

RCSL: 43-GL4053

AUTHOR: PG

EDITED: 77-04-18

01 ;
 02 ;
 03 ;
 04 ;
 05 ;
 06 ;
 07 ;
 08 ;
 09 ;
 10 ;
 11 ;
 12 ;
 13 ;
 14 ;
 15 ;
 16 ;
 17 ;
 18 ;
 19 ;
 20 ;
 21 ;
 22 ;
 23 ;
 24 ;
 25 ;
 26 ;
 27 ;
 28 ;
 29 ;
 30 ;
 31 ;
 32 ;
 33 ;
 34 ;
 35 ;
 36 ;
 37 ;
 38 ;
 39 ;
 40 ;
 41 ;
 42 ;
 43 ;
 44 ;

MJP03

; KEYWORD: MUS, PAGING SYSTEM, LISTING.
 ; ABSTRACT: MUS PAGING SYSTEM.
 ; ASCII PAPER TAPE: RCSL 43-GL4052.
 ; REL. BINARY PAPER TAPE: RCSL 43-GL4054.

; DESCRIPTION OF ENTRY POINTS:

; PROCEDURE COMON(SUBROUTINE CALL);

; EXECUTES THE SUBROUTINE CALL AND ARRANGES FOR A

; PROPER RETURN. MAY CHANGE THE PAGE MAP.

; CALL RETURN LINK

; AC0 *) *) +0: SUBROUTINE CALL

; AC1 *) *) +1: RETURN

; AC2 *) *)

; AC3 LINK *) *) AS FOR THE SUBROUTINE CALL

; PROCEDURE CALL(POINT);

; EXECUTES A SUBROUTINE JUMP TO THE POINT.

; MAY CHANGE THE PAGE MAP.

; CALL CONTINUATION LINK

; AC0 UNCHANGED +0: POINT

; AC1 UNCHANGED +1: POSSIBLE RETURN

; AC2 UNCHANGED

; AC3 LINK LINK+1

; PROCEDURE GOTO(POINT);

; EXECUTES A JUMP TO THE POINT.

; MAY CHANGE THE PAGE MAP.

; CALL CONTINUATION LINK

; AC0 UNCHANGED +0: POINT

; AC1 UNCHANGED

; AC2 UNCHANGED

; AC3 LINK DESTROYED

; PROCEDURE GETADR(POINT, ADDRESS);

; COMPUTES THE ADDRESS OF THE POINT GIVEN.

; MAY CHANGE THE PAGE MAP

; CALL RETURN LINK

; AC0 POINT UNCHANGED +0: RETURN

; AC1 UNCHANGED

; AC2 UNCHANGED

; AC3 LINK ADDRESS

; PROCEDURE GETPOINT(ADDRESS, POINT);

; COMPUTES THE POINT CORRESPONDING TO THE ADDRESS GIVEN.

; CALL RETURN LINK

; AC0 ADDRESS UNCHANGED +0: RETURN

; AC1 UNCHANGED

; AC2 UNCHANGED

; AC3 LINK POINT

01
02
03
05
06
07
08
09
10
11
13
14
16
17
19
20
22
24
25
26
27
28
29

.TITLE MUP03

000012 .RDX 10

000001 .TXTM 1

; TABLE OF PAGE ZERO ENTRY POINTS

000354 .LOC COMON-GOS ; PROCEDURE COMON
M00354 000354 PS0 ;

000355 .LOC CALL-GOS ; PROCEDURE CALL
M00355 000355 PS1 ;

000356 .LOC GOTO-GOS ; PROCEDURE GOTO
M00356 000356 PS2 ;

000357 .LOC GETADR-GOS ; PROCEDURE GETADR
M00357 000357 PS3 ;

000360 .LOC GETPOINT-GOS ; PROCEDURE GETPOINT
M00360 000360 PS4 ;

.NREL

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

; RELATIVE DISPLACEMENTS IN PROGRAMS

```

000007 PSIZE = NAME+3 ; PAGE SIZE
000010 PMASK = PSIZE+1 ; PAGE MASK
000011 PBLK = PMASK+1 ; BLOCKING FACTOR
000012 PTAB = PBLK+1 ; ADR. PAGETABLE/ADR. NO OF PAGES
000013 PMAP = PTAB+1 ; ADR. PAGEMAP/ADR. SEMAPHORE
000014 PSTAT = PMAP+1 ; ADR. PAGE STATISTICS PROCEDURE
000015 PFOF = PSTAT+1 ; FIRST OF FRAMES
000016 PTOF = PFOF+1 ; TOP OF FRAMES
000017 PVIC = PTOF+1 ; VICTIM
000020 PREAD = PVIC+1 ; NO OF PAGES READ
000021 PWRIT = PREAD+1 ; NO OF PAGES WRITTEN
000022 PAGEIN = PWRIT+1 ; PAGE TO BE LOADED INTO CORE
000023 PAGEOUT = PAGEIN+1 ; PAGE TO BE SAVED ONTO DISK
000024 PINADR = PAGEOUT+1 ; ADR. INPUT MESSAGE
000025 PINMES = PINADR+1 ; INPUT MESSAGE
000031 POUTADR = PINMES+4 ; ADR. OUTPUT MESSAGE
000032 POUTMES = POUTADR+1 ; OUTPUT MESSAGE
000036 PFLAG = POUTMES+4 ; PAGER FLAG(NOT USED!)

000037 PAC0 = PFLAG+1 ; SAVED AC0
000040 PAC1 = PAC0+1 ; SAVED AC1
000041 PAC2 = PAC1+1 ; SAVED AC2
000042 PAC3 = PAC2+1 ; SAVED AC3: CURRENT ADDRESS
000043 PREF = PAC3+1 ; VIRTUAL REFERENCE
000044 PLINK = PREF+1 ; LINK IN PAGEFAULT ROUTINE

000045 PWRK1 = PLINK+1 ; WORKING LOCATIONS
000046 PWRK2 = PWRK1+1 ;
000047 PWRK3 = PWRK2+1 ;
000050 PWRK4 = PWRK3+1 ;
000051 PDRIVE = PWRK4+1 ; ADR. NAME OF DISC DRIVER
000052 PDISP = PDRIVE+1 ; KIT DISPLACEMENT

```

; RELATIVE DISPLACEMENTS IN COROUTINES

```

000005 PCAC0 = CAC1SAVE+1 ; SAVED AC0
000006 PCAC1 = PCAC0+1 ; SAVED AC1
000007 PCAC2 = PCAC1+1 ; SAVED AC2
000010 PCAC3 = PCAC2+1 ; SAVED AC3: CURRENT POINT
000011 PCREF = PCAC3+1 ; VIRTUAL REFERENCE
000012 PCLINK = PCREF+1 ; LINK IN PAGEFAULT ROUTINE

```

01
02
03
04
05

; EXTERNAL PROCEDURE COMON(SUBROUTINE CALL);

```

PM00000'060277 PS0: INTDS ; DISABLE;
07 00001'040551 STA 0 PSAC0 ; SAVE(AC0);
08 00002'044551 STA 1 PSAC1 ; SAVE(AC1);
09 00003'050551 STA 2 PSAC2 ; SAVE(AC2);
10 00004'021400 LDA 0 0,3 ; AC0:= SUBROUTINE CALL;
11 00005'040412 STA 0 PS01 ; STORE(SUBROUTINE CALL);
12 00006'161400 INC 3,0 ; AC0:=RETURNADR:= CALLADR+1;
13 00007'004511 JSR PS7 ; AC3:= COMPUTEPOINT(RETURNADR);
14 00010'161000 MOV 3,0 ; AC0:= RETURNPOINT;
15 00011'034017 LDA 3 COROUT ; AC3:= CURRENT COROUTINE;
16 00012'041412 STA 0 PCLINK,3 ; CCOROUTINE.PCLINK:= RETURNPOINT;
17 00013'020537 LDA 0 PSAC0 ; RESTORE(AC0);
18 00014'024537 LDA 1 PSAC1 ; RESTORE(AC1);
19 00015'030537 LDA 2 PSAC2 ; RESTORE(AC2);
20 00016'060177 INTEN ; ENABLE AND
21 00017'000000 PS01: 0 ; EXECUTE(SUBROUTINE CALL);
22 ; RETURN FROM SUBROUTINE:
23 00020'060277 INTDS ; DISABLE;
24 00021'040531 STA 0 PSAC0 ; SAVE(AC0);
25 00022'034017 LDA 3 COROUT ;
26 00023'054532 STA 3 PSAC3 ; SAVE(AC3= COROUT);
27 00024'021412 LDA 0 PCLINK,3 ; AC3:=RETURNADR:=
28 00025'004450 JSR PS6 ; TAKEPOINT(CCOROUTINE.PCLINK);
29 00026'054524 STA 3 PSAC0 ; SAVE(RETURNADR);
30 00027'034526 LDA 3 PSAC3 ; RESTORE(AC3);
31 00030'060177 INTEN ; ENABLE AND
32 00031'002521 JMP@ PSAC0 ; RETURN;
33
34

```

01
02
03
04
05

; EXTERNAL PROCEDURE CALL(POINT);

```

PM00032'060277 PS1: INTDS ; DISABLE;
07 00033'040517 STA 0 PSAC0 ; SAVE(AC0);
08 00034'054521 STA 3 PSAC3 ; SAVE(AC3);
09 00035'021400 LDA 0 0,3 ; AC0:=POINT:= PARAMETER;
10 00036'004437 JSR PS6 ; AC3:= TAKEPOINT(POINT);
11 00037'054513 STA 3 PSAC0 ; SAVE(CONTINUEADR);
12 00040'034515 LDA 3 PSAC3 ; RESTORE(AC3);
13 00041'175400 INC 3,3 ; AC3:=RETURNADR:= CALLADR+1;
14 00042'060177 INTEN ; ENABLE AND
15 00043'002507 JMP@ PSAC0 ; EXIT TO CONTINUE ADDRESS;

```

16
17
18
19

; EXTERNAL PROCEDURE GOTO(POINT);

```

PM00044'060277 PS2: INTDS ; DISABLE;
21 00045'040505 STA 0 PSAC0 ; SAVE(AC0);
22 00046'021400 LDA 0 0,3 ; AC0:=POINT:= PARAMETER;
23 00047'176400 SUB 3,3 ; AC3:= 0; I.E. NO RETURNADR;
24 00050'054505 STA 3 PSAC3 ; SAVE(AC3);
25 00051'004424 JSR PS6 ; AC3:= TAKEPOINT(POINT);
26 00052'060177 INTEN ; ENABLE AND
27 00053'001400 JMP 0,3 ; EXIT TO CONTINUE ADDRESS;

```

28
29
30
31

; EXTERNAL PROCEDURE GETADR(POINT);

```

PM00054'060277 PS3: INTDS ; DISABLE;
33 00055'040475 STA 0 PSAC0 ; SAVE(AC0);
34 00056'054477 STA 3 PSAC3 ; SAVE(AC3);
35 00057'004416 JSR PS6 ; AC3:= TAKEPOINT(POINT);
36 00060'060177 INTEN ; ENABLE AND
37 00061'002474 JMP@ PSAC3 ; RETURN;

```

38
39
40
41

; EXTERNAL PROCEDURE GETPOINT(ADR);

```

PM00062'060277 PS4: INTDS ; DISABLE;
43 00063'040467 STA 0 PSAC0 ; SAVE(AC0);
44 00064'044467 STA 1 PSAC1 ; SAVE(AC1);
45 00065'050467 STA 2 PSAC2 ; SAVE(AC2);
46 00066'054467 STA 3 PSAC3 ; SAVE(AC3);
47 00067'004431 JSP PS7 ; AC3:= COMPUTEPOINT(ADR);
48 00070'020462 LDA 0 PSAC0 ; RESTORE(AC0);
49 00071'024462 LDA 1 PSAC1 ; RESTORE(AC1);
50 00072'030462 LDA 2 PSAC2 ; RESTORE(AC2);
51 00073'060177 INTEN ; ENABLE AND
52 00074'002461 JMP@ PSAC3 ; RETURN;

```

53
54

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

```
; PROCEDURE TAKEPOINT(POINT);
; COMPUTES THE ADDRESS OF A POINT
;           CALL          RETURN          LINK
; AC0      POINT        PSAC0          +0: RETURN
; AC1
; AC2
; AC3      LINK          ADDRESS
```

```
PS6:  STA      3  PSLINK      ; SAVE(LINK);
      MOVL    0,3  SNC        ; IF POINTREF(0:0) = 0
      JMP     PS60         ; THEN GOTO POINT TAKEN;
      ADDOR   0,0         ; AC0:=VIRTUALREF:= POINTREF-180;
      STA     0  PSREF      ; SAVE(VIRTUALREF);
      STA     2  PSAC2     ; SAVE(AC2);
PFRETURN:
      LDA     2  CUR        ; RETURN FROM PAGEFAULT:
      LDA     2  PROG,2    ; AC2:=PROCESS:= CUR;
      JSR    PS8          ; AC2:=PROGRAM:= PROCESS.PROG;
      LDA     3  PMAP,2    ; AC0:= COMPUTEPAGE(VITUALREF);
      ADD     0,3         ;
      LDA     3  0,3       ; AC3:= PMAP(PAGE);
      LDA     2  PSREF     ; AC2:= VIRTUALREF;
      ADDL   2,3  SZC      ; AC3:=BYTEADR:=
      JMP     PAGEFAULT   ; 2*(VIRTUALREF+PMAP(PAGE));
      LDA     2  PSAC2    ; IF BYTEADR>2**16 THEN PAGEFAULT;
      PS60:  MOVZR  3,3     ; RESTORE(AC2);
      LDA     0  PSAC0    ; POINT TAKEN:
      JMP@   PSLINK      ; AC3:=ABSADR:= BYTEADR/2;
                       ; RESTORE ORIGINAL(AC0);
                       ; RETURN;
```

```

01
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 00120'054000 PS7: STA 3 0 ; SAVE(LINK);
13 00121'030040 LDA 2 CUR ; AC2:=PROCESS:= CUR;
14 00122'031012 LDA 2 PROG,2 ; AC2:=PROGRAM:= PROCESS.PROG;
15 00123'004524 JSR PS100 ; COMPUTE FRAME AND REL;
16 00124'000406 JMP PS70 ; IF ADR IN FRAMES THEN
17 00125'035400 LDA 3 0,3 ; BEGIN AC3:= FIRST WORD ON FRAME;
18 00126'025010 LDA 1 PMASK,2 ; AC1:= PROGRAM.MASK;
19 00127'137400 AND 1,3 ; AC3:=POINT:=
20 00130'117000 ADD 0,3 ; PAGEBASE+REL+1B0
21 00131'177241 ADDR 3,3 SKP ; END
22 00132'115000 PS70: MOV 0,3 ; ELSE AC3:=POINT:= ADR;
23 00133'002000 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 00134'055045 PS8: STA 3 PWRK1,2 ; SAVE(AC3);
35 00135'035010 LDA 3 PMASK,2 ; AC3:=MASK:=PROGRAM.MASK;
36 00136'163705 ANDS 3,0 SNR ; AC0:= PAGE*BLOCKING FACTOR;
37 00137'002412 JMP@ .ERR ; IF PAGE = 0 THEN GOTO ERROR;
38 00140'035011 LDA 3 PBLK,2 ; BLFAC:= PROGRAM.BLFAC;
39 00141'175222 MOVZR 3,3 SZC ; FOR C:=BLFAC MOD 2,BLFAC:=BLFAC/2
40 00142'000403 JMP .+3 ; WHILE C=0 DO
41 00143'101220 MOVZR 0,0 ; AC0:=AC0/2
42 00144'000775 JMP .-3 ; NOW AC0=PAGE;
43 00145'037012 LDA@ 3 FTAB,2 ; AC3:=LASTPAGE:=PROGRAM.PTAB(0);
44 00146'116432 SUBZ# 0,3 SZC ; IF PAGENO <= LASTPAGE
45 00147'003045 JMP@ PWRK1,2 ; THEN RETURN;
46 00150'002401 JMP@ .ERR ; GOTO ERROR;
47 00151'000443 .ERR: ERR1 ; STEPPING STONE;
48
49 ; WORKING VARIABLES USED IN DISABLED MODE:
50 00152'000000 PSAC0: 0 ; SAVED AC0
51 00153'000000 PSAC1: 0 ; SAVED AC1
52 00154'000000 PSAC2: 0 ; SAVED AC2
53 00155'000000 PSAC3: 0 ; SAVED AC3/MODIFIED AC3
54 00156'000000 PSREF: 0 ; VIRTUAL REFERENCE
55 00157'000000 PSLINK: 0 ; RETURN ADR FROM PAGEFAULT/COMON
56
57

```



```

01
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 PAGEFAULT: ; PAGEFAULT:
20 00160'044773 STA 1 PSAC1 ; SAVE(AC1);
21 00161'030017 LDA 2 COROUT ;
22 00162'151015 MOV# 2,2 SNR ; IF COROUT<>0 THEN
23 00163'000407 JMP PS101 ; BEGIN
24 00164'020422 LDA 0 COLOWER ; TEST FOR COSPEC CALL:
25 00165'024422 LDA 1 COUPPER ;
26 00166'034767 LDA 3 PSAC3 ;
27 00167'162033 ADCZ# 3,0 SNC ; IF CALL ADDRESS < COLOWER OR
28 00170'166033 ADCZ# 3,1 SNC ; CALL ADDRESS >= COUPPER
29 00171'000417 JMP COSPEC ; THEN GOTO COROUTINE SPECIAL;
30 PS101: ; END COROUTINE;
31 00172'004507 JSR PMOVE ; MOVEVARIABLES(PAGER,PROGRAM);
32 00173'000322" PSAC0-1*2 ;
33 00174'000171 PAC0-1*4+1 ;
34 00175'060177 INTEN ; ENABLE;
35 00176'000533 JMP PSCHECK ; GOTO CHECK AND ADVANCE VICTIM;
36
37 EXITPAGEFAULT: ; EXITPAGEFAULT:
38 00177'060277 INTDS ; DISABLE;
39 00200'004501 JSR PMOVE ; MOVEVARIABLES(PROGRAM,PAGER);
40 00201'000171 PAC0-1*4+1 ;
41 00202'000322" PSAC0-1*2 ;
42 PS102: ; EXIT FROM COROUTINESPECIAL:
43 00203'020753 LDA 0 PSREF ; AC0:= PSREF;
44 00204'024747 LDA 1 PSAC1 ; RESTORE(AC1);
45 00205'000676 JMP PFRETURN ; RETURN FROM PAGEFAULT;
46 00206'000210'COLOWER: COSPEC
47 00207'000247'COUPPER: PS100
48
49

```

```

01
02
03
04 COSPEC: ; COROUTINE SPECIAL:
05 00210'020745 LDA 0 PSAC3 ; ACO:=CURADR:= PSAC3;
06 00211'004707 JSR PS7 ; PSAC3:=CURPOINT:=
07 00212'054743 STA 3 PSAC3 ; COMPUTEPOINT(CURADR);
08 00213'004466 JSR PMOVE ; MOVEVARIABLES(PAGER,COROUTINE);
09 00214'000322" PSACU-1*2 ;
10 00215'000023 PCACU-1*4+3 ;
11 00216'060177 INTEN ; ENABLE;
12 00217'030040 LDA 2 CUR ; AC2:=PROCESS:= CUR;
13 00220'031012 LDA 2 PROG,2 ; AC2:=PROGRAM:= PROCESS.PROG;
14 00221'025013 LDA 1 PMAP,2 ; AC1:=SEMAPHORE:=PMAP(0);
15 00222'006335 WAITSEMAPHORE ; WAITSEMAPHORE(PAGER READY);
16 00223'021411 LDA 0 PCREF,3 ; IF VICTIM = REFPAGE
17 00224'004441 JSR PS105 ; THEN ADVANCE VICTIM;
18 00225'021010 LDA 0 PCAC3,2 ;
19 00226'103240 ADDOR 0,0 ; IF VICTIM = CURPAGE
20 00227'004436 JSR PS105 ; THEN ADVANCE VICTIM;
21 00230'103240 ADDOR 0,0 ;
22 00231'006357 GETADR ; CCOROUT.CURADR:=
23 00232'055010 STA 3 PCAC3,2 ; GETADR(CURPOINT);
24 00233'021011 LDA 0 PCREF,2 ;
25 00234'103240 ADDOR 0,0 ;
26 00235'006357 GETADR ; GETADR(REFPOINT);
27 00236'030040 LDA 2 CUR ; AC2:=PROCESS:= CUR;
28 00237'031012 LDA 2 PROG,2 ; AC2:=PROGRAM:=PROCESS.PROG;
29 00240'025013 LDA 1 PMAP,2 ; AC1:= PROGRAM.SEMAPHORE;
30 00241'006343 SIGNAL ; SIGNAL(PROGRAM.SEMAPHORE);
31 00242'060277 INTDS ; DISARLE;
32 00243'004436 JSR PMOVE ; MOVE VARIABLES(COROUTINE, PAGER);
33 00244'000023 PCACU-1*4+3 ;
34 00245'000322" PSACU-1*2 ;
35 00246'000735 JMP PS102 ; GOTO EXIT FROM COROUTINE SPECIAL;
36
37

```

```

01
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 00247'175400 PS100:INC 3,3 ; NORMALRETURN:= LINK+1;
13 00250'055045 STA 3 PWRK1,2 ; SAVE(NORMAL RETURN);
14 00251'025016 LDA 1 PTOF,2 ; TOPOFFRAMES:= PROGRAM.PTOF;
15 00252'106033 ADCZ# 0,1 SNC ; IF ADR >= TOPOFFRAMES
16 00253'001777 JMP -1,3 ; THEN RETURN(NO SPLIT);
17 00254'025015 LDA 1 PFOF,2 ; FIRSTOFFRAMES:= PROGRAM.PFOF;
18 00255'106032 ADCZ# 0,1 SZC ; IF ADR< FIRSTOFFRAMES
19 00256'001777 JMP -1,3 ; THEN RETURN(NO SPLIT);
20 00257'122400 SUB 1,0 ; AC0:= ADR-FIRSTOFFRAMES;
21 00260'035010 LDA 3 PMASK,2 ; AC3:= PROGRAM.PMASK;
22 00261'117400 AND 0,3 ; AC3:=FRAME-FIRSTOFFRAMES;
23 00262'162400 SUB 3,0 ; AC0:= REL;
24 00263'137000 ADD 1,3 ; AC3:= FRAME;
25 00264'003045 JMP@ PWRK1,2 ; RETURN;
26
27
28 ; PROCEDURE CHECK VIRTUALREF AGAINST VICTIM
29 ; CHECKS AND ADVANCES VICTIM IF VIRTUALREF CORRESPONDS
30 ; TO VICTIM. USED IN CONNECTION WITH COROUTINE SPECIAL;
31 ; CALL RETURN LINK
32 ; AC0 VIRTUALREF UNCHANGED +0: RETURN
33 ; AC1 DESTROYED
34 ; AC2 COROUT
35 ; AC3 LINK UNCHANGED
36
37 00265'030040 PS105:LDA 2 CUR ; AC2:=PROCESS:= CUR;
38 00266'031012 LDA 2 PROG,2 ; AC2:=PROGRAM:= PROCESS.PROG;
39 00267'055045 STA 3 PWRK1,2 ; PROGRAM.WRK1:= RETURNADR;
40 00270'027017 LDA@ 1 PVIC,2 ; AC1:=REFVICT:=
41 00271'035010 LDA 3 PMASK,2 ; PROGRAM.VICTIM.VIRTUALREF;
42 00272'167400 AND 3,1 ; AC3:=REFPARAM:=
43 00273'117400 AND 0,3 ; VIRTUALREF/PAGESIZE*PAGESIZE;
44 00274'136415 SUB# 1,3 SNR ; IF REFVICT = REFPARAM
45 00275'004553 JSR PS110 ; THEN ADVANCE VICTIM;
46 00276'035045 LDA 3 PWRK1,2 ; AC3:=RETURNADR:=PROGRAM.WRK1;
47 00277'030017 LDA 2 COROUT ; AC2:= COROUT;
48 00300'001400 JMP 0,3 ; RETURN;
49
50

```

```

01
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 00301'054021 PMOVE: STA 3 17 ; SAVE(RETURN);
16 00302'025400 LDA 1 0,3 ; GET(SOURCE PARAMETER);
17 00303'004415 JSR PGETAREA ;
18 00304'044020 STA 1 16 ; SAVE(ADR SOURCE);
19 00305'026021 LDA@ 1 17 ; GET(DESTINATION PARAMETER);
20 00306'004412 JSR PGETAREA ;
21 00307'034021 LDA 3 17 ; GET(RETURN);
22 00310'044021 STA 1 17 ; SAVE(ADR DESTINATION);
23 00311'024123 LDA 1 ,6 ;
24 00312'044016 STA 1 14 ; COUNT:= 6;
25 00313'026020 LDA@ 1 16 ; REPEAT
26 00314'046021 STA@ 1 17 ; MOVE 1 WORD;
27 00315'014016 DSZ 14 ; COUNT:= COUNT-1;
28 00316'000775 JMP , -3 ; UNTIL COUNT = 0;
29 00317'001401 JMP 1,3 ; RETURN;
30
31 PGETAREA: ; GETAREAADDRESS:
32 00320'125223 MOVZR 1,1 SNC ; IF PARAM(15:15) = 0
33 00321'001400 JMP 0,3 ; THEN ADR:= PARAM/2
34 00322'030040 LDA 2 CUR ; ELSE
35 00323'031012 LDA 2 PROG,2 ; ADR:= PARAM/4
36 00324'125222 MOVZR 1,1 SZC ; +(IF PARAM(14:14) = 0
37 00325'030017 LDA 2 COROUT ; THEN PROCESS PROGRAM
38 00326'147000 ADD 2,1 ; ELSE COROUT);
39 00327'001400 JMP 0,3 ; RETURN;
40
41 00330'000134'.PS8: PS8 ; STEPPING STONE
42
43

```

```

01
02
03
04 PSCHECK: ; CHECK AND ADVANCE VICTIM:
05 00331'021042 LDA 0 PAC3,2 ; ACO:= PROGRAM.CURADR;
06 00332'004715 JSR PS100 ; COMPUTE CALL FRAME AND REL;
07 00333'176400 SUB 3,3 ; IF NOTINFRAMES THEN CALLFRAME:= 0;
08 00334'055046 STA 3 PWRK2,2 ; PROGRAM.PWRK2:= CALLFRAME;
09 00335'021017 LDA 0 PVIC,2 ; ACO:=ORGVICT:= PROGRAM.PVIC;
10 00336'115000 MOV 0,3 ; AC3:=VICTIM:= ORGVICT;
11 PS120: ; CHECK VICTIM:
12 00337'025046 LDA 1 PWRK2,2 ;
13 00340'136415 SUB# 1,3 SNR ; IF VICTIM = CALL FRAME
14 00341'000405 JMP PS121 ; OR
15 00342'025400 LDA 1 0,3 ; FIRSTWORD ON VICTIM(14:15)
16 00343'034121 LDA 3 .3 ; >= 2
17 00344'167625 ANDZR 3,1 SNR ; THEN
18 00345'000405 JMP PS122 ; BEGIN
19 00346'004502 PS121:JSR PS110 ; AC3:= ADVANCE VICTIM;
20 00347'116415 SUB# 0,3 SNR ; IF VICTIM = ORGVICT
21 00350'000474 JMP ERR2 ; THEN GOTO ERROR2;
22 00351'000766 JMP PS120 ; GOTO CHECK VICTIM;
23 ; END;
24 PS122: ; VICTIM FOUND:
25 00352'025017 LDA 1 PVIC,2 ;
26 00353'127000 ADD 1,1 ; AC1:= VICTIM*2;
27 00354'102463 SUBC 0,0 SNC ; IF CARRY ON ENTRY THEN
28 00355'000410 JMP PS123 ; BEGIN C. PAGE SHOULD BE SAVED;
29 00356'023017 LDA@ 0 PVIC,2 ; PAGEREF:= CORE(VICTIM);
30 00357'006751 JSR@ .PS8 ; ACO:= COMPUTEPAGE(PAGEREF);
31 00360'035012 LDA 3 PTAB,2 ; AC3:=PAGETABLE:=PROGRAM.PTAB;
32 00361'117000 ADD 0,3 ; AC3:=SECTOR:=PAGETABLE(PAGE);
33 00362'035400 LDA 3 0,3 ;
34 00363'045034 STA 1 POUTMES+2,2; PROGRAM.OUTMESS2:=VICTIM*2;
35 00364'055035 STA 3 POUTMES+3,2; PROGRAM.OUTMESS3:=SECTOR;
36 00365'041023 PS123:STA 0 PAGEOUT,2 ; PROGRAM.PAGEOUT:=PAGE
37 ; END ELSE PAGEOUT:= 0;
38 00366'021043 LDA 0 PREF,2 ; VIRTUALREF:=PROGRAM.PREF;
39 00367'006741 JSR@ .PS8 ; ACO:= COMPUTEPAGE(VIRTUALREF);
40 00370'035012 LDA 3 PTAB,2 ; AC3:=PAGETABLE:=PROGRAM.PTAB;
41 00371'117000 ADD 0,3 ; AC3:=SECTOR:=PAGETABLE(PAGE);
42 00372'035400 LDA 3 0,3 ;
43 00373'045027 STA 1 PINMES+2,2; PROGRAM.INMESS2:=VICTIM*2;
44 00374'055030 STA 3 PINMES+3,2; PROGRAM.INMESS3:=SECTOR;
45 00375'041022 STA 0 PAGEIN,2 ; PROGRAM.PAGEIN:=PAGE;
46
47 00376'035014 LDA 3 PSTAT,2 ; CALL STAT PROC:
48 00377'175005 MOV 3,3 SNR ; IF PROGRAM.STATPROC <> 0
49 00400'000404 JMP PS124 ; THEN CALL(STATPROC);
50 00401'005400 JSR 0,3 ;
51 00402'000402 JMP PS124 ;
52 00403'000726 JMP PSCHECK ; IF NEW VICTIM THEN GOTO
53 ; CHECK AND ADVANCE VICTIM;
54
55

```

01

02

03

```

04 00404'023017 PS124:LDA@ 0 PVIC,2 ; TRANSFER OF PAGES:
05 00405'101015 MOV# 0,0 SNR ; IF CORE(VICTIM) <> 0 THEN
06 00406'000415 JMP PS125 ; BEGIN
07 00407'006721 JSR@ .PS8 ; PAGE:= COMPUTEPAGE(
08 00410'035013 LDA 3 PMAP,2 ; CORE(VICTIM));
09 00411'117000 ADD 0,3 ;
10 00412'102620 SUBZR 0,0 ;
11 00413'041400 STA 0 0,3 ; PMAP(PAGE):= 180;
12 00414'021023 LDA 0 PAGEOUT,2 ;
13 00415'101015 MOV# 0,0 SNR ; IF PAGEOUT <> 0 THEN
14 00416'000405 JMP PS125 ; BEGIN
15 00417'025031 LDA 1 POUTADR,2 ; AC1:= OUTPUT MESSAGE ADDRESS;
16 00420'004445 JSR PS130 ; TRANSFER;
17 00421'011021 ISZ PWRIT,2 ; PAGESWRITTEN:= PAGESWRITTEN+1;
18 00422'000401 JMP .+1 ; END OUTPUT
19 PS125: ; END OLD PAGE;
20 00423'025024 LDA 1 PINADR,2 ; AC1:= INPUT MESSAGE ADDRESS;
21 00424'004441 JSR PS130 ; TRANSFER;
22 00425'011020 ISZ PREAD,2 ; PAGEREAD:= PAGEREAD+1;
23 00426'000401 JMP .+1 ;
24 00427'021022 LDA 0 PAGEIN,2 ; AC0:= PAGE;
25 00430'035013 LDA 3 PMAP,2 ; AC3:= PROGRAM.PMAP;
26 00431'117000 ADD 0,3 ;
27 00432'023017 LDA@ 0 PVIC,2 ; AC1:= PAGEIN*PAGESIZE =
28 00433'025010 LDA 1 PMASK,2 ; CORE(VICTIM) AND MASK;
29 00434'107400 AND 0,1 ;
30 00435'021017 LDA 0 PVIC,2 ; AC0:= VICTIM;
31 00436'122400 SUB 1,0 ; PMAP(PAGEIN):=
32 00437'041400 STA 0 0,3 ; VICTIM - PAGEIN*PAGESIZE;
33 00440'004410 JSR PS110 ; ADVANCE VICTIM;
34 00441'002401 JMP@ .+1 ; GOTO EXITPAGEFAULT;
35 00442'000177' EXITPAGEFAULT ;
36
37 ERR1: ; ERROR1: ADDRESSING ERROR;
38 00443'126401 SUB 1,1 SKP ; AC1:= 0; GOTO ERROR3;
39 ERR2: ; ERROR2: FRAME ERROR;
40 00444'126520 SUBZL 1,1 ; AC1:= 1;
41 ERR3: ; ERROR3: DISK ERROR;
42 00445'020124 LDA 0 .7 ; AC1 = DISK STATUS;
43 00446'030040 LDA 2 CUR ;
44 00447'006012 BREAKPROCESS ; BREAKPROCESS(CUR, 7);
45
46

```

```

01
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 00450'055047 PS110:STA     3   PWRK3,2 ; ADVANCE VICTIM: SAVE RETURN;
13 00451'035017     LDA     3   PVIC,2  ; PROGRAM.VICTIM:=
14 00452'025007     LDA     1   PSIZE,2  ; PROGRAM.VICTIM+PAGESIZE;
15 00453'137000     ADD     1,3          ;
16 00454'055017     STA     3   PVIC,2  ;
17 00455'137000     ADD     1,3          ; PAGETOP:=VICTIM+PAGESIZE;
18 00456'025016     LDA     1   PTOF,2  ;
19 00457'166432     SUBZ#   3,1 SZC      ; IF PAGETOP>TOPOFFRAMES THEN
20 00460'000403     JMP     PS111        ; BEGIN
21 00461'035015     LDA     3   PFOF,2  ; VICTIM:= FIRSTOFFRAMES;
22 00462'055017     STA     3   PVIC,2  ; PROGRAM.VICTIM:= VICTIM;
23 PS111:          ; END;
24 00463'035017     LDA     3   PVIC,2  ; AC3:= VICTIM;
25 00464'003047     JMP@    PWRK3,2  ; RETURN;
26
27
28

```

```

01
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 00465'055050 PS130:STA 3 PWRK4,2 ; TRANSFER: SAVE RETURN;
13 00466'045046 STA 1 PWRK2,2 ; SAVE(MESSAGEADDRESS);
14 00467'135000 MOV 1,3 ;
15 00470'025052 LDA 1 PDISP,2 ;
16 00471'021403 LDA 0 3,3 ; MESS3:= MESS3
17 00472'123000 ADD 1,0 ; + PROGRAM.PDISP;
18 00473'041403 STA 0 3,3 ;
19 00474'020122 LDA 0 .5 ; COUNT:=5;
20 00475'041047 STA 0 PWRK3,2 ;
21 PS131: ; TRY AGAIN:
22 00476'025046 LDA 1 PWRK2,2 ; AC1:= MESSAGE ADDRESS;
23 00477'031051 LDA 2 PDRIVE,2 ; AC2:= PROGRAM.PDRIVE;
24 00500'006004 SENDMESSAGE ; SEND MESSAGE;
25 00501'034017 LDA 3 COROUT ; IF COROUTINE
26 00502'175004 MOV 3,3 SZR ; THEN
27 00503'006337 CWANSWER ; COROUTINE WAITANSWER;
28 00504'006005 WAITANSWER ; WAIT ANSWER;
29 00505'031412 LDA 2 PROG,3 ; AC2:=PROGRAM;
30 00506'105005 MOV 0,1 SNR ; IF STATUS = 0
31 00507'003050 JMP@ PWRK4,2 ; THEN RETURN;
32 00510'015047 DSZ PWRK3,2 ; COUNT:= COUNT - 1;
33 00511'000402 JMP .+2 ; IF COUNT = 0
34 00512'000733 JMP ERR3 ; THEN GOTO ERROR3;
35 00513'024055 LDA 1 .0 ; C. BECAUSE OF DISK DRIVER
36 00514'030126 LDA 2 .10 ; ERROR;
37 00515'006003 WAITINTERRUPT ; DELAY(200MS);
38 00516'024407 LDA 1 PS133 ; CLEAN DISK DRIVER
39 00517'031412 LDA 2 PROG,3 ;
40 00520'031051 LDA 2 PDRIVE,2 ;
41 00521'006004 SENDMESSAGE
42 00522'006005 WAITANSWER
43 00523'031412 LDA 2 PROG,3 ; AC2:= PROGRAM;
44 00524'000752 JMP PS131 ; GOTO TRY AGAIN;
45
46 00525'000526'PS133: .+1 ; CONTROL MESSAGE
47 00526'000000 0 ;
48 00527'000000 0 ;
49 00530'000000 0 ;
50 00531'000000 0 ;
51
52
53
54 .END ; END OF PAGING SYSTEM

```

0010 SOURCE LINES IN ERROR

0017 MUP03

COLOW	000206'	9/24	9/46						
COSPE	000210'	9/29	9/46	10/04					
CO P	000207'	9/25	9/47						
ERR1	000443'	8/47	14/37						
ERR2	000444'	13/21	14/39						
ERR3	000445'	14/41	16/34						
EXITP	000177'	9/37	14/35						
PACO	000037	4/25	4/26	9/33	9/40				
PAC1	000040	4/26	4/27						
PAC2	000041	4/27	4/28						
PAC3	000042	4/28	4/29	13/05					
PAGEF	000160'	7/28	9/19						
PAGEI	000022	4/17	4/18	13/45	14/24				
PAGEO	000023	4/18	4/19	13/36	14/12				
PHLK	000011	4/08	4/09	8/38					
PCACO	000005	4/42	4/43	10/10	10/33				
PCAC1	000006	4/43	4/44						
PCAC2	000007	4/44	4/45						
PCAC3	000010	4/45	4/46	10/18	10/23				
PCLIN	000012	4/47	5/16	5/27					
PCREF	000011	4/46	4/47	10/16	10/24				
PD P	000052	4/37	16/15						
PDRIV	000051	4/36	4/37	16/23	16/40				
PFLAG	000036	4/23	4/25						
PFOF	000015	4/12	4/13	11/17	15/21				
PERET	000103'	7/18	9/45						
PGETA	000320'	12/17	12/20	12/31					
PINAD	000024	4/19	4/20	14/20					
PINME	000025	4/20	4/21	13/43	13/44				
PLINK	000044	4/30	4/32						
PMAP	000013	4/10	4/11	7/22	10/14	10/29	14/08	14/25	
PMASK	000010	4/07	4/08	8/18	8/35	11/21	11/41	14/28	
PMOVE	000301'	9/31	9/39	10/08	10/32	12/15			
POUTA	000031	4/21	4/22	14/15					
POUTM	000032	4/22	4/23	13/34	13/35				
PREAD	000020	4/15	4/16	14/22					
PREF	000043	4/29	4/30	13/38					
PS0	000354	5/06							
PS01	000017'	5/11	5/21						
PS1	000355	3/15	6/06						
PS0	000247'	8/15	9/47	11/12	13/06				
PS101	000172'	9/23	9/30						
PS102	000203'	9/42	10/35						
PS105	000265'	10/17	10/20	11/37					
PS110	000450'	11/45	13/19	14/33	15/12				
PS111	000463'	15/20	15/23						
PS120	000337'	13/11	13/22						
PS121	000346'	13/14	13/19						
PS122	000352'	13/18	13/24						
PS123	000365'	13/28	13/36						
PS124	000404'	13/49	13/51	14/04					
PS125	000423'	14/06	14/14	14/19					
PS130	000465'	14/16	14/21	16/12					
PS131	000476'	16/21	16/44						
PS133	000525'	16/38	16/46						
PS2	000356	3/18	6/20						
PS3	000357	3/21	6/32						
PS4	000360	3/24	6/42						
PS6	000075'	5/28	6/10	6/25	6/35	7/12			
PS	000115'	7/14	7/30						

0018 MUP03

PS7	000120'	5/13	6/47	8/12	10/06				
PS70	000132'	8/16	8/22						
PS	000134'	7/21	8/34	12/41					
PSAC0	000152'	5/07	5/17	5/24	5/29	5/32	6/07	6/11	
		6/15	6/21	6/33	6/43	6/48	7/32	8/50	
		9/32	9/41	10/09	10/34				
PSAC1	000153'	5/08	5/18	6/44	6/49	8/51	9/20	9/44	
PSAC2	000154'	5/09	5/19	6/45	6/50	7/17	7/29	8/52	
PSAC3	000155'	5/26	5/30	6/08	6/12	6/24	6/34	6/37	
		6/46	6/52	8/53	9/26	10/05	10/07		
PSCHE	000331'	9/35	13/04	13/52					
PSIZE	000007	4/06	4/07	15/14					
PSLIN	000157'	7/12	7/33	8/55					
PSREF	000156'	7/16	7/25	8/54	9/43				
PSTAT	000014	4/11	4/12	13/47					
PTAB	000012	4/09	4/10	8/43	13/31	13/40			
PTOF	000016	4/13	4/14	11/14	15/18				
PVIC	000017	4/14	4/15	11/40	13/09	13/25	13/29	14/04	
		14/27	14/30	15/13	15/16	15/22	15/24		
PWRIT	000021	4/16	4/17	14/17					
PWRK1	000045	4/32	4/33	8/34	8/45	11/13	11/25	11/39	
		11/46							
PWRK2	000046	4/33	4/34	13/08	13/12	16/13	16/22		
PWRK3	000047	4/34	4/35	15/12	15/25	16/20	16/32		
PWRK4	000050	4/35	4/36	16/12	16/31				
ERR	000151'	8/37	8/46	8/47					
PS8	000330'	12/41	13/30	13/39	14/07				