BOOT

```
0001  MUU01
01                ;                                   RCSL:    43-GL524
02                ;                                   AUTHOR:  JBP
03                ;                                   EDITED:  74.10.09
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21                ;                   MUU01
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39                ; KEYWORD:          MUS, DRIVER, UTILITY, LISTING.
40                ; ABSTRACT:         MUS SYSTEM, UTILITY PROCEDURES.
41                ;                   ASCII PAPER TAPE: RCSL 43-GL525.
42                ;                   REL. BINARY PAPER TAPE: RCSL 43-GL526.
43
```

```
↑ 0002  MUU01
01                ;    ******    SYSTEM UTILITY PROCEDURES    ******
02
03                .TITL MUU01
04                .NREL
05          000012 .RDX 10
06          000001 .TXTM 1
07
08
09          000164 .LOC NEXTOP-GOS
10  00164   000000'       A34          ; NEXT OPERATION
11          000167 .LOC WAITOP-GOS
12  00167   000051'       A340         ; WAIT OPERATION
13          000165 .LOC RETURN-GOS
14  00165   000067'       A35          ; RETURN ANSWER
15          000171 .LOC SETRES-GOS
16  00171   000110'       A37          ; SET RESERVATION
17          000172 .LOC SETCON-GOS
18  00172   000120'       A38          ; SET CONVERSION
19          000173 .LOC CONBYTE-GOS
20  00173   000124'       A39          ; CONBYTE
21          000174 .LOC GETBYTE-GOS
22  00174   000130'       A40          ; GETBYTE
23          000175 .LOC PUTBYTE-GOS
24  00175   000140'       A41          ; PUTBYTE
25          000176 .LOC MULTIPLY-GOS
26  00176   000155'       A42          ; MULTIPLY
27          000177 .LOC DIVIDE-GOS
28  00177   000171'       A43          ; DIVIDE
29          000232 .LOC BINDEC-GOS
30  00232   000206'       BD00         ; BINDEC
31          000233 .LOC DECBIN-GOS
32  00233   000243'       DB00         ; DECBIN
33
34                .NREL
35
```

```
↑ 0003  MUU01
01                ; PROCEDURE NEXT OPERATION(MODE,COUNT,BUF);
02                ; WAITS FOR AN EVENT ARRIVING AT THE EVENT QUEUE OF THE
03                ; CALLING PROCESS. ANSWERS ARE UNTOUCHED. TRANSPUT MESSAGES
04                ; WITH COUNT EQUAL TO ZERO ARE ANSWERED WITH STATUS ZERO.
05                ;
06                ;         CALL:                 RETURN:                 LINK
07                ; AC0                           MODE=OP(0:13)             +0: CONTROL
08                ; AC1                           COUNT                     +1: INPUT
09                ; AC2     CUR                   CUR                       +2: OUTPUT
10                ; AC3     LINK                  BUF
11
12 00000'055024 A34:   STA     3    SAVE,2     ; NEXT OPERATION:
13 00001'152400 A341:  SUB     2,2             ;   BUF:= 0;
14 00002'006006        WAITEVENT               ;   WAIT EVENT(OP,COUNT,BUF,
15 00003'000777        JMP           .-1       ;     +0: WAIT NEXT);
16 00004'024444        LDA     1    A347       ;
17 00005'045426 A3411:STA      1    ADDRESS,3  ;   ADDRESS.CUR:=NEXT OP;
18 00006'025004        LDA     1    SENDER,2   ;
19 00007'021430        LDA     0    RESERVER,3 ;
20 00010'101004        MOV     0,0  SZR        ;   IF RESERVER.CUR<>0 AND
21 00011'106405        SUB     0,1  SNR        ;    SENDER.BUF<>RESERVER.CUR THEN
22 00012'000403        JMP           .+3       ;   GOTO REJECT IT;
23 00013'020107        LDA     0    SILLEGAL   ;
24 00014'000427        JMP           A345      ;
25 00015'021006        LDA     0    MESS0,2    ;
26 00016'101223        MOVZR   0,0  SNC        ;   IF OP(15:15)=0 THEN
27 00017'000427        JMP           A346      ;     GOTO SET CONTROL;
28 00020'025425        LDA     1    BUF,3      ;
29 00021'125405        INC     1,1  SNR        ;   IF BUF.CUR=-1 THEN
30 00022'000417        JMP           A343      ;    GOTO RETURN IT;
31 00023'025007        LDA     1    MESS1,2    ;
32 00024'125005        MOV     1,1  SNR        ;   IF COUNT=0 THEN
33 00025'000415        JMP           A344      ;     GOTO RETURN IT;
34 00026'011424        ISZ           SAVE,3    ;   RETURN:= INPUT;
35 00027'101222        MOVZR   0,0  SZC        ;   IF OP(14:14)=1 THEN
36 00030'011424        ISZ           SAVE,3    ;     RETURN:= OUTPUT;
37 00031'155000 A342:  MOV     2,3             ;
38 00032'030040        LDA     2    CUR        ;
39 00033'055025        STA     3    BUF,2      ;   BUF.CUR:=BUF;
40 00034'025410        LDA     1    MESS2,3    ;
41 00035'045026        STA     1    ADDRESS,2  ;   ADDRESS.CUR:=MESS2.BUF;
42 00036'025407        LDA     1    MESS1,3    ;
43 00037'045027        STA     1    COUNT,2    ;   COUNT.CUR:=MESS1.BUF;
44 00040'003024        JMP@          SAVE,2    ;   RETURN;
45                A343:                         ; RETURN IT:
46 00041'102521        SUBZL   0,0  SKP        ;   STATUS:=1 OR
47 00042'102400 A344:  SUB     0,0             ;   STATUS:=0;
48 00043'126400 A345:  SUB     1,1             ;
49 00044'006007        SENDANSWER              ;   SEND ANSWER(STATUS,0,BUF);
50 00045'003426        JMP@          ADDRESS,3 ;   GOTO ADDRESS.CUR;
51                A346:                         ; SET CONTROL:
52 00046'101220        MOVZR   0,0             ;   RETURN:= CONTROL;
53 00047'000762        JMP           A342      ;   RETURN;
54
55 00050'000001'A347:   A341                   ; NEXT EVENT
56
```

```
↑ 0004  MUU01
01                ; PROCEDURE WAIT OPERATION(TIMER,DEVICE,BUF.CUR,
02                ;                          BUF, MODE, COUNT);
03                ; WAITS FOR AN INTERRUPT OR TIMER OR EVENT AFTER BUF.CUR.
04                ; TRANSPUT MESSAGES ARE HANDLED AS FOR NEXTOPERATION, WITH THE
05                ; EXCEPTION THAT THE PROCEDURE RETURNS AFTER A SENDANSWER.
06                ;       CALL:              RETURN:         LINK:
07                ; AC0     TIMER              TIMER(MODE(0:13))
08                ; AC1     DEVICE             DEVICE(COUNT)
09                ; AC2     CUR                CUR(CUR)
10                ; AC3     LINK                 CUR(BUF)
11                ;
12                ; RETURNS:
13                ; +0:     TIMER EXPIRED
14                ; +1:     DEVICE INTERRUPT
15                ; +2:     EMPTY MESSAGE RETURNED
16                ; +3:     CONTROL
17                ; +4:     INPUT
18                ; +5:     OUTPUT
19                ;
20 00051'175400 A340: INC     3,3              ; WAIT OPERATION:
21 00052'055024       STA     3   SAVE,2       ;
22 00053'031025       LDA     2   BUF,2        ;   BUF:=BUF.CUR;
23 00054'151415       INC#    2,2 SNR          ;   IF BUF.CUR=-1 THEN
24 00055'152400       SUB     2,2              ;    BUF:=0;
25 00056'006002       WAIT                     ;   WAIT(TIMER,DEVICE,BUF,MODE,COUNT);
26 00057'015424 A3401:DSZ         SAVE,3       ; +0: TIMER;
27 00060'030040       LDA     2   CUR          ; +1: INTERRUPT;
28 00061'003424       JMP@        SAVE,3       ; +2: ANSWER(IRR);
29 00062'011424       ISZ         SAVE,3       ; +3: MESSAGE:
30 00063'011424       ISZ         SAVE,3       ;
31 00064'024402       LDA     1   A3404        ;   ADDRESS.CUR:= RETURN(2);
32 00065'000720       JMP         A3411        ;   GOTO TEST MESSAGE;
33
34 00066'000057'A3404:A3401                    ;
35
```

```
↑ 0005  MUU01
01                ; PROCEDURE RETURN ANSWER(STATUS);
02                ; CALCULATES THE NUMBER OF TRANSFERRED BYTES AND RETURNS
03                ; THE BUFFER. IF STATUS CONTAINS ONE OR MORE CLEAN BITS,
04                ; BUF OF CURRENT IS SET TO -1.
05                ;         CALL:                 RETURN:
06                ; AC0     STATUS                STATUS
07                ; AC1     SPECIAL ANSWER        DESTROYED
08                ; AC2                           CUR
09                ; AC3     LINK                  DESTROYED
10
11 00067'030040 A35:  LDA     2   CUR         ; RETURN ANSWER:
12 00070'055024       STA     3   SAVE,2      ;
13 00071'031025       LDA     2   BUF,2       ;   BUF:= BUF.CUR;
14 00072'035010       LDA     3   MESS2,2     ;   BYTES:= ADDR.CUR;
15 00073'045010       STA     1   MESS2,2     ;   MESS2.BUF:=SPECIAL;
16 00074'030040       LDA     2   CUR         ;
17 00075'025026       LDA     1   ADDRESS,2   ;   BYTES:=BYTES-ADDRESS.CUR;
18 00076'031025       LDA     2   BUF,2       ;
19 00077'166400       SUB     3,1             ;
20 00100'006007       SENDANSWER              ;   SEND ANSWER(STATUS,BYTES,BUF);
21 00101'171000       MOV     3,2             ;
22 00102'024405       LDA     1   A350        ;   MASK:= CLEAN BITS;
23 00103'107404       AND     0,1 SZR         ;   IF (STATUS AND MASK)<>0 THEN
24 00104'126000       ADC     1,1             ;    BUF.CUR:=-1
25 00105'045025       STA     1   BUF,2       ;   ELSE BUF.CUR:=0;
26 00106'003024       JMP@        SAVE,2      ;     RETURN;
27
28 00107'161776 A350: 7B2+7B8+7B11+7B14       ; CLEAN BITS: 0-2, 6-14
29
30                ; PROCEDURE SET RESERVATION(MODE);
31                ;         CALL:                 RETURN:
32                ; AC0     MODE(0:13)            MODE(0:12)
33                ; AC1                           DESTROYED
34                ; AC2     CUR                   CUR
35                ; AC3     LINK                  DESTROYED
36
37                A37:                            ; SET RESERVATION:
38 00110'031025       LDA     2   BUF,2       ;   BUF:= BUF.CUR;
39 00111'025007       LDA     1   MESS1,2     ;   SENDER:= MESS1.BUF;
40 00112'125004       MOV     1,1 SZR         ;   IF SENDER<>0 THEN
41 00113'025004       LDA     1   SENDER,2    ;     SENDER:= SENDER.BUF;
42 00114'030040       LDA     2   CUR         ;
43 00115'101222       MOVZR   0,0 SZC         ;   IF MODE(13:13)<>0 THEN
44 00116'045030       STA     1   RESERVER,2;     RESERVER.CUR:= SENDER;
45 00117'001400       JMP        +0,3         ;   RETURN;
46
47                ; PROCEDURE SET CONVERSION(MODE);
48                ;         CALL:                 RETURN:
49                ; AC0     MODE(0:12)            MODE(0:11)
50                ; AC1                           DESTROYED
51                ; AC2     CUR                   CUR
52                ; AC3     LINK                  DESTROYED
53
54                A38:                            ; SET CONVERSION
55 00120'025026       LDA     1   ADDRESS,2 ;   TABLE:=MESS2.BUF.CUR;
56 00121'101222       MOVZR   0,0 SZC         ;   IF MODE(12:12)=1 THEN
57 00122'045031       STA     1   CONVT,2    ;     CONV TABLE.CUR:= TABLE;
58 00123'001400       JMP        +0,3         ;   RETURN;
59
```

```
↑ 0007  MUU01
01                ; PROCEDURE CONBYTE(BYTE);
02                ;         CALL:                 RETURN:
03                ; AC0     BYTE                  BYTE (CONVERTED)
04                ; AC1                           DESTROYED
05                ; AC2     CUR                   CUR
06                ; AC3     LINK                  DESTROYED
07
08 00124'025031 A39:  LDA     1   CONVT,2    ; CONBYTE:
09 00125'125005       MOV     1,1 SNR         ;   IF CONV TABLE.CUR<>0 THEN
10 00126'001400       JMP       +0,3          ;     GETBYTE(BYTE+CONV TABLE.CUR,BYTE)
11 00127'107000       ADD     0,1             ;   RETURN;
12
13                ; PROCEDURE GETBYTE(ADDR,BYTE);
14                ;         CALL:                 RETURN:
15                ; AC0                           BYTE
16                ; AC1     ADDR                  ADDR
17                ; AC2                           CUR
18                ; AC3     LINK                  DESTROYED
19
20 00130'131220 A40:  MOVZR   1,2             ; GETBYTE:
21 00131'021000       LDA     0  +0,2         ;   VALUE:= 0.(ADDR//2);
22 00132'101003       MOV     0,0 SNC         ;   IF ADDR(15:15)=0 THEN
23 00133'101300       MOVS    0,0             ;     BYTE:= VALUE(0:7)
24 00134'030143       LDA     2  .255         ;   ELSE
25 00135'143400       AND     2,0             ;     BYTE:= VALUE(8:15);
26 00136'030040 AAAAAA LDA     2  CNR          ;
27 00137'001400       JMP       +0,3          ;   RETURN;
28
29                ; PROCEDURE PUTBYTE(ADDR,BYTE);
30                ; BYTE MUST BE IN THE RANGE 0 TO 255.
31                ;         CALL:                 RETURN:
32                ; AC0     BYTE                  BYTE
33                ; AC1     ADDR                  ADDR
34                ; AC2                           CUR
35                ; AC3     LINK                  DESTROYED
36
37 00140'131220 A41:  MOVZR   1,2             ; PUTBYTE:
38 00141'025000       LDA     1  +0,2         ;   VALUE:= 0.(ADDR//2);
39 00142'055000       STA     3  +0,2 LINK4   ;   SAVE(LINK);
40 00143'034147       LDA     3  .M256        ;
41 00144'175003       MOV     3,3 SNC         ;   IF ADDR(15:15)=0 THEN
42 00145'125300       MOVS    1,1             ;     VALUE(0:7):= BYTE
43 00146'167400       AND     3,1             ;   ELSE
44 00147'107003       ADD     0,1 SNC         ;     VALUE(8:15):= BYTE;
45 00150'125300       MOVS    1,1             ;
46 00151'035000       LDA     3  +0,2         ;   RESTORE(LINK);
47 00152'045000       STA     1  +0,2         ;   0.(ADDR//2):= VALUE;
48 00153'145100       MOVL    2,1             ;   COMMENT: CARRY CONTAINS ADDR(15:15)
49 00154'000762       JMP@      A409 LINK4    ;   RETURN;
50
```

```
↑ 0008  MUU01
                ; PROCEDURE MULTIPLY(OP1,OP2,RESULT);
                ; COMPUTES RESULT:= OP1*OP2.
                ;         CALL:                 RESULT:
                ; AC0     OP1                   RESULT(0:15)
                ; AC1     OP2                   RESULT(16:31)
                ; AC2                           CUR
                ; AC3     LINK                  DESTROYED

00155'030040 A42:  LDA     2   CUR        ; MULTIPLY:
00156'055024       STA     3   SAVE,2     ;   SAVE(LINK);
00157'152400       SUB     2,2            ;   RESULT(0:15):= 0;
00160'034146       LDA     3  .M16        ;   STEPS:= 16;
00161'125203 A420: MOVR    1,1 SNC        ;   COMMENT:
00162'151201       MOVR    2,2 SKP        ;       MULTIPLICATION
00163'113220       ADDZR   0,2            ;       AS SHOWN IN
00164'175404       INC     3,3 SZR        ;       HOW TO USE
00165'000774       JMP         A420       ;       THE NOVA
00166'125260       MOVCR   1,1            ;       COMPUTERS;
00167'141000       MOV     2,0            ;
00170'000414       JMP         A431       ;   RETURN;

                ; PROCEDURE DIVIDE(DIVIDEND,DIVISOR,QUOTIENT,REMAINDER);
                ;         CALL:                 RETURN:
                ; AC0     DIVIDEND              QUOTIENT
                ; AC1     DIVISOR               DIVISOR
                ; AC2                           CUR
                ; AC3     LINK                  REMAINDER

00171'030040 A43:  ~~LDA     2   CUR~~        ; DIVIDE:
00172'055024       STA     3   ~~SAVE,2~~ LINK ;   SAVE(LINK);
00173'176400       SUB     3,3            ;   HIGH PART:= 0;
00174'030146       LDA     2  .M16        ;   STEPS:= 16;
00175'101120       MOVZL   0,0            ;
00176'175100 A430: MOVL    3,3            ;   COMMENT:
00177'136412       SUB#    1,3 SZC        ;       DIVISION
00200'136400       SUB     1,3            ;       AS SHOWN IN
00201'101100       MOVL    0,0            ;       HOW TO USE
00202'151404       INC     2,2 SZR        ;       THE NOVA
00203'000773       JMP         A430       ;       COMPUTERS;
00204'030040 A431: ~~LDA     2   CUR~~        ;
00205'003024       JMP@        ~~SAVE,2~~     ; RETURN;
                                LINK
                LINK 4:
```

```
↑ 0009  MUU01
01                ; PROCEDURE BINDEC(WORD,ADDR,CUR);
02                ;         CALL:                 RETURN:
03                ; AC0     WORD                  DESTROYED
04                ; AC1     ADDR                  DESTROYED
05                ; AC2     CUR                   CUR
06                ; AC3     LINK                  DESTROYED
07 00206'055024 BD00: STA     3    SAVE,2     ; SAVE:=LINK;
08 00207'045027       STA     1    SAVE+3,2   ; COUNT:=ADDR;
09 00210'041025       STA     0    SAVE1,2    ; SAVE1:=WORD;
10 00211'034423       LDA     3    BD04       ;
11 00212'055026       STA     3    SAVE+2,2   ; SAVE+2:=ADDR.TENTABLE;
12                                             ;  NEW:
13 00213'025025 BD01: LDA     1    SAVE1,2    ;   WORD:=SAVE1;
14 00214'037026       LDA@    3    SAVE+2,2   ;   TEN:=WORD(SAVE2);
15 00215'011026       ISZ           SAVE+2,2   ;   SAVE2:=SAVE2+1;
16 00216'161005       MOV     3,0 SNR          ;   IF TEN=0 THEN
17 00217'000407       JMP           BD03       ;   GOTO PUTBYTE;
18 00220'020135       LDA     0    .48         ;   BASE:= "0";
19                                             ;  CREATE:
20 00221'166422 BD02: SUBZ    3,1 SZC          ;   WORD:=WORD-TEN;
21                                             ;   IF TEN<WORD THEN
22 00222'101401       INC     0,0   SKP        ;   BASE:=BASE+1 ELSE
23 00223'167001       ADD     3,1   SKP        ;   BEGIN
24                                             ;   WORD:=WORD+TEN;
25 00224'000775       JMP           BD02       ;   SAVE1:=WORD;
26 00225'045025       STA     1    SAVE1,2    ;   GOTO PUTBYTE;
27                                             ;   END;
28 00226'025027 BD03: LDA     1    SAVE+3,2   ;  PUTBYTE:
29 00227'011027       ISZ           SAVE+3,2   ;   ADDR:=ADDR+1;
30 00230'006175       PUTBYTE                  ;    PUTBYTE(WORD,ADDR);
31 00231'101004       MOV     0,0 SZR          ;   IF WORD<>0 THEN
32 00232'000761       JMP           BD01       ;   GOTO NEW;
33 00233'003024       JMP@          SAVE,2     ; RETURN;
34
35 00234'000235'BD04: .+1                      ;
36 00235'023420       10000
37 00236'001750       1000
38 00237'000144       100
39 00240'000012       10
40 00241'000001       1
41 00242'000000       0
42
```

```
↑ 0010  MUII01
01                ; PROCEDURE DECBIN(ADDR,CUR,WORD);
02                ;          CALL:                  RETURN:
03                ; AC0                             DESTROYED
04                ; AC1     ADDR                    BINARY NUMBER
05                ; AC2     CUR                     CUR
06                ; AC3     LINK                    DESTROYED
07 00243'055024 DB00: STA     3    SAVE,2       ; SAVE:=LINK;
08 00244'045025       STA     1    SAVE1,2      ; SAVE1:=ADDR;
09 00245'126400       SUB     1,1               ;
10 00246'045026       STA     1    SAVE+2,2     ; WORD:=0;
11 00247'025025 DB01: LDA     1    SAVE1,2      ;
12 00250'011025       ISZ           SAVE1,2      ; ADDR:=ADDR+1;
13 00251'006174       GETBYTE                    ;  GETBYTE(WORD,ADDR);
14 00252'024135       LDA     1    .48          ;
15 00253'034415       LDA     3    DB03         ;
16 00254'116032       ADCZ#   0,3  SZC          ;    IF NUMBER<10 THEN
17 00255'122423       SUBZ    1,0  SNC          ;     IF NUMBER>=0
18 00256'000410       JMP           DB02         ;     BEGIN
19 00257'025026       LDA     1    SAVE+2,2     ;
20 00260'135120       MOVZL   1,3               ;
21 00261'175120       MOVZL   3,3               ;
22 00262'167120       ADDZL   3,1               ;
23 00263'107000       ADD     0,1               ; WORD:=WORD*10+CHAR-48;
24 00264'045026       STA     1    SAVE+2,2     ;    SAVE2:=WORD;
25 00265'000762       JMP           DB01         ;   GOTO NEW;
26 00266'025026 DB02: LDA     1    SAVE+2,2     ;
27 00267'003024       JMP@          SAVE,2       ;
28
29 00270'000072 DB03: 58                        ;
30
31
32                ;   ******   END OF SYSTEM UTILITY PROCEDURES ******
33
34                .END
```

A34	000000'	2/10	3/12
A340	000051'	2/12	4/20
A3401	000057'	4/26	4/34
A3[illegible]4	000066'	4/31	4/34
A341	000001'	3/13	3/55
A3411	000005'	3/17	4/32
A342	000031'	3/37	3/53
A343	000041'	3/30	3/45
A344	000042'	3/33	3/47
A345	000043'	3/24	3/48
A346	000046'	3/27	3/51
A347	000050'	3/16	3/55
A35	000067'	2/14	5/11
A350	000107'	5/22	5/28
A37	000110'	2/16	5/37
A38	000120'	2/18	5/54
A39	000124'	2/20	7/08
A40	000130'	2/22	7/20
A400	000136'	7/26	7/49
A41	000140'	2/24	7/37
A42	000155'	2/26	8/09
A420	000161'	8/13	8/17
A4[illegible]	000171'	2/28	8/29
A430	000176'	8/34	8/39
A431	000204'	8/20	8/40
BD00	000206'	2/30	9/07
BD01	000213'	9/13	9/32
BD02	000221'	9/20	9/25
BD03	000226'	9/17	9/28
BD04	000234'	9/10	9/35
DB00	000243'	2/32	10/07
DB01	000247'	10/11	10/25
DB02	000266'	10/18	10/26
DB03	000270'	10/15	10/29