

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

MUM12

; KEYWORDS: MUS, MONITOR, LISTING.

; ABSTRACT: MULTIPROGRAMMING UTILITY SYSTEM
MONITOR PROCEDURES.

; ASCII PAPER TAPE: RCSL 43-GL10256
; REL. BINARY PAPER TAPE: RCSL 43-GL10258

WIL

MUM 12 PAGE 0002

MISSING

20041209 PHK

```

01          ; ***** MONITOR FUNCTION ENTRIES *****
02
03          000000 .LOC 0          ; MONITOR START IN WORD 0
04
05 00000 000000          0          ; 0: SAVED PC AT INTERRUPT, ALSO WORK CELL
06 00001 000067'        A0         ; 1: INTERRUPT RESPONSE ADDRESS
07
08          F2:              ; 2: WAIT , (START OF MUI MODULE RESTORED BY MUI )
09          000003 .LOC .+1
10 00003 000417'F3:      A20        ; 3: WAIT INTERRUPT
11 00004 000417'F4:      A20        ; 4: SEND MESSAGE
12 00005 000000 F5:      0          ; 5: WAIT ANSWER ( RESTORED BY MUI MODULE)
13 00006 000417'F6:      A20        ; 6: WAIT EVENT
14 00007 000417'F7:      A20        ; 7: SEND ANSWER
15 00010 000417'F8:      A20        ; 8: SEARCH ITEM
16 00011 000417'F9:      A20        ; 9: CLEAN PROCESS
17 00012 000417'F10:     A20        ; 10: BREAK PROCESS
18 00013 000417'F11:     A20        ; 11: STOP PROCESS
19 00014 000417'F12:     A20        ; 12: START PROCESS
20 00015 000417'F13:     A20        ; 13: RECHAIN
21 00016 000000 WORK:    0          ; 14: PAGE ZERO LOCATION
22 00017 000000 LINK:    0          ; 15: PAGE ZERO LOCATION
23
24          FUNCTION:        ; MONITOR FUNCTION ENTRIES:
25 00020 000000          0          ; 16: AUTO-INCREMENTING LOCATION
26 00021 000000          0          ; 17: AUTO-INCREMENTING LOCATION
27 00022 000452'        A22        ; 18: WAIT(Delay,DEVICE,FIRST,SECOND,BUF)
28 00023 000470'        A23        ; 19: WAIT INTERRUPT(DEVICE,DELAY)
29 00024 000510'        A24        ; 20: SEND MESSAGE(ADDR,NAME ADDR,BUF)
30 00025 000620'        A25        ; 21: WAIT ANSWER(FIRST,SECOND,BUF)
31 00026 000627'        A26        ; 22: WAIT EVENT(FIRST,SECOND,BUF)
32 00027 000637'        A27        ; 23: SEND ANSWER(FIRST,SECOND,BUF)
33 00030 000644'        A28        ; 24: SEARCH ITEM(CHAIN,NAME ADDR,ITEM)
34 00031 000650'        A29        ; 25: CLEAN PROCESS(PROC)
35 00032 000705'        A30        ; 26: BREAK PROCESS(PROC,ERROR NUMBER)
36 00033 000720'        A31        ; 27: STOP PROCESS(PROC)
37 00034 000733'        A32        ; 28: START PROCESS(PROC)
38 00035 000737'        A33        ; 29: RECHAIN(OLD,NEW,ELEM)
39 00036 000000          0          ; 30: AUTO-DECREMENTING LOCATION
40 00037 000000          0          ; 31: AUTO-DECREMENTING LOCATION

```

```

01 ; ***** MONITOR PROCESS DESCRIPTOR *****
02
03 ; THE MONITOR PROCESS DESCRIPTOR HOLDS THE MONITOR TABLE, THE
04 ; MONITOR CODE, THE SYSTEM UTILITY PROCEDURES, AND THE INPUT/
05 ; OUTPUT UTILITY PROCEDURES.
06
07 ; ***** MONITOR TABLE *****
08
09 ;DUSR M= . ; MONITOR PROCESS DESCRIPTOR:
10 ;DUSR CUR= . ;
11 00040 000040 CUR ; NEXT: FIRST PROCESS IN RUNNING QUEUE
12 00041 000040 CUR ; PREV: LAST PROCESS IN RUNNING QUEUE
13 ; HEAD OF RUNNING QUEUE AND HEAD OF
14 ; PROCESS CHAIN.
15
16 00042 000000 0 ; CHAIN: PROCESS DESCRIPTOR OF FIRST
17 ; PROCESS IN PROCESS CHAIN.
18 00043 000000 WORK1: 0 ; SIZE: MONITOR WORK
19 00044 000000 WORK2: 0 ; NAME: MONITOR WORK
20 ;DUSR TABLE=.
21 00045 000370 DEVTA ; +1: DEVICE TABLE
22 ; CONTAINS A WORD FOR EACH DEVICE
23 ; NUMBER HOLDING PROCESS DESCRIPTORS
24 ; FOR INTERRUPT REQUESTING PROCESSES.
25 ;DUSR TOPTA=.
26 00046 000464 TOPDE ; +2: TOP OF DEVICE TABLE
27 DFIRS:
28 00047 000047 DFIRS ; EVENT: FIRST PROCESS IN DELAY QUEUE
29 00050 000047 DFIRS ; +1: LAST PROCESS IN DELAY QUEUE
30 ; HEAD OF DELAY QUEUE
31 00051 000706 'R301: A301 ; BUFFER: POINT
32 ;DUSR PFIRS=.
33 00052 000000 0 ; PROG: FIRST IN PROGRAM CHAIN
34
35 00053 000000 0 ; STATE: MONITOR STATE: ALWAYS ZERO
36 ;DUSR RUNNI=.
37 00054 000040 CUR ; TIMER: RUNNING QUEUE
38 ;DUSR PROCE=RUNNI ; REFERENCE TO HEAD OF RUNNING QUEUE AND
39 ;DUSR MONIT=RUNNI ; HEAD OF PROCESS CHAIN: MONITOR PROCESS
40
41 00055 000000 0 ; PRIOR: MONITOR PRIORITY: LOWEST POSSIBLE: ZERO
42 ;DUSR EXIT=.
43 00056 000136' A1 ; BREAD: MONITOR EXIT
44 ;DUSR EFIRS=.
45 00057 000000 0 ; AC0: FIRST IN FREE CORE
46 ;DUSR FFIRS=.
47 00060 000367 247 ; AC1: LAST IN FREE CORE
48 ;DUSR DELAY=.
49 00061 000047 DFIRS ; AC2: DELAY QUEUE
50 00062 000062 JMP . ; AC3: PC.MONITOR: JMP .+0
51 00063 000062 .-1 ; PSW: PSW.MONITOR: PC.MONITOR
52 ;DUSR AREAP=. ; HEAD OF AREA PROCESS CHAIN
53 00064 000063 AFIRS=CHAIN ;
54 ;DUSR AFIRS=. ; FIRST IN AREA PROCESS CHAIN
55 00065 000000 0
56
57
58

```

```
01          ; PAGE ZERO VARIABLES:
02
03          ;DUSR FREQU=.      ; FREQUENCY OF RTC:
04 00066 000000      0      ; 0: 50 HZ, 1: 10 HZ, 2: 100 HZ, 3: 1000 HZ
05          ;DUSR MASK=.      ;
06 00067 000000      INTER ; INTERRUPT MASK
07          ;DUSR CORES=.     ;
08 00070 000000      0      ; CORE SIZE
09          ;DUSR PROGR=.     ;
10 00071 000050      PFIRS=CHAIN ; REFERENCE TO HEAD OF PROGRAM CHAIN
11 00072 000743'     A360      ; CLINT: CLEARDEVICE
12 00073 000155'     A2       ; CLINT.RTC: REAL TIME CLOCK
13          ;DUSR RTIME=.     ; REAL TIME COUNT (DOUBLE WORD)
14 00074 000000      0      ;
15 00075 000000      0      ;
16          ;DUSR POWIN=.     ; POWER INTERRUPT COUNT
17 00076 000000      0      ;
18          ;DUSR CDUMP=.     ; CORE DUMP ENTRY
19 00077 000000      0      ; NOT USED
20 00100 000000      0      ;
```

```
21
22          ; PAGE ZERO CONSTANTS:
```

```
23
24 00101 100000 B1B0: 1B0
25 00102 040000 B1B1: 1B1
26 00103 020000 B1B2: 1B2
27 00104 010000 B1B3: 1B3
28 00105 004000 B1B4: 1B4
29 00106 002000 B1B5: 1B5
30 00107 001000 B1B6: 1B6
31 00110 000400 B1B7: 1B7
32 00111 000200 B1B8: 1B8
33 00112 000100 B1B9: 1B9
34 00113 000040 B1B10: 1B10
35 00114 000020 B1B11: 1B11
36 00115 000010 B1B12: 1B12
37 00116 000004 B1B13: 1B13
38 00117 000002 B1B14: 1B14
39 00120 000001 B1B15: 1B15
40
```

10006 MUM12

01
02 00121 000003 C3: 3
03 00122 000005 C5: 5
04 00123 000006 C6: 6
05 00124 000007 C7: 7
06 00125 000011 C9: 9
07 00126 000012 C10: 10
08 00127 000014 C12: 12
09 00130 000015 C13: 13
10 00131 000017 C15: 15
11 00132 000030 C24: 24
12 00133 000031 C25: 25
13 00134 000050 C40: 40
14 00135 000060 C48: 48
15 00136 000070 C56: 56
16 00137 000074 C60: 60
17 00140 000077 C63: 63
18 00141 000170 C120: 120
19 00142 000177 C127: 127
20 00143 000377 C255: 255
21 00144 177775 CM3: -3
22 00145 177774 CM4: -4
23 00146 177760 CM16: -16
24 00147 177400 CM256: -256
25
26
27

28 00150 000032 Z ; OPTIONAL.ZONE

01 ; REFERENCES:

02

03 00151 000132'R00: A00 ; INTERRUPT ACCEPT
04 00152 000212'R6: A6 ; POWER RESTART POINT
05 00153 000241'R100: A100 ; REMOVE PROCESS(CUR,STATE)
06 00154 000242'R10: A10 ; REMOVE PROCESS(STATE)
07 00155 000243'R11: A11 ; REMOVE(ELEM)
08 00156 000253'R12: A12 ; LINK PROCESS(PROC)
09 00157 000265'R13: A13 ; LINK(HEAD,ELEM)
10 00160 000275'R14: A14 ; RECHAIN(OLD,NEW,ELEM)
11 00161 000313'R15: A15 ; SEARCH(CHAIN,NAME ADDR,ITEM)
12 00162 000337'R160: A160 ; SETBUF(CUR,BUF)
13 00163 000354'R17: A17 ; DELIVER ANSWER(BUF)

14

15 ; REFERENCES TO REENTRANT SYSTEM UTILITY PROCEDURES:

16

17 00164 000000 R34: 0 ; NEXT OPERATION(MODE,COUNT,BUF)
18 00165 000000 R35: 0 ; RETURN ANSWER(STATUS)
19 00166 000746'R36: A36 ; CLEAR(DEVICE)
20 00167 000000 R340: 0 ; WAIT OPERATION
21 00170 000754'R361: A361 ; SETINTERRUPT(DEVICE)
22 00171 000000 R37: 0 ; SET RESERVATION(MODE)
23 00172 000000 R38: 0 ; SET CONVERSION(MODE)
24 00173 000000 R39: 0 ; CONBYTE(BYTE)
25 00174 000000 R40: 0 ; GETBYTE(ADDR,BYTE)
26 00175 000000 R41: 0 ; PUTBYTE(ADDR,BYTE)
27 00176 000000 R42: 0 ; MULTIPLY(OP1,OP2,RESULT)
28 00177 000000 R43: 0 ; DIVIDE(DIVIDEND,DIVISOR,QUOTIENT,REMAINDER)

29

```

01          ; REFERENCES TO REENTRANT INPUT/OUTPUT UTILITY PROCEDURES:
02
03 00200 000000 R50: 0      ; GETREC(ZONE,BYTES,ADDR,SPAN)
04 00201 000000 R51: 0      ; PUTREC(ZONE,BYTES,ADDR,SPAN)
05 00202 000000 R52: 0      ; WAIT TRANSFER(ZONE)
06 00203 000000 R528: 0     ; REPEATSHARE(ZONE)
07 00204 000000 R53: 0      ; TRANSFER(ZONE,OP,LENGTH)
08 00205 000000 R54: 0      ; INBLOCK(ZONE)
09 00206 000000 R55: 0      ; OUTBLOCK(ZONE)
10 00207 000000 R56: 0      ; INCHAR(ZONE,CHAR)
11 00210 000000 R57: 0      ; FREESHARE(ZONE)
12 00211 000000 R580: 0     ; OUTSPACE(ZONE)
13 00212 000000 R58: 0      ; OUTCHAR(ZONE,CHAR)
14 00213 000000 R590: 0     ; OUTNL(ZONE)
15 00214 000000 R59: 0      ; OUTEND(ZONE,CHAR)
16 00215 000000 R60: 0      ; OUTTEXT(ZONE,ADDR)
17 00216 000000 R61: 0      ; OUTOCTAL(ZONE,VALUE)
18 00217 000000 R70: 0      ; SETPOSITION(ZONE,FILE,BLOCK)
19 00220 000000 R71: 0      ; CLOSE(ZONE,RELEASE)
20 00221 000000 R72: 0      ; OPEN(ZONE,OP)
21 00222 000000 R700: 0     ; WAITZONE(ZONE)
22 00223 000000 R82: 0      ; INNAME(ZONE,NAMEADDR);
23 00224 000000 R83: 0      ; MOVE(PARAMADDR)
24 00225 000000 R84: 0      ; INTPRETE
25 00226 002227 R85: JMP# .+1; INTGIVEUP
26 00227 000000          0      ;
27 00230 002231 R86: JMP# .+1; INTBREAK
28 00231 000000          0      ;
29 00232 000000 R90: 0      ; BINDEC(WORD,ADDR,CUR);
30 00233 000000 R91: 0      ; DECBIN(ADDR,WORD,CUR);
31
32
33          ;DUSR MZSTART=.          ; INTERPRETER PAGE ZERO START:
34 00234 000100 .BLK 64          ; RESERVED FOR INTERPRETER
35
36          ; COROUTINE MONITOR ENTRIES
37          ;
38 00334 000000 CR0: 0      ; DELAY(TIME);
39 00335 000000 CR1: 0      ; WAITSEM(SEMAPHORE);
40 00336 000000 CR2: 0      ; WAITCHAINED(SEMAPHORE);
41 00337 000000 CR3: 0      ; WAITANSWER(BUF);
42 00340 000000 CR4: 0      ; TESTOUT REGISTERS;
43 00341 000000 CR5: 0      ; TESTOUT(RECORD);
44 00342 000000 CR6: 0      ; TESTGENERAL;
45 00343 000000 CR7: 0      ; SIGNAL(SEMAPHORE);
46 00344 000000 CR8: 0      ; SIGNAL CHAINED(SEMAPHORE);
47 00345 000000 CR9: 0      ; PASS;

```



```
01          ; FILE SYSTEM ENTRIES
02 00346 000000 FS0: 0      ; CREATEENTRY(ZONE,TYPE,SIZE);
03 00347 000000 FS1: 0      ; LOOKUPENTRY(ZONE,ADDRESS);
04 00350 000000 FS2: 0      ; CHANGEENTRY(ZONE,ADDRESS);
05 00351 000000 FS3: 0      ; REMOVE ENTRY(ZONE);
06 00352 000000 FS4: 0      ; INIT CATALOG(ZONE,INITKIND,DRIVENO);
07 00353 000000 FS5: 0      ; SETENTRY(ZONE,ADDRESS);
08
09
10          ; PAGE SYSTEM ENTRIES
11 00354 000000 PS0: 0      ; COROUTINE MONITOR CALL(COROUTINECALL)
12 00355 000000 PS1: 0      ; CALL(POINT)
13 00356 000000 PS2: 0      ; GOTO(POINT)
14 00357 000000 PS3: 0      ; GETADR(POINT)
15 00360 000000 PS4: 0      ; GETPOINT(ADR)
16
17
18          ; HEAD OF CORE CHAIN
19          ;DUSR CORE      = .
20 00361 000000          0
21          ;DUSR COMLIST = .      ; ADDR OF PROC XCOMX
22 00362 000000          0
23          ;DUSR COMNO   = .
24 00363 000000          0
25
26
27          ; EXTENDED COROUTINE MONITOR ENTRIES
28          ;
29
30 00364 000000 CR10: 0      ; CSENDMESSAGE
31 00365 000000 CR11: 0      ; SIGNAL GENEREL
32 00366 000000 CR12: 0      ; WAITGENEREL
33 00367 000000 CR13: 0      ; COROUTINE TESTOUTPUT FUNCTION
```

10010 MUM12

```
01      000376 .LOC 254      ;
02 00376 063530      SKPBZ      24      ; WHILE(BUSY MT0) DO;
03 00377 000376      JMP      .-1      ; GOTO SYSTEM START;
04 00400 002002      JMP@      2      ;
05 00401 000000      0      ;
06
07      ; DEVICE TABLE:
08      ; THIS TABLE IS USED TO SELECT THE PROPER PROCESS DESCRIPTOR
09      ; WHEN A DEVICE ISSUES AN INTERRUPT. THE TABLE IS CLEARED BY
10      ; THE MONITOR INITIALIZATION.
11
12      .NREL
13      ;DUSR DEVTA=256-8      ; DEVICE TABLE:
14      ;DUSR TOPDE=DEVTA+63-3      ; TOP OF DEVICE TABLE:
15 00000'000067 .BLK 63-8      ;
16
17      ; ***** END OF MONITOR TABLE *****
18
19
20
```

```

01 ; ***** INTERRUPT RESPONSE *****
02 ; THE ACCUMULATORS ETC ARE SAVED IN THE PROCESS DESCRIPTOR
03 ; OF THE CURRENT PROCESS.
04
05 00067'054016 A0: STA 3 WORK ; INTERRUPT RESPONSE:
06 00070'034040 LDA 3 CUR ;
07 00071'041417 STA 0 AC0,3 ; AC0.CUR:= AC0;
08 00072'045420 STA 1 AC1,3 ; AC1.CUR:= AC1;
09 00073'051421 STA 2 AC2,3 ; AC2.CUR:= AC2;
10 00074'020000 LDA 0 0 ;
11 00075'041423 STA 0 PSW,3 ; PSW.CUR:= WORD(0);
12 00076'101100 MOVL 0,0 ;
13 00077'041403 STA 0 SIZE,3 ; SIZE.CUR(15):=CARRY;
14 00100'020016 LDA 0 WORK ;
15 00101'041422 STA 0 AC3,3 ; AC3.CUR:= AC3;
16
17 ; GET THE DEVICE NUMBER OF THE NEAREST DEVICE ON THE BUS
18 ; WHICH IS ISSUING AN INTERRUPT, AND SELECT A PROCESS
19 ; DESCRIPTOR FROM THE DEVICE TABLE.
20 00102'063777 SKPDZ CPU ; IF POWER FAULT THEN
21 00103'000500 JMP AS ; GOTO POWER FAILURE;
22 00104'065477 INTA 1 ; DEV:= INTERRUPTING DEVICE;
23 00105'034045 LDA 3 TABLE ;
24 00106'137000 ADD 1,3 ;
25 00107'031400 LDA 2 +0,3 ; PROC:= DEVICE TABLE(DEV);
26 00110'151220 MOVZR 2,2 ; PROC:= PROC SHIFT -1;
27 00111'141140 MOVOL 2,0 ; DEVTA(DEV):= PROC SHIFT 1+1;
28 00112'041400 STA 0 +0,3 ;
29 ; AC1=DEVICE AC2=PROC MUST NOT BE DESTROYED BY CLINT.PROC
30 00113'035032 LDA 3 CLINT,2 ; ENTRY:=PROC.CLINT;
31 00114'175133 A01: MOVZL# 3,3 SNC ; WHILE ENTRY < 0 DO
32 00115'000405 JMP A001 ; BEGIN
33 00116'175120 MOVZL 3,3 ;
34 00117'175220 MOVZR 3,3 ; ENTRY:=ENTRY EXTRACT 15;
35 00120'035400 LDA 3 +0,3 ; ENTRY:=WORD(ENTRY);
36 00121'000773 JMP A01 ; END;
37 00122'005400 A001: JSR +0,3 ; CALL(ENTRY);
38 00123'021013 LDA 0 STATE,2 ;
39 00124'107014 ADD# 0,1 SZR ; IF DEV=
40 00125'106415 SUB# 0,1 SNR ; ABS(STATE.PROC) THEN
41 00126'000402 JMP +2 ; BEGIN
42 00127'000407 JMP A1 ;
43 00130'034045 LDA 3 TABLE ;
44 00131'137000 ADD 1,3 ; INTERRUPT ACCEPT:
45 00132'015400 A00: DSZ +0,3 ; DECR(DEVTA(DEV));
46 00133'011023 ISZ PSW,2 ; PSW.PROC:= PSW.PROC+1;
47 00134'006503 JSR# .A11 ; REMOVE(PROC);
48 00135'006503 JSR# .A12 ; LINK PROCESS(PROC);
49 ; END
50 ; THIS IS THE STANDARD MONITOR EXIT. THE ACCUMULATORS ETC
51 ; ARE RESTORED AND THE INTERRUPT SYSTEM IS ENABLED AGAIN.
52 00136'102400 A1: SUB 0,0 ; INTERRUPT RETURN:
53 00137'040000 STA 0 0 ; CATCH JUMP TO ZERO;
54 00140'034040 LDA 3 CUR ;
55 00141'021441 LDA 0 CCOROUT,3 ;
56 00142'040017 STA 0 COROUT ; COROUT:=NONSENSE IN MOST CASES
57 00143'021403 LDA 0 SIZE,3 ; CARRY:= SIZE.CUR(15);
58 00144'101220 MOVZR 0,0 ;
59 00145'021423 LDA 0 PSW,3 ; RETURNADDR:=PSW.CUR;
60 00146'040016 STA 0 WORK ;

```

0012 MUM12

```
01 00147'021417   LDA    0    AC0,3   ;   AC0:= AC0.CUR;
02 00150'025420   LDA    1    AC1,3   ;   AC1:= AC1.CUR;
03 00151'031421   LDA    2    AC2,3   ;   AC2:= AC2.CUR;
04 00152'035422   LDA    3    AC3,3   ;   AC3:= AC3.CUR;
05 00153'060177   INTEN                ;   INTERRUPT ENABLE;
06 00154'002016   JMP@    WORK        ;   RETURN;
```

```

01      ; AT EACH TIMER INTERRUPT CURRENT PROCESS IS RELINKED AS LAST
02      ; PROCESS WITHIN ITS PRIORITY
03      ; TIMER COUNT IS DECREASED BY ONE FOR ALL DELAYED PROCESSES.
04      ; WHEN TIMER COUNT FOR A PROCESS BECOMES ZERO, IT IS STARTED
05
06      000014 RTC= 12
07
08 00155'010075 A2:  ISZ      RTIME+1  ; REAL TIME CLOCK:
09 00156'000403      JMP      .+3      ;
10 00157'010074      ISZ      RTIME+0  ; RTIME:=RTIME+1;
11 00160'000401      JMP      .+1      ;
12 00161'020066      LDA      0  FREQUENCY ;
13 00162'061114      DOAS    0  RTC      ; START(RTC,FREQUENCY);
14 00163'030040      LDA      2  CUR      ; PROC:=CUR;
15 00164'006453      JSR@    .A11     ; REMOVE(PROC);
16 00165'156414      SUB#    2,3  SZR      ; IF PROC<>NEXT.PROC THEN
17 00166'006452      JSR@    .A12     ; LINK PROCESS(PROC);
18 00167'030047      LDA      2  DFIRST  ; PROC:= FIRST IN DELAY QUEUE;
19 00170'021000 A3:  LDA      0  NEXT,2  ; NEXT DELAYED:
20 00171'040016      STA      0  WORK    ; NEXT:= NEXT.PROC;
21 00172'024061      LDA      1  DELAY   ;
22 00173'146415      IEQ      2,1    ; IF PROC=DELAY QUEUE HEAD THEN
23 00174'002056      JMP@    EXIT     ; EXIT;
24 00175'015014      DSZ      TIMER,2  ; IF DECR(TIMER COUNT.PROC)<>0 THEN
25 00176'000403      JMP      A4      ; GOTO GET NEXT;
26 00177'006440      JSR@    .A11     ; REMOVE(PROC);
27 00200'006440      JSR@    .A12     ; LINK PROCESS(PROC);
28      A4:          ; GET NEXT:
29 00201'030016      LDA      2  WORK    ; PROC:= NEXT;
30 00202'000766      JMP      A3      ; GOTO NEXT DELAYED;
31
32      ; AT POWER FAULT THIS CODE WILL BE ENTERED
33
34 00203'020432 A5:  LDA      0  A8      ; POWER FAILURE:
35 00204'040000      STA      0  0      ; WORD(0):=JMP@ POWER RESTART
36 00205'010076      ISZ      POWINT  ; INCR(POWER INTERRUPT COUNT);
37 00206'063601      SKPDN   1      ; IF MEMORY EXTENSION OFF THEN
38 00207'102400      SUB      0,0    ; WORK:=0
39 00210'040016      STA      0  WORK    ; ELSE WORK:=(A8);
40 00211'063077      HALT     ; HALT;
41
42 00212'062677 A6:  IORST    ; POWER RESTART:
43 00213'020016      LDA      0  WORK    ; MEM. EXT. FLAG BEFORE POWER FALIUR
44 00214'101014      MOV#    0,0  SZR      ; IF MEM.EXT. WAS ON THEN
45 00215'062701      DICP    0,1    ; SET MEM.EXT.ON AGAIN
46 00216'020420      LDA      0  A9      ; INDEX:=TOPTABLE-TABLE+1;
47 00217'040016      STA      0  WORK    ;
48 00220'034045 A7:  LDA      3  TABLE  ; REPEAT
49 00221'024016      LDA      1  WORK    ;
50 00222'137000      ADD      1,3    ;
51 00223'031402      LDA      2  2,3    ; PROC:=DEVTA(INDEX-1)//2;
52 00224'024147      LDA      1  .M256  ;
53 00225'151222      MOVZR   2,2  SZC      ; IF INTERRUPT THEN
54 00226'015402      DSZ      2,3    ; DECR(DEVTA(INDEX-1));
55 00227'020145      LDA      0  .M4    ;
56 00230'147414      AND#    2,1  SZR      ; IF -,PROC IN PAGE ZERO THEN
57 00231'006051      JSR@    R301   ; BREAK IT(-4,PROC);
58 00232'014016      DSZ      WORK    ; INDEX:=INDEX-1
59 00233'000765      JMP      A7      ; UNTIL INDEX=0;
60 00234'000721      JMP      A2      ; GOTO REAL TIME CLOCK;

```

```
0014 MUM12
01 00235'002152 A8:  JMP#      R6      ; CONSTANT;
02 00236'000074 A9:   60
03 00237'000243'.A11: A11
04 00240'000253'.A12: A12
05
06          ;  *****  END OF INTERRUPT RESPONSE  *****
07
08
09
```

```

01      ; ***** MONITOR UTILITY PROCEDURES *****
02
03
04      ; PROCEDURE REMOVE PROCESS(PROC,STATE);
05      ; REMOVES THE PROCESS FROM A QUEUE AND SETS ITS STATE.
06      ;      CALL:      RETURN:
07      ; AC0      STATE      PREV.ELEM
08      ; AC1      UNCHANGED
09      ; AC2      PROC      PROC
10      ; AC3      LINK      NEXT.ELEM
11
12 00241'030040 A100: LDA      2      CUR      ; PROC:= CUR;
13
14      A10:
15 00242'041013      STA      0      STATE,2 ; REMOVE PROCESS:
16      ; STATE.PROC:= STATE;
17      ; REMOVE(PROC);
18      ; RETURN;
19
20      ; PROCEDURE REMOVE(ELEM);
21      ; REMOVES A GIVEN ELEMENT FROM A QUEUE.
22      ;      CALL:      RETURN:
23      ; AC0      PREV.ELEM
24      ; AC1      UNCHANGED
25      ; AC2      ELEM      ELEM
26      ; AC3      LINK      NEXT.ELEM
27 00243'054000 A11:  STA      3      0      ; REMOVE:
28 00244'035000      LDA      3      NEXT,2 ;
29 00245'057001      STA@   3      PREV,2 ; NEXT.PREV.ELEM:= NEXT.ELEM;
30 00246'021001      LDA      0      PREV,2 ;
31 00247'041401      STA      0      PREV,3 ; PREV.NEXT.ELEM:= PREV.ELEM;
32 00250'051000      STA      2      NEXT,2 ; NEXT.ELEM:= ELEM;
33 00251'051001      STA      2      PREV,2 ; PREV.ELEM:= ELEM;
34 00252'002000      JMP@    0      ; RETURN;
35
36      ; PROCEDURE LINK PROCESS(PROC);
37      ; LINKS A PROCESS TO THE RUNNING QUEUE AS THE LAST PROCESS
38      ; AMONG PROCESSES OF SAME PRIORITY.
39      ;      CALL:      RETURN:
40      ; AC0      DESTROYED
41      ; AC1      DESTROYED
42      ; AC2      PROC      PROC
43      ; AC3      LINK      DESTROYED
44
45 00253'054000 A12:  STA      3      0      ; LINK PROCESS:
46 00254'126400      SUB      1,1 ;
47 00255'045013      STA      1      STATE,2 ; STATE.PROC:= RUNNING(=0);
48 00256'034054      LDA      3      RUNNING ; HEAD:= RUNNING QUEUE;
49 00257'021015      LDA      0      PRIORITY,2;
50      A120: ; NEXT PRIORITY:
51 00260'035400      LDA      3      NEXT,3 ; HEAD:= NEXT.HEAD;
52 00261'025415      LDA      1      PRIORITY,3; IF PRIORITY.PROC
53 00262'106432      ING      0,1 ; <=PRIORITY.HEAD THEN
54 00263'000775      JMP      A120 ; GOTO NEXT PRIORITY;
55 00264'000403      JMP      A130 ; LINK(HEAD,PROC);
56      ; RETURN;

```

```

01      ; PROCEDURE LINK(HEAD,ELEM);
02      ; LINKS A GIVEN ELEMENT TO THE END OF A QUEUE.
03      ;      CALL:      RETURN:
04      ; AC0      PREV,HEAD
05      ; AC1      HEAD      HEAD
06      ; AC2      ELEM      ELEM
07      ; AC3      LINK      HEAD
08
09 00265'054000 A13:  STA      3      0      ; LINK:
10 00266'135000      MOV      1,3      ;
11 00267'021401 A130: LDA      0      PREV,3  ; OLD PREV:= PREV.HEAD;
12 00270'051401      STA      2      PREV,3  ; PREV.HEAD:= ELEM;
13 00271'055000      STA      3      NEXT,2  ; NEXT.ELEM:= HEAD;
14 00272'041001      STA      0      PREV,2  ; PREV.ELEM:= OLD PREV;
15 00273'053001      STA@    2      PREV,2  ; NEXT.PREV.ELEM:= ELEM;
16 00274'002000      JMP@    0      ; RETURN;
17
18      ; PROCEDURE RECHAIN(OLD,NEW,ELEM);
19      ; EXCLUDES THE ELEMENT FROM THE OLD CHAIN AND INCLUDES IT
20      ; IN THE NEW CHAIN.
21      ;      CALL:      RETURN:
22      ; AC0      OLD      CHAIN,NEW
23      ; AC1      NEW      NEW
24      ; AC2      ELEM      ELEM
25      ; AC3      LINK      CUR
26
27 00275'054000 A14:  STA      3      0      ; RECHAIN:
28 00276'115000 A141: MOV      0,3      ; HEAD:= OLD;
29 00277'021402      LDA      0      CHAIN,3  ; OLD:= CHAIN.OLD;
30 00300'101005      MOV      0,0      SNR      ; IF OLD=0 THEN
31 00301'000543      JMP      A213      ; GOTO MONITOR ERROR 3;
32 00302'112414      INE      0,2      ; IF OLD<>ELEM THEN
33 00303'000773      JMP      A141      ; GOTO RECHAIN;
34 00304'021002      LDA      0      CHAIN,2  ;
35 00305'041402      STA      0      CHAIN,3  ; CHAIN.HEAD:= CHAIN.ELEM;
36 00306'135000      MOV      1,3      ;
37 00307'021402      LDA      0      CHAIN,3  ;
38 00310'041002      STA      0      CHAIN,2  ; CHAIN.ELEM:= CHAIN.NEW;
39 00311'051402      STA      2      CHAIN,3  ; CHAIN.NEW:= ELEM;
40 00312'000423      JMP      A152      ; RETURN;
41
42
43

```



```

01 ; PROCEDURE SEARCH(CHAIN,NAME ADDR,ITEM);
02 ; SEARCHES THE CHAIN FOR AN ITEM WITH A GIVEN NAME AND
03 ; DELIVERS IT IF PRESENT, AND A ZERO IF THE NAME IS NOT
04 ; FOUND IN THE CHAIN.
05 ; CALL: RETURN:
06 ; AC0 DESTROYED
07 ; AC1 CHAIN DESTROYED
08 ; AC2 NAME ADDR ITEM
09 ; AC3 LINK CUR
10
11 00313'054000 A15: STA 3 0 ; SEARCH:
12 00314'135000 MOV 1,3 ; ITEM:= CHAIN;
13 A150: ; NEXT ITEM:
14 00315'035402 LDA 3 CHAIN,3 ; ITEM:= CHAIN.ITEM;
15 00316'175005 MOV 3,3 SNR ; IF ITEM=0 THEN
16 00317'000415 JMP A151 ; RETURN;
17 00320'021404 LDA 0 +0+NAME,3 ;
18 00321'025000 LDA 1 +0,2 ;
19 00322'106414 INE 0,1 ; IF 0.NAME.ITEM<>0.NAME ADDR THEN
20 00323'000772 JMP A150 ; GOTO NEXT ITEM;
21 00324'021405 LDA 0 +1+NAME,3 ;
22 00325'025001 LDA 1 +1,2 ;
23 00326'106414 INE 0,1 ; IF 1.NAME.ITEM<>1.NAME ADDR THEN
24 00327'000766 JMP A150 ; GOTO NEXT ITEM;
25 00330'021406 LDA 0 +2+NAME,3 ;
26 00331'025002 LDA 1 +2,2 ;
27 00332'106414 INE 0,1 ; IF 2.NAME.ITEM<>2.NAME ADDR THEN
28 00333'000762 JMP A150 ; GOTO NEXT ITEM;
29 00334'171000 A151: MOV 3,2 ;
30 00335'034040 A152: LDA 3 CUR ;
31 00336'002000 JMP# 0 ; RETURN;
32
33 ; PROCEDURE SETBUF(PROC,BUF);
34 ; SAVES BUF AND THE TWO FIRST BUFFER WORDS IN THE GIVEN
35 ; PROCESS DESCRIPTOR.
36 ; CALL: RETURN:
37 ; AC0 DESTROYED
38 ; AC1 PROC PROC
39 ; AC2 BUF BUF
40 ; AC3 LINK PROC
41
42 00337'024040 A160: LDA 1 CUR ; PROC:= CUR;
43
44 00340'054000 A16: STA 3 0 ; SETBUF:
45 00341'135000 MOV 1,3 ;
46 00342'051421 STA 2 AC2,3 ; AC2.PROC:= BUF;
47 00343'021006 LDA 0 MESS0,2 ;
48 00344'041417 STA 0 AC0,3 ; AC0.PROC:= MESS0.BUF;
49 00345'021007 LDA 0 MESS1,2 ;
50 00346'041420 STA 0 AC1,3 ; AC1.PROC:= MESS1.BUF;
51 00347'021005 LDA 0 RECEIVER,2;
52 00350'100503 NEGL 0,0 SNC ; IF RECEIVER.BUF<=0 THEN
53 00351'002000 JMP# 0 ; RETURN;
54 00352'011423 ISZ PSW,3 ; PSW.PROC:= PSW.PROC+1;
55 00353'002000 JMP# 0 ; RETURN;
56
57
58

```

```

01 ; PROCEDURE DELIVER ANSWER(BUF);
02 ; DELIVERS AN ANSWER FROM A RECEIVER AND STARTS THE SENDER
03 ; IF IT IS WAITING FOR AN ANSWER OR AN EVENT.
04 ; CALL: RETURN:
05 ; AC0 DESTROYED
06 ; AC1 DESTROYED
07 ; AC2 BUF DESTROYED
08 ; AC3 LINK DESTROYED
09
10 00354'054017 A17: STA 3 LINK ; DELIVER ANSWER:
11 00355'004666 JSR A11 ; REMOVE(BUF);
12 00356'035004 LDA 3 SENDER,2 ; PROC:= SENDER,BUF;
13 00357'054016 STA 3 WORK ; SAVE(PROC);
14 00360'021005 LDA 0 RECEIVER,2;
15 00361'100400 NEG 0,0 ; RECEIVER,BUF:=
16 00362'041005 STA 0 RECEIVER,2; -RECEIVER,BUF;
17 00363'021413 LDA 0 STATE,3 ; COMMENT: ANSWER NOW PENDING;
18 00364'142404 SUB 2,0 SZR ; IF STATE.PROC<>BUF THEN
19 00365'000407 JMP A171 ; GOTO DELIVER EVENT;
20 ; RELEASE BUFFER:
21 00366'041005 STA 0 RECEIVER,2; RECEIVER,BUF:= 0;
22
23 A170: ; START PROC:
24 00367'024016 LDA 1 WORK ; RESTORE(PROC);
25 00370'004750 JSR A16 ; SETBUF(PROC,BUF);
26 00371'131000 MOV 1,2 ;
27 00372'034017 LDA 3 LINK ; LINK PROCESS(PROC);
28 00373'000660 JMP A12 ; RETURN;
29
30 A171: ; DELIVER EVENT:
31 00374'024016 LDA 1 WORK ; RESTORE(PROC);
32 00375'020124 LDA 0 ,EVENT ; HEAD:= EVENT QUEUE HEAD,PROC;
33 00376'107000 ADD 0,1 ;
34 00377'004666 JSR A13 ; LINK(HEAD,BUF);
35 00400'034016 LDA 3 WORK ; RESTORE(PROC);
36 00401'021413 LDA 0 STATE,3 ;
37 00402'101415 INC# 0,0 SNR ; IF STATE.PROC=-1 THEN
38 00403'000764 JMP A170 ; GOTO START PROC;
39 00404'101127 MOVZL 0,0 SBN ; IF STATE.PROC>=2
40 ; OR STATE.PROC=1B0 THEN
41 00405'002017 JMP# LINK ; RETURN;
42 00406'025423 A172: LDA 1 PSW,3 ; EVENT AFTER WAIT:
43 00407'020117 LDA 0 ,2 ;
44 00410'107000 ADD 0,1 ; PSW,PROC:= PSW.PROC+2;
45 00411'045423 STA 1 PSW,3 ;
46 00412'145000 MOV 2,1 ;
47 00413'171000 MOV 3,2 ;
48 00414'004627 JSR A11 ; REMOVE(PROC);
49 00415'131000 MOV 1,2 ;
50 00416'000751 JMP A170 ; GOTO START PROC;
51
52 ; ***** END OF MONITOR UTILITY PROCEDURES *****
53
54
55

```

```

01 ; ***** MONITOR FUNCTIONS *****
02
03 ; MONITOR FUNCTION ENTRY:
04 ; ALL MONITOR FUNCTIONS ARE HANDLED HERE AFTER THE INITIAL
05 ; CALL.
06
07 A20: ; MONITOR FUNCTION ENTRY:
08 00417'060277 INTDS ; INTERRUPT DISABLE;
09 00420'054000 STA 3 0 ; WORD(0):= LINK;
10 00421'034040 LDA 3 CUR ;
11 00422'041417 STA 0 AC0,3 ; AC0.CUR:= AC0;
12 00423'045420 STA 1 AC1,3 ; AC1.CUR:= AC1;
13 00424'051421 STA 2 AC2,3 ; AC2.CUR:= AC2;
14 00425'055422 STA 3 AC3,3 ; AC3.CUR:= CUR;
15
16 00426'030000 LDA 2 0 ; UNPACK CALL:
17 00427'145100 MOVL 2,1 ;
18 00430'045403 STA 1 SIZE,3 ; SIZE.CUR(15):=CARRY;
19 00431'051423 STA 2 PSW,3 ; PSW.CUR:= LINK;
20 00432'031377 LDA 2 -1,2 ; CALL:= (-1).LINK;
21 00433'024131 LDA 1 .15 ;
22 00434'133400 AND 1,2 ; INDEX:= CALL(12:15);
23 00435'025020 LDA 1 FUNCTION,2;
24 00436'044016 STA 1 WORK ;
25 00437'024124 LDA 1 .EVENT ;
26 00440'167000 ADD 3,1 ; AC1:= EVENT QUEUE HEAD.CUR;
27 00441'031421 LDA 2 AC2,3 ; AC2:= AC2.CUR;
28 00442'002016 JMP# WORK ; GOTO FUNCTION(INDEX);
29
30 ; WHEN THE CODE TO HANDLE THE FUNCTION IS ENTERED WE HAVE:
31 ;
32 ; AC0 AC0.CUR
33 ; AC1 EVENT QUEUE HEAD.CUR
34 ; AC2 AC2.CUR
35 ; AC3 CUR
36
37 ; MONITOR CALL ERRORS:
38 ; GENERALLY THE ERROR NUMBERS ARE NEGATIVE AND ARE GENERATED
39 ; VIA THE FOLLOWING ENTRIES. AT ERROR RETURN WE HAVE:
40 ; SAC0 ERR NO
41 ; SAC1 DESTROYED
42 ; SAC2 CUR
43 ; SAC3 LINK
44
45 00443'152401 A212: SUB 2,2 SKP ; ERROR(-2):
46 00444'152520 A213: SUBZL 2,2 ; ERROR(-3):
47 00445'151400 INC 2,2 ;
48 00446'150000 COM 2,2 ;
49 00447'034040 LDA 3 CUR ; PROC:= CUR;
50 00450'051421 STA 2 AC2,3 ; AC2.PROC:=ERR NO
51 00451'002056 JMP# EXIT ; GOTO EXIT;
52
53
54

```

```

01 ; FUNCTION WAIT(Delay,Device,First,Second,Buf);
02 ; WORKS AS A COMBINATION OF THE FUNCTIONS, WAIT INTERRUPT
03 ; AND WAIT EVENT. THE RETURN PARAMETERS HAVE TWO POSSIBILITIES,
04 ; THE FIRST USED IN CASE OF TIME OUT OR INTERRUPT, THE SECOND
05 ; USED IN CASE OF AN EVENT.
06 ; CALL: RETURN: LINK
07 ; SAC0 DELAY DELAY OR FIRST +0: TIME OUT
08 ; SAC1 DEVICE DEVICE OR SECOND +1: INTERRUPT
09 ; SAC2 BUF CUR OR BUF +2: ANSWER
10 ; SAC3 LINK CUR +3: MESSAGE

```

```

11 ;
12 ; PRECONDITION:
13 ; BUF=0 OR BUF IN EVENT QUEUE OF THE CALLING PROCESS.
14 ; DEVICE MUST BE ZERO OR IN THE RANGE 8-62.
15 ;

```

```

16 A22: ; WAIT:
17 00452'151005 MOV 2,2 SNR ; IF BUF=0 THEN
18 00453'131000 MOV 1,2 ; BUF:= EVENT QUEUE HEAD.CUR;
19 00454'031000 LDA 2 NEXT,2 ; BUF:= NEXT.BUF;
20 00455'146415 IEQ 2,1 ; IF BUF=EVENT QUEUE HEAD.CUR THEN
21 00456'000405 JMP A220 ; GOTO SET STATE;
22 00457'054016 STA 3 WORK ; SAVE(CUR);
23 00460'024056 LDA 1 EXIT ;
24 00461'044017 STA 1 LINK ; LINK:= EXIT;
25 00462'000724 JMP A172 ; GOTO EVENT AFTER WAIT;
26 A220: ; SET STATE:
27 00463'111000 MOV 0,2 ; DELAY:= AC0.CUR;
28 00464'021420 LDA 0 AC1,3 ; STATE:=-DEVICE.CUR
29 00465'100405 NEG 0,0 SNR ; IF STATE=0 THEN
30 00466'102120 ADCZL 0,0 ; STATE:= -2
31 00467'000402 JMP A230 ;
32
33

```

```

01
02 ; FUNCTION WAIT INTERRUPT(DEVICE,DELAY);
03 ; CHECKS IF AN INTERRUPT IS PENDING. IF NOT THEN
04 ; THE PROCESS IS LINKED TO THE DELAY QUEUE. IF DEVICE IS ZERO
05 ; ONLY THE DELAY FUNCTION IS ACTIVE.
06 ; CALL: RETURN: LINK
07 ; SAC0 UNCHANGED +0: TIME OUT
08 ; SAC1 DEVICE DEVICE +1: INTERRUPT
09 ; SAC2 DELAY CUR
10 ; SAC3 LINK CUR
11
12 ; PRECONDITION:
13 ; DEVICE MUST BE ZERO OR IN THE RANGE 8-62.
14
15 00470'021420 A23: LDA 0 AC1,3 ; WAIT INTERRUPT:
16 00471'051414 A230: STA 2 TIMER,3 ; TIMER COUNT,CUR:= DELAY;
17 00472'055421 STA 3 AC2,3 ; AC2,CUR:= CUR;
18 00473'171000 MOV 3,2 ; PROC:=CUR;
19 00474'025420 LDA 1 AC1,3 ; DEV:= DEVICE.CUR;
20 00475'125005 MOV 1,1 SNR ; IF DEV<>0 THEN
21 00476'000406 JMP A231 ; BEGIN
22 00477'034045 LDA 3 TABLE ;
23 00500'137000 ADD 1,3 ;
24 00501'031400 LDA 2 +0,3 ; INT:=DEVTA(DEV);
25 00502'151222 MOVZR 2,2 SZC ; IF INT(15) THEN
26 00503'002151 JMP@ R00 ; GOTO INTERRUPT ACCEPT;
27 A231: ; END;
28
29
30 00504'006153 JSR@ R100 ; REMOVE PROCESS(CUR,STATE);
31 00505'024061 LDA 1 DELAY ; HEAD:= DELAY QUEUE HEAD;
32 00506'034056 LDA 3 EXIT ; LINK(HEAD,OLD);
33 00507'002157 JMP@ R13 ; EXIT;
34
35
36

```

```

01
02 ; FUNCTION SEND MESSAGE(ADDR,NAME ADDR,BUF);
03 ; SENDS A MESSAGE TO A GIVEN NAMED PROCESS.
04 ; CALL: RETURN:
05 ; SAC0 UNCHANGED
06 ; SAC1 ADDR ADDR
07 ; SAC2 NAME ADDR BUF
08 ; SAC3 LINK CUR
09
10 00510'034056 A24: LDA 3 EXIT ; SEND MESSAGE:
11 00511'054017 STA 3 LINK ; LINK:= EXIT;
12 00512'024054 LDA 1 PROCESS ; CHAIN:= PROCESS CHAIN;
13 00513'006161 JSR R15 ; SEARCH(CHAIN,NAME ADDR,PROC);
14 00514'145000 MOV 2,1 ;
15 00515'050016 STA 2 WORK ; SAVE(PROC);
16 00516'031411 LDA 2 BUFFER,3 ; BUF:= BUFFER.CUR;
17 00517'151005 A241: MOV 2,2 SNR ; REP:
18 00520'000724 JMP A213 ; IF BUF=0 THEN ERROR(-3);
19 00521'021005 LDA 0 RECEIVER,2;
20 00522'101005 MOV 0,0 SNR ; IF RECEIVER.BUF=0 THEN
21 00523'000403 JMP A242 ; GOTO SET MESSAGE;
22 00524'031002 LDA 2 CHAIN,2 ; BUF:=CHAIN.BUF;
23 00525'000772 JMP A241 ; GOTO REP;
24 A242: ; SET MESSAGE:
25 00526'125004 MOV 1,1 SZR ; IF PROC=0 THEN
26 00527'000450 JMP A246 ; BEGIN
27 00530'050016 STA 2 WORK ; SAVE(BUF);
28 00531'030363 LDA 2 COMNO ; IF NEW XCOMX THEN
29 00532'151404 INC 2,2 SZR ; BEGIN
30 00533'000410 JMP A2421 ;
31 00534'034362 LDA 3 COMLIST ; GET XCOMX ADDRESS;
32 00535'175015 MOV# 3,3 SNR ; IF XCOMX NOT LOADED THEN
33 00536'000705 JMP A212 ; GOTO MONITOR ERROR -2;
34 00537'165000 MOV 3,1 ;
35 00540'031777 LDA 2 -1,3 ;
36 00541'035776 LDA 3 -2,3 ; TABADD:=COMLIST;
37 00542'000410 JMP A243 ; END ELSE BEGIN
38 00543'024054 A2421:LDA 1 PROCESS ; CHAIN:=PROCESS CHAIN;
39 00544'030450 LDA 2 A248 ; NAME ADDR:=ADDR OF XCOMX NAME;
40 00545'006161 JSR R15 ; SEARCH(CHAIN,NAME,ADDR,PROC);
41 00546'145005 MOV 2,1 SNR ; IF PROC=0 THEN
42 00547'000674 JMP A212 ; GOTO NOMITOR ERROR -2;
43 00550'034362 LDA 3 COMLIST ;
44 00551'030363 LDA 2 COMNO ; TABADDR:=COMLIST;
45 ; END;
46 00552'150000 A243: COM 2,2 ; COUNT := -COMNO-1;
47 00553'151405 INC 2,2 SNR ; NEXT: IF INC(COUNT)=0 THEN
48 00554'000667 JMP A212 ; GOTO MONITOR ERROR -2;
49 00555'021400 LDA 0 +0,3 ; IF TAB(TABADD)<>0 THEN
50 00556'101005 MOV 0,0 SNR ; BEGIN
51 00557'000404 JMP A244 ;
52 00560'020116 LDA 0 .4 ;
53 00561'117000 ADD 0,3 ; TABADD:=TABADD+4;
54 00562'000770 JMP A243 ; GOTO NEXT
55 ; END;

```

```

01
02 00563'030040 A244: LDA      2   CUR      ;
03 00564'031021      LDA      2   AC2,2    ;
04 00565'021000      LDA      0   +0,2     ;
05 00566'041400      STA      0   +0,3     ;
06 00567'021001      LDA      0   +1,2     ;
07 00570'041401      STA      0   +1,3     ;      NAME(TABADD):=RECIEVERNAME;
08 00571'021002      LDA      0   +2,2     ;
09 00572'041402      STA      0   +2,3     ;
10 00573'030016      LDA      2   WORK     ;      RESTORE(BUF);
11 00574'051403      STA      2   +3,3     ;      BUF(TABADD):= BUF;
12 00575'044016      STA      1   WORK     ;      SAVE(PROC);
13 00576'034040      LDA      3   CUR      ;      END;
14
15
16
17 00577'051421 A246: STA      2   AC2,3    ;      AC2.CUR:=BUF;
18 00600'045005      STA      1   RECEIVER,2;      RECEIVER.BUF:= PROC;
19 00601'035420      LDA      3   AC1,3    ;
20 00602'021400      LDA      0   +0,3     ;
21 00603'041006      STA      0   MESS0,2   ;      MESS0.BUF:= 0.AC1.CUR;
22 00604'021401      LDA      0   +1,3     ;
23 00605'041007      STA      0   MESS1,2   ;      MESS1.BUF:= 1.AC1.CUR;
24 00606'021402      LDA      0   +2,3     ;
25 00607'041010      STA      0   MESS2,2   ;      MESS2.BUF:= 2.AC1.CUR;
26 00610'021403      LDA      0   +3,3     ;
27 00611'041011      STA      0   MESS3,2   ;      MESS3.BUF:= 3.AC1.CUR;
28 00612'002401      JMP@      .+1          ;      GOTO DELIVER EVENT;
29 00613'000374'      A171          ;
30
31 00614'000615' A248: .+1
32 00615'054103 .TXT  .XCOMX.
33      047515
34      054000
35
36      ; FUNCTION WAIT ANSWER(FIRST,SECOND,BUF);
37      ; WAITS FOR THE ANSWER TO A GIVEN MESSAGE.
38      ;      CALL:      RETURN:
39      ; SAC0      FIRST
40      ; SAC1      SECOND
41      ; SAC2      BUF
42      ; SAC3      LINK      CUR
43
44 00620'141000 A25:  MOV      2,0      ; WAIT ANSWER;
45 00621'006153      JSR@      R100      ; REMOVE PROCESS(CUR,BUF);
46 00622'031013      LDA      2   STATE,2 ; RESTORE(BUF);
47 00623'021005      LDA      0   RECEIVER,2;
48 00624'103057      ADDO#    0,0 SBN      ; IF RECEIVER.BUF<=0 THEN
49 00625'006163      JSR@      R17      ; DELIVER ANSWER(BUF);
50 00626'002056      JMP@      EXIT      ; EXIT;
51
52
53

```

```

01 ; FUNCTION WAIT EVENT(FIRST,SECOND,BUF);
02 ; DELAYS THE CALLING PROCESS UNTIL AN EVENT ARRIVES IN
03 ; ITS QUEUE FOLLOWING A GIVEN BUFFER.
04 ; CALL: RETURN: LINK
05 ; SAC0 FIRST +0: ANSWER
06 ; SAC1 SECOND +1: MESSAGE
07 ; SAC2 BUF BUF
08 ; SAC3 LINK CUR
09
10 ; PRECONDITION:
11 ; BUF=0 OR BUF IN EVENT QUEUE OF THE CALLING PROCESS.
12
13 A26: ; WAIT EVENT:
14 00627'151005 MOV 2,2 SNR ; IF BUF=0 THEN
15 00630'131000 MOV 1,2 ; BUF:= EVENT QUEUE HEAD.CUR;
16 00631'031000 LDA 2 NEXT,2 ; BUF:= NEXT.BUF;
17 00632'102000 ADC 0,0 ;
18 00633'034056 LDA 3 EXIT ; IF BUF=EVENT.CUR THEN
19 00634'146415 IEQ 2,1 ; REMOVE PROCESS(CUR,WAIT EVENT=-1
20 00635'002153 JMP@ R100 ; ELSE SETBUF(CUR,BUF);
21 00636'002162 JMP@ R160 ; EXIT;
22
23 ; FUNCTION SEND ANSWER(FIRST,SECOND,BUF);
24 ; DELIVERS AN ANSWER TO THE SENDER.
25 ; CALL: RETURN:
26 ; SAC0 FIRST FIRST
27 ; SAC1 SECOND SECOND
28 ; SAC2 BUF BUF
29 ; SAC3 LINK CUR
30
31 ; PRECONDITION:
32 ; BUF IN EVENT QUEUE OF THE CALLING PROCESS.
33
34 A27: ; SEND ANSWER:
35 00637'041006 STA 0 MESS0,2 ; MESS0.BUF:= AC0.CUR;
36 00640'021420 LDA 0 AC1,3 ;
37 00641'041007 STA 0 MESS1,2 ; MESS1.BUF:= AC1.CUR;
38 00642'034056 LDA 3 EXIT ; DELIVER ANSWER(BUF);
39 00643'002163 JMP@ R17 ; EXIT;
40
41 ; FUNCTION SEARCH ITEM(CHAIN,NAME ADDR,ITEM);
42 ; IF THE CHAIN CONTAINS AN ITEM WITH THE GIVEN NAME, THE ITEM
43 ; ADDRESS IS DELIVERED, OTHERWISE A ZERO IS DELIVERED.
44 ; CALL: RETURN:
45 ; SAC0 UNCHANGED
46 ; SAC1 CHAIN CHAIN
47 ; SAC2 NAME ADDR ITEM
48 ; SAC3 LINK CUR
49
50 00644'025420 A28: LDA 1 AC1,3 ; SEARCH ITEM:
51 00645'006161 JSR@ R15 ; SEARCH(CHAIN,NAME ADDR,ITEM);
52 00646'051421 STA 2 AC2,3 ; AC2.CUR:= ITEM;
53 00647'002056 JMP@ EXIT ; EXIT;
54
55
56

```



```

01      ; FUNCTION CLEAN PROCESS(PROC);
02      ; MESSAGES TO THE PROCESS ARE ANSWERED WITH STATUS = 0.
03      ; ANSWERS TO THE PROCESS ARE RELEASED. MESSAGES FROM IT
04      ; ARE RELEASED AND THE RECEIVERS ARE
05      ; BROKED WITH ERROR NUMBER=1.
06      ;      CALL:      RETURN:
07      ; SAC0      UNCHANGED
08      ; SAC1      UNCHANGED
09      ; SAC2      PROC      PROC
10      ; SAC3      LINK      CUR
11
12      A29:      ; CLEAN PROCESS:
13 00650'050043      STA      2      WORK1      ; SAVE(PROC);
14 00651'035011      LDA      3      BUFFER,2      ; CLEAN BUFFER CHAIN:
15      ;      BUF:=BUFFER,PROC;
16 00652'161005      A292: MOV      3,0      SNR      ; NEXT BUF:
17 00653'000417      JMP      A291      ; IF BUF=0 THEN GOTO CLEAN EVENT QUE
18 00654'031405      LDA      2      RECEIVER,3 ; RECEIVER:= RECEIVER.BUF;
19 00655'151005      MOV      2,2      SNR      ; IF RECEIVER=0 THEN
20 00656'000412      JMP      A293      ; GOTO NEXTBUF;
21 00657'102400      SUB      0,0      ;
22 00660'041405      STA      0      RECEIVER,3 ; RECEIVER.BUF:=0;
23 00661'054016      STA      3      WORK      ;
24 00662'102520      SUBZL   0,0      ;
25 00663'151113      MOVL#   2,2      SNC      ; IF RECEIVER>0 THEN
26 00664'004422      JSR      A301      ; BREAK IT(RECEIVER,1);
27 00665'030016      LDA      2      WORK      ; RESTORE(BUF);
28 00666'006155      JSR@    R11      ; REMOVE(BUF);
29 00667'034016      LDA      3      WORK      ;
30 00670'035402      A293: LDA      3      CHAIN,3      ; BUF:=NEXT.BUF;
31 00671'000761      JMP      A292      ; GOTO NEXT BUF;
32 00672'030043      A291: LDA      2      WORK1      ;
33 00673'031007      A290: LDA      2      EVENT,2      ; CLEAN EVENT QUEUE:
34 00674'021001      LDA      0      PREV,2      ; BUF:= EVENT.PROC;
35 00675'142405      SUB      2,0      SNR      ; IF BUF=PREV.BUF THEN
36 00676'002056      JMP@    EXIT      ; GOTO EXIT;
37 00677'102400      SUB      0,0      ;
38 00700'041006      STA      0      MESS0,2      ; MESS0.BUF:=
39 00701'041007      STA      0      MESS1,2      ; MESS1.BUF:=0;
40 00702'006163      JSR@    R17      ; DELIVER ANSWER(BUF);
41 00703'030043      LDA      2      WORK1      ;
42 00704'000767      JMP      A290      ; GOTO CLEAN EVENT QUEUE;
43
44

```

```

01 ; FUNCTION BREAK PROCESS(PROC,ERROR NUMBER);
02 ; ERROR NUMBER SHOULD BE GREATER THAN ZERO. THE BROKED PROCESS IS
03 ; STARTED AT ITS BREAK ADDRESS WITH THE FOLLOWING ACCUMULATOR
04 ; CONTENTS:
05 ; AC0 ERROR NUMBER
06 ; AC1 UNCHANGED
07 ; AC2 PROC
08 ; AC3 PSW,PROC//2 (ITS OLD PROGRAM COUNTER)
09 ; CALL: RETURN:
10 ; SAC0 ERROR NUMBER ERROR NUMBER
11 ; SAC1 UNCHANGED
12 ; SAC2 PROC PROC
13 ; SAC3 LINK CUR
14
15 A30: ; BREAK PROCESS:
16 00705'034056 LDA 3 EXIT ; LINK:=EXIT;
17 00706'054017 A301: STA 3 LINK ; BREAK IT:
18 00707'041017 STA 0 AC0,2 ; AC0.PROC:=ERR NO;
19 00710'051021 STA 2 AC2,2 ; AC2.PROC:=PROC;
20 00711'021023 LDA 0 PSW,2 ;
21 00712'041022 STA 0 AC3,2 ; AC3.PROC:=PSW.PROC;
22 00713'021016 LDA 0 BREADDR,2 ;
23 00714'041023 STA 0 PSW,2 ; PSW.PROC:=BREAK ADDR.PROC;
24 00715'006155 JSR# R11 ; REMOVE(PROC);
25 00716'006156 JSR# R12 ; LINK PROCESS(PROC);
26 00717'002017 JMP# LINK ; RETURN;
27
28 ; FUNCTION STOP PROCESS(PROC);
29 ; THE PROCESS IS
30 ; REMOVED IF PLACED IN A QUEUE, AND ITS STATE IS SET TO STOPPED.
31 ; IF IT WAS WAITING FOR EVENT OR ANSWER, PSW IS DECREASED BY
32 ; TWO, ENSURING THE CALL TO BE REPEATED IF THE PROCESS IS
33 ; STARTED AGAIN.
34 ; CALL: RETURN:
35 ; SAC0 UNCHANGED
36 ; SAC1 UNCHANGED
37 ; SAC2 PROC PROC
38 ; SAC3 LINK CUR
39
40 00720'102620 A31: SUBZR 0,0 ; STOP PROCESS:
41 00721'025013 LDA 1 STATE,2 ; OLD:=STATE.PROC;
42 00722'006154 JSR# R10 ; REMOVE PROCESS(PROC,180);
43 00723'125415 INC# 1,1 SNR ; IF OLD =-1 THEN
44 00724'000405 JMP A310 ; GOTO FUNCTION;
45 00725'020112 LDA 0 .64 ;
46 00726'125134 MOVZL# 1,1 SZR ; IF OLD=0 OR OLD=180
47 00727'106532 SUBZL# 0,1 SZC ; OR OLD<64 THEN
48 00730'002056 JMP# EXIT ; EXIT;
49 A310: ; FUNCTION:
50 00731'015023 DSZ PSW,2 ; PSW.PROC:=PSW.PROC-1;
51 00732'002056 JMP# EXIT ; EXIT;

```

10027 MUM12

```
01 ; FUNCTION START PROCESS(PROC);
02 ; IF STATE OF THE PROCESS IS STOPPED, THE PROCESS IS LINKED
03 ; TO RUNNING QUEUE.
04 ; CALL: RETURN:
05 ; SAC0 UNCHANGED
06 ; SAC1 UNCHANGED
07 ; SAC2 PROC PROC
08 ; SAC3 LINK CUR
09
10 00733'021013 A32: LDA 0 STATE,2 ; START PROCESS:
11 00734'103245 ADDOR 0,0 SNR ; IF STATE.PROC=180 THEN
12 00735'006156 JSR@ R12 ; LINK PROCESS(PROC);
13 00736'002056 JMP@ EXIT ; EXIT;
14
15 ; FUNCTION RECHAIN(OLD,NEW,ELEMENT);
16 ; EXCLUDES THE ELEMENT FROM THE OLD CHAIN AND INCLUDES IT IN
17 ; THE NEW CHAIN.
18 ; CALL: RETURN:
19 ; SAC0 OLD OLD
20 ; SAC1 NEW NEW
21 ; SAC2 ELEM ELEM
22 ; SAC3 LINK CUR
23
24 00737'025420 A33: LDA 1 AC1,3 ; RECHAIN:
25 00740'034056 LDA 3 EXIT ; RECHAIN(OLD,NEW,ELEM);
26 00741'002160 JMP@ R14 ; EXIT;
27
28
29
```

10028 MUM12

```
01 ; PROCEDURE CLEAR(DEVICE);
02 ; CALL: RETURN:
03 ; AC0 DESTROYED
04 ; AC1 DEVICE DEVICE
05 ; AC2 UNCHANGED
06 ; AC3 LINK DESTROYED
07
08
09 00742'000000 0 ; UNKNOWN INT COUNT
10 00743'010777 A360: ISZ .-1 ;
11 00744'000401 JMP .+1 ;
12 00745'034056 LDA 3 EXIT ; LINK := EXIT;
13 00746'020405 A36: LDA 0 A362 ; CLEAR DEVICE:
14 00747'123000 ADD 1,0 ; INSTRUCTION:= NIOC 0+DEVICE;
15 00750'040401 STA 0 .+1 ;
16 00751'060200 NIOC 0 ; EXECUTE(INSTRUCTION);
17 00752'001400 JMP +0,3 ; RETURN;
18 00753'060200 A362: NIOC 0 ; CONSTANTE
19
20
21
22 ; PROCEDURE SET INTERRUPT(DEVICE);
23 ; CALL: RETURN:
24 ; AC0 DESTROYED
25 ; AC1 DEVICE DEVICE
26 ; AC2 UNCHANGED
27 ; AC3 LINK CUR
28
29 00754'060277 A361: INTDS ; SET INTERRUPT:
30 00755'054016 STA 3 WORK ; DISABLE;
31 00756'004770 JSR A36 ; CLEAR(DEVICE);
32 00757'034045 LDA 3 TABLE ;
33 00760'137000 ADD 1,3 ;
34 00761'020040 LDA 0 CUR ;
35 00762'101120 MOVZL 0,0 ;
36 00763'041400 STA 0 +0,3 ; DEVTA(DEV):= CUR*2;
37 00764'115220 MOVZR 0,3 ;
38 00765'060177 INTEN ;
39 00766'002016 JMP @WORK ; RETURN;
40
41 ; ***** END OF MONITOR FUNCTIONS *****
42
43 .END
0000 SOURCE LINES IN ERROR
```

A0	000067'	3/06	11/05					
A00	000132'	7/03	11/45					
A001	000122'	11/32	11/37					
A01	000114'	11/31	11/36					
A1	000136'	4/43	11/42	11/52				
A10	000242'	7/06	15/14					
A100	000241'	7/05	15/12					
A11	000243'	7/07	14/03	15/27	18/11	18/48		
A12	000253'	7/08	14/04	15/45	18/28			
A120	000260'	15/50	15/54					
A13	000265'	7/09	16/09	18/34				
A130	000267'	15/55	16/11					
A14	000275'	7/10	16/27					
A141	000276'	16/28	16/33					
A15	000313'	7/11	17/11					
A150	000315'	17/13	17/20	17/24	17/28			
A151	000334'	17/16	17/29					
A152	000335'	16/40	17/30					
A16	000340'	17/44	18/25					
A160	000337'	7/12	17/42					
A17	000354'	7/13	18/10					
A170	000367'	18/23	18/38	18/50				
A171	000374'	18/19	18/30	23/29				
A172	000406'	18/42	20/25					
A2	000155'	5/12	13/08	13/60				
A20	000417'	3/10	3/11	3/13	3/14	3/15	3/16	3/17
		3/18	3/19	3/20	19/07			
A212	000443'	19/45	22/33	22/42	22/48			
A213	000444'	16/31	19/46	22/18				
A22	000452'	3/27	20/16					
A220	000463'	20/21	20/26					
A23	000470'	3/28	21/15					
A230	000471'	20/31	21/16					
A231	000504'	21/21	21/27					
A24	000510'	3/29	22/10					
A241	000517'	22/17	22/23					
A242	000526'	22/21	22/24					
A2421	000543'	22/30	22/38					
A243	000552'	22/37	22/46	22/54				
A244	000563'	22/51	23/02					
A246	000577'	22/26	23/17					
A248	000614'	22/39	23/31					
A25	000620'	3/30	23/44					
A26	000627'	3/31	24/13					
A27	000637'	3/32	24/34					
A28	000644'	3/33	24/50					
A29	000650'	3/34	25/12					
A290	000673'	25/33	25/42					
A291	000672'	25/17	25/32					
A292	000652'	25/16	25/31					
A293	000670'	25/20	25/30					
A3	000170'	13/19	13/30					
A30	000705'	3/35	26/15					
A301	000706'	4/31	25/26	26/17				
A31	000720'	3/36	26/40					
A310	000731'	26/44	26/49					
A32	000733'	3/37	27/10					
A33	000737'	3/38	27/24					
A36	000746'	7/19	28/13	28/31				
A360	000743'	5/11	28/10					

A361	000754'	7/21	28/29
A362	000753'	28/13	28/18
A4	000201'	13/25	13/28
A5	000203'	11/21	13/34
A6	000212'	7/04	13/42
A7	000220'	13/48	13/59
A8	000235'	13/34	14/01
A9	000236'	13/46	14/02
B180	000101	5/24	
B181	000102	5/25	
B1810	000113	5/34	
B1811	000114	5/35	
B1812	000115	5/36	
B1813	000116	5/37	
B1814	000117	5/38	
B1815	000120	5/39	
B182	000103	5/26	
B183	000104	5/27	
B184	000105	5/28	
B185	000106	5/29	
B186	000107	5/30	
B187	000110	5/31	
B188	000111	5/32	
B189	000112	5/33	
C10	000126	6/07	
C12	000127	6/08	
C120	000141	6/18	
C127	000142	6/19	
C13	000130	6/09	
C15	000131	6/10	
C24	000132	6/11	
C25	000133	6/12	
C255	000143	6/20	
C3	000121	6/02	
C40	000134	6/13	
C48	000135	6/14	
C5	000122	6/03	
C56	000136	6/15	
C6	000123	6/04	
C60	000137	6/16	
C63	000140	6/17	
C7	000124	6/05	
C9	000125	6/06	
CM16	000146	6/23	
CM256	000147	6/24	
CM3	000144	6/21	
CM4	000145	6/22	
CR0	000334	8/38	
CR1	000335	8/39	
CR10	000364	9/30	
CR11	000365	9/31	
CR12	000366	9/32	
CR13	000367	9/33	
CR2	000336	8/40	
CR3	000337	8/41	
CR4	000340	8/42	
CR5	000341	8/43	
CR6	000342	8/44	
CR7	000343	8/45	
CR8	000344	8/46	

CR9	000345	8/47						
DFIRS	000047	4/27	4/28	4/29	4/49	13/18		
F10	000012	3/17						
F11	000013	3/18						
F12	000014	3/19						
F13	000015	3/20						
F2	000002	3/08						
F3	000003	3/10						
F4	000004	3/11						
F5	000005	3/12						
F6	000006	3/13						
F7	000007	3/14						
F8	000010	3/15						
F9	000011	3/16						
FS0	000346	9/02						
FS1	000347	9/03						
FS2	000350	9/04						
FS3	000351	9/05						
FS4	000352	9/06						
FS5	000353	9/07						
FUNCT	000020	3/24	19/23					
INTER	000000	2/02	5/06					
LINK	000017	3/22	18/10	18/27	18/41	20/24	22/11	26/17
		26/26						
PS0	000354	9/11						
PS1	000355	9/12						
PS2	000356	9/13						
PS3	000357	9/14						
PS4	000360	9/15						
R00	000151	7/03	21/26					
R10	000154	7/06	26/42					
R100	000153	7/05	21/30	23/45	24/20			
R11	000155	7/07	25/28	26/24				
R12	000156	7/08	26/25	27/12				
R13	000157	7/09	21/33					
R14	000160	7/10	27/26					
R15	000161	7/11	22/13	22/40	24/51			
R160	000162	7/12	24/21					
R17	000163	7/13	23/49	24/39	25/40			
R301	000051	4/31	13/57					
R34	000164	7/17						
R340	000167	7/20						
R35	000165	7/18						
R36	000166	7/19						
R361	000170	7/21						
R37	000171	7/22						
R38	000172	7/23						
R39	000173	7/24						
R40	000174	7/25						
R41	000175	7/26						
R42	000176	7/27						
R43	000177	7/28						
R50	000200	8/03						
R51	000201	8/04						
R52	000202	8/05						
R528	000203	8/06						
R53	000204	8/07						
R54	000205	8/08						
R55	000206	8/09						
R56	000207	8/10						

R57	000210	8/11						
R58	000212	8/13						
R580	000211	8/12						
R59	000214	8/15						
R590	000213	8/14						
R6	000152	7/04	14/01					
R60	000215	8/16						
R61	000216	8/17						
R70	000217	8/18						
R700	000222	8/21						
R71	000220	8/19						
R72	000221	8/20						
R82	000223	8/22						
R83	000224	8/23						
R84	000225	8/24						
R85	000226	8/25						
R86	000230	8/27						
R90	000232	8/29						
R91	000233	8/30						
RTC	000014	13/06	13/13					
WORK	000016	3/21	11/05	11/14	11/60	12/06	13/20	13/29
		13/39	13/43	13/47	13/49	13/58	18/13	18/24
		18/31	18/35	19/24	19/28	20/22	22/15	22/27
		23/10	23/12	25/23	25/27	25/29	28/30	28/39
WORK1	000043	4/18	25/13	25/32	25/41			
WORK2	000044	4/19						
.A11	000237'	11/47	13/15	13/26	14/03			
.A12	000240'	11/48	13/17	13/27	14/04			

11/05	11/14	11/60	12/06	13/20	13/29			
		13/39	13/43	13/47	13/49	13/58	18/13	18/24
		18/31	18/35	19/24	19/28	20/22	22/15	22/27