

Overview

About This Course.....	1
Introduction.....	1
Course Description	2
Prerequisites	3
Resources	3
Course Organization	4
Course Goals	5
Course Nongoes	5
Course Conventions.....	6

Module 1 — Logging In and Out of an OpenVMS System

About This Module.....	2
Introduction.....	2
Objectives.....	2
Resources	2
Gaining Access to an OpenVMS System	3
Overview	3
Local Access through a Directly Connected Terminal.....	4
Text-based Login.....	5
Example: Successful Login.....	6
Example: Unsuccessful Login.....	6
Graphical Login.....	8
DECwindows or DECwindows Motif Login.....	8
Common Desktop Environment Login.....	9
Network Access through Dialup	10
Network Access through a Serial Line and the Point-to-Point Protocol Utility (PPPD)	10
Example: PPPD Login	10
Example Notes: PPPD Login Example	10
Network Login through a Terminal Server.....	11
Overview	11
Example: Login through a Terminal Server.....	12
Example Notes: Login through a Terminal Server.....	12
Terminal Server Commands.....	13
SHOW SERVICES Command.....	13
Example: List of Services	13
Example Notes: List of Services	13
Connecting to a Service	13
Example: Connecting to a Service Successfully	13
Example: Connecting to a Service Unsuccessfully	14
Displaying a List of Active Sessions.....	15

Example: SHOW SESSIONS Command	15
Disconnection during a Session	16
Example: Disconnected Session.....	16
When Reconnection Works.....	16
Multiple Sessions.....	17
How many Sessions may I have?	17
How to Create Another Session	17
How Sessions are Organized.....	17
Moving Between Sessions	17
Example: Multiple Sessions	18
Example Notes: Multiple Sessions	19
The Forward Switch Character.....	20
Example: Changing Sessions	20
Duration of the Forward Switch Character	20
Example: Using a Forward Switch Character	21
Example Notes: Using a Forward Switch Character	21
Deleting a Forward Switch Character.....	22
Network Login through the OpenVMS SET HOST Command.....	22
Example: SET HOST command	22
Example Notes: SET HOST Command.....	22
Capturing Screen Output of Remote Sessions	22
How the System Validates a User	23
Identifying Users	23
User Names	23
Passwords.....	24
Successful Validation	25
Unsuccessful Validation	26
Using Simple User Security Techniques	27
Overview	27
Changing a User Password.....	27
Password Screening by the System	27
Reasons for Changing a Password	27
Guidelines for Selecting a Password	28
How to Set a User Password.....	29
Example: Using the SET PASSWORD Command.....	29
Example: Possible Errors in Setting a Password.....	30
Example Notes: Possible Errors in Setting a Password.....	31
Example: Password about to Expire	31
Example: Password Expired	31
Example: System-Generated Passwords.....	32
Example Notes: System-Generated Passwords.....	32
Locking a Terminal Session on a Terminal Server	33
Making Sure Your Terminal Is Secure	33
Example: Locking Your Terminal.....	33
Example Notes: Locking Your Terminal.....	33
Locking a DECwindows Session	33
Locking a Common Desktop Environment Session.....	33
Terminating an OpenVMS Session	34

Overview	34
Logging Out of OpenVMS.....	35
LOGOUT Command	35
Example: Displaying Accounting Information at Logout	35
Example: Logging Out of a Terminal Server.....	35
Exiting DECwindows	36
Exiting the Common Desktop Environment.....	36
Exiting a PPP Session.....	36
Exiting a Telnet Session.....	36
Summary	37
Concepts.....	37
Gaining Access to an OpenVMS System	37
Accessing a Remote OpenVMS System.....	37
Using Simple User Security Techniques	37
Terminating an OpenVMS Session	37
Commands.....	38
Gaining Access to an OpenVMS System	38
Accessing a Remote OpenVMS System.....	38
Using Simple User Security Techniques	39
Terminating an OpenVMS Session	39

Module 2 — Issuing Commands

About This Module.....	2
Introduction	2
Objectives.....	2
Resources	2
Constructing a DCL Command.....	3
Overview	3
Constructing DCL Commands	4
DCL Commands.....	4
Components of a DCL Command.....	4
Figure: DCL Command Structure	5
Figure Notes: DCL Command Structure	5
Rules for Constructing a DCL Command.....	6
Modifying DCL Commands.....	7
Types of Qualifiers	7
Command Qualifiers.....	7
Positional Qualifiers	7
Using DCL to Interact with the System.....	8
Overview	8
Entering DCL Commands.....	9
Long Command Lines	9
Abbreviating DCL Commands.....	9
Comment Character.....	10
How the Comment Character Works.....	10
Understanding System Messages	11
Error Messages	11
Figure: Format of a System Message	12

Figure Notes: Format of a System Message	12
Severity Levels	13
HELP/MESSAGE Utility.....	14
Using DCL Features	15
Overview	15
DCL Command Line Prompting	16
Example: DCL Prompting	16
Using Wildcards	17
File Specification Wildcards	17
Uses for Wildcards	17
Using the Control Key	18
Control Key.....	18
Sample Control Key Sequences	18
Effect of Ctrl/O on Command Execution	18
Recalling DCL Commands	19
RECALL Command	19
Example: Displaying All Commands in the Recall Buffer.....	19
Example Notes: Displaying All Commands in the Recall Buffer.....	19
RECALL/PAGE Command	19
Recalling and Entering a Command.....	20
Up and Down Arrow Keys.....	20
Editing Command Lines	21
Entering Commands	21
Editing Commands	21
Enabling Command Line Editing.....	21
Two Editing Modes	21
Selecting an Editing Mode.....	21
Control Keys Used in Command Line Editing	22
Example: Command Line Editing Session	23
Combining DCL Commands.....	24
Using PIPE to Combine DCL Commands.....	24
Format of the PIPE Command	24
Examples of the PIPE Command	24
Summary	25
Concepts.....	25
Constructing a DCL Command.....	25
Using DCL to Interact with the System.....	25
Using DCL Features	25
Commands.....	26

Module 3 — Naming and Storing Files

About This Module	2
Introduction	2
Objectives.....	2
Resources	2
Constructing an OpenVMS File Specification.....	3
Overview	3
OpenVMS File Specifications.....	4

File Specification.....	4
Format of a File Specification.....	4
Parts of a File Specification	4
Sample File Specifications.....	5
Guidelines for Naming Files	5
Creating and Identifying Subdirectories	6
Overview	6
OpenVMS Directory Structure.....	7
System and User Directories	7
Figure: Directory Structure	7
Figure Notes: Directory Structure	8
Maximum Number of Subdirectory Levels	8
Creating Subdirectories.....	9
Identifying Subdirectories.....	9
Rules for Naming Subdirectories	9
Creating a Subdirectory	10
Example: Creating a Subdirectory.....	10
Common Error in Creating a Subdirectory.....	10
Using Subdirectories.....	11
Uses for Subdirectories	11
Displaying Your Default Device and Directory.....	11
Navigating a Directory Structure	11
Specifying a Higher Directory Level.....	12
Example: Using SET and SHOW DEFAULT	12
Example Notes: Using SET and SHOW DEFAULT	12
Common Error: Setting Default to a Nonexistent Directory	14
Common Error: Using an Incomplete Subdirectory Name.....	14
Specifying Many Directory Levels.....	15
Displaying Information About Files	16
Overview	16
DIRECTORY Command	17
Listing Files.....	17
Example: DIRECTORY Display.....	17
Controlling the DIRECTORY Display.....	18
Displaying Detailed Directory Information	18
Example: Displaying Detailed Directory Information	19
Writing a Directory Listing to a File	19
Using Wildcards	20
Wildcard Characters	20
Example: Using Wildcard Characters	20
Summary	21
Concepts.....	21
Constructing an OpenVMS File Specification.....	21
Creating and Identifying Subdirectories	21
Displaying Information About Files.....	21
Commands.....	22
Creating and Identifying Subdirectories	22
Displaying Information About Files.....	22

Module 4 — Creating Memos, Reports, and Data Files

About This Module	2
Introduction	2
Objectives	2
Resources	2
Performing Basic EVE Operations	3
Overview	3
Invoking EVE	4
EDIT/TPU Command	4
Editing a New File	4
To edit an existing file:	4
Figure: EVE Screen Layout	5
Figure Notes: EVE Screen Layout	5
Entering EVE Commands	6
Predefined Keys	6
Figure: VT200 Keypad Definitions	7
Figure Notes: VT200 Keypad Definitions	7
Typing EVE Commands	8
Moving the Cursor	9
Cursor Movement Keys	9
Cursor Movement Commands	9
Creating and Modifying a Text File	10
Overview	10
Entering Text	11
General Text Entry Pattern	11
Two Modes of Text Entry	11
Replacing Text	12
REPLACE Command	12
Replacement Options	13
Requesting Help with EVE	14
Overview	14
EVE Online Help	15
How to Request Help	15
Kinds of Help Topics	15
Help Key	15
Using Some Advanced Features of EVE	16
Overview	16
Including Existing Files into a File	17
INCLUDE FILE Command	17
Example: Including a File	17
Executing DCL Commands from EVE	18
Entering a DCL Command from the EVE Command Line	18
SPAWN Command	18
Creating Alternate Key Definitions	19
Defining Your Keypad	19
Simulating the EDT Keypad	19
Simulating the WPS Keypad	20

Saving an Edited File.....	21
Overview	21
Ending a Session.....	22
Two Ways to End a Session.....	22
Quitting a Session.....	22
Exiting from a Session.....	22
Saving a File Without Ending Your Editing Session	23
Two Ways to Save a File	23
SAVE FILE Command.....	23
SAVE FILE AS Command	23
File Recovery.....	24
Journal File.....	24
Types of Journaling	24
How to Recover a Lost Editing Session	25
Summary	26
Concepts.....	26
Performing Basic EVE Operations.....	26
Creating and Modifying a Text File	26
Requesting Help with EVE	26
Using Some Advanced Features of EVE	26
Saving an Edited File.....	26
Commands.....	27
Performing Basic EVE Operations.....	27
Creating and Modifying a Text File	27
Requesting Help with EVE	27
Using Some Advanced Features of EVE	27
Saving an Edited File.....	28

Module 5 — Manipulating Files

About This Module.....	2
Introduction	2
Objectives.....	2
Resources	2
Making Copies of Files.....	3
Overview	3
COPY Command.....	4
Copying Files	4
Format of the COPY Command.....	4
Requirement for the COPY Command.....	4
Sample COPY Operations and Commands	5
Modifying the COPY Command.....	6
Qualifiers.....	6
Requesting Confirmation of the COPY Operation.....	6
Example: Copy Operation Using /CONFIRM	7
Example Notes: Copy Operation Using /CONFIRM	7
Displaying a Log of the COPY Operation	7
Example: Copy Operation Using /LOG.....	8
Example Notes: Copy Operation Using /LOG.....	8

Setting a New Protection Code on a Copied File	8
Example: Copy Operation Using /PROTECTION	8
Common Errors in Copying Files.....	9
Errors in Copying Files.....	9
Example: Error in Input File	9
Example: Error in Output File.....	9
Displaying the Contents of Files	10
Overview	10
TYPE Command.....	11
Displaying the Contents of a File	11
Format of the TYPE Command.....	11
Defaults for the TYPE Command	11
Format of the TYPE Command.....	11
Example: Using the TYPE Command	11
Requirements and Restrictions for the TYPE Command	12
Requirement for the TYPE Command.....	12
What Kinds of Files Can be Displayed.....	12
Files that Should Not be Typed.....	12
What to Do if You Display a Non-ASCII File.....	12
Different Ways to Use the TYPE Command	13
Displaying Multiple Files.....	13
Displaying One Screen at a Time	13
Displaying a File in a Different Location	13
Printing Files	15
Overview	15
PRINT Command	16
Printing Files	16
Format of the PRINT Command.....	16
Requirement for the PRINT Command	16
Entry Number	16
Modifying the Action of the PRINT Command.....	17
Qualifiers for the PRINT Command.....	17
Specifying a Queue.....	17
Specifying the Number of Copies Wanted	18
Asking the System to Notify on Completion	18
Displaying Information About Print Jobs	19
Monitoring Print Jobs	19
SHOW ENTRY Command.....	19
Example: Listing Queue Entries.....	19
Example: Displaying a Specific Queue Entry.....	19
Example: Displaying Detailed Queue Information	19
SHOW QUEUE Command.....	21
Example: Displaying Queue Information	21
Controlling Print Jobs	22
Controlling Print Jobs	22
Changing the Characteristics of Your Print Job.....	22
Job Characteristics You Can Change	22
Example: Changing Characteristics of a Print Job.....	23

Example: Holding and Releasing a Job	23
Removing Your Print Job from the Queue	24
Requirements for Deleting a Job	24
Example: Removing a Print Job from the Queue	25
Common Errors in Manipulating Print Jobs	26
Errors in Manipulating Print Jobs	26
Error: Nonexistent File	26
Error: Nonexistent Queue	26
Error: Nonexistent Job	27
Error: Nonexistent Entry	27
Renaming Files	28
RENAME Command	29
Renaming a File	29
Format of the RENAME Command	29
Requirements for the RENAME Command	29
Sample Rename Operations	29
Using Version Numbers with the RENAME Command	30
Modifying the RENAME Command	31
Qualifiers	31
Requesting Confirmation of the Rename Operation	31
Example: Rename Operation Using /CONFIRM	31
Displaying a Log of a Rename Operation	32
Example: Rename Operation Using /LOG	32
Common Errors in Renaming Files	33
Errors in Renaming Files	33
Example: Error in Input File	33
Example: Error in Output File	33
Removing Files from Disk	34
Overview	34
DELETE Command	35
Deleting Files	35
Format of the DELETE Command	35
Specifying Version Numbers	35
Requirements for the DELETE Command	35
Modifying the DELETE Command	36
Qualifiers	36
Requesting Confirmation of a Delete Operation	36
Requesting a Log of a Delete Operation	36
Common Errors in Deleting Files	37
Errors in Deleting Files	37
No Version Number Specified	37
Trying to Delete a Protected File	37
PURGE Command	38
Purging Files	38
Format of the PURGE Command	38
Requirement for the PURGE Command	38
Modifying the PURGE Command	39
Qualifiers	39

Requesting Confirmation of the Purge Operation.....	39
Example: Purge Operation Using /CONFIRM	39
Displaying a Log of the Purge Operation	40
Example: Purge Operation Using /LOG	40
Keeping More than One Version of a File.....	40
Example: Purge Operation Using /KEEP	41
Example: Combining PURGE Command Qualifiers	41
Common Errors in Purging Files.....	42
Errors in Purging Files.....	42
Example: Specifying a Version Number When Purging Files.....	42
Example: Trying to Purge a Protected File.....	42
Example: Trying to Purge a File in a Protected Directory.....	43
Erasing the Contents of Deleted Files.....	44
Contents of Deleted Files.....	44
Erasing Files.....	44
Summary	45
Concepts.....	45
Making Copies of Files.....	45
Displaying the Contents of Files	45
Printing Files	45
Renaming Files.....	46
Removing Files from Disk.....	46
Commands.....	47
Making Copies of Files.....	47
Displaying the Contents of Files	47
Printing Files	47
Renaming Files.....	47
Removing Files from Disk.....	47

Module 6 — Communicating with Other Users

About This Module.....	2
Introduction.....	2
Objectives.....	2
Resources	2
Using the Mail Utility	3
Overview	3
Getting Started with the Mail Utility	4
Overview	4
Invoking the Mail Utility	4
Mail Folders	4
Figure: Default Mail Folders.....	6
Example: Identifying Your Current Folder.....	6
Exiting from the Mail Utility	6
Displaying Information About Messages	7
Listing Mail Messages	7
Example: Listing Mail Messages	7
Example Notes: Listing Mail Messages	7
Displaying a List of All Your Folders	8

Specifying a Folder.....	8
Reading Messages	9
Notification About New Messages.....	9
Disabling Notification About New Messages.....	9
READ Command.....	10
Sample Read Operations.....	10
Controlling Display of Screen Information.....	11
Example: Reading a Mail Message	12
Example Notes: Reading a Mail Message	12
Sending Messages.....	13
Ways to Send a Message.....	13
SEND Command	13
Editing Your Message	14
Example: Sending a Mail Message	14
Example Notes: Sending a Mail Message.....	14
Sending a File as a Message.....	15
Copying People on Your Messages.....	15
Using a Distribution List to Send a Message	15
FORWARD Command.....	16
Example: Forwarding a Mail Message	16
Example Notes: Forwarding a Mail Message	16
Using a Signature File.....	17
REPLY Command.....	17
Example: Reading and Replying to a Mail Message.....	18
Example Notes: Reading and Replying to a Mail Message.....	18
Modifying the Action of the REPLY Command.....	19
Deleting Messages.....	20
DELETE Command.....	20
How a Deleted Message Is Handled.....	20
Sample DELETE Commands.....	20
Retrieving a Deleted Message.....	20
Example: Deleting and Retrieving a Mail Message	21
Example Notes: Deleting and Retrieving a Mail Message	21
PURGE Command	22
Additional Mail Features	23
Other Features	23
Multipurpose Internet Mail Extension (MIME) Utility.....	24
Purpose.....	24
Summary	25
Concepts.....	25
Using the Mail Utility	25
Commands.....	26
Using the Mail Utility	26
Using the MIME Utility.....	27

Module 7 — Getting Help

About This Module	2
Introduction	2
Objectives	2
Resources	2
Using Online Help	3
Overview	3
OpenVMS Help Facility	4
Overview	4
Selecting Topics in HELP	4
Ending a Help Session	4
Example: Help Session	5
Example Notes: Help Session	6
Selecting a Topic from the DCL Command Line	7
Faster Use of Help	7
Help Message Utility	8
HELP/MESSAGE Command	8
Using Status Codes	8
Resources	8
Using the OpenVMS Documentation Set	9
Overview	9
Components of OpenVMS Documentation	10
Three Sets of Documentation	10
Who Should Use Each Set of Documentation	10
Documentation for OpenVMS Alpha and OpenVMS VAX	11
OpenVMS Master Index	11
OpenVMS Glossary	11
Summary	12
Concepts	12
Using Online Help	12
Using the OpenVMS Documentation Set	12
Commands	13
Using Online Help	13

Module 8 — Protecting Your Data

About This Module	2
Introduction	2
Objectives	2
Resources	2
Using File Protection	3
Overview	3
Security Profiles	4
Protected Objects	4
Contents of a User Security Profile	4
Contents of an Object's Security Profile	4
User Identification Codes, UICs	5
Purpose of UICs	5

UIC Components.....	5
UIC Formats.....	5
Example: Displaying Your UIC.....	6
Example: Displaying the UIC of a File	6
Protection Codes.....	7
Types of Access.....	7
Types of File Access.....	7
User Categories	7
Assigning UICs to Categories	8
Figure: Elements of a Protection Code	9
Figure Notes: Elements of a Protection Code	9
Displaying File Security Information	10
Example: Displaying File Protection.....	11
Default Protection.....	12
Example: Displaying Default Protection	12
Example: Changing Default Protection	12
Example: Effect of Changing Default Protection.....	13
Example Notes: Effect of Changing Default Protection.....	13
Manipulating Protection Codes	14
Setting File Protection	14
Example: Setting File Protection.....	14
Protection on Directory Files	14
Steps in Deleting a Directory File	15
Example: Deleting a Subdirectory.....	15
Example Notes: Deleting a Subdirectory.....	15
Common Errors in Deleting Directory Files.....	16
Common Error: Deleting a Directory That Is Not Empty	16
Common Error: Deleting a Protected Directory File.....	16
Saving Copies of Files	17
Overview	17
Backup Utility	18
What the Backup Utility Can Do	18
BACKUP Command	18
Save Sets	18
Example: Making a Save Set of All Files in a Directory.....	19
Example: Listing the Contents of a Save Set.....	19
Restoring a File from a Save Set.....	20
Summary	21
Concepts.....	21
Using File Protection	21
Saving Copies of Files	21
Commands.....	22
Using File Protection	22
Saving Copies of Files	22

Module 9 — Customizing Your Working Environment

About This Module.....	2
Introduction.....	2
Objectives.....	2
Resources	2
Using Logical Names.....	3
Overview	3
Creating Logical Names.....	4
Commands Used to Create Logical Names	4
Rules for Creating Logical Names	4
Logical Names in Equivalence Strings.....	4
Duration of a Logical Name.....	4
Using Logical Names.....	5
Common Uses for Logical Names.....	5
Example: Using a Logical Name for a Directory Specification.....	5
Example: Using a Logical Name for a Complete File Specification	5
Example: Using a Logical Name in the Left Part of File Specification	6
Common Error in Using Logical Names	6
Using Logical Names with the Mail Utility.....	7
Determining the Equivalence of a Logical Name	8
Translation of Logical Names	8
SHOW TRANSLATION Command.....	8
Example: SHOW TRANSLATION Command	8
SHOW LOGICAL Command.....	8
Example: SHOW LOGICAL Command	9
Removing Logical Names.....	10
DEASSIGN Command	10
Delete All Process Logical Names	10
Logical Names the System Creates for You	11
Logical Names Created by the System.....	11
SYS\$LOGIN	11
SYS\$LOGIN_DEVICE	11
Using Symbols.....	12
Overview	12
Introduction to Symbols.....	13
Symbols.....	13
Local Symbols.....	13
Global Symbols	13
Duration of Symbols.....	13
Data Types for Symbols.....	14
Creating Symbols.....	15
Equating Symbols to Character Strings	15
Sample String Assignments	15
Equating Symbols to Integers	15
Creating a Symbol as a Command Synonym.....	16
Creating Symbols that Can Be Abbreviated	16
Displaying Symbols.....	17

Sample SHOW SYMBOL Commands.....	17
Deleting Symbols.....	17
DELETE/SYMBOL Command	17
Using Keypad Definitions.....	18
Overview	18
Creating Keypad Definitions.....	19
Keys that Can Be Defined.....	19
Defining Keys.....	19
Scope and Duration of Keypad Definitions	19
Tailoring Keypad Definitions.....	20
Example: Enabling and Defining a Key	20
Example Notes: Enabling and Defining a Key	20
Common Errors in Defining Keys.....	20
Displaying Keypad Definitions	21
Displaying Keypad Definitions.....	21
Example: Displaying a Specific Keypad Definition.....	21
Example: Displaying Detailed Keypad Definitions	21
Example: Displaying All Keypad Definitions	21
Deleting Keypad Definitions.....	22
Deleting a Single Keypad Definition.....	22
Deleting All Keypad Definitions.....	22
Writing Command Procedures	23
Overview	23
Developing Command Procedures	24
Steps for Developing Command Procedures	24
Rules for Developing Command Procedures.....	24
Guidelines for Developing Command Procedures.....	25
Example: Command Procedure.....	25
Executing a Command Procedure	26
How a Command Procedure Is Executed	26
Example: Executing a Command Procedure	26
Testing Your Command Procedure.....	26
Example: Using SET VERIFY	27
Example Notes: Using SET VERIFY.....	27
Halting Execution.....	28
Executing a Command Procedure in Batch	28
Writing a Login Command Procedure.....	29
LOGIN.COM	29
Bypassing LOGIN.COM	29
Example: LOGIN.COM.....	30
Example Notes: LOGIN.COM.....	31
Common Errors in Login Command Procedures	31
Summary	33
Concepts.....	33
Using Logical Names.....	33
Using Symbols	33
Using Keypad Definitions.....	33
Writing Command Procedures	33

Commands.....	34
Using Logical Names.....	34
Using Symbols	34
Using Keypad Definitions.....	34
Writing Command Procedures.....	35

Learning Check

Learning Check Answers

Understanding System Resources

Module 1 Labs

Virtual Memory Management Concepts

Written Exercise: Memory Management

Supply the best answer to the following questions:

1. One page of physical memory on a VAX computer consists of ____ bytes.
.....
2. Virtual memory is divided into these two spaces:
.....
.....
3. Each process has a _____ that maps virtual pages to physical pages.
.....
4. The set of physical pages of memory currently being used by a process is known as the.
.....
5. A page fault occurs when a process refers to a page that
.....
.....
6. A process can be in one of these three states:
.....
.....
.....
7. The two types of nodes in a DECnet for OpenVMS network are a routing node and an:
.....
.....

Written Exercise Solution: Memory Management

1. One page of physical memory on a VAX computer consists of 512 bytes.
2. Virtual memory is divided into these two spaces: System space and process space.
3. Each process has a page table that maps virtual pages to physical pages.
4. The set of physical pages of memory currently being used by a process is known as the working set.
5. A page fault occurs when a process refers to a page that is not in its working set.
6. A process may be in one of these three states: Current, computable, or waiting.
7. The two types of nodes in a DECnet for OpenVMS network are a routing node and an end node.

Managing Batch and Print Requests

Module 2 Labs

Managing Queues and Spooled Devices

Written Exercise 2–1: Queue Creation

1. List the five steps taken to establish a print execution queue.
.....
.....
.....
.....
.....
2. Write the DCL command that you would use to display all device queues.
.....
3. Write the DCL command that you would use to create and start a print queue.
.....

Written Solution 2–1: Queue Creation

1. List the five steps taken to establish a print execution queue.
 - a. Set physical attributes of device.
 - b. Spool each printing device.
 - c. Initialize and start an execution queue for each printing device.
 - d. Initialize and start a generic print queue.
 - e. Initialize and start a generic terminal queue.
2. Write the DCL command that you would use to display all device queues.
`SHOW QUEUE/DEVICE`
3. Write the DCL command that you would use to create and start a print queue.
`INITIALIZE/QUEUE/START/ON=LPA0 SYS$PRINT`

Lab Exercise 2–2: Queue Management

The following exercises allow you to practice DCL commands needed to create, manipulate, view, and delete queues.

Perform the following laboratory exercises at an interactive terminal.

Use the command procedure LOGIN.COM in your main directory as a sample to place in the print and batch queues.

1. Place your LOGIN.COM in two print queues. Include the /HOLD qualifier so the jobs will not execute.
2. Examine the print queues.
3. Place several copies of the LOGIN.COM file into the batch queues. Include the /HOLD qualifier so the jobs will not execute.

The system returns a job number for each SUBMIT command you enter. Record these numbers for later use.

4. Examine the batch queues.
Notice the different information obtained from each batch queue command. Examine the information given with each job.
5. Using the job entry numbers for each print and batch job, delete those jobs from the queues.

Lab Solution 2–2: Queue Management

1. Place your LOGIN.COM in two print queues. Include the /HOLD qualifier so the jobs will not execute.

To place several entries in the print queues, enter the following commands:

```
$ PRINT/HOLD LOGIN.COM
$ PRINT/HOLD/QUEUE=LPA0 LOGIN.COM
```

2. Examine the print queues.

```
$ SHOW QUEUE queue_name
$ SHOW QUEUE/DEVICES/ALL
```

3. Place several copies of the LOGIN.COM file into the batch queues. Include the /HOLD qualifier so the jobs will not execute.

The system returns a job number for each SUBMIT command you enter.

Record these numbers for later use.

```
$ SUBMIT/HOLD LOGIN.COM
```

4. Examine the batch queues.

```
$ SHOW QUEUE/BATCH
$ SHOW QUEUE/BATCH/FULL
```

Notice the different information obtained from each batch queue command. Examine the information given with each job.

5. Using the job entry numbers for each print and batch job, delete those jobs from the queues.

```
$ DELETE/ENTRY=job_entry_number queue_name
```

For example, if the system returns:

```
JOB SYSTEM (queue SYS$PRINT, entry 390) holding
```

delete the job with the DCL command:

```
$ DELETE/ENTRY=390 SYS$PRINT
```

Lab Exercise 2–3: Batch Queues

Preparation for this lab exercise:

- Use an editor or the CREATE command to create a command procedure that you can use in the following exercise. Name this file LOOP.COM. Include the following commands:
\$ LOOP :
\$ SHOW TIME
\$ SHOW SYSTEM
\$ SHOW QUEUE
\$ SHOW ENTRY/FULL
\$ GOTO LOOP
1. Use one command to create and start a batch queue called username_BATCH (for example: MGR1_BATCH).

If you get an error message indicating insufficient privilege, remind your instructor that you need OPER privilege to complete this laboratory exercise.
 2. Submit LOOP.COM on hold to be run as a batch job in your batch queue.
 3. Look at the contents of the queue username_BATCH. The first line of output from a SHOW QUEUE command lists the type, name, and status of the queue. Get used to checking this line to see if the queue is stopped or paused.

For the rest of the laboratory exercise, look at the contents of all batch queues whenever you change anything.
 4. Create and start a second batch queue called SLO_username (for example: SLO_MGR1) with a base priority of 2. This time use separate commands to create and start the queue.
 5. Submit LOOP.COM as a batch job to the SLO_username queue. Examine its priority.
 6. Use the SHOW SYSTEM command to see the name of the batch process executing LOOP.COM.
 7. Submit LOOP.COM on hold to the SLO_username queue.
 8. Submit LOOP.COM to the SLO_username queue.
 9. Submit LOOP.COM on hold to the username_BATCH queue.
 10. Stop the username_BATCH and SLO_username queues, canceling the executing jobs.
 11. Delete the holding entries in the queues.
 12. Delete the username_BATCH and SLO_username queues. List all batch queues.

Lab Solution 2–3: Batch Queues

1. Use one command to create and start a batch queue called `username_BATCH`.
`$ INITIALIZE/QUEUE/BATCH/START username_BATCH`
2. Submit `LOOP.COM` on hold to be run as a batch job in your batch queue.
`$ SUBMIT/HOLD/QUEUE=username_BATCH LOOP.COM`
3. Look at the contents of the queue `username_BATCH`. The first line of output from a `SHOW QUEUE` command lists the type, name, and status of the queue. Get used to checking this line to see if the queue is stopped or paused.

`$ SHOW QUEUE username_BATCH`

The following command gives additional information:

`$ SHOW QUEUE/FULL username_BATCH`

The status column indicates the status of the job you submitted.

4. Create and start a second batch queue called `SLO_username` with a base priority of 2. This time use separate commands to create and start the queue.

`$ INITIALIZE/QUEUE/BATCH/BASE_PRIORITY=2 SLO_username`

`$ START/QUEUE SLO_username`

`$ SHOW QUEUE/FULL/BATCH`

5. Submit `LOOP.COM` as a batch job to the `SLO_username` queue. Examine its priority.

`$ SUBMIT/QUEUE=SLO_username LOOP.COM`

`$ SHOW QUEUE/FULL SLO_username`

Notice that the priority of the job is different from the priority of the queue. Again, note the status of the job you submitted.

6. Use the `SHOW SYSTEM` command to see the name of the batch process executing `LOOP.COM`.

`$ SHOW SYSTEM`

The process name is `BATCHxxx`, where `xxx` is the job entry number.

7. Submit LOOP.COM on hold to the SLO_username queue.

```
$ SUBMIT/HOLD/QUEUE=SLO_username LOOP.COM  
$ SHOW QUEUE/BATCH
```

The order of the qualifiers is not important, as long as they are all included in the command.

8. Submit LOOP.COM to the SLO_username queue.

```
$ SUBMIT/QUEUE=SLO_username LOOP.COM  
$ SHOW QUEUE SLO_username
```

9. Submit LOOP.COM on hold to the username_BATCH queue.

```
$ SUBMIT/HOLD/QUEUE=username_BATCH LOOP.COM  
$ SHOW QUEUE/BATCH
```

10. Stop the username_BATCH and SLO_username queues, canceling the executing jobs.

```
$ STOP/QUEUE/RESET SLO_username  
$ STOP/QUEUE/RESET username_BATCH  
$ SHOW QUEUE/BATCH
```

11. Delete the holding entries in the queues.

```
$ DELETE/ENTRY=entry-number queuename
```

Repeat this command for each entry on hold in the queues.

12. Delete the username_BATCH and SLO_username queues. List all batch queues.

```
$ DELETE/QUEUE SLO_username  
$ DELETE/QUEUE username_BATCH  
$ SHOW QUEUE/FULL/BATCH
```

Lab Exercise 2–4: Print Queues

The following exercises allow you to practice DCL commands needed to create, manipulate, view, and delete print queues. Keep in mind that the commands used are similar to the commands for batch queues, except they are missing the /BATCH qualifier.

Preparation for this lab exercise:

- Copy your LOOP.COM file to A.DAT so you will have two files to work with.
- 1. Create and start a generic queue named SYS\$PRINT.
- 2. Create a print execution queue. SYS\$PRINT sends jobs to the queue you create. The print execution queue should be given the name of the fastest printer on your system (or the printer assigned to you for the course). All jobs printed through this queue should include a flag page. Start the queue.
- 3. Create the symbol SQ to look at the contents of each device queue.
`$ SQ=="SHOW QUEUE/FULL/DEVICE"`
Use this symbol instead of the command. The symbol is faster to enter, and you will learn more about queues if you look at their contents often (especially after each change).
- 4. Send A.DAT on hold to the SYS\$PRINT generic queue to print.
- 5. Send LOOP.COM directly to the print execution queue to print. Look at the contents of each device queue (quickly).
- 6. Release the A.DAT job in the SYS\$PRINT queue so that it will print.
- 7. Look at the contents of the device queues, and check the line printer output.

8. While the files are printing, pretend that the line printer is jammed. Fix the line printer, and requeue the file.
9. Abort the current job printing on the line printer. Display the contents of the queue and note the status of the job you just aborted.
10. Enable your terminal as an operator's console.
11. Print the file A.DAT on SYS\$PRINT, using the /OPERATOR qualifier to send a message to the operator.
12. When notified that operator assistance is requested, enter a SHOW QUEUE/FULL command on the print execution queue. Note the first line, which reports the status of the device.
13. Start the queue again.
14. Display the contents of the execution queue.

Lab Solution 2–4: Print Queues

The *n* in some of the answers corresponds to the entry number given to the job when you entered the PRINT command. Each job has a unique entry number. When you enter these commands to see the answers, substitute the entry numbers given by your system in the appropriate places.

Replace the device name LPA0: with the device name of the printer you are using so the command will work on your system.

1. Create and start a generic queue named SYS\$PRINT.
\$ INITIALIZE/QUEUE/GENERIC=(LPA0:)/START SYS\$PRINT
2. Create a print execution queue. SYS\$PRINT sends jobs to the queue you create. The print execution queue should be given the name of the fastest printer on your system (or the printer assigned to you for the course). All jobs printed through this queue should include a flag page. Start the queue.

```
$ INITIALIZE/QUEUE/SEPARATE=FLAG/START LPA0
```

3. Create the symbol SQ to look at the contents of each device queue.

```
$ SQ=="SHOW QUEUE/FULL/DEVICE"  
$ SQ
```

The symbol lists the complete contents of all device queues. You should see two queues, named LPA0 and SYS\$PRINT.

4. Send A.DAT on hold to the SYS\$PRINT generic queue to print.

```
$ PRINT/QUEUE=SYS$PRINT/HOLD A.DAT
```

The PRINT command automatically sends jobs to the SYS\$PRINT queue if it exists, so the following command does the same thing:

```
$ PRINT/HOLD A.DAT
```

5. Send LOOP.COM directly to the print execution queue to print. Look at the contents of each device queue (quickly).

```
$ PRINT/QUEUE=LPA0 LOOP.COM  
$ SQ
```

The contents of the queues should be A.DAT in SYS\$PRINT on hold, and LOOP.COM in LPA0 with printing status.

6. Release the A.DAT job in the SYSS\$PRINT queue so that it will print.
\$ SET ENTRY/RELEASE *n*
(Where *n* is the entry number)
7. Look at the contents of the device queues, and check the line printer output.
\$ SQ
 - A.DAT should have printing status on the LPA0 queue.
 - LOOP.COM should have printed first, followed by the file A.DAT.
8. While the files are printing, pretend that the line printer is jammed. Fix the line printer, and requeue the file.
\$ STOP/QUEUE/REQUEUE LPA0
9. Abort the current job printing on the line printer. Display the contents of the queue and note the status of the job you just aborted.
\$ STOP/QUEUE/ABORT LPA0
10. Enable your terminal as an operator's console.
\$ REPLY/ENABLE
11. Print the file A.DAT on SYSS\$PRINT:, using the /OPERATOR qualifier to send a message to the operator.
\$ PRINT/OPERATOR="Long job coming through -- check paper."
12. When notified that operator assistance is requested, enter a SHOW QUEUE/FULL command on the print execution queue. Note the first line, which reports the status of the device.
\$ SHOW QUEUE/FULL LPA0
13. Start the queue again.
\$ START/QUEUE LPA0
14. Display the contents of the execution queue.
\$ SHOW QUEUE/FULL LPA0

Setting Up a Remote Print Queue

Lab Exercise: Setting Up a Remote Print Queue

This exercise allows you to practice setting up a terminal as a virtual print queue. It requires two terminals and privileged access to a terminal server.

1. Choose one of the terminals to act as a printer. It does not have to be a hardcopy terminal. Set up a service on the terminal server called `username_PRINT`, using your own user name. Assign that service to the terminal port to which your chosen terminal is connected. Also, make sure the port characteristics are set correctly for printing.
2. On an OpenVMS system, use LATCP to set up a port that can be used to print to your chosen terminal.
3. On the same system, set up a print queue that queues jobs to the port you created with LATCP. Give the queue the same name as your user name.
4. Test your work by submitting a small text file to the queue you created. It should appear on your chosen terminal.
5. Delete the print queue and the LAT port on the OpenVMS system.
6. Delete the service you created on the terminal server, and restore the terminal characteristics so that it functions as an interactive terminal. Log in to the terminal to make sure it is usable.

Lab Solution: Setting Up a Remote Print Queue

The following solution uses port number 6. Substitute your printer port number for 6.

1. Choose one of the terminals to act as a printer. It does not have to be a hardcopy terminal. Set up a service on the terminal server called `username_PRINT`, using your own user name. Assign that service to the terminal port to which your chosen terminal is connected. Also, make sure the port characteristics are set correctly for printing.

```
Local> SET PRIVILEGED
```

```
Password>
```

```
Local> SET PORT 6 ACCESS REMOTE
```

```
Local> DEFINE SERVICE SYSNET9_PRINT PORT 6 "Printer port"
```

```
Local> SET SERVICE SYSNET9_PRINT PORT 6 "Printer port"
```

```
Local> LOGOUT PORT 6
```

2. On an OpenVMS system, use LATCP to set up a port that can be used to print to your chosen terminal.

This example uses LAT unit number 24. Substitute any available unit number.

```
$ RUN SYS$SYSTEM:LATCP
```

```
LATCP> CREATE PORT LTA24:/APPLICATION
```

```
LATCP> SET PORT LTA24:/SERVICE=SYSNET9_PRINT
```

3. On the same system, set up a print queue that queues jobs to the port you created with LATCP. Give the queue the same name as your user name.

```
$ SET DEVICE/SPOOLED LTA24:
```

```
$ INITIALIZE/QUEUE/ON=LTA24:/START SYSNET9
```

4. Test your work by submitting a small text file to the queue you created. It should appear on your chosen terminal.

```
$ PRINT/QUEUE=SYSNET9 SYS$MANAGER:LAT$SYSTARTUP.TEMPLATE
```

5. Delete the print queue and the LAT port on the OpenVMS system.

```
$ STOP/QUEUE/NEXT SYSNET9
```

```
$ DELETE/QUEUE SYSNET9
```

```
$ SET DEVICE/NOSPOOLED LTA24:
```

```
$ RUN SYS$SYSTEM:LATCP
```

```
LATCP> DELETE PORT LTA24:
```

```
LATCP> EXIT
```

```
$
```

6. Delete the service you created on the terminal server, and restore the terminal characteristics so that it functions as an interactive terminal. Log in to the terminal to make sure it is usable.

[Break]

```
Local> CLEAR SERVICES SYSNET9_PRINT
```

```
Local> PURGE SERVICES SYSNET9_PRINT
```

```
Local> SET PORT 6 ACCESS LOCAL
```

```
Local> LOGOUT PORT 6
```

```
Local>
```


Preparing Volumes for Use

Written Exercise 4–1: DCL Commands

Supply the DCL command to perform the following actions:

- | | | |
|----|-------|---|
| a. | | Clear the volume and establish volume parameters. |
| b. | | Establish volume parameters without clearing the volume. |
| c. | | Make a volume known to the system. |
| d. | | Display full information regarding a device, including volume protection codes. |
| e. | | Reserve a device for your process |
| f. | | Mount disk in drive \$2\$DUA1, with the label WORK1, for system access. |

Written Solution 4–1: DCL Commands

Supply the DCL command to perform the following actions:

INITIALIZE	Clear the volume and establish volume parameters.
SET VOLUME	Establish volume parameters without clearing the volume.
\$MOUNT	Make a volume known to the system.
\$SHOW DEVICE/FULL	Display full information regarding a device, including volume protection codes.
\$ALLOCATE	Reserve a device for your process.
\$MOUNT/SYSTEM \$2\$DUA1 WORK1	Mount disk in drive \$2\$DUA1, with the label WORK1, for system access.

Introduction to Laboratory Exercise 4–2

This lab can be done as a group exercise.

You need a disk drive and a disk volume, or a tape drive and a tape volume to complete these exercises.

Lab Exercise 4–2: Create a Public Volume

1. Initialize and mount a disk volume or a tape volume for public use.
 - a. Allocate the device specified by your to your process.
 - b. Physically load the volume on the device. If you are not sure how to load the volume, ask your
 - c. Initialize the disk or tape volume so that all users will have access to it.
 - d. Mount the disk or tape volume so that all users on the system will have access to it without having to mount it themselves.
 - e. Display the characteristics of the device on which you have just mounted the volume.
2. Dismount the volume and de-allocate the device so that it is available for other users.

Lab Solution 4–2: Create a Public Volume

Substitute the name of the device you are using for DRA1:, your own logical name for DRIVE, and your own label for MYVOL.

1. Initialize and mount a disk volume or a tape volume for public use.
 - a. Allocate the device specified by your to your process.

```
$ ALLOCATE DRA1: DRIVE
```
 - b. Physically load the volume on the device. If you are not sure how to load the volume, ask your
No solution needed.
 - c. Initialize the disk or tape volume so that all users will have access to it.

```
$ INITIALIZE/SYSTEM DRIVE MYVOL
```
 - d. Mount the disk or tape volume so that all users on the system will have access to it without having to mount it themselves.

```
$ MOUNT/SYSTEM/NOUNLOAD DRIVE MYVOL
```
 - e. Display the characteristics of the device on which you have just mounted the volume.

```
$ SHOW DEVICE/FULL DRIVE
```
2. Dismount the volume and de-allocate the device so that it is available for other users.

```
$D ISMOUNT/UNLOAD DRIVE  
$ DEALLOCATE DRA1:
```

Introduction to System Customization

Module 5 Labs

Customizing System Startup Files

Lab Exercise 5–1: System Startup Files

1. Type (or print) the following system startup files and examine them:
 - a. SYS\$MANAGER:SYLOGICALS.COM
 - b. SYS\$MANAGER:SYPAAGSWPFILES.COM
 - c. SYS\$MANAGER:SYCONFIG.COM
 - d. SYS\$MANAGER:SYSTARTUP_VMS.COM
2. (Optional) Create TERMINALS.COM in your own directory. This procedure should set the permanent characteristics of the terminals on your system according to the following information.
 - a. Assume that you have eight terminals.
 - b. Set up some fast terminals and some slow terminals.
 - c. Protect at least three terminals from allocation by processes other than those with a [001,004] UIC.
 - d. Four of the terminals are VT200 series terminals.
 - e. Three are VT300 series terminals.
 - f. One is an LA120 hardcopy terminal.
 - g. The LA120 is attached through a modem.
 - h. The other terminals are attached through direct lines.
 - i. Include the name of the owner and office number where the terminal is located in a comment for each terminal.

3. (Optional) Create an alternate SYSTARTUP_VMS.COM in your own directory. Do not execute this procedure to verify your work. Instead, look at the answers provided in this chapter. The procedure should include commands to do the following:
 - a. Mount the class disk (label = CLASS) on the device DUA1:. Be sure it will be accessible to all users. Assign the disk the logical name CLASS_DISK.
 - b. Assume that a directory named [PROGRAMS] has been created on the system disk to contain site-specific programs. Define a system logical name for this directory.
 - c. Create and start the following queues (if possible):
 - ♦ SY\$PRINT (generic queue)
A print execution queue
Make sure the printer supports lowercase characters and includes a flag page on each job printed.
 - ♦ SY\$BATCH (batch execution queue)
Set the job limit at 2 and the priority at 3.
 - ♦ BIGJOB (batch execution queue)
Set the job limit at 1 and the priority at 2.
Do not start this queue.
 - d. Invoke SY\$MANAGER:TERMINALS.COM.
 - e. Define the logical names SY\$ANNOUNCE and SY\$WELCOME.
 - f. Rename the second highest version of the operator log to OPERATOR.OLD. Print OPERATOR.OLD, and have the system delete the file for you after it has been printed.
 - g. Restrict the number of interactive users to 35.
 - h. Send a message to all terminals telling users that the system is now up and ready for use, and log out.

Lab Solution 5–1: System Startup Files

1. Type (or print) the following system startup files and examine them.

.
.
.

The purpose of looking at the system startup files is to familiarize yourself with the contents of these files as a whole. The following are the commands that you would use to type out the system startup files.

- a. `$ TYPE SYS$MANAGER:SYLOGICALS.COM`
 - b. `$ TYPE SYS$MANAGER:SYPAGSWPFILES.COM`
 - c. `$ TYPE SYS$MANAGER:SYCONFIG.COM`
 - d. `$ TYPE SYS$MANAGER:SYSTARTUP_VMS.COM`
2. (Optional) Create TERMINALS.COM in your own directory. This procedure should set the permanent characteristics of the terminals on your system according to the following information:
 - a. Assume that you have eight terminals.
 - b. Set up some fast terminals and some slow terminals.
 - c. Protect at least three terminals from allocation by processes other than those with a [001,004] UIC.
 - d. Four of the terminals are VT200 series terminals.
 - e. Three are VT300 series terminals.
 - f. One is an LA120 hardcopy terminal.
 - g. The LA120 is attached through a modem.
 - h. The other terminals are attached through direct lines.
 - i. Include the name of the owner and office number where the terminal is located in a comment for each terminal.

The following is a sample TERMINALS.COM file. This procedure meets the specifications of the problem, but it is not the only answer. If your procedure contains most of the same kinds of statements, it is probably also correct.

```
$!TERMINALS.COM
$!
$! This file sets up the permanent characteristics of the
$! terminals on this system. This procedure is typically
$! invoked from SYSTARTUP_VMS.COM.
$!
$!-----
$!
$ SET NOON
$!
$ SET TERM/PERM/NOMODEM-
/VT200/SPEED=300 TTA0:  !J.Smith E15
$ SET TERM/PERM/NOMODEM-
/VT200/SPEED=2400 TTA1:  !N.Hae E16
$ SET TERM/PERM/NOMODEM-
/VT200/SPEED=9600 TTA2:  !F.Chi E17
$ SET TERM/PERM/NOMODEM-
/VT200/SPEED=9600 TTA3:  !P.Jones E18
$ SET TERM/PERM/NOMODEM-
/VT300/SPEED=2400 TTA4:  !A.Steel E19
$ SET TERM/PERM/NOMODEM-
/VT300/SPEED=9600 TTA5:  !J.Howland F01
$ SET TERM/PERM/NOMODEM-
/VT300/SPEED=9600 TTA6:  !M.Carter F02
$ SET TERM/PERM/MODEM-
/AUTOBAUD/LA120 TTA7:  !D.Trevor Dial-up
$!
$! NOTE: For the LA120 terminal, the speed is
$! set automatically by /AUTOBAUD to 9600.
$!
$!Give [001,004] ownership of three terminals:
$!
$ SET SECURITY/PROT=(S,0:R,G,W)/DEVICE/OWNER_UIC=[001,004] TTA0:
```

\$ SET SECURITY/PROT=(S,O:R,G,W)/DEVICE/OWNER_UIC=[001,004] TTA5:

\$ SET SECURITY/PROT=(S,O:R,G,W)/DEVICE/OWNER_UIC=[001,004] TTA7:

\$!

\$! NOTE: All users can log in on these terminals, but users

\$! with a UIC of [001,004] can also allocate them.

3. (Optional) Create an alternate SYSTARTUP_VMS.COM in your own directory. Do not execute this procedure to verify your work.

The following is a sample SYSTARTUP_VMS.COM procedure. This procedure meets the specifications of the problem, but it is not the only answer. If your procedure contains most of the same kinds of statements, it is probably also correct.

```
$! SAMPLE_SYSTARTUP_VMS.COM
$!
$! This procedure sets up the system environment according
$! to the resources available and functions performed on
$! this system.
$!
$!-----
$!
$ MOUNT/SYSTEM DUA1: CLASS CLASS_DISK
$!
$ ASSIGN/SYSTEM SYS$SYSDEVICE:PROGRAMS.DIR PROGRAMS
$!
$! Start up PRINT and BATCH queues
$!
$ INITIALIZE/QUEUE/FLAG/START LPA0
$!
$ INITIALIZE/QUEUE/GENERIC=LPA0/START SYS$PRINT
$!
$ SET PRINTER LPA0:/LOWER
$ SET DEVICE LPA0:/SPOOLED
$!
$ INITIALIZE/QUEUE/BATCH/JOB=2/PRIORITY=3/START SYS$BATCH
$!
$ INITIALIZE/QUEUE/BATCH/JOB=1/PRIORITY=2 BIGJOB
$!
$ @SYS$MANAGER:TERMINALS.COM
$!
$ ASSIGN/SYSTEM "System number 239 - OpenVMS " SYS$ANNOUNCE
$ ASSIGN/SYSTEM "welcome to system 239" SYS$WELCOME
$!
$ RENAME OPERATOR.LOG; -1 OPERATOR.OLD
```

```
$ PRINT/DELETE OPERATOR.OLD
$!
$ SET LOGIN/INTERACTIVE=35
$!
$! NOTE: The SET LOGIN/INT command should be placed at the end
$!   of the procedure, so users cannot log in while the
$!   procedure is executing.
$!
$!
$ REPLY/BELL/ALL "System 239 is ready for use"
$!
$ LOGOUT/BRIEF
```

Customizing Login Command Procedures

Lab Exercise 5–2: Create a Captive Account

You need write access to the MFD of your disk and to the system authorization files to do this lab.

1. Create an account called PAYROLL. The group number assigned to workers in payroll is 322. Choose any member number from 60-377 (octal) that is not in use. Also create a directory and a disk quota entry for the account.
2. Log in to the new account and create a small file in its directory to verify your work.
3. Create a captive login command procedure for PAYROLL. Include commands to:
 - a. Log out if Ctrl/Y is typed
 - b. Display process information
 - c. Display information about system processes
 - d. Log out
4. Log in to PAYROLL and type Ctrl/Y while the commands are executing. What happens?
5. Log in to PAYROLL again, specifying /NOCOMMAND on the user name. What happens?
6. Log in to PAYROLL again, specifying /NOCOMMAND on the user name. What happens?
7. Remove the PAYROLL user from the system.

Lab Solution 5–2: Create a Captive Account

1. Create an account called PAYROLL. The group number assigned to workers in payroll is 322. Choose any member number from 60-377 (octal) that is not in use. Also create a directory and a disk quota entry for the account.

```
$ SET PROCESS/PRIVILEGE=SYSPRV
```

```
$ SET DEFAULT SYS$SYSTEM
```

```
$ RUN AUTHORIZE
```

```
UAF> SHOW/BRIEF [322,*]
```

(Output shows 300 has not been used as a member number yet)

```
UAF> ADD PAYROLL/PASSWORD=TESTING/UIC=[322,300]-
```

```
_UAF> /DEV=DISK$TESTSPACE/DIR=[PAYROLL]/FLAG=NODISUSER
```

```
UAF> EXIT
```

```
$ SET PROC/PRIV=OPER
```

```
$ RUN SYSMAN
```

```
SYSMAN> DISKQUOTA ADD
```

```
[322,300]/DEVICE=DISK$TESTSPACE/PERM=500/OVER=500
```

```
SYSMAN> EXIT
```

```
$ CREATE/DIRECTORY/OWNER=[322,300] DISK$TESTSPACE:[PAYROLL]
```

```
$
```

2. Log in to the new account and create a small file in its directory to verify your work.

```
$ SET HOST 0
```

```
Welcome to VAX/VMS F7.0
```

```
Username: PAYROLL
```

```
Password:
```

```
Welcome to OpenVMS VAX version F7.0 on node TIDY
```

```
Your password has expired; you must set a new password to log in
```

```
New password:
```

```
Verification:
```

```
$ CREATE TEST.FILE
```

```
.  
.   
. 
```

```
$ DIR TEST.FILE
```

```
Directory DISK$TESTSPACE: [PAYROLL]
```

```
TEST.FILE;1
```

```
Total of 1 file.
```

```
$ LOGOUT
```

```
PAYROLL   logged out at 20-NOV-1995 10:56:45.47
```

Check the owner UIC of the directory if you receive protection errors. Check the contents of the QUOTA.SYS file on your disk if you receive quota errors.

3. Create a captive login command procedure for PAYROLL. Include commands to:

- a. Log out if Ctrl/Y is typed
- b. Display process information
- c. Display information about system processes
- d. Log out

```
$ CREATE SYS$LOGIN:LOOPLOGIN.COM
```

```
$ ON CONTROL_Y THEN LOGOUT
```

```
$ SHOW PROCESS/ALL
```

```
$ SHOW SYSTEM
```

```
$ LOGOUT
```

```
[Ctrl/Z]
```

```
$ SET SECURITY/PROTECTION=W:RE SYS$LOGIN:LOOPLOGIN.COM
```

```
$ RUN AUTHORIZE
```

```
UAF> MODIFY PAYROLL/LGICOM=SYS$LOGIN:LOOPLOGIN.COM
```

```
UAF> EXIT
```

4. Log in to PAYROLL and type Ctrl/Y while the commands are executing. What happens?

```
$ SET HOST 0
```

```
Welcome to VAX/VMS F7.0
```

```
Username: PAYROLL
```

```
Password:
```

```
Welcome to OpenVMS VAX version F7.0 on node TIDY
```

```
  Last interactive login on Monday, 20-NOV-1995 10:22
```

```
20-NOV-1995 10:59:47.33  User: PAYROLL      Process ID: 25A00D68
```

```
                Node: TIDY      Process name: "PAYROLL"
```

```
Terminal:      RTA3: (TIDY::MGR1)
```

```
User Identifier: [PAYROLL]
```

```
Base priority:  4
```

```
Default file spec: DISK$TESTSPACE:[PAYROLL]
```

```
Devices allocated: TIDY$RTA3:
```

```
[ Interrupt ]
```

```
  PAYROLL  logged out at 20-NOV-1995 11:00:15.24
```

```
%REM-S-END, control returned to node _TIDY::
```

When Ctrl/Y is typed, the process is logged out.

5. Log in to PAYROLL again, specifying /NOCOMMAND on the user name.
What happens?

```
$ SET HOST 0
```

```
Welcome to VAX/VMS F7.0
```

```
Username: PAYROLL/NOCOMMAND
```

```
Password:
```

```
Welcome to OpenVMS VAX version F7.0 on node TIDY
```

```
  Last interactive login on Monday, 20-NOV-1995 10:59
```

```
$
```

The login command procedure does not execute, and the DCL prompt is displayed.

6. Modify the UAF record for PAYROLL to include the CAPTIVE flag.
\$ SET DEFAULT SYS\$SYSTEM
\$ RUN AUTHORIZE
UAF> MODIFY PAYROLL/FLAGS=CAPTIVE
UAF> EXIT
\$
7. Log in to PAYROLL again, specifying /NOCOMMAND on the user name.
What happens?
\$ SET HOST 0

Welcome to VAX/VMS F7.0

Username: PAYROLL/NOCOMMAND

Password:

Qualifier not allowed in captive accounts

Username:

8. Remove the PAYROLL user from the system.

To remove the account completely, you must:

- ❶ Delete all subdirectories and files in [PAYROLL]
 - ❷ Remove PAYROLL.DIR
 - ❸ Remove the PAYROLL entry from the quota file
 - ❹ Remove the UAF record from the SYSUAF.DAT file
- ```
❶ $ DELETE DISK$TESTSPACE:[PAYROLL...]*.*;*
$ SET PROT=W:REWD DISK$TESTSPACE:[000000]PAYROLL.DIR
❷ $ DELETE DISK$TESTSPACE:[000000]PAYROLL.DIR;
$ SET PROCESS/PRIVILEGE=OPER
$ RUN SYS$SYSTEM:SYSMAN
❸ SYSMAN> DISKQUOTA DELETE
[322,300]/DEVICE=DISK$TESTSPACE
SYSMAN> EXIT
$ SET PROCESS/PRIVILEGE=SYSPRV
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
❹ UAF> REMOVE PAYROLL
UAF> EXIT
$
```

## Maintaining the DECnet for OpenVMS Remote Node Database

### Lab Exercise 5–3: Using NCP

At an interactive terminal, perform the following tasks:

1. Enter the DCL command that starts an NCP session.
2. At the NCP> prompt, enter the command SHOW EXECUTOR.

Observe that this command shows you:

- The name and numeric address of the local node
  - Whether the DECnet for OpenVMS software is running (on)
  - What version of the DECnet for OpenVMS software is running
3. Exit the session and return to the DCL prompt.

## Lab Solution 5–3: Using NCP

At an interactive terminal, perform the following tasks:

1. Enter the DCL command that starts an NCP session.  
`$ RUN SYS$SYSTEM:NCP`
2. At the NCP> prompt, enter the command SHOW EXECUTOR.

Observe that this command shows you:

- The name and numeric address of the local node
- Whether the DECnet for OpenVMS software is running (on)
- What version of the DECnet for OpenVMS software is running

```
NCP>SHOW EXECUTOR
```

```
Node Volatile Summary as of 20-NOV-1995 11:34:32
```

```
Executor node = 2.7 (BOSTON)
```

```
State = on
```

```
Identification = DECnet for OpenVMS VAX V6.1
```

3. Exit the session and return to the DCL prompt.

```
NCP>EXIT or NCP> [ctrl/Z]
```

## Introduction to Lab Exercise 5–4

You need to find an unused node name and two unused addresses for these exercises. Do one of the following:

- Ask your or a system manager or network administrator at your site, to provide them.
- Run NCP and enter the SHOW KNOWN NODES command. Select a node name and two addresses that do not appear on the list.
- If the output from SHOW KNOWN NODES is very long, select a node name at random and then enter a SHOW NODE command to find out whether it is in the remote node database; keep trying until you find a node name that is not in the database. Choose two addresses similarly.

### Lab Exercise 5–4: Adding Remote Nodes

For this exercise, choose one node name and one address. At an interactive terminal, perform the following tasks:

1. At the DCL prompt, try to use the SET HOST command to connect to the node whose name you chose. What happens?
2. Run NCP and enter commands to find out whether that node is registered in both the permanent and volatile remote node databases.
3. Define the node in the permanent database.
4. Exit from NCP and try step 1 again. What happens and why?
5. Run NCP again and enter a command to copy the node name and address from the permanent to the volatile database.
6. Exit from NCP and try step 1 again. What happens and why?

## Lab Solution 5-4: Adding Remote Nodes

These solutions demonstrate the use of node name SHIRT and address 5.123. Substitute the name and address that you are using.

1. At the DCL prompt, try to use the SET HOST command to connect to the node whose name you chose. What happens?

You see this message:

```
%SYSTEM-F-NOSUCHNODE, remote node is unknown
```

2. Run NCP and enter commands to find out whether that node is registered in both the permanent and volatile remote node databases.

If the node is in neither database, you see this output:

```
$ RUN SYS$SYSTEM:NCP
```

```
NCP>LIST NODE SHIRT
```

```
Node Permanent Summary as of 20-NOV-1995 11:37:45
```

```
%NCP-W-UNRCMP, Unrecognized component , Node
```

```
NCP>SHOW NODE SHIRT
```

```
Node Volatile Summary as of 20-NOV-1995 11:38:53
```

```
%NCP-W-UNRCMP, Unrecognized component , Node
```

3. Define the node in the permanent database.  
NCP>DEFINE NODE SHIRT ADDRESS 5.123  
(or)  
NCP>DEFINE NODE 5.123 NAME SHIRT
4. Exit from NCP and try step 1 again. What happens and why?

```
NCP>[Ctrl/Z]
```

```
$ SET HOST SHIRT
```

```
%SYSTEM-F-NOSUCHNODE, remote node is unknown
```

The node is still not defined in the volatile database, and the volatile database is where the DECnet software obtains its data during operation.

5. Run NCP again and enter a command to copy the node name and address from the permanent to the volatile database.

```
$ RUN SYS$SYSTEM:NCP
```

```
NCP>SET NODE SHIRT ALL
```

(or)

```
$ RUN SYS$SYSTEM:NCP
```

```
NCP>SET NODE 5.123 ALL
```

(or)

```
$ RUN SYS$SYSTEM:NCP
```

```
NCP>SET NODE SHIRT ADDRESS 5.123
```

(or)

```
$ RUN SYS$SYSTEM:NCP
```

```
NCP>SET NODE 5.123 NAME SHIRT
```

6. Exit from NCP and try step 1 again. What happens and why?

If the node exists and is running, you see the login prompt. If it does not exist or is not running, you see the following message:

```
%SYSTEM-F-UNREACHABLE, remote node is not currently
reachable
```

## **Lab Exercise 5–5: Changing a Remote Node Entry**

In the exercise on adding a remote node, you added a node entry to the remote node database.

Now change its address to the other address of the two that you chose or were given. Change it in both the permanent and the volatile database.

## Lab Solution 5–5: Changing a Remote Node Entry

This solution assumes that you are changing the address of node SHIRT to 5.124. Substitute the name and address that you are using.

```
$ RUN SYS$SYSTEM:NCP
```

```
NCP>DEFINE NODE SHIRT ADDRESS 5.124
```

```
NCP>SET NODE SHIRT ALL
```

(or)

```
$ RUN SYS$SYSTEM:NCP
```

```
NCP>DEFINE NODE SHIRT ADDRESS 5.124
```

```
NCP>SET NODE SHIRT ADDRESS 5.124
```

## **Lab Exercise 5–6: Removing Remote Nodes**

In the exercise on adding a remote node, you added a node entry to the remote node database; in the exercise on changing a remote node entry, you changed its address.

Now use NCP to remove that node entry from both databases.

## Lab Solution 5–6: Removing Remote Nodes

These solutions demonstrate the use of node name SHIRT and address 5.124. Substitute the name and address that you are using.

```
NCP>CLEAR NODE SHIRT ALL
```

```
NCP>PURGE NODE SHIRT ALL
```

(or)

```
NCP>CLEAR NODE 5.124 ALL
```

```
NCP>PURGE NODE 5.124 ALL
```

---

# ***Installing Layered Products***

## ***Module 6 Labs***

### **Managing Product Licenses**

#### **Lab Exercise 6–1: License Management Facility**

1. Enter the LMF command to display a list of licenses registered in the license database for:
  - a. All licenses registered
  - b. A specific layered product
2. Enter the LMF command that displays active licenses on the current node for:
  - a. All active licenses
  - b. A specific layered product

## Lab Solution 6–1: License Management Facility

1. Enter the LMF command to display a list of licenses registered in the license database for:
  - a. All licenses registered  
\$ LICENSE LIST
  - b. A specific layered product  
\$ LICENSE LIST FORTRAN

Press Ctrl/Z to exit, use arrow keys to scroll.  
License Management Facility

License Database File:  
SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB;1  
Created on: 9-MAY-1999  
Created by user: SYSTEM  
LMF Version: V1.0

-----  
FORTRAN DEC  
FORTRAN DEC  
[End Of List]

The purpose of displaying the list of all registered licenses is to begin to familiarize yourself with the contents of the license database. This command allows you to gain information on the entire database or just on specific products.

2. Enter the LMF command that displays active licenses on the current node for:

a. All active licenses

\$ SHOW LICENSE

b. A specific layered product

\$ SHOW LICENSE FORTRAN

Active licenses on node SUPER:

```
----- Product ID ----- ---- Rating ----- -- Version --
Product Producer Units Avail Activ Version Release
Termination
FORTRAN DEC 0 F 0 0.0 23-AUG-1999 (none)
```

The SHOW LICENSE command allows you to find out what products have already been loaded on the current node. This command will give you the entire list of active products, or you can request information on a specific product.

## **Lab Exercise 6–2: License Management Utility (LICENSE)**

1. Create a license database in your own user directory. Enter the SHOW DEFAULT command to find out the exact device and directory name that you need to specify when creating the database.

2. Using the information from the following PAK, register and list the sample product ED\_SERV\_FORTRAN. You will need to specify the /DATABASE qualifier to the REGISTER command to ensure that you register it in your private database.

PAK ID:

Issuer: DEC

Authorization Number: SQM003058

PRODUCT ID:

Product Name: ED\_SERV\_FORTRAN

Producer: DEC

NUMBER OF UNITS:

Number of units: 500

KEY LEVEL:

Version: 1.0

Product Release Date:

KEY TERMINATION DATE:

Key Termination Date:

RATING:

Availability Table Code:

Activity Table Code:

MISCELLANEOUS:

Key Options:

Product Token:

Hardware-Id:

Checksum: 1-PAOI-MJNP-NCJL-DEKP

\*\*\*\*\*

## Lab Solution 6–2: License Management Utility (LICENSE)

1. Create a license database in your own user directory. Enter the SHOW DEFAULT command to find out the exact device and directory name that you need to specify when creating the database.

The following is the command that you would use to create a new license database called PRACTICE.LDB in the directory [MGR1]. Your database location would be different than the one used in this solution. A DIRECTORY command entered afterwards verifies that the file was created in the requested directory.

```
$
```

```
$ LICENSE CREATE /DATABASE=1DUA1:[MGR1]PRACTICE.LDB
```

```
$
```

```
$ DIRECTORY
```

```
Directory 1DUA1:[MGR1]
```

```
PRACTICE.LDB;1
```

```
Total of 1 file.
```

2. Using the information from the following PAK, register and list the sample product ED\_SERV\_FORTRAN. You will need to specify the /DATABASE qualifier to the REGISTER command to ensure you register it to your private database.

The following are the LICENSE commands that you would enter to register and then list the layered product ED\_SERV\_FORTRAN.

```
$ LICENSE REGISTER "ED_SERV_FORTRAN" -
_$ /ISSUER="DEC" -
_$ /AUTHORIZATION=SQM003058 -
_$ /PRODUCER=DEC -
_$ /UNITS=500 -
_$ /VERSION=1.0 -
_$ /CHECKSUM=1-PAOI-MJNP-NCJL-DEKP -
_$ /DATABASE=1DUA1:[MGR1]PRACTICE.LDB
$
$ LICENSE LIST ED_SERV_FORTRAN
/DATABASE=1DUA1:[MGR1]PRACTICE.LDB
$
```

Note that the use of the SHOW LICENSE command here will default to the actual system database, not your practice database. There is no /DATABASE qualifier to the SHOW LICENSE command.



---

# ***Monitoring System Resources***

## ***Module 7 Labs***

## Using the Monitor Utility

### Lab Exercise 7-1

1. Using a video terminal, run the Monitor utility to display the MONITOR> prompt. At the prompt, enter a command to the utility to display the PAGE class with an update interval of 10 seconds. Allow the utility to run for 5 to 10 minutes. Watch the display and notice how (if you are on an active system) the display changes with time.
2. Return to the MONITOR> prompt and display the SYSTEM class.
3. Return to the MONITOR> prompt and display the I/O class.
4. Use MONITOR to create a file called SUMMARY.DAT that contains summary information about the PROCESSES class. Allow the utility to write several screens of information to the file before you enter the Ctrl/Z key sequence to return to the DCL prompt. (Each time the values in the display change, the utility writes a copy of the display to the file.) Display the file on your terminal screen after you exit from the utility.
5. Issue the command to monitor the cluster. This may take a while to build a display because a network process must be created on each node.
6. Issue the DCL command to display the cluster information once, then return to the DCL prompt. Select one other node name to be used for the next step.
7. Monitor your system plus the other selected node.
8. Have a lab partner monitor your node, then pause by returning to the MONITOR prompt. While your partner's MONITOR display is alternately active and inactive, issue the DCL command to find out the priorities of this created process. Note the process name/priority as opposed to the PID.
9. Return to the MONITOR> prompt and display any other classes that you want. Use the HELP facility in the Monitor utility to list the names of the classes and possible qualifiers.
10. Use the SHOW USERS command to obtain a list of the interactive users on the system. Use the SHOW SYSTEM command to obtain a list of the processes on the system. Compare the names of the interactive users with the list of process names. Note which processes appear in one but not the other.
11. Use the SHOW PROCESS/CONTINUOUS command to examine your own process. What image does the program indicate you are running? Try the V key (get help).
12. Which DCL command gives you information on physical memory and usage?

13. Which DCL command gives you information on the page files and swap files and their usage?
14. Use the SHOW MEMORY command to observe current memory usage.
15. Use the SHOW PROCESS/CONTINUOUS command to monitor:
  - a. another student's process, by PID (SHOW PROCESS/CONTINUOUS/
  - b. another student's process, by name (SHOW PROCESS/CONTINUOUS <name>) (Use \$ SHOW SYSTEM to look up the pid and name for use in b and c.)

## Lab Solution 7–1

1. Using a video terminal, run the Monitor utility to display the MONITOR> prompt. At the prompt, enter a command to the utility to display the PAGE class with an update interval of 10 seconds. Allow the utility to run for 5 to 10 minutes. Watch the display and notice how (if you are on an active system) the display changes with time.  

```
$ MONITOR
MONITOR> MONITOR PAGE
```
2. Return to the MONITOR> prompt and display the SYSTEM class.  

```
[Ctrl/C]
MONITOR> MONITOR SYSTEM
```
3. Return to the MONITOR> prompt and display the I/O class.  

```
[Ctrl/C]
MONITOR> MONITOR IO
```
4. Use MONITOR to create a file called SUMMARY.DAT that contains summary information about the PROCESSES class. Allow the utility to write several screens of information to the file before you enter the Ctrl/Z key sequence to return to the DCL prompt. (Each time the values in the display change, the utility writes a copy of the display to the file.) Display the file on your terminal screen after you exit from the utility.  

```
[Ctrl/C]
MONITOR> MONITOR PROCESS/SUMMARY=SUMMARY.DAT
[Ctrl/Z]
$ TYPE SUMMARY.DAT
```
5. Issue the command to monitor the cluster. This may take a while to build a display because a network process must be created on each node.  

```
MONITOR> MONITOR CLUSTER
```
6. Issue the DCL command to display the cluster information once, then return to the DCL prompt. Select one other node name to be used for the next step.  

```
[Ctrl/Z]
$ SHOW CLUSTER
```
7. Monitor your system plus the other selected node.  

```
$ MONITOR CLUSTER/NODE=(yours,selected_other)
```

8. Have a lab partner monitor your node, then pause by returning to the MONITOR prompt. While your partner's MONITOR display is alternately active and inactive, issue the DCL command to find out the priorities of this created process. Note the process name/priority as opposed to the PID.

[Ctrl/Z]

...as the MONITOR CLUSTER process begins

\$ SHOW SYSTEM/NETWORK

.  
.  
.

...after the MONITOR CLUSTER process starts collecting data

\$ SHOW SYSTEM/NETWORK

.  
.  
.

...after the MONITOR CLUSTER process is interrupted by a [Ctrl/Z]

\$ SHOW SYSTEM/NETWORK

.  
.  
.

...after the Monitor utility is terminated

\$ SHOW SYSTEM/NETWORK

.  
.  
.

9. Return to the MONITOR> prompt and display any other classes that you want. Use the HELP facility in the Monitor utility to list the names of the classes and possible qualifiers.

Enter the Ctrl/C key sequence to cancel any display and return to the MONITOR> prompt.

```
$ MONITOR
```

```
MONITOR> HELP MONITOR
```

10. Use the SHOW USERS command to obtain a list of the interactive users on the system. Use the SHOW SYSTEM command to obtain a list of the processes on the system. Compare the names of the interactive users with the list of process names. Note which processes appear in one but not the other.

There should be more processes listed by SHOW SYSTEM than by SHOW USERS.

SHOW SYSTEM lists the system processes, interactive processes, processes from batch queues, and subprocesses. SHOW USERS lists interactive users attached to terminals.

11. Use the SHOW PROCESS/CONTINUOUS command to examine your own process. What image does the program indicate you are running? Try the V key (get help).

SHOW PROCESS/CONTINUOUS shows that you are executing  
SYS\$SYSROOT:[SYSEXE]SHOW.EXE.

12. Which DCL command gives you information on physical memory and usage?

```
S SHOW MEMORY/PHYSICAL
```

13. Which DCL command gives you information on the page files and swap files and their usage?

```
S SHOW MEMORY/FILES/FULL
```

14. No solution needed.

15. No solution needed.

## Using the SHOW CLUSTER Utility

### Lab Exercise 7-2

SHOW CLUSTER covers approximately 100 fields of data organized in terms of fields and classes.

The classes are:

CLUSTER

SYSTEMS

MEMBERS

CIRCUITS

CONNECTIONS

COUNTERS

CREDITS

LOCAL\_PORT

ERROR

The fields vary for each class. For example, the CLUSTER class has the following fields:

- CL\_EXPECTED\_VOTES
- CL\_MEMBERS
- CL\_QDVOTES
- CL\_QUORUM
- CL\_VOTES
- FORMED
- LAST\_TRANSITION
- MEMSEQ
- QD\_NAME
- QF\_VOTE

The SHOW CLUSTER utility has the following basic commands:

ADD

REMOVE

SET

INIT

HELP

EXIT

SAVE

WRITE

@file-spec

Execute SHOW CLUSTER/CONTINUOUS and enter a series of commands that change the display.

## Lab Solution 7–2

Issue the following commands:

|                                 |                                                                              |
|---------------------------------|------------------------------------------------------------------------------|
| 1. SHOW CLUSTER/CONTINUOUS      |                                                                              |
| 2. ADD DIR_WT                   | (This adds one field.)                                                       |
| 3. REMOVE MEMBER                | (This removes the entire class. )                                            |
| 4. ADD VOTES,EXPECTED_VOTES     | (This adds two fields. )                                                     |
| 5. ADD CIRCUITS                 | (This adds one class. Each class will have its own default fields. )         |
| 6. REMOVE RPORT                 | (This removes a field. )                                                     |
| 7. REMOVE CIRCUIT               | (This removes a class. )                                                     |
| 8. ADD TRANSITION_TIME          | (This adds a field.)                                                         |
| 9. SET VOTES/WIDTH=2            | (This will adjust the width of the column. )                                 |
| 10. SAVE MYFILE.SCD             | (This will write back to disk the format of the display. )                   |
| 11. INIT                        | (This will return you to the default display without doing ADD or REMOVE. )  |
| 12. WRITE MYFILE.DISPLAY        | (This takes a snapshot of the terminal display and writes it back to disk. ) |
| 13. CONTROL/Z or EXIT)          |                                                                              |
| 14. \$ TYPE MYFILE.DISPLAY/PAGE | (Here is the information that was on the screen earlier.)                    |
| 15. \$ SHOW CLUSTER/CONTINUOUS  |                                                                              |
| 16. @MYFILE.SCD                 | (This file will re-create the display without your using ADD commands.)      |
| 17. CONTROL/Z or EXIT           |                                                                              |

---

# ***Maintaining System Security***

## ***Module 8 Labs***

### **General System Security**

#### **Written Exercise 8–1: Physical Security**

List two ways a system manager can provide physical security for a computer system.

.....

.....

## Written Solution 8–1: Physical Security

List two ways a system manager can provide physical security for a computer system.

Examples:

- Keep the console terminal in a locked or restricted area.
- Do not post dialup line phone numbers in public locations.
- Keep all media storage in a locked or restricted area.

## Written Exercise 8–2: Software Security

List five ways a system manager can maintain software security.

.....

.....

.....

.....

.....

## Written Solution 8–2: Software Security

List five ways a system manager can maintain software security.

Examples:

- Keep SYSUAF.DAT, RIGHTSLIST.DAT, NETPROXY.DAT, and NET\$PROXY.DAT (VAX systems) up to date.
- Use flags and hourly restrictions with AUTHORIZE.
- Insist on nontrivial passwords.
- Do not publicize dialup numbers.
- Do not permit WORLD access to SYSUAF.DAT or RIGHTSLIST.DAT.
- Encourage use of file protection codes and ACLs.
- Restrict user activity with captive accounts.
- Restrict user privileges.
- Create accounting reports to check system usage.
- Label disks and tapes in a systematic manner.
- Require secondary passwords.
- Implement erase-on-delete and erase-on-allocate for files.
- Use login security.
- Use break-in detection.
- Use security auditing.
- Implement security alarms on files.
- Protect subsystems.

## Restricting User Activities

### Written Exercise 8–3: Process Parameters

The following example displays the characteristics of a process that has several privileges. Using the information displayed in the example, determine the value of each of the parameters. (Turn the page for this exercise.)

| Answers | Parameters                                 |
|---------|--------------------------------------------|
|         | Account name                               |
|         | Default device and directory specification |
|         | Interactive terminal specification         |
|         | Process identification number              |
|         | Process name                               |
|         | User identification code                   |
|         | User name                                  |
|         | Priority                                   |
|         | CPU limit                                  |
|         | Open file quota                            |
|         | Working set limit                          |
|         | Working set quota                          |
|         | Privileges (list them)                     |

\$ SHOW PROCESS/ALL

19-NOV-1999 17:05:08.99 User: GABRIEL Process ID: 20200242

Node: TRMPET Process name: Bugler

Terminal: LTA28: (ZK1123/LC-4-2)

User Identifier: [GABRIEL]

Base priority: 4

Default file spec: DISK\$COMPACT:[GABRIEL]

Devices allocated: TIDY\$LTA28:

Process Quotas:

Account name: MUSIC

CPU limit: Infinite Direct I/O limit: 100

Buffered I/O byte count quota: 50000 Buffered I/O limit: 100

Timer queue entry quota: 80 Open file quota: 50

Paging file quota: 49978 Subprocess quota: 8

Default page fault cluster: 64 AST quota: 100

Enqueue quota: 600 Shared file limit: 0

Max detached processes: 0 Max active jobs: 0

Accounting information:

```
Buffered I/O count: 8241 Peak working set size: 10268
Direct I/O count: 763 Peak virtual size: 24376
Page faults: 37681 Mounted volumes: 1
Images activated: 35
Elapsed CPU time: 0 00:13:37.81
Connect time: 0 01:25:13.34
```

Authorized privileges:

NETMBX SETPRV TMPMBX

Process privileges:

```
CMKRNL may change mode to kernel
NETMBX may create network device
OPER operator privilege
TMPMBX may create temporary mailbox
```

Process rights identifiers:

INTERACTIVE

LOCAL

SYS\$NODE\_TIDY

Process Dynamic Memory Area

|                       |       |                            |      |
|-----------------------|-------|----------------------------|------|
| Current Size (bytes)  | 25600 | Current Total Size (pages) | 50   |
| Free Space (bytes)    | 21744 | Space in Use (bytes)       | 3856 |
| Size of Largest Block | 21712 | Size of Smallest Block     | 8    |
| Number of Free Blocks | 4     | Free Blocks LEQU 32 Bytes  | 1    |

Processes in this tree:

Satchmo

Bugler (\*)

\$

\$ SHOW WORKING\_SET

Working Set /Limit= 1024 /Quota= 4096 /Extent= 8192

Adjustment enabled Authorized Quota= 4096 Authorized Extent= 8192

\$

## Written Solution 8–3: Process Parameters

| Answers                      | Parameters                                 |
|------------------------------|--------------------------------------------|
| MUSIC                        | Account name                               |
| DISK\$COMPACT:[GABRIEL]      | Default device and directory specification |
| LTA28:                       | Interactive terminal specification         |
| 20200242                     | Process identification number              |
| Bugler                       | Process name                               |
| [ULTIMATE,GABRIEL]           | User identification code                   |
| GABRIEL                      | User name                                  |
| 4                            | Priority                                   |
| Infinite                     | CPU limit                                  |
| 50                           | Open file quota                            |
| 1024                         | Working set limit                          |
| 4096                         | Working set quota                          |
| CMRKNL, TMPMBX, OPER, NETMBX | Privileges (list them)                     |

## Lab Exercise 8–4: Setting Login Flag Parameters

You must have write access to the system authorization file and SECURITY privilege to do this lab.

1. Create a UAF record, using a vacant UIC. Set the expiration date on the new UAF record to be 5 minutes after the current time.
2. Wait at least 5 minutes and attempt to log in to the new account. What happens?
3. Log in to your own account and delete the extra UAF record.
4. Change the minimum length of the password to 10 on your own account, set a flag to require password generation, exit from the utility, log out, and log in.
5. Change your password with a DCL command that uses one of the choices listed by the generator (or generate a new list).

## Lab Solution 8–4: Setting Login Flag Parameters

1. Create a UAF record, using a vacant UIC. Set the expiration date on the new UAF record to be 5 minutes after the current time.

```
$ SET DEFAULT SYS$SYSTEM
```

```
$ RUN AUTHORIZE
```

```
UAF> ADD
```

```
TEST_ACCT/PASS=XYZZY/EXPIRATION=15:50/uic=[200,5300]/FLAGS=NOD
ISUSER
```

```
%UAF-I-ADDMSG, user record successfully added
```

```
%UAF-I-RDBADDMSGU, identifier TEST_ACCT value [000200,005300]
added to rights data base
```

```
UAF>
```

2. Wait at least 5 minutes and attempt to log in to the new account. What happens?

```
Welcome to OpenVMS 7.2
```

```
Username: TEST_ACCT
```

```
Password:
```

```
Your account has expired; contact your system manager
```

```
You will not be able to log in to the new account because it has expired.
```

3. Log in to your own account and delete the extra UAF record.  
\$ RUN AUTHORIZE  
UAF> REMOVE TEST\_ACCT
4. Change the minimum length of the password to 10 on your own account, set a flag to require password generation, exit from the utility, log out and log in.  
Use the /FLAGS=GENPWD qualifier to the MODIFY command in the AUTHORIZE utility to do this.  
\$ SET DEFAULT SYS\$SYSTEM  
\$ RUN AUTHORIZE  
UAF> MODIFY MGR1/PWDMINIMUM=10/FLAGS=GENPWD  
UAF> EXIT

5. Change your password with a DCL command that uses one of the choices listed by the generator (or generate a new list).

```
$ SET PASSWORD
```

```
Old password:
```

```
cyetadbyac cye-tad-byac
faillavplan fail-lav-plan
artpabruan art-pa-bru-an
aswiasdyay a-swi-as-dyay
uzangyacboa u-zang-yac-boa
```

```
Choose a password from this list, or press RETURN to get a new
list
```

```
New password:
```

```
Verification:
```

```
$
```

When you enter the SET PASSWORD command, it automatically generates a list of passwords from which you can choose.

## Using Passwords to Maintain System Security

### Lab Exercise 8–5: Filtered Passwords

1. Change your password and then try to change it again to the original password.
2. Try to change your password to any English word.

## Lab Solution 8–5: Filtered Passwords

1. Change your password and then try to change it again to the original password.  
\$ SET PASSWORD  
Old password:  
New password:  
Verification:  
%SYSTEM-F-PWDINHIS, password found in history list; please choose  
a new password

When you set your password it adds that password to your password history file. The next time you set your password it checks the password you type against the password history file; if it finds the password there it displays the error message.

2. Try to change your password to any English word.  
\$ SET PASSWORD  
Old password:  
New password:  
%SYSTEM-F-PWDINDIC, password found in system dictionary; please  
choose another  
string

When you type a new password, the first thing the system does (even before it prompts you to type the new password verification) is to check the system password dictionary. If it finds the password you typed in the dictionary, you will receive an error message.