# RSX-11M/M-PLUS
## Utilities and Commands
## A Self-Paced Course

Exercises and Solutions

**digital**

# RSX-11M/M-PLUS
## Utilities and Commands
## A Self-Paced Course

Exercises and Solutions

Prepared by Educational Services
of
Digital Equipment Corporation

# CONTENTS

# 8  CONTROLLING TASK EXECUTION

# 9  LIBRARIES

# 10  ADVANCED MAINTENANCE OPERATIONS

# RSX-11M/M-PLUS System Overview

## LABORATORY EXERCISES

1. A computer system is made up of the _____ and the _____.


2. A _____ is a set of software that makes computer system operation more efficient.


3. List six system features of the RSX-11M/M-PLUS operating system.

   a)

   b)

   c)

   d)

   e)

   f)


4. A _____ is the smallest executable piece of code in the RSX-11M/M-PLUS operating system.

## LABORATORY EXERCISES

5. Match the terms in Column A with their meanings in Column B.

   Column A                          Column B

   _____ System Task            a) An operating system data structure
         Directory                 that is a list of all tasks
                                   competing for system resources.
   _____ Active Task List
                                b) A segment of memory in which a task
   _____ Round Robin              can reside.
         Scheduler
                                c) An optional system routine that simulates
   _____ Task State               time sharing.

   _____ Priority               d) A system function that allows a higher-
                                   priority task to gain the memory space
   _____ Shuffling                held by a lower-priority task.

   _____ Partition              e) An operating system data structure
                                   that is a list of all tasks known
   _____ Checkpointing            to the system.

   _____ User-Controlled        f) A number in the range of 1-250
         Partition                 assigned to a task to indicate
                                   its importance in obtaining system
   _____ System-Controlled        resources.
         Partition
                                g) A segment of memory whose subdivision
                                   is controlled by the operating system
                                   (dynamic).

                                h) An act accomplished by the operating
                                   system to free up a fragmented
                                   memory space within a partition.

                                i) A segment of memory whose subdivision
                                   is controlled by the user (static).

                                j) The condition of a task that is taken
                                   into consideration when the operating
                                   system is scheduling resources.

6. CPU time is allocated to the highest _____ ready- _____ task.

3

# RSX-11M/M-PLUS System Overview

## SOLUTIONS

```
>;     Solutions to Suggested Laboratory Exercises
>;
>;
>;
>;
>;     Module 1
>;
>;
>;     1.  Hardware and Software.
>;
>;     2.  Operating System
>;
>;     3.  Real Time
>;         Multiprogramming
>;         Multiuser
>;         Interactive
>;         Disk-Based
>;         Intertask Communication
>;
>;     4.  Task
>;
>;     5.  E
>;         A
>;         C
>;         J
>;         F
>;         H
>;         B
>;         D
>;         I
>;         G
>;
>;     6.  Highest priority ready-to-run task.
```

# Getting Started on the System

## LABORATORY EXERCISES

1. Log in using the proper command sequence. Note your terminal number.

2. Determine which CLI is active at your terminal. If it is not DCL, change the setting by typing the command:

   >SET /DCL=TI:

3. Using the HELP command, obtain information about the SHOW command. (Read information on all command options.)

4. Display all the system devices and complete the following table.

   **Device Name**                   **Physical Device Type**

   Disks

   Line Printers

   Magtape Units

   DECtape Units

   Floppy Drive Units

**LABORATORY EXERCISES**

5.  How many terminals are attached to the system?

    With what devices are the following pseudo devices associated?

    LB:  _____
    CO:  _____
    CL:  _____
    SY:  _____


6.  Which devices are public devices?


7.  Display the day and time.  What are the least amount of characters needed to represent the command?


8.  Display the characteristics of your terminal.

    a.  What terminal number are you?  _____

    b.  What CLI is active?  _____

    c.  What buffer size is your terminal using?  _____

    d.  What type of terminal does the O.S. think you are?  ____

# Getting Started on the System

## LABORATORY EXERCISES

9. Determine how many users are on the system.

10. Show your default device and user file directory.

11. What is the UIC for the operating system?

12. Show the tasks that are installed on the system.

    a. What task has the highest priority? What is its name and what partition does it execute in?

    b. In which partition and at what priority do the dispatcher CLI tasks ..MCR and ...DCL execute?

13. Display all tasks that are active on the system.

14. Display the partitions that make up memory.

15. A computer system has the following devices:

    2 RKØ5 cartridge disks
    2 RPØ5 disks
    1 RXØ1 floppy disk
    1 TU16 magnetic tape drive

    Give the file specification for a Macro source file called PAYROLL on each device. Use your own UIC for the UFD.

# Getting Started on the System

## LABORATORY EXERCISES

16. Write the file specification to indicate all Text files on your UFD.

17. Give the file specification for all versions of all file types whose names begin with the first two characters EL.

18. Set your terminal to MCR.

19. Using the HELP command, obtain information about the following commands:

    SET, DEVICES, TIME, TASKLIST, ATL, PARTITION

20. Repeat items 5 through 14 using MCR commands.

21. Log off the system.

# Getting Started on the System

## SOLUTIONS

1. >HELLO VANTWUYVER
   Password:

   RSX-11M BL32   [4,54] System      QUASAR
   16-DEC-81 10:34 Logged on Terminal TT56:

   Good Morning

   16-DEC-81       S. Adams

           QUASAR will be down today at lunch for DK: repairs.


   10-DEC-81       D SUMMERS

           A few changes have been made to DXDRV and the modified
           driver is loaded.  Please report any problems to Dave
           Summers KERMIT::SUMMERS.

   2-DEC-81        Parmenter

           Latest and greatest HELP now on QUASAR::LB:  Let me know if
           you have trouble getting help.

   >
2. DCL>SET TERMINAL MCR
   >
   MCR>SET /DCL=TI:
   >,
   >;      3.  To obtain help on any topic or sub-topic, simple enter HELP
   >;          topic sub-topic [...].  For example, to get help on the SHOW
   >;          command, type HELP SHOW.  To get help on  devices type HELP SHOW
   >;          DEVICES
   >;
   >;
3. >Help Show
     SHOW thing

     The SHOW command can be used to show something. The following things
     can be shown with this command:

     ACCOUNTING      ASSIGNMENTS      GROUPFLAGS      QUEUE    LIBRARY
     CLOCK_QUEUE     SYSTEM           COMMON          MEMORY   TASKS
     DEFAULT         TERMINAL         DEVICES         PARTITIONS
     [DAY]TIME       PROCESSOR        USERS

     Abbreviation: S

## SOLUTIONS

```
4.  >;       4. The chart is show below.
    >Show Devices
    HT0: Offline Loaded
    HT1: Offline Loaded
    HT2: Offline Loaded
    HT3: Offline Loaded
    LP0: Loaded
    DB0: Loaded Type=RP05
    DB1: Public Mounted Loaded Type=RP06
    DB2: Public Mounted Loaded Type=RP06
    DD0: Offline Loaded
    DD1: Offline Loaded
    DK0: Loaded
    DK1: Loaded
    DK2: Loaded
    DK3: Loaded
    DL0: Loaded Type=RL01
    DL1: Loaded Type=RL01
    DL2: Loaded Type=RL02
    DM0: Loaded Type=RK06
    DM1: Loaded Type=RK06
    DM2: Loaded Type=RK07
    DR0: Public Mounted Loaded Type=RP07
    DR1: Loaded Type=RM03
    DS0: Offline Unloaded Type=RS04
    DS1: Offline Unloaded Type=RS04
    DT0: Loaded
    DT1: Loaded
    DX0: TT7: - Private Mounted Loaded
    DX1: Loaded
    EM0: Offline Unloaded Type=ML11
    MM0: Loaded
    MM1: Loaded
    C00: Loaded
    NS0: Public Mounted Loaded
    TT0: Loaded
    TT1: Loaded
    TT2: Loaded
    TT3: Loaded
    TT4: Loaded
    TT5: Loaded
    TT6: [7,374] - Lossed in  Loaded
    TT7: [7,202] - Lossed in  Loaded
    TT10: [7,43] - Lossed in  Loaded
    TT11: [7,32] - Lossed in  Loaded
    TT12: [7,302] - Lossed in  Loaded
    TT13: Loaded
    TT14: Loaded
    TT15: [7,113] - Lossed in  Loaded
    TT16: [7,103] - Lossed in  Loaded
    TT17: Loaded
    TT20: [301,26] - Lossed in  Loaded
    TT21: Loaded
    TT22: Loaded
    TT23: [301,333] - Lossed in  Loaded
```

**SOLUTIONS**

```
5.  TT24: Loaded
    TT25: Loaded
    TT26: [7,42] - Lossed in  Loaded
    TT27: Loaded
    TT30: Loaded
    TT31: Loaded
    TT32: Loaded
    TT33: Loaded
    TT34: Loaded
    TT35: Loaded
    TT36: Loaded
    TT37: Loaded
    TT40: Loaded
    TT41: Loaded
    TT42: Loaded
    TT43: Loaded
    TT44: Loaded
    TT45: Loaded
    TT46: Loaded
    TT47: Loaded
    TT50: Loaded
    TT51: Loaded
    TT52: Loaded
    TT53: Loaded
    TT54: Loaded
    TT55: Loaded
    TT56: [305,303] - Lossed in  Loaded
    TT57: Loaded
    TT60: Loaded
    NL0:
    TI0:
    CL0:     LP0:
    LB0:     DR0:
    SY0:     DR0:
    >;
    >;
    >;
    >;      Device Name             Physical Device Type
    >;
    >;      Disks                   DB:  RP05(1),RP06(2)
    >;                              DD:  TU58(2)
    >;                              DK:  TC11(3)
    >;                              DL:  RL01(2),RL02(1)
    >;                              DM:  RK06(2),RK07(1)
    >;                              DR:  RP07(1),RM03(1)
    >;                              DS:  RS04(2)
    >;
    >;
    >;      Line Printers           LA11(1)
    >;
    >;      Magnetic Tape Units     MM:  (2)
    >;
    >;      DECtape Units           DT:  (2)
    >;                              DT:  (2)
    >;
    >;
    >;      Floppy Drive Units      DX:  (2)
    >;
```

11

# Getting Started on the System

## SOLUTIONS

```
>;
>;        5.  For this particular system, there are 60 terminals attached.
>;
>;
>;              LB:   DR0:
>;              CO:
>;              CL:   LP0:
>;              SY:   DR0:
>;
>;        6.  For this particular system, the following devices at the time
>;              of this run are public divices:   DB1:,DB2:,DR0:
>;
>;
>;        7.
```

7.
```
>SHOW DAYTIME
16:31:11 09-JAN-82
>SHDAY
INS -- File not found
>SH DAY
16:31:23 09-JAN-82
>SH D
SHOW  --  Illegal function
SH D
   ^
>SH DA
16:31:31 09-JAN-82
>
```

8.
```
>SHOW  TERMINAL
TT56:   [305,303]          [305,303]
        CLI   = DCL      BUF   = 132,     HFILL = 0         SPEED=(9600:9600)
        LINES = 66,      TERM  = LA120    OWNER = none      BRO      NOABAUD
        LOWER    NOPRIV  NOHOLD  NOSLAVE NOESC    NOCRT    FORM     NOREMOTE
        ECHO     NOVFILL NOHHT   NOFDX    WRAP    NORPA    NOEBC    TYPEAHEAD
>;
>;
```

9.
```
>SHOW USERS
TT6:  [7,374]
TT7:  [7,202]
TT10: [7,43]
TT11: [7,32]
TT12: [7,302]
TT15: [7,113]
TT16: [7,103]
TT20: [301,26]
TT23: [301,333]
TT26: [7,42]
TT56: [305,303]
>;
>;
>;
```

10.
```
>SHOW DEFAULTS
   DR0:[305,303]  TT56:
>;
>;        10,
```

11.
```
>SHOW SYSTEM
SYSUIC=[4,54]
>;
```

12

## SOLUTIONS

12.
```
>SHOW TASKS /INSTALLED
LDR...  12.19  LDRPAR  248.  00002500  LB0:-01330246  FIXED
TKTN    04.8   TKNPAR  248.  00010000  LB0:-01334700
RMDEMO  V1.05  GEN     225.  00027100  LB0:-01363011
MTAACP  0013   GEN     200.  00014400  SY0:-01350674
F11MSG  V0012  GEN     200.  00005600  LB0:-01350440
NETACP  V02.00 GEN     200.  00025600  SY0:-01323421
EVP...  V01.00 GEN     199.  00013100  SY0:-01321702  CHECKPOINTED
MCR...  2.02   SYSPAR  160.  00012000  LB0:-01335065
...MCR  1.1    GEN     160.  00020400  LB0:-01335446
SHOT56  1.1    GEN     160.  00020400  LB0:-01335446
...DCL  00     GEN     160.  00037100  SY0:-01327224
...HEL  01.0   GEN     150.  00037600  LB0:-01346314
...CA.  4.01   GEN     150.  00005100  LB0:-01350427
NVP...  V01.00 GEN     150.  00012400  SY0:-01324406
...BYE  04.1   GEN     150.  00012400  SY0:-01120306
...CVT  4.02   GEN     150.  00052200  SY0:-01007774
...DLG  4.02   GEN     150.  00052200  SY0:-01007774
F11ACP  M0401  GEN     149.  00044000  LB0:-01342047
SYOFCP  M0401  GEN     149.  00044000  LB0:-01357463
WKFFCP  M0401  GEN     149.  00044000  LB0:-01362650
DB1FCP  M0401  GEN     149.  00044000  SY0:-01361711
DB2FCP  M0401  GEN     149.  00024000  SY0:-01362057
PMT...  01.52  GEN     148.  00006300  LB0:-01330235
ERRLOG  V1.05  GEN     148.  00040000  LB0:-01350571
COT...  01.02  GEN     145.  00017100  LB0:-01346146  CHECKPOINTED
...DMO  23.26  GEN     140.  00014600  LB0:-01335712
...INI  22.03  GEN     140.  00034700  LB0:-01337502
...MOU  2503   GEN     140.  00037700  LB0:-01336751
...UFD  V0412  GEN     140.  00005700  LB0:-01335120
PMD...  07.2   GEN     130.  00017200  LB0:-01342114
SHF...  5.08   SYSPAR  105.  00012000  LB0:-01340414
...INS  06     GEN     100.  00034400  LB0:-01334757
...SEN  22JUL  GEN     100.  00104400  LB0:-00662645
RMHACP  V02.00 GEN     100.  00023300  SY0:-01111131
MAL$$$  11RX   GEN     100.  00132300  SY0:-00663342
MAL.1   11RX   GEN     100.  00132300  SY0:-00663342
MAL.0   11RX   GEN     100.  00132300  SY0:-00663342
PMR$$$  V01.05 GEN     100.  00034300  SY0:-00660507
QMG...  1.8    GEN      70.  00032200  LB0:-01353625  CHECKPOINTED
LPO     1.10   GEN      70.  00014000  LB0:-01354614  CHECKPOINTED
PRT...  1.2    GEN      70.  00001100  LB0:-01350452
...BRU  1.04   GEN      70.  00156400  LB0:-01426072
...TEC  V36    GEN      65.  00177100  SY0:-00074254
...EDI  M11.04 GEN      65.  00123500  SY0:-01476622
...SOS  07.00  GEN      65.  00073100  SY0:-01421320
...EDT  02.00  GEN      65.  00130700  LB0:-01416243
...MAI  02NOV  GEN      65.  00123700  SY0:-00322226
RMT...  V02.01 GEN      65.  00027300  SY0:-01321574
...RVT  X02.19 GEN      65.  00032500  SY0:-00663556
MAIT10  02NOV  GEN      65.  00123700  SY0:-00322226
...AT.  02     GEN      64.  00060000  LB0:-01336321
...BRO  V04    GEN      50.  00030300  LB0:-01343656
LPINIT  01.03  GEN      50.  00015500  LB0:-01340626
...QUE  1.10   GEN      50.  00020600  LB0:-01354525
...PRI  1.10   GEN      50.  00020600  LB0:-01354525
...PIP  M1344  GEN      50.  00131600  SY0:-01500271
...MAC  M1200  GEN      50.  00107700  SY0:-01446173
...TKB  M40.02 GEN      50.  00076600  SY0:-01345333
...LBR  06.00  GEN      50.  00154200  SY0:-01476674
```

## SOLUTIONS

```
CRF... V01     GEN      50. 00116500 SY0:-01477246
...MUN V36     GEN      50. 00177100 SY0:-00074254
...CMP'V0109   GEN      50. 00151300 SY0:-01477276
...RNO M0101   GEN      50. 00061500 SY0:-01477620
...SLP 11.03   GEN'     50. 00027200 SY0:-01476520
...SRD 4.1F    GEN      50. 00034200 SY0:-01476077
...BCK 0002CM  GEN      50. 00065600 SY0:-00065660
...DOC 06DEC   GEN      50. 00061600 SY0:-00063566
...DTR V01.10  GEN      50. 00175000 SY0:-00063652
MAL... 30OCT   GEN      50. 00057700 SY0:-00553461
NICE.. V02.00  GEN      50. 00026600 SY0:-01275172
...NCP V01.00  GEN      50. 00037100 SY0:-01326011
FAL... V3.01   GEN      50. 00075500 SY0:-01236514
...NFT V02.00  GEN      50. 00035300 SY0:-01275315
.CMTS. V02.00  GEN      50. 00002000 SY0:-01120301
TCL... V02.00  GEN      50. 00002400 SY0:-01120274
...NTD V01.00  GEN      50. 00021200 SY0:-01272475
NTD... V01.00  GEN      50. 00011700 SY0:-01321730
...RMT V02.00  GEN      50. 00002100 SY0:-01120323
...PRV 01      GEN      50. 00001600 SY0:-00230052
...VMR         GEN      50. 00037500 DM0:-00022624
```

13. 
```
>
>SHOW TASKS/ACTIVE/ALL
LDR...
RMDEMO
NETACP
EVP...
MCR...
SHOT56
F11ACP
DB2FCP
PMT...
COT...
...SEN
QMG...
LPO
...BRU
...AT.
```

## SOLUTIONS

```
14.   >SHOW PARTITIONS
      CEXPAR 114234 00114300 00003500 MAIN COM
      TTPAR  114170 00120000 00040000 MAIN TASK
      EXCOM1 114124 00160000 00014700 MAIN COM
      EXCOM2 114060 00174700 00006100 MAIN COM
      SYSPAR 114014 00203000 00012000 MAIN TASK
      TKNPAR 113750 00215000 00010000 MAIN TASK
      DRVPAR 113704 00225000 00030400 MAIN SYS
             113640 00225000 00002100 SUB  DRIVER -DB:
             113540 00227100 00001200 SUB  DRIVER -DK:
             113440 00230300 00003100 SUB  DRIVER -DM:
             113340 00233400 00003000 SUB  DRIVER -DR:
             113140 00237500 00001100 SUB  DRIVER -DT:
             113040 00240600 00001400 SUB  DRIVER -DX:
             112740 00242200 00002200 SUB  DRIVER -DL:
             112640 00244400 00003000 SUB  DRIVER -DD:
             112540 00247400 00001300 SUB  DRIVER -LP:
             112404 00250700 00004300 SUB  DRIVER -MM:
             111750 00255200 00000200 SUB  DRIVER -CO:
      LDRPAR 111704 00255400 00002500 MAIN TASK
      BASIC2 111640 00260100 00040000 MAIN COM
      FCSRES 111574 00320100 00040000 MAIN COM
      TSTPAR 070674 00360100 00100000 MAIN TASK
      GEN    111464 00460100 03317700 MAIN SYS
             045100 00460100 00006300 SUB  (PMT...)
             057250 00466400 00024000 SUB  (DB2FCP)
             045620 00520600 00025600 SUB  (NETACP)
             072650 00546400 00020400 SUB  (SHOT56)
             046200 00577000 00044000 SUB  (F11ACP)
             045034 00643000 00032200 SUB  (QMG...)
             045554 01001100 00027100 SUB  (RMDEMO)
             070010 01120200 00060000 SUB  (...AT.)
             047640 01200200 00104400 SUB  (...SEN)
             067600 01304600 00132300 SUB  (MAL.O )
             045360 01552100 00156400 SUB  (...BRU)
      NT.PCL 067534 03671600 00003500 SUB  DYNAMIC
      NT.NSP 060070 03675300 00016000 SUB  DYNAMIC
      NT.AUX 044770 03713300 00001100 SUB  DYNAMIC
      POOL.. 057560 03714400 00061700 SUB  DYNAMIC
             113240 03776300 00001500 SUB  DRIVER -HT:
```

15

# SOLUTIONS

```
15.   >;
      >;            A.   DK:[305,303]PAYROLL.MAC
      >;                 DK1:[305,303]PAYROLL.MAC
      >;            B.   DB:[305,303]PAYROLL.MAC
      >;                 DB1:[305,303]PAYROLL.MAC
      >;            C.   DX:[305,303]PAYROLL.MAC
      >;                 DX01:[305,303]PAYROLL.MAC
      >;            D.   MM:[305,303]PAYROLL.MAC
      >;                 MM1:[305,303]PAYROLLMAC
      >;
      >;
16.   >;            DB:[305,303]*.TXT
      >;
17.   >;            DB:[305,303]EL*.*
      >;
      >;      17.
```

```
      >SET TERMINAL/MCR
18.   >HELP SET
19.   The general form of the SET system and/or device characteristics
      command is:
                  SET /keyword     where keyword can be one of:

      Device Characteristics:
            /[NO]ABAUD[=TTnn:]         /[NO]BRO[=TTnn:]        /BUF=ddnn:[size]
            /CLI=TTnn:[cli]           /[NO]CRT[=TTnn:]        /DCL[=TTnn:]
            /[NO]EBC[=TTnn:]          /[NO]ESCSEQ[=TTnn:]     /[NO]ECHO[=TTnn:]
            /[NO]FDX[=TTnn:]          /[NO]FORMFEED[=TTnn:]   /HFILL=TTnn:[value]
            /[NO]HHT[=TTnn:]          /[NO]HOLD[=TTnn:]       /LINES=TTnn:[value]
            /[NO]LOWER[=ddnn:]        /MCR[=TTnn:]            /[NO]PRIV[=TTnn:]
            /[NO]PUB[=ddnn:]          /[NO]REMOTE[=TTnn:[sp]] /[NO]RPA[=TTnn:]
            /[NO]SLAVE[=TTnn:]        /SPEED=TTnn:[recv:xmit] /TERM=TTnn:[value]
            /[NO]TYPEAHEAD[=TTnn:]    /UIC[=[uic][:TTnn:]]    /[NO]VFILL[=TTnn:]
            /[NO]WCHK[=ddnn:]         /[NO]WRAP[=TTnn:]

      System Parameters:
            /[NO]LOGON               /MAXPKT[=n]             /SYSUIC[=[uic]]

      Memory Allocation:
            /MAXEXT[=size]           /POOL=[top]             /TOP
            /PLCTL[=[high][:[low][:[frz][:[base]]]]]
            /MAIN=pname[:base:size:type]       /NOMAIN=pname
            /SUB=mname:sname[:base:size]       /NOSUB=mname:sname

      Console Logging (COT) commands:
            SET /COLOG/option

      For more information, type HELP SET keyword.
      >
      >
```

# SOLUTIONS

```
20.  >DEVICES
     HT0: Offline Loaded
     HT1: Offline Loaded
     HT2: Offline Loaded
     HT3: Offline Loaded
     LP0: Loaded
     DB0: Loaded Type=RP05
     DB1: Public Mounted Loaded Type=RP06
     DB2: Public Mounted Loaded Type=RP06
     DD0: Offline Loaded
     DD1: Offline Loaded
     DK0: Loaded
     DK1: Loaded
     DK2: Loaded
     DK3: Loaded
     DL0: Loaded Type=RL01
     DL1: Loaded Type=RL01
     DL2: Loaded Type=RL02
     DM0: Loaded Type=RK06
     DM1: Mounted Loaded Type=RK06
     DM2: Loaded Type=RK07
     DR0: Public Mounted Loaded Type=RP07
     DR1: Loaded Type=RM03
     DS0: Offline Unloaded Type=RS04
     DS1: Offline Unloaded Type=RS04
     DT0: Loaded
     DT1: Loaded
     DX0: Loaded
     DX1: Loaded
     EM0: Offline Unloaded Type=ML11
     MM0: Loaded
     >TIME
     10:44:33 16-DEC-81

     >SET /CLI=TI:
     CLI=TT56:MCR
     >
     MCR>SET /BUF=TI:
     BUF=TI0:132.
     >SET /BUF=TI:80.
     >SET /BUF=TI:
     BUF=TI0:80.

     >SET /CRT
     CRT=TT1:
     CRT=TT3:
     CRT=TT6:
     CRT=TT7:
     CRT=TT10:
     CRT=TT11:
     CRT=TT12:
     CRT=TT13:
     CRT=TT14:
     CRT=TT15:
     CRT=TT16:
     CRT=TT17:
     CRT=TT20:
     CRT=TT21:
     CRT=TT22:
     CRT=TT23:
     CRT=TT24:
```

## SOLUTIONS

```
CRT=TT25:
CRT=TT26:
CRT=TT27:
CRT=TT30:
CRT=TT31:
CRT=TT32:
CRT=TT33:
CRT=TT34:
CRT=TT35:
CRT=TT36:
CRT=TT37:
CRT=TT40:
CRT=TT41:
CRT=TT42:
CRT=TT43:
CRT=TT44:
CRT=TT47:
CRT=TT52:
CRT=TT53:
CRT=TT55:
CRT=TT57:
CRT=TT60:
>
>DEV TT:
TT0: Loaded
TT1: Loaded
TT2: Loaded
TT3: Loaded
TT4: Loaded
TT5: Loaded
TT6: [7,374] - Lossed in  Loaded
TT7: Loaded
TT10: [7,43] - Lossed in  Loaded
TT11: [7,32] - Lossed in  Loaded
TT12: [7,302] - Lossed in  Loaded
TT13: Loaded
TT14: [7,110] - Lossed in  Loaded
TT15: [7,113] - Lossed in  Loaded
TT16: [301,303] - Lossed in  Loaded
TT17: Loaded
TT20: Loaded
TT21: [7,372] - Lossed in  Loaded
TT22: Loaded
TT23: Loaded
TT24: Loaded
TT25: Loaded
TT26: [7,42] - Lossed in  Loaded
TT27: Loaded
TT30: Loaded
TT31: Loaded
TT32: Loaded
TT33: Loaded
TT34: Loaded
TT35: Loaded
TT36: Loaded
TT37: Loaded
TT40: Loaded
TT41: Loaded
TT42: Loaded
TT43: Loaded
TT44: Loaded
```

# SOLUTIONS

```
TT45: Loaded
TT46: Loaded
TT47: [7,110] - Lossed in  Loaded
TT50: Loaded
TT51: [320,3] - Lossed in  Loaded
TT52: Loaded
TT53: Loaded
TT54: Loaded
TT55: Loaded
TT56: [305,303] - Lossed in  Loaded
TT57: Loaded
TT60: Loaded

>SET /UIC
UIC=[305,303]
>SET /SYSUIC
SYSUIC=[2,54]

>TAS
LDR... 12.19   LDRPAR 248.  00002500 LB0:-01241163 FIXED
TKTN   04.8    TKNPAR 248.  00010000 LB0:-01242225
RMDEMO V1.05   GEN    225.  00027100 LB0:-01260644
MTAACP 0013    GEN    200.  00014400 SY0:-01246560
F11MSG V0012   GEN    200.  00005600 LB0:-01242420
NETACP V02.00  GEN    200.  00025600 SY0:-01416535
EVF... V01.00  GEN    199.  00013100 SY0:-01262477 CHECKPOINTED
MCR... 2.02    SYSPAR 160.  00012000 LB0:-01242053
...MCR 1.1     GEN    160.  00020400 LB0:-01243440
...DCL 00      GEN    160.  00037200 SY0:-01245616
...HEL 01.0    GEN    150.  00037600 LB0:-01246365
...CA. 4.01    GEN    150.  00005100 LB0:-01242474
NVP... V01.00  GEN    150.  00012400 SY0:-01416117
...BYE 04.1    GEN    150.  00012400 SY0:-01371726
F11ACP M0401   GEN    149.  00044000 LB0:-01245171
SYOFCP M0401   GEN    149.  00044000 LB0:-01260256
WKFFCP M0401   GEN    149.  00044000 LB0:-01260577
DB1FCP M0401   GEN    149.  00044000 SY0:-01260367
DB2FCP M0401   GEN    149.  00024000 SY0:-01260460
PMT... 01.52   GEN    148.  00006300 LB0:-01123626
ERRLOG V1.05   GEN    148.  00040000 LB0:-01246455
COT... 01.02   GEN    145.  00017100 LB0:-01245535 CHECKPOINTED
...DMO 23.26   GEN    140.  00014600 LB0:-01242576
...INI 22.03   GEN    140.  00034700 LB0:-01244244
...MOU 2503    GEN    140.  00037700 LB0:-01244335
...UFD V0412   GEN    140.  00005700 LB0:-01242311
PMD... 07.2    GEN    130.  00017200 LB0:-01245236
SHF... 5.08    SYSPAR 105.  00012000 LB0:-01242627
...INS 06      GEN    100.  00034400 LR0:-00216226
...SEN 22JUL   GEN    100.  00104400 LB0:-00662645
RMHACP V02.00  GEN    100.  00023300 SY0:-01265452
MAL$$$ 11RX    GEN    100.  00132300 SY0:-00663342
PMR$$$ V01.05  GEN    100.  00034300 SY0:-00660507
QMG... 1.8     GEN     70.  00032200 LB0:-01247466 CHECKPOINTED
LPO    1.10    GEN     70.  00014000 LB0:-01247612 CHECKPOINTED
PRT... 1.2     GEN     70.  00001100 LB0:-01242245
...BRU 1.04    GEN     70.  00156400 LB0:-01254320
...TEC V36     GEN     65.  00177100 SY0:-00074254
...EDI M11.04  GEN     65.  00123500 SY0:-01256107
...SOS 07.00   GEN     65.  00073100 SY0:-01252640
...EDT 02.00   GEN     65.  00130700 LB0:-01252440
...MAI 02NOV   GEN     65.  00123700 SY0:-00322226
```

# SOLUTIONS

```
RMT...  V02.01 GEN    65. 00027300 SY0:-01416475
...RVT  X02.19 GEN    65. 00032500 SY0:-00663556
FMTT51  01.08  GEN    65. 00070000 LB0:-01253466
EDTT25  02.00  GEN    65. 00130700 LB0:-01252440
EDTT7   02.00  GEN    65. 00130700 LB0:-01252440
...AT,  02     GEN    64. 00060000 LB0:-01243706
...BRO  V04    GEN    50. 00030300 LB0:-01245502
LPINIT  01.03  GEN    50. 00015500 LB0:-01242634
...QUE  1.10   GEN    50. 00020600 LB0:-01247523
...PRI  1.10   GEN    50. 00020600 LB0:-01247523
...PIP  M1344  GEN    50. 00131600 SY0:-01257162
...MAC  M1200  GEN    50. 00107700 SY0:-01254542
...TKB  M40.02 GEN    50. 00076600 SY0:-01255446
...LBR  06.00  GEN    50. 00154200 SY0:-01256161
CRF...  V01    GEN    50. 00116500 SY0:-01256631
...MUN  V36    GEN    50. 00177100 SY0:-00074254
...CMP  V0109  GEN    50. 00151300 SY0:-01256761
...RNO  M0101  GEN    50. 00061500 SY0:-01256661
...SLP  11.03  GEN    50. 00027200 SY0:-01252240
...SRD  4.1F   GEN    50. 00034200 SY0:-01256733
...BCK  0002CM GEN    50. 00065600 SY0:-00065660
...DOC  06DEC  GEN    50. 00061600 SY0:-00063566
...DTR  V01.10 GEN    50. 00175000 SY0:-00063652
MAL..,  30OCT  GEN    50. 00057700 SY0:-00553461
NICE..  V02.00 GEN    50. 00026600 SY0:-01374201
...NCP  V01.00 GEN    50. 00037100 SY0:-01417214
FAL...  V3.01  GEN    50. 00075500 SY0:-01373420
...NFT  V02.00 GEN    50. 00035300 SY0:-01373520
.CMTS.  V02.00 GEN    50. 00002000 SY0:-01371326
TCL...  V02.00 GEN    50. 00002400 SY0:-01262513
...NTD  V01.00 GEN    50. 00021200 SY0:-01265411
NTD...  V01.00 GEN    50. 00011700 SY0:-01416464
...RMT  V02.00 GEN    50. 00002100 SY0:-01371332
```

# SOLUTIONS

```
>ATL
LDR... 111350  LDRPAR 111704 00255400-00260100  Pri - 248.  Dpri - 248.
    Status:  -CHK FXD STP -PMD PRV NSD
    TI - COO:  IOC - 0.  BIO - 0.  Efls - 000001 000000  PS - 170000
    PC - 120354 Ress 0-6 120212 011457 177777 107146 045620 107114 120166
RMDEMO 051404  GEN     047640 00643000-00672100  Pri - 225.  Dpri - 225.
    Status:  -CHK WFR -PMD PRV MCR
    TI - TTO:  IOC - 0.  BIO - 0.  Efls - 000021 040000  PS - 170010
    PC - 125250 Ress 0-6 000000 136357 132110 000000 136421 134654 121166
NETACP 056750  GEN     056640 00546400-00574200  Pri - 200.  Dpri - 200.
    Status:  STP ACP -PMD PRV NSD
    TI - COO:  IOC - 0.  BIO - 0.  Efls - 000000 000000  PS - 170000
    PC - 120452 Ress 0-6 000006 000002 071134 000000 053004 065654 120242
EVP... 057064  GEN     057200 01006000-01022200  Pri - 199.  Dpri - 199.
    Status:  OUT CKP STP -PMD PRV
    TI - COO:  IOC - 0.  BIO - 0.  Efls - 000001 000000
MCR... 107460  SYSPAR 114014 00203000-00215000  Pri - 160.  Dpri - 160.
    Status:  STP -PMD PRV MCR CLI NSD CAL
    TI - TT26:  IOC - 0.  BIO - 0.  Efls - 000001 040000  PS - 170000
    PC - 122462 Ress 0-6 000000 120476 120515 000003 010753 044144 120366
...MCR 107114  GEN     045620 00746700-00767300  Pri - 160.  Dpri - 160.
    Status:  CKD -PMD PRV MCR NSD
    TI - TT56:  IOC - 0.  BIO - 0.  Efls - 000001 040000  PS - 170000
    PC - 121502 Ress 0-6 120424 121324 121734 000000 121330 000000 000742
F11ACP 111050  GEN     046200 00577000-00643000  Pri - 149.  Dpri - 149.
    Status:  STP ACP -PMD PRV NSD CAL
    TI - COO:  IOC - 0.  BIO - 0.  Efls - 000002 000002  PS - 170000
    PC - 135350 Ress 0-6 000000 000070 000016 151460 046054 045620 120310
DB1FCP 047360  GEN     045664 00702700-00746700  Pri - 149.  Dpri - 149.
    Status:  -CHK STP ACP -PMD PRV NSD
    TI - COO:  IOC - 0.  BIO - 0.  Efls - 000002 000001  PS - 170000
    PC - 135350 Ress 0-6 034024 000070 000016 150364 047474 056330 120310
PMT... 105754  GEN     045100 00460100-00466400  Pri - 148.  Dpri - 148.
    Status:  STP -PMD PRV CAL
    TI - COO:  IOC - 0.  BIO - 0.  Efls - 000200 000000  PS - 170000
    PC - 121736 Ress 0-6 000074 177604 000000 140164 157700 000000 120212
COT... 056010  GEN     046704 00515700-00535000  Pri - 145.  Dpri - 145.
    Status:  STP -PMD PRV NSD
    TI - TTO:  IOC - 0.  BIO - 0.  Efls - 001000 140000  PS - 170000
    PC - 121564 Ress 0-6 121150 013400 000002 132622 130631 130631 120576
QMG... 050464  GEN     045364 01072600-01125000  Pri - 70.  Dpri - 70.
    Status:  OUT CKP STP -PMD PRV
    TI - COO:  IOC - 0.  BIO - 0.  Efls - 000000 000000
LPO    050600  GEN     045034 00466400-00502400  Pri - 70.  Dpri - 70.
    Status:  OUT CKP STP PRV SLV
    TI - COO:  IOC - 0.  BIO - 0.  Efls - 000400 040000
RMT... 061450  GEN     045554 00466400-00515700  Pri - 65.  Dpri - 65.
    Status:  STP -PMD PRV MCR CLI NSD
    TI - COO:  IOC - 1.  BIO - 0.  Efls - 000003 000000  PS - 170000
    PC - 132316 Ress 0-6 000030 124263 000015 041004 124224 121372 121216
FMTT51 064350  GEN     044770 01125600-01215600  Pri - 65.  Dpri - 65.
    Status:  WFR -PMD REM MCR
    TI - TT51:  IOC - 1.  BIO - 0.  Efls - 000002 040000  PS - 170000
    PC - 002242 Ress 0-6 000003 033304 000060 004044 005462 005456 001172
EDTT25 065760  GEN     067400 01346500-01477400  Pri - 65.  Dpri - 65.
    Status:  WFR -PMD REM MCR
    TI - TT25:  IOC - 1.  BIO - 0.  Efls - 000001 140000  PS - 170000
    PC - 001452 Ress 0-6 010212 036116 000446 000012 140060 000001 000674
EDTT7  066500  GEN     060034 01477400-01630300  Pri - 65.  Dpri - 65.
    Status:  STP WFR -PMD REM MCR
    TI - TT7:  IOC - 0.  BIO - 1.  Efls - 001001 140000  PS - 170000
    PC - 022152 Ress 0-6 001124 000011 000000 000000 140060 000000 001122
```

# SOLUTIONS

```
>PAR
CEXPAR 114234 00114300 00003500 MAIN COM
TTPAR  114170 00120000 00040000 MAIN TASK
EXCOM1 114124 00160000 00014700 MAIN COM
EXCOM2 114060 00174700 00006100 MAIN COM
SYSPAR 114014 00203000 00012000 MAIN TASK
TKNPAR 113750 00215000 00010000 MAIN TASK
DRVPAR 113704 00225000 00030400 MAIN SYS
       113640 00225000 00002100 SUB  DRIVER -DB:
       113540 00227100 00001200 SUB  DRIVER -DK:
       113440 00230300 00003100 SUB  DRIVER -DM:
       113340 00233400 00003000 SUB  DRIVER -DR:
       113140 00237500 00001100 SUB  DRIVER -DT:
       113040 00240600 00001400 SUB  DRIVER -DX:
       112740 00242200 00002200 SUB  DRIVER -DL:
       112640 00244400 00003000 SUB  DRIVER -DD:
       112540 00247400 00001300 SUB  DRIVER -LP:
       112404 00250700 00004300 SUB  DRIVER -MM:
       111750 00255200 00000200 SUB  DRIVER -CO:
LDRPAR 111704 00255400 00002500 MAIN TASK
BASIC2 111640 00260100 00040000 MAIN COM
FCSRES 111574 00320100 00040000 MAIN COM
TSTPAR 111530 00360100 00100000 MAIN TASK
GEN    111464 00460100 03317700 MAIN SYS
       045100 00460100 00006300 SUB  (PMT...)
       045554 00466400 00027300 SUB  (RMT...)
       056640 00546400 00025600 SUB  (NETACP)
       046200 00577000 00044000 SUB  (F11ACP)
       047640 00643000 00027100 SUB  (RMDEMO)
       045664 00702700 00044000 SUB  (DB1FCP)
       046704 00746700 00017100 SUB  (COT...)
       056330 00767300 00130700 SUB  (...EDT)
       044770 01125600 00070000 SUB  (FMTT51)
       045164 01215600 00060000 SUB  (...AT.)
       067734 01275600 00020400 SUB  (MCRT56)
       067400 01346500 00130700 SUB  (EDTT25)
       060034 01477400 00130700 SUB  (EDTT7 )
       066140 01630300 00132300 SUB  (MAL.0 )
       066364'01762600 00034400 SUB  (...INS)
NT.PCL 057244 03671600 00003500 SUB  DYNAMIC
NT.NSP 056704 03675300 00016000 SUB  DYNAMIC
NT.AUX 047574 03713300 00001100 SUB  DYNAMIC
POOL.. 056374 03714400 00061700 SUB  DYNAMIC
       045230 03776300 00001500 SUB  DRIVER -HT:
>
>
>
>
>BYE
Have a Good Afternoon
23-DEC-81 14:45 TT56: logged off QUASAR
```

# Creating and Modifying Files

## LABORATORY EXERCISES

1. Using Example 3-1 of the Student Workbook, create a file with EDT containing the text as shown in the listing. Do not be concerned with typographical errors; you will soon learn how to correct them. Be sure you understand:

   a. How to invoke the editor specifying a new file.

   b. How to enter input mode, and how to terminate it.

   c. How to save the results of your editing session in a file.

2. Using the file NEW.TXT created in exercise 1 above, do the following:

   a. Add the following lines of text to the end of the file using EDT line mode:

      Nothing motivates a man more than to see his boss putting in an honest days work.

      When all else fails read the instructions.

      Everything that goes up must come down.

   b. Display these newly added lines on your terminal.

   c. Using the substitute command, correct all typos.

   d. Move the last sentence, "Everything that goes up must come down," to the beginning of the file.

   e. Duplicate the line beginning with "Nothing motivates a man" using the copy command.

   f. Display HELP on the resequence command.

   g. Resequence your file's line numbers to begin at 100 and increment by 10.

   h. Save your terminal session in a file called NEWER.TXT.

# Creating and Modifying Files

## LABORATORY EXERCISES

3. Edit the file NEWER.TXT created in exercise 2. Using the Change command, put the Editor in character mode. Do the following:

   a. Use the HELP key and obtain help for the following: ↑,↓,→, ← . Then type each key several times to learn its function.

   b. Use the HELP key and obtain help on the following keys: ADVANCE, BACKUP, BLINE, EOL and WORD. Using these keys, move the cursor forwards and backwards in the file to understand each of their functions.

4. Position the cursor after the words "when all else fails read the instructions" and add the following lines of text by typing them on the standard keyboard:

   THE OTHER LINE MOVES FASTER.
   ANYTHING CAN BE MADE TO WORK IF YOU FIDDLE WITH IT LONG ENOUGH.
   IT WORKS BETTER IF YOU PLUG IT IN.
   IF YOU CANNOT CONVINCE THEM, CONFUSE THEM.

5. Obtain HELP on the following keys: DEL L, DEL W, DEL C. Using these keys, delete the line "Everything that goes up must come down."

   In the line "Anything can be made to work ..." delete the words "fiddle with" and add the words "keep at".

6. Save the results of your editing session in a file NEWEST.TXT.

# Creating and Modifying Files

# SOLUTIONS

1.

a. &gt;EDIT/EDT NEW.TXT
   Input file does not exist
   [EOB]

b. *I

```
                    TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
                    AND HE'LL BELIVE YOU.  TELL HIM A BENCH HAS WET PAINT
                    ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.

                    UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
                    TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
                    ORGANISM WILL DO AS IT DARN WELL PLEASES.

                    ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.


                    THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
                    TO THE LEVEL OF MANAGEMENT
                    ^Z
```
   [EOB]

c. *EXIT
   DR0:[305,303]NEW.TXT;1 13 lines

# SOLUTIONS

```
2.       >
         >EDIT/EDT NEW.TXT
             1                TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
 a. *I E
                             NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS IN A
                             HONEST DAYS WORK.

                             WHEN ALL ELSE FAILS READ THE INSTRUCTIONS.

                             EVERYTHING THAT GOES UP MUST COME DOWWN.
                             ^Z
      [EOB]
 b. *%WH
             1                TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
             2                AND HE'LL BELIVE YOU.  TELL HIM A BENCH HAS WET PAINT
             3                ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
             4
             5                UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
             6                TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
             7                ORGANISM WILL DO AS IT DARN WELL PLEASES.
             8
             9                ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
            10
            11
            12                THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
            13                TO THE LEVEL OF MANAGEMETN.
            14                NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS IN A
            15                HONEST DAYS WORK.
            16
            17                WHEN ALL ELSE FAILS READ THE INSTRUCTIONS.
            18
            19                EVERYTHING THAT GOES UP MUST COME DOWWN.
      [EOB]
      *I 14


                             ^Z
            14                NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS IN A
      *%WH
             1                TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
             2                AND HE'LL BELIVE YOU.  TELL HIM A BENCH HAS WET PAINT
             3                ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
             4
             5                UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
             6                TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
             7                ORGANISM WILL DO AS IT DARN WELL PLEASES.
             8
             9                ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
            10
            11
            12                THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
            13                TO THE LEVEL OF MANAGEMETN.
            13.1
            14                NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS IN A
            15                HONEST DAYS WORK.
            16
            17                WHEN ALL ELSE FAILS READ THE INSTRUCTIONS.
            18
            19                EVERYTHING THAT GOES UP MUST COME DOWWN.
      [EOB]
```

# SOLUTIONS

```
c.  *S/METN/MENT/13
        13              TO THE LEVEL OF MANAGEMENT.
    1 substitution
    *S/BOSS/BOSS PUTTING/14
        14              NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS PUTTING IN A
    1 substitution
    *S/WW/W/19
        19              EVERYTHING THAT GOES UP MUST COME DOWN.
    1 substitution
    *%WH
        1               TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
        2               AND HE'LL BELIVE YOU.  TELL HIM A BENCH HAS WET PAINT
        3               ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
        4
        5               UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
        6               TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
        7               ORGANISM WILL DO AS IT DARN WELL PLEASES.
        8
        9               ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
        10
        11
        12              THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
        13              TO THE LEVEL OF MANAGEMENT.
        13.1
        14              NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS PUTTING IN A
        15              HONEST DAYS WORK.
        16
        17              WHEN ALL ELSE FAILS READ THE INSTRUCTIONS.
        18
        19              EVERYTHING THAT GOES UP MUST COME DOWN.
    [EOB]
d.  *M 19 TO 1
    1 line moved
    *0:5
        0.1             EVERYTHING THAT GOES UP MUST COME DOWN.
        1               TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
        2               AND HE'LL BELIVE YOU.  TELL HIM A BENCH HAS WET PAINT
        3               ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
        4
        5               UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
    *18:E
        18
    [EOB]
    *%WH
        0.1             EVERYTHING THAT GOES UP MUST COME DOWN.
        1               TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
        2               AND HE'LL BELIVE YOU.  TELL HIM A BENCH HAS WET PAINT
        3               ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
        4
        5               UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
        6               TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
        7               ORGANISM WILL DO AS IT DARN WELL PLEASES.
        8
        9               ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
        10
        11
        12              THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
        13              TO THE LEVEL OF MANAGEMENT.
        13.1
        14              NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS PUTTING IN A
        15              HONEST DAYS WORK.
        16
        17              WHEN ALL ELSE FAILS READ THE INSTRUCTIONS.
        18
    [EOB]
```

# SOLUTIONS

```
e. *COPY 14:15 TO E
   2 lines copied
   *18:E
       18
       19              NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS PUTTING IN A
       20              HONEST DAYS WORK.
   [EOB]
f. *HELP RES
   The RESEQUENCE (abbreviation: RES) command assigns new line numbers
   to a range of lines.

       Format:          RESEQUENCE [range] [/SEQUENCE:init:incr]

   The specified range of lines must be contiguous. When you do not specify
   a range, all lines in the  current buffer are resequenced.  If the
   new numbers assigned would cause duplicate or nonsequential line
   numbers, EDT will either disallow the command or  renumber lines
   beyond the specified range.


   Additional information available:
   /SEQUENCE
g. *RES /SEQ:100:10
   22 lines resequenced
   *%WH
      100              EVERYTHING THAT GOES UP MUST COME DOWN.
      110              TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
      120              AND HE'LL BELIVE YOU.  TELL HIM A BENCH HAS WET PAINT
      130              ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
      140
      150              UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
      160              TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
      170              ORGANISM WILL DO AS IT DARN WELL PLEASES.
      180
      190              ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
      200
      210
      220              THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
      230              TO THE LEVEL OF MANAGEMENT.
      240
      250              NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS PUTTING IN A
      260              HONEST DAYS WORK.
      270
      280              WHEN ALL ELSE FAILS READ THE INSTRUCTIONS.
      290
      300              NOTHING MOTIVATES A MAN MORE THAN TO SEE HIS BOSS PUTTING IN A
      310              HONEST DAYS WORK.
   [EOB]
h. *EXIT NEWER.TXT
   DRO:[305,303]NEWER.TXT;1 22 lines
   >LO
   Have a Good Afternoon
   21-DEC-81 13:58 TT56: logged off QUASAR
   >
```

## File and Directory Maintenance

## WRITTEN EXERCISES

1. The term "volume" refers to the _____ on which data is stored. The term "device" refers to the _____ on which a volume is mounted.

2. Each volume has two levels of directory files, the ___ and the ___.

3. The DCL command ___ _____ is used to change a user's default device and UFD. The ____ _____ command will display the user's current default settings.

4. When deleting files the _____ of a file specification must be supplied. If not supplied, the operating system will query you on each file that meets the file specification.

5. Using the directory listing for DK2:[305,303] (item 2 in Example 4-2 of the Student Workbook) write the file names that would be displayed by issuing the following commands:

>DIR *.MAC

_____

_____

_____

>DIR *.MAC;*

_____

_____

_____

_____

_____

>DIR PROG.*

_____

_____

_____

_____

# File and Directory Maintenance

## WRITTEN EXERCISES

```
>DIR PROG.*;*
```

_____

_____

_____

_____

_____

_____

_____

6.  Next to each file maintenance operation, write the letter that corresponds to the command best suited to accomplish it. Specify each command at least once.

Commands:

A.  COPY
B.  DELETE
C.  DIRECTORY
D.  DIRECTORY/PRINTER
E.  PURGE/KEEP:
F.  PRINT
G.  PURGE
H.  RENAME
I.  TYPE
J.  DELETE/QUERY

Operations:

\_\_\_\_\_    Display the contents of a file at your terminal.

\_\_\_\_\_    Display the contents of your default directory at your terminal.

\_\_\_\_\_    Remove a specified file from your default directory.

# File and Directory Maintenance

## WRITTEN EXERCISES

_____ Remove all but the most recent version of a specified file from your default directory.

_____ Create an exact duplicate of a file in your default directory.

_____ List the contents of the file at the default system printer.

_____ Add the contents of one file to another.

_____ Change the version number of a file to some specified value.

_____ Determine the number of blocks remaining for your use on your default disk.

_____ Display the name of each file in your default directory and remove or retain it by entering a "Y" or "N" at your terminal.

_____ Remove from your default directory all but the two most recent versions of your files of type LST.

_____ Display the contents of your default directory at the system printer (M-plus) systems only.

# File and Directory Maintenance

## WRITTEN EXERCISES

7. Next to each directory maintenance operation write the letter of the command best suited to perform the job. You may use each command more than once.

Commands:

A. Copy
B. Delete
C. Directory
D. Rename
E. Set Default
F. Set Protection
G. Show Protection
H. Show Default
I. Create/Directory
J. Delete/Directory


Operations:

_____ Display the name of your current default directory

_____ Display the contents of a directory

_____ Remove a directory from a voluem

_____ Add a directory to a volume

_____ Move files from one directory to another

_____ Change your current default directory

_____ Display the name of your current default devices

_____ Change your current default device

_____ Show the protection code of a file

_____ Change the file type of a file

_____ Delete a file from one directory

# File and Directory Maintenance

## SOLUTIONS

1.  The term "volume" refers to the magnetic media on which data
    is stored. The term "device" refers to the hardware on which
    a volume is mounted.

2.  Each volume has two levels of directory files, the MFD and the
    UFD.

3.  The DCL command SET DEFAULT is used to change a user's default
    device and UFD. The SHOW DEFAULT command will display the
    user's current default settings.

4.  When deleting files the version of a file specification must
    be supplied. If not supplied, the operating system will query
    you on each file that meets the file specification.

5.  Using the directory listing for DK2:[305,303] (item 2 in
    Example 4-2 of the Student Workbook) write the file names that
    would be displayed by issuing the following commands:


>DIR *.MAC
     XYZ.MAC;1
     VIP.MAC;5
     MACIO.MAC;10


>DIR *.MAC;*
     XYZ.MAC;1
     VIP.MAC;5
     MACIO.MAC;7
     MACIO.MAC;6
     MACIO.MAC;10

# File and Directory Maintenance

## SOLUTIONS

```
>DIR  PROG.*
        PROG.LST;1
        PROG.OBJ;2
        PROG.FTN;3
        PROG.TSK;1


>DIR  PROG.*;*
        PROG.FTN;1
        PROG.OBJ;1
        PROG.LST;1
        PROG.OBJ;2
        PROG.FTN;2
        PROG.FTN;3
        PROG.TSK;1
```

6.  Next to each file maintenance operation, write the letter that
    corresponds to the command best suited to accomplish it.
    Specify each command at least once.


    Commands:


    A.  COPY
    B.  DELETE
    C.  DIRECTORY
    D.  DIRECTORY/PRINTER
    E.  PURGE/KEEP:
    F.  PRINT
    G.  PURGE
    H.  RENAME
    I.  TYPE
    J.  DELETE/QUERY

# File and Directory Maintenance

## SOLUTIONS

Operations:

I     Display the contents of a file at your terminal.

C     Display the contents of your default directory at your terminal.

B     Remove a specified file from your default directory.

G     Remove all but the most recent version of a specified file from your default directory.

A     Create an exact duplicate of a file in your default directory.

E     List the contents of a file at the default system printer.

A     Add the contents of one file to another.


H     Change the version number of a file to some specified value.

C     Determine the number of blocks remaining for your use on your default disk.

J     Display the name of each file in your default directory and remove or retain it by entering a "Y" or "N" at your terminal.

F     Remove from your default directory all but the two most recent versions of files of type LST.

D     Display the contents of your default directory at the system printer (M-PLUS Systems only)>

# File and Directory Maintenance

## SOLUTIONS

7.  Next to each directory maintenance operation write the letter of the command best suited to perform the job. You may use each command more than once.

    Commands:

    A.  Copy
    B.  Delete
    C.  Directory
    D.  Rename
    E.  Set Default
    F.  Set Protection
    G.  Show Protection
    H.  Show Default
    I.  Create/Directory
    J.  Delete/Directory

    Operations

    H       Display the name of your current default
            directory.

    J       Remove a directory from a volume

    I       Add a directory to a volume

    A       Move files from one directory to another

    F       Change your current default directory

    E       Change the protection code of a directory
            file

    B       Display the name of your current default
            device

    F       Change your current default device

    G       Show the protection code of a file

    D       Change the file type of a file

    B       Delete a file from a directory

# File and Directory Maintenance

## LABORATORY EXERCISES

Exercise 1 involves use of the DIR, TYPE, PRINT, COPY, RENAME, PURGE, DELETE and SET PROTECTION commands. Obtain Help on each command before performing the exercise pertaining to the command.

1. Using the directory command and its qualifiers, do the following:

   a) Display all the files in your user file directory on your terminal.

      **1** What device are you working on?

      **2** How many files do you own?

      **3** How many blocks are used for your files?

      **4** How many blocks were allocated for your files?

   b) Display all the indirect command files in your UFD.

   c) Display all versions of NEW.TXT in your UFD.

   d) What is the protection on the latest version of NEW.TXT? Who is the owner? What is the Header Number and Sequence Number?

   e) Without listing all your files, obtain the number of blocks used in your UFD.

   f) Obtain the number of free blocks on the volume.

   g) Obtain a directory of the following:

## File and Directory Maintenance

## LABORATORY EXERCISES

1 UFD of someone else in your group.

2 UFD of someone in the system group.

3 UFD of someone outside of either your group or the system group.

4 LB:[1,2]*.HLP.  What are these files?  What physical device is LB:?

5 SY:[0,0]

6 LB:[200,1]

2. Using the TYPE command, display the contents of NEW.TXT on your terminal.

3. Using the PRINT command, obtain a hardcopy of NEW.TXT.

4. Using the COPY command, make another copy of the file, NEW.TXT, and name that copy LATE.TXT.

5. Using the RENAME command, rename the latest version of NEW.TXT to LATEST.TXT.  What is the difference between the results of the COPY command and the results of the RENAME command?

6. Using the PURGE command, purge your UFD of all but the latest copy of each file.

7. Using the selective DELETE command, delete some but not all (especially not your LOGIN.CMD file) of your .CMD files.

8. Set the protection on LATEST.TXT so that only you can read, write, extend or delete the file.

# File and Directory Maintenance

# SOLUTIONS

```
>;      CHAPTER 4
>;
>;      LABORATORY EXERCISE SOLUTIONS
>;
>;
>;
1. a) >DIR
```

❶

```
Directory DR0:[305,303]
22-DEC-81 12:54

FILES.MAC;1          2.              03-SEP-81 14:51
MYLIB.MLB;1          9.          C   03-SEP-81 14:54
DOC.MRO;1            1.              05-OCT-81 09:45
LIBRARY.CMD;2        1.              03-SEP-81 15:07
MAIN.FOR;1           4.              07-OCT-81 14:35
BUILD.CMD;1          1.              07-OCT-81 14:33
COMPARE.TXT;1        2.              14-SEP-81 11:15
CMP.TXT;2            3.              14-SEP-81 11:17
DIF.TXT;3            2.              15-SEP-81 15:22
LOGOUT.CMD;1         1.              17-SEP-81 14:37
INPUT.CMD;1          2.              22-SEP-81 13:38
SYMBOLS.CMD;1        2.              22-SEP-81 13:38
CONTROL.CMD;1        2.              22-SEP-81 13:38
SPECIAL.CMD;1        2.              22-SEP-81 13:38
OPERATING.CMD;2      3.              22-SEP-81 14:19
ASSEMBLE.CMD;2       1.              22-SEP-81 14:24
TEST.LST;2           2.              22-SEP-81 14:24
LOGICAL.CMD;1        2.              23-SEP-81 10:48
TEST.MAC;2           1.              25-SEP-81 09:24
TEST.OBJ;3           1.              25-SEP-81 09:24
A0.MAC;1             1.              02-OCT-81 14:39
VIP.TSK;1            7.          C   02-OCT-81 14:04
A1.MAC;1             1.              02-OCT-81 14:39
B0.MAC;1             1.              02-OCT-81 14:39
B1.MAC;1             1.              02-OCT-81 14:39
C0.MAC;1             1.              02-OCT-81 14:39
A2.MAC;2             1.              02-OCT-81 14:40
B2.MAC;1             1.              02-OCT-81 14:40
A0.OBJ;1             1.              02-OCT-81 14:40
A1.OBJ;1             1.              02-OCT-81 14:40
A2.OBJ;1             1.              02-OCT-81 14:41
B0.OBJ;1             1.              02-OCT-81 14:41
B1.OBJ;1             1.              02-OCT-81 14:41
B2.OBJ;1             1.              02-OCT-81 14:41
C0.OBJ;1             1.              02-OCT-81 14:41
CNTRL.MAC;1          1.              02-OCT-81 14:41
CNTRL.OBJ;1          1.              02-OCT-81 14:41
TEST.TSK;7           7.          C   03-OCT-81 08:56
CNTRL.ODL;1          1.              02-OCT-81 15:09
CNTRL.TSK;2          36.         C   02-OCT-81 15:12
CONTROL.ODL;6        1.              02-OCT-81 15:20
CONTROL.TSK;5        36.         C   02-OCT-81 15:22
CONTROL.MAP;2        15.             02-OCT-81 15:22
TEST.MAP;1           3.              03-OCT-81 08:56
EX2D4T.LST;4         8.              15-OCT-81 16:08
SORT.MAC;1           2.              16-OCT-81 09:37
SORT.MAC;2           2.              16-OCT-81 09:37
TEMP.TXT;1           2.              19-OCT-81 08:33
```

## SOLUTIONS

```
SORT.MAC;3          2.        16-OCT-81 09:37
SORT.MAC;4          2.        16-OCT-81 09:37
SORT.MAC;5          2.        16-OCT-81 09:37
SORT.MAC;6          2.        16-OCT-81 09:38
ODTST.MAC;1         1.        24-NOV-81 13:54
ODTST.OBJ;1         1.        24-NOV-81 13:56
ODTST.OBJ;2         1.        24-NOV-81 13:59
ODTST.LST;1         3.        24-NOV-81 13:59
DIF.TXT;4           1.        11-DEC-81 09:58
SOLUTIONS.CMD;1     5.        18-DEC-81 15:02
SAMPLE.TXT;1        1.        15-DEC-81 10:58
SAMPLE.TXT;2        1.        15-DEC-81 10:59
SAMPLE.TXT;3        1.        15-DEC-81 11:00
SAMPLE.TXT;4        1.        15-DEC-81 11:02
TEMP.TXT;2          2.        15-DEC-81 11:15
SAMPLE.TXT;5        2.        15-DEC-81 16:51
SOLUTIONS.CMD;2     5.        21-DEC-81 12:58
CHAPSOL.CMD;1       2.        22-DEC-81 11:01
NEW.TXT;1           1.        21-DEC-81 13:48
NEWER.TXT;1         2.        21-DEC-81 13:57

Total of 218./377. blocks in 68. files
>;        ❸   ❹          ❷
>;
>;
```

b) >DIR *.CMD

```
Directory DR0:[305,303]
22-DEC-81 12:54

LIBRARY.CMD;2       1.        03-SEP-81 15:07
BUILD.CMD;1         1.        07-OCT-81 14:33
LOGOUT.CMD;1        1.        17-SEP-81 14:37
INPUT.CMD;1         2.        22-SEP-81 13:38
SYMBOLS.CMD;1       2.        22-SEP-81 13:38
CONTROL.CMD;1       2.        22-SEP-81 13:38
SPECIAL.CMD;1       2.        22-SEP-81 13:38
OPERATING.CMD;2     3.        22-SEP-81 14:19
ASSEMBLE.CMD;2      1.        22-SEP-81 14:24
LOGICAL.CMD;1       2.        23-SEP-81 10:48
SOLUTIONS.CMD;2     5.        21-DEC-81 12:58
CHAPSOL.CMD;1       2.        22-DEC-81 11:01

Total of 24./38. blocks in 12. files

>;
```

c) >;
```
>DIR NEW.TXT;*

Directory DR0:[305,303]
22-DEC-81 12:55

NEW.TXT;1           1.        21-DEC-81 13:48

Total of 1./5. blocks in 1. file
```

# SOLUTIONS

```
   >;
   >;        4.
d) >
   >DIR/FULL NEW.TXT


   Directory DR0:[305,303]
   22-DEC-81 13:12

   NEW.TXT;5            (12673,32)       2./5.               15-DEC-81 16:51
      [305,303][RWED,RWED,RWED,R]

   Total of 2./5. blocks in 1. file

   >;
   >;
e) >DIR/SUMMARY


   Storage used/allocated for Directory DR0:[305,303]
   22-DEC-81 12:55


   Total of 218./377. blocks in 68. files

   >;
   >;
f) >DIR/FREE

   DR0: has 540600. blocks free, 467400. blocks used out of 1008000.
   Largest contiguous space = 229377. blocks
   19273. file headers are free, 16727. headers used out of 36000.

   >;
   >;
   >;
```

## SOLUTIONS

```
      >
      >
      >
g)  >DIR [305,301]
```

① `Directory DR0:[305,301]`
`22-DEC-81 13:13`

```
    TASKD.TSK;1        8.      C   03-SEP-81 12:04
    SPRGDEV.CMD;1      2.          03-SEP-81 12:09
    PROGMACS.MLB;1     62.         03-SEP-81 12:13
    PROGSUBS.OLB;1     24.     C   03-SEP-81 12:13
    TASKC.OBJ;2        3.          03-SEP-81 12:14
    TASKC.MAC;2        3.          03-SEP-81 12:14
    TASKC.LST;2        6.          03-SEP-81 12:14
    TASKC.TSK;3        8.      C   03-SEP-81 12:14
    SPAWNT2.MAC;1      4.          04-SEP-81 14:49
    SPAWNT2.TSK;1      8.      C   04-SEP-81 14:49
    PASSIT.MAC;1       3.          08-SEP-81 11:38
    PMSPAWN.TSK;1      9.      C   08-SEP-81 11:38
    PASSIT.OBJ;1       3.          08-SEP-81 11:39
    PASSIT.LST;1       7.          08-SEP-81 11:39
    PASSIT.TSK;1       8.      C   08-SEP-81 11:39
    PASSIT.OBJ;2       3.          08-SEP-81 11:41
    PASSIT.MAC;2       3.          08-SEP-81 11:41
    PASSIT.LST;2       7.          08-SEP-81 11:41
    PASSIT.TSK;2       8.      C   08-SEP-81 11:41
    SPAWN.TSK;1        8.      C   08-SEP-81 11:42
    SPAWN.MAC;1        4.          08-SEP-81 12:07
    SPAWN.TSK;2        9.      C   08-SEP-81 12:07
    SPAWN.TSK;3        9.      C   08-SEP-81 12:41
    PASSIT.MAC;3       4.          08-SEP-81 12:41
    PASSIT.OBJ;3       3.          08-SEP-81 12:41
    PASSIT.LST;3       8.          08-SEP-81 12:41
    PASSIT.TSK;3       8.      C   08-SEP-81 12:42
    PMSPAWN.TSK;2      9.      C   08-SEP-81 12:44
    PMSPAWNT.TSK;1     9.      C   08-SEP-81 12:53
    GSPAWN.TSK;1       9.      C   08-SEP-81 15:39
    RECV1.MAC;1        3.          09-SEP-81 13:05
    SEND1.MAC;1        2.          09-SEP-81 13:05
    RECV1.TSK;1        8.      C   09-SEP-81 13:05
    SEND1.TSK;1        8.      C   09-SEP-81 13:05
    RECV2.TSK;1        8.      C   09-SEP-81 13:08
    SEND2.TSK;1        8.      C   09-SEP-81 13:08
    RECV2.MAC;1
```

## SOLUTIONS

**2** >
>DIR [7,5]


Directory DR0:[7,5]
22-DEC-81 13:14

```
SWITCH.INI;1          1.            09-JAN-78 14:04
FRCASM.CMD;1          1.            09-JAN-78 14:04
ZAPMSG.TXT;1          2.            09-FEB-79 16:33
FRCBLD.CMD;2          1.            19-OCT-79 08:38
160BPI.HOW;1          1.            09-JAN-78 14:05
KMCLPBLD.CMD;3        1.            06-DEC-78 10:00
MAKBIGFIL.FTN;1       1.            21-FEB-78 15:48
KMCLP.CMO;17          1.            06-DEC-78 09:55
KMCLPASM.CMD;1        1.            06-DEC-78 10:01
CHESS.MAC;2           170.          03-FEB-78 14:18
CHESS.HLP;1           4.            03-FEB-78 14:18
CHESS.TSK;1           24.      C    03-FEB-78 14:18
ODTPRVBLD.CMD;1       1.            30-OCT-78 08:47
PRINT.TSK;3           130.     C    14-FEB-78 08:26
SALPHA.DAT;1          105.     L    14-FEB-78 08:26
PRINT.BAS;1           12.           14-FEB-78 08:26
SCRIPT.FOR;1          59.           21-FEB-78 10:58
SCRIPT.FTN;1          59.           21-FEB-78 13:03
GETLBN.BAS;3          1.            30-MAR-78 16:42
KTEST.FTN;1           4.            19-APR-78 13:17
FIND.TEC;4            1.            12-MAY-78 09:56
ODTPRE.MAC;4          1.            02-MAR-79 16:25
COPYRIGHT.CMD;1       1.            04-JAN-77 09:18
DIRTYJOKE.CMD;1       2.            01-FEB-78 16:49
MAKTC.CMD;1           1.            24-JAN-78 10:09
MEMO.CMD;5            1.            30-JUN-78 16:13
PNGBLD.CMD;1          1.            08-JAN-78 15:09
T.CMD;2               1.            29-OCT-78 15:58
T.CMD;1               1.            29-AUG-78 10:51
TCBLD.CMD;1           1.            24-JAN-78 10:10
VTKASM.CMD;1          1.            03-DEC-78 17:43
VTKBLD.CMD;3          1.            03-DEC-78 17:43
VTKODTBLD.CMD;2       1.            07-MAY-78 11:11
VTKODTBLD.CMD;2       1.            29-AUG-78 10:23
```

## SOLUTIONS

```
    >
  3 >DIR [200,200]


   Directory DR0:[200,200]
   22-DEC-81 13:15

   IECLIB.IMG;1        58.       14-AUG-79 16:32
   PEAINS.CMD;1        11.       08-SEP-80 10:41
   COPYNEW.CMD;1        1.       07-FEB-80 15:01
   COPYNEW.CMD;2        1.       21-MAR-80 15:57
   COPYNEW.CMD;3        1.       24-MAR-80 11:35
   AUTOPATCH.CMD;1     58.       19-MAR-81 13:50
   DIR.DIR;1           15.       13-FEB-80 13:23
   FCSRES.OBJ;1         1.       11-JUN-80 11:33
   COPYNEW.CMD;4        1.       23-JUN-80 08:26
   CHRIS.LIS;1          1.       06-AUG-80 10:52
   CHRIS.COM;1          1.       06-AUG-80 10:49
   BAS.CMD;1            1.       20-AUG-80 14:01
   FCSRS1.CMD;1         3.       10-OCT-80 08:55
   TECFOO.TMP;1         1.       20-MAR-81 13:01
   RSOVL1.MAC;1         1.       10-OCT-80 08:56
   FCSRS1.CMD;2         3.       10-OCT-80 08:56
   RSOVL1.OBJ;1         1.       10-OCT-80 08:56
   RSOVL2.MAC;1         1.       10-OCT-80 08:56
   RSOVL2.OBJ;1         1.       10-OCT-80 08:56
   FCSRES.MAC;2        11.       20-OCT-80 14:31
   11SGEN2.CMD;2      107.       20-OCT-80 14:32
   11SCOPY.CMD;2        6.       20-OCT-80 14:32
   NUM.DAT;1            1.       29-OCT-80 10:53
   AUTOPATCH.DOC;1    118.       23-NOV-80 13:04
   SYSSAVED.DAT;6       3.       19-FEB-81 08:45
   SYSGEN3.CMD;1       13.       26-FEB-81 16:32
   POOLTEST.DAT;1       0.       04-MAR-81 13:24
   POOLTEST.DAT;2       4.       04-MAR-81 13:55
   MARK.TST;1           1.       09-MAR-81 09:12
   MARK.TST;2           1.       09-MAR-81 09:13
   MARK.TST;3           0.       09-MAR-81 09:14
   DUTEST.CMD;1         4.       13-MAR-81 14:34
   DUTEST.DAT;1         4.       13-MAR-81 14:34
```

## SOLUTIONS

**4**
```
>'
>;
>DIR LB:[1,2]*.HLP


Directory DR0:[1,2]
22-DEC-81 12:55

HELPTTY.HLP;1        5.           02-DEC-81 14:30
CFE.HLP;5            21.          31-JAN-80 00:43
CFEDEF.HLP;4         37.          31-JAN-80 00:43
CFELIS.HLP;4         5.           31-JAN-80 00:43
DAC.HLP;3            7.           13-FEB-79 15:38
QUASARHLP.HLP;6      1.           04-DEC-81 11:28
DCLARCH.HLP;1        33.    C     23-OCT-81 15:41
HELPSTAT.HLP;233     1.           22-DEC-81 08:02
DEBUG16.HLP;1        2.           14-JUL-81 14:36
DEMO.HLP;1           2.           01-DEC-81 11:54
HELPDISK.HLP;12      6.           24-MAR-81 16:47
HELPDOC.HLP;3        3.           26-NOV-80 12:40
HELPMAIL.HLP;77777   18.          01-JAN-81 14:44
LCTOP.HLP;1          3.           29-NOV-79 10:43
NCP.HLP;5            31.          31-JAN-80 00:43
NCPCLE.HLP;5         9.           31-JAN-80 00:44
NCPLOA.HLP;5         7.           31-JAN-80 00:44
NCPLOO.HLP;5         7.           31-JAN-80 00:44
NCPSET.HLP;5         33.          31-JAN-80 00:44
NCPSHO.HLP;5         7.           31-JAN-80 00:44
NCPTRI.HLP;5         4.           31-JAN-80 00:44
NCPZER.HLP;5         2.           31-JAN-80 00:44
NET.HLP;2            1.           04-DEC-79 16:57
NEWWEEK.HLP;5        1.           10-NOV-81 11:32
NEXTWEEK.HLP;7       1.           10-NOV-81 11:31
NFT.HLP;5            12.          31-JAN-80 00:44
```

# File and Directory Maintenance

## SOLUTIONS

**5** >DIR SY:[0,0]


Directory DR0:[0,0]
22-DEC-81 12:56

```
INDEXF.SYS;1        34017.        08-OCT-81 13:25
BITMAP.SYS;1        248.          08-OCT-81 13:25
BADBLK.SYS;1        50.           08-OCT-81 13:25
000000.DIR;1        13.     C     08-OCT-81 13:25
CORIMG.SYS;1        0.      C     08-OCT-81 13:25
RSX11.SYS;1
Read attributes error - Privilege violation
001001.DIR;1        8.      C     08-OCT-81 13:25
001002.DIR;1        16.     C     08-OCT-81 13:25
001003.DIR;1        1.      C     08-OCT-81 13:25
001004.DIR;1        2.      C     08-OCT-81 13:25
001006.DIR;1        1.      C     08-OCT-81 13:25
001020.DIR;1        1.      C     08-OCT-81 13:25
001024.DIR;1        2.      C     08-OCT-81 13:25
001030.DIR;1        1.      C     08-OCT-81 13:25
001034.DIR;1        1.      C     08-OCT-81 13:25
001050.DIR;1        1.      C     08-OCT-81 13:25
001054.DIR;1        6.      C     08-OCT-81 13:25
001060.DIR;1        1.      C     08-OCT-81 13:25
001064.DIR;1        1.      C     08-OCT-81 13:25
002007.DIR;1        1.      C     08-OCT-81 13:25
002020.DIR;1        1.      C     08-OCT-81 13:25
002054.DIR;1        6.      C     08-OCT-81 13:25
002200.DIR;1        1.      C     08-OCT-81 13:25
002310.DIR;1        1.      C     08-OCT-81 13:25
002320.DIR;1        1.      C     08-OCT-81 13:25
002325.DIR;1        1.      C     08-OCT-81 13:25
002330.DIR;1        1.      C     08-OCT-81 13:25
003054.DIR;1        7.      C     08-OCT-81 13:25
004054.DIR;1        7.      C     08-OCT-81 13:25
005054.DIR;1        1.      C     08-OCT-81 13:25
306016.DIR;1        1.      C     08-OCT-81 13:25
007004.DIR;1        1.      C     08-OCT-81 13:25
007005.DIR;1        3.      C     08-OCT-81 13:25
100200.DIR;1        1.      C     08-OCT-81 13:25
```

## SOLUTIONS

```
>;
>;
>DIR LB:[200,1]
DIR -- Cannot find directory file
LB:[200,1]
>;
>;
>;      2.
```
🔵 **6**

2.
```
>TYPE NEW.TXT
TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
AND HE'LL BELIVE YOU.  TELL HIM A BENCH HAS WET PAINT
ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.

UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
ORGANISM WILL DO AS IT DARN WELL PLEASES.

ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.


THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
TO THE LEVEL OF MANAGEMETN.
>;
>;      3.
```

3.
```
>PRINT NEW.TXT
PRI - Job 646, name "NEW       ", submitted to queue "PRINT "
>;
>;      4.
```

4.
```
>COPY NEW.TXT LATE.TXT
>;
>;
```

5.
```
>RENAME NEW.TXT LATEST.TXT
>;
>;
```

6.
```
>PURGE *.*
>;
>;

>.
```

7.
```
>DEL/QUERY *.CMD
Delete file    DR0:[305,303]LIBRARY.CMD;2       [Y/N/G/Q]? Y
Delete file    DR0:[305,303]BUILD.CMD;1         [Y/N/G/Q]? Y
Delete file    DR0:[305,303]LOGOUT.CMD;1        [Y/N/G/Q]? Y
Delete file    DR0:[305,303]INPUT.CMD;1         [Y/N/G/Q]? N
Delete file    DR0:[305,303]SYMBOLS.CMD;1       [Y/N/G/Q]? N
Delete file    DR0:[305,303]CONTROL.CMD;1       [Y/N/G/Q]? N
Delete file    DR0:[305,303]SPECIAL.CMD;1       [Y/N/G/Q]? Q
>.
>.
```

8.
```
>SET PROTECTION
File? LATEST.TXT
Code? (SY:,OW:RWED,GR:,WO:)
>.
>.
>.
>.
```

# Program Development

## WRITTEN EXERCISES

1. The process of entering code into a file, compiling it and linking it is called _____ _____.

2. All _____ run on the system have been through the program development process.

3. There are ___ basic types of programming languages in use today.

4. A programming language where each source statement is translated into machine code and then executed before processing the next source statement is called an _____ language.

5. In an _____ language, each source statement generates one machine instruction.

6. In a _____ language, each source statement may generate one or more machine instructions.

7. The _____ command is used to invoke an assembly of a MACRO-11 source file.

8. Write the command to perform an assembly on a file called VIP.MAC producing a cross-reference list.

_____

9. Write the command to perform an assembly on a file called VIP.MAC, and to search a library file called USERMAC.MLB.

_____

10. Write the command to perform an assembly on a file called VIP.MAC, creating just a listing file.

_____

11. The _____ command is used to compile a FORTRAN source file.

49

## Program Development

## WRITTEN EXERCISES

12. Write the command that invokes the FORTRAN IV-PLUS compiler to compile a source file called FUNCTIONS.FTN.

    _____

13. Write the command that invokes the FORTRAN-77 compiler, producing both a
    listing and object files from the source file called ROUTINES.FTN.

    _____

14. Write the command that invokes the FORTRAN IV compiler to compile the
    following input files: FUNCTIONS.FTN, ROUTINES.FTN, SUBROUT.FTN.

    _____

15. Output from the FORTRAN compiler consists of:

    _____
    _____

16. Object files must be _____ to produce a file that can be run on the system.

17. The ____ command is used to invoke the Task Builder.

18. Input to the Task Builder includes:

    _____
    _____

19. Output from the Task Builder includes:

    _____
    _____
    _____

20. To execute, a task must be _____ in the STD.

21. The ___ command will install a task in the STD, run the task and then remove the task entry from the STD.

# Program Development

## SOLUTIONS

1. The process of entering code into a file, compiling it and linking it is called program development.

2. All tasks run on the system have been through the program development process.

3. There are 3 basic types of programming languages in use today.

4. A programming language where each source statement is translated into machine code and then executed before processing the next source statement is called an Interpreted language.

5. In an Assembled language, each source statement generates one machine instruction.

6. In a Compiled language, each source statement may generate one or more machine instructions.

7. The MACRO command is used to invoke an assembly of a MACRO-11 source file.

8. Write the command to perform an assembly on a file called VIP.MAC producing a cross-reference list.

    MACRO/CROSSREFERENCE VIP

9. Write the command to perform an assembly on a file called VIP.MAC, and to
search a library file called USERMAC.MLB.

    MACRO USERMAC.MLB/LIBRARY,VIP

10. Write the command to perform an assembly on a file called VIP.MAC, creating
just a listing file.

    MACRO/LIST/NOOBJECT VIP

11. The FORTRAN command is used to compile a FORTRAN source file.

51

# Program Development

## SOLUTIONS

12. Write the command that invokes the FORTRAN IV-PLUS compiler to compile a
    source file called FUNCTIONS.FTN.

    <u>FORTRAN/F4P FUNCTIONS</u>

13. Write the command that invokes the FORTRAN-77 compiler, producing both a
    listing and object files from the source file called ROUTINES.FTN.

    <u>FORTRAN/F77/LIST ROUTINES</u>

14. Write the command that invokes the FORTRAN IV compiler to compile the
    following input files: FUNCTIONS.FTN, ROUTINES.FTN, SUBROUT.FTN.

    <u>FORTRAN FUNCTIONS,ROUTINES,SUBROUT</u>

15. Output from the FORTRAN compiler consists of:

    a)      <u>an object file</u>
    b)      <u>a listing file</u>

16. Object files must be <u>linked</u> to produce a file that can be
    run on the system.

17. The <u>LINK</u> command is used to invoke the Task Builder.

18. Input to the Task Builder includes:

    <u>object file(s)</u>
    <u>object library</u> file(s)

19. Output from the Task Builder includes:

    <u>task image file</u>
    <u>map file</u>
    <u>symbol table file</u>

20. To execute, a task must be <u>installed</u> in the STD.

21. The <u>RUN</u> command will install a task in the STD, run the task,
    and then remove the task entry from the STD.

# Program Development

## LABORATORY EXERCISES

Students should perform either exercises 1 and 2 and/or 3 and 4, and then proceed to 5 through 7.

1.  Assemble PROGRAM.MAC creating a listing file and an object file.

2.  Obtain a hardcopy of the listing file generated in exercise 1 and identify the following:

    a.  Source statement line numbers
    b.  Program addresses
    c.  Machine code
    d.  Source lines
    e.  Symbol table
    f.  Assembly diagnostics

3.  Compile PROGRAM.FTN creating a listing file and an object file, using the applicable language command qualifier for your system.

4.  Obtain a hardcopy of the listing file generated in exercise 3 and identify the following:

    a.  Source statement line numbers
    b.  Source statements
    c.  Storage map

5.  Using the object module generated in exercise 3, build your task using the Link command in the following manner:

    • Generate a map file.

    • Generate a task image file using your name as the file name and TSK as the extension.

**Program Development**

## LABORATORY EXERCISES

6.  Obtain a hardcopy of the map file generated in exercise 5  and
    identify the following:

    a.   The task name

    b.   The partition in which the task will be loaded

    c.   The UIC under which the task will be  run  for  time-based
         scheduled requests

    d.   Task size


7.  Using the RUN command, run the task generated in exercise 5.

# Program Development

# SOLUTIONS

```
>;
>;
1.  >;      SOLUTION - EXERCISE #1
>;
>MACRO/LIST/NOWIDE  PROGRAM
>;
```

```
2.
   a.1                                    .TITLE  SYNCHQ
      2                                    .IDENT  /01/
      3                                    .ENABL  LC               ; ENABLE LOWER CASE
      4                            ;+
      5                            ; FILE SYNCHQ.MAC
      6                            ;
      7                            ; This task reads a line of text from the terminal,
      8                            ; converts all upper case characters to lower case, and
      9                            ; prints the converted message back at the terminal. It
     10                            ; uses synchronous QIO directives.
     11                            ;-
     12                                    .MCALL  QIOW$C,QIOW$S,EXIT$S ; External system
     13                                                             ;  macros
     14  b.
     15 000000             IOSB:   .BLKW   2                        ; I/O Status Block
     16 000004             BUFF:   .BLKB   80.                      ; Text buffer
     17           c.                      d.
     18 000124 005005      START:  CLR     R5                       ; Error Count
     19 000126 005004              CLR     R4                       ; Error indicator - 0
     20                                                             ;   means directive error
     21                                                             ;   (DSW in R3), nes
     22                                                             ;   means I/O error
     23                                                             ;   (I/O status in R3)
     24 000130              QIOW$C  IO.RVB,5,1,,IOSB,,<BUFF,80.> ; Issue
     25                                                             ;  read
     26 000136 103472               BCS     ERR1                    ; Branch on dir error
     27 000140 105767 177634        TSTB    IOSB                    ; Check for I/O error
     28 000144 002461               BLT     ERR1A                   ; Branch on I/O error
     29 000146 016700 177630        MOV     IOSB+2,R0               ; Get count of characters
     30                                                             ;  typed in
     31 000152 005001               CLR     R1                      ; Offset into buffer to
     32                                                             ;  character
     33 000154 126127 000004' 000101 LOOP:  CMPB  BUFF(R1),#'A      ; Check for upper case
     34                                                             ;  ASCII character
     35 000162 002412               BLT     NEXT                    ; Branch if below range
     36 000164 126127 000004' 000132 CMPB   BUFF(R1),#'Z
     37 000172 003006               BGT     NEXT                    ; Branch if above range
     38                            ; Here if upper case, move to register R2 and convert
     39 000174 116102 000004'       MOVB    BUFF(R1),R2             ; Move to register
     40 000200 062702 000040        ADD     #32.,R2                 ; Convert to lower case
     41 000204 110261 000004'       MOVB    R2,BUFF(R1)             ; Replace in message
     42 000210 005201      NEXT:    INC     R1                      ; Increment offset into
     43                                                             ;  buffer to next char
     44 000212 077020               SOB     R0,LOOP                 ; Decrement count of
     45                                                             ;  characters left to check
     46 000214              QIOW$S  #IO.WVB,#5,#1,,#IOSB,,<#BUFF,IOSB+2,#40>
     47                                                             ; Write text
     48 000270 103414               BCS     ERR2                    ; Branch on dir error
     49 000272 105767 177502        TSTB    IOSB                    ; Check for I/O error
     50 000276 002403               BLT     ERR2A                   ; Branch on I/O error
     51 000300              EXIT$S                                  ; Exit
     52                            ;
     53                            ; Error code
     54                            ;
     55 000306 005205      ERR2A:   INC     R5                      ; Up error count - 2nd QIO
     56 000310 005205      ERR1A:   INC     R5                      ;                 - 1st QIO
     57 000312 116703 177462        MOVB    IOSB,R3                 ; I/O error, I/O status
```

# Program Development

# SOLUTIONS

```
58                                                      ; to R3.
59 000316  005304                    DEC    R4          ; Negative value in R4
60                                                      ; means I/O error
61 000320  000004                    IOT                ; Trap and display
62                                                      ; resisters
63 000322  005205          ERR2:     INC    R5          ; Up error count - 2nd QIO
64 000324  005205          ERR1:     INC    R5          ;             - 1st QIO
65 000326  016703  000000G           MOV    $DSW,R3     ; Directive error. DSW
66                                                      ; to R3, leave R4=0.
67 000332  000004                    IOT                ; Trap and display
68                                                      ; resisters
69         000124'                   .END START
```

SYMBOL TABLE

```
BUFF   000004R    IOSB    000000R    Q.IOAE= 000012   Q.IOPL= 000014   $DSW  = ****** GX
ERR1   000324R    IO.RVB= ****** GX  Q.IOEF= 000006   Q.IOPR= 000007   $$$   = 000000R     002
ERR1A  000310R    IO.WVB= ****** GX  Q.IOFN= 000002   Q.IOSB= 000010   $$$ARG= 000002
ERR2   000322R    LOOP    000154R    Q.IOLU= 000004   START   000124R  $$$OST= 000014
ERR2A  000306R    NEXT    000210R
```

```
. ABS.  000000    000
        000334    001   f.
$DPB$$  000030    002
ERRORS DETECTED:   0
```

```
VIRTUAL MEMORY USED: 9396 WORDS  ( 37 PAGES)
DYNAMIC MEMORY: 10042 WORDS  ( 38 PAGES)
ELAPSED TIME:  00:00:23
PROGRAM,PROGRAM/SP=PROGRAM
```

# Program Development

# SOLUTIONS

3.  ```
    >FORTRAN/LIST PROGRAM
    SYNCHQ
    >
    >
    >
    ```

4.  ```
    FORTRAN IV      V02.2-1          SUN 10-JAN-82 11:44:59          PAGE 001
    ```
a.  ```
    PROGRAM,PROGRAM/SP/LI:3=PROGRAM
    0001          PROGRAM      SYNCHQ
              C
              C FILE SYNCHQ.FTN
              C
              C This program reads a line of text from the terminal,
              C converts any upper case characters to lower case and
              C prints the converted message back at the terminal.
              C It uses synchronous QIO directives.
    ```
b.  ```
              C
    0002          BYTE IOSB(4),IBUF(80)
    0003          DIMENSION IPAR(6)
    0004          EQUIVALENCE (NUM,IOSB(3))
    0005          DATA IOWVB/"11000/
    0006          DATA IORVB/"10400/
    0007          DATA IVFC/"40/
              C Set up values for the QIO
    0008          IUNIT=5
    0009          IPAR(2)=80
    0010          IPAR(3)=IVFC
              C Get the address of the I/O buffer
    0011          CALL GETADR(IPAR(1),IBUF(1))
              C     Issue the QIO
    0012          CALL WTQIO(IORVB,IUNIT,1,,IOSB,IPAR,IDS)
              C Check the directive and I/O statuses
    0013          IF (IDS .LT. 0) GO TO 800
    0015          IF (IOSB(1) .LT. 0) GO TO 810
              C Check for uppercase characters and convert them to lowercase
    0017          DO 100 I=1,NUM
    0018          IF (IBUF(I) .LT. 'A') GO TO 100
    0020          IF (IBUF(I) .GT. 90) GO TO 100          IZ is 90(10)
    0022          IBUF(I)=IBUF(I)+32
    0023  100     CONTINUE
              C Place the number of characters to write in the I/O parameter list
    0024          IPAR(2)=NUM
              C Write the converted line to the terminal
    0025          CALL WTQIO(IOWVB,IUNIT,1,,IOSB,IPAR,IDS)
              C Check directive and I/O status
    0026          IF (IDS .LT. 0) GO TO 820
    0028          IF (IOSB(1) .LT. 0) GO TO 830
    0030          GO TO 850
    0031  800     WRITE(5,900)IDS
    0032          GO TO 850
    0033  810     WRITE(5,910)IOSB(1)
    0034          GO TO 850
    0035  820     WRITE(5,920)IDS
    0036          GO TO 850
    0037  830     WRITE(5,930)IOSB(1)
    0038  850     CALL EXIT
    0039  900     FORMAT(' DIRECTIVE ERROR ON READ, CODE = ',I4)
    0040  910     FORMAT(' I/O ERROR ON READ, CODE = ',I4)
    0041  920     FORMAT(' DIRECTIVE ERROR ON WRITE, CODE = ',I4)
    0042  930     FORMAT(' I/O ERROR ON WRITE, CODE = ',I4)
    0043          END
    ```

# SOLUTIONS

C. FORTRAN IV STORAGE MAP FOR PROGRAM UNIT SYNCHQ

    LOCAL VARIABLES, .PSECT $DATA, SIZE = 000160 (   56. WORDS)

| NAME | TYPE | OFFSET | NAME | TYPE | OFFSET | NAME | TYPE | OFFSET |
|------|------|--------|------|------|--------|------|------|--------|
| I | I*2 | 000156 | IDS | I*2 | 000154 | IORVB | I*2 | 000144 |
| IOWVB | I*2 | 000142 | IUNIT | I*2 | 000152 | IVFC | I*2 | 000146 |
| NUM | I*2 | 000004 Eqv | | | | | | |

    LOCAL AND COMMON ARRAYS:

| NAME | TYPE | SECTION | OFFSET | ------SIZE----- | | DIMENSIONS |
|------|------|---------|--------|------|------|------------|
| IBUF | L*1 | $DATA | 000006 | 000120 ( | 40.) | (80) |
| IOSB | L*1 | $DATA | 000002 | 000004 ( | 2.) | (4) |
| IPAR | I*2 | $DATA | 000126 | 000014 ( | 6.) | (6) |

    SUBROUTINES, FUNCTIONS, STATEMENT AND PROCESSOR-DEFINED FUNCTIONS:

| NAME | TYPE | NAME | TYPE | NAME | TYPE | NAME | TYPE | NAME | TYPE |
|------|------|------|------|------|------|------|------|------|------|
| EXIT | R*4 | GETADR | R*4 | WTQIO | R*4 | | | | |

5.

```
>
>LINK/MAP/CODE:FPP/EXECUTABLE:STUDENT PROGRAM,[1,1]FOROTS/LIB
>DIR STUDENT.*


Directory DR0:[305,303]
10-JAN-82 12:10

STUDENT.TSK;2        35.     C  10-JAN-82 12:10

Total of 35./35. blocks in 1. file
```

# Program Development

# SOLUTIONS

6.    STUDENT.TSK   Memory allocation map  TKB M40.02      Page 1
                         10-JAN-82   12:10

b.  Partition name : GEN
    Identification : FORV02
c.  Task  UIC     : [305,303]
    Stack    limits: 000260 001257 001000 00512.
    PRG xfr address: 020376
    Task attributes: FP
    Total address windows: 1.
d.  Task image size : 8416. words
    Task address limits: 000000 040623
    R-W disk blk limits: 000002 000042 000041 00033.

a.  *** Root segment: PROGRA

    R/W mem  limits: 000000 040623 040624 16788.
    Disk blk limits: 000002 000042 000041 00033.

    Memory allocation synopsis:

| Section | | | | Title | Ident | File |
|---------|--|--|--|-------|-------|------|
| . BLK.:(RW,I,LCL,REL,CON) | 001260 000442 00290. | | | | | |
| CVSSF :(RW,I,GBL,REL,OVR) | 001722 001756 01006. | | | | | |
| | 001722 001756 01006. | | | SCONVF | F40004 | FOROTS.OLB;2 |
| CVSSI :(RW,I,GBL,REL,OVR) | 003700 000406 00262. | | | | | |
| | 003700 000406 00262. | | | SCONVI | F40004 | FOROTS.OLB;2 |
| CVSSL :(RW,I,GBL,REL,OVR) | 004306 000106 00070. | | | | | |
| | 004306 000106 00070. | | | SCONVL | F40004 | FOROTS.OLB;2 |
| OTSSF :(RW,I,GBL,REL,CON) | 004414 001726 00982. | | | | | |
| | 004414 001726 00982. | | | SFIO | F40010 | FOROTS.OLB;2 |
| OTSSI :(RW,I,LCL,REL,CON) | 006342 011406 04870. | | | | | |
| | 006342 000044 00036. | | | SIADDS | F40002 | FOROTS.OLB;2 |
| | 006406 000054 00044. | | | SBRAS | F40002 | FOROTS.OLB;2 |
| | 006462 000036 00030. | | | SCALL | F40002 | FOROTS.OLB;2 |
| | 006520 000152 00106. | | | SCONV2 | F40002 | FOROTS.OLB;2 |
| | 006672 000024 00020. | | | SLPCMP | F40002 | FOROTS.OLB;2 |
| | 006716 000164 00116. | | | SEOL | F40005 | FOROTS.OLB;2 |
| | 007102 000004 00004. | | | SEXIT | F40001 | FOROTS.OLB;2 |
| | 007106 000034 00028. | | | SINCR | F40002 | FOROTS.OLB;2 |
| | 007142 000126 00086. | | | SIFW | F40002 | FOROTS.OLB;2 |
| | 007270 000014 00012. | | | SISNLS | F40001 | FOROTS.OLB;2 |
| | 007304 000102 00066. | | | SIMOVS | F40002 | FOROTS.OLB;2 |
| | 007406 000116 00078. | | | SLMOVS | F40002 | FOROTS.OLB;2 |
| | 007524 000050 00040. | | | SRETS | F40002 | FOROTS.OLB;2 |
| | 007574 000026 00022. | | | SLVEC | F40002 | FOROTS.OLB;2 |
| | 007622 000032 00026. | | | STESTS | F40002 | FOROTS.OLB;2 |
| | 007654 000024 00020. | | | SLTEST | F40002 | FOROTS.OLB;2 |
| | 007700 000172 00122. | | | SVTRAN | F40002 | FOROTS.OLB;2 |
| | 010072 000246 00166. | | | SOTI | F40003 | FOROTS.OLB;2 |
| | 010340 000000 00000. | | | SOTV | F40001 | FOROTS.OLB;2 |
| | 010340 000252 00170. | | | SSTPPA | F40001 | FOROTS.OLB;2 |
| | 010612 000160 00112. | | | SSAVRE | F40002 | FOROTS.OLB;2 |
| | 010772 000036 00030. | | | SFCHNL | F40001 | FOROTS.OLB;2 |
| | 011030 000550 00360. | | | SOPEN | F40004 | FOROTS.OLB;2 |

# Program Development

# SOLUTIONS

```
                                011600 000066 00054. SINITI F40002 FOROTS.OLB;2
                                011666 000032 00026. SPUTRE F40001 FOROTS.OLB;2
                                011720 001750 01000. SERRPT F40003 FOROTS.OLB;2
                                013670 000116 00078. SVINIT F40005 FOROTS.OLB;2
                                014006 000274 00188. SERQIO F40004 FOROTS.OLB;2
                                014302 000124 00084. SCLOSE F40001 FOROTS.OLB;2
                                014426 003206 01670. SERTXT F40004 FOROTS.OLB;2
                                017634 000114 00076. SR50   F40001 FOROTS.OLB;2
OTSSP  :(RW,D,GBL,REL,OVR)      017750 000036 00030.
                                017750 000036 00030. SCONVI F40004 FOROTS.OLB;2
                                017750 000036 00030. SFIO   F40010 FOROTS.OLB;2
                                017750 000036 00030. SCONVF F40004 FOROTS.OLB;2
                                017750 000036 00030. SCONVL F40004 FOROTS.OLB;2
PICSI  :(RO,I,LCL,REL,CON)      020006 000370 00248.
SCODE  :(RW,I,LCL,REL,CON)      020376 000606 00390.
                                020376 000606 00390. SYNCHQ FORV02 PROGRAM.OBJ;2
SDATA  :(RW,D,LCL,REL,CON)      021204 000160 00112.
                                021204 000160 00112. SYNCHQ FORV02 PROGRAM.OBJ;2
SDATAP:(RW,D,LCL,REL,CON)       021364 000234 00156.
                                021364 000234 00156. SYNCHQ FORV02 PROGRAM.OBJ;2
SSAOTS:(RW,D,LCL,REL,CON)       021620 000764 00500.
                                021620 000764 00500. SOTV   F40001 FOROTS.OLB;2
SSDEVT:(RW,D,LCL,REL,OVR)       022604 001210 00648.
                                022604 000000 00000. SOTV   F40001 FOROTS.OLB;2
SSFSR1:(RW,D,GBL,REL,OVR)       024014 004100 02112.
                                024014 000000 00000. SOTV   F40001 FOROTS.OLB;2
SSFSR2:(RW,D,GBL,REL,CON)       030114 000114 00076.
SSFSR3:(RW,D,GBL,REL,OVR)       030230 000000 00000.
SSFSR4:(RW,D,GBL,REL,OVR)       030230 000000 00000.
SSIOB1:(RW,D,LCL,REL,OVR)       030230 000204 00132.
                                030230 000204 00132. SOTV   F40001 FOROTS.OLB;2
SSIOB2:(RW,D,LCL,REL,OVR)       030434 000000 00000.
                                030434 000000 00000. SOTV   F40001 FOROTS.OLB;2
SSOBF1:(RW,D,LCL,REL,CON)       030434 000110 00072.
                                030434 000110 00072. SOTV   F40001 FOROTS.OLB;2
SSOBF2:(RW,I,LCL,REL,CON)       030544 000000 00000.
                                030544 000000 00000. SOTV   F40001 FOROTS.OLB;2
SSRESL:(RO,I,LCL,REL,CON)       030544 010044 04132.
SSTSKP:(RW,I,LCL,REL,CON)       040610 000012 00010.
                                040610 000000 00000. SOTV   F40001 FOROTS.OLB;2
```

Global symbols:

```
ADISIA 006362-R   BEQS   006432-R   CIDS   006546-R   DCISA  007136-R
ADISIM 006366-R   BGES   006442-R   CIFS   006556-R   DCISM  007130-R
ADISIS 006356-R   BGTS   006440-R   CILS   006664-R   DCISP  007134-R
ADISMA 006376-R   BLES   006430-R   CLCS   006546-R   DCISS  007124-R
ADISMM 006402-R   BLTS   006452-R   CLDS   006546-R   DCOS   002756-R
ADISMS 006372-R   BNES   006450-R   CLFS   006556-R   ECOS   002750-R
ADISSA 006346-R   BRAS   006444-R   CLIS   006670-R   EOLS   006774-R
ADISSM 006352-R   CAIS   006462-R   CMLSPI 006700-R   EXIT   007102-R
ADISSS 006342-R   CALS   006470-R   CMLSPM 006710-R   EXITS  010524-R
BAHS   010442-R   CICS   006546-R   CMLSPS 006672-R   FCOS   002744-R


FOOS   010514-R   IFWS   007142-R   MOISMM 007340-R
F.BFHD 000020     IFWSS  007206-R   MOISMS 007334-R
F.FDB  000154     ISNS   007270-R   MOISSA 007314-R
GCOS   002736-R   LCIS   004306-R   MOISSM 007310-R
ICIS   003706-R   LCOS   004354-R   MOISSS 007304-R
ICISA  007120-R   LSNS   007276-R   MOISOA 007360-R
ICISM  007112-R   MOISIA 007330-R   MOISOM 007354-R
ICISP  007116-R   MOISIM 007324-R   MOISOS 007350-R
ICISS  007106-R   MOISIS 007320-R   MOIS1A 007400-R
ICOS   004110-R   MOISMA 007344-R   MOIS1M 007372-R
```

# Program Development

# SOLUTIONS

```
MOIS1S 007364-R   OCIS   003700-R   TSDSM 007636-R   TVPS  007730-R
MOLS1A 007506-R   OCOS   004102-R   TSDSP 007646-R   TVQS  007722-R
MOLSIM 007500-R   PSES   010406-R   TSDSS 007626-R   V007A 000000
MOLSIP 007514-R   RCIS   001722-R   TSFSI 007642-R   SATT  014006-R
MOLSIS 007320-R   RELS   007320-R   TSFSM 007636-R   SBINAS 013160-R
MOLSMA 007432-R   RETS   007540-R   TSFSP 007646-R   SCLOSE 014302-R
MOLSMM 007426-R   RETSF  007530-R   TSFSS 007632-R   SCVTCB 006534-R
MOLSMP 007450-R   RETSI  007536-R   TSISI 007642-R   SCVTCI 006534-R
MOLSMS 007416-R   RETSL  007524-R   TSISM 007636-R   SCVTDB 006534-R
MOLSPA 007472-R   SALSIM 007574-R   TSISP 007646-R   SCVTDI 006534-R
MOLSPM 007460-R   SALSMM 007612-R   TSISS 007622-R   SCVTFB 006520-R
MOLSPP 007444-R   SALSSM 007576-R   TSLSI 007664-R   SCVTFI 006520-R
MOLSPS 007466-R   SAVRGS 010612-R   TSLSM 007660-R   SDET  014112-R
MOLSSA 007412-R   STPS   010510-R   TSLSP 007672-R   SDI   006546-R
MOLSSM 007406-R   SVLSIM 007602-R   TSLSS 007654-R   SECI  003722-R
MOLSSP 007436-R   SVLSMM 007616-R   TVDS  007714-R   SEOL  006772-R
MOLSSS 007304-R   SVLSSM 007604-R   TVFS  007706-R   SERRAA 012124-R
NMIS1I 006420-R   THRDS  010770-R   TVIS  007736-R   SERRTB 013500-R
NMIS1M 006406-R   TSDSI  007642-R   TVLS  007700-R   SERRTE 013670-R
```

```
SERRWT 014176-R   SPUTRE 011666-R   STVI  007736-R
SERRZA 013116-R   SRI    006556-R   STVL  007700-R
SERTXT 014426-R   SR50   017634-R   STVP  007730-R
SERXIT 012372-R   SSEQC  021670-R   STVQ  007722-R
SEXIT  010524-R   SSST   022564-R   SVIRIN 013670-R
SEXITS 012024-R   SSST0  011720-R   SVRINT 013670-R
SFCHNL 010772-R   SSST1  011726-R   SSFIO 005174-R
SFILL  013234-R   SSST2  011740-R   SSIFW 007152-R
SFIO   005170-R   SSST3  011746-R   SSOTI 010074-R
SFLDEF 011506-R   SSST4  011754-R   SSVIR 013670-R
SIFW   007146-R   SSST5  011762-R   .MOLUN 021676-R
SINITI 011600-R   SSST6  012064-R   .NLUNS 021674-R
SIOEXI 006716-R   SSST7  011772-R   .VIR  000002
SNAMC  021672-R   SSTP   010510-R
SOPEN  011030-R   SSTPS  010502-R
SOTI   010072-R   SSVTKS 013472-R
SOTSVA 021670-R   STSKP  040610-R
SPSE   010412-R   STVD   007714-R
SPSES  010452-R   STVF   007706-R
```

```
*** Task builder statistics:

     Total work file references: 35119.
     Work file reads: 0.
     Work file writes: 0.
     Size of core pool: 8220. words (32. pages)
     Size of work file: 4608. words (18. pages)

     Elapsed time:00:00:22
```

7.
```
>RUN STUDENT
ANOTHER UNQUALIFIED SUCCESS
another unqualified success
>
>
>
>
```

## Using the Editor Effectively

## LABORATORY EXERCISES

Perform the following exercises using command mode of EDT.

1.  Edit the file PROSE.TXT located in your UFD.


2.

    a.  Create a buffer called MORE.

    b.  Using the include command, read in the file MORE.TXT
        (located in your UFD) into the buffer.

    c.  Display the contents of the buffer.


3.  Create another buffer called SOME and move lines 1 through  13
    of MORE (buffer) into it.


4.  Save the contents of SOME in a file called SOME.TXT.

5.

    a.  Perform a character search for all occurrences of the
        character string "anything" in the main buffer.

    b.  Change every occurrence of the string "anything" to
        "everything".


6.  Define a macro called EDIT that:

    a.  Searches for the next occurrence of a ';'

    b.  Inserts a line of asterisks at that point

    c.  Substitutes a ';' for nothing

## LABORATORY EXERCISES

7.  Invoke that macro on the main buffer two or three times.

8.

   a.  Use the set command to set EDT terminal parameter to a VT1ØØ. (If your terminal is a VT1ØØ, set it to a VT52.)

   b.  Verify that it happened with the SHOW command. Also display your terminal screen width.

9.

   a.  Locate the first occurrence of the string "everything".

   b.  Substitute the word "anything" for the word "everything".

   c.  Use the SUBSTITUTE NEXT command to change all other occurrences of that string.

1Ø. Create a startup command file with the name EDTINI.EDT to:

   a.  Automatically put you in character mode upon entering EDT.

   b.  Automatically create a buffer and include the file MORE.TXT in that buffer.

   c.  Set EDT to search for exact matches when string searches.

   Exit the editor and reenter using PROSE.TXT as your input file.

## Using the Editor Effectively

## LABORATORY EXERCISES

Perform the following exercises using character mode of EDT.

11.

    a.  Move the cursor to the next section using the SECTION key.

    b.  Move the cursor to the top of the buffer using the TOP key.

    c.  Move the cursor to the bottom of the buffer using the BOTTOM key.

    d.  Backup two sections and insert a form feed. Backup two more sections and insert another form feed.

    e.  Move the cursor to the top of the buffer and using the PAGE key move through the main buffer to the end.

12. Using the CHNGCASE and SELECT keys change all occurrences of the word "LAW" to lowercase.

13.

    a.  Ensure that the caps lock key is up.

    b.  Beginning at the top of the file, search for all occurrences of the word LAW.

    c.  Press the caps lock key and repeat the exercise. What was the difference?

# Using the Editor Effectively

## LABORATORY EXERCISES

14. Move the lines beginning with the string "Boling's Postulate" and ending with "you'll get over it" to the end of the file using the CUT and PASTE keys.

15. Position the cursor at the beginning of a line and delete the first five words in that line in one command.

16. Using the COMMAND key, enter the line mode command to exit the editor.

# Using the Editor Effectively

# SOLUTIONS

```
1.    >EDIT/EDT PROSE.TXT
          1                 ;LAWS OF COMPUTER PROGRAMMING:
      *=MORE
2.a.  [EOB]
  b.  *INCLUDE MORE.TXT =MORE
      *=MORE
  c.      1            RULE OF THE GREAT:
          2
          3                   When somebody you greatly admire and respect
          4                   appears to be thinking deep thoughts, they
          5                   probably are thinking about lunch.
          6
          7            CLARKE'S LAW OF REVOLUTIONARY IDEAS;
          8
          9                   Every revolutionary idea - in Science, Politics,
         10                   Art or Whatever - evokes three stages of reaction.
         11                   They may be summed up by the three phrases:
         12
         13                   1.  "It is impossible - don't waste my time."
         14                   2.  "It is possible, but it is not worth doing."
         15                   3.  "I said it was a good idea all along."
         16
         17            NEWTON'S LITTLE-KNOWN SEVENTH LAW:
         18
         19                   A bird in the hand is safer than one overhead.
         20
         21            WHITE'S CHAPPAGQUIDICK THEOREM:
         22
         23                   The sooner and in more detail you announce the
         24                   bad news, the better.
      [EOB]


3.    *=SOME
      [EOB]
      *M =MORE 1:13 TO =SOME
      13 lines moved
      *=SOME
          1            RULE OF THE GREAT:
          2
          3                   When somebody you greatly admire and respect
          4                   appears to be thinking deep thoughts, they
          5                   probably are thinking about lunch.
          6
          7            CLARKE'S LAW OF REVOLUTIONARY IDEAS;
          8
          9                   Every revolutionary idea - in Science, Politics,
         10                   Art or Whatever - evokes three stages of reaction.
         11                   They may be summed up by the three phrases:
         12
         13                   1.  "It is impossible - don't waste my time."
      [EOB]
```

# SOLUTIONS

```
4.      *WRITE SOME.TXT =SOME
        DR0:[305,303]SOME.TXT;2 13 lines
        *=MAIN 1
            1               ;LAWS OF COMPUTER PROGRAMMING:
5.a.    *%ALL 'ANYTHING'
           16                       If anything can go wrong, it will.
           21                       2.  ANYTHING takes longer than you think.
           33                       8.  It is impossible to make anything foolproof
           63                       Once a job is fouled up, anything done to improve
           68                       90% of ANYTHING is crud.
           85                       Anything can be made to work if you fiddle with it
  b.    *S/ANYTHING/EVERYTHING/WH
           16                       If EVERYTHING can go wrong, it will.
           21                       2.  EVERYTHING takes longer than you think.
           33                       8.  It is impossible to make EVERYTHING foolproof
           63                       Once a job is fouled up, EVERYTHING done to improve
           68                       90% of EVERYTHING is crud.
           85                       EVERYTHING can be made to work if you fiddle with it
        6 substitutions
6.      *DEFINE MACRO EDIT
        *=EDIT
        [EOB]
        *I
  a.                      ';'
  b.                      I;**********
  c.                      S/;//
                         ^Z
        [EOB]
        *%WH
            1             ';'
            2             I;**********
            3             S/;//
        [EOB]
        *=MAIN 1
            1             ;LAWS OF COMPUTER PROGRAMMING:
7.      *EDIT
            1             ;LAWS OF COMPUTER PROGRAMMING:
            1             LAWS OF COMPUTER PROGRAMMING:
        *EDIT
           14             ;MURPHY'S LAW
           14             MURPHY'S LAW
        *EDIT
           38             ;BOLING'S POSTULATE:
           38             BOLING'S POSTULATE:
```

# Using the Editor Effectively

# SOLUTIONS

```
*0:39
    0.1         **********
    1           LAWS OF COMPUTER PROGRAMMING:
    2
    3                   1,  Any given program, when running is obsolete.
    4                   2.  Any given program costs more and takes longer.
    5                   3.  If a program is useful, it will have to be changed.
    6                   4.  If a program is useless, it will have to be document
ed.
    7                   5.  Any given program will expand to fill all available
    8                       memory.
    9                   6.  The value of a program is proporitional to the
   10                       weight of its output.
   11                   7.  Program complexity grows until it exceeds the
   12                       capability of the programmer who must maintain it.
   13
   13.1        **********
   14          MURPHY'S LAW
   15
   16                   If EVERYTHING can go wrong, it will.
   17
   18               Corollaires:
   19
   20                   1.  Nothing is as easy as it looks.
   21                   2.  EVERYTHING takes longer than you think.
   22                   3.  If there is a possibility of several things going
   23                       wrong, the one that will cause the most damage will
   24                       be the one to go wrong.
   25                   4.  If you perceive that there are four possible ways
   26                       in which a procedure can go wrong, and circumvent
   27                       these, then a fifth way will promptly develop.
   28                   5.  Left to themselves, things tend to go from bad
   29                       to worse.
   30                   6.  Whenever you set out to do something, something
   31                       else must be done first.
   32                   7.  Every solution breeds new problems.
   33                   8.  It is impossible to make EVERYTHING foolproof
   34                       because fools are so ingenious.
   35                   9.  Nature always sides with the hidden flaw.
   36                   10.  Mother nature is a bitch.
   37
   37.1        **********
   38          BOLING'S POSTULATE:
   39
8.a.  *SET TERMINAL VT52
      *SHOW TERMINAL
      VT52
  b.  *SET TERMINAL HCPY
      *SHOW TERMINAL
      Hardcopy
```

69

# Using the Editor Effectively

# SOLUTIONS

```
*SET SCREEN 40
*1;20
     1              LAWS OF COMPUTER PROGRAM
MING:
     2
     3                      1,  Any siven pr
osram, when runnins is obsolete.
     4                      2.  Any siven pr
osram costs more and takes lonser.
     5                      3.  If a prosram
 is useful, it will have to be chansed.
     6                      4.  If a prosram
 is useless, it will have to be document
ed.
     7                      5.  Any siven pr
osram will expand to fill all available
     8                          memory.
     9                      6.  The value of
 a prosram is proporitional to the
    10                          weisht of it
s output.
    11                      7.  Prosram comp
lexity srows until it exceeds the
    12                          capability o
f the prosrammer who must maintain it.
    13
    13.1          **********
    14             MURPHY'S LAW
    15
    16                      If EVERYTHING ca
n so wrons, it will.
    17
    18             Corollaires:
    19
    20                      1,  Nothins is a
 seasy as it looks.
*SHOW SCREEN
40
*SET SCREEN 80
    16              If EVERYTHING can so wrons, it will.
```

9. a. 
```
*S/EVERYTHING/ANYTHING/
    16              If ANYTHING can so wrons, it will.
```

b. 
```
1 substitution
*SN
     ^
Unrecosnized command
```

c. 
```
*S N
    21              2.  ANYTHING takes lonser than you think.
*SN\N\ N
    33              8.  1t is impossible to make ANYTHING foolproof
*S N
    63              Once a Job is fouled up, ANYTHING done to improve
*S N
    68              90% of ANYTHING is crud.
*S N
    85              ANYTHING can be made to work if you fiddle with it
*S N
*EDIT/EDT \ TDE/TIDE\EXIT
DR0:[305,303]PROSE.TXT;3 138 lines
```

70

## SOLUTIONS

```
        >
        >
        >
10.     >EDIT/EDT EDTINI.EDT
        Input file does not exist
        [EOB]
        *I
                a. INCLUDE MORE.TXT =MORE
                b. SET SEARCH EXACT
                c. SET MODE C
                   ^Z
        [EOB]
        *%WH
            1               INCLUDE MORE.TXT =MORE
            2               SET SEARCH EXACT
            3               SET MODE C
        [EOB]
        *EX
        DR0:[305,303]EDTINI.EDT;1 3 lines
        >EDIT/EDT PROSE.TXT
```

11-16.  No Solution Provided

# Using Indirect Command Files

# LABORATORY EXERCISES

1. Write an indirect command file that prompts for a file type and then displays on your terminal all files under your UIC that are of that particular file type.

2. Write an indirect command file that will set your terminal to the following characteristics:

   VT52 or VT100
   HOLD SCREEN
   DCL

   Give the file the name LOGIN.CMD, then logout and log back in.

3. To your LOGIN.CMD file created in exercise 2 above, add the following features:

   a. Ask the user if he wants to read a message file (MESSAGE.TXT)

   b. If the user answers yes, display the file contents on the terminal. If the user answers no, exit the indirect command file.

   Create a file MESSAGE.TXT with a note in it. Logoff and log back in to test the indirect command file.

# Using Indirect Command Files

## LABORATORY EXERCISES

4. Write an indirect command file to get the following system information and put it into a file:

   a. the time and date

   b. the user's UIC

   c. the command language interpreter the user is currently using

   d. what physical device the user is on

   e. whether the system is mapped or not

   f. which operating system is running


5. Write a LOGIN.CMD file to ask the user for another password. If it is not the correct password, ask the question again twice. If the answer is still not correct, automatically log the user off. If the password is correct, ask the user for his name and group and save it in a file along with the time and date.


6. Write an indirect command file that asks the user if he wants to assemble/compile or task-build a task.

   If he wants to assemble, invoke another command file that asks the file name to be assembled.

7. Write an indirect command file that will purge your directory of every version but the latest two.

   Modify this to ask for a particular file type to be purged.

   Modify it to ask for the number of latest versions to be saved.

## Using Indirect Command Files

## LABORATORY EXERCISES

8.  Obtain a listing of  LB:[1,2]STARTUP.CMD  and  SHUTUP.CMD  and
    read for understanding.

    What function do these two indirect command files perform?


9.  Write an indirect command file to broadcast  a  message  to  a
    particular  terminal  every X seconds.  Ask the user to supply
    the terminal number, number of seconds, and the message.  Also
    make it flexible enough to process messages from a file.


10. Create an indirect  command  file  to  perform  the  following
    actions:

    a.  Prompt for a file name

    b.  Prompt for a UFD

    c.  Copy the file with the indicated  name  belonging  to  the
        specified UFD, to your UFD

    d.  Ask if there are any more files to  be  copied.   If  yes,
        repeat the above steps;  if no, exit.

# Using Indirect Command Files

# SOLUTIONS

```
SOLUTION - EXERCISE #1

.ENABLE SUBSTITUTION
.ASKS[1:3] TYPE FILE TYPE FOR DIRECTORY
DIR *.'TYPE'



SOLUTION - EXERCISE #2



.IF <CLI> EQ "MCR" SET /DCL=TI:
SET TERMINAL/VT52
SET TERMINAL/HOLD



SOLUTION - EXERCISE #3



.IF <CLI> EQ "MCR" SET /DCL=TI:
SET TERMINAL/VT52
SET TERMINAL/HOLD
.ASK MESS DO YOU WANT TO READ YOUR MESSAGE FILE
.ENABLE QUIET
.IFT MESS TYPE MESSAGE.TXT



SOLUTION - EXERCISE #4



.OPEN INFO
.ENABLE SUBSTITUTION
.DATA '<DATE>'
.DATA '<TIME>'
.DATA '<SYDISK>' '<SYUNIT>'
.DATA '<CLI>'
.IFT <MAPPED> .DATA MAPPED SYSTEM
.IFF <MAPPED> .DATA UNMAPPED SYSTEM
.IF <SYSTEM> EQ 6 .DATA RSX-11M-PLUS SYSTEM
.IF <SYSTEM> EQ 1 .DATA RSX-11M SYSTEM
.CLOSE INFO
```

# Using Indirect Command Files

# SOLUTIONS

```
;
;          SOLUTION - EXERCISE #5
;
;
;
;
;
;
           .SETN COUNT 1
.LOOP:
           .ASKS PSSWRD ENTER PASSWORD PLEASE
           .IF PSSWRD EQ "FUN" .GOTO 10
           .IF COUNT EQ 3 .GOTO 20
           .INC COUNT
           .GOTO LOOP
.10:
           .ASKS GROUP ENTER YOUR GROUP NUMBER PLEASE
           .ASKS MEMBER ENTER YOUR MEMBER NUMBER PLEASE
           .ENABLE SUBSTITUTION
           .OPEN USERS
           .DATA 'GROUP' 'MEMBER' '<TIME>' '<DATE>'
           .CLOSE USERS
           .EXIT
.20:       .IF <CLI> EQ "DCL" LO
           .IF <CLI> EQ "MCR" BYE


;
;
;          SOLUTION - EXERCISE #6
;
;
;
           .ASKS OPER DO YOU WANT TO COMPILE,ASSEMBLE OR TASKBUILD(C,A,T)
           .IF OPER EQ "C" @COMPILE.CMD
           .IF OPER EQ "A" @ASSEMBLE.CMD
           .ASK CONT DO YOU WISH TO TASKBUILD NOW
           .IFT CONT .SETS OPER "T"
           .IF OPER EQ "T" @TASKBUILD.CMD


;
;          COMPILE.CMD
;
;
.ASKS FILE WHAT FILE DO YOU WANT TO COMPILE
.ASKS LANG WHAT LANGUAGE DO YOU WANT TO USE (FOR, F4P,F77)
.ENABLE SUBSTITUTION
.IF LANG EQ "FOR" FORTRAN/FOR/LIST 'FILE'
.IF LANG EQ "F4P" FORTRAN/F4P/LIST 'FILE'
.IF LANG EQ "F77" FORTRAN/F77/LIST 'FILE'
.EXIT
;
;
;          ASSEMBLE.CMD
;
;
           .ASKS FILE WHAT FILE TO ASSEMBLE
           .ENABLE SUBSTITUTION
           MACRO/LIST 'FILE'
           .EXIT


;
;          TASKBUILD.CMD
;
           .ASKS FILE WHAT FILE TO TASKBUILD
           .ENABLE SUBSTITUTION
           LINK/MAP 'FILE'
```

77

# Using Indirect Command Files

# SOLUTIONS

```
;
;        SOLUTION - EXERCISE #7
;
;
;

        .ASKS TYPE WHAT FILE TYPE TO BE PURGED
        .ASKS NUM HOW MANY VERSIONS TO KEEP
        .ENABLE SUBSTITUTION
        PURGE *.'TYPE'/KEEP:'NUM'/LOG


;
;
;        SOLUTION - EXERCISE #8
;
;        No Solution Provided
;
;
;
;        SOLUTION - EXERCISE #9
;                                                    ;
;                                                    ;
;                                                    ;

        .ASKS TTY TO WHAT TERMINAL
        .ASKS TIM HOW OFTEN (SECONDS)
        .ASKS MES WHAT MESSAGE
        .SETN COUNT 1                                ;
.LOOP:                                               ;
        .ENABLE SUBSTITUTION                         ;
        BRO TT'TTY': 'MES'                           ;
        .INC COUNT                                   ;
        .IF COUNT EQ 6 .EXIT                         ;
        .DELAY 'TIM'S                                ;
        .GOTO LOOP                                   ;
;
;
;                                                    ;
;                                                    ;
;        SOLUTION - EXERCISE #10                     ;
;                                                    ;
;                                                    ;
;                                                    ;
.LOOP:                                               ;
        .ASKS FILE FILE NAME TO BE COPIED
        .ASKS UFD FROM WHAT UFD
        .ENABLE SUBSTITUTION
        COPY 'UFD' 'FILE' *.*
        .ASK RPT DO YOU WANT TO REPEAT THIS OPERATI
        .IFT RPT .GOTO LOOP
;                                                    ;
;                                                    ;
;                                                    ;
                                                     ;
                                                     ;

                                                     ;
                                                     ;
                                                     ;
```

# Controlling Task Execution

## WRITTEN EXERCISES

1. Using Figure 8-1, complete the following chart. (Fill in as many of the states as apply to each of the tasks.)

```
TASK     ____ _____
TASKA    _____ _____
TASKB    _____ _____ _____
TASKC    _____ _____
TASKD    _____ _____ _____
TASKE    _____ _____ _____ _____ _____
TASKF    _____ _____
TASKG    _____ _____ _____
TASKH    _____ _____
TASKI    _____ _____
TASKJ    _____ _____ _____
TASKK    _____
```

2. List at least four conditions that are significant events.

   a.

   b.

   c.

   d.

   e.

   f.

   g.

3. Using Figure 8-1, suppose TASKE completes and TASKI becomes unblocked. What task would become the active task?

   _____

# Controlling Task Execution

## WRITTEN EXERCISES

4. A real-time task will be built with a priority in the range of
_____.

5. Most programs in use on the system run at a priority in the range of 1 - ___, and in the partition called ___.

6. A real-time task generally will/will not (circle one) be built to run in the GEN partition.

7. Building a task to be checkpointable allows the operating system to _____ the task's memory image out to a system checkpoint file.

8. The DCL command LINK/CHECKPOINT:TASK will create checkpoint space for the task in the ____ _____ _____.

9. You achieve _____ _____ when you delay assigning a logical unit to a physical device until just before running a program.

1Ø. A logical unit represents an ___ _____.

11. The priority of an installed task may be changed by a privileged user by issuing the MCR ___ command.

12. The task name of an installed task can/cannot (circle one) be changed at run time.

13. List the Task Builder defaults for the following task attributes:

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| Priority | _____ | | | | | |
| Partition | _____ | | | | | |
| Checkpointability | ___ _____ | | | | | |
| Task Name | _____ ___ _____ ___ | | | | | |
| | ____ _____ _____ | | | | | |
| Task Units | __ __ __ __ ___ __ _____ | | | | | |

# Controlling Task Execution

## WRITTEN EXERCISES

14. Only a _____ user may permanently install a task in the STD.

15. When a task is installed in the STD, its task state changes from _____ to _____ ____ _____.

16. The privileged command _____ is used to remove a task from the STD.

17. Uninstalled tasks are invoked by issuing the ___ command.

18. Installed tasks which are not MCR spawnable are invoked by issuing the ___ _____.

19. An MCR spawnable installed task is invoked by issuing the last _____ _____ __ ___ ____ ____.

20. The ____ is used with the RUN command to indicate that the task image file is located in the System UFD (System and Library UFD for M-PLUS systems).

21. To RUN a task at a specific time, the _____ qualifier of the RUN command is used.

22. To RUN a task at repeated intervals, the _____ qualifier of the RUN command is used.

# Controlling Task Execution

## SOLUTIONS

1. Using Figure 8-1, complete the following chart. (Fill in as many of the states as apply to each of the tasks.)

| TASK | TASK STATE(S) | | |
|------|------|------|------|
| TASKA | KNOWN, | DORMANT | |
| TASKB | KNOWN, | ACTIVE, | BLOCKED |
| TASKC | KNOWN, | DORMANT | |
| TASKD | KNOWN, | ACTIVE, | BLOCKED |
| TASKE | KNOWN, | ACTIVE, | READY-TO-RUN, CURRENT |
| TASKF | KNOWN, | DORMANT | |
| TASKG | KNOWN, | ACTIVE, | BLOCKED |
| TASKH | KNOWN, | DORMANT | |
| TASKI | KNOWN, | ACTIVE, | BLOCKED |
| TASKJ | KNOWN, | ACTIVE, | READY-TO-RUN |
| TASKK | UNKNOWN | | |

2. List at least four conditions that are significant events.

    a. TASK COMPLETION
    b. I/O COMPLETION
    c. RUN COMMAND IS ISSUED
    d. A TASK DECLARES A SIGNIFCANT EVENT
    e. TASK ENTERS WAIT STATE
    f. MARK TIME EXPIRATION
    g. PROGRAM ISSUES A SEND DATA DIRECTIVE

3. Using Figure 8-1, suppose TASKE completes and TASKI becomes unblocked. What task would become the active task?

    TASKI

# Controlling Task Execution

## SOLUTIONS

4. A real-time task will be built with a priority in the range of <u>150-250</u>.

5. Most programs in use on the system run at a priority in the range of <u>1</u> - <u>150</u> and in the partition called <u>GEN</u>.

6. A real-time task generally will/<u>will</u> <u>not</u> (circle one) be built to run in the GEN partition.

7. Building a task to be checkpointable allows the operating system to <u>SWAP</u> the task's memory image out to a system checkpoint <u>file</u>.

8. The DCL command LINK/CHECKPOINT:TASK will create checkpoint space for the task in the <u>task image file</u>.

9. You achieve <u>DEVICE INDEPENDENCE</u> when you delay assigning a logical unit to a <u>physical device</u> until just before running a program.

10. A logical unit represents an <u>I/O device</u>.

11. The priority of an installed task may be changed by a privileged user by issuing the MCR <u>ACT</u> command.

12. The task name of an installed task can/<u>cannot</u> (circle one) be changed at run time.

13. List the Task Builder defaults for the following Task attributes:

| | |
|---|---|
| Priority | <u>50</u> |
| Partition | <u>GEN</u> |
| Checkpointability | <u>Not checkpointable</u> |
| Task Name | <u>First six characters of the first input file</u> |
| Task Units | <u>6 (7 in the case of FORTRAN)</u> |

# SOLUTIONS

14. Only a <u>privileged</u> user may permanently install a task in the STD.

15. When a task is installed in the STD, its task state changes from <u>UNKNOWN</u> to <u>KNOWN AND DORMANT</u>.

16. The privileged command <u>REMOVE</u> is used to remove a task from the STD.

17. Uninstalled tasks are invoked by issuing the <u>RUN</u> command.

18. Installed tasks which are not MCR spawnable are invoked by issuing the <u>RUN Command</u>.

19. An MCR spawnable installed task is invoked by issuing the last <u>three characters of the task name</u>.

20. The <u>$</u> is used with the RUN command to indicate that the task image file is located in the System UFD (System and Library UFD for M-PLUS systems).

21. To RUN a task at a specific time, the <u>/SCHEDULE</u> qualifier of the RUN command is used.

22. To RUN a task at repeated intervals, the <u>/INTERVAL</u> qualifier of the RUN command is used.

# Controlling Task Execution

## LABORATORY EXERCISES

1. Using the object modules created in the Program Development module, build your task using the /OPTIONS switch of the LINK command with: a priority of 75, a task name of XXABC, and a partition of SYSPAR. Make this task checkpointable to a system checkpoint file. List a directory on the task image file and note the number of file blocks.

2. Rebuild your task making it checkpointable to the task image file. Do a directory on the task image file. Note how many blocks the file contains. Did the task image file decrease or increase? Why?

   To perform the following exercises, log into a privileged account.

3. Locate the LABONE.TSK in your UFD. Install this task into the system task directory using your name as the task name, at a priority of 45 (10). Using the SHOW command, display your task characteristics.

4. Determine the task logical unit assignments. To which device is logical unit 5 assigned?

5. Run the task. Display all active tasks initiated from your terminal. What name is your task running under? Abort your task.

6. Remove your task from the System Task Directory. Run your task using the install-run-remove option of the RUN command. Determine the task name by showing the active tasks at your terminal. Abort the task.

7. Install your task so that it is MCR spawnable. Run it under MCR without using the RUN command. What is its task name? Abort it.

8. Remove your task. Reinstall it specifying the partition in which it will run as XYZZY. After answering the following questions, remove the task.

   a. Was there an error message?
   b. Was a STD entry created?
   c. What is the partition name in the STD entry?

# Controlling Task Execution

# SOLUTIONS

1.  &gt;LINK/OPTIONS/CHECKPOINT:SYSTEM PROGRAM
    Option? TASK=XXXABC
    Option? PAR=DRVPAR
    Option? PRI=75
    Option?

    &gt;DIR PROGRAM.TSK


    Directory DR0:[305,303]
    22-JAN-82 14:25

    PROGRAM.TSK;1        4.      C  22-JAN-82 14:24

    Total of 4./4. blocks in 1. file

2.  &gt;LINK/OPTIONS/CHECKPOINT:TASK PROGRAM
    Option? TASK=XXXABC
    Option? PAR=DRVPAR
    Option? PRI=75
    Option?
    &gt;
    &gt;
    &gt;DIR PROGRAM.TSK;*


    Directory DR0:[305,303]
    22-JAN-82 14:26

    PROGRAM.TSK;1        4.      C  22-JAN-82 14:24
    PROGRAM.TSK;2        6.      C  22-JAN-82 14:26

    Total of 10./10. blocks in 2. files

    &gt;
    &gt;
3.  &gt;INSTALL/TASK_NAME:JIM/PRIORITY:75 LABONE.TSK
    &gt;SHOW TASKS:JIM/INSTALLED
    JIM     0356    GEN      75. 00053400 SY0:-01115075
    &gt;
4.  &gt;SHOW TASKS:JIM/LOGICAL_UNITS
    SY0:  1.
    SY0:  2.
    SY0:  3.
    SY0:  4.
    TI0:  5.
    CL0:  6.
    &gt;
5.  &gt;RUN JIM
    ENTER UPPER LIMIT
    &gt;
    DCL&gt;SHOW TASKS ACTIVE
    PRIME #1 IS 2
    MCR...
    SHOT54
    JIM
    PRIME #2 IS 3

# SOLUTIONS

```
DCL>ABO JIM

14:43:38  Task "JIM   " terminated
          Aborted via directive or CLI
          And with pending I/O requests


          >
6.        >REMOVE JIM
          >RUN LABONE.TSK
          ENTER UPPER LIMIT

          DCL>SHOW TASKS/ACTIVE
          PRIME #1 IS 2
          MCR...
          SHOT54
          TT54
          PRIME #2 IS 3


          >
          >RUN LABONE.TSK
          ENTER UPPER LIMIT
          DCL>ABO

14:44:53  Task "TT54   " terminated
          Aborted via directive or CLI
          And with pending I/O requests


7.        >INSTALL/TASK_NAME:...JIM LABONE.TSK
          >JIM
          ENTER UPPER LIMIT

          DCL>SHOW TASK ACTIVE
          PRIME #1 IS 2
          PRIME #2 IS 3
          SHOT54
          >
            ...JIM

          >SHOW TASKS/ACTIVE
          MCR...
          SHOT54
          >JIM
          ENTER UPPER LIMIT

          DCL>ABO JIM
          PRIME #1
14:51:59  Task "...JIM" terminated
          Aborted via directive or CLI
          And with pending I/O requests
          REM JIM
          >


          >
8.        >INSTALL/PARTITION:XYZZY LABONE.TSK
          INS -- Partition XYZZY  not in system, defaulting to GEN
          >SHOW TASKS:LABONE /INSTALLED
          LABONE 0356   GEN     50. 00053400 SY0:-01115075
          >REM LABONE
```

# Libraries

## WRITTEN EXERCISES

1. List the three types of libraries.

   _____

   _____

   _____

2. The _____ Table in a library is used to locate a global label in a library module.

3. The _____ Table in a library is used to locate a module contained within a library.

4. The DCL _____ command is used to invoke the task that maintains a library file.

5. Write the file specification of the default System Macro library.

   _____

6. Write the file specification of the default System Object library.

   _____

7. The modules READ, WRITE, SINE, and COSINE are contained in a file MYRTNS.OBJ. Write the DCL command to create a User Object library called MYOBJECT. Take the system defaults on the size of the file, the number of module entries and global entries.

   _____

8. Write the command to extract the WRITE module from the library created above. Give the output file the name WRITE.OBJ.

   _____

# Libraries

## WRITTEN EXERCISES

9.  Write the command to replace the SINE module in MYOBJECT.  The input file containing the module is ROUTINES.OBJ.

    _____

10. List the two system tasks that automatically search a library file.

    _____

    _____

11. Write the command to list the entry points in the default System Object library.

    _____

# Libraries

## SOLUTIONS

1.  List the three types of libraries.

    <u>Macro</u> <u>Source</u> <u>Libraries</u>

    <u>Object</u> <u>Libraries</u>

    <u>Universal</u> <u>Libraries</u>

2.  The <u>Entry Point</u> Table in a library is used to locate a global <u>label</u> in a library module.

3.  The <u>Module Name</u> Table in a library is used to locate a module contained within a library.

4.  The DCL <u>LIBRARY</u> command is used to invoke the task that maintains a library file.

5.  Write the file specification of the default System Macro library.

    <u>LB:[1,1]RSXMAC.SML</u>

6.  Write the file specification of the default System Object library.

    <u>LB:[1,1]SYSLIB.OLB</u>

7.  The modules READ, WRITE, SINE, and COSINE are contained in a file MYRTNS.OBJ. Write the DCL command to create a User Object library called MYOBJECT. Take the system defaults on the size of the file, the number of module entries and global entries.

    <u>LIBRARY/CREATE</u> <u>MYOBJECT</u> <u>MYRTNS</u>

8.  Write the command to extract the WRITE module from the library created above. Give the output file the name WRITE.OBJ.

    <u>LIBRARY/EXTRACT/OUTPUT:WRITE.OBJ</u> <u>MYOBJECT</u> <u>WRITE</u>

9.  Write the command to replace the SINE module in MYOBJECT. The input file containing the module is ROUTINES.OBJ.

    <u>LIBRARY/REPLACE</u> <u>MYOBJECT</u> <u>ROUTINES</u>

# Libraries

## SOLUTIONS

10. List the two system tasks that automatically search a library file.

    MACRO-11 Assembler

    Task Builder

11. Write the command to list the entry points in the default System Object library.

    LIBRARY/LIST/NAMES LB:[1,1]SYSLIB

# Libraries

## LABORATORY EXERCISES

1. Read the information provided by the HELP command on LIBRARY.

2. Using the DCL LIBRARY command, obtain a directory of the System Macro library on LB:[1,1] RSXMAC.SML.

3. Obtain a full directory including global symbols of the same library and put it in a file rather than having it come to your terminal. Print that file on the line printer. What additional information does this type of listing give you? When would this information be most beneficial?

4. Extract the following modules from the library putting each module in a separate file (in your UFD) on your disk area:

   |          |        |
   |----------|--------|
   | ALUN$S   | DIR$   |
   | CALL     | OFF$   |
   | EXIT$C   | CRAW$  |

   You may want to inspect the contents of a file to see how a macro source file looks.

5. Now, create a Macro library on your own UFD called MYMAC that contains the modules extracted in exercise 3. Do a brief directory on MYMAC to ensure that all modules are there. Module names in a Source library equate to the first six characters in the last .TITLE statement for a macro source file.

6. Obtain information on available file space and recoverable deleted space on the library. Then delete DIR$ and CRAW$ from the library and compare the new figures on available file space and recoverable deleted space with the old figures.

7. Replace ALUN$S, CALL, and EXIT$C with the same macro source files and again look at the new figures on available free space and recoverable deleted space.

93

# Libraries

## LABORATORY EXERCISES

8.  Compress the library to recover the free deleted space.

9.  You can repeat the exercises above for object modules using the library LB:[1,1]SYSLIB.OLB.

10. You can repeat the exercises above using MCR commands.

# Libraries

# SOLUTIONS

```
        >;
1.      >;      #1
        >;
        >HELP LIBRARY
        The LIBRARY command performs maintenance operations on a library
        file. A library file is a direct access file containing one or more
        modules.

        There are three types of libraries. Object libraries hold object
        modules and can be used as input to the Task Builder. Macro
        libraries hold source macros and can be called in by the macro
        assembler. Universal libraries can have any contents.  See the
        RSX-11M/M-PLUS Command Language Manual and the discussion of the LBR
        utility in the RSX11M/M-PLUS Utilities Manual for more information.

        The LIBRARY command performs the following eight operations:

        CREATE          DELETE          EXTRACT         INSERT          LIST
        REPLACE         REMOVE          COMPRESS

        For more information on a particular LIBRARY operation, see HELP
        LIBRARY operation.
        >;
2.      >;      #2
        >;
        >LIBRARY/LIST LB:[1,1]RSXMAC.SML

        Directory of file RSXMAC.SML;133
        Macro library created by:  LBR V06.00
        Last insert occurred 18-DEC-81 at 18:32:33
        MNT entries allocated: 512; Available: 85
        EPT entries allocated: 0; Available: 0
        File space available: 00938 words

        ABRT$
        ABRT$C
        ABRT$S
        AFF$
        ALTP$
        ALTP$C
        ALTP$S
        ALUN$
        ALUN$C
        ALUN$S
        ASTX$
        ASTX$C
        ASTX$S
        ATRG$
        ATRG$C
        ATRG$S
        BDOFF$
        CALL
        CALLR
        CBYTE$
        CCML$
        CGET$B
        CGET$W
        CINT$
        CINT$C
        CINT$S
        CLEF$
        CLEF$C
        CLEF$S
        CLOSE$
```

# Libraries

## SOLUTIONS

```
CMKT$
CMKT$C
CMKT$S
CMOV$B
CMOV$W
CMOV$2
CNCT$
ENCP$
ENCP$C
ENCP$S
ERR$
EXIF$
EXIF$C
EXIF$S
EXIT$
EXIT$C
EXIT$S
EXST$
EXST$C
EXST$S
EXTK$
EXTK$C
EXTK$S
FCSBT$
FCSMC$
FDAT$A
FDAT$R
FDBDF$
FDBF$A
FDBF$R
FDBK$A
FDBK$R
FDBSZ$
FDOFF$
FDOF$L
FDO
```

3.
```
>;
>;        #3
>;
>LIBRARY/LIST:[305,303]LIBDIR.LST/FULL LB:[1,1]RSXMAC.SML
>PRINT LIBDIR.LST
PRI -- Job 765, name 'LIBDIR   ', submitted to queue 'PRINT '
>;
```
4.
```
>;        #4
>;
>LIBRARY/EXTRACT/OUTPUT:[305,303]ALUNS.MAC LB:[1,1]RSXMAC.SML ALUN$S
>LIBRARY/EXTRACT/OUTPUT:[305,303]CALL.MAC LB:[1,1]RSXMAC.SML CALL
>LIBRARY/EXTRACT/OUTPUT:[305,303]EXITC.MAC LB:[1,1]RSXMAC.SML EXIT$C
>LIBRARY/EXTRACT/OUTPUT:[305,303]CRAW.MAC LB:[1,1]RSXMAC.SML CRAW$
>LIBRARY/EXTRACT/OUTPUT:[305,303]DIR.MAC LB:[1,1]RSXMAC.SML DIR$
>LIBRARY/EXTRACT/OUTPUT:[305,303]OFF.MAC LB:[1,1]RSXMAC.SML OFF$
```

# SOLUTIONS

```
        >;
        >;
  5     >;        #5
        >;
        >LIBRARY/CREATE/MACRO MYMAC ALUNS,CALL,EXITC,CRAW,DIR,OFF
        >LIBRARY/LIST MYMAC.MLB

        Directory of file MYMAC.MLB;1
        Macro library created by:  LBR V06.00
        Last insert occurred 30-DEC-81 at 09:39:16
        MNT entries allocated: 256; Available: 250
        EPT entries allocated: 0; Available: 0
        File space available: 23830 words

        ALUN$S
        CALL
        CRAW$
        DIR$
        EXIT$C
        OFF$

        >;
  6.    >;        #6
        >;
        >LIBRARY/LIST/FULL MYMAC.MLB

        Directory of file MYMAC.MLB;1
        Macro library created by:  LBR V06.00
        Last insert occurred 30-DEC-81 at 09:39:16
        MNT entries allocated: 256; Available: 250
        EPT entries allocated: 0; Available: 0
        File space available: 23830 words

        ALUN$S  Size:00079  Inserted:30-DEC-81

        CALL    Size:00032  Inserted:30-DEC-81

        CRAW$   Size:00085  Inserted:30-DEC-81

        DIR$    Size:00082  Inserted:30-DEC-81

        EXIT$C  Size:00092  Inserted:30-DEC-81

        OFF$    Size:00120  Inserted:30-DEC-81


        >LIBRARY/DELETE MYMAC.MLB DIR$,CRAW$
        Modules deleted:
        DIR$
        CRAW$

        >LIBRARY/LIST/FULL MYMAC.MLB

        Directory of file MYMAC.MLB;1
        Macro library created by:  LBR V06.00
        Last insert occurred 30-DEC-81 at 09:39:16
        MNT entries allocated: 256; Available: 252
        EPT entries allocated: 0; Available: 0
        File space available: 23830 words
        Recoverable deleted space: 00167 words

        ALUN$S  Size:00079  Inserted:30-DEC-81

        CALL    Size:00032  Inserted:30-DEC-81

        EXIT$C  Size:00092  Inserted:30-DEC-81

        OFF$    Size:00120  Inserted:30-DEC-81
```

# SOLUTIONS

```
      >;
      >;
7.    >;      #7
      >;
      >LIBRARY/REPLACE MYMAC.MLB ALUNS,CALL,EXITC
      Module 'ALUN$S' replaced

      Module 'CALL  ' replaced

      Module 'EXIT$C' replaced

      >LIBRARY/LIST/FULL MYMAC.MLB

      Directory of file MYMAC.MLB;1
      Macro library created by:  LBR V06.00
      Last insert occurred 30-DEC-81 at 09:39:27
      MNT entries allocated: 256; Available: 252
      EPT entries allocated: 0; Available: 0
      File space available: 23627 words
      Recoverable deleted space: 00370 words

      ALUN$S   Size:00079  Inserted:30-DEC-81

      CALL     Size:00032  Inserted:30-DEC-81

      EXIT$C   Size:00092  Inserted:30-DEC-81

      OFF$     Size:00120  Inserted:30-DEC-81



      >;
8.    >;      #8
      >;
      >LIBRARY/COMPRESS MYMAC.MLB NEWMYMAC
      >;
      >;
```

# Advanced Maintenance Operations

## WRITTEN EXERCISES

1. _____ devices process records one at a time.

2. Devices that collect records into a file and store them for future use are called _____ devices.

3. The _____ software system creates and maintains the file structure on disks and DECtape.

4. The FILES-11 file structure has a ___ level directory structure consisting of the _____ and the

   _____.

5. List the five system files that must be present on every FILES-11 volume.

   a. INDEXF.SYS
   b. BITMAP.SYS
   c. BADBLK.SYS
   d. 000000.DIR
   e. CORIMG.SYS

6. Every file has two parts, the _____ and a _____.

7. The header of a file is located in the _____ system file.

8. A _____ device is one that can be accessed by all users of the system.

9. A _____ device is one that has been allocated by a user.

10. You use the _____ command to logically inform the operating system about a volume.

11. You use the _____ command to make a second copy of a volume for safety precautions.

12. The _____ will transfer files from one volume to another, and in addition will convert the file to the output volume structure.

99

# Advanced Maintenance Operations

## WRITTEN EXERCISES

13. The _____ is software that manages output devices (like printers, plotters) for efficient use.

14. You use the _____ command to place a file into a queue for printing at the line printer, when the Queue Manager is installed on your system.

15. The _____ command is used to display the jobs in a queue.

16. You use the _____ to inspect the contents of a file when debugging a program.

17. You use the _____ command to compare the contents of two ASCII files.

# Advanced Maintenance Operations

## SOLUTIONS

1. <u>Record-oriented</u> devices process records one at a time.

2. Devices that collect records into a file and store them for future use are called <u>file-structured</u> devices.

3. The <u>FILES-11</u> software system creates and maintains the file structure on disks and DECtape.

4. The FILES-11 file structure has a <u>two</u> level directory structure consisting of the <u>Master File Directory</u> and the <u>User File Directories</u>.

5. List the five system files that must be present on every FILES-11 volume.

   a. INDEXF.SYS
   b. BITMAP.SYS
   c. BADBLK.SYS
   d. 000000.DIR
   e. CORIMG.SYS

6. Every file has two parts, the <u>body</u> and a <u>header</u>.

7. The header of a file is located in the <u>INDEX.SYS</u> system file.

8. A <u>public</u> device is one that can be accessed by all users of the system.

9. A <u>private</u> device is one that has been allocated by a user.

10. You use the <u>MOUNT</u> command to logically inform the operating system about a volume.

11. You use the <u>BACKUP</u> command to make a second copy of a volume for safety precautions.

12. The <u>File Transfer Utility</u> will transfer files from one volume to another, and in addition will convert the file to the output volume structure.

# Advanced Maintenance Operations

# SOLUTIONS

13. The Queue Manager is software that manages output devices (like printers, plotters) for efficient use.

14. You use the PRINT command to place a file into a queue for printing at the line printer, when the Queue Manager is installed on your system.

15. The SHOW QUEUE command is used to display the jobs in a queue.

16. You use the File Dump Utility to inspect the contents of a file when debugging a program.

17. You use the DIFFERENCES command to compare the contents of two ASCII files.

# Advanced Maintenance Operations

## LABORATORY EXERCISES

1.  Give the commands needed to perform the following actions:

    a.  Run FMT and verify that the operation was successful.

    b.  Run BAD and list the bad blocks at your terminal.

    c.  Initialize a RKØ5 disk pack to use FILES-11.

    d.  Create a UFD on a RKØ5 pack for [3Ø5,3ØØ]. Allow only yourself to have write, delete and extend privileges.

    e.  Create a backup copy of a RKØ5 pack on 8ØØ BPI tape.


2.  Write the command to copy a file from a magnetic tape in DOS-11 format to your UFD on a FILES-11 volume.

3.  Write the command to show the differences between two versions of an ASCII file:

    a.  Standard output format
    b.  Change bar format

# Advanced Maintenance Operations

# SOLUTIONS

```
         >
1.a.  >ALL DK0:
      >MCR FMT DK0:/VE

      ** WARNING - Data will be lost on DK0: **

      Continue? [Y OR N]: Y

      Start formatting

      Start verification

      Operation complete


      >MCR BAD DK0:/LI
   b. BAD -- DK0: Total bad blocks= 0.
         >
         :


      >INITIALIZE
   c. Device? DK0:
      Label? STUDENT
         >
         >



         >
   d.    >
      >CREATE/DIRECTORY/PROTECTION:(SY:,OW:RWED,GR:,WO:)
      Device, UFD? DK0:[305,300]
      UFD -- Volume not mounted
      >MOUNTDK0:
      MCR -- Syntax error
      >MOUNT DK0:
      Volume ID? STUDENT

         >
      >CREATE/DIRECTORY/PROTECTION:(SY:,OW:RWED,GR:,WO:)
      Device, UFD? DK0:[305,300]
      >DIR *.TXT

         :
      >MOUNT DK0:ELIZABETH
      >BACKUP
      From? DK0:
      To? MM0:
      BAC -- *FATAL* -- Privilege violation on DK0:
      I/O error code -16

   e. >BACKUP/MOUNTED
      From? DK0:
      To? MM0:
      BAC - Starting Tape 1 on MM0:

      BAC - End of Tape 1 on MM0:

      BAC - Completed
```

# SOLUTIONS

2.
```
>MCR FLX
FLX>MM0!/ZE/RW
FLX>/RS
FLX>MM0)-[305,303]*,*
FLX:MM0)/RW/LI


Directory       MM0![305,303]
09-Jan-82

FILES.MAC            2,      09-Jan-82 <233>
MYLIB.MLB            9,      09-Jan-82 <233>
DOC.MRO             1,      09-Jan-82 <233>
PROSE.TXT           8,      09-Jan-82 <233>
MAIN.FOR            4,      09-Jan-82 <233>
MORE.TXT            2,      09-Jan-82 <233>
COMPARE.TXT         2,      09-Jan-82 <233>
CMP.TXT             3,      09-Jan-82 <233>
NEW.TXT             2,      09-Jan-82 <233>
INPUT.CMD           2,      09-Jan-82 <233>
SYMBOLS.CMD         2,      09-Jan-82 <233>
CONTROL.CMD         2,      09-Jan-82 <233>
SPECIAL.CMD         2,      09-Jan-82 <233>
OPERATING.CMD       3,      09-Jan-82 <233>
ASSEMBLE.CMD        1,      09-Jan-82 <233>
TEST.LST            2,      09-Jan-82 <233>
LOGICAL.CMD         2,      09-Jan-82 <233>
TEST.MAC            1,      09-Jan-82 <233>


FLX>/DO
FLX>=MM0!PROGRAM.MAP
FLX>^Z
```

# Advanced Maintenance Operations

# SOLUTIONS

```
3.    DRO:[305,303]COMPARE.TXT;2 27 lines
  a. >DIFFERENCE COMPARE.TXT;1 COMPARE.TXT;2
     ************************************************
       1)  DRO:[305,303]COMPARE.TXT;1
         15           GENERATE A LISTING IN THE FORM OF ONE LIST,
     ****************
       2)  DRO:[305,303]COMPARE.TXT;2
         15   THESE LINES WILL BE THE DIFFERENCE BETWEEN THE TWO FILES
         16   THE DIFFERENCE COMMAND IS EXCELLENT FOR COMPARING THE DIFERENCES
         17   BETWEEN TWO VERSIONS OF PROGRAM SOURCE FILLES TO DETERMINE
         18   WHAT EDITS HAVE BEEN MADE.
         19
         20           GENERATE A LISTING IN THE FORM OF ONE LIST,

         1 differences found
     TI:/BL/FF/-CB=COMPARE.TXT;1,COMPARE.TXT;2
     >
  b. >DIFFERENCE/CHANGE_BAR COMPARE.TXT;1 COMPARE.TXT;2
          1           THIS IS A TEST TO SHOW THE RESULTS OF THE COMPARE
          2           FILE UTILITY (CMP),  THIS UTILITY COMPARES TWO
          3           ASCII FILES.  THE FILES ARE COMPARED LINE BY LINE TO
          4           DETERMINE WHETHER PARALLEL RECORDS ARE IDENTICAL.
          5           USING CMP, YOU CAN PERFORM THE FOLLOWING FILE-
          6           COMPARE FUNCTIONS:
          7
          8               GENERATE A LISTING SHOWING THE DIFFERENCES
          9               BETWEEN THE TWO FILES.  EACH DIFFERENCE IS
         10               LISTED AS A PAIR; FIRST, THE LINES FROM
         11               THE FIRST FILE THAT ARE BEING COMPARED TO
         12               LINES IN THE SECOND FILE, THEN THE LINES
         13               FROM THE SECOND FILE.
         14
         15  !    THESE LINES WILL BE THE DIFFERENCE BETWEEN THE TWO FILES
         16  !    THE DIFFERENCE COMMAND IS EXCELLENT FOR COMPARING THE DIFERENCES
         17  !    BETWEEN TWO VERSIONS OF PROGRAM SOURCE FILLES TO DETERMINE
         18  !    WHAT EDITS HAVE BEEN MADE.
         19  !
         20  !        GENERATE A LISTING IN THE FORM OF ONE LIST,
         21           WITH DIFFERENCES MARKED BY CHANGE BARS.
         22
         23           GENERATE OUTPUT SUITABLE FOR INPUT TO THE SLP
         24           UTILITY.  THIS OUTPUT CONTAINS THE SLP COMMANDS
         25           AND INPUT REQUIRED TO MAKE THE FIRST INPUT
         26           FILE IDENTICAL TO THE SECOND INPUT FILE.
         27

         1 differences found
     TI:/BL/FF/CB/VB:041=COMPARE.TXT;1,COMPARE.TXT;2
```