK
29                ; AC0   VIRTUALADR    PAGENO          +0: RETURN
30                ; AC1                 UNCHANGED
31                ; AC2   PROGRAM       PROGRAM
32                ; AC3   LINK          DESTROYED
33
34 00147'055045 PS8:  STA     3   PWRK1,2    ;   SAVE(AC3);
35 00150'035010       LDA     3   PMASK,2    ;   AC3:=MASK:=PROGRAM.MASK;
36 00151'163705       ANDS    3,0 SNR        ;   AC0:= PAGE*BLOCKING FACTOR;
37 00152'002412       JMP@        .ERR       ;   IF PAGE = 0 THEN GOTO ERROR;
38 00153'035011       LDA     3   PBLK,2     ;   BLFAC:= PROGRAM.BLFAC;
39 00154'175222       MOVZR   3,3 SZC        ;   FOR C:=BLFAC MOD 2,BLFAC:=BLFAC/2
40 00155'000403       JMP         .+3        ;   WHILE C=0 DO
41 00156'101220       MOVZR   0,0            ;     AC0:=AC0/2
42 00157'000775       JMP         .-3        ;   NOW AC0=PAGE;
43 00160'037012       LDA@    3   PTAB,2     ;   AC3:=LASTPAGE:=PROGRAM.PTAB(0);
44 00161'116432       SUBZ#   0,3 SZC        ;   IF PAGENO <= LASTPAGE
45 00162'003045       JMP@        PWRK1,2    ;     THEN RETURN;
46 00163'002401       JMP@        .ERR       ;   GOTO ERROR;
47 00164'000174*.ERR: ERR1                   ;   STEPPING STONE;
48
49
50                .NREL
51
52                ; WORKING VARIABLES USED IN DISABLED MODE:
53 00165'000000 PSAC0:        0   ;   SAVED AC0
54 00166'000000 PSAC1:        0   ;   SAVED AC1
55 00167'000000 PSAC2:        0   ;   SAVED AC2
56 00170'000000 PSAC3:        0   ;   SAVED AC3/MODIFIED AC3
57 00171'000000 PSREF:        0   ;   VIRTUAL REFERENCE
58 00172'000000 PSLINK:       0   ;   RETURN ADR FROM PAGEFAULT/COMON
59
60
```

```
01                ; PG 75-11-27   PAGING SYSTEM                          PAGE..09..
02
03
04                ; PROCEDURE PAGEFAULT
05                ; TAKES CARE OF PAGEFAULTS BY BRINGING PAGES INTO CORE
06                ; AND MODIFY VARIABLES SO THAT THE CALL IS TRANSPARENT
07                ; TO THE PROGRAMS
08                ;       CALL          RETURN        RETURN TO PFRETURN
09                ; AC0                 PSREF
10                ; AC1                 UNCHANGED
11                ; AC2                 DESTROYED
12                ; AC3                 DESTROYED
13                ; PSAC0 SAVED AC0     UNCHANGED
14                ; PSAC2 SAVED AC2     UNCHANGED
15                ; PSAC3 CURADR        CURADR/MODIFIED CURADR
16                ; PSREF VIRTUALREF    UNCHANGED
17                ; PSLINK RETURNADR     UNCHANGED
18
19
20
21
22                PAGEFAULT:                      ; PAGEFAULT:
23 00173'044773         STA     1    PSAC1      ;   SAVE(AC1);
24 00174'030017         LDA     2    COROUT     ;
25 00175'151015         MOV#    2,2  SNR        ;   IF COROUT<>0 THEN
26 00176'000407         JMP          PS101      ;   BEGIN
27 00177'020422         LDA     0    COLOWER    ;     TEST FOR COSPEC CALL:
28 00200'024422         LDA     1    COUPPER    ;
29 00201'034767         LDA     3    PSAC3      ;
30 00202'162033         ADCZ#   3,0  SNC        ;     IF CALL ADDRESS < COLOWER OR
31 00203'166033         ADCZ#   3,1  SNC        ;        CALL ADDRESS >= COUPPER
32 00204'000422         JMP          COSPEC     ;     THEN GOTO COROUTINE SPECIAL;
33                PS101:                          ;   END COROUTINE;
34 00205'006416         JSR@          .PMOVE    ;   MOVEVARIABLES(PAGER,PROGRAM);
35 00206'000350"        PSAC0-1*2               ;
36 00207'000171         PAC0-1*4+1              ;
37 00210'060177         INTEN                   ;   ENABLE;
38 00211'002414         JMP@        .PSCHECK    ;   GOTO CHECK AND ADVANCE VICTIM;
39
40                EXITPAGEFAULT:                  ; EXITPAGEFAULT:
41 00212'060277         INTDS                   ;   DISABLE;
42 00213'006410         JSR@          .PMOVE    ;   MOVEVARIABLES(PROGRAM,PAGER);
43 00214'000171         PAC0-1*4+1              ;
44 00215'000350"        PSAC0-1*2               ;
45                PS102:                          ; EXIT FROM COROUTINESPECIAL:
46 00216'020753         LDA     0    PSREF      ;   AC0:= PSREF;
47 00217'024747         LDA     1    PSAC1      ;   RESTORE(AC1);
48 00220'002404         JMP@         .PFRETURN  ;   RETURN FROM PAGEFAULT;
49 00221'000226'COLOWER:        COSPEC
50 00222'000000*COUPPER:        PS100
51 00223'000032*.PMOVE:         PMOVE
52 00224'000115'.PFRETURN:      PFRETURN
53 00225'000062*.PSCHECK:       PSCHECK
```

```
!0010 MUP11
01                ; PG 75-11-27   PAGING SYSTEM                          PAGE..10..
02
03
04                COSPEC:                          ; COROUTINE SPECIAL:
05  00226'020742          LDA     0   PSAC3        ;   AC0:=CURADR:= PSAC3;
06  00227'034743          LDA     3   PSLINK       ;
07  00230'177240          ADDOR   3,3              ;
08  00231'101112          MOVL#   0,0 SZC          ;   IF PSAC3 = (ADDR IN HIGH CORE)
09  00232'054740          STA     3   PSLINK       ;   THEN PLINK:=PLINK+1B0;
10  00233'006444          JSR@        .PS7         ;   PSAC3:=CURPOINT:=
11  00234'054734          STA     3   PSAC3        ;     COMPUTEPOINT(CURADR);
12  00235'006766          JSR@         .PMOVE      ;   MOVEVARIABLES(PAGER,COROUTINE);
13  00236'000350"         PSAC0-1*2                ;
14  00237'000023          PCAC0-1*4+3              ;
15  00240'060177          INTEN                    ;   ENABLE;
16  00241'030040          LDA     2   CUR          ;   AC2:=PROCESS:= CUR;
17  00242'031012          LDA     2   PROG,2       ;   AC2:=PROGRAM:= PROCESS.PROG;
18  00243'025013          LDA     1   PMAP,2       ;   AC1:=SEMAPHORE:=PMAP(0);
19  00244'006335          WAITSEMAPHORE            ;   WAITSEMAPHORE(PAGER READY);
20  00245'021411          LDA     0   PCREF,3      ;   IF VICTIM = REFPAGE
21  00246'006432          JSR@        .PS105       ;   THEN ADVANCE VICTIM;
22  00247'025012          LDA     1   PCLINK,2     ;   IF PLINK AND 1B0 <> 0 THEN
23  00250'125123          MOVZL   1,1 SNC          ;   !CALL FROM HIGH CORE!
24  00251'000404          JMP         COSP1        ;   PLINK:=PLINK-1B0
25  00252'125220          MOVZR   1,1              ;   ELSE
26  00253'045012          STA     1   PCLINK,2     ;   BEGIN
27  00254'000407          JMP         COSP2        ;
28  00255'021010 COSP1:   LDA     0   PCAC3,2      ;
29  00256'103240          ADDOR   0,0              ;   IF VICTIM = CURPAGE
30  00257'006421          JSR@        .PS105       ;   THEN ADVANCE VICTIM;
31  00260'103240          ADDOR   0,0              ;
32  00261'006357          GETADR                   ;   CCOROUT.CURADR:=
33  00262'055010          STA     3   PCAC3,2      ;     GETADR(CURPOINT);
34                                                 ;   END;
35  00263'021011 COSP2:   LDA     0   PCREF,2      ;
36  00264'103240          ADDOR   0,0              ;
37  00265'006357          GETADR                   ;   GETADR(REFPOINT);
38  00266'030040          LDA     2   CUR          ;   AC2:=PROCESS:= CUR;
39  00267'031012          LDA     2   PROG,2       ;   AC2:=PROGRAM:=PROCESS.PROG;
40  00270'025013          LDA     1   PMAP,2       ;   AC1:= PROGRAM.SEMAPHORE;
41  00271'006343          SIGNAL                   ;   SIGNAL(PROGRAM.SEMAPHORE);
42  00272'060277          INTDS                    ;   DISABLE;
43  00273'006730          JSR@        .PMOVE       ;   MOVE VARIABLES(COROUTINE, PAGER);
44  00274'000023          PCAC0-1*4+3              ;
45  00275'000350"         PSAC0-1*2                ;
46  00276'000720          JMP         PS102        ;   GOTO EXIT FROM COROUTINE SPECIAL;
47
48
49  00277'000133'.PS7:      PS7
50  00300'000016*.PS105:    PS105
```

```
10011 MUP11
01                ; PG 75-11-27   PAGING SYSTEM                          PAGE..11..
02
03
04                ; PROCEDURE COMPUTE FRAME AND REL
05                ; SPLITS ADR IN (FRAME, REL) IF POSSIBLE
06                ;       CALL          RETURN          LINK
07                ; AC0   ADR           REL/(ADR)       +0: RETURN IF NO SPLIT
08                ; AC1                 DESTROYED       +1: RETURN
09                ; AC2   PROGRAM       PROGRAM
10                ; AC3   LINK          FRAME
11
12
13                .XREL
14
15 00000*175400 PS100:INC     3,3             ;   NORMALRETURN:= LINK+1;
16 00001*055045        STA     3   PWRK1,2     ;   SAVE(NORMAL RETURN);
17 00002*025016        LDA     1   PTOF,2      ;   TOPOFFRAMES:= PROGRAM.PTOF;
18 00003*106033        ADCZ#   0,1 SNC         ;   IF ADR >= TOPOFFRAMES
19 00004*001777        JMP         -1,3        ;   THEN RETURN(NO SPLIT);
20 00005*025015        LDA     1   PFOF,2      ;   FIRSTOFFRAMES:= PROGRAM.PFOF;
21 00006*106032        ADCZ#   0,1 SZC         ;   IF ADR< FIRSTOFFRAMES
22 00007*001777        JMP         -1,3        ;   THEN RETURN(NO SPLIT);
23 00010*122400        SUB     1,0             ;   AC0:= ADR-FIRSTOFFRAMES;
24 00011*035010        LDA     3   PMASK,2     ;   AC3:= PROGRAM.PMASK;
25 00012*117400        AND     0,3             ;   AC3:=FRAME-FIRSTOFFRAMES;
26 00013*162400        SUB     3,0             ;   AC0:= REL;
27 00014*137000        ADD     1,3             ;   AC3:= FRAME;
28 00015*003045        JMP@        PWRK1,2     ;   RETURN;
29
30
31                ; PROCEDURE CHECK VIRTUALREF AGAINST VICTIM
32                ; CHECKS AND ADVANCES VICTIM IF VIRTUALREF CORRESPONDS
33                ; TO VICTIM. USED IN CONNECTION WITH COROUTINE SPECIAL;
34                ;       CALL          RETURN          LINK
35                ; AC0   VIRTUALREF    UNCHANGED       +0: RETURN
36                ; AC1                 DESTROYED
37                ; AC2                 COROUT
38                ; AC3   LINK          UNCHANGED
39
40 00016*030040 PS105:LDA     2   CUR         ;   AC2:=PROCESS:= CUR;
41 00017*031012        LDA     2   PROG,2      ;   AC2:=PROGRAM:= PROCESS.PROG;
42 00020*055045        STA     3   PWRK1,2     ;   PROGRAM.WRK1:= RETURNADR;
43 00021*027017        LDA@    1   PVIC,2      ;   AC1:=REFVICT:=
44 00022*035010        LDA     3   PMASK,2     ;     PROGRAM.VICTIM.VIRTUALREF;
45 00023*167400        AND     3,1             ;   AC3:=REFPARAM:=
46 00024*117400        AND     0,3             ;     VIRTUALREF/PAGESIZE*PAGESIZE;
47 00025*136415        SUB#    1,3 SNR         ;   IF REFVICT = REFPARAM
48 00026*004553        JSR         PS110       ;   THEN ADVANCE VICTIM;
49 00027*035045        LDA     3   PWRK1,2     ;   AC3:=RETURNADR:=PROGRAM.WRK1;
50 00030*030017        LDA     2   COROUT      ;   AC2:= COROUT;
51 00031*001400        JMP         0,3         ;   RETURN;
52
53
```

```
10012 MUP11
01                ; PG 75-11-27   PAGING SYSTEM                          PAGE..12..
02
03
04                ; PROCEDURE PMOVE
05                ; MOVES VARIABLES IN DISABLE MODE
06                ;       CALL          RETURN          LINK
07                ; AC0                 UNCHANGED       +0: SOURCE PARAMETER
08                ; AC1                 DESTROYED       +1: DESTINATION PARAMETER
09                ; AC2                 PROG/COROUT     +2: RETURN
10                ; AC3   LINK          DESTROYED
11                ; PARAM(15:15)=0: (ABS ADR - 1) * 2
12                ; PARAM(14:15)=1: (PROGRAM RELATIVE - 1) * 4 + 1
13                ; PARAM(14:15)=3: (COROUTINE RELATIVE - 1) * 4 + 3
14
15 00032*054021 PMOVE:STA    3   17          ;   SAVE(RETURN);
16 00033*025400       LDA    1   0,3         ;   GET(SOURCE PARAMETER);
17 00034*004415       JSR        PGETAREA    ;
18 00035*044020       STA    1   16          ;   SAVE(ADR SOURCE);
19 00036*026021       LDA@   1   17          ;   GET(DESTINATION PARAMETER);
20 00037*004412       JSR        PGETAREA    ;
21 00040*034021       LDA    3   17          ;   GET(RETURN);
22 00041*044021       STA    1   17          ;   SAVE(ADR DESTINATION);
23 00042*024123       LDA    1   .6          ;
24 00043*044016       STA    1   14          ;   COUNT:= 6;
25 00044*026020       LDA@   1   16          ;   REPEAT
26 00045*046021       STA@   1   17          ;     MOVE 1 WORD;
27 00046*014016       DSZ        14          ;     COUNT:= COUNT-1;
28 00047*000775       JMP        .-3         ;   UNTIL COUNT = 0;
29 00050*001401       JMP        1,3         ;   RETURN;
30
31                PGETAREA:                  ; GETAREAADDRESS:
32 00051*125223       MOVZR  1,1 SNC         ;   IF PARAM(15:15) = 0
33 00052*001400       JMP        0,3         ;   THEN ADR:= PARAM/2
34 00053*030040       LDA    2   CUR         ;   ELSE
35 00054*031012       LDA    2   PROG,2      ;   ADR:= PARAM/4
36 00055*125222       MOVZR  1,1 SZC         ;   +(IF PARAM(14:14) = 0
37 00056*030017       LDA    2   COROUT      ;     THEN PROCESS.PROGRAM
38 00057*147000       ADD    2,1             ;     ELSE COROUT);
39 00060*001400       JMP        0,3         ;   RETURN;
40
41 00061*000147'.PS8: PS8                    ; STEPPING STONE
42
43
```

```
02
03
04                PSCHECK:                          ; CHECK AND ADVANCE VICTIM:
05 00062*021042         LDA     0    PAC3,2       ;   AC0:= PROGRAM.CURADR;
06 00063*004715         JSR          PS100        ;   COMPUTE CALL FRAME AND REL;
07 00064*176400         SUB     3,3               ;   IF NOTINFRAMES THEN CALLFRAME:= 0;
08 00065*055046         STA     3    PWRK2,2      ;   PROGRAM.PWRK2:= CALLFRAME;
09 00066*021017         LDA     0    PVIC,2       ;   AC0:=ORGVICT:= PROGRAM.PVIC;
10 00067*115000         MOV     0,3               ;   AC3:=VICTIM:= ORGVICT;
11                PS120:                            ; CHECK VICTIM:
12 00070*025046         LDA     1    PWRK2,2      ;
13 00071*136415         SUB#    1,3  SNR          ;   IF VICTIM = CALL FRAME
14 00072*000405         JMP          PS121        ;     OR
15 00073*025400         LDA     1    0,3          ;   FIRSTWORD ON VICTIM(14:15)
16 00074*034121         LDA     3    .3           ;     >= 2
17 00075*167625         ANDZR   3,1  SNR          ;   THEN
18 00076*000405         JMP          PS122        ;   BEGIN
19 00077*004502 PS121:  JSR          PS110        ;     AC3:= ADVANCE VICTIM;
20 00100*116415         SUB#    0,3  SNR          ;     IF VICTIM = ORGVICT
21 00101*000474         JMP          ERR2         ;     THEN GOTO ERROR2;
22 00102*000766         JMP          PS120        ;     GOTO CHECK VICTIM;
23                                                ;   END;
24                PS122:                            ; VICTIM FOUND:
25 00103*025017         LDA     1    PVIC,2       ;
26 00104*127000         ADD     1,1               ;   AC1:= VICTIM*2;
27 00105*102463         SUBC    0,0  SNC          ;   IF CARRY ON ENTRY THEN
28 00106*000410         JMP          PS123        ;   BEGIN C. PAGE SHOULD BE SAVED;
29 00107*023017         LDA@    0    PVIC,2       ;     PAGEREF:= CORE(VICTIM);
30 00110*006751         JSR@         .PS8         ;     AC0:= COMPUTEPAGE(PAGEREF);
31 00111*035012         LDA     3    PTAB,2       ;     AC3:=PAGETABLE:=PROGRAM.PTAB;
32 00112*117000         ADD     0,3               ;     AC3:=SECTOR:=PAGETABLE(PAGE);
33 00113*035400         LDA     3    0,3          ;
34 00114*045034         STA     1    POUTMES+2,2; ;     PROGRAM.OUTMESS2:=VICTIM*2;
35 00115*055035         STA     3    POUTMES+3,2; ;     PROGRAM.OUTMESS3:=SECTOR;
36 00116*041023 PS123:  STA     0    PAGEOUT,2    ;     PROGRAM.PAGEOUT:=PAGE
37                                                ;   END ELSE PAGEOUT:= 0;
38 00117*021043         LDA     0    PREF,2       ;   VIRTUALREF:=PROGRAM.PREF;
39 00120*006741         JSR@         .PS8         ;   AC0:= COMPUTEPAGE(VIRTUALREF);
40 00121*035012         LDA     3    PTAB,2       ;   AC3:=PAGETABLE:=PROGRAM.PTAB;
41 00122*117000         ADD     0,3               ;   AC3:=SECTOR:=PAGETABLE(PAGE);
42 00123*035400         LDA     3    0,3          ;
43 00124*045027         STA     1    PINMES+2,2;  ;   PROGRAM.INMESS2:=VICTIM*2;
44 00125*055030         STA     3    PINMES+3,2;  ;   PROGRAM.INMESS3:=SECTOR;
45 00126*041022         STA     0    PAGEIN,2     ;   PROGRAM.PAGEIN:=PAGE;
46
47 00127*035014         LDA     3    PSTAT,2      ; CALL STAT PROC:
48 00130*175005         MOV     3,3  SNR          ;   IF PROGRAM.STATPROC <> 0
49 00131*000404         JMP          PS124        ;   THEN CALL(STATPROC);
50 00132*005400         JSR          0,3          ;
51 00133*000402         JMP          PS124        ;
52 00134*000726         JMP          PSCHECK      ;   IF NEW VICTIM THEN GOTO
53                                                ;   CHECK AND ADVANCE VICTIM;
54
55
```

```
02
03
04 00135*023017 PS124:LDA@   0   PVIC,2     ; TRANSFER OF PAGES:
05 00136*101015       MOV#   0,0 SNR         ;   IF CORE(VICTIM) <> 0 THEN
06 00137*000415       JMP        PS125       ;   BEGIN
07 00140*006721       JSR@       .PS8        ;     PAGE:= COMPUTEPAGE(
08 00141*035013       LDA    3   PMAP,2      ;       CORE(VICTIM));
09 00142*117000       ADD    0,3             ;
10 00143*102620       SUBZR  0,0             ;
11 00144*041400       STA    0   0,3         ;     PMAP(PAGE):= 1B0;
12 00145*021023       LDA    0   PAGEOUT,2   ;
13 00146*101015       MOV#   0,0 SNR         ;     IF PAGEOUT <> 0 THEN
14 00147*000405       JMP        PS125       ;     BEGIN
15 00150*025031       LDA    1   POUTADR,2   ;       AC1:= OUTPUT MESSAGE ADDRESS;
16 00151*004445       JSR        PS130       ;       TRANSFER;
17 00152*011021       ISZ        PWRIT,2     ;       PAGESWRITTEN:= PAGESWRITTEN+1;
18 00153*000401       JMP        .+1         ;     END OUTPUT
19                PS125:                     ;   END OLD PAGE;
20 00154*025024       LDA    1   PINADR,2    ;   AC1:= INPUT MESSAGE ADDRESS;
21 00155*004441       JSR        PS130       ;   TRANSFER;
22 00156*011020       ISZ        PREAD,2     ;   PAGEREAD:= PAGEREAD+1;
23 00157*000401       JMP        .+1         ;
24 00160*021022       LDA    0   PAGEIN,2    ;   AC0:= PAGE;
25 00161*035013       LDA    3   PMAP,2      ;   AC3:= PROGRAM.PMAP;
26 00162*117000       ADD    0,3             ;
27 00163*023017       LDA@   0   PVIC,2      ;   AC1:= PAGEIN*PAGESIZE =
28 00164*025010       LDA    1   PMASK,2     ;     CORE(VICTIM) AND MASK;
29 00165*107400       AND    0,1             ;
30 00166*021017       LDA    0   PVIC,2      ;   AC0:= VICTIM;
31 00167*122400       SUB    1,0             ;   PMAP(PAGEIN):=
32 00170*041400       STA    0   0,3         ;     VICTIM - PAGEIN*PAGESIZE;
33 00171*004410       JSR        PS110       ;   ADVANCE VICTIM;
34 00172*002401       JMP@       .+1         ;   GOTO EXITPAGEFAULT;
35 00173*000212'      EXITPAGEFAULT          ;
36
37                ERR1:                      ; ERROR1: ADDRESSING ERROR;
38 00174*126401       SUB    1,1 SKP         ;   AC1:= 0; GOTO ERROR3;
39                ERR2:                      ; ERROR2: FRAME ERROR;
40 00175*126520       SUBZL  1,1             ;   AC1:= 1;
41                ERR3:                      ; ERROR3: DISK ERROR;
42 00176*020124       LDA    0   .7          ;   AC1 = DISK STATUS;
43 00177*030040       LDA    2   CUR         ;
44 00200*006012       BREAKPROCESS           ;   BREAKPROCESS(CUR, 7);
45
46
```

```
10015 MUP11
01                ; PG 75-11-27   PAGING SYSTEM                          PAGE..15..
02
03
04                ; PROCEDURE ADVANCE VICTIM
05                ;       CALL          RETURN          LINK
06                ; AC0                 UNCHANGED       +0: RETURN
07                ; AC1                 DESTROYED
08                ; AC2   PROGRAM       PROGRAM
09                ; AC3   LINK          VICTIM
10                ; PROGRAM.VICTIM IS UPDATED CYCLICALLY
11
12 00201*055047 PS110:STA     3   PWRK3,2    ; ADVANCE VICTIM: SAVE RETURN;
13 00202*035017       LDA     3   PVIC,2     ;   PROGRAM.VICTIM:=
14 00203*025007       LDA     1   PSIZE,2    ;     PROGRAM.VICTIM+PAGESIZE;
15 00204*137000       ADD     1,3            ;
16 00205*055017       STA     3   PVIC,2     ;
17 00206*137000       ADD     1,3            ;   PAGETOP:=VICTIM+PAGESIZE;
18 00207*025016       LDA     1   PTOF,2     ;
19 00210*166432       SUBZ#   3,1 SZC        ;   IF PAGETOP>TOPOFFRAMES THEN
20 00211*000403       JMP         PS111      ;   BEGIN
21 00212*035015       LDA     3   PFOF,2     ;     VICTIM:= FIRSTOFFRAMES;
22 00213*055017       STA     3   PVIC,2     ;     PROGRAM.VICTIM:= VICTIM;
23                PS111:                     ;   END;
24 00214*035017       LDA     3   PVIC,2     ;   AC3:= VICTIM;
25 00215*003047       JMP@        PWRK3,2    ;   RETURN;
26
27
28
```

```
!0016 MUP11
01                ; PG 76-07-26   PAGING SYSTEM                          PAGE..16..
02
03
04                ; PROCEDURE TRANSFER
05                ; TAKES CARE OF DISC I/O
06                ;       CALL          RETURN          LINK
07                ; AC0                 DESTROYED       +0: RETURN
08                ; AC1   ADR MESSAGE   DESTROYED
09                ; AC2   PROGRAM       PROGRAM
10                ; AC3   LINK          DESTROYED
11
12 00216*055050 PS130:STA     3   PWRK4,2    ; TRANSFER: SAVE RETURN;
13 00217*045046       STA     1   PWRK2,2    ;   SAVE(MESSAGEADDRESS);
14 00220*135000       MOV     1,3            ;
15 00221*025052       LDA     1   PDISP,2    ;
16 00222*021403       LDA     0   3,3        ;   MESS3:= MESS3
17 00223*123000       ADD     1,0            ;     + PROGRAM.PDISP;
18 00224*041403       STA     0   3,3        ;
19 00225*020122       LDA     0   .5         ;   COUNT:=5;
20 00226*041047       STA     0   PWRK3,2    ;
21                PS131:                     ; TRY AGAIN:
22 00227*025046       LDA     1   PWRK2,2    ;   AC1:= MESSAGE ADDESS;
23 00230*031051       LDA     2   PDRIVE,2   ;   AC2:= PROGRAM.PDRIVE;
24 00231*006004       SENDMESSAGE            ;   SEND MESSAGE;
25 00232*034017       LDA     3   COROUT      ;   IF COROUTINE
26 00233*175004       MOV     3,3 SZR        ;   THEN
27 00234*006337       CWANSWER               ;     COROUTINE WAITANSWER;
28 00235*006005       WAITANSWER             ;   WAIT ANSWER;
29 00236*031412       LDA     2   PROG,3      ;   AC2:=PROGRAM;
30 00237*105005       MOV     0,1 SNR        ;   IF STATUS = 0
31 00240*003050       JMP@        PWRK4,2    ;   THEN RETURN;
32 00241*015047       DSZ         PWRK3,2    ;   COUNT:= COUNT - 1;
33 00242*000402       JMP         .+2        ;   IF COUNT = 0
34 00243*000733       JMP         ERR3       ;   THEN GOTO ERROR3;
35 00244*024055       LDA     1   .0         ;   C. BECAUSE OF DISK DRIVER
36 00245*030126       LDA     2   .10        ;      ERROR;
37 00246*006003       WAITINTERRUPT          ;   DELAY(200MS);
38 00247*024407       LDA     1   PS133      ;   CLEAN DISK DRIVER
39 00250*031412       LDA     2   PROG,3     ;
40 00251*031051       LDA     2   PDRIVE,2   ;
41 00252*006004       SENDMESSAGE
42 00253*006005       WAITANSWER
43 00254*031412       LDA     2   PROG,3      ;   AC2:= PROGRAM;
44 00255*000752       JMP         PS131      ;   GOTO TRY AGAIN;
45
46 00256*000257*PS133:.+1                    ;   CONTROL MESSAGE
47 00257*000000       0                      ;
48 00260*000000       0                      ;
49 00261*000000       0                      ;
50 00262*000000       0                      ;
51
52
53
54                .END                       ; END OF PAGING SYSTEM
0000 SOURCE LINES IN ERROR
```

000047	4/34	4/35	15/12	15/25	16