EY-1168E-SG-0001

Miguel DURAN

5-70-65-93

VAX/VMS System Management

Student Guide

Prepared by Educational Services
of
Digital Equipment Corporation

1st Edition, September 1982

Printed in U.S.A.

10/83-15

BACKUP 132
Major Steps in Using a Disk or Tape. 135
PROTECTING A VOLUME 137
THE MOUNT COMMAND 139
MOUNTING A DISK WITH AN UNKNOWN LABEL 140
Public Volumes 141
MAINTAINING PUBLIC VOLUMES. 142
Using the BACKUP Utility to Maintain
Public Volumes 142
BACKING UP THE SYSTEM DISK. 144
THE VERIFY UTILITY. 147
Modes of Operation. 147
MULTIVOLUME SETS ON TAPE. 149
MULTIVOLUME SETS ON DISK. 152
OTHER FILE TRANSFER UTILITIES 154
UNDERSTANDING THE DISK FILE STRUCTURE ON VAX/VMS. . . 156
ODS-2 Reserved Files. 157
ODS-2 Home Block. 158
Files on a FILES-11 Disk Volume 159
ODS-2 File Headers. 161
File ID 161
Choosing a Cluster Size for Disk Volumes. 164
ODS-2 Directories 165
Directory File Structure. 166
Comparing the ODS-1 and ODS-2 Structures. 167
Understanding the VAX/VMS Implementation
of the ANSI Level 3 Tape File Structure 168
Magnetic Tape Volume Label. 169
Tape File Header Labels 170
Tape File Trailer Labels. 171

5 MANAGING VMS

INTRODUCTION 175
OBJECTIVES 176
RESOURCES. 176
OVERVIEW 177
CUSTOMIZING THE SYSTEM START-UP FILES. 180
Assigning System Logical Names 182
Mounting Site-Specific Volumes 184
Starting Up System Processes 185
Installing Known Images. 185
Connecting All Standard Devices. 190
Enabling RMS File Sharing. 190
Recording Device Characteristics 192
Announcing That the System Is Up 194
Allowing Users to Log In 194

CUSTOMIZING THE DCL ENVIRONMENT. 195
Adding DCL Commands. 196
Adding HELP Text 197
Adding Error Messages. 198
SECURITY OF SYSTEM FILES 199
Overview 199
System Directories 200
File and Directory Protection. 203

6 SYSTEM START-UP AND SHUTDOWN

INTRODUCTION 207
OBJECTIVES 208
RESOURCES. 208
STARTING UP A VAX/VMS SYSTEM 209
Phases of Start-Up 209
VAX Console Subsystem. 211
Front Panel Switches 213
Console Processor and Console Device 217
Default System Device. 219
Automatic Restart. 226
Alternative System Devices 228
Stand-Alone Utilities and Diagnostics. 234
CONSOLE COMMANDS 235
Issuing Console Commands at System Start-Up. . . 236
Issuing Console Commands While VMS Is Running. . 237
Customizing Start-Up and Automatic Restart . . . 240
SYSTEM PARAMETERS. 245
Using SYSBOOT to Modify System Parameters. . . . 246
SPECIFYING THE SYSTEM CONFIGURATION. 250
VAX-11/782 MULTIPROCESSING SYSTEM. 251
Introduction 251
Starting Up the VAX-11/782 252
Controlling Multiprocessing. 253
SHUTDOWN 254

7 SYSTEM INSTALLATION AND UPDATE

INTRODUCTION 263
OBJECTIVES 264
RESOURCES. 264
INSTALLING OR UPGRADING VMS. 265
Overview 265
Building a System Volume on a VAX-11/780
or VAX-11/750. 266
Customizing the System Volume and Console Volume
on a VAX-11/780 or VAX-11/750. 270

UPGRADING VMS. 273
Additional Steps After Installation or Update. . 273
INSTALLING VMS ON A VAX-11/730 274
Systems With an R80 System Disk. 274
Systems With an RL02 System Disk 274
INSTALLING VMS ON A VAX-11/782
MULTIPROCESSING SYSTEM 275
ACTIVATING SINGLE-NODE DECnet ON VMS 276
RUNNING UETP AFTER INSTALLING VMS. 278
INSTALLING OPTIONAL SOFTWARE ON VMS
AND UPDATING VMS 279

8 SYSTEM PROBLEMS

INTRODUCTION 283
OBJECTIVES 284
RESOURCES. 284
SYSTEM DUMPS 285
System Dump File 285
Analyzing the System Dump File 286
Backing Up the System Dump File. 290
DEVICE ERRORS. 291
Error Logging. 291
Hardware Diagnostics 296
On-Line Diagnostics. 297
Remote Diagnosis 302
UETP 303
SOFTWARE PERFORMANCE REPORTS 304
Reporting Software Problems. 304

9 PERFORMANCE

INTRODUCTION 307
OBJECTIVES 308
RESOURCES. 308
OVERVIEW 309
MONITORING SYSTEM ACTIVITY 310
Overview 310
Monitoring Active Processes. 311
Monitoring System Processes. 314
Obtaining Information About a Device 316
Monitoring Memory Resources. 317
Monitoring Print and Batch Queues. 318
MONITOR Utility. 319
Information on a Specific Process. 323
Interactive Users. 324

IMPROVING SYSTEM PERFORMANCE. 325
Overview 325
Reconfiguring the System with AUTOGEN. 327
Reconfiguring the System with SYSGEN 329
Performance Tuning 332

FIGURES

1-1 The User Environment. 13
1-2 VAX-11/730 Hardware Configuration 15
1-3 VAX-11/750 Hardware Configuration 15
1-4 VAX-11/780 Hardware Configuration 16
1-5 VAX-11/782 Hardware Configuration 16

2-1 User Authorization File 35
2-2 User File Directory 46
2-3 Volume Quota File 49
2-4 System Accounting File. 66

3-1 Physical and Generic Print Queues 81
3-2 Physical, Generic, and Logical Print Queues 84
3-3 Print Job Header Pages and Header Lines 88
3-4 Batch Queues. 92

4-1 Volume Manipulation Commands. 123
4-2 Creating and Using a Disk or Tape Volume. 127
4-3 Relationship Between the Blocks, Clusters, and Extents of a Typical Disk File. 160
4-4 Map Pointers to the Data Blocks of a Disk File. . . . 162
4-5 Use of ODS-2 File Headers to Access a File. 163
4-6 Format of a Directory File Record 166
4-7 Format of an ANSI Tape Containing Two Files 168

6-1 Start-Up Phases of VAX/VMS. 210
6-2 VAX Console Control Panels. 214
6-3 Effect of VMS Start-Up on System Parameters 247

9-1 Using the SYSGEN Utility to Modify System Parameters 331

TABLES

1-1 Disk Comparison Chart 20
1-2 Tape Comparison Chart 20

1-3	Terminal and Print Comparison Chart	21
1-4	Card Reader Comparison Chart	22
2-1	Methods of Regulating VMS Processes	28
2-2	Standard User Records in the User Authorization File	30
2-3	Fields in a UAF Record Usually Unique to Each User	32
2-4	Fields in a UAF Record Usually Common to Groups of Users	33
2-5	Process Interaction Controlled by UIC	34
2-6	Getting the AUTHORIZE Utility Started	36
2-7	Managing the User Authorization File with the AUTHORIZE Utility	37
2-8	Displaying the Contents of the UAF with the AUTHORIZE Utility	38
2-9	Other AUTHORIZE Functions	40
2-10	Typical Login Command Procedures	42
2-11	Fields in a UAF Record Used to Enforce Additional Restrictions	44
2-12	Creating a User's Default Directory or Other UFD	45
2-13	Fields in a Quota File Record	47
2-14	Getting the DISKQUOTA Utility Started	50
2-15	Managing Individual Records in the Volume Quota File with the DISKQUOTA Utility	51
2-16	Managing the Volume Quota File as a Whole with the DISKQUOTA Utility	52
2-17	Displaying the Contents of the Volume Quota File, and Other DISKQUOTA Functions	53
2-18	Controlling Processes	58
2-19	Controlling Files and Volumes	59
2-20	Operations Requiring Operator Assistance	60
2-21	Enabling and Disabling Operator Terminals	62
2-22	Providing Operator Assistance	63
2-23	Controlling the Operator's Log	64
2-24	Sending Messages to Users	64
2-25	Recording Accounting Information	69
2-26	Creating an Accounting Report	73
3-1	Queueing a Print Job	80
3-2	Specifying the Priority of a Print Job	86
3-3	Setting the Characteristics of a Print Job	87
3-4	Submitting a Batch Job	91
3-5	Logical Name Definitions for Interactive and Batch Processes	94
3-6	Controlling the Batch Log File	95
3-7	Specifying the Priority of a Batch Job	96
3-8	Specifying the Characteristics of Batch and Print Jobs	100
3-9	Modifying a Batch or Print Job	101

3-10	Specifying When a Job Is Released for Processing	102
3-11	Job States	103
3-12	Managing Queues	106
3-13	Characteristics of Print Queues	108
3-14	Characteristics of Batch Queues	109
3-15	Stopping Logical and Generic Print Queues	112
3-16	Pausing Physical Print Queues to Start a New Box of Printer Paper or Repair Jammed Paper	113
3-17	Stopping Physical Print Queues to Change the Queue Characteristics or Delete the Queue	114
3-18	Reassigning Logical Queues	115
3-19	Moving Jobs from One Queue to Another	115
3-20	Stopping Batch Queues	116
4-1	Obtaining Information About Devices	128
4-2	Preparing Disk and Tape Volumes for Use	130
4-3	Sequence of Commands for Using Disk and Tape Volumes	131
4-4	Definitions of BACKUP Terms	132
4-5	Commonly Used BACKUP Functions and Commands	133
4-6	Protecting Volumes in the Initializing Phase	138
4-7	Effects of $MOUNT Qualifiers on a Volume	139
4-8	Backing Up Disk Volumes	143
4-9	Using the VERIFY Utility	147
4-10	Transferring Files Between VAX and PDP-11 Systems	155
4-11	Volume Information Contained in ODS-2 Reserved Files	157
4-12	ODS-1 and ODS-2 Comparison	167
4-13	Contents of the Magnetic Tape Volume Label	169
4-14	Contents of Tape File Header Labels	170
5-1	Files Controlling the VMS Environment	179
5-2	Assigning System Logical Names	182
5-3	Installing Known Images	186
5-4	Functions of the INSTALL Utility	187
5-5	Enabling RMS File Sharing	190
5-6	Recording Permanent Characteristics of Terminals	192
5-7	Establishing Ownership and Protection of Terminals and Other Nonsharable Devices	193
5-8	Adding a DCL Command	196
5-9	Adding HELP Text	197
5-10	System Directories	200
6-1	VAX Console Subsystem	212
6-2	VAX Console Switches	215
6-3	VAX Console Devices	218
6-4	Starting Up a VAX/VMS System from Power Off, Using the Default System Device	220

6-5 Starting Up a VAX/VMS System from Power On, Using the Default System Device 225
6-6 Automatic Powerfail Recovery. 227
6-7 Starting Up a VAX/VMS System, Explicitly Specifying the System Default Device. 229
6-8 Device Codes Used at System Start-Up. 230
6-9 Typical Console Commands. 235
6-10 Issuing Console Commands While VMS Is Running 238
6-11 Disabling the Use of Console Mode 239
6-12 CCL Command Files Used at Start-Up. 241
6-13 Using SYSBOOT During Conversational Start-Up. 246
6-14 Starting Up a VAX/VMS System Conversationally 248
6-15 Customizing the System Configuration. 250
6-16 Controlling Multiprocessing on the VAX-11/782 253
6-17 Emergency Shutdown from CCL 260

8-1 Analyzing a System Dump 286
8-2 Reports Generated by the System Dump Analyzer 287
8-3 Enabling and Disabling Error Logging. 291
8-4 Analyzing the Error Log 292
8-5 Typical Error Reports 293
8-6 Diagnostic Supervisor Programs. 297
8-7 Enabling Remote Diagnosis 302

9-1 System, Process, and Device Monitoring. 310
9-2 System States 313
9-3 Definition of System Processes. 314
9-4 Results of System Process Deletion. 315
9-5 SHOW DEVICES Command. 316
9-6 MONITOR Class Names 319
9-7 MONITOR PROCESSES Class Qualifiers. 319
9-8 System Files. 325
9-9 Running AUTOGEN When System Physical Resources Change. 327
9-10 Running AUTOGEN to Specify a System Parameter Value or File Size Manually 328
9-11 Using the SYSGEN Utility. 330

EXAMPLES

2-1 Sample UAF Record 31
2-2 Sample Brief List of UAF Records. 39
2-3 Typical SYLOGIN.COM Login Command Procedure 43
2-4 Sample List of Volume Quota File Records. 48
2-5 Adding a System User. 55
2-6 Removing a System User. 56

2-7 Command Procedure DELTREE.COM to Delete a UFD and its SFDs. 56
2-8 Request to Operator by REQUEST/REPLY. 61
2-9 Request to Operator by MOUNT. 61
2-10 Message to Operator by Request, No Reply Required 61
2-11 Accounting Report, Full Format. 68
2-12 Accounting Report, Brief Format 71
2-13 Accounting Report, Summary Format 72

3-1 Issuing the PRINT Command 79
3-2 Queue Status Display Corresponding to Figure 3-1. . . 82
3-3 Queue Status Display Corresponding to Figure 3-2. . . 85
3-4 Full Format Queue Status Display Corresponding to Figure 3-2 89
3-5 Issuing the SUBMIT Command. 90
3-6 Full Format Queue Status Display Corresponding to Figure 3-4 93
3-7 Sample Batch Log File 99
3-8 JOB_CONTROL and Related Processes 105
3-9 Sample Start-Up Commands Corresponding to Figure 3-2 107
3-10 Sample Start-Up Commands Corresponding to Figure 3-4 108

4-1 Preparing and Transferring Files to a Disk Volume . . 129
4-2 Transferring Files to a Tape Volume Using BACKUP. . . 134
4-3 Restoring a Directory Hierarchy from Tape (Saved with BACKUP) and Showing an Operator-Assisted MOUNT. . . . 136
4-4 Mounting a Disk with an Unknown Label 140
4-5 Initializing and Mounting a Public Disk 141
4-6 Backing Up a VAX-11/780 Using Stand-Alone BACKUP. . . 146
4-7 Using the VERIFY Utility. 148
4-8 Creating a Multivolume Tape Set 150
4-9 Reading a Multivolume Tape Set. 151
4-10 Creating a Disk Volume Set from an Existing Volume. . 153

5-1 Assigning System Logical Names with STARTUP.COM . . . 183
5-2 Assigning System Logical Names with SYSTARTUP.COM . . 183
5-3 Mounting Site-Specific Volumes with SYSTARTUP.COM . . 184
5-4 Typical INSTALL Commands in VMSIMAGES.DAT 188
5-5 Installing Known Images with SYSTARTUP.COM. 189
5-6 Connecting All Known Devices with STARTUP.COM 190
5-7 Enabling RMS File Sharing with STARTUP.COM. 191
5-8 Increasing RMS File Sharing with SYSTARTUP.COM. . . . 191
5-9 Recording Device Characteristics with SYSTARTUP.COM. 193
5-10 Announcing That the System Is Up with SYSTARTUP.COM. 194

5-11 Allowing Users to Log In with STARTUP.COM 194

6-1 Default Start-Up of a VAX-11/780. 221
6-2 Default Start-Up of a VAX-11/750, Not Using Console Volume. 222
6-3 Default Start-Up of a VAX-11/750, Using Console Volume. 223
6-4 Default Start-Up of a VAX-11/730. 224
6-5 Start-Up of a VAX-11/780 Specifying an RP05 as the System Device. 231
6-6 Start-Up of a VAX-11/750 Specifying an RM03 as the System Device. 232
6-7 Start-Up of a VAX-11/730 Specifying an RL02 as the System Device. 233
6-8 File DB0BOO.CMD from a VAX-11/780 Console Volume. . . 242
6-9 Customizing the Default System Device for Start-Up. 243
6-10 File RESTAR.CMD from a VAX-11/780 Console Volume. 244
6-11 Typical Conversational Start-Up 249
6-12 Performing an Orderly System Shutdown 257
6-13 User View of Orderly System Shutdown. 258

7-1 Installing VMS. 268
7-2 Installing VMS. 271
7-3 Installing the DECnet License 280

8-1 Report Produced by SDA Command SHOW CRASH 288
8-2 Report Produced by SDA Command SHOW SUMMARY 289
8-3 Backing Up the Dump File at Start-Up. 290
8-4 Error Log Summary Report. 294
8-5 Formatting an RK07 Volume 299
8-6 Formatting an RM03 Volume 300
8-7 Formatting an RP06 Volume 300
8-8 Formatting an RK07 Volume, Using a Configuration Command File. 301

9-1 Output from the SHOW SYSTEM Command 312
9-2 Sample SHOW MEMORY Output 317
9-3 MONITOR Screen Display of the POOL Class. 321
9-4 MONITOR PROCESSES/TOPCPU Screen Display 322
9-5 Output from SHOW PROCESS/CONTINUOUS 323
9-6 Output from SHOW USERS. 324

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" Y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO,D.F.

STUDENT GUIDE

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985.

COURSE DESCRIPTION

The VAX/VMS System Management course is designed for students who operate and manage a VAX/VMS system, including:

- Managing system resources
- Installing, updating, and upgrading the system and its layered products
- Monitoring and improving system performance

PREREQUISITES

- VAX/VMS Utilities and Commands course

COURSE GOALS

- Install and update VMS and typical layered products on standard configurations
- Set up user authorization and accounting files and procedures
- Set up queue and device characteristics
- Perform file back-up and restore operations
- Respond to user requests for operator assistance
- Identify potential system problems (hardware or software) by examining error log files
- Write and submit Software Performance Reports (SPRs)
- Monitor system activity and ensure satisfactory system performance

COURSE ORGANIZATION

This course is divided into units, called modules. Each module covers a skill or related group of skills required to fulfill the course goals.

In this textbook, each module consists of:

- An introduction to the subject matter of the module.
- One or more objectives, which describe what you intend to achieve by studying the module.
- A list of resources, the manuals you need to study the module. Some of these manuals are included with your course materials. Others may be available for reference in your classroom or lab.
- The text of each module, which includes outlines, tables, figures, and examples.

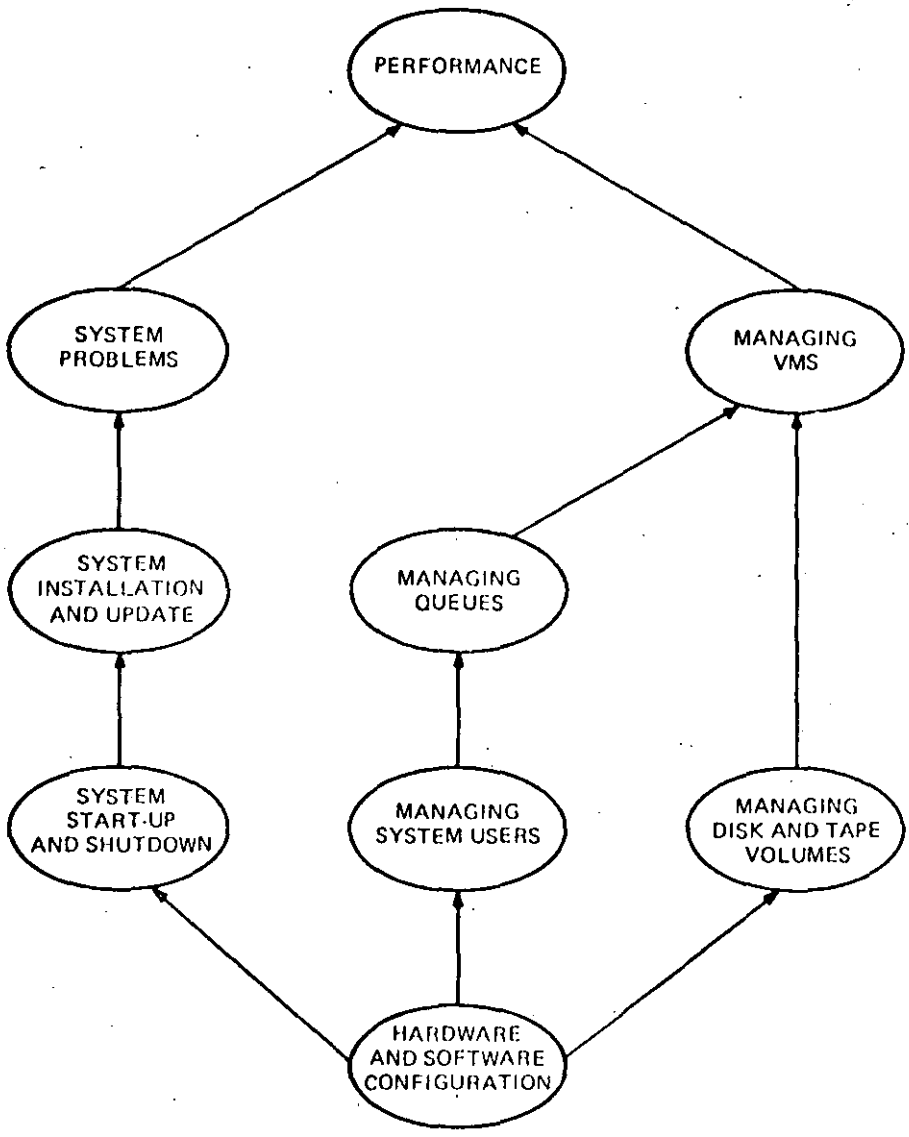
In addition, the Tests and Exercises booklet contains laboratory exercises for use in studying each module.

This course textbook is designed for use in a lecture classroom. Your instructor will follow its topic outlines in covering much of the subject matter of this course. As the instructor explains a particular subject, you can take notes in this textbook in the white space surrounding the subject outline. The outline will help you organize your notes. The large number of tables, figures, and exmaples that accompany the outline should reduce the amount of writing required during lecture, enabling you to channel your energies into understanding the subject matter and away from recording it.

The course map shows how each module is related to other modules and to the course as a whole. Before learning a module, you should master all its prerequisite modules. These prerequisites are below the selected module in the course map and are connected to it by a chain of upward-pointing arrows.

This material can be effectively presented in many different ways. By providing prerequisite information as needed, your instructor may choose to use a module organization different from the one in the course map.

COURSE MAP


PERFORMANCE
SYSTEM PROBLEMS
MANAGING VMS
SYSTEM INSTALLATION AND UPDATE
MANAGING QUEUES
SYSTEM START-UP AND SHUTDOWN
MANAGING SYSTEM USERS
MANAGING DISK AND TAPE VOLUMES
HARDWARE AND SOFTWARE CONFIGURATION
TK-9236

RESOURCES

Some of these resources will be given to you. Others will be available for reference in your classroom or lab.

The following manuals are heavily used throughout this course:

1. VAX/VMS System Management and Operations Guide
2. VAX-11 Utilities Reference Manual

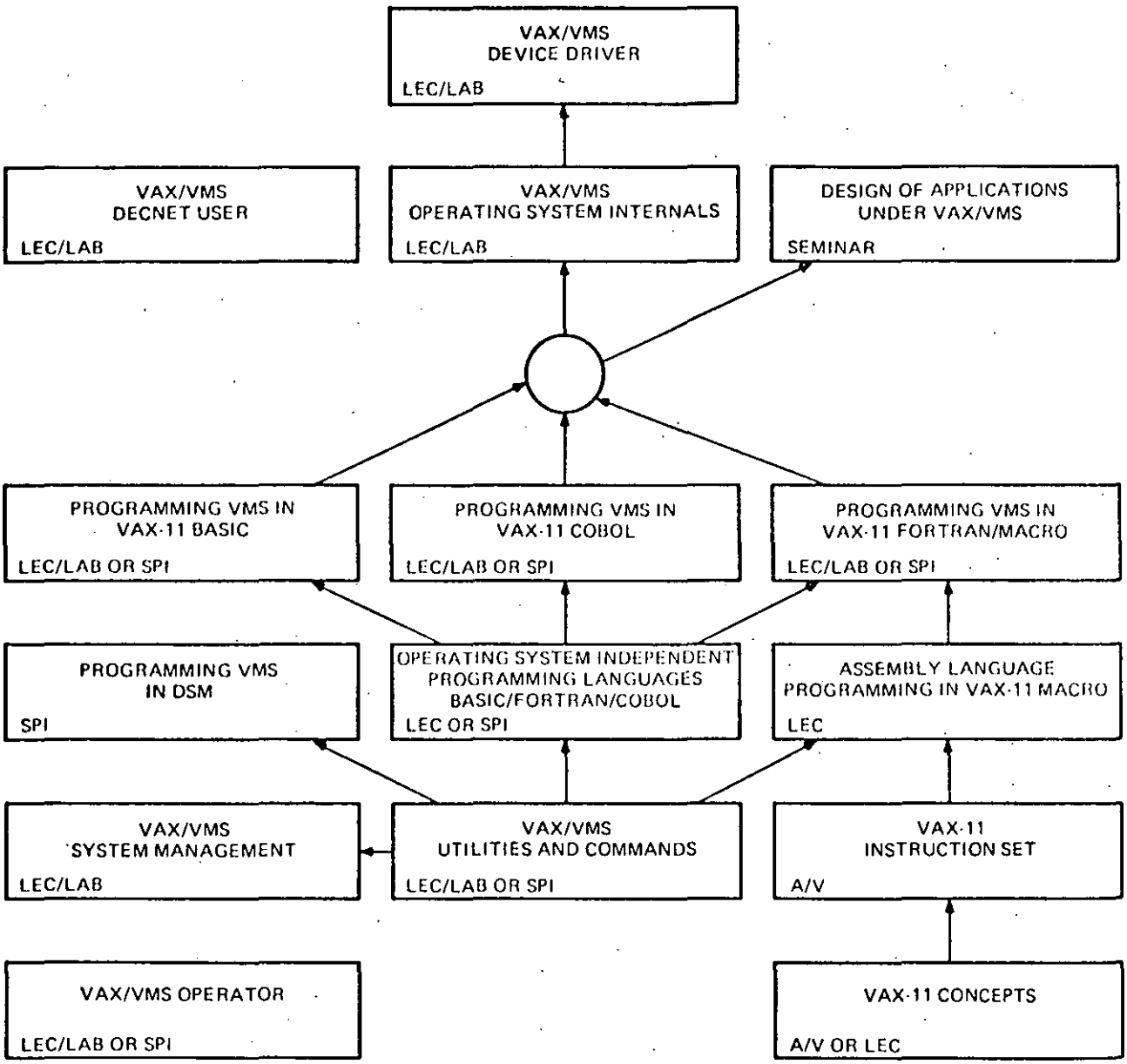
The following manuals are of use in more than one module.

1. VAX/VMS Release Notes
2. VAX/VMS UETP User's Guide
3. VAX-11/780 Software Installation Guide
4. VAX-11/750 Software Installation Guide
5. VAX-11/730 Software Installation Guide
6. VAX-11/782 User's Guide

The following manuals and documents are also useful.

1. VAX/VMS System Dump Analyzer Reference Manual
2. VMS Version V3.0 Software Product Description
3. VAX Hardware Handbook

VAX/VMS CURRICULUM



KEY

LEC	=	LECTURE
LEC/LAB	=	LECTURE AND LAB
SPI	=	SELF-PACED INSTRUCTION
A/V	=	AUDIOVISUAL INSTRUCTION

TK-9040

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO,D.F.

HARDWARE AND SOFTWARE CONFIGURATION

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985

INTRODUCTION

As system manager, you should be familiar with available VAX hardware and VMS software. This knowledge will be of value in understanding the capability of your current system and planning for its future expansion.

The official document describing the hardware, operating system software, and optional software supported by the VMS operating system is the Software Product Description (SPD) for the VAX/VMS operating system. This document is reissued with each update of VMS. The information in this module corresponds to the SPD for VAX/VMS Version 3.Ø. Your instructor may provide you with additional information from a more recent SPD.

OBJECTIVES

A system manager must identify VAX/VMS hardware and software, including:

- Types of hardware supported by VMS
- Types of software supported by VMS

RESOURCES

1. VAX/VMS Software Product Description

THE USER ENVIRONMENT

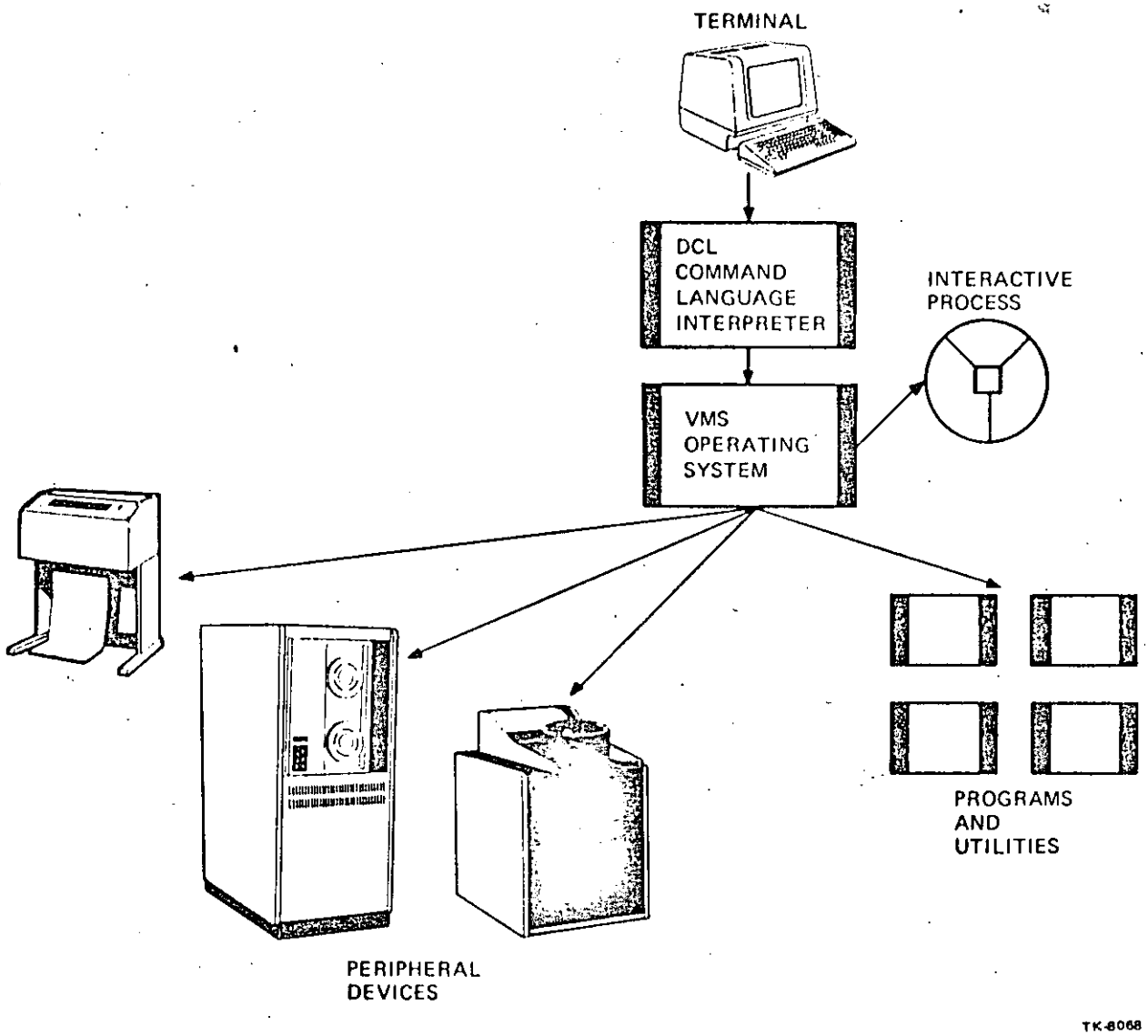


Figure 1-1 The User Environment

USER HARDWARE ENVIRONMENT

- Central Processing Unit (CPU)
- The Console Subsystem
  - Operator's console
  - Console storage device
- The Main Memory Subsystem
  - Battery backup
- The Input/Output Subsystem
  - MASSBUS
    - Fast disk units
    - Tape units
  - UNIBUS
    - Card readers
    - Line printers
    - Terminals
    - Slower disk units
    - Slower tape units
  - IDC (Integrated Disk Controller)
    - Disk units on VAX-11/730
- Communications
  - Dial-in phone lines
  - Processor-to-processor communication link

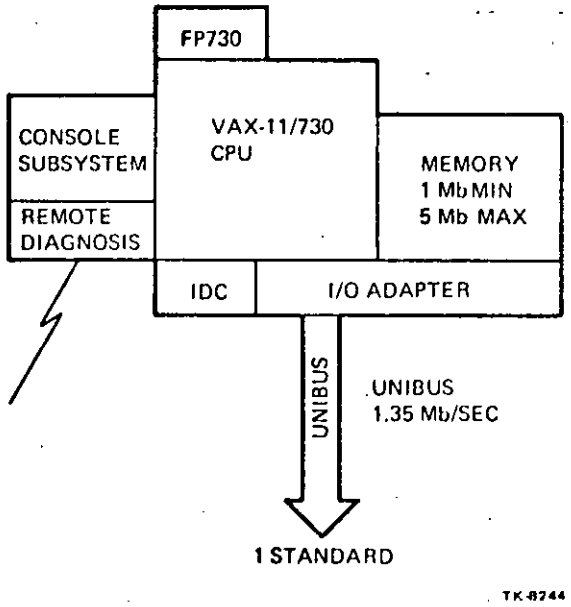


Figure 1-2 VAX-11/730 Hardware Configuration

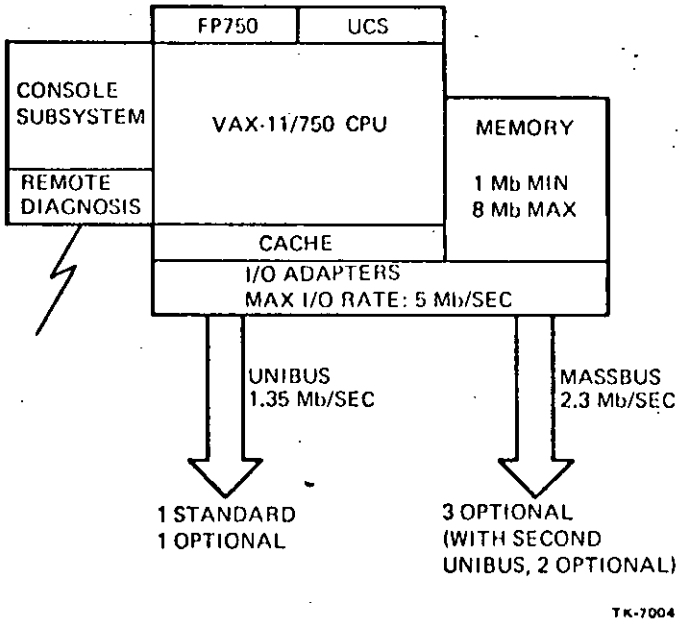


Figure 1-3 VAX-11/750 Hardware Configuration

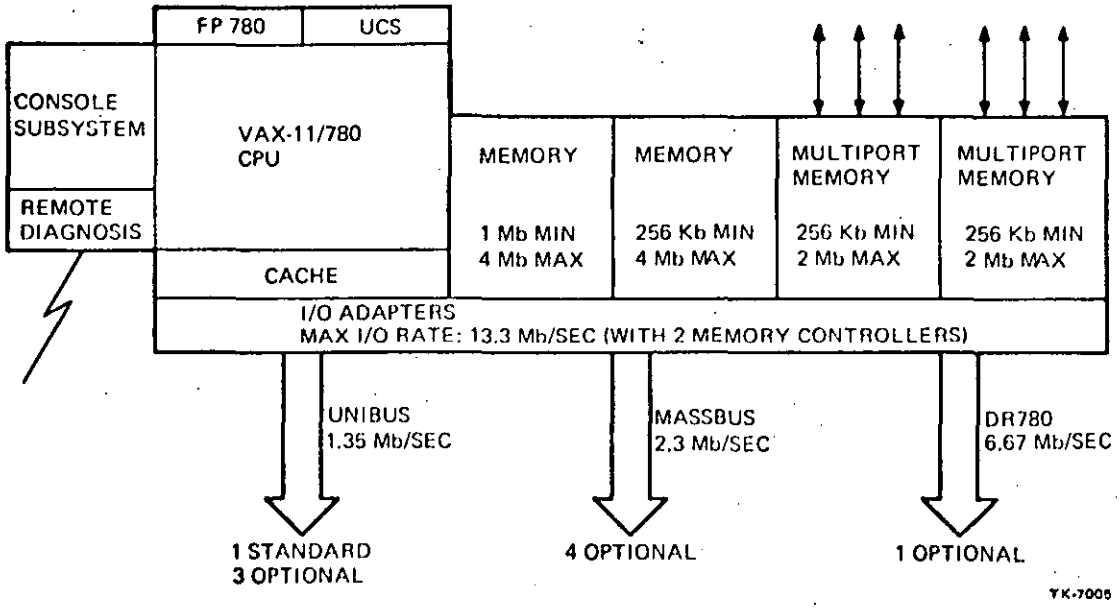


Figure 1-4 VAX-11/780 Hardware Configuration

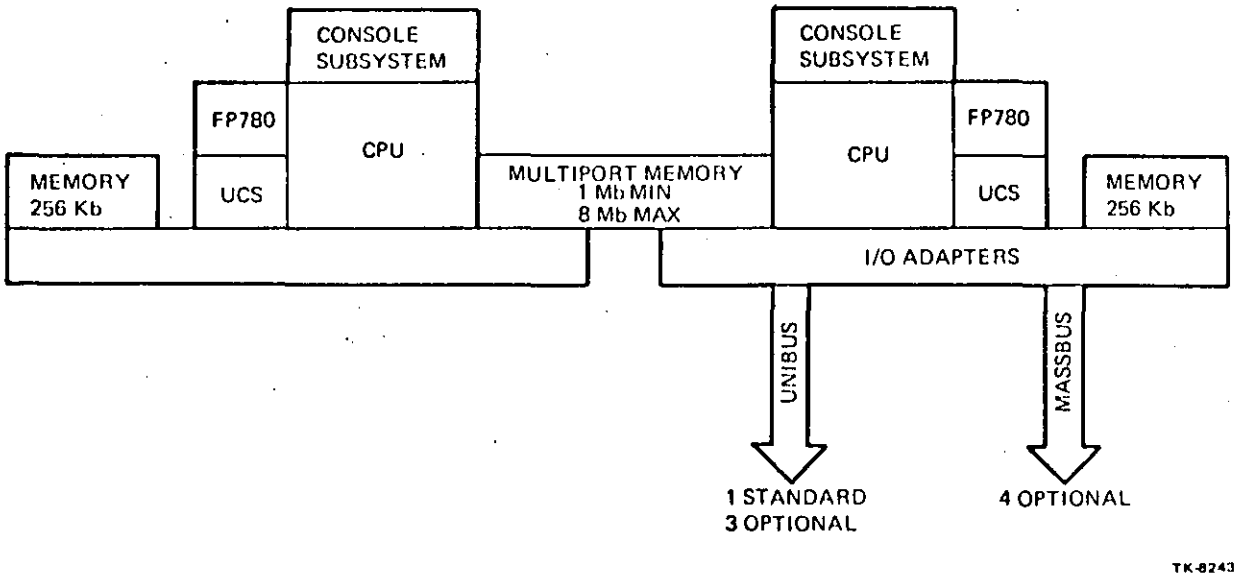


Figure 1-5 VAX-11/782 Hardware Configuration

USER SOFTWARE ENVIRONMENT

Major Software Components

- VMS Operating System
  - Controls peripheral devices
  - Manages user processes
- Applications Migration Executive (AME)
  - Allows RSX-11M programs to execute on VAX hardware
- VAX-11 Record Management Services (RMS)
  - General file/record management tools
- Command Language Interpreters (CLI)
  - Interface between user and system
  - DIGITAL Command Language (DCL)
  - Monitor Console Routine (MCR)
- Program Development Tools
  - Text editors (EDT, SOS, TECO, EDI)
  - VAX-11 MACRO Assembler
  - VAX-11 Linker
  - VAX-11 Symbolic Debugger
  - VAX-11 Librarian
  - VAX-11 System Libraries

Other Software Components

- Word Processing Tools
  - Interactive Text Editors
  - RUNOFF Text Formatter
  - SLP Batch Text Editor
- Volume Manipulation Tools
  - INITIALIZE Utility
  - MOUNT Utility
  - BACKUP Utility
  - Volume Verification Utility
- Other Utilities
  - File Sorting and Merging Utility
  - Informational Utilities
  - Personal Mail Utility
  - Telephone Utility
  - Text String Search Utility

Optional Software Products

- Information Management Tools
  - Data Base Management Services (DBMS)
  - DATATRIEVE
  - Common Data Dictionary (CDD)
  - Forms Management Services (FMS)
- Programming Languages
  - BASIC
  - BLISS
  - C
  - COBOL
  - CORAL-66
  - DIBOL
  - DIGITAL Standard MUMPS (DSM)
  - FORTRAN
  - PASCAL
  - PL/1
- Communication Packages
  - DECnet/VAX
  - VAX-11 2780/3780 Protocol Emulator
  - VAX-11 3271 Terminal Emulator
  - VAX-11 MUX200 Terminal Emulator
  - DX/VMS
- Application Management Tools
  - DECCMS

Table 1-1 Disk Comparison Chart

Model	Controller Interface	MB/ Drive	Drives/ Controller	Media Type	Average Access Time(msec)	Peak Transfer (MB/sec)	Access Dual Option
RP07	MASSBUS	516	8	DISK PACK	31.3	1.3/2.2*	YES
RM05	MASSBUS	256	8	DISK PACK	38.3	1.2	YES
RP06	MASSBUS	176	8	DISK PACK	38.3	.806	YES
RM80	MASSBIS	124	8	FIXED	33.3	1.2	YES
RM03	MASSBUS	67	8	DISK PACK	38.3	1.2	YES
RK07	UNIBUS	28	8	CARTRDIGE	49	.538	YES
RL02	UNIBUS/IDC	10.4	4	CARTRIDGE	67.5	.512	NO
RX02	UNIBUS	0.5	2	DISKETTE	263	.062	NO

*Interleaved memory is required for a transfer rate of 2.2 MB/sec.

Table 1-2 Tape Comparison Chart

Model	Controller Interface	Number of Tracks	Drives/ Controller	Density (BPI)	Speed (IPS)	Capacity/ Reel (MB)	Data Transfer (KB/sec)
TU58*	UNIBUS	N/A	2	800	30	.25	38.4
TS11	UNIBUS	9	1	1600	45	32	72
TE16	MASSBUS	9	8	800-1600	45	32	72
TU45	MASSBUS	9	8	800-1600	75	32	120
TU77	MASSBUS	9	8	800-1600	125	32	200
TU78	MASSBUS	9	8	1600-6250	125	145	781

*The TU58 is treated like a disk by VAX/VMS.

Table 1-3 Terminal and Printer Comparison Chart

Model	Max. Print Speed (char/sec)	Number of Columns	Printing Characters	Printing Type	Font Sizes	Bidirectional Printing
Hardcopy						
LA34/LA38	30	132*	96	9X7 Dot Matrix	4	NO
LA120	180	132*	96	7X7 Dot Matrix	8	YES
Video						
VT100	1920	132*	94	9X7 Dot Matrix	4	N/A
Graphing						
VT105**	1920	132*	96	9X7 Dot Matrix	4	N/A
VT125	1920	132*	96	9X7 Dot Matrix	4	N/A
GIGI**						
VS11						
Line Printers	(lines/min)					
LP11-AA	300	132	64	Band	1	N/A
LP11-BA	300	132	64	Band	1	N/A
	215		96			
LP11-CA	900	132	64	Impact	1	N/A
LP11-DA	600	132	96	Impact	1	N/A
LP11-VA	300	132	64	Impact	1	N/A
LP11-WA	240	132	96	Impact	1	N/A
LP11-YA	600	132	64	Impact	1	N/A
LP11-ZA	436	132	96	Impact	1	N/A
Printer/ Plotters						
LXY11**	300	132	96			
	170			Dot Matrix		
LXY21**	600	132	96			
	320			Dot Matrix		

* Number of columns varies with font size.
** Optional plotting/graphing software available.

Table 1-4 Card Reader Comparison Chart

Model	Speed (cards/min)	Card Size
CR11-A	300	80
CR11-BC	600	80
CME11-KA	250	40/80

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" Y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO,D.F.

MANAGING SYSTEM USERS

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985

INTRODUCTION

The resources of a single VAX/VMS system are limited. A given configuration has a fixed amount of physical memory, a fixed amount of disk storage, and a fixed number of peripheral devices. These resources must be shared by all system users.

VMS enforces restrictions on each user's access to system resources. By controlling these restrictions, you can promote fair and effective resource sharing.

VMS also restricts the ability of each user to affect other users and to affect VMS itself. By controlling these restrictions, you can maintain system security.

The principal methods available on VMS to regulate individual processes are:

- Priority
- Privileges
- Limits
- Disk quotas
- Protection
- Access mode

The system resource or ability regulated by each of these methods is explained in Table 2-1.

To define restrictions on individual user processes and to record system activity, you use VMS utilities and DCL commands that create and maintain the following files, discussed in this module:

- User authorization file (SYS$SYSTEM:SYSUAF.DAT)
- Quota files on disk volumes ([000000]QUOTA.SYS)
- Login files (for example, SYS$MANAGER:SYLOGIN.COM)
- Accounting file (SYS$MANAGER:ACCOUNTNG.DAT)

OBJECTIVES

To share limited system resources among users and processes, a system manager must:

- Set up user authorization records, directories, and disk quotas
- Collect accounting information
- Restrict the abilities of certain users or applications
- Respond to requests for operator assistance

RESOURCES

1. VAX/VMS Command Language User's Guide
2. VAX/VMS System Management and Operations Guide
3. VAX-11 Utilities Reference Manual

MANAGING SYSTEM RESOURCES

- System resources
  - Physical memory
  - Disk storage
  - Peripheral devices
- Must promote fair and effective resource sharing
  - VMS enforces restrictions on user's access
  - System manager controls restrictions
- Must maintain system security
  - Restrict ability of one user to affect another
  - Restrict user ability to affect VMS itself
- Files defining user restrictions and recording system activity
  - User authorization file

    (SYS$SYSTEM:SYSUAF.DAT)
  - Quota files on disk volumes

    ([000000]QUOTA.SYS)
  - Login files

    (for example, SYS$MANAGER:SYLOGIN.COM)
  - Accounting file

    (SYS$MANAGER:ACCOUNTNG.DAT)
- Methods of regulating individual processes
  - Priority
  - Privileges
  - Limits
  - Disk quotas
  - Protection
  - Access mode

Table 2-1 Methods of Regulating VMS Processes

Resource or Ability Regulated	Method of Regulation	Comments
Access to CPU	Priority	• VMS uses process priority to schedule process execution. • Priority ranges from 0 (lowest) to 31 (highest). • Timesharing processes have priorities 0-15. VMS can automatically adjust the priorities of timesharing processes to promote greater CPU sharing. • Real-time processes have priorities 16-31. VMS does not automatically adjust the priorities of real-time processes.
Ability to affect other VMS users and to affect VMS itself	Privileges	• In many cases, exercising a privilege results in the use of physical memory or other system resources. • There is no restriction on the number of times a process can exercise a privilege.
Ability to use certain system resources	Limits	• Limits restrict the amount of a system resource a process can use at a given time. • Limits are expressed numerically. • Limits are also called quotas in some documentation. Do not confuse limits with disk quotas, described below.
Ability to store files on a disk volume	Disk quotas	• Disk quotas restrict the number of blocks of storage a process can use on a volume. • Disk quotas are enforced volume by volume, not device by device or system wide.

Table 2-1 Methods of Regulating VMS Processes (Cont)

Resource or Ability Regulated	Method of Regulation	Comments
Access to devices, volumes, and files on volumes; also, ability of processes to exchange information and affect each other	Protection	• Every process has a UIC. • Devices, volumes, files, and interprocess communication structures have an owner UIC and a protection code. • VMS protection is based on a comparison of the process UIC with the owner UIC and protection code of the protected structure.
Ability to modify or delete certain data structures, such as logical names	Access mode	• Data structures created at more powerful access modes can be used at less powerful access modes; however, they cannot be modified or deleted at less powerful access modes. • The access modes are numbered from 0 (most powerful) to 3 (least powerful). • The access modes, and the routines that typically run in them, are: 0 Kernel — VMS kernel routines 1 Executive — RMS routines 2 Supervisor — DCL, MCR routines 3 User — User programs

AUTHORIZING AND MANAGING SYSTEM USERS

User Authorization File

- A record for each authorized user name
- Should reflect the needs of users and resources of the system
- Four reserved user names
  - SYSTEM
  - FIELD
  - SYSTEST
  - DEFAULT

Table 2-2 Standard User Records in the User Authorization File

User Name (and Initial Password) of UAF Record	Reasons to Log In With This User Name	Comment
*SYSTEM (MANAGER)	For software installation, system bootstrapping, and system problem diagnosis	Do not log in as SYSTEM for routine system management functions. There are too many dangerous privileges, particularly BYPASS. Create a UAF record with UIC [1,4] and SETPRV, but without dangerous privileges, to use for routine system management functions.
+SYSTEST (UETP)	For running the User Environment Test Package (UETP), which tests VAX/VMS hardware and software	
+FIELD (SERVICE)	For running hardware diagnostics	
*DEFAULT (USER)	Not possible to log in as DEFAULT	This record is used by the AUTHORIZE command ADD to supply default values for new UAF records.

* These records are present whenever a UAF is created. They cannot be removed or renamed. The password of DEFAULT cannot be changed. System security requires that you change the password of SYSTEM.

+ These records are present in the UAF that is distributed with VMS. It is possible to remove them, but it is not recommended. System security requires that you change the passwords of SYSTEST and FIELD.

- Fields in the UAF record

```
(1) Username:  BROWN      (2) Owner:  MARY BROWN
(3) Account:   GRP32      (4) UIC:    [011,015]
(5) CLI:       DCL        (6) LGICMD:
(7) [Default Device: DISK$USER:
    Default Directory: [BROWN]
(8) Login Flags:
(9) [Primary days:   Mon Tue Wed Thu Fri
    Secondary days:                  Sat Sun
    No hourly restrictions
    PRIO:     4  BYTLM:        4096  BIOLM:          6
    PRCLM:    2  PBYTLM:          0  DIOLM:          6
(10)ASTLM:   10  WSDEFAULT:     150  FILLM:         20
    ENQLM:   10  WSQUOTA:       200  SHRFILLM:       0
    TQELM:   10  WSEXTENT:      500  CPU:      no limit
    MAXJOBS:  0  MAXACCTJOBS:     0  PGFLQUOTA:  10000
(11)[Privileges:
      TMPMBX NETMBX
```

Example 2-1 Sample UAF Record

- Certain fields usually unique to each user

Table 2-3 Fields in a UAF Record Usually Unique to Each User

Field	Meaning	AUTHORIZE Qualifier	Key to Example 2-1
User name	Identifies user for whom VMS creates process. Typed in response to "Username:" prompt.	None. Parameter, not qualifier.	1
Password	Used to confirm identity of user. Typed in response to "Password:" prompt.	/PASSWORD	Not displayed, to preserve security.
UIC	Used to enforce protection of devices, volumes, and files. Also used to regulate ability of process to communicate with and affect other processes.	/UIC	4
Owner	Used to record full name of user, for convenience of system manager. Not used by VMS. To allow spaces in name, place entire name between quotation marks.	/OWNER	2
Default directory	Used by VMS to establish both initial default device and initial default directory. Note that it is preferable to identify the default device by a logical name based on the volume label (such as DISK$USER), not by a physical device name (such as DBA0:).	/DEVICE /DIRECTORY	7

- Other fields usually common to groups of users

Table 2-4 Fields in a UAF Record Usually Common to Groups of Users

Field	Meaning	AUTHORIZE Qualifier	Key to Example 2-1
Account	Used by ACCOUNTING utility to identify processes whose activities are billed in common.	/ACCOUNT	3
CLI	Command language interpreter. Usually, DCL.	/CLI	5
LGICMD	User's login command file. Commands in this file are executed before the user can issue commands interactively or through a batch command file. If this field is left blank, VMS assumes that the file name of the login command file is LOGIN, located in the default directory.	/LGICMD	6
Privileges	Privileges to add to user record or remove from user record.	/PRIVILEGES This qualifier has a keyword for each privilege. For example, GRPNAM (to add this privilege) and NOGRPNAM (to delete it).	11
Limits	Limits to record in user record.	One qualifier for each limit. For example, /TQELM, /ASTLM.	10

Process Interaction Controlled by UIC

- In assigning UICs to users, determine the extent to which the user processes need to:
  - Share access to files, volumes, and devices
  - Communicate interactively with one another
  - Affect or control one another
- For greater interaction, place users in same UIC group
- For greater protection, place users in different UIC groups
- UIC groups 1 through 10 are reserved for VMS and the system manager

Table 2-5 Process Interaction Controlled by UIC

Interaction Available Among Processes in the Same UIC Group	Comment
Sharing files, volumes, and devices, yet denying access to processes in other groups	Accomplished by setting the protection codes of the files, volumes, and devices.
Communicating with each other by means of structures that cannot be accessed by processes in other groups	For example: • Group logical names • Group mailbox logical names • Group global section names • Common event flag clusters • Process names
Affecting and controlling other processes in same UIC group	Requires GROUP privilege.

Running the AUTHORIZE Utility

- AUTHORIZE utility accesses User Authorization File SYS$SYSTEM:SYSUAF.DAT

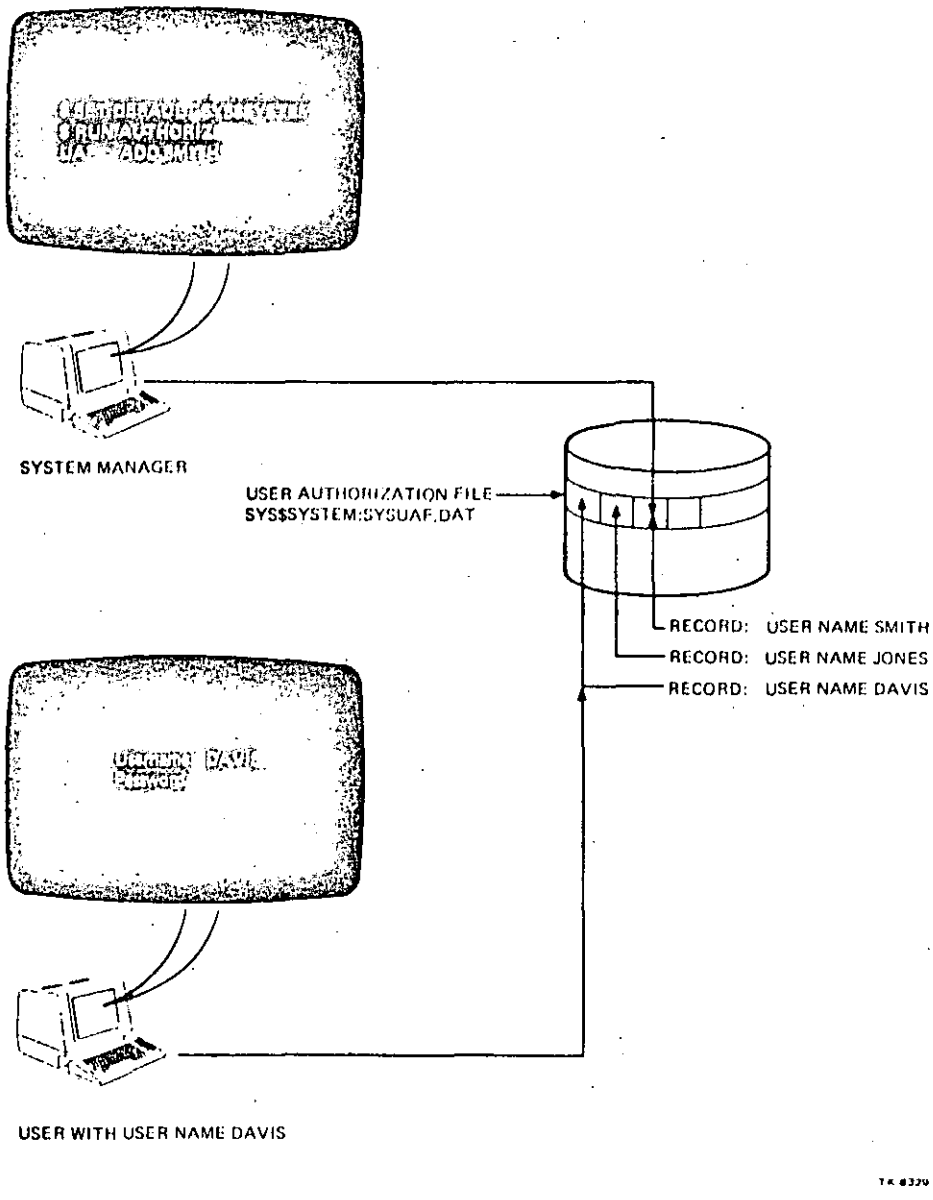


Figure 2-1 User Authorization File

Table 2-6 Getting the AUTHORIZE Utility Started

DCL Commands Used to Run AUTHORIZE	Comments
Step 1: $ SET UIC [1,4]	This command requires CMKRNL privilege. If the UIC of your interactive process is [1,4], this step is unnecessary.
Step 2: $ SET DEFAULT SYS$SYSTEM	The AUTHORIZE utility manages the file SYSUAF.DAT in your default device and directory. To affect SYS$SYSTEM:SYSUAF.DAT, set your default to SYS$SYSTEM.
Step 3: $ RUN AUTHORIZE	If no SYSUAF.DAT exists, AUTHORIZE asks whether you want to create a SYSUAF.DAT. If you answer "yes," it creates one. If you know SYS$SYSTEM:SYSUAF.DAT really does exist, answer "no." Then: A) Make sure that the logical name SYSUAF, if it exists, points to SYS$SYSTEM:SYSUAF.DAT. B) Follow steps 1-3 again.

- AUTHORIZE can:
  - Add user records
  - Modify user records
  - Remove user records

Table 2-7 Managing the User Authorization File with the AUTHORIZE Utility

Operation on the UAF	Command Format
Adding a new user record, taking all values not specified by qualifiers from the DEFAULT record (If the /PASSWORD qualifier is omitted, the password USER is supplied.)	UAF>ADD new-user-name [/qualifier...]
Adding a new user record, taking all values (except password) not specified by qualifiers from an existing user record (The password must be specified, even if it is to remain the same.)	UAF>COPY existing-user-name new-user-name - _/PASSWORD=password [/qualifier...]
Modifying an existing user record	UAF>MODIFY existing-user-name /qualifier [...]
Modifying all existing user records	UAF>MODIFY * /qualifier [...]
Modifying all existing user records with UICs in a particular group	UAF>MODIFY [group-number, *] /qualifier [...]
Renaming an existing user record (The password must be specified, even if it is to remain the same.)	UAF>RENAME existing-user-name new-user-name - _/PASSWORD=password
Modifying the DEFAULT record (The qualifier /PASSWORD does not affect the default password, which is always USER.)	UAF>MODIFY DEFAULT /qualifier [...] or UAF>DEFAULT /qualifier [...]
Removing an existing user record	UAF>REMOVE existing-user-name

- AUTHORIZE can display UAF records

Table 2-8 Displaying the Contents of the UAF with the AUTHORIZE Utility

	Command	Qualifier	Parameter
To direct display:			
• To your terminal	SHOW		
• To listing file SYSUAF.LIS	LIST		
To use display format:			
• Full		/FULL (default for SHOW)	
• Brief		/BRIEF (default for LIST)	
To select and sort records by:			
• User name			user-name (Wild card characters * and % are permitted.)
• UIC			[UIC] (Wild card character * is permitted for entire group or user number.)

Examples:

```
UAF>SHOW JONES
UAF>LIST *
UAF>SHOW/BRIEF J*
UAF>LIST/FULL [1,4]
UAF>LIST [31,*]
```

```
      Owner          Username        UIC      Account  Privs Pri Default Directory

MARY BROWN           BROWN       [011,015] GRP32    Normal  4 DISK$USER:[BROWN]
JOHN BUFORD          BUFORD      [011,010] CC3F5    All     4 DISK$USER:[BUFORD]
                     DECNET      [051,001]          Normal  4 SYS$SYSROOT:[DECNET]
                     DEFAULT     [011,000] CC3F5    All     4 DISK$USER:[COURSE]
JACK DORSEY          DORSEY      [011,030] CC3F5    All     4 DISK$USER:[DORSEY]
                     FIELD       [001,010]          All     4 DRA0:[SYSMAINT]
ALAN FRIEDMAN        FRIEDMAN    [011,040] CC3F5    All     4 DISK$USER:[FRIEDMAN]
LIZ HELLER           HELLER      [011,110] CC3F5    All     4 DISK$USER:[HELLER]
MERYL HLYNKA         HLYNKA      [011,120] CC3F5    All     4 DISK$USER:[HLYNKA]
LIZ HUNT             HUNT        [011,140] CC3F5    All     4 DISK$USER:[HUNT]
FRED MARSH           MARSH       [011,220] CC3F5    All     4 DISK$USER:[MARSH]
EMILIE MELVIN        MELVIN      [011,240] CC3F5    All     4 DISK$USER:[MELVIN]
VIK MUIZNIKES        MUIZNIEKS   [011,250] CC3F5    All     4 DISK$USER:[MUIZNIEKS]
JIM MILLER/TOM KIMBL NETMGR      [013,002]          All     4 SYS$SYSDISK:[NETMGR]
                     OPER        [041,020]          All     4 SYS$SYSROOT:[OPER]
KUAN OU              OU          [011,310] CC3F5    All     4 DISK$USER:[OU]
ROCKY CARDALISCO     ROCKY       [011,320] CC3F5    All     4 DISK$USER:[ROCKY]
CHARLOTTE SIGMON     SIGMON      [011,330] CC3F5    All     4 DISK$USER:[SIGMON]
SYSTEM MANAGER       SYSTEM      [001,004] SYSTEM   All     4 SYS$SYSROOT:[SYSMGR]
DEBBIE TOWLE         TOWLE       [011,340] CC3F5    All     4 DISK$USER:[TOWLE]
```

Example 2-2 Sample Brief List of UAF Records

- AUTHORIZE has HELP facility

Table 2-9 Other AUTHORIZE Functions

Operation	Command Format
Displaying information about AUTHORIZE commands on your terminal	UAF>HELP [command [/qualifier]]
Exiting from AUTHORIZE	UAF>EXIT

Customizing the User Environment

- Define the user environment using system manager's login command procedures
- Login command procedures are executed each time interactive process or batch job is created
- How VMS finds these command procedures
  - System logical name SYS$SYLOGIN
  - Default user login file LOGIN.COM
  - LGICMD field in each user's UAF record
- VMS login command sequence:
  - If SYS$SYLOGIN is defined, execute command file it designates
  - If LGICMD is not blank, execute command file it designates
  - If LGICMD is blank, execute LOGIN.COM, if it exists in user's login default directory

- Three common uses of login command procedure
  - Personal login only
  - System and personal login
  - Captive login

Table 2-10 Typical Login Command Procedures

Function of Login Command Files	Definition of System Logical Name SYS$SYLOGIN and UAF Record Field LGICMD
Personal login only: Each DCL user creates the file LOGIN.COM in his own login default directory to customize his own environment.	SYS$LOGIN undefined LGICMD undefined
System and personal login: The system manager creates the DCL command file SYS$MANAGER:SYLOGIN.COM to customize a common user environment. In addition, each DCL user creates the file LOGIN.COM in his own login default directory to customize his own environment.	SYS$SYLOGIN defined as SYS$MANAGER:SYLOGIN LGICMD undefined
Captive login: The system manager creates a file in the SYS$MANAGER directory, such as SYS$MANAGER: CAPTIVE.COM, to customize the user environment and prevent a user from changing it. Typically, the command procedure examines each DCL command typed by the user, to decide whether or not to allow it to be executed. It may even implement its own private command language. A captive command procedure must contain a loop to prevent it from exiting, and the FLAGS field of the UAF record must specify the CAPTIVE, DISCTLY, and LOCKPWD flags. This FLAGS field is mentioned in Table 2-11.	SYS$SYLOGIN definition optional, affects captive and non-captive users. LGICMD defined as SYS$MANAGER:CAPTIVE for captive users, and /FLAGS=CAPTIVE set as well.

```
$       V = 'F$VERIFY(0)'       !Save verification status.
$       SET NOCONTROL = Y       !Make sure that entire
$       SET NOON                !command procedure executes.
$
$ !
$ !     ENABLE <CTRL/T> HANDLING BY DCL
$ !
$       SET CONTROL = T
$ !
$ !     DEFINE SYMBOLS FOR COMMONLY USED DCL COMMANDS AND INSTALLED UTILITIES
$ !
$       DISPLAY         ==      "MONITOR PROCESS/TOPC"
$       NCP             ==      "$NCP"
$       DMAIL           ==      "$DMAIL"
$       SPELL           ==      "$SPELL"
$       USERS           ==      "SHOW USERS"
$       ERASE           ==      "$ERASE"
$       BYE             ==      "$ERASE LOGOUT"
$       INFO            ==      "SHOW PROCESS/CONTINUOUS"
$       SUSPEND         ==      "SET PROCESS/SUSPEND"
$       RESUME          ==      "SET PROCESS/RESUME"
$ !
$ !     DEFINE SYMBOLS FOR COMMONLY USED COMMAND PROCEDURES
$ !
$       BATCH           ==      "@SYS$LIBRARY:BATCH"
$       FORK            ==      "@SYS$LIBRARY:FORK"
$       CALC            ==      "@SYS$LIBRARY:CALC"
$       CHECKBLD        ==      "@WRKD$:[CHECKIN]CHECKBLD"
$       CHECKIN         ==      "@WRKD$:[CHECKIN]CHECKIN"
$       CHECKOUT        ==      "@WRKD$:[CHECKIN]CHECKOUT"
$       NOTICES         ==      "@SYS$MANAGER:NOTICES"
$ !
$ !     DEFINE SYMBOLS FOR COMMONLY USED NOTES FILES
$ !
$       SYSNOTES        ==      "NOTES SYS$NOTES:SYSNOTES"
$       NETNOTES        ==      "NOTES SYS$NOTES:NETNOTES"
$       CRASH           ==      "NOTES SYS$NOTES:CRASH"
$ !
$ !     FOR INTERACTIVE PROCESSES ONLY, DISPLAY CURRENT SYSTEM NOTICES
$ !     FOR ALL PROCESSES, RESTORE ORIGINAL STATUS
$ !
$       IF F$MODE() .EQS. "INTERACTIVE" THEN TYPE SYS$MANAGER:NOTICE.TXT
$       IF V THEN SET VERIFY    !Restore verification status
$       SET CONTROL = Y         !and <CTRL/Y> response.
$       EXIT
```

Example 2-3 Typical SYLOGIN.COM Login Command Procedure

Restricting User Activity

- Captive login
  - /FLAGS = (CAPTIVE, DISCTLY, LOCKPWD)
- SYSTEM record
  - /PFLAGS = (DISNETWORK, DISDIALUP)
  - /SFLAGS = (DISNETWORK, DISDIALUP)
- SYSTEST and FIELD records
  - /FLAGS = DISUSER when not needed
  - /FLAGS = NODISUSER when needed

Table 2-11 Fields in a UAF Record Used to Enforce Additional Restrictions

Field	Meaning	AUTHORIZE Qualifier	Key to Example 2-1
Login flags	Used to prevent a user from overriding certain fields in the UAF record, such as: • Default CLI • Default device • Login command procedure Also, used to disable the use of: • SET PASSWORD • <CTRL/Y>	/FLAGS	8
Primary days Primary hours and flags Secondary hours and flags	Used to divide the days of the of the week into two categories, primary and secondary. For each category, used to specify: • Hours when allowed to log in • Whether allowed to log in over dial-up line • Whether allowed to log in using SET HOST from another network node	/PRIMEDAYS /PHOURS /PFLAGS /SHOURS /SFLAGS	9

Managing Disk Volumes

- Each user's UAF record logically defines
  - Default device
  - Default directory
- AUTHORIZE puts values in UAF
- CREATE/DIRECTORY DCL command physically creates the default directory
- DISKQUOTA utility limits amount of disk storage space available

Creating a User File Directory

Table 2-12 Creating a User's Default Directory or Other UFD

Steps in Creating a UFD	Comments
Step 1: $ SHOW DEVICE volume-log-name	Read the information displayed by this command to find out whether the volume on which the directory is to be created is mounted. Although the SHOW DEVICE command accepts a physical device name, it is preferable to use a logical name to refer to the volume. For example, if the label of the desired volume is USER, the volume logical name DISK$USER can be used. This logical name is created by the DCL command MOUNT.
Step 2: If the desired volume is not mounted, mount it and return to step 1.	See Module 4 of this course for instructions on how to mount a volume.
Step 3: $ SET DEFAULT volume-log-name	This command selects the desired volume for the command in step 4.
Step 4: $ CREATE/DIRECTORY - $_/OWNER=[uic] [ufd-name]	The UIC supplied with the /OWNER qualifier should be the user's own UIC, as recorded in the UAF. A user should always own his own UFD. The /OWNER qualifier requires SYSPRV.

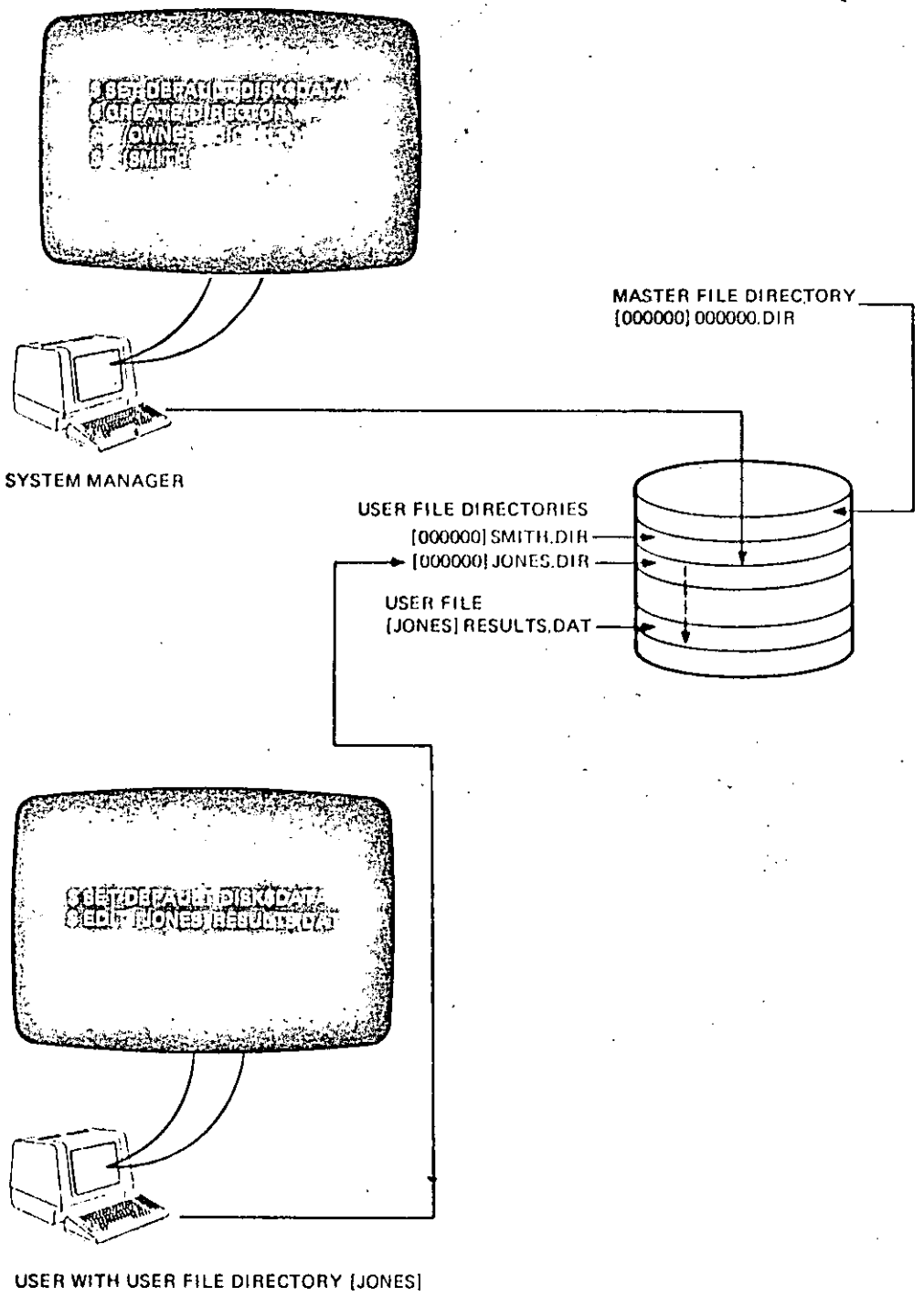


Figure 2-2 User File Directory

Regulating Disk Usage

- Volume quota file [000000]QUOTA.SYS
- One record for each UIC allowed to store files on a volume
- DISKQUOTA utility manages quota file records
- Fields in a quota file record
  - UIC
  - Usage
  - Quota
  - Overdraft

Table 2-13 Fields in a Quota File Record

Field	Meaning	DISKQUOTA Qualifier
UIC	Identifies the user who is permitted to use the volume. Note that files are owned by UICs, not user names. When you log in, VMS uses your UAF record to determine your UIC.	None. Parameter, not qualifier.
Usage	Records the number of blocks of storage owned by this UIC.	None. Maintained by VMS, not entered by system manager.
Quota	Determines the number of blocks of storage this UIC can own before VMS refuses to create new files.	/PERMQUOTA
Overdraft	Determines the number of blocks above the quota this UIC can own before VMS refuses to create new files and extend existing files.	/OVERDRAFT

UIC	Usage	Permanent Quota	Overdraft Limit
[000000,000000]	0	1000	200
[000001,000004]	88535	100000	200
[000011,000140]	6496	10000	200
[000011,000240]	6677	10000	200
[000011,000010]	2846	20000	200
[000011,000030]	3231	10000	200
[000011,000040]	3401	10000	200
[000011,000110]	6837	10000	200
[000011,000120]	3237	10000	200
[000011,000250]	9049	10000	200
[000011,000310]	2051	10000	200
[000011,000320]	8443	10000	200
[000011,000330]	1497	10000	200
[000011,000340]	5087	10000	200
[000011,000220]	10883	20000	200
[000011,000015]	20	500	100
[000041,000020]	47	1000	200
[000051,000001]	60	1000	200

Example 2-4 Sample List of Volume Quota File Records

Running the DISKQUOTA Utility

- The DISKQUOTA utility accesses volume quota file [000000]QUOTA.SYS

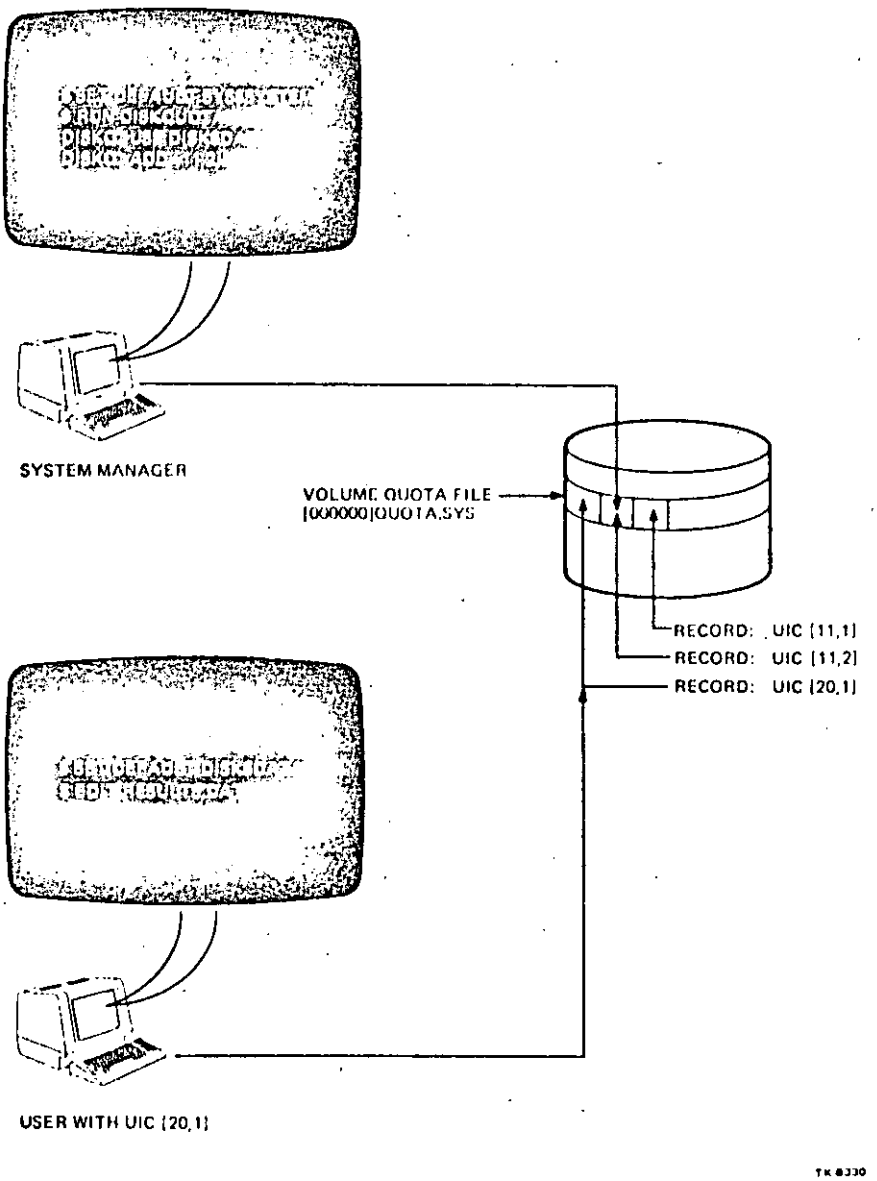


Figure 2-3 Volume Quota File

- DISKQUOTA can:
  - Add UIC records
  - Modify UIC records
  - Remove UIC records
- QUOTA.SYS must be contiguous

Table 2-14 Getting the DISKQUOTA Utility Started

Commands	Comments
Step 1: $ SET UIC [1,4]	This command requires CMKRNL privilege. If the UIC of your interactive process is [1,4], this step is unnecessary.
Step 2: $ RUN SYS$SYSTEM:DISKQUOTA	
Step 3: DISKQ>USE volume	For "volume" you can specify either the device on which the volume is mounted (such as DBA0:) or a logical name for that device (such as DISK$USER).

Table 2-15 Managing Individual Records in the Volume Quota File with the DISKQUOTA Utility

Operation on the Volume Quota File	Command Format
Adding a new record, taking all values not specified by qualifiers from the record for [0,0]	DISKQ>ADD [uic] [/PERMQUOTA=blks1][/OVERDRAFT=blks2]
Modifying an existing record	DISKQ>MODIFY [uic] [PERMQUOTA=blks1][/OVERDRAFT=blks2]
Modifying the record for [0,0], which is used to supply defaults ([0,0] should never own any files.)	DISKQ>MODIFY [0,0] [/PERMQUOTA=blks1][/OVERDRAFT=blks2]
Modifying all records for UICs in a particular group	DISKQ>MODIFY [group-number,*][/PERMQUOTA=blks1][/OVERDRAFT=blks2]
Modifying all records, including the default	DISKQ>MODIFY [*,*][/PERMQUOTA=blks1][/OVERDRAFT=blks2]
Removing an existing record	DISKQ>REMOVE [uic]

- DISKQUOTA can:
  - Create or rebuild the volume quota file
  - Enable or disable disk quotas

Table 2-16 Managing the Volume Quota File as a Whole with the DISKQUOTA Utility

Operation on the Volume Quota File	Command Format	Comment
Establishing quotas on the volume for the first time	DISKQ>CREATE DISKQ>MODIFY [0,0]/PERM=b1/OVER=b2 Establish quota records. See Table 11-15 DISKQ>REBUILD	Creates the file [000000]QUOTA.SYS. Establishes default values for PERMQUOTA and OVERDRAFT. Brings usage records up to date.
Reestablishing quotas on the volume, after they have been disabled for a while	DISKQ>REBUILD	Brings usage records up to date.
Reestablishing quotas on the volume, after they have been disabled for a while, without bringing usage records up to date	DISKQ>ENABLE	Quotas must be enabled to use the commands ADD, MODIFY, and REMOVE. The commands ENABLE, REBUILD, and CREATE enable quotas.
Disabling quotas temporarily	DISKQ>DISABLE	
Disabling quotas and deleting the volume quota file	DISKQ>DISABLE DISKQ>EXIT $ DELETE [000000]QUOTA.SYS;1	Quotas must be disabled to delete the volume quota file.

- DISKQUOTA can:
  - Display UIC records
  - Display HELP

Table 2-17 Displaying the Contents of the Volume Quota File, and Other DISKQUOTA Functions

Operation	Command Format
Displaying the record of a particular user on your terminal	DISKQ>SHOW [uic]
Displaying the records of all users with UICs in a particular group	DISKQ>SHOW [group-number,*]
Displaying the records of all users	DISKQ>SHOW [*,*]
Displaying the DISKQUOTA commands on your terminal	DISKQ>HELP
Exiting from DISKQUOTA	DISKQ>EXIT

- Disk usage displayed by:
  - DISKQUOTA command SHOW
  - DCL command SHOW QUOTA
- Smaller disk usage displayed by:
  - DCL command DIRECTORY/SIZE=ALLOCATED
- QUOTA accounting includes overhead not accounted for by DIRECTORY

ADDING AND REMOVING USERS FROM THE SYSTEM

```
$ SET DEFAULT SYS$SYSTEM
$ RUN DISKQUOTA
DISKQ>USE DISK$USER
DISKQ>ADD [11,15]/PERMQUOTA=500/OVERDRAFT=100
DISKQ>EXIT
$ CREATE/DIRECTORY -
$_/OWNER_UIC=[11,15] DISK$USER:[BROWN]
$ RUN AUTHORIZE
UAF>ADD BROWN/PASSWORD=ENIGMA/UIC=[11,15] -
_/DEVICE=DISK$USER/DIRECTORY=[BROWN] -
_/OWNER="Mary Brown"/ACCOUNT=GRP32
user record successfully added
UAF>EXIT
SYSUAF.DAT;modified
$
```

Example 2-5 Adding a System User

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF>REMOVE BROWN
record removed from SYSUAF.DAT
UAF>EXIT
SYSUAF.DAT modified
$ @DELTREE DISK$USER BROWN
$ RUN DISKQUOTA
DISKQ>USE DISK$USER
DISKQ>REMOVE [11,15]
DISKQ>EXIT
$
```

Example 2-6 Removing a System User

```
$       !Command procedure to delete a UFD and all its SFDs.
$       !Requires SYSPRV privilege or a system UIC.
$       !Back up all useful files before running this procedure.
$       !Use this procedure ONLY on VMS V3.0 and later versions.
$
$       SET NOON
$       IF P1 .EQS. "" THEN INQUIRE P1 "Device, omit colon"
$       IF P2 .EQS. "" THEN INQUIRE P2 "Directory (UFD, omit brackets)"
$       SET PROTECTION=(SY:RWED) 'P1:[''P2...]*.*;*
$       COUNTER = 8
$ LOOP:
$       DELETE/LOG 'P1:[''P2...]*.*;*
$       COUNTER = COUNTER - 1
$       IF COUNTER .GT. 0 THEN GOTO LOOP
$       SET PROTECTION=(SY:RWED) 'P1:[000000]'P2.DIR
$       DELETE/LOG 'P1:[000000]'P2.DIR;1
```

Example 2-7 Command Procedure DELTREE.COM to Delete a UFD and its SFDs

CONTROLLING USER PROCESSES

- Users may require system manager or operator assistance to control their:
  - Processes
  - Files
  - Input/Output
- Most users have sufficient privilege to control their own process
- System manager and operator, with additional privileges, can control any process on the system

Controlling a Process

- Suspending execution
- Resuming execution
- Changing base priority
- Stopping execution and deleting process
- Required privileges to control:
  - Own process and subprocesses - None
  - Processes in same UIC group - GROUP
  - All processes - WORLD

Table 2-18 Controlling Processes

Operation	Command Format and Examples	Comments
Suspending a process	$ SET PROCESS/SUSPEND - $_[IDENTIFICATION=process-id] [process-name] $ SET PROCESS/SUSPEND - $_/IDENTIFICATION=3F0018	Every process has a unique process identification (PID), a hexadecimal number. To refer to a process by PID, as shown in the example, use the /IDENTIFICATION qualifier. The SHOW SYSTEM command displays PIDs.
Resuming a suspended process	$ SET PROCESS/RESUME - $_[IDENTIFICATION=process-id] [process-name] $ SET PROCESS/RESUME - $_IDENTIFICATION=3F0018	
Changing the base priority of a process	$ SET PROCESS/PRIORITY=n - $_[IDENTIFICATION=process-id] - $ SET PROCESS/PRIORITY=3 JONES	Every process has a process name, unique within its UIC group. To refer to a process in your UIC group by process name, supply it as the command parameter. To refer to a process not in your UIC group, you must use its PID, not its process name.
Stopping and deleting a process	$ STOP[/IDENTIFICATION=process-id] - $_[process-name] $ STOP/IDENTIFICATION=3F0018	

Controlling Files and Volumes

- To affect files you do not own, need system UIC or SYSPRV
- To affect volumes you do not own, need VOLPRO

Table 2-19 Controlling Files and Volumes

Operation	Command Format	Comments
Changing the ownership a file	$ SET FILE/OWNER_UIC=[uic] - $_file-spec	This operation requires of SYSPRV privilege or a system UIC.
Establishing automatic housekeeping for a particular file	$ SET FILE/VERSION_LIMIT=n - $_file-spec	VMS automatically purges the lowest numbered file versions to keep the total number of versions less than the version limit.
Disabling special protection on a directory file	$ SET FILE/NODIRECTORY file-spec	This command enables you to delete a corrupted directory file. It is not intended for use on healthy directory files.
Changing the ownership a directory	$ SET DIRECTORY/OWNER_UIC=[uic] - $_[directory-spec]	This operation requires of SYSPRV privilege or a system UIC.
Establishing automatic housekeeping for all files in a directory	$ SET DIRECTORY/VERSION_LIMIT=n - $_[directory-spec]	
Changing the ownership of a volume	$ SET VOLUME/OWNER_UIC=[uic] - $_device-spec	This operation requires VOLPRO privilege.

Controlling Input/Output

- User may request operator assistance
  - Explicity
  - Implicitly
- At some installations, users perform their own operator functions
- User and operator communication performed by OPCOM process
- Operations requiring operator assistance
  - User cannot issue new DCL commands until assistance is provided

Table 2-20 Operations Requiring Operator Assistance

Operator Function	DCL Command Causing Request	Category of Operator Notified	Key for Table 2-22
Mounting a disk or tape volume	MOUNT	DISKS or TAPES	A
Performing any user-requested function	REQUEST/REPLY	All operators; can be more specific by using the qualifier /TO = (OPER [,...]).	B
Mounting a new tape reel for the file system, when it is reading a multi-reel volume set	Any command that reads a file	TAPES	C
Mounting a new tape reel for the file system, when it is writing a multi-reel volume set	Any command that writes a file	TAPES	D

- Typical messages received by operators

```
%OPCOM,  7-MAY-1982 15:17:37.09, request 1, from user FRIEDMAN
__TTA7:, Please connect remote sensor 3A.
```

Example 2-8 Request to Operator by REQUEST/REPLY

```
%OPCOM,  7-MAY-1982 16:54:23.27, request 3, from user FRIEDMAN
Please mount volume DATA15 in device _MTA0:
```

Example 2-9 Request to Operator by MOUNT

```
%OPCOM,  7-MAY-1982 15:22:02.97, message from user FRIEDMAN
__TTA7:, Please check the line printer.
```

Example 2-10 Message to Operator by Request, No Reply Required

- Can respond to user requests only from operator terminals

Table 2-21 Enabling and Disabling Operator Terminals

Operation	Command Format and Examples (Requires OPER Privilege)	Comments
Enabling your terminal to receive and respond to all operator requests	$ REPLY/ENABLE	A terminals is disabled when the operator logs out. The only exception is the console terminal, OPA0:, which is enabled permanently by default when the sytem is initialized.
Disabling your terminal from receiving and responding to any operator requests	$ REPLY/DISABLE	An enabled terminal displays operator requests, but a terminal user needs OPER privilege to respond to these requests.
Enabling your operator terminal selectively	$ REPLY/ENABLE=(oper [,...]) $ REPLY/ENABLE=(DISKS, TAPES)	System generated operator messages are sent only to interested categories of operators. Users can also direct their requests to specific categories. The categories include DISKS, TAPES, PRINTERS, CARDS, and NETWORK.
Disabling your operator terminal selectively	$ REPLY/DISABLE=(oper [,...]) $ REPLY/DISABLE=DISKS	
Listing pending status requests for which your terminal is enabled	$ REPLY/STATUS	

- Responding to requests for operator assistance

Table 2-22 Providing Operator Assistance

Circumstance	Command Format (Requires OPER Privilege)	Type of Request for Which Response is Appropriate (See Table 2-20)
Cannot perform requested function; must abort request	$ REPLY/ABORT=request-id - $_"message-text"	A, B, C, D
May be able to perform requested function; want to notify interactive user but leave request pending	$ REPLY/PENDING=request-id - $_"message-text"	A, B, C, D
Can perform requested function:		
a) User-requested function	$ REPLY/TO=request-id - $_"message-text"	B
b) Mounting specified volume on specified device	Physically mounting the specified volume is the only response required, when the volume is a continuation tape in a multi-reel volume set. In that case:	A
	$ REPLY/TO=request-id - $_"message-text"	C
c) Mounting specified volume on an alternate device	$ REPLY/TO=request-id - $_"SUBSTITUTE alternate-device " - $_"message-text"	A
d) Mounting a scratch tape when the file system is writing a multi-reel volume set	$ REPLY/INITIALIZE_TAPE=request-id - $_"tape-label message-text"	D
e) Mounting a blank tape when the file system is writing a multi-reel volume set	$ REPLY/BLANK_TAPE=request-id - $_"tape-label message-text"	D

- Operator's log file
  - SYS$MANAGER:OPERATOR.LOG
  - Entries made by OPCOM process
  - Logs operator requests and replies
  - Controlled by system manager

Table 2-23 Controlling the Operator's Log

Operation	Command Format	Comments
Closing the operator's log and opening a new one	$ REPLY/LOG	Requires OPER privilege.
Closing the operator's log but not opening a new one	$ REPLY/NOLOG	Requires OPER privilege.
Printing the operator's log file	$ PRINT SYS$MANAGER:- $_OPERATOR.LOG;n	You can print (or type) any version except the current one, which is locked by OPCOM.

- Sending messages to users

Table 2-24 Sending Messages to Users

Operation	Command Format and Examples (Requires OPER Privilege)
Sending a message to all terminals on the system	$ REPLY/ALL "message-text" $ REPLY/ALL/BELL - $_"System down for PM at 2."
Sending a message to all terminals where users are logged on	$ REPLY/USER "message-text"
Sending a message to a specified terminal or terminals	$ REPLY/TERMINAL=(term [,...]) - $_"message-text"

Note: To ring or buzz the terminal receiving the message, use the additional qualifier /BELL.

KEEPING ACCOUNTING RECORDS ON PROCESSES

System Accounting File

- Name: SYS$MANAGER:ACCOUNTNG.DAT
- Contents are used to:
  - Charge for system resources used
  - Analyze system activity and performance
  - Monitor system activity for security reasons
- Can be read by VMS ACCOUNTING utility to produce formatted reports
- Can be read by user written utility
- Input/output control by system manager with DCL commands:
  - SET ACCOUNTING
  - ACCOUNTING

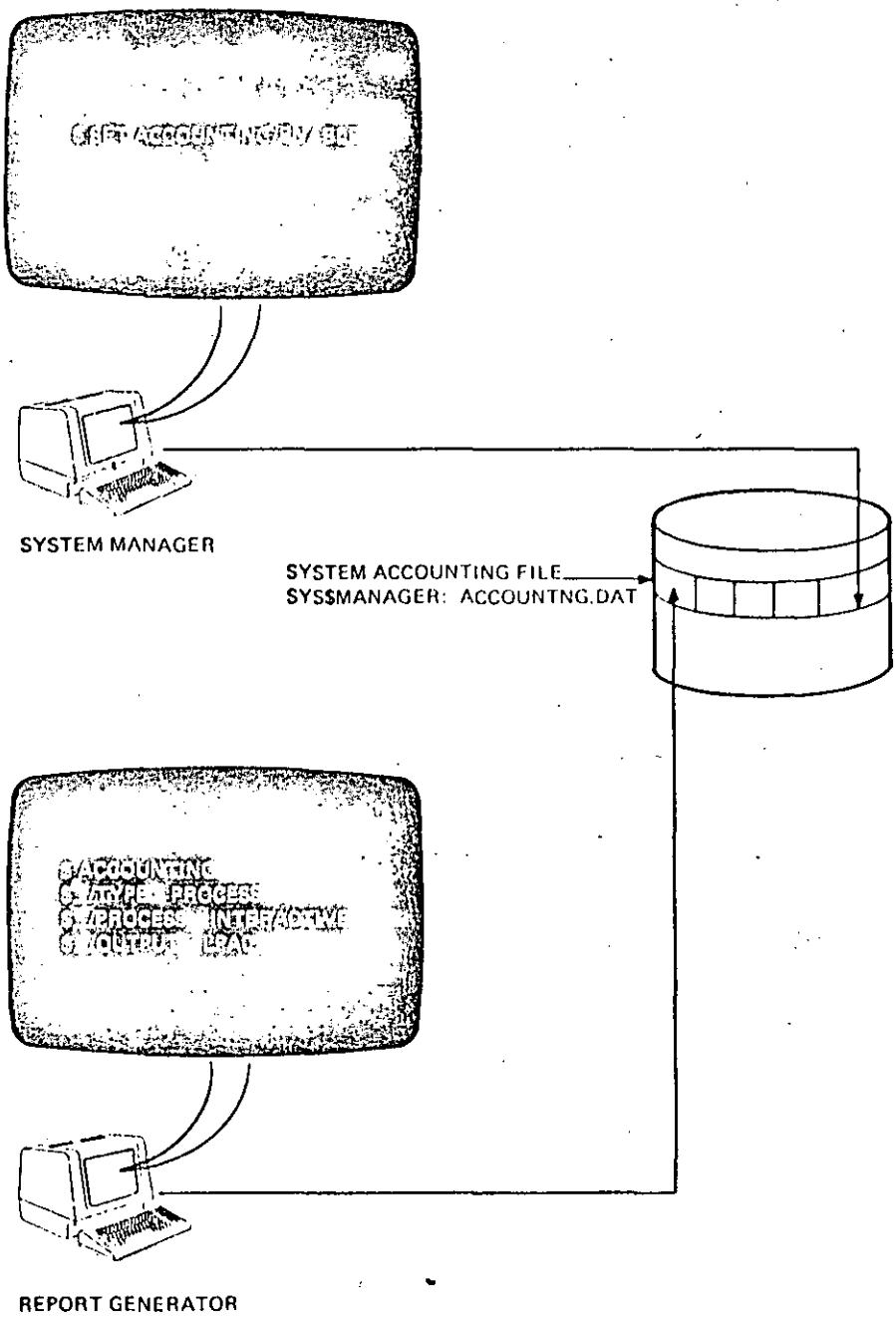


Figure 2-4 System Accounting File

Recording Accounting Information

- JOB_CONTROL writes record to the file when the following events occur:
  - Process deletion
  - Print job completion
  - LOGIN failure
  - Image exit
- Contents of account record
  - System resource usage
  - Identity of resource user

```
INTERACTIVE Process Termination
-------------------------------

Username:           FORGEY              Finish time:       19-JAN-1982 06:41:58.88
Account:            VMS                 Start time:        19-JAN-1982 06:30:45.15
UIC:                [ 11,161]           Elapsed time:                0 00:11:13.73
Process ID:         000F0045            Processor time:              0 00:00:01.89
Owner ID:                               Priority:           4
Terminal name:      RTA1:               Privilege <31-00>:  0010C100
Remote node addr:     224               Privilege <63-32>:  00000000
Remote node name:   AURORA              Queue entry:
Remote ID:          00250039            Queue name:
Final status code:  00000001            Job name:
Final status text:  %SYSTEM-S-NORMAL, normal successful completion

Page faults:              354           Direct IO:                  12
Page fault reads:          13           Buffered IO:                75
Peak working set:         120           Volumes mounted:             0
Peak page file:           185           Images executed:             5
```

Example 2-11 Accounting Report, Full Format

- Recording enabled by default (except image activity)
- SET ACCOUNTING command controls types of resource users about which accounting information is recorded:
  - BATCH
  - INTERACTIVE
  - MESSAGE
  - LOGIN_FAILURE
  - PRINT
  - PROCESS
  - IMAGE

Table 2-25 Recording Accounting Information

Operation	Command Format and Examples (Requires OPER Privilege)
Enabling the recording of all accounting information except image accounting	$ SET ACCOUNTING/ENABLE
Disabling the recording of all accounting information	$ SET ACCOUNTING/DISABLE
Enabling the recording of accounting information selectively	$ SET ACCOUNTING/ENABLE=(record-type [,...]) $ SET ACCOUNTING/ENABLE=(PRINT, LOGIN_FAILURE)
Disabling the recording of accounting information selectively	$ SET ACCOUNTING/DISABLE=(record-type [,...]) $ SET ACCOUNTING/DISABLE=(LOGIN_FAILURE)
Closing the current accounting file and opening a new one	$ SET ACCOUNTING/NEW_FILE

Producing Accounting Reports

- ACCOUNTING utility
  - Reads system accounting file
  - Produces reports
- Qualifiers to DCL command ACCOUNTING control:
  - Which accounting records to analyze
  - How much detail to display
    - Full
    - Brief
    - Summary
  - Order in which records are displayed
- Full format accounting report example generated by:

```
$ ACCOUNTING -
$_/FULL -
$_/TYPE=PROCESS -
$_/PROCESS=INTERACTIVE
```

- Brief format accounting report example generated by:

```
$ ACCOUNTING -
$_/SINCE=19-JAN-1982:6:00 -
$_/BEFORE=19-JAN-1982:8:00
```

Date / Time	Type	Subtype	Username	ID	Source	Status
19-JAN-1982 06:34:53	FILE			00000000		00000000
19-JAN-1982 06:36:35	PROCESS	INTERACTIVE	JEFFREYS	00010034	TTD2:	00000001
19-JAN-1982 06:41:17	PROCESS	INTERACTIVE	FORGEY	000E0040	TTC6:	00000001
19-JAN-1982 06:41:28	LOGFAIL		FORGEY	000F0040	TTC6:	00000000
19-JAN-1982 06:41:34	LOGFAIL		FORGEY	00100040	TTC6:	00000000
19-JAN-1982 06:41:58	PROCESS	INTERACTIVE	FORGEY	000F0045	AURORA	00000001
19-JAN-1982 06:42:05	PROCESS	INTERACTIVE	FORGEY	00110040	TTC6:	00000001
19-JAN-1982 06:42:35	PROCESS	NETWORK	NETNONPRIV	00120040	AURORA	00000001
19-JAN-1982 06:43:15	PROCESS	NETWORK	NETNONPRIV	0022003F	AURORA	00000001
19-JAN-1982 06:43:17	PROCESS	NETWORK	NETNONPRIV	00130040	KOBAL	00000001
19-JAN-1982 06:43:26	PROCESS	NETWORK	NETNONPRIV	00140040	AURORA	00000001
19-JAN-1982 06:43:34	PROCESS	NETWORK	NETNONPRIV	00150040	AURORA	00000001
19-JAN-1982 06:44:06	PROCESS	NETWORK	NETNONPRIV	0023003F	AURORA	00000001
19-JAN-1982 06:46:49	PROCESS	INTERACTIVE	STOREY	00160040	KOBAL	00000001
19-JAN-1982 06:55:07	PROCESS	SUBPROCESS	FORGEY	00170040		00000001
19-JAN-1982 06:55:22	PRINT		FORGEY	00170040		00040001
19-JAN-1982 07:16:59	PROCESS	BATCH	FORGEY	0024003F		100311F0
19-JAN-1982 07:21:41	PROCESS	BATCH	FORGEY	0025003F		10000001
19-JAN-1982 07:24:36	PROCESS	INTERACTIVE	STOREY	00180040	KOBAL	00000001
19-JAN-1982 07:27:57	PROCESS	BATCH	FORGEY	0026003F		10000001
19-JAN-1982 07:38:14	PROCESS	INTERACTIVE	HOBBS	00190040	DELPHI	00000001
19-JAN-1982 07:46:34	PROCESS	NETWORK	NETNONPRIV	001A0040	ALIEN	00000001
19-JAN-1982 07:47:30	PROCESS	NETWORK	NETNONPRIV	001A003E	STAR	00000001
19-JAN-1982 07:51:42	PROCESS	INTERACTIVE	DUMONT	0027003F	TTG0:	00000001
19-JAN-1982 07:54:39	PROCESS	NETWORK	NETNONPRIV	002B003F	CHIPS	100020E4
19-JAN-1982 07:56:42	PROCESS	NETWORK	NETNONPRIV	001B003E	CHIPS	00000001
19-JAN-1982 07:57:26	PROCESS	BATCH	FORGEY	001B0040		10000001
19-JAN-1982 07:59:34	PROCESS	NETWORK	NETNONPRIV	0029003F	CHIPS	00000001
19-JAN-1982 07:59:48	PROCESS	NETWORK	NETNONPRIV	002A003F	CHIPS	00000001

Example 2-12 Accounting Report, Brief Format

- Summary format accounting report example generated by:

```
$ ACCOUNTING -
$_/SINCE=19-JAN-1982:6:00 -
$_/BEFORE=19-JAN-1982:8:00 -
$_/TYPE=PROCESS -
$_/SUMMARY=(HOUR,USER) -
$_/REPORT=BUFFERED_IO
```

```
From: 19-JAN-1982 06:00                          To: 19-JAN-1982 08:00
HH Username       Buffered
                     I/O
-------------------------
06 FORGEY             1668
06 JEFFREYS           5834
06 NETNONPRIV          818
06 STOREY              187
07 DUMONT              115
07 FORGEY             1172
07 HOBBS               112
07 NETNONPRIV          470
07 STOREY              459
```

Example 2-13 Accounting Report, Summary Format

- Steps in creating an accounting report

Table 2-26 Creating an Accounting Report

Operation	Qualifier of ACCOUNTING Command	Comment
Step 1: Selecting an accounting file	Parameter, not qualifier	By default, SYS$MANAGER:ACCOUNTNG.DAT
Step 2: Selecting type of record to analyze for the report	/TYPE	By default, all types.
Step 3: Selecting records to analyze, based on contents of specific fields in the records	Many qualifiers. For example, /USER, /UIC, /ACCOUNT, /NODE	Individual fields may be present in some record types and not in others.
Step 4: Sorting the selected records	/SORT	Affects display order of full and brief formats.
Step 5: Choosing format of report:		
a) For each record, every field is displayed, one record per terminal screen or listing page.	/FULL	See Example 2-11.
b) For each record, selected fields are displayed, one record per line.	default (brief)	See Example 2-12.
c) The contents of a single resource field is summarized as a function of specified identifying fields.	/SUMMARY /REPORT	See Example 2-13.

- Using command files containing qualifiers to create accounting reports

  - Full command string

```
$ ACCOUNTING -
$_/SINCE=19-JAN-1982:6:00 -
$_/BEFORE=19-JAN-1982:8:00 -
$_/TYPE=PROCESS -
$_/SUMMARY=(HOUR,USER) -
$_/REPORT=BUFFERED_IO
```

  - Abbreviated command string

```
$ ACCOUNTING -
$_/SINCE=19-JAN-1982:6:00 -
$_/BEFORE=19-JAN-1982:8:00@BUFFSUM
```

  - Contents of command file BUFFSUM.COM

```
/TYPE=PROCESS/SUMMARY=(HOUR,USER)/REPORT=BUFFERED_IO
```

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" Y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO, D.F.

MANAGING QUEUES

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985

INTRODUCTION

When a user issues the PRINT command to print a file, all the system printers may already be in use. Because this condition often occurs, VMS maintains a list of all print requests in the order in which they occur. This ordered list is called a print queue, and the requests are called print jobs. The PRINT command places the user's job in the queue. When the job moves up to the front of the queue, VMS passes it to the first available printer, which prints the file.

Similarly, when a user issues the SUBMIT command, system resources may not be sufficient to support the immediate creation of a batch process to execute the submitted command procedure. To handle this condition, VMS maintains a batch queue. The SUBMIT command places the user's request, called a batch job, in the queue. When the batch job moves up to the front of the queue, and when the number of batch processes on the system becomes small enough, VMS creates a batch process to execute the job.

For each VMS system, batch and print queues are customized by the system manager. By matching your system queue structure to the uses and resources of the system, you can improve system performance.

Usually, your customized queue structure is created when you start up the system. While the system is running, you may also find it necessary to manage the system queues. For example, during times of heavy system usage, you may want to reduce batch job activity to improve system interactive response.

System operators often share your queue management responsibilities. For example, if the paper jams in a line printer, or if a printer runs out of paper, it may be the job of an operator to return the printer to service.

OBJECTIVES

A system manager must:

- Select and install batch and print queues
- Handle malfunctioning queues and jobs

RESOURCES

1. VAX/VMS Command Language User's Guide
2. VAX/VMS System Management and Operations Guide

PRINTING A FILE

PRINT Command in DCL

- Issuing the PRINT command

```
$ PRINT MEMO.TXT
  Job 2863 entered on queue SYSSPRINT
$
```

Example 3-1 Issuing the PRINT Command

- Job numbers indicate the order in the queue
- SYS$PRINT is the standard system print queue
- Job is printed by the first available printer

Using a Particular Printer

- Ways to print jobs and display status

Table 3-1 Queuing a Print Job

Operation	Command Format and Example	Comments
Printing a job consisting of a single file	$ PRINT file-spec $ PRINT PROG	The PRINT command uses a default file type of LIS. In this example, the file printed is PROG.LIS.
Printing a job consisting of more than one file	$ PRINT file-spec [,...] $ PRINT MEMO.TXT, PROG.LIS	The files MEMO.TXT and PROG.LIS form a single print job.
Printing a job on a specified printer	$ PRINT/QUEUE=queue-name - $_file-spec [,...] $ PRINT/QUEUE=LPA0 MEMO.TXT	You can use either the qualifier /QUEUE or the qualifier /DEVICE. They are equivalent.
Displaying a list print jobs	$ SHOW QUEUE/DEVICES $ SHOW QUEUE/DEVICES/ALL $ SHOW QUEUE/DEVICES/FULL $ SHOW QUEUE/DEVICES/FULL/ALL	By default, the only jobs of displayed other than your own are those currently being printed. To display everyone's jobs, add the qualifier /ALL. For more information on each job, add the qualifier /FULL.
Displaying a list of print jobs on a particular queue	$ SHOW QUEUE queue-name $ SHOW QUEUE SYS$PRINT $ SHOW QUEUE/ALL queue-name $ SHOW QUEUE/ALL SYS$PRINT	In any SHOW QUEUE command, you can specify a particular queue name instead of using /DEVICES. You can also use the qualifiers /FULL and /ALL.

- Can print a job on a specific printer
- Physical queue associated with each printer
- Generic queue
  - SYS$PRINT - Standard system print queue
  - Jobs transferred to any physical queue with generic printing enabled
- Generic queue/physical queue
  - Current job (physical queues only)
  - Pending jobs

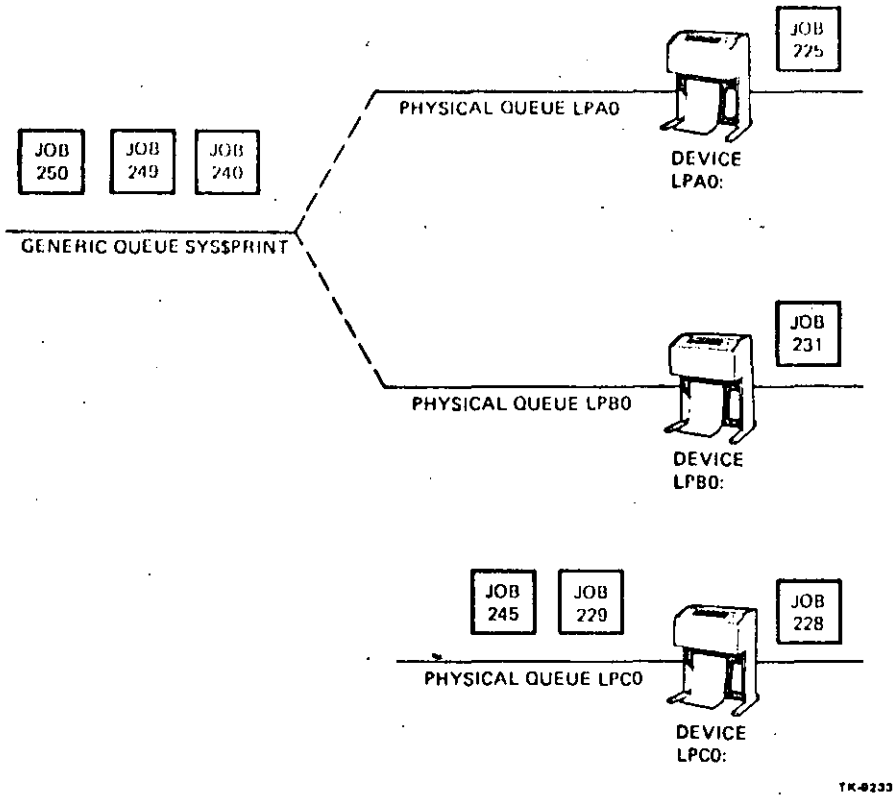


Figure 3-1 Physical and Generic Print Queues

- Queue status list

  $ SHOW QUEUE/DEVICES/ALL

  - Job status
  - Job number
  - User name
  - Job name
  - Job priority
  - Date and time job was queued
  - Job size in blocks of 512 bytes

```
* Generic Device queue "SYS$PRINT" Flag

 PEND    240 KRYCKA          DAPWORK    Pri=4 19-OCT-1981 11:02 Size=85
 PEND    249 KENAH           FORKCNTRL  Pri=4 19-OCT-1981 11:05 Size=59
 PEND    250 LIPMAN          DVIDEF     Pri=4 19-OCT-1981 11:06 Size=4

* Device queue "LPA0" Forms=0, Genprt Lower Flag

 CURR    225 HALVORSEN       RMSOLSTCH  Pri=4 19-OCT-1981 10:50 Size=265

* Device queue "LPB0" Forms=0, Genprt Lower Flag

 CURR    231 KENAH           SYSCREPRC  Pri=4 19-OCT-1981 11:00 Size=268

* Device queue "LPC0" Forms=0, Nogen Lower

 CURR    228 MALIK           RFATEST    Pri=4 19-OCT-1981 10:56 Size=5
 PEND    229 DAVIS           MAIL       Pri=4 19-OCT-1981 10:57 Size=9
 PEND    245 GOLDSTEIN       MAIL       Pri=4 19-OCT-1981 11:04 Size=20
```

Example 3-2 Queue Status Display Corresponding to Figure 3-1

Referring to a Particular Printer by Function

- Assigned
  - Logical queue assigned to a physical queue
- Reassigned
  - Logical queue can be reassigned to a different physical queue
- Not assigned at all
  - Logical queue does not have to be assigned to a physical queue at all
  - Holding jobs
  - For long print jobs
  - For special print jobs

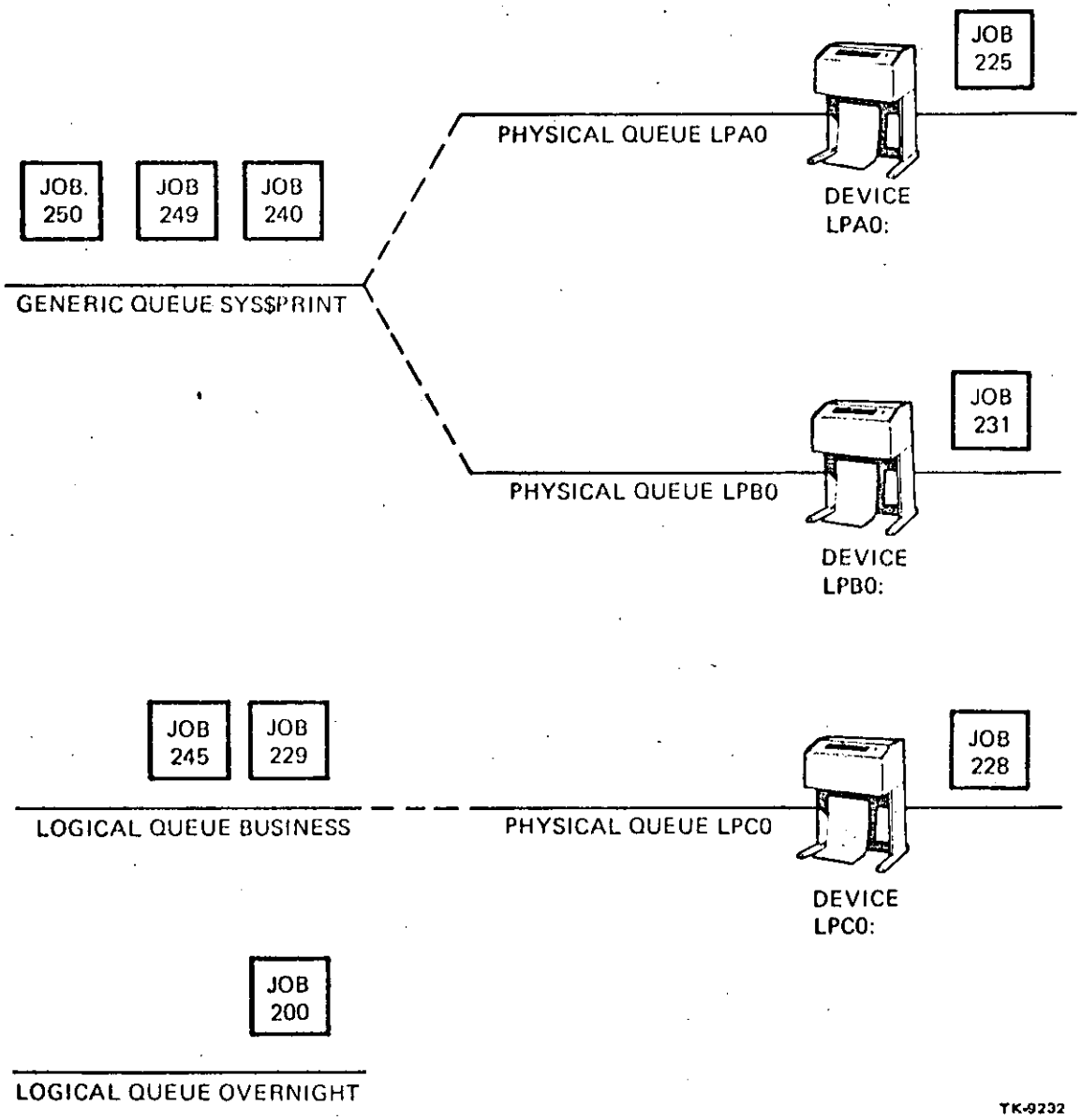


Figure 3-2 Physical, Generic, and Logical Print Queues

```
* Generic Device queue "SYSSPRINT" Flag

 PEND   240 KRYCKA        DAPWORK   Pri=4 19-OCT-1981 11:02 Size=85
 PEND   249 KENAH         FORKCNTRL Pri=4 19-OCT-1981 11:05 Size=59
 PEND   250 LIPMAN        DVIDEF    Pri=4 19-OCT-1981 11:06 Size=4

* Device queue "LPA0" Forms=0, Genprt Lower Flag

 CURR   225 HALVORSEN     RMSOLSTCH Pri=4 19-OCT-1981 10:50 Size=265

* Device queue "LPB0" Forms=0, Genprt Lower Flag

 CURR   231 KENAH         SYSCREPRC Pri=4 19-OCT-1981 11:00 Size=268

* Device queue "LPC0" Forms=0, Nogen Lower

 CURR   228 MALIK         RFATEST   Pri=4 19-OCT-1981 10:56 Size=5

* Device queue "BUSINESS" Forms=0, Nogen

        This queue assigned to LPC0
 PEND   229 DAVIS         MAIL      Pri=4 19-OCT-1981 10:57 Size=9
 PEND   245 GOLDSTEIN     MAIL      Pri=4 19-OCT-1981 11:04 Size=20

* Device queue "OVERNIGHT" Forms=0, Nogen Stopped Flag

 HOLD   200 RAPPAPORT     DUDRIVER  Pri=4 19-OCT-1981 10:15 Size=1300
```

Example 3-3 Queue Status Display Corresponding to Figure 3-2

Assigning Priority to Print Jobs

- VMS selects a current job based on:
  - Time the job was queued
  - Priority, an integer between 0 and 31
- Highest priority jobs printed first
- For same priority jobs, the first queued is selected
- By default, job priority is the same as base priority of process queueing job

Table 3-2 Specifying the Priority of a Print Job

Operation	Command Format and Example	Comments
Printing a job with default priority	$ PRINT file-spec $ PRINT PROG	By default, VMS uses the base priority of the process that queues a print job as the print job priority. Because the base priority of most processes is 4, the priority of most print jobs is 4.
Printing a job with specified priority	$ PRINT/PRIORITY=n file-spec $ PRINT/PRIORITY=2 PROG	OPER or ALTPRI privilege is a required to specify a print job priority higher than process base priority.

- Process priority/print job priority
  - Both take the values 0 - 31
  - Both usually 4
  - Process priority - For scheduling CPU
  - Print job priority - Determines order in queue

Specifying the Characteristics of Print Jobs

- Number of copies
- How many header pages precede each job
- Whether each page has a header line
- Single spacing or double spacing
- Whether top and bottom page margins are provided
- Whether a special printer setup is required

Table 3-3 Setting the Characteristics of a Print Job

Characteristic	PRINT Qualifier Format and Example	Comments
Number of copies	/COPIES=number $ PRINT/COPIES=2 MEMO.TXT /JOB_COUNT=number $ PRINT/JOB_COUNT=3 - $_MEMO.TXT/COPIES=2, - $_MEMO1	/COPIES is a file qualifier. It can be different for each file in a job, or it can be the same for all. /JOB_COUNT is a command qualifier. It specifies how many times to duplicate the entire job.
Header pages and lines	/[NO]FLAG_PAGE $ PRINT/NOFLAG PROG.FOR /[NO]BURST $ PRINT/FLAG/BURST PROG.FOR /[NO]HEADER $ PRINT/HEADER PROG.FOR	See Figure 3-3 for an explanation of flag and burst pages and of header lines. The default values of /FLAG and /BURST are set by the system manager. For header lines, the default is /NOHEADER.
Spacing between lines and pages	/SPACE $ PRINT MEMO.TXT/SPACE /[NO]FEED $ PRINT PROG/NOFEED	By default, files are printed single spaced; with the /SPACE qualifier, double spaced. By default, the printer advances (feeds) to a new page without printing on the bottom four lines of the preceding page.
Paper and ribbon installed in printer	/FORMS=type $ PRINT/FORMS=3 INVOICES $ PRINT/FORMS=CARBON - $_INVOICES	As an alternative to logical queues, forms type can be used to indicate a job that requires a special printer setup. The system manager assigns a number or name (such as REDINK or CARBON) to each printer setup allowed. A job queued with a special forms type will be held on the queue until the system manager sets up the printer to match. By default, forms type is 0 or NORMAL.

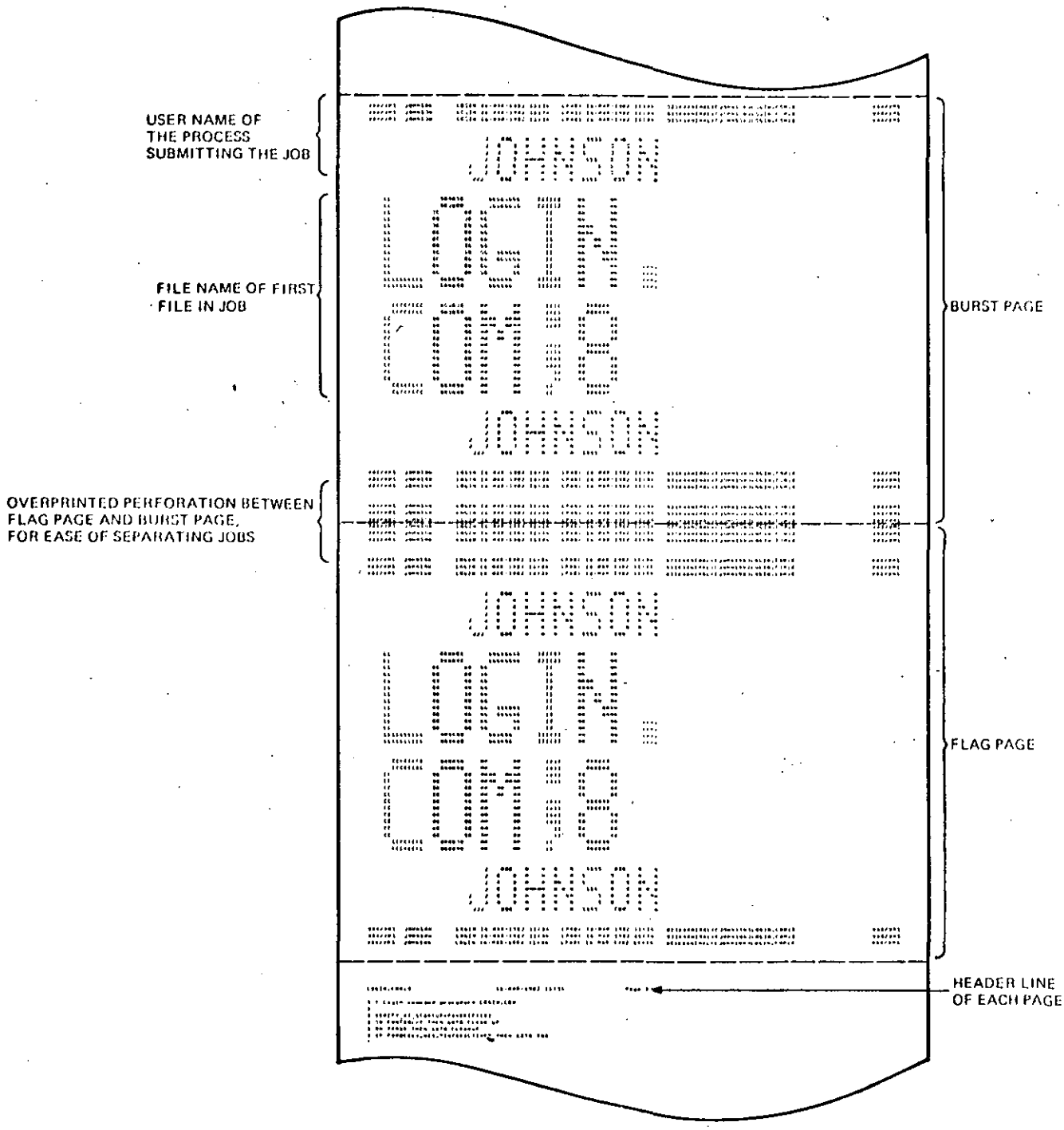


Figure 3-3 Print Job Header Pages and Header Lines

- Queue status list

  $ SHOW QUEUE/DEVICES/FULL/ALL

```
* Generic Device queue "SYS$PRINT"  Flag

 PEND    240 KRYCKA        DAPWORK    Pri=4 19-OCT-1981 11:02 Size=85
         WRKD$:[KRYCKA.MEM]DAPWORK.TXT;1 Copies=1, Flag
         WRKD$:[KRYCKA.MEM]DAPLIST.TXT;1 Copies=1,
 PEND    249 KENAH         FORKCNTRL Pri=4 19-OCT-1981 11:05 Size=59
         RESD$:[SYS.LIS]FORKCNTRL.LIS;1 Copies=1, Flag
 PEND    250 LIPMAN        DVIDEF     Pri=4 19-OCT-1981 11:06 Size=4
         WRKD$:[LIPMAN.SRC]DVIDEF.MDL;7 Copies=1, Flag

* Device queue "LPA0" Forms=0, Genprt Lower Flag

 CURR    225 HALVORSEN     RMSOLSTCH Pri=4 19-OCT-1981 10:50 Size=265

* Device queue "LPB0" Forms=0, Genprt Lower Flag

 CURR    231 KENAH         SYSCREPRC Pri=4 19-OCT-1981 11:00 Size=268

* Device queue "LPC0" Forms=0, Nogen Lower

 CURR    228 MALIK         RFATEST   Pri=4 19-OCT-1981 10:56 Size=5

* Device queue "BUSINESS" Forms=0, Nogen

         This queue assigned to LPC0
 PEND    229 DAVIS         MAIL      Pri=4 19-OCT-1981 10:57 Size=9
         WRKD$:[]MAIL.LIS; Copies=1, Delete
 PEND    245 GOLDSTEIN     MAIL      Pri=4 19-OCT-1981 11:04 Size=20
         WRKD$:[]MAIL.LIS; Copies=1, Delete

* Device queue "OVERNIGHT" Forms=0, Nogen Stopped Flag

 HOLD    200 RAPPAPORT     DUDRIVER  Pri=4 19-OCT-1981 10:15 Size=1300
         WRKD$:[RAPPAPORT.LIS.CLASS]DUDRIVER.LIS;1 Copies=1, Hurst Flag
```

Example 3-4 Full Format Queue Status Display Corresponding to Figure 3-2

SUBMITTING A BATCH JOB

SUBMIT Command in DCL

```
$ SUBMIT ACTION.COM
  Job 2864 entered on queue SYS$BATCH
$
```

Example 3-5 Issuing the SUBMIT Command

- Job numbers indicate the order in the queue
- SYS$BATCH is the standard system batch queue
- Each job in the queue consists of a command procedure
- VMS creates a batch process to execute the command procedure

Using a Particular Batch Queue

- Ways to queue batch jobs and display status
- Can pass parameters using /PARAMETERS qualifier

Table 3-4 Submitting a Batch Job

Operation	Command Format and Example	Comments
Submitting a job with no parameters	$ SUBMIT file-spec $ SUBMIT ACTION	The SUBMIT command uses a default file type of COM. In this example, the file that is submitted is ACTION.COM.
Submitting a job that requires parameters	$ SUBMIT/PARAMETERS=(P1,[,...]) - $_file-spec $ SUBMIT/PARAMETERS=(3,SUM) - $_MATH	You can specify up to 8 parameters.
Submitting a job to a specified queue	$ SUBMIT/QUEUE=queue-name - $_file-spec $ SUBMIT/QUEUE=SLOWBATCH - $_ACTION	By default, the system batch queue SYS$BATCH is used.
Displaying a list of batch jobs	$ SHOW QUEUE/BATCH $ SHOW QUEUE/BATCH/ALL $ SHOW QUEUE/BATCH/FULL $ SHOW QUEUE/BATCH/FULL/ALL	By default, the only jobs displayed other than your own are those currently being executed. To display everyone's jobs, add the qualifier /ALL. For more information on each job, add the qualifier /FULL.
Displaying a list of batch jobs on a particular queue	$ SHOW QUEUE queue-name $ SHOW QUEUE SYS$BATCH $ SHOW QUEUE/ALL queue-name $ SHOW QUEUE/ALL SYS$BATCH	In any SHOW QUEUE command, you can specify a queue name instead of using /BATCH. You can also use the qualifiers /FULL and /ALL.

- SYS$BATCH
  - The standard system batch queue
  - Job limit of the batch queue
- Additional batch queues
  - For batch jobs with special system resource requirements
- Status of jobs on queues
  - Current jobs
  - Pending jobs
  - Holding jobs

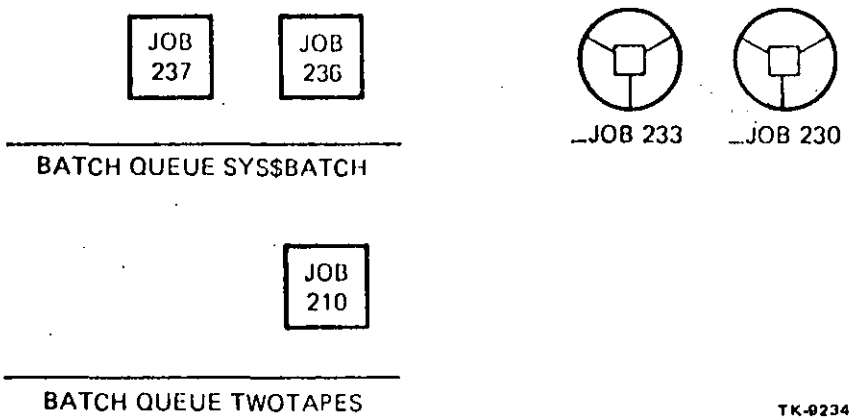


Figure 3-4 Batch Queues

- Queue status list

  $ SHOW QUEUE/BATCH/FULL/ALL

```
* Batch queue "SYS$BATCH" Joblim=2, Basepri=4, Swap

    WS default= None          WS quota= None          WS extent= None
  Def CPU time= None                                Max CPU time= None

 CURR    230 MUIZNIEKS     LISTS      Pri=4 19-OCT-1981 10:59
   WS default= None  WS quota= None  WS extent= None  CPU time= None
 CURR    233 FOWLER        COMPI      Pri=4 19-OCT-1981 11:00
   WS default= None  WS quota= None  WS extent= None  CPU time= None
 PEND    236 MUIZNIEKS     TEST       Pri=4 19-OCT-1981 11:01
   WS default= None  WS quota= None  WS extent= None  CPU time= None
     WRKD$:[MUIZNIEKS.LEVEL1]TEST.COM;1
 PEND    237 LIONEL        COPY       Pri=4 19-OCT-1981 11:02
   WS default= None  WS quota= None  WS extent= None  CPU time= None
     WRKD$:[LIONEL.MISC]COPY.COM;1

* Batch queue "TWOTAPES" Joblim=1, Basepri=4, Swap Stopped

    WS default= None          WS quota= None          WS extent= None
  Def CPU time= None                                Max CPU time= None

 HOLD    210 MORSE         CHECKIN    Pri=4 19-OCT-1981 10:20
   WS default= None  WS quota= None  WS extent= None  CPU time= None
     WRKD$:[MORSE]CHECKTEMP.COM;3  Delete
```

Example 3-6 Full Format Queue Status Display Corresponding to Figure 3-4

How a Batch Job Executes

- Batch job's relationship to the job that submitted it
  - Runs independently
  - Uses same UAF characteristics
  - Executes same LOGIN.COM
- Batch job's SYS$OUTPUT assigned to batch log file
  - Created in login default directory
  - File name same as name of batch command procedure file
  - File type LOG
- Logical name assignments for batch processes

Table 3-5 Logical Name Definitions for Interactive and Batch Processes

Logical Name	Definition When Interactive Process Begins to Execute	Definition When Batch Process Begins to Execute
SYS$INPUT	interactive terminal	batch command file
SYS$OUTPUT	interactive terminal	batch log file
SYS$COMMAND	interactive terminal	batch command file
SYS$ERROR	interactive terminal	batch log file

- Control of batch log file
  - Name
  - Printing on successful completion of job
  - Deletion after printing

Table 3-6 Controlling the Batch Log File

Characteristic	SUBMIT Qualifier Format and Example	Comments
Name of batch log file	/[NO]LOG_FILE=file-spec $ SUBMIT/LOG_FILE=OUTPUT - $_ACTION	By default, the log file has the same file name as the command file, and file type LOG. In this example, the log file is named OUTPUT.LOG instead of the default ACTION.LOG.
Printing of batch log file	/[NO]PRINTER=queue-name $ SUBMIT/PRINTER=LPC0 - $_ACTION	By default, the log file is queued to SYS$PRINT when the batch job finishes executing. In this example, the log file is queued to LPC0 instead of the default SYS$PRINT.
Saving of batch log file	/[NO]KEEP $ SUBMIT/KEEP ACTION	By default, the log file is deleted after it is printed. In this example, the log file is retained in the login directory. The qualifier /NOPRINT implies the qualifier /KEEP.

Assigning Priority to Batch Jobs

- VMS selects a current job based on
  - Time the job was queued
  - Priority, an integer between 0 and 31
- Highest priority jobs executed first
- For same priority jobs, the first queued is selected
- By default, job priority is the same as base priority of process queueing job

Table 3-7 Specifying the Priority of a Batch Job

Operation	Command Format and Example	Comments
Submitting a job with default priority	$ SUBMIT file-spec $ SUBMIT MAKER	By default, VMS uses the base priority of the process that queues a batch job as the batch job priority. Because the base priority of most processes is 4, the priority of most batch jobs is 4. Note that the base priority at which the batch process executes is determined by the system manager. It is not affected by batch job priority, which only controls access to the front of the batch queue.
Submitting a job with a specified priority	$ SUBMIT/PRIORITY=n file-spec $ SUBMIT/PRIORITY=2 MAKER	OPER or ALTPRI privilege is required to specify a batch job priority higher than process base priority.

- Process priority/batch job priority
  - Both take the values Ø - 31
  - Both usually 4
  - Batch job priority determines order in queue
  - Once batch job executes, its process priority is determined by the system manager
- Uses for alternate batch queues
  - System manager can create special queues for jobs requiring:

    Different working set size from UAF values

    Different CPU time from UAF values

Writing a Batch Command Procedure

- Two ways to run a command procedure
  - Interactive
  - Batch
- For command procedure run in batch, consider:
  - Batch process executes LOGIN.COM first
  - By default, error or severe error terminates batch job execution
  - Batch process's default directory is the one specified in UAF
  - Verification is on in batch process by default

```
$       ! LOGIN.COM
$
$       ! The following command procedure defines logical
$       ! names and symbols for processes created by USER.
$       ! These definitions are required only if this is
$       ! an interactive job.
$
$ IF F$MODE() .NES. 'INTERACTIVE' THEN EXIT
$!FORTEST.COM
$
$! FORTEST.COM COMPILES, LINKS, AND
$! RUNS THE PROGRAM, TEST.FOR
$
$ FORTRAN TEST
$ LINK TEST
$ RUN TEST
          1
          2
          3
          4
          5
$ EXIT
  USER         Job terminated at  4-MAY-1982 13:49:06.49

  Accounting information:
  Buffered I/O count:            71      Peak working set size:    200
  Direct I/O count:              41      Peak page file size:      425
  Page faults:                 1092      Mounted volumes:            0
  Elapsed CPU time:     0 00:00:03.22    Elapsed time:     0 00:00:07.15
```

Example 3-7 Sample Batch Log File

HANDLING BATCH AND PRINT JOBS

Characteristics common to both batch and print jobs:

- The name used to identify the job
- The VAX node on which the job is processed
- Whether the job number is displayed when the job is queued
- Whether the user is notified when the job completes
- Whether the input file is deleted after the job completes

Table 3-8 Specifying the Characteristics of Batch and Print Jobs

Characteristic	PRINT or SUBMIT Qualifier Format and Example	Comments
Name of job	/NAME=job-name $ SUBMIT/NAME=MYJOB - $_ACTION $ PRINT/NAME=SAVETHIS $_MEMO.TXT	By default, VMS uses the file name of the first file in the job. The job name appears on the flag and burst pages of print jobs, as the default file name of the batch log, and in the queue status list.
Notification that job has been queued	/[NO]IDENTIFY $ PRINT/NOIDENTIFY PROG.FOR	By default, the user receives a message like the ones in Examples 3-1 and 3-5.
Notification that job processing is complete	/[NO]NOTIFY $ SUBMIT/NOTIFY MATH	By default, the user receives no notification.
Deletion of queued file after it is processed	/[NO]DELETE $ PRINT/DELETE TEMP	By default, queued files are not deleted. Note that using the /DELETE qualifier is not equivalent to issuing a DELETE command just after the file is queued. If you delete the file as soon as you queue it, VMS will not find the file when the job is ready to become current.
Processing of job on another VAX node	/REMOTE $ SUBMIT/REMOTE - $_VAX2::[BUSINESS]PAYROLL	The queued file must already be on the remote system where it is to be processed. No qualifiers besides /REMOTE are allowed.

Modifying a Print or Batch Job

- Change job characteristics after a job is queued (non-current only)
- Move a job to another queue
- Delete a job from a queue (current and non-current)
- Privileges required to affect jobs
  - Queued by you - None
  - Queued by processes in your UIC group - GROUP
  - Queued by anyone - WORLD or OPER

Table 3-9 Modifying a Batch or Print Job

Operation	Command Format and Example	Comments
Changing the characteristics of a job	$ SET QUEUE/ENTRY=job-number - $_/qualifier [/...] queue-name $ SET QUEUE/ENTRY=113 - $_/PRIORITY=3/JOB_COUNT=2 - $_LPA0	The job number for the /ENTRY qualifier can be obtained from the queue status list. If a job is current, its characteristics cannot be changed.
Moving a job to another queue	$ SET QUEUE/ENTRY=job-number - $_/REQUEUE=queue-name - $_[/qualifier...] queue-name $ SET QUEUE/ENTRY=256 - $_/REQUEUE=TWOTAPE SYS$BATCH	
Deleting a job from a queue	$ DELETE/ENTRY=job-number - $_queue-name $ DELETE/ENTRY=125 SYS$BATCH	A job on a queue can be deleted, including a current job. If you delete a current job, its processing stops immediately.

Delaying the Processing of a Job

- A job can be queued but processing is delayed until:
  - A particular time arrives
  - A command is given to release it

Table 3-10 Specifying When a Job is Released for Processing

Operation	Command Format and Example	Comments
Scheduling a job for processing at a specific time	$ PRINT/AFTER=time - $_file-spec [,...] $ SUBMIT/AFTER=time - $_file-spec [,...] $ SET QUEUE/ENTRY=job-number - $_/AFTER=time queue-name $ PRINT/AFTER=21:30 - $_SYMBOLS.LIS	VMS releases the job for processing at the specified time. It is processed some time after that. The time can be expressed absolutely, relative to the current time, or relative to an absolute time.
Holding a job on the queue until someone explicitly releases it	$ PRINT/HOLD file-spec [,...] $ SUBMIT/HOLD file-spec [,...] $ SET QUEUE/ENTRY=job-number - $_/HOLD queue-name $ SET QUEUE/ENTRY=321 - $_/HOLD SYS$BATCH	VMS waits for the SET QUEUE/RELEASE command to release the job for processing.
Releasing a held job	$ SET QUEUE/ENTRY=job-number - $_/RELEASE queue-name $ SET QUEUE/ENTRY=321 - $_/RELEASE SYS$BATCH	

- Status of jobs

Table 3-11 Job States

State and Abbreviation	Meaning for Print Jobs	Meaning for Batch Jobs
current CURR	Being printed	Being executed
pending PEND	Waiting for a printer to become available	Waiting for VMS to create a batch process
after AFTER	Waiting for specified time to be released for printing	Waiting for specified time to be released for execution
holding HOLD	Waiting for job to be explicitly released, waiting for queue to be started, or waiting on physical queue for the printer to have the required setup	Waiting for job to be explicitly released, or waiting for queue to be started

HOW VMS HANDLES BATCH AND PRINT QUEUES

- JOB_CONTROL process creates and deletes necessary processes
- Processes created by JOB_CONTROL
  - Print symbionts

    One for each active physical print queue
  - Batch processes

    One for each current batch job

    Deleted when batch job finishes executing
  - Input symbionts

    One for each card reader performing batch input

    Copies the deck to a command file in the default directory

    This command file is then submitted to a batch queue

    Requirements for batch job submitted on cards

    First DCL command is JOB command

    Next DCL command is PASSWORD command

    Last DCL command is EOJ command
  - Interactive processes

    One for each user login

```
 VAX/VMS V1A5    Processes on  29-OCT-1981 09:35:30.95    Uptime  1 04:51:22
  Pid     Process Name     UIC  State Pri Dir. I/O    CPU      Page flts Ph.Mem
00010000 NULL          000,000 COM    0        0 12:21:50.17         0    0
00010001 SWAPPER       000,000 HIB   16        0 00:00:38.12         0    0
00020018 _TTD4:        211,002 LEF    7       67 00:00:02.28       178  130
00020019 _JOB3874      010,040 COM    3        1 00:00:00.02        16   40 B
0003001C HUNT          360,005 LEF    7       63 00:00:06.68       988   84
0007001D _TTG5:        211,002 COM    4       58 00:03:34.83       186  100
000A0020 SYSTEM        001,004 LEF    9       27 00:00:01.54       231   67
00010023 SIGMON        360,025 LEF    4       53 00:00:03.27       361   66
00060026 _TTE1:        350,000 LEF    7      133 00:00:03.11       467  150
0002002D J_OSBORN      360,013 LEF    7       61 00:00:22.46      1812  123
0003002E MELVIN        360,011 LEF    4       58 00:00:03.85      1206   70
000B0033 _TTJ2:        211,002 LEF    8       40 00:00:01.40       176  131
000A0038 ST_GEORGE     320,001 LEF    8      597 00:00:19.07      2548   60
0004003A LPERRON       303,007 LEF    4      790 00:00:17.31      1154   79
00070040 YERASKA       360,016 LEF    9       15 00:00:01.04       142   87
000B0042 HARP          360,023 LEF    8      443 00:01:13.37      2908  106
003C0045 DORSEY        360,024 LEF    4       17 00:00:01.17       199   72
002A0046 FRIEDMAN      360,002 CUR    4      469 00:02:05.44      3322  119
003E0047 FIELD         001,010 LEF    4       13 00:00:01.45       188   64
00030048 REMACP        001,003 HIB    9        1 00:02:10.50        32   26
00020049 NETACP        001,004 HIB   10    13406 00:07:45.67     32799  171
000B004A ZKINFO        350,000 LEF    7       44 00:00:01.25       327  150
002B004B MCGRATH       360,017 LEF    5       20 00:00:01.46       219   87
0001004C PRTSYMB2      001,004 HIB    8       28 00:00:03.71        25   44 S
0001004D PRTSYMB1      001,004 HIB    8       46 00:00:08.70        17   45 S
0001004E ERRFMT        001,006 HIB    8      197 00:00:02.47        30   32
00010050 OPCOM         001,004 LEF    8       52 00:00:00.27        34   42
00010051 JOB_CONTROL   001,004 HIB    8     1034 00:00:13.83       115  100
00010052 DBA0ACP       001,003 HIB    7     9645 00:05:28.19      6297  267
00020053 DRC2ACP       001,003 HIB    7      922 00:00:21.14       242  231
```

Example 3-8 JOB_CONTROL and Related Processes

Print Spooling

- Permits alternate user and program access to printer served by queue
- If spooling not enabled, only access to printer is through PRINT command
- If spooling is enabled, other I/O methods can be used
- When printer I/O occurs, VMS writes output to a file on disk
- When file is closed, VMS queues file to print queue
- Transparent to user
- Should be enabled on all printers served by queues

SETTING UP BATCH AND PRINT QUEUES

- OPER privilege required in managing queues
- DCL commands used in managing queues

Table 3-12 Managing Queues

Operation	Command Format
Setting the characteristics of a line printer	$ SET PRINTER/qualifiers device-name
Setting the characteristics of a terminal used as a printer	$ SET TERMINAL/PERMANENT - $_/NOBROADCAST/NOTYPE_AHEAD/NOTTSYNC - $_/other-qualifiers device-name
Enabling spooling on a printer	$ SET DEVICE/SPOOLED device-name
Creating a print queue	$ INITIALIZE/QUEUE queue-name
Creating a batch queue	$ INITIALIZE/QUEUE/BATCH queue-name
Starting the processing of jobs on a queue	$ START/QUEUE queue-name
Stopping the processing of jobs on a queue	$ STOP/QUEUE queue-name
Deleting a queue	$ DELETE/QUEUE queue-name
Assigning a logical print queue to a physical print queue	$ ASSIGN/QUEUE physical-queue logical-queue
Cancelling a logical print queue assignment	$ DEASSIGN/QUEUE logical-queue

Queue Commands Executed at System Start-Up

- Included in SYS$MANAGER:SYSTARTUP.COM

```
$!Start a physical queue for each line printer.
$!Enable generic printing on LPA0 and LPB0.
$!Disable generic printing on LPC0.
$
$       SET PRINTER/LOWER LPA0:
$       SET DEVICE/SPOOLED LPA0:
$       START/QUEUE LPA0        !If this START command succeeds,
$       !$STATUS equals 1, and the following branch to LPA0_DONE
$       !is taken.  Otherwise, the branch is not taken,
$       !and the queue is initialized, then started.
$       IF $STATUS THEN GOTO LPA0_DONE
$       INITIALIZE/QUEUE/FLAG LPA0
$       START/QUEUE LPA0
$LPA0_DONE:
$       SET PRINTER/LOWER LPB0:
$       SET DEVICE/SPOOLED LPB0:
$       START/QUEUE LPB0
$       IF $STATUS THEN GOTO LPB0_DONE
$       INITIALIZE/QUEUE/FLAG LPB0
$       START/QUEUE LPB0
$LPB0_DONE:
$       SET PRINTER/LOWER LPC0:
$       SET DEVICE/SPOOLED LPC0:
$       START/QUEUE LPC0
$       IF $STATUS THEN GOTO LPC0_DONE
$       INITIALIZE/QUEUE/NOENABLE_GENERIC_PRINTING LPC0
$       START/QUEUE LPC0
$LPC0_DONE:
$
$!Start the generic print queue SYS$PRINT.
$
$       START/QUEUE SYS$PRINT
$       IF $STATUS THEN GOTO SYS$PRINT_DONE
$       INITIALIZE/QUEUE/GENERIC/FLAG SYS$PRINT
$       START/QUEUE SYS$PRINT
$SYS$PRINT_DONE:
$
$!Start the logical queue BUSINESS and assign it to the
$!physical queue LPC0:.
$
$       START/QUEUE BUSINESS
$       IF $STATUS THEN GOTO BUSINESS_DONE
$       INITIALIZE/QUEUE/NOENABLE_GENERIC_PRINTING BUSINESS
$       ASSIGN/QUEUE LPC0 BUSINESS
$       START/QUEUE BUSINESS
$BUSINESS_DONE:
$
$!Make sure the logical queue OVERNIGHT exists.
$!Do not start it.
$
$       !The following command is present only to find out if the
$       !logical queue exists.
$       STOP/QUEUE OVERNIGHT
$       !The following command checks for the "no such queue" error
$       !status.  Any other status indicates that the queue exists.
$       IF ($STATUS .NE. %X4803A) THEN GOTO OVERNIGHT_DONE
$       INITIALIZE/QUEUE/NOENABLE_GENERIC_PRINTING OVERNIGHT
$OVERNIGHT_DONE:
```

Example 3-9 Sample Start-Up Commands Corresponding to Figure 3-2

- Typical qualifiers used in initializing print queues

Table 3-13 Characteristics of Print Queues

Characteristic of Print Queue	Qualifiers of INITIALIZE and START Commands	Comments
Flag page precedes each job	/FLAG	A flag page is a header page that makes it easy to separate one job from another.
Flag and burst pages precede each job	/FLAG/BURST	A burst page is a second flag page. Two header pages make it even easier to separate one job from another.
Special printer setup	/FORMS_TYPE=type	Forms type can be used to identify color of ink, width of paper, design of forms, or any other characteristic of printer setup.

```
$!Start the batch queue SYS$BATCH
$
$       START/QUEUE SYS$BATCH   !If this START command succeeds,
$       !$STATUS equals 1, and the following branch to SYS$BATCH_DONE
$       !is taken.  Otherwise, the branch is not taken,
$       !and the queue is initialized, then started.
$       IF $STATUS THEN GOTO SYS$BATCH_DONE
$       INITIALIZE/QUEUE/BATCH/JOB_LIMIT=2  SYS$BATCH
$       START/QUEUE SYS$BATCH
$SYS$BATCH_DONE:
$
$!Make sure the batch queue TWOTAPES exists.
$!Do not start it.
$
$       !The following command is present only to find out if the
$       !logical queue exists.
$       STOP/QUEUE TWOTAPES
$       !The following command checks for the "no such queue" error
$       !status.  Any other status indicates that the queue exists.
$       IF ($STATUS .NE. %X4803A) THEN GOTO TWOTAPES_DONE
$       INITIALIZE/QUEUE/BATCH TWOTAPES
$TWOTAPES_DONE:
```

Example 3-10 Sample Start-Up Commands Corresponding to Figure 3-4

- Typical qualifiers used in initializing batch queues

Table 3-14 Characteristics of Batch Queues

Characteristic of Batch Queue	Qualifiers of INITIALIZE and START Commands
Number of current jobs queue can create	/JOB_LIMIT=number
Size of the working set of each current batch job	/WSDEFAULT=number /WSQUOTA=number /WSEXTENT=number
Total CPU time for each current batch queue	/CPUDEFAULT=time /CPUMAXIMUM=time
Ability of VMS to swap a current batch job	/[NO]DISABLE_SWAPPING

Rules for Establishing Queues

- For every physical printer
  - Set the physical characteristics of the device

    Line printer:

    ```
    $ SET PRINTER/qualifiers device-name
    ```

    Printing terminal:

    ```
    $ SET TERMINAL/PERMANENT -
    $_/NOBROADCAST -
    $_/NOTYPEAHEAD -
    $_/NOTTSYN -
    $_/other-qualifiers device-name
    ```

  - Set the device spooled

    ```
    $ SET DEVICE/SPOOLED device-name
    ```

  - Initialize and start a physical queue for the device

    ```
    $ START/QUEUE device-name
    $ IF $STATUS THEN GOTO label
    $ INITIALIZE/QUEUE/qualifiers device-name
    $ START/QUEUE device-name
    $ label:
    ```

    If you do not want to enable generic printing on the queue, use the INITIALIZE qualifier:

    ```
    /NOENABLE_GENERIC_PRINTING
    ```

- Generic line printer queue

  ```
  $ START/QUEUE SYS$PRINT
  $ IF $STATUS THEN GOTO label
  $ INITIALIZE/QUEUE/GENERIC/qualifiers SYS$PRINT
  $ START/QUEUE SYS$PRINT
  $ label:
  ```

- Generic terminal queue (if desired)

```
$ START/QUEUE queue-name
$ IF $STATUS THEN GOTO label
$ INITIALIZE/QUEUE/GENERIC/TERMINAL/qualifiers -
$_queue-name
$ START/QUEUE queue-name
$ label:
```

- Logical queues (See Example 3-9)

- Forms type (SYS$MANAGER:FORMSTYPE.DAT)

```
%forms-name forms-number comment
%NORMAL Ø   This forms name is standard.
```

- Default batch queue

```
$ START/QUEUE SYS$BATCH
$ IF $STATUS THEN GOTO label
$ INITIALIZE/QUEUE/BATCH/qualifiers SYS$BATCH
$ START/QUEUE SYS$BATCH
$ label:
```

- Other batch queues

MANAGING QUEUES ON A RUNNING SYSTEM

- OPER privilege required in managing queues
- Typical items to manage
  - Line printer needs paper
  - Line printer breaks
  - Batch job uses too much of system resources

Managing Print Queues

- Logical and generic print queues
  - No current job

Table 3-15 Stopping Logical and Generic Print Queues

Operation on a Logical or Generic Queue	Command Format
Stopping the queue and restarting it with new characteristics	$ STOP/QUEUE queue-name $ START/QUEUE[/qualifier...] queue-name
Stopping the queue and deleting it	$ STOP/QUEUE queue-name $ DELETE/QUEUE queue-name

- Physical print queues
  - Pause for printer repair
  - Must specify what happens to current job

Table 3-16 Pausing Physical Print Queues to Start a New Box of Printer Paper or Repair Jammed Paper

Operation on a Physical Queue	Command Format
Pausing a queue, halting the current job	$ STOP/QUEUE physical-queue
Resuming a paused queue, with the following effect on the current job:	
Continuing with same page of current job	$ START/QUEUE physical-queue
Continuing with preceding page of current job	$ START/QUEUE/BACKSPACE physical-queue
Restarting the current job from the beginning	$ START/QUEUE/TOP_OF_FILE physical-queue
Aborting the current job	$ START/QUEUE/NEXT physical-queue

- Physical print queues
  - Stop to change characteristics of queue or delete queue
  - Must specify what happens to current job

Table 3-17 Stopping Physical Print Queues to Change the Queue Characteristics or Delete the Queue

Steps in Changing the Characteristics of or Deleting a Physical Print Queue	Command Format
Step 1: Stopping the queue	$ STOP/QUEUE/NEXT physical-queue
Step 2: Specifying the effect on the current job	
Allowing current job to complete	--no command --
Aborting current job	$ STOP/QUEUE/ABORT physical-queue
Moving current job to the end of the queue	$ STOP/QUEUE/REQUEUE physical-queue
Moving current job to specified queue with specified priority	$ STOP/QUEUE/REQUEUE=queue-name - $_/PRIORITY=n physical-queue
Step 3: Modifying the queue	
Restarting the queue with new characteristics	$ START/QUEUE[/qualifier...] physical-queue
Deleting the queue, along with all the jobs on the queue	$ DELETE/QUEUE physical-queue

- Reassigning logical queues

Table 3-18 Reassigning Logical Queues

Operation on a Logical Queue	Command Format
Changing the assignment of the queue	$ STOP/QUEUE logical-queue $ ASSIGN/QUEUE other-queue logical-queue $ START/QUEUE logical-queue
Removing the assignment of the queue	$ STOP/QUEUE logical-queue $ DEASSIGN/QUEUE logical-queue

- Moving jobs from one queue to another

Table 3-19 Moving Jobs From One Queue to Another

Operation	Command Format	Example: LPA0 Needs Repairs, LPB0 is Operating
Moving all jobs (except current jobs) from one queue to another	$ ASSIGN/MERGE - $_destination-queue source-queue	$ STOP/QUEUE/NEXT LPA0 $ STOP/QUEUE/REQUEUE LPA0 $ ASSIGN/MERGE - $_LPB0 LPA0 $ DELETE/QUEUE LPA0

Managing Batch Jobs

- Interrupting and restarting
  - Can NOT be interrupted and restarted automatically
  - Must be started again from the beginning or restarted manually
- Restarting manually
  - Examine the batch log
  - Edit the batch command procedure
  - Submit the edited command procedure as a batch job
- Managing batch queues

Table 3-20 Stopping Batch Queues

Operation on a Batch Queue	Command Format
Stopping the queue and restarting it with new characteristics	$ STOP/QUEUE batch-queue $ START/QUEUE[/qualifier...] batch-queue
Stopping the queue and deleting it	$ STOP/QUEUE batch-queue Wait for all current jobs to complete, or abort current jobs with DELETE/ENTRY command. $ DELETE/QUEUE batch-queue

Cautions on Command Names

- $ STOP/QUEUE
  - Stops batch queue
  - Stops generic or logical print queue
  - Does NOT stop physical print queue (must use STOP/QUEUE/NEXT)
  - Pauses physical print queue
- $ STOP/QUEUE/REQUEUE and STOP/QUEUE/ABORT
  - Does NOT stop physical print queue
  - Does NOT pause physical print queue
  - Affects the current job, NOT the queue
- $ ASSIGN/MERGE
  - Does NOT assign a logical queue
  - Moves jobs from one queue to another

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" Y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO,D.F.

MANAGING DISK AND TAPE VOLUMES

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985

INTRODUCTION

The system manager must coordinate disk and tape management with the system's users and operators. The system manager is responsible for deciding how the user data disks are initialized, mounted, backed up, and, when necessary, verified. Although the system manager establishes the guidelines, the operators usually perform the actual operations.

Data stored on mass-storage volumes must be organized so it can be retrieved quickly, and protected from access by unauthorized users. For this purpose, disks used on a VAX/VMS system are formatted according to the FILES-11 on-disk structure, and magnetic tapes comply with the ANSI Level 3 tape file standard.

The appropriate file structure is given to a disk or tape volume at initialization time by an ancillary control process (ACP). The ACP is responsible for maintaining the structure of and controlling access to the volume. It is the ACP which makes entries in directories, creates and deletes files and keeps track of storage allocation of the blocks on a disk.

This module is separated into two parts: The first part covers how disk and tape volumes are managed; the second part covers how they are structured.

OBJECTIVES

A system manager must:

- Assist users with their private volumes
- Create and manage public volumes
- Describe the FILES-11 structure of disk volumes
- Describe the VAX/VMS ANSI Level 3 structure of tape volumes

RESOURCES

To complete this module you should have access to the following documents:

1. VAX/VMS Utilities Reference Manual
2. VAX/VMS Magnetic Tape User's Guide
3. VAX/VMS System Management and Operations Guide

DISK AND TAPE VOLUMES

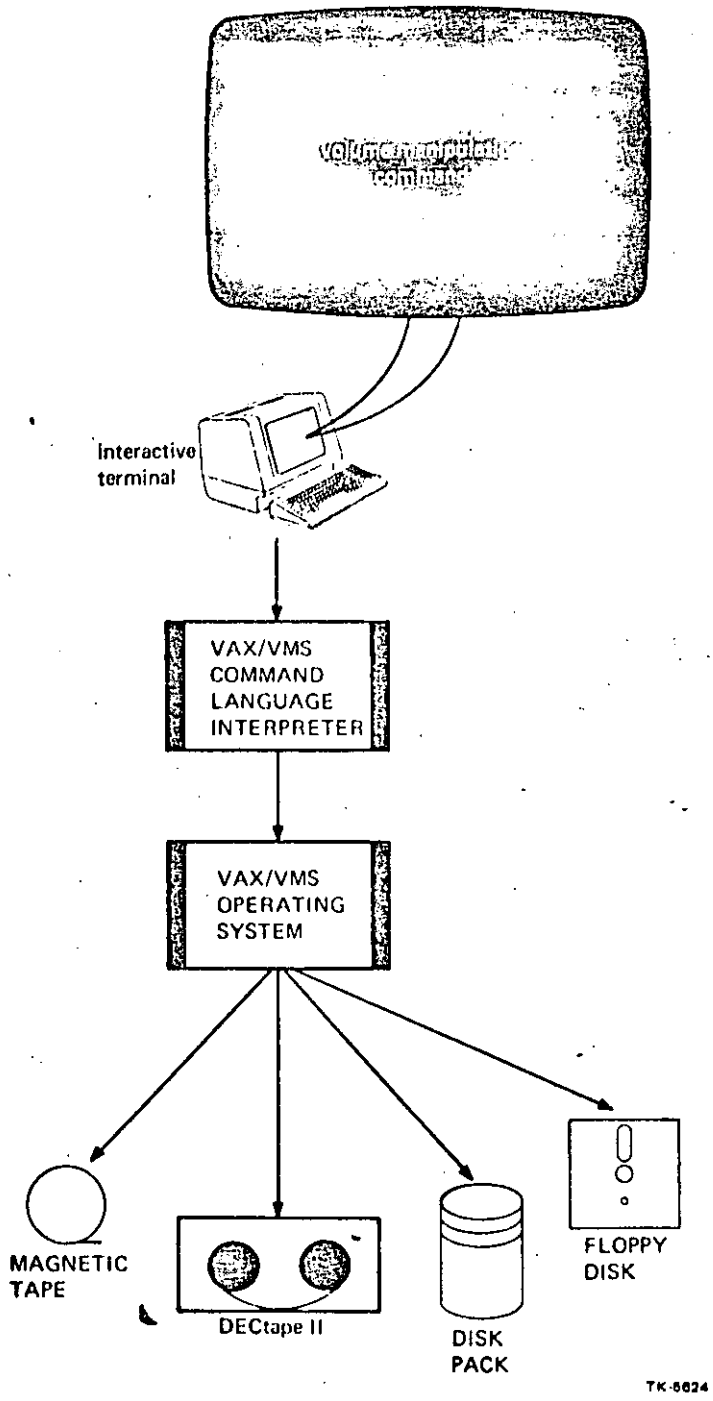


Figure 4-1 Volume Manipulation Commands

TYPES OF DISK AND TAPE VOLUMES

Private Volumes

- Reasons for using private disks and tapes
  - File backup
  - Private environment (protection)
  - More space (quotas)
  - Ease file transfer among systems

Public Volumes

- Reasons for use
  - Contain files users create for private or public use
  - Contain public files
    - Operating system
    - Utilities
    - Diagnostic and test programs
    - System libraries
    - HELP files
    - Optional software
- Available to all users
- Access to files controlled by protection

USING DISK AND TAPE VOLUMES

- To copy information to or from a volume, use:
  - COPY
  - BACKUP
  - FLX
  - BCK
  - RST

Major Steps in Creating and Using a Disk or Tape Volume

- $ALLOCATE device-name[:] [logical-name[:]]
- [$INITIALIZE[/qualifiers] device-name[:] label]
- $MOUNT[/qualifiers] device-name[:] [label] [log-name[:]]

  .
  .
  .
  use device - often via logical name
  .
  .
  .
- $DISMOUNT[/qualifiers] device-name[:]
- $DEALLOCATE device-name[:]

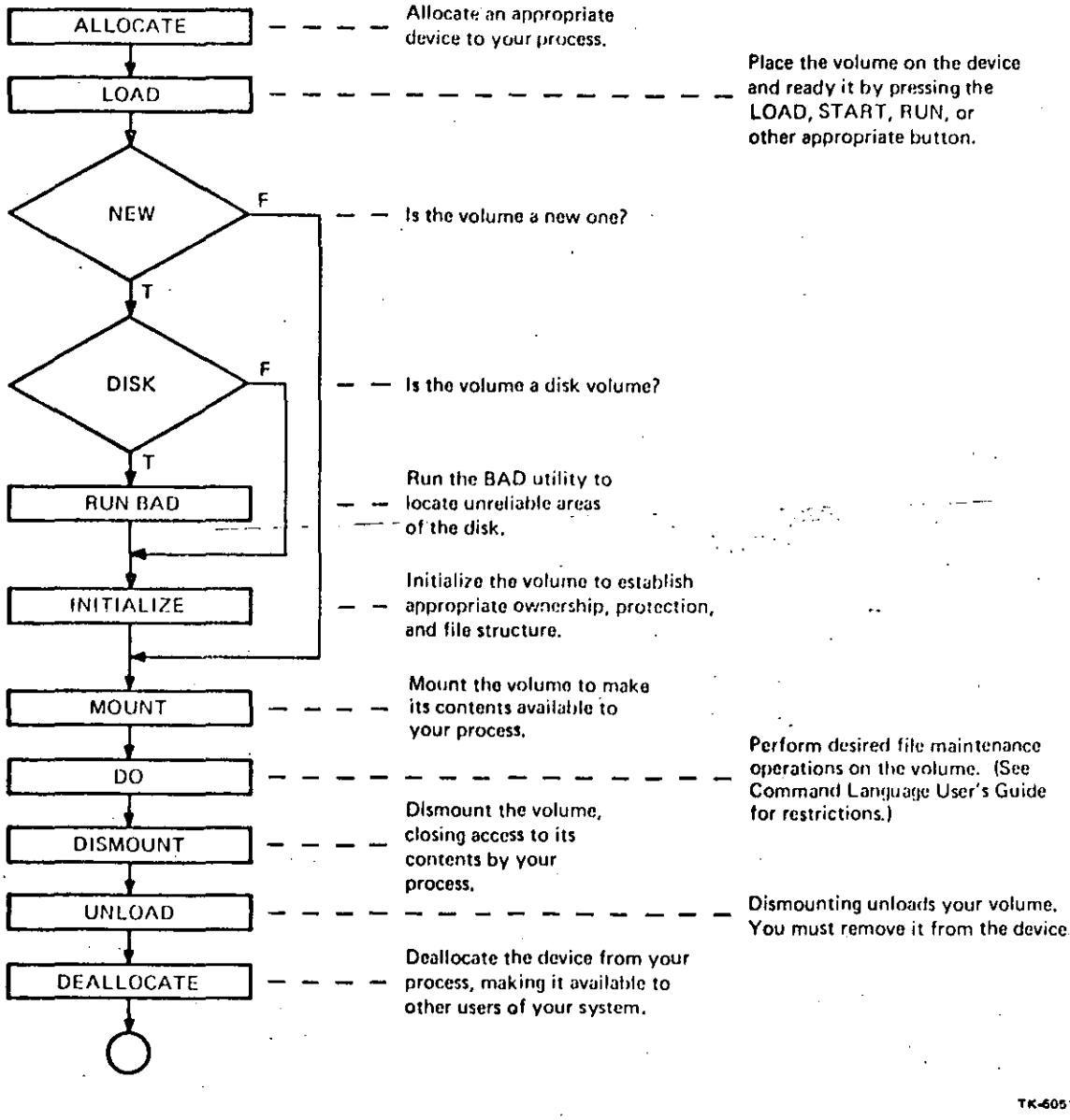


Figure 4-2 Creating and Using a Disk or Tape Volume

Table 4-1 Obtaining Information About Devices

Operation	Command String	Comments
Displaying the characteristics of a device	$ SHOW DEVICE[/FULL] device	
	$ SHOW DEVICE/FULL DMA2:	Displays the characteristics of device DMA2.
Displaying the characteristics of a volume	$ SHOW DEVICE/FULL device	
	$ SHOW DEVICE/FULL MYDISK	Displays the characteristics of the volume currently mounted on the device whose logical name is MYDISK.

```
$ SHOW DEVICE DY:
  List of Devices                on      25-MAR-1982 15:50:07.35
  Device    Device      Device       Err.    Volume         Free  Trans Mount
  Name      Status  Characteristics Count     Label        Blocks Count Count
  DYA0:     on line                   0
  DYA1:     on line                   0
$ ALLOCATE DY: DISK
  _DYA0: allocated
$ MOUNT/FOREIGN DISK
%MOUNT-I-MOUNTED, mounted on _DYA0:
$ RUN SYS$SYSTEM:BAD
BAD>DY0:
BAD -- DY0: TOTAL BAD BLOCKS= 0.
BAD>^Z
$ DISMOUNT/NOUNLOAD DISK
$ INITIALIZE DISK USER_DISK
$ MOUNT DISK USER_DISK
%MOUNT-I-MOUNTED, USER_DISK    mounted on _DYA0:
$ CREATE/DIRECTORY DISK:[HELP]
$ DIRECTORY DISK:[HELP]
  No files found.
$ COPY SYS$HELP:NOTES.HLB DISK:[HELP]
$ DIRECTORY DISK:[HELP]

Directory _DYA0:[HELP]

NOTES.HLB;1

Total of 1 file.
$ SHOW DEVICE/FULL DISK


  Device   DYA0:          25-MAR-1982 15:51:32.21

       on line
       Mounted
       Error Logging Enabled
       Allocated

    Error count                    0    Owner process id          0011003A
    Operations completed         530    Owner process name        DORSEY
    Reference count                1    Default buffer size           512

    Volume label       USER_DISK        Free blocks                   440
    Owner UIC          [011,030]        Transaction count               1
    Volume protection  0000             Mount count                     1
    Volume status                       Relative volume no.             0
    ACP process name   DYA0ACP          Cluster size                    1
    Quota cache size           64       Max. files allowed            123
    File ID cache size         32       Extent cache size              64
$ DISMOUNT DISK
$ DEALLOCATE DISK
```

Example 4-1 Preparing and Transferring Files to a Disk Volume

Table 4-2 Preparing Disk and Tape Volumes for Use

Operation	Command String/Example	Comments
Allocating a device	$ ALLOCATE device [log-name]	
	$ ALLOCATE DM DISK	Finds first available RK06/RK07 and assigns it to your process. The logical name DISK is placed in your process logical name table and assigned the name of the allocated device. Other users are unable to access the device. When the ALLOCATE command is used, the DISMOUNT command does not free the device for other users.
Finding unreliable areas on an unused disk	$ MOUNT/FOREIGN device $ RUN SYS$SYSTEM:BAD BAD>device/LI . . . BAD><CTRL/Z> $ DISMOUNT/NOUNLOAD device	Finds unreliable areas of the disk and records their locations in a special block. The process obliterates all previous data on the disk. If the disk is not your own, VOLPRO privilege is required.
Creating a file structure on a tape or disk	$ INITIALIZE device label	
	$ INITIALIZE DMA2: TEST_DISK	Builds FILES-11 On-Disk-Structure on the volume loaded on DMA2:. The volume is labeled TEST_DISK. You are declared owner of the disk; all user groups are allowed all types of access (RWED). You must own the disk, or process VOLPRO privilege.
	$ INITIALIZE MTA0: TSTTAP	Builds ANSI level 3 tape structure on the volume located on device MTA0. The volume receives the label TSTTAP. By default, you are declared the owner; all user groups are allowed all types of access (RWED).

Table 4-3 Sequence of Commands for Using Disk and Tape Volumes

Function	Action of Command/Example	Comments
Create a link between your process and a or tape device and specify the volume to be placed on that device	$ MOUNT device_name - $_volume-label logical-name $ MOUNT MT: XFILES TAPE	Issuing the mount command with a generic device disk specification causes the system to select an available device of the specified type to your process. Once this command is issued, either someone must mount the volume or you must issue <CTRL/C> to cancel the request.
Complete the link between your process the specified volume	Place the volume on the assigned device. Press LOAD, START, RUN or other appropriate button.	This activity is performed by an operator on some systems, by the user on others.
Use the Volume		
Breaking the link between the Volume and your Process	$ DISMOUNT[/NOUNLOAD]device	If you want VMS to keep the order on-line when you dismount it, use the qualifier /NOUNLOAD.
	$ DISMOUNT DMA2:	Dismounts and automatically unloads the volume on DMA2. Deletes the logical name assignment of MOUNT.
Deallocating a Device	$ DEALLOCATE device	
	$ DEALLOCATE DMA2:	Deallocates device DMA2. Frees the device for use by other users. Does not delete a logical name assigned by the ALLOCATE command. If the ALLOCATE command was not used to assign the device to your process, the DISMOUNT command breaks the link between your process and that device, so that it may be assigned to other users.
Physically remove the volume from the device	Remove volume from device and store in an appropriate place.	This activity is performed by the operator in some systems, the user on others.

BACKUP

- Files specified are placed in a save-set
- Format of information in save-set is understood by BACKUP
- A save-set can exist on tape or disk
- Format of the BACKUP command:

  BACKUP/qualifier input-specifier output-specifier

- TAPE must be mounted /FOREIGN
- When used with tape volumes, BACKUP can create and access save-sets only

Table 4-4 Definitions of BACKUP Terms

Term	Definition
SAVE-SET	A file in BACKUP format, created by BACKUP
SAVE-SET NAME	Any legal VMS filename, file type and version number
SAVE-SET SPECIFIER	A device and a save-set name. The format is: device:save-set name

Table 4-5 Commonly Used BACKUP Functions and Commands

Function	Definition
Compare	Examine two input and output specifications for differences in file specifications.
	$ BACKUP/COMPARE save-set-specifier file-specification(s)
	$ BACKUP/COMPARE MTA0:FEB10.BCK [MARSH...]*.*
Copy	Create one or more new disk files from one or more old disk files.
	$ BACKUP file-specification(s) file-specification(s)
	$ BACKUP DBA1:[MARSH...] DBA2:[MARSH...]
Duplicate	Create a duplicate of a disk including location of information.
	$ BACKUP/PHYSICAL input-device output-device
	$ BACKUP/PHYSICAL DBA1: DBA2:
List	Display the file specifications for files inside the save-set.
	$ BACKUP/LIST save-set-specifier
	$ BACKUP/LIST MTA0:FEB10.BCK
Save	Copy one or more files into a save-set. The directory structure is recorded with file information.
	$ BACKUP file-specification(s) save-set-specifier
	$ BACKUP DBA1:[MARSH...]*.* MTA0:FEB16.BCK
Restore	Copy one or more files from a save-set to a disk or floppy. Directories are created if not found.
	$ BACKUP save-set-specifier file-specification(s)
	$ BACKUP MTA0:FEB16.BCK DBA2:[MARSH]

```
$ SHOW DEVICE MT:
  List of Devices                on       5-JUN-1982 13:08:10.71
  Device   Device      Device       Err.    Volume          Free  Trans Mount
  Name     Status  Characteristics Count    Label         Blocks Count Count
  MTA0:    on line                  13
$ ALLOCATE MT: MYTAPE
  _MTA0: allocated
$ INITIALIZE/DENSITY=1600 MYTAPE UTLBCK
$ MOUNT/FOREIGN MYTAPE
%MOUNT-I-MOUNTED, UTLBCK mounted on _MTA0:
$ BACKUP [COURSE.UTLCOM...]*.* MYTAPE:MARCH10.BCK
$ BACKUP/LIST MYTAPE:MARCH10.BCK
Listing of save set

Save set:          MARCH10.BCK
Written by:        DORSEY
UIC:               [011,030]
Date:               5-JUN-1982 13:08:13.11
Command:           BACKUP [COURSE.UTLCOM...]*.* MYTAPE:MARCH10.BCK
Operating system:  VAX/VMS version V3.0
BACKUP version:    V3.0
CPU ID register:   0130A095
Node name:         _HARDY::
Written on:        _MTA0:
Block size:        8464
Group size:        10
Buffer count:      3

[000000]000000.DIR;1                                    11  26-APR-1982
 04:34
[000000]COURSE.DIR;1                                     1  24-MAR-1982
 11:48
[COURSE]UTLCOM.DIR;1                                     1  24-MAR-1982
 12:19
[COURSE.UTLCOM]BAS.DIR;1                                 1  24-MAR-1982
 12:21
[COURSE.UTLCOM.BAS]BADCNVRT.BAS;1                        4  22-APR-1980
 10:31
[COURSE.UTLCOM.BAS]BADFACT.BAS;1                         2  22-APR-1980
 09:33
[COURSE.UTLCOM.BAS]BADROMAN.BAS;1                        2  22-APR-1980
 09:36
[COURSE.UTLCOM.BAS]CNVRT.BAS;1                           4  22-APR-1980
 10:32
[COURSE.UTLCOM.BAS]CREATOR.BAS;1                         3   7-MAY-1980
 11:06
.
.
.

Total of 212 files, 546 blocks
End of save set
$ SHOW DEVICE/FULL MYTAPE

  Device   MTA0:          5-JUN-1982 13:08:48.76

       on line
       Mounted
       Error Logging Enabled
       Allocated
       Foreign

    Error count                 13    Owner process id          004C0048
    Operations completed      9953    Owner process name        DORSEY
    Reference count              1    Default buffer size            512
    Density                1600 BPI

    Volume label       UTLBCK           Record size                    0
    Owner UIC          [011,030]        Transaction count              1
    Volume protection  FF00             Mount count                    1
    ACP process name                    Relative Volume No.            0
$ DISMOUNT MYTAPE
$ DEALLOCATE MYTAPE
$ DEASSIGN MYTAPE
```

Example 4-2 Transferring Files to a Tape Volume Using BACKUP

Major Steps in Using a Disk or Tape

- $ MOUNT [/qualifiers] device-name[:] [label] -
  $_[logical-name[:]]

  .
  .
  .
  use device - can refer to it by logical name assigned by MOUNT
  .
  .
  .

- $ DISMOUNT [/qualifiers] device-name[:]

  or

- $ DISMOUNT [/qualifiers] logical-name[:]

```
$ SHOW DEVICE MT:
  List of Devices                    on        5-JUN-1982 13:35:20.22
  Device    Device      Device          Err.     Volume          Free  Trans Mount
  Name      Status  Characteristics     Count     Label         Blocks Count Count
  MTA0:     on line                      13
$ SET DEFAULT [COURSE.UTLCOM]
$ MOUNT/FOREIGN MT: UTLBCK MYTAPE
%MOUNT-I-MOUNTED, MARCH1 mounted on _MTA0:
$ DIRECTORY
  No files found.
$ BACKUP/LIST MYTAPE:MARCH10.BCK
Listing of save set

Save set:           MARCH10.BCK
Written by:         DORSEY
UIC:                [011,030]
Date:                5-JUN-1982 13:08:13.11
Command:            BACKUP [COURSE.UTLCOM...]*.* MYTAPE:MARCH10.BCK
Operating system:   VAX/VMS version V3.0
BACKUP version:     V3.0
CPU ID register:    0130A095
Node name:          _HARDY::
Written on:         _MTA0:
Block size:         8464
Group size:         10
Buffer count:       3

[000000]000000.DIR;1                                    11  26-APR-1982 04:34
[000000]COURSE.DIR;1                                     1  24-MAR-1982 11:48
[COURSE]UTLCOM.DIR;1                                     1  24-MAR-1982 12:19
[COURSE.UTLCOM]BAS.DIR;1                                 1  24-MAR-1982 12:21
[COURSE.UTLCOM.BAS]BADCNVRT.BAS;1                        4  22-APR-1980 10:31
[COURSE.UTLCOM.BAS]BADFACT.BAS;1                         2  22-APR-1980 09:33
[COURSE.UTLCOM.BAS]BADROMAN.BAS;1                        2  22-APR-1980 09:36
[COURSE.UTLCOM.BAS]CNVRT.BAS;1                           4  22-APR-1980 10:32
[COURSE.UTLCOM.STR]FILE6.LIS;1                           1  25-SEP-1980 10:24

Total of 212 files, 546 blocks
End of save set
$ BACKUP MYTAPE:MARCH10.BCK [*...]
$ DIRECTORY

Directory WORK:[COURSE.UTLCOM]

BAS.DIR;1           COB.DIR;1           COM.DIR;1           EDT.DIR;1
FIL.DIR;1           FOR.DIR;1           JOB.DIR;1           LIB.DIR;1
LOG.DIR;1           MAR.DIR;1           PRN.DIR;1           SRT.DIR;1
STR.DIR;1

Total of 13 files.
$ DISMOUNT MYTAPE
```

Example 4-3 Restoring a Directory Hierarchy from Tape (Saved with BACKUP) and Showing an Operator-Assisted MOUNT

PROTECTING A VOLUME

- Protection codes are set when volume is:
  - Initialized
  - Mounted
- You have control over:
  - Volume owner's UIC
  - Volume protection code
- Separate owner UICs and protection codes for:
  - Volume
  - Directories
  - Files
- To set volume protection code:
  - Volume must be blank, or
  - Volume owner UIC matches yours, or
  - You have VOLPRO privilege

Table 4-6 Protecting Volumes in the Initializing Phase

COMMAND FORMAT

$ INITIALIZE[/qualifier(s)] device label

Purpose/Comment	Qualifier	Effects
Protecting tapes or disks; can use any combination of these two qualifiers	none	owner UIC: [g,m] * protection: S:RWED,O:RWED,G:RWED,W:RWED
	/PROTECTION=code	owner UIC: no effect protection: specified in code
	/OWNER_UIC=uic	owner UIC: uic specified protection: no effect
Protecting disks; can use listed combinations of qualifiers only	/NOSHARE	owner UIC: [g,m] * protection: S:RWED,O:RWED,G,W
	/GROUP	owner UIC: [g,0] * protection: S:RWED,O:RWED,G:RWED,W:RWED
	/GROUP/NOSHARE	owner UIC: [g,0] * protection: S:RWED,O:RWED,G:RWED,W
	/SYSTEM	owner UIC: [1,1] protection: S:RWED,O:RWED,G:RWED,W:RWED

* In [g,m], g = user's group number, m = user's member number

THE MOUNT COMMAND

Table 4-7 Effects of $MOUNT Qualifiers on a Volume

COMMAND FORMAT	$MOUNT[/qualifier(s)] device label log-name	
Purpose	**$MOUNT Qualifier**	**Effects/Required Privilege**
Override volume parameters while volume is mounted		
Override owner UIC of tape or disk	/OWNER_UIC=[uic]	- specified UIC is owner while volume is mounted - volume ownership or VOLPRO required
Override disk protection code while mounted	/PROTECTION=code	- specified code is protection while disk is mounted - volume ownership or VOLPRO required
Extend disk access to others		
Make disk access available to your group	/GROUP	- assigns logical name to group table - group members access the volume without mounting it - group members must still pass volume protection code - GRPNAM required
Make disk access available to all users (public)	/SYSTEM	- assigns logical name to system table - all users access the volume without mounting it - users must still pass volume protection code - SYSNAM required
Make disk available to all users issuing a $MOUNT/SHARE command	/SHARE	- other users must still pass volume protection - no special privileges required
Mounting a volume of unknown label	/OVERRIDE=IDENTIFICATION	- requires VOLPRO privilege or ownership of volume

MOUNTING A DISK WITH AN UNKNOWN LABEL

- Requires disk ownership or VOLPRO privilege

```
$ MOUNT/OVERRIDE=IDENTIFICATION DY: UNKNOWN MYDISK
%MOUNT-I-MOUNTED, USER_DISK     mounted on _DYA0:
$ SHOW DEVICE/FULL MYDISK

  Device   DYA0:           7-JUN-1982 10:39:25.07

       on line
       Mounted
       Error Logging Enabled
       Allocated

    Error count                    0    Owner process id          0008002E
    Operations completed         255    Owner process name        DORSEY
    Reference count                1    Default buffer size           512

    Volume label       USER_DISK        Free blocks                   446
    Owner UIC          [100,200]        Transaction count               1
    Volume protection  FFOF             Mount count                     1
    Volume status                       Relative volume no.             0
    ACP process name   DYA0BACP         Cluster size                    1
    Quota cache size             64     Max. files allowed            123
    File ID cache size           32     Extent cache size              64
$ DISMOUNT MYDISK
$
```

Example 4-4 Mounting a Disk with an Unknown Label

Public Volumes

- Initializing a disk /SYSTEM sets
  - UIC=[1,1]
  - Protection = (S:RWED,O:RWED,G:RWED,W:RWED)
- Mounting a disk /SYSTEM
  - Deallocates the device
  - Places logical name is system logical name table
  - Allows all users to access volume without mounting it

```
$ ALLOCATE DB: PUBDISK
  _DBA0: allocated
$ MOUNT/FOREIGN PUBDISK
%MOUNT-I-MOUNTED,                mounted on _DBA0:
$ RUN SYS$SYSTEM:BAD
BAD>DB0:
BAD -- DB0: TOTAL BAD BLOCKS= 0.
BAD>^Z
$ DISMOUNT/NOUNLOAD PUBDISK
$ INITIALIZE/SYSTEM PUBDISK MYVOL
$ MOUNT/SYSTEM PUBDISK MYVOL
%MOUNT-I-MOUNTED, MYVOL          mounted on _DBA0:
$ SHOW LOGICAL/SYSTEM

  Contents of system logical name table :

  CDD$DICTIONARY   =  "SYS$SYSTEM:"
  DBG$INPUT   =  "SYS$INPUT:"
  DBG$OUTPUT   =  "SYS$OUTPUT:"
  DISK$ANIMAL   =  "__DRA0:"
  DISK$MYVOL   =  "__DBA0:"
```

Example 4-5 Initializing and Mounting a Public Disk

MAINTAINING PUBLIC VOLUMES

The recommended procedure is to perform regular maintenance on public volumes.

- To provide backup copies of volumes
- To improve I/O performance of the volume
- To recover blocks allocated to files that were not properly closed

Using the BACKUP Utility to Maintain Public Volumes

The BACKUP utility performs the following:

- Saves disk files onto tape or disk in a file called a save-set
- Restores a save-set to disk
- Lists information about the files in a save-set
- Copies files from disk to disk
- Compares disk files or files in a save-set with other disk files

BACKUP is used to:

- Back up and recover disk volumes
- Create more efficient copies of disk volumes to improve I/O performance
- Copy system volumes (stand-alone BACKUP)

Table 4-8 Backing Up Disk Volumes

Function	Command/Example	Comments
Save a disk volume to a save-set on tape	$BACKUP/IMAGE dev-name - $_save-set-specifier $BACKUP/IMAGE DBA1: - $_MTA0:MAR24.BCK	Creates a save-set on tape to restore data in case of failures or to temporarily store data on a less expensive medium.
Restore a disk volume from an image save-set	$BACKUP/IMAGE save-set-specifier - $_dev-name $BACKUP/IMAGE MTA0:MAR24.BCK - $_DBA1:	Restore a save-set to recover lost data or move data out of temporary storage.
Copy a disk volume to another disk and create a efficient volume	$BACKUP/IMAGE dev-name dev-name $BACKUP/IMAGE OLDDISK NEWDISK	Make a copy of the volume specified by the logical name OLDDISK onto the volume assigned to volume NEWDISK. In the new volume, the files are made more contiguous, making I/O more efficient.
Do an incremental backup	$BACKUP/RECORD - $_dev-name/SINCE=BACKUP - $_save-set-specifier $BACKUP/RECORD - $_DB2:[*...]/SINCE=BACKUP - $_MTA0:19JUN.BCK	Back up those files that were created or modified since the last save operation.

Note: See Chapter 5 of the VAX/VMS System Management and Operations Guide for more information on backing up public files and volumes.

BACKING UP THE SYSTEM DISK

To back up and restore a system disk on a system with at least two drives capable of handling the system disk:

- Log in at the console terminal
- Run the shutdown command procedure
- Stop the processor by pressing <CTRL/P> and typing HALT
- Boot stand-alone BACKUP. See your Software Installation Guide for instructions
- Physically load the scratch volume on the disk device
- Issue the command string

  BACKUP/IMAGE source-dev destination-dev

  to copy the system disk onto the scratch disk
- Halt the processor by pressing <CTRL/P> and then typing HALT
- Boot the system
- If the copy is needed in the future, mount it as the system disk when booting the system

To back up and restore a system disk on a system without two volumes capable of handling the system disk or on a system with a fixed system disk:

- Log in at the console terminal
- Run the shutdown command procedure
- Stop the processor by pressing <CTRL/P> and typing HALT
- Boot stand-alone BACKUP. See your Software Installation Guide for instructions
- Physically load the scratch disk or tape on the appropriate device
- Issue the command string

  BACKUP/IMAGE device-name save-set-specifier

  to copy the system disk into a save set on the scratch volume
- Halt the processor by pressing <CTRL/P> and then typing HALT
- Boot the system
- If the copy is needed in the future, use stand-alone BACKUP to restore the save set to the system disk

```
$ SET DEFAULT SYS$SYSTEM
$ @SHUTDOWN.COM
   .
   .
   .
^P HALT
>>>BOOT CS1

Please insert the first system diskette or
cartridge and press return

Please insert the second system diskette or
cartridge and press return

$ BACKUP/IMAGE DMA0: MTA0:MAY20SYS.BCK
$^P HALT
.
.
.
>>>BOOT
$
```

Example 4-6 Backing Up a VAX-11/780 Using Stand-Alone BACKUP

THE VERIFY UTILITY

The VERIFY utility checks the readability and validity of the file structure.

Modes of Operation

- Error reporting with no repairs
- Error reporting with repairs
- Error reporting with user-controlled selective repairs

Table 4-9 Using the VERIFY Utility

COMMAND FORMAT	$ ANALYZE/DISK_STRUCTURE dev-name/[qualifier]	
Function	**Command/Example**	**Comments**
Running VERIFY to find errors in the file structure	$ ANALYZE/DISK_STRUCTURE dev-name $ ANALYZE/DISK_STRUCTURE MYDISK	Using no parameter qualifier invokes VERIFY in the error reporting mode.
Running VERIFY to report and repair errors	$ ANALYZE/DISK_STRUCTURE - $_dev-name/REPAIR $ ANALYZE/DISK_STRUCTURE - $_MYDISK/REPAIR	The /REPAIR qualifier causes automatic repair of all errors located in the file structure.
Running VERIFY to report errors and allow the user to decide what gets repaired	$ ANALYZE/DISK_STRUCTURE - $_dev-name/REPAIR/CONFIRM $ ANALYZE/DISK_STRUCTURE - $_MYDISK/REPAIR/CONFIRM	The /REPAIR/CONFIRM qualifiers causes the utility to display errors, asks the user if the error should be corrected, and carries out the user's request.

UNDERSTANDING THE DISK FILE STRUCTURE ON VAX/VMS

- FILES-11 Structure Level 2
- Also called On-Disk Structure 2 (ODS-2)
- Used by ACPs to maintain and control data on disk volumes

ODS-2 Reserved Files

- Define the FILES-11 disk file structure
- Created when a volume is initialized
- Cataloged in the Master File Directory [0,0] of the volume

Table 4-11 Volume Information Contained in ODS-2 Reserved Files

Information Needed	Reserved File
Location of file data blocks on a volume	Index File (INDEXF.SYS)
Location of available clusters on a volume	Storage Bit Map File (BITMAP.SYS)
Properties of a volume	Index File (INDEXF.SYS)
Properties of files on a volume (i.e., ownership, protection)	Index File (INDEXF.SYS)
Areas of volume not suitable for use	Bad Block File (BADBLK.SYS)
Areas of volume which are suspected of being unsuitable for use	Pending Bad Block Log File (BADLOG.SYS)
Pointers to all User File Directories	Master Directory File (000000.DIR)
Labels of other volumes in the volume set (if there is more than one)	Volume Set List File (INDEXF.SYS)
History of backups done on the volume	Backup Log File (BACKUP.SYS)

ODS-2 Home Block

- Identifies a disk volume with information including:
  - A label specifying that the volume is in ODS-2 format
  - The volume label (in ASCII)
  - The name and UIC of its owner
  - Volume protection code
  - The maximum number of files allowed at any one time
- Provides entry point into the file structure
- Backup copies are stored in the volume index file

Files on a FILES-11 Disk Volume

- Consist of two major data structures:
  - File header with identification and protection information
  - One or more data blocks
- Composed of logical disk blocks
- Each logical block has a virtual block number
- Blocks are grouped into clusters
- Contiguous clusters are called "extents"

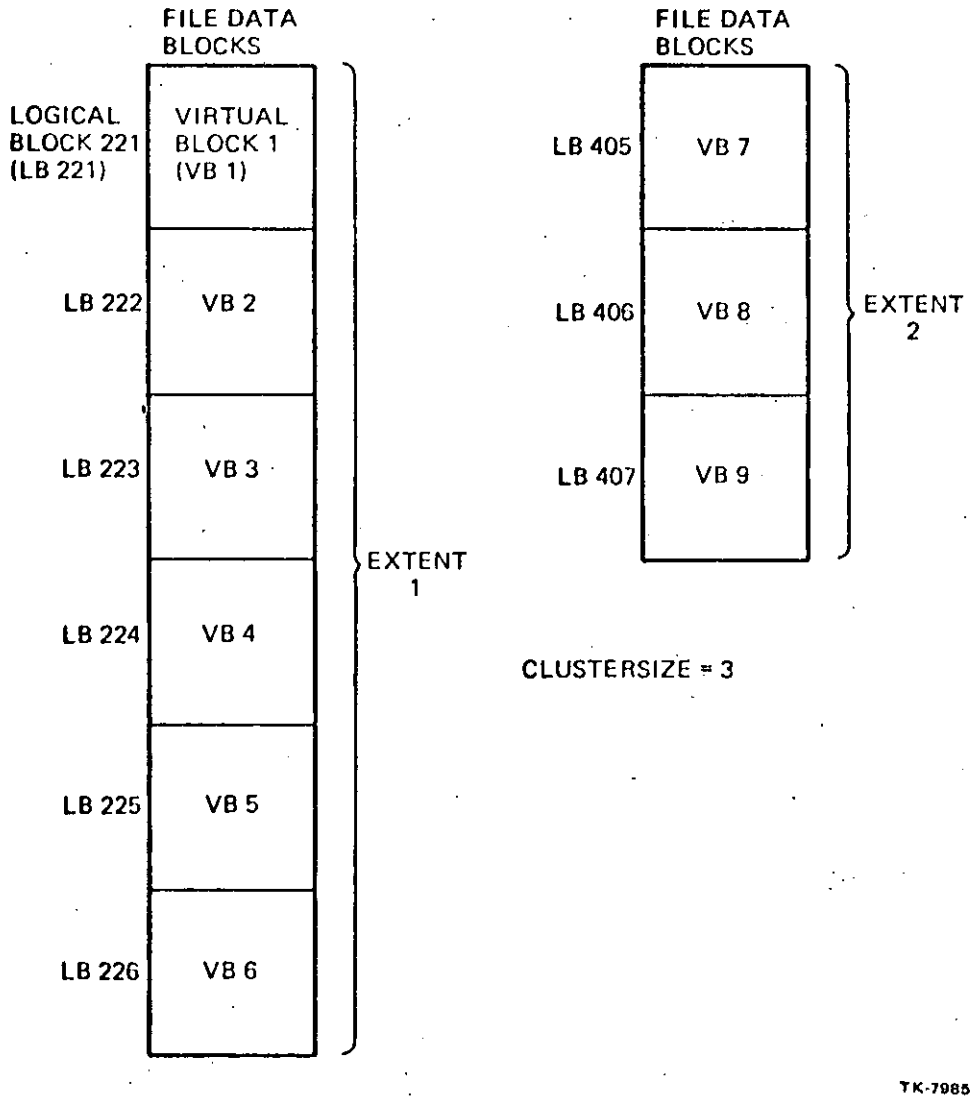


Figure 4-3 Relationship Between the Blocks, Clusters, and Extents of a Typical Disk File

ODS-2 File Headers

- Part of the volume index file
- Not part of the file it describes
- Divided into six areas:
  - Header area - Contains basic information for checking access validity
  - Indent area - Contains identification and accounting information
  - Map area - Contains pointers to the blocks allocated to the file (see Figure 4-4)
  - Access area - Currently unused
  - Reserved area - Reserved for use by customers and CSS
  - Checksum - Validity check on header's contents (the last word of header)
- Used to locate file data blocks on disk

File ID

- Unique three-word identification for file
- File number - Number of file header in index file
- File sequence number - Count of times this header has been used for different files
- Relative volume number - Number of volume on which file resides

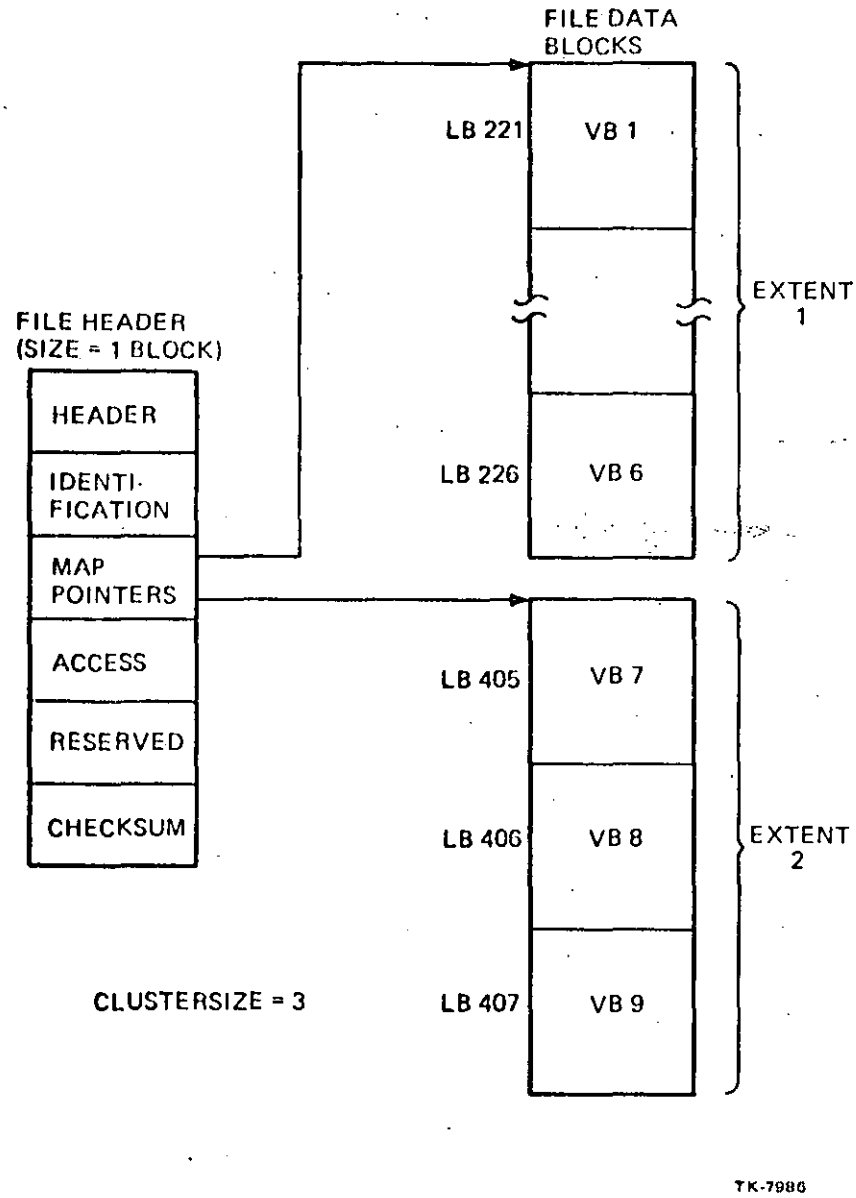


Figure 4-4 Map Pointers to the Data Blocks of a Disk File

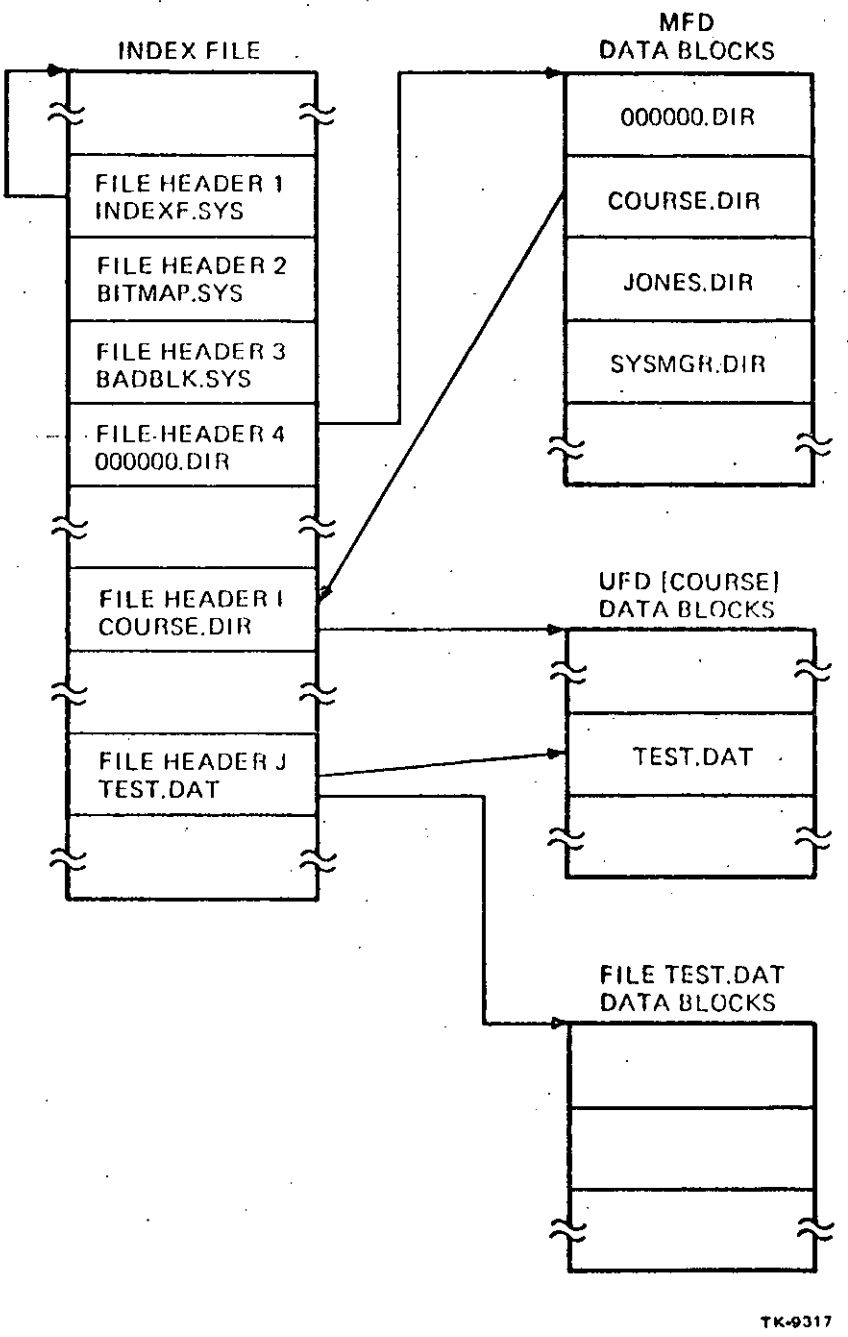


Figure 4-5 Use of ODS-2 File Headers to Access a File

Choosing a Cluster Size for Disk Volumes

- Cluster size is defined when a volume is initialized
- A larger cluster size can waste space, but allows for faster allocation of additional blocks
- A smaller cluster size can conserve space, but causes files to be fragmented

ODS-2 Directories

- Associate symbolic file names with file IDs
- User File Directories (UFDs) are entered in the Master File Directory (MFD)
- Subfile directories (SFDs) are entered in their parent directory file
- Directory names may be:
  - In UIC format with the group and member fields in the range 0 to 377 octal, or
  - In alphanumeric format of not more than eight characters
- Subdirectory names must be alphanumeric
- Directory files are:
  - alphanumeric-name.DIR;1
  - UIC-formatted directory names are converted to their numeric equivalent:

    [123,12] = 123012.DIR;1
- VAX/VMS supports up to seven levels of subdirectories beneath the UFD level

Directory File Structure

- Directory file characteristics
  - Contiguous
  - Sequential
  - Variable length records
  - Directory bit is set in the file header
  - File protection code of S:RWE,O:RWE,G:RWE,W:RE
  - File entries are stored in alphabetical order

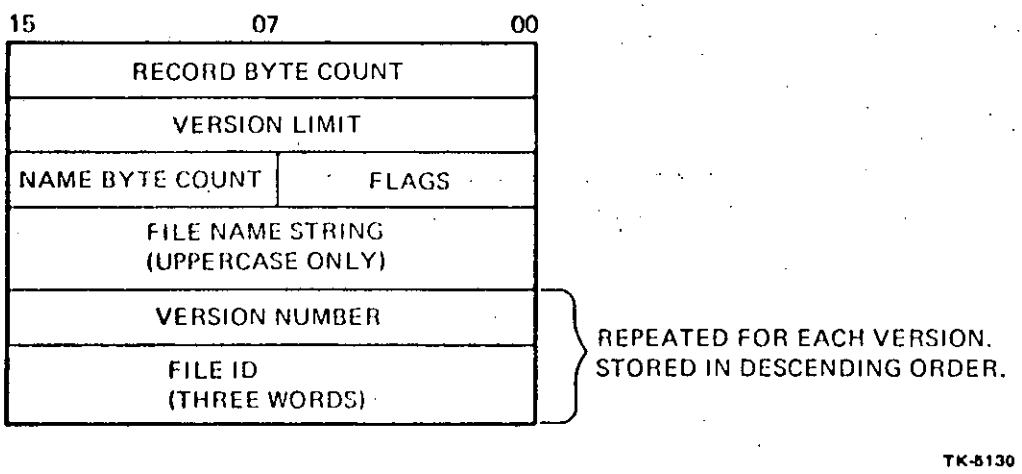


Figure 4-6 Format of a Directory File Record

Comparing the ODS-1 and ODS-2 Structures

Table 4-12 ODS-1 and ODS-2 Comparison

Characteristic	ODS-1	ODS-2
Transportable to RSX/IAS	Yes	No
Subdirectories	No	Yes
Alphanumeric directory names	No	Yes
Alphabetized directory files	No	Yes
Clustered allocation	No	Yes
Tightly-coupled volumes sets	No	Yes
Backup home block	No	Yes
Extended map pointers	No	Yes
Meaning of protection code E	Extend	Execute

Understanding the VAX/VMS Implementation of the ANSI Level 3 Tape File Structure

- Sections of an ANSI standard magnetic tape
  - Physical BOT and EOT markers
  - Volume label
  - File header labels
  - File data blocks
  - End-of-file labels
  - Logical end-of-tape marker

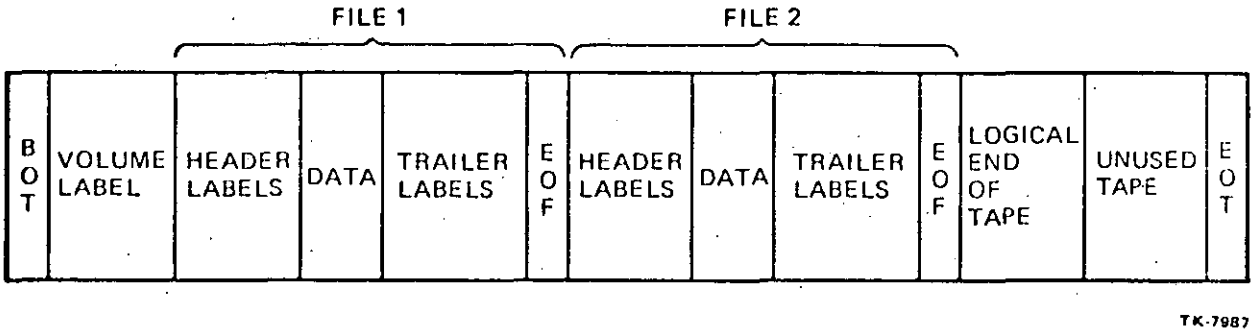


Figure 4-7 Format of an ANSI Tape Containing Two Files

Magnetic Tape Volume Label

Table 4-13 Contents of the Magnetic Tape Volume Label

Information	Purpose
Label identifier (VOL1)	Specify that this is the ANSI volume label
Volume name	Used as the volume label for mount operations (consists of 1-6 alphanumeric or special characters)
Accessibility	Provides compatibility with some non-DIGITAL systems
Owner identifier	Specifies the owner of the volume; used for volume protection
DIGITAL standard version	(Numeric character 1)
Label standard version	Identifies the version of the ANSI standard being used (version 3)

Tape File Header Labels

- Precede each file on a tape volume
- Identify and describe a tape file
- VAX/VMS supports three file header labels (HDR1, HDR2, and HDR3)
- The third header (HDR3) is optional

Table 4-14 Contents of Tape File Header Labels

Label	Contents
HDR1	File name File section number (for multi-volume files) File sequence number Creation and expiration dates Accessibility code Code specifying which system created the file.
HDR2	Record format (fixed or variable-length) Maximum record size Maximum block length Carriage control definition
HDR3	VAX-11 RMS attributes of the file

Tape File Trailer Labels

- Follow each file on a tape volume
- Also called end-of-file labels (EOF)
- The three EOF labels correspond to the three header labels
- Identical to HDR counterparts except:
  - The label identifier field contains EOFn instead of HDRn (where n is the label number), and
  - The EOF labels contain a count of the data blocks in the file

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" Y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO, D.F.

MANAGING VMS

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985

INTRODUCTION

The VMS environment is established in two stages. At system start-up, the default environment for all processes is established. When an interactive or batch process is created, the environment of that process is customized.

The two principal methods of defining the process environment are:

- Setting system parameters
- Issuing DCL commands

System parameters, which affect all processes, are usually set by the system manager. These parameters control the physical structure of VMS and are used to improve system performance. System parameters are covered in other modules of this course, particularly Module 7, System Installation and Update, and Module 9, Performance.

Each time the system is initialized, the system-wide VMS environment is set up in accordance with the system parameters and the system start-up files. The system start-up files, which contain DCL commands, are usually controlled by the system manager. Commands typically executed in the system start-up files are covered in this module.

When a user logs in or a batch job runs, the VMS environment for that interactive or batch process is customized by DCL commands in the system manager's and user's login command files. These files are covered in Module 2, Managing System Users.

The DCL commands used in the start-up and login command files can also be issued interactively. In this case, their effect is only temporary. For a command to take effect at every start-up or every login, it must be placed in the appropriate file.

Security is an important aspect of the VMS environment. All system files are stored in specific directories on the system disk. VMS protects these files against unauthorized access.

OBJECTIVES

A system manager must:

- Set up system-wide logical names
- Specify the characteristics of a device
- Increase process sharing of physical memory pages by installing global sections and known images

RESOURCES

1. VAX/VMS Command Language User's Guide
2. VAX/VMS System Management and Operations Guide
3. VAX-11 Utilities Reference Manual
4. VAX-11/xxx Software Installation Guide
   (/xxx = /780, /750, or /730)

OVERVIEW

- Environment within which interactive user works, batch job runs is called a process
- Process default environment is established in two stages:
  - At system start-up:

    System-wide default established
  - When a process is created:

    Individual process environment customized
- Process environment can also be customized interactively
- Principal methods of defining process environment:
  - Setting system parameters
  - Issuing DCL commands

- System-wide VMS environment is set up
  - When system is initialized
  - In accordance with:

    System parameters

    System start-up command procedures
- Single process VMS environment is set up
  - When user logs in
  - When batch job runs
  - By DCL commands in:

    Systems manager's login command procedure

    User's login command procedure

Table 5-1 Files Controlling the VMS Environment

File	Information in File that Controls VMS Environment	When Information in File Is Automatically Used
SYS$SYSTEM:SYS.EXE	System parameters	System start-up
SYS$SYSTEM:STARTUP.COM	DCL commands	System start-up
SYS$MANAGER:SYSTARTUP.COM	DCL commands	System start-up
SYS$MANAGER:SYLOGIN.COM	DCL commands	Interactive or batch process creation
SYS$LOGIN:LOGIN.COM	DCL commands	Interactive or batch process creation

- Security issues

CUSTOMIZING THE SYSTEM START-UP FILES

- Two start-up command procedures
  - SYS$SYSTEM:STARTUP.COM
    - Site-independent
    - Distributed as part of VMS
    - Commands supplied by DIGITAL
    - Do not modify
  - SYS$MANAGER:STARTUP.COM
    - Site-specific
    - Distributed as part of VMS
    - Initially empty
    - Add site-specific customization commands
    - Override inappropriate commands in STARTUP.COM
  - Executed in order when VMS initialized

- Functions performed by STARTUP.COM, covered in this module:
  - Assigning logical names required by certain VMS software
  - Assigning logical names to the VMS system directories
  - Starting up system processes JOB_CONTROL, OPCOM, and ERRFMT
  - Installing known images
  - Connecting all standard devices
  - Enabling RMS file sharing
  - Calling the site-specific start-up command procedure
  - Allowing users to log in
- Functions performed by SYSTARTUP.COM, covered in this module:
  - Assigning system logical names
  - Mounting all volumes except the system disk
  - Installing known images
  - Reinstalling known images installed in STARTUP.COM
  - Allocating additional space to RMS file sharing
  - Establishing device characteristics
  - Announcing that the system is up
- Functions performed by SYSTARTUP.COM, covered in other modules:
  - Establishing batch and print queues (Module 3)
  - Deleting old versions of the operator's log (Module 2)
  - Starting up DECnet (Module 7)
  - Analyzing the last system crash dump (Module 8)

Assigning System Logical Names

- To permit device-independent references to:
  - System devices
  - System directories
  - System files
- To inform VMS of the location of certain files and text strings
  - SYS$SYLOGIN
  - SYS$ANNOUNCE
  - SYS$WELCOME

Table 5-2 Assigning System Logical Names

Operation	Command Format and Example	Comments
Creating or replacing a system logical name	$ ASSIGN/SYSTEM eqv-name log-name $ DEFINE/SYSTEM log-name eqv-name $ ASSIGN/SYSTEM SYS$SYSTEM:NOTICE.TXT NOTICE	Requires SYSNAM privilege
Deleting a system logical name	$ DEASSIGN/SYSTEM log-name $ DEASSIGN NOTICE	Requires SYSNAM privilege

```
$ !
$ ! Assign debugger systemwide logical names
$ !
$ ASSIGN /SYSTEM SYS$INPUT: DBG$INPUT:
$ ASSIGN /SYSTEM SYS$OUTPUT: DBG$OUTPUT:
$ !
$ ! Assign systemwide logical names for libraries and help files
$ !
$ DEASSIGN /SYSTEM SYS$SYSTEM
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSEXE] SYS$SYSTEM:
$ DEASSIGN /SYSTEM SYS$MESSAGE
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSMSG] SYS$MESSAGE:
$ DEASSIGN /SYSTEM SYS$SHARE
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSLIB] SYS$SHARE:
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSLIB] SYS$LIBRARY:
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSHLP] SYS$HELP:
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSHLP.EXAMPLES] SYS$EXAMPLES:
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSMGR] SYS$MANAGER:
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSUPD] SYS$UPDATE:
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSTEST] SYS$TEST:
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSMAINT] SYS$MAINTENANCE:
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSERR] SYS$ERRORLOG:
$ ASSIGN /SYSTEM SYS$SYSROOT:[SYSCBI] SYS$INSTRUCTION:
```

Example 5-1 Assigning System Logical Names with STARTUP.COM

```
$ !
$ ! Define site-specific logical names
$ !
$ ASSIGN /SYSTEM SYS$MANAGER:SYLOGIN    SYS$SYLOGIN
$ ASSIGN /SYSTEM DISK$USER:[TOOLS]      SYS_TOOLS
$ ASSIGN /SYSTEM "      MENTOR  -  VMS Course Development"      SYS$ANNOUNCE
```

Example 5-2 Assigning System Logical Names with SYSTARTUP.COM

Mounting Site-Specific Volumes

- Mounted in site-specific start-up command procedure
- No volumes are mounted in the site-independent start-up command procedure (not even the system disk)
- By default in SYSTARTUP.COM, MOUNT means MOUNT/NOASSIST
- MOUNT creates logical names that are used for device independence in
  - Issuing commands
  - Displaying file specifications in listings (indicated by two underscores in equivalence string device name)

```
$ !
$ !     Mount public disk volumes
$ !     (MOUNT /NOASSIST is the default when SYSTARTUP.COM is
$ !     executed indirectly from STARTUP.COM.)
$ !
$ MOUNT /SYSTEM DBA0: DATA13 DATADISK
$ MOUNT /SYSTEM DBA1: USER
```

Example 5-3 Mounting Site-Specific Volumes with SYSTARTUP.COM

Starting Up System Processes

- Three system processes
  - JOB_CONTROL
  - OPCOM
  - ERRFMT
- By DCL command RUN in STARTUP.COM

Installing Known Images

- Reasons for installing
  - Reducing time required for image activation
  - Enabling the image to execute with privileges that the process lacks
  - Promoting sharing of physical memory
  - Adding a user-written system service
- Do not confuse two different VMS uses of word "install":
  - Install known images (covered here)
  - Install VMS operating system, optional software (covered in Module 7)

Table 5-3 Installing Known Images

Steps in Running the INSTALL Utility	Comments
1. $ RUN SYS$SYSTEM:INSTALL	Requires CMKRNL, PRMGBL, and SYSGBL privileges.
2. INSTALL > [file-spec] [/qualifier...]	Table 5-4 shows the INSTALL commands that install and remove known images and display information about them. Note that INSTALL uses the following file specifier defaults: Device and directory: SYS$SYSTEM File type: .EXE Version number: highest The version number should never be specified.
3. INSTALL > /EXIT	

Table 5-4 Functions of the INSTALL Utility

Operation	Command Format and Example	Comments
Installing a known image and specifying optional characteristics	INSTALL> file-spec [/qualifier...] INSTALL> EDT/OPEN/SHARED	Qualifiers used to specify optional characteristics include: /OPEN /HEADER_RESIDENT /PRIVILEGED /SHARED /WRITEABLE /PROTECTED
Installing a new version of a known image with the same characteristics as the existing version	INSTALL> file-spec/REPLACE INSTALL> EDT/REPLACE	
Installing a new version of a known image with different characteristics from the existing version	INSTALL> file-spec/DELETE INSTALL> file-spec [qualifier...] INSTALL> EDT/DELETE INSTALL> EDT/OPEN/SHARED/HEADER	
Removing a known image	INSTALL> file-spec/DELETE INSTALL> EDT/DELETE	
Displaying information about a known image or about INSTALL	INSTALL> [file-spec]/LIST INSTALL> [file-spec]/FULL INSTALL> /GLOBAL INSTALL> /HELP INSTALL> EDT/FULL	

- How INSTALL is run in STARTUP.COM
  - SYS$MANAGER:STARTUP.COM runs SYS$MANAGER:VMSIMAGES.COM
  - SYS$MANAGER:VMSIMAGES.COM runs INSTALL, gets commands from SYS$MANAGER:VMSIMAGES.DAT

```
SYS$SYSTEM:LINK          /OPEN /HEADER
SYS$SHARE:CRFSHR         /OPEN /HEADER /SHARED
SYS$SYSTEM:SUBMIT        /PRIV=(TMPMBX) /OPEN /HEADER /SHARED
SYS$SHARE:DISMNTSHR      /PROTECT /OPEN /HEADER /SHARE !/NOREMOVE
SYS$SHARE:MOUNTSHR       /PROTECT /OPEN /HEADER /SHARE
SYS$SYSTEM:RTPAD         /PRIV=(TMPMBX) /OPEN /HEADER /SHARED
SYS$LIBRARY:DEBUG        /OPEN /HEADER /SHARED
SYS$SYSTEM:PHONE         /PRIV=(NETMBX,OPER,PRMMBX,WORLD) /OPEN /HEADER
SYS$SYSTEM:LOGINOUT      /PRIV=(CMKRNL,TMPMBX,EXQUO,SYSPRV) /OPEN /HEADER /SHARED
SYS$SHARE:DCLTABLES      /OPEN /SHARED /HEADER
SYS$SYSTEM:DCL           /OPEN /SHARED /HEADER
SYS$SYSTEM:DELETE        /OPEN /HEADER /SHARED
SYS$SYSTEM:DIRECTORY     /OPEN /HEADER /SHARED
SYS$SYSTEM:COPY          /OPEN /HEADER /SHARED
SYS$SYSTEM:SET           /PRIV=(CMKRNL,SYSPRV,TMPMBX) /OPEN /HEADER /SHARED
SYS$SYSTEM:SHOW          /PRIV=(CMKRNL,NETMBX) /OPEN /HEADER /SHARED
SYS$SYSTEM:MAIL          /PRIV=(SYSPRV,OPER,WORLD,NETMBX) /OPEN /HEADER /SHARED
SYS$SYSTEM:EDT           /OPEN /SHARED /HEADER
SYS$SHARE:SCRSHR         /OPEN /SHARED /HEADER
SYS$SHARE:LBRSHR         /OPEN /SHARED /HEADER
SYS$SHARE:VMSRTL         /OPEN /SHARED /HEADER
```

Example 5-4 Typical INSTALL Commands in VMSIMAGES.DAT

- How INSTALL is run in SYSTARTUP.COM
  - Can install images not in VMSIMAGES.DAT
  - Can reinstall images in VMSIMAGES.DAT with modifications
  - Install most frequently activated images last

```
$ !
$ ! Install known images, most frequently activated last
$ !
$ RUN SYS$SYSTEM:INSTALL
SYS$SYSTEM:REPLY          /PRIV=OPER
SYS$SYSTEM:BLISS32        /OPEN /SHARED
SYS$SYSTEM:MACRO32        /OPEN /SHARED
SYS$SYSTEM:NCP            /OPEN /SHARED
!
! Re-install SHOW with WORLD priv
!
SYS$SYSTEM:SHOW /DELETE
SYS$SYSTEM:SHOW           /OPEN /SHARED /HEADER_RESIDENT -
                          /PRIV=(CMKRNL,WORLD,NETMBX)
!
! Re-install the linker to be shared in memory
!
SYS$SYSTEM:LINK /DELETE
SYS$SYSTEM:LINK           /OPEN /SHARED /HEADER_RESIDENT
```

Example 5-5 Installing Known Images with SYSTARTUP.COM

Connecting All Standard Devices

- Accomplished by SYSGEN command AUTOCONFIGURE/ALL

```
$ !
$ !     Automatically configure I/O data base to include all
$ !     standard devices and their required drivers.
$ !
$ RUN SYS$SYSTEM:SYSGEN
AUTOCONFIGURE ALL
```

Example 5-6 Connecting All Known Devices with STARTUP.COM

Enabling RMS File Sharing

- Allows more than one user access to a file at the same time

Table 5-5 Enabling RMS File Sharing

Steps in Running the RMSSHARE Utility	Comments
1. $ RUN SYS$SYSTEM:RMSSHARE	Requires CMEXEC privilege.
2. RMS file sharing is currently enabled... Maximum allocation allowed: nn Number of pages allocated: nn Max pages used: nn Current number of pages in use: nn Enter max pages: number-of-pages	The numbers of pages you specify is used as the new value for number of pages allocated. It must be at least the current value for number of pages allocated and no greater than the maximum allocation allowed.
3. . . . . Enter max pages: EXIT	

- System memory set aside by command in STARTUP.COM for the file sharing data structures

```
$ !
$ ! Enable RMS file sharing.
$ !
$ RUN SYS$SYSTEM:RMSSHARE
20
```

Example 5-7 Enabling RMS File Sharing with STARTUP.COM

- Increased by command in SYSTARTUP.COM (initial value can NOT be decreased)

```
$ !
$ ! Increase the amount of system memory reserved for RMS file sharing
$ !
$ RUN SYS$SYSTEM:RMSSHARE
30
```

Example 5-8 Increasing RMS File Sharing with SYSTARTUP.COM

Recording Device Characteristics

- STARTUP.COM AUTOCONFIGURE step records the number of terminal lines the system can support
- SYSTARTUP.COM records the physical characteristics of the attached terminals
- Can modify permanent characteristics interactively, but effective only until next start-up

Table 5-6 Recording Permanent Characteristics of Terminals

Characteristic	Qualifier for Command $ SET TERMINAL/PERMANENT and Examples (Requires LOG_IO or PHY_IO Privilege)	Comments
Line speed, for direct lines	/SPEED/NOMODEM $ SET TERMINAL/PERMANENT - $_/SPEED=9600/NOMODEM TTF5:	By default, all lines have the speed specified by system parameter TTY_SPEED.
Line speed, for lines connected through modems	/AUTOBAUD/MODEM $ SET TERMINAL/PERMANENT - $_/AUTOBAUD/MODEM TTA1:	VMS does not send broadcast messages to autobaud lines unless the terminal is logged in.
Terminal type, for direct lines	/DEVICE_TYPE=type $ SET TERMINAL/PERMANENT - $_/DEVICE_TYPE=VT100 TTC2:	Among the recognized device types are: LA36, LA120, VT52, and VT100. This one qualifier is a substitute for many individual qualifiers, each specifying only one characteristic.

```
$ SET TERMINAL/PERMANENT/NOMODEM/SPEED=9600/DEVICE=VT100  TTA0:  !M. BROWN
$ SET TERMINAL/PERMANENT/NOMODEM/SPEED=1200/DEVICE=LA120  TTA1:  !R. JONES
$ SET TERMINAL/PERMANENT/AUTOBAUD/MODEM                   TTA2:  !DIAL-UP
```

Example 5-9 Recording Device Characteristics with SYSTARTUP.COM

- Use SET PROTECTION/DEVICE to control device ownership and protection

Table 5-7 Establishing Ownership and Protection of Terminals and Other NonSharable Devices

Operation	Format of $ SET PROTECTION/DEVICE Command and Examples (Requires OPER Privilege)	Comments
Establishing protection ownership	$ SET PROTECTION=code/DEVICE - $_/OWNER_UIC=[uic] device $ SET PROTECTION=(SY:R,OW:R)/DEVICE - $_/OWNER_UIC=[11,21] TTA3:	By default all terminals have the owner specified by the system parameter TTY_OWNER and the protection specified by the system parameter TTY_PROT.
Allowing all users access to a device	$ SET PROTECTION/DEVICE device $ SET PROTECTION/DEVICE TTA3:	

Announcing That the System Is Up

- Command near end of STARTUP.COM
- Control returns to STARTUP.COM

```
$ !
$ ! Announce system is up
$ !
$ REPLY /ALL /BELL "Start-up complete: MENTOR is up and running."
```

Example 5-10 Announcing That the System Is Up with SYSTARTUP.COM

Allowing Users to Log In

- By default, 64 interactive users permitted
- To reduce number, add these commands to very end of SYSTARTUP.COM:

  ```
  $ SET LOGIN/INTERACTIVE = number
  $ LOGOUT/BRIEF
  ```

- Unless LOGOUT command is included, control returns from SYSTARTUP.COM to STARTUP.COM

```
$ !
$ ! Invoke system manager's installation-specific startup file.
$ !
$SITE_SPECIFIC:
$ MOUNT == "MOUNT /NOASSIST"
$ DEFINE SYS$LOGIN SYS$SYSTEM:
$ @SYS$MANAGER:SYSTARTUP
$ !
$ ! System initialization complete
$ !
$ SET LOGIN /INTERACTIVE=64
$ LOGOUT/BRIEF
```

Example 5-11 Allowing Users to Log In with STARTUP.COM

CUSTOMIZING THE DCL ENVIRONMENT

- VMS allows customization of:
  - DCL commands
  - HELP messages
  - Error messages
- For the most part, modifying this interface requires programming skills beyond the scope of this course
- The following material is only to acquaint you with what is possible

Adding DCL Commands

- Two files needed to add a new DCL command:
  - A command description file (.CLD)
  - An executable image file (.EXE)
- Steps to follow in adding a DCL command

Table 5-8 Adding a DCL Command

Steps to Follow	Comment
Step 1: $ COPY command-image-file SYS$SYSTEM	It is assumed that you have two files in your default directory: commd-descr-file (file type .CLD) commd-image-file (file type .EXE)
Step 2: $ SET COMMAND commd-descr-file - $_/TABLES=SYS$LIBRARY:DCLTABLES - $_/OUTPUT=SYS$LIBRARY:DCLTABLES	This creates a new version of the DCL tables, adding the new command (or replacing the existing command).
Step 3: $ RUN SYS$SYSTEM:INSTALL INSTALL> SYS$LIBRARY:DCLTABLES/REPLACE INSTALL> /EXIT	This installs the new DCL tables. Be sure to back up the old DCL tables, if you plan to delete them.

Adding HELP Text

- Add HELP information on a new DCL command
  - Need a text file (.HLP) that contains the information
  - Steps to follow in adding HELP text

Table 5-9 Adding HELP Text

Operation	Comment
$ LIBRARY/HELP - $_SYS$HELP:HELPLIB help-text-file	It is assumed that you have this file in your default directory: help-text-file (file type .HLP) This adds new HELP text (or replaces existing HELP text).

Adding Error Messages

- Error messages of a new command can be either:
  - Part of its executable image file
  - In a separate message file (.EXE), copied into SYS$MESSAGE

SECURITY OF SYSTEM FILES

Overview

- System files - The image files, data files, and other files required by VMS for its own operation

  Example: LINK.EXE

- System directories - System files are grouped into directories according to their characteristics

  Examples: [SYSEXE] LINK.EXE

  [SYSLIB] STARLET.OLB

- System volume: The disk volume where the system files and directories are stored

  - Files other than system files may also be stored on this volume

  - Operating systems other than the current version of VMS may also be stored on this volume (and started up from it)

    Stand-alone BACKUP

    Previous versions of VAX/VMS, V.3.0 and later

- System device: The disk drive on which the system volume is mounted

- Maintaining security on the system volume is the system manager's responsibility

System Directories

Table 5-10 System Directories

Directory Name Used in Combination with SYS$SYSROOT:	Logical Name	Principal Contents
[SYSEXE]	SYS$SYSTEM	Executable images, user authorization file, site-independent start-up and shutdown command procedures, system parameter files, paging file, swapping file, crash dump file
[SYSLIB]	SYS$LIBRARY SYS$SHARE	Macro libraries, object libraries, shareable image libraries, shareable images
[SYSHLP]	SYS$HELP	Help libraries
[SYSMSG]	SYS$MESSAGE	Error message files
[SYSMGR]	SYS$MANAGER	Site-specific start-up and shutdown command procedures, print forms definitions, operator's log, accounting file
[SYSMAINT]	SYS$MAINTENANCE	Hardware diagnostics (not distributed with VMS)
[SYSTEST]	SYS$TEST	UETP command procedures, images, data
[SYSERR]	SYS$ERRORLOG	Error log file
[SYSUPD]	SYS$UPDATE	Command procedures used for system installation, upgrade, and update
[SYSUPD.EXAMPLES]	SYS$EXAMPLES	Sample source programs such as device drivers and system services
[SYSCBI]	SYS$INSTRUCTION	Computer-based instruction software

- Each system directory has been assigned its own logical name
  - When referring to a system directory, use logical name SYS$SYSROOT to refer to the system volume
  - When referring to directories that are not system directories, use logical name SYS$SYSDEVICE to refer to the system volume
- System directory names conceal level 1 (root) directory of form SYSn, n=Ø,...,F (hex)
  - Supports multiple systems on the same disk
  - Under usual circumstances, concealed root directory is SYSØ
  - Concealment accomplished by assignment of logical name SYS$SYSROOT to equivalence string __ddcu:[SYSn]

- Example of steps in logical name translation

  1. SYS$SYSTEM:LINK.EXE
  2. SYS$SYSROOT:[SYSEXE]LINK.EXE
  3. __DRA0:[SYS0.SYSEXE]LINK.EXE (concealed)

File and Directory Protection

- To allow a user to run an executable image but not copy it, deny R access but grant E access
- To allow a user access to files in a directory by explicit file name and file type but not by wildcard, deny R directory access but grant E directory access
- Deny W access to all directories containing sensitive files
- After using EDT to modify a file, use SET PROTECTION to preserve original protection. EDT assigns your default protection to the files it creates

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" Y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO, D.F.

SYSTEM START-UP AND SHUTDOWN

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985

INTRODUCTION

A VAX/VMS system can usually remain running for long periods of time. Under certain circumstances, however, the system must be shut down. Such circumstances include:

- Performing hardware or software maintenance
- Backing up the system disk
- Moving or reconfiguring system hardware
- Upgrading or reconfiguring system software
- Losing electrical power or air conditioning
- Losing operating system response

VMS provides a command procedure that shuts the system down at a time you specify and notifies system users as that time approaches. Other shutdown methods are also available, in case the usual procedure is not effective.

There are many methods of starting up a VAX/VMS system. Because VMS keeps a record of your standard system configuration, starting up this configuration can be as simple as pressing a button. Using other start-up methods, you can override standard system characteristics and specify alternative system configurations. Furthermore, if you want to change to a different standard system configuration, you can modify the information recorded by VMS.

This module assumes that VMS software has already been configured to run on your VAX system. The subject of running VMS Version 3.Ø software on your VAX system for the first time is covered in Module 7, System Installation and Update.

OBJECTIVES

To install and use a system, a system manager must:

- Start the system up and shut it down
- Customize the bootstrap command files
- Customize the startup and shutdown files

RESOURCES

1. VAX/VMS System Management and Operations Guide
2. VAX-11 Hardware Handbook
3. VAX-11/xxx Software Installation Guide
   (/xxx = /780, /750, or /730)
4. VAX-11/782 User's Guide

STARTING UP A VAX/VMS SYSTEM

Phases of Start-Up

1. Hardware loads VMB.EXE, primary bootstrap program.
2. VMB.EXE loads SYS$SYSTEM:SYSBOOT.EXE, secondary bootstrap program.
3. SYSBOOT.EXE loads SYS$SYSTEM:SYS.EXE, executive image.
4. SYS.EXE initializes VMS data base, creates process running SYS$SYSTEM:SYSINIT.EXE.
5. SYSINIT.EXE continues system initialization, creates batch process executing DCL commands in SYS$SYSTEM:STARTUP.COM.
6. STARTUP.COM executes its own DCL commands and those in nested command procedure SYS$MANAGER:SYSTARTUP.COM, completes system initialization.

- Phase 1 - Hardware dependent
- Phases 2-6 - Mostly hardware independent
- Phases 1-3 - Called system bootstrapping

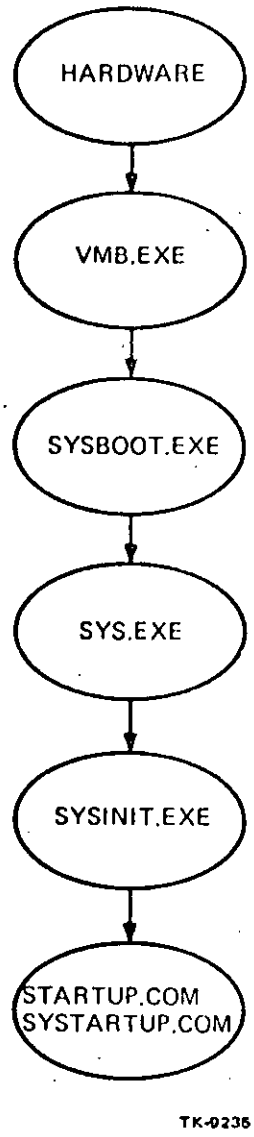


Figure 6-1 Start-Up Phases of VAX/VMS

VAX Console Subsystem

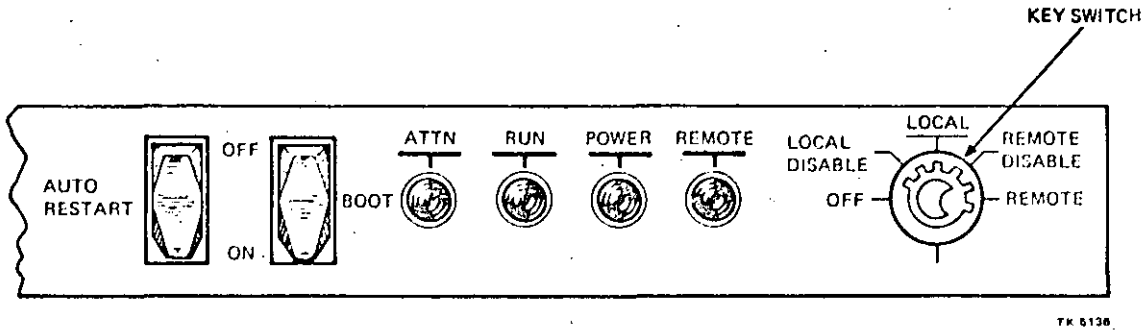
- Enables you to exercise control over the rest of the VAX hardware
- Six parts:
  - Console processor
  - Console command Language (CCL)
  - Console terminal
  - Console device
  - Front panel switches and lights
  - Remote diagnosis port (optional)

Table 6-1 VAX Console Subsystem

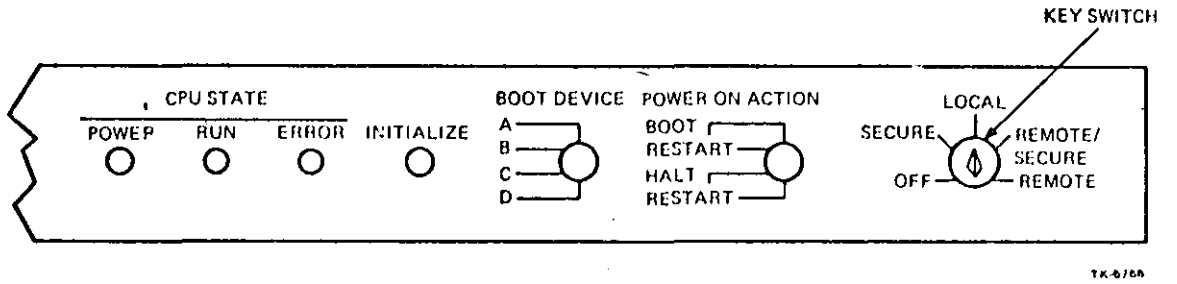
Component	Function
Console processor	Interprets and executes console commands affecting VAX hardware. Does not require the presence of VMS software.
Console command language (CCL)	Used to express console commands affecting VAX hardware.
Console terminal	Printing terminal used to enter CCL commands and display responses. When VMS is running, can also serve as interactive user terminal.
Console device	Holds console volume, which may contain console processor microcode, VAX microcode, and/or CCL command files. Used to update VMS, install optional VAX software, and run diagnostics.
Front panel switches and lights	Control system start-up, recovery from system failure, and remote diagnosis. Display system status. Key switch turns power on and off.
Remote diagnosis port	Enables a terminal at the DIGITAL Diagnosis Center to function as remote console terminal.

Front Panel Switches

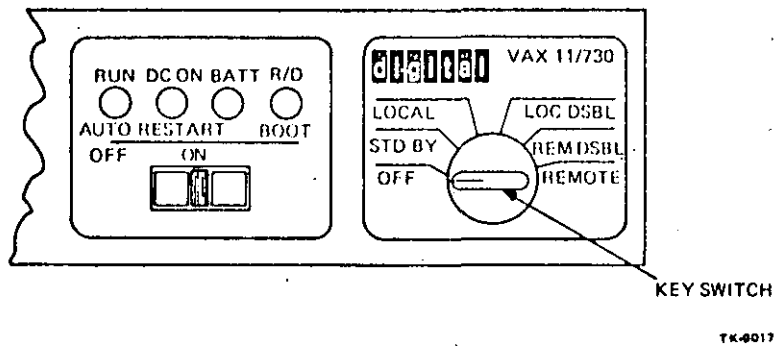
- Used to start up system
- Switches control processor
- Lights indicate processor status



(a) VAX-11/780



(b) VAX-11/750



(c) VAX-11/730

Figure 6-2 VAX Console Control Panels

Table 6-2 VAX Console Switches

Function	Switch		
	VAX-11/780	VAX-11/750	VAX-11/730
Turns power on and off. Enables and disables functions of console terminal and remote diagnosis port.	Keyswitch	Keyswitch	Keyswitch
Affects recovery from system failure.	AUTO RESTART	POWER ON ACTION	AUTO RESTART/BOOT
Requests system initialization.	BOOT	INITIALIZE	AUTO RESTART/BOOT
Selects system device at system start-up and automatic restart.	----	BOOT DEVICE	----

- Note terminology in course tables for settings of POWER ON ACTION switch of VAX-11/750 (counterclockwise from top):
  - BOOT
  - RESTART/BOOT
  - HALT
  - RESTART/HALT

Console Processor and Console Device

- For VAX-11/780 and VAX-11/730:
  - Console processor accomplishes the first start-up phase
  - Console volume must be mounted in console device
  - Console volume contains files needed by console processor

    Processor microcode

    Primary bootstrap program

    CCL command procedures
- For VAX-11/750:
  - Console volume NOT required for start-up
  - Necessary start-up code stored in read-only memory (ROM)
  - Use BOOT DEVICE switch to select ROM
  - Alternatively, you can start up using optional console volume, run BOOT58 program
  - BOOT58 command language is an extension of CCL

Table 6-3 VAX Console Devices

Processor	Console Device	Location	Device Name
VAX-11/780	RX01 diskette drive	Inside right door of processor cabinet, at bottom. Pivots out for access.	CSA1:
VAX-11/750	TU58 cartridge tape drive	On console front panel.	CSA1:
VAX-11/730	Dual TU58 cartridge tape drive	On console front panel.	CSA1:
		Inside processor cabinet. No access without sliding out processor.	CSA2:

Default System Device

- Each VAX processor supports a method of specifying a default system device
- If system volume is mounted on default system device, start-up is almost automatic

Table 6-4 Starting Up a VAX/VMS System from Power Off, Using the Default System Device

	VAX-11/780	VAX-11/750 No Console Volume	VAX-11/750 Using Console Volume	VAX-11/730
Step 1	Set AUTO RESTART switch to OFF. Place console volume in console device.	Set POWER ON ACTION switch to HALT. Set BOOT DEVICE switch to default setting for system device (B, C, or D).	Set POWER ON ACTION switch to HALT. Place console volume in console device. Set BOOT DEVICE switch to setting for console device, A.	Set AUTO RESTART/ BOOT switch to OFF. (Internal console volume is always in internal console device.)
Step 2	Set key switch from OFF to LOCAL.	Set key switch from OFF to LOCAL.	Set key switch from OFF to LOCAL.	Set key switch from OFF to LOCAL.
Step 3	When the >>> prompt appears on console terminal, toggle BOOT switch from OFF to ON.	When the >>> prompt appears on console terminal, set POWER ON ACTION switch to BOOT. Press INITIALIZE button.	When the >>> prompt appears on console terminal, set POWER ON ACTION switch to BOOT. Press INITIALIZE button. When the BOOT58> prompt appears, type BOOT<RET>.	When the >>> prompt appears on console terminal, toggle AUTO RESTART/BOOT switch from OFF to BOOT.
Step 4	Set key switch from LOCAL to LOCAL DISABLE. Set AUTO RESTART switch to ON.	Set key switch from LOCAL to SECURE. Set POWER ON ACTION switch to RESTART/ BOOT.	Set key switch from LOCAL to SECURE. Set BOOT DEVICE switch to default setting for system device (B, C, or D), and set POWER ON ACTION switch to RESTART/BOOT.	Set key switch from LOCAL to LOC DSBL. (AUTO RESTART/BOOT switch is set to ON.)

```
        CPU HALTED,SOMM CLEAR,STEP=NONE,CLOCK=NORM
        RAD=HEX,ADD=PHYS,DAT=LONG,FILL=00,REL=00000000
        INIT SEQ DONE
        HALTED AT 00000000

        (RELOADING WCS)
        LOAD DONE, 00003200 BYTES LOADED
        VER: PCS=01 WCS=0D-13 FPLA=0D CON=V04-03-L
>>>
        (BOOTING)
        CPU HALTED
        INIT SEQ DONE
        HALT INST EXECUTED
        HALTED AT 200034F9

        G  0000000E  00000200
        LOAD DONE, 00004200 BYTES LOADED

  VAX/VMS Version V3.0 26-APR-1982 16:21


%OPCOM, 27-JUN-1982 18:24:05.30, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
  Login quotas - Interactive limit=64, Current interactive value=0
  SYSTEM       Job terminated at 27-JUN-1982 18:24:24.33
```

Example 6-1 Default Start-Up of a VAX-11/780

```
%%

00000000  16
>>>
%
%%

  VAX/VMS Version V3.0 26-APR-1982 16:21

%OPCOM, 30-JUN-1982 17:16:40.42, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
  Login quotas - Interactive limit=32, Current interactive value=0
  SYSTEM       Job terminated at 30-JUN-1982 17:17:13.14
```

Example 6-2 Default Start-Up of a VAX-11/750, Not Using Console Volume

```
%%

00000000  16
>>>
%
%%

BOOT58>BOOT

!

!        DB0 BOOT COMMAND FILE - DB0B00.CMD

!

D/G 0 0                  ! DISK PACK DEVICE TYPE

D/G 1 F28000             ! BASE OF MBA0 ADDRESS SPACE

D/G 2 0                  ! ADAPTER UNIT = 0

D/G 3 0                  ! CONTROLLER UNIT = 0

D/G 4 0                  ! BOOT BLOCK LBN        (UNUSED)

D/G 5 0                  ! SOFTWARE BOOT FLAGS

D/G E 200                ! ADDRESS OF WORKING MEMORY + ^X200

LOAD VMB.EXE/START:200   ! LOAD PRIMARY BOOTSTRAP

START 200                ! START PRIMARY BOOTSTRAP

  VAX/VMS Version V3.0 26-APR-1982 16:21

%OPCOM, 30-JUN-1982 17:25:47.56, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
  Login quotas - Interactive limit=32, Current interactive value=0
  SYSTEM       Job terminated at 30-JUN-1982 17:26:19.99
```

Example 6-3 Default Start-Up of a VAX-11/750, Using Console Volume

```
CONSOLE
VERSION 89.01
>>>@POWER.CMD
>>>L/C CONSLE.CPU           !VERSION 52
>>>L/C/S:0800 MMIE.CPU      !VERSION 52
>>>L/C/S:0E00 POWER.CPU     !VERSION 52
>>>S/C 0B
>>>W
>>>@CODE03.CMD
>>>L/C/S:0E00 FPSP.CPU      !VERSION 52
>>>L/C/S:1A00 BITFLD.CPU    !VERSION 52
>>>L/C/S:1D00 CM.CPU        !VERSION 52
>>>L/C/S:2200 IRDFPA.CPU    !VERSION 52
>>>L/C/S:3B00 QUEUE.CPU     !VERSION 52
>>>L/C/S:4000 IDC.CPU       !VERSION 52
>>>I
>>>


BOOT IN PROGRESS

>>>@DD1:DEFBOO.CMD
>>>I
>>>D/G/L 0 00A80003
>>>D/G/L 1 3
>>>D/G/L 2 3FB86
>>>D/G/L 3 0
>>>D/G/L 4 0
>>>D/G/L 5 0
>>>E SP
G 0000000E 00000200
>>>L/P/S:@ VMB.EXE
>>>S @


  VAX/VMS Version V3.0 26-APR-1982 16:21


%OPCOM, 30-JUN-1982 12:58:11.94, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
%OPCOM, 30-JUN-1982 12:58:33.89, message from user NETACP
DECnet starting
%RUN-S-PROC_ID, identification of created process is 00010040
%PURGE-W-SEARCHFAIL, error searching for SYS$SYSROOT:[SYSEXE]RTTLOAD.LOG;*
-RMS-E-FNF< file not found
%PURGE-I-NOFILPURG, no files purged for SYS$SYSROOT:[SYSEXE]RTTLOAD.LOG;
  Login quotas - Interactive limit=64, Current interactive value=0
  SYSTEM       Job terminated at 30-JUN-1982 12:58:46.20
```

Example 6-4 Default Start-Up of a VAX-11/730

Table 6-5 Starting Up a VAX/VMS System from Power On, Using the Default System Device

	VAX-11/780	VAX-11/750 No Console Volume	VAX-11/750 Using Console Volume	VAX-11/730
Step 1	Set AUTO RESTART switch to OFF. Place console volume in console device.	Set POWER ON ACTION switch to HALT. Set BOOT DEVICE switch to default setting for system device (B, C, or D).	Set POWER ON ACTION switch to HALT. Place console volume in console device. Set BOOT DEVICE switch to setting for console device, A.	Set AUTO RESTART/ BOOT switch to OFF. (Internal console volume is always in internal console device.)
Step 2	Set key switch to LOCAL.	Set key switch to LOCAL.	Set key switch to LOCAL.	Set key switch to LOCAL.
Step 3	Toggle BOOT switch from OFF to ON. If nothing happens, at console terminal type <CTRL/P>. When >>> prompt appears, type HALT<RET>.	Set POWER ON ACTION switch to BOOT. Press INITIALIZE button.	Set POWER ON ACTION switch to BOOT. Press INITIALIZE button. When the BOOT58> prompt appears, type: BOOT<RET>.	Toggle AUTO RESTART/BOOT switch from OFF to BOOT. If nothing happens, at console terminal type <CTRL/P>. When >>> prompt appears, type B<RET>.
Step 4	Set key switch from LOCAL to LOCAL DISABLE. Set AUTO RESTART switch to ON.	Set key switch from LOCAL to SECURE. Set POWER ON ACTION switch to RESTART/ BOOT.	Set key switch from LOCAL to SECURE. Set BOOT DEVICE switch to default setting for system device (B, C, or D), and set POWER ON ACTION switch to RESTART/BOOT.	Set key switch from LOCAL to LOC DSBL. Set AUTO RESTART/BOOT switch to ON.

Automatic Restart

- Battery backup makes system recovery possible after power failure
- Recovery can occur automatically
- If memory is valid after recovery, all processes and batch jobs resume where they left off
- If memory is not valid after recovery, system is initialized

Table 6-6 Automatic Powerfail Recovery

Restart Response After Power Failure	VAX-11/780 AUTO RESTART Switch Setting	VAX-11/750 POWER ON ACTION Switch Setting	VAX-11/730 AUTO RESTART/BOOT Switch Setting
If memory is valid, try to recover from system failure. Otherwise, perform fresh start-up.	ON	RESTART/BOOT	ON
If memory is valid, try to recover from system failure. Otherwise, halt processor.	----	RESTART/HALT	----
Halt processor.	OFF	HALT	OFF
Perform fresh start-up.	----	BOOT	----

Alternative System Devices

- Can specify system device explicitly at system start-up
  - Console volume not customized for default start-up
  - Override customized default
- Device codes used at start-up are NOT necessarily standard VMS device codes

Table 6-7 Starting Up a VAX/VMS System, Explicitly Specifying the System Device

	VAX-11/780	VAX-11/750 no console volume (requires ROM for specified device)	VAX-11/750 Using Console Volume	VAX-11/730
Step 1	Set AUTO RESTART switch to OFF. Place console volume in console device.	Set POWER ON ACTION switch to HALT.	Set POWER ON ACTION switch to HALT. Place console volume in console device. Set BOOT DEVICE switch to setting for console device, A.	Set AUTO RESTART/ BOOT switch to OFF. (Internal console volume is always in internal console device.)
Step 2	Set key switch to LOCAL.	Set key switch to LOCAL.	Set key switch to LOCAL.	Set key switch to LOCAL.
Step 3	If >>> prompt is not displayed on console terminal, obtain it by typing <CTRL/P>. Type HALT <RET>. Then, type BOOT ddu <RET>, where ddu is the start-up code for the system device.	If >>> prompt is not displayed on console terminal, obtain it by typing <CTRL/P>. Type B ddcu <RET>, where ddcu is the start-up code for the system device.	Set POWER ON ACTION switch to BOOT. Press INITIALIZE button. When the BOOT58> prompt appears on console terminal, type BOOT ddcu <RET>, where ddcu is the start-up code for the system device.	If >>> prompt is not displayed on console terminal, obtain it by typing <CTRL/P>. Type B ddu <RET>, where ddu is the start-up code for the system device.
Step 4	Set key switch from LOCAL to LOCAL DISABLE. Set AUTO RESTART switch to ON.	Set key switch from LOCAL to SECURE. Set BOOT DEVICE switch to default setting for system device (B, C, or D), and set POWER ON ACTION switch to RESTART/ BOOT.	Set key switch from LOCAL to SECURE. Set BOOT DEVICE switch to default setting for system device (B, C, or D), and set POWER ON ACTION switch to RESTART/BOOT.	Set key switch from LOCAL to LOC DSBL. Set AUTO RESTART/BOOT switch to ON.

Table 6-8 Device Codes Used at System Start-up

Device	VAX-11/780 Support	VAX-11/750 Support	VAX-11/730 Support	Device Code
(MASSBUS)				DB
RP05	S	----	----	
RP06	S	D	----	
RM03	S	S	----	
RM05	S	S	----	
RM80	S	S	----	
RP07	S	----	----	
(UNIBUS)				DM
RK06	D	----	----	
RK07	S	S	D	
(UNIBUS)				DL
RL02	D	D	D	
(IDC)				DQ
RL02	----	----	S	
(IDC)				DQ
R80	----	----	S	
(CONSOLE DEVICE)				CS
RX01	D	----	----	
(CONSOLE DEVICE)				CS
TU58	----	D	D	
				DD

S = Device supported as system device or data device
D = Device supported as data device
---- = Device not supported

For the VAX-11/780 and VAX-11/730, the full device code format is ddu, where dd is the code in the above table and u is the unit number.

For the VAX-11/750, the full device code format is ddcu, where dd is the code in the above table, c is the channel letter, and u is the unit number. The channel letter, which is hardware dependent, is usually A.

```
>>>BOOT DB0

        CPU HALTED
        INIT SEQ DONE
        HALT INST EXECUTED
        HALTED AT 200034F9

        G  0000000E  00000200
        LOAD DONE, 00004200 BYTES LOADED

  VAX/VMS Version V3.0 26-APR-1982 16:21


%OPCOM, 27-JUN-1982 18:28:27.82, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
  Login quotas - Interactive limit=64, Current interactive value=0
  SYSTEM       Job terminated at 27-JUN-1982 18:28:46.69
```

Example 6-5 Start-Up of a VAX-11/780 Specifying an RP05 as the System Device

```
>>>B DBA0
%%

  VAX/VMS Version V3.0 26-APR-1982 16:21

%OPCOM, 30-JUN-1982 17:27:47.31, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
  Login quotas - Interactive limit=32, Current interactive value=0
  SYSTEM       Job terminated at 30-JUN-1982 17:28:19.79
```

Example 6-6 Start-Up of a VAX-11/750 Specifying an RM03 as the System Device

```
>>>D DQO
>>>@DD1:DQOBOO.CMD
>>>I
>>>D/G/L 0 00A80003
>>>D/G/L 1 3
>>>D/G/L 2 3FB86
>>>D/G/L 3 0
>>>D/G/L 4 0
>>>D/G/L 5 0
>>>E SP
G 0000000E 00000200
>>>L/P/S:@ VMB.EXE
>>>S @

  VAX/VMS Version V3.0 26-APR-1982 16:21


%OPCOM, 30-JUN-1982 12:56:10.25, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
%OPCOM, 30-JUN-1982 12:56:32.22, message from user NETACP
DECnet starting
%RUN-S-PROC_ID, identification of created process is 00010040
%PURGE-W-SEARCHFAIL, error searching for SYS$SYSROOT:[SYSEXE]RTTLOAD.LOG;*
-RMS-E-FNF, file not found
%PURGE-I-NOFILPURG, no files purged for SYS$SYSROOT:[SYSEXE]RTTLOAD.LOG;
  Login quotas - Interactive limit=64, Current interactive value=0
  SYSTEM       Job terminated at 30-JUN-1982 12:56:44.43
```

Example 6-7 Start-Up of a VAX-11/730 Specifying an RL02 as the System Device

Stand-Alone Utilities and Diagnostics

- Can run on VAX processor without VMS
  - Hardware diagnostics
  - VMS utilities
- Started up like VMS, using console
  - Larger selection of start-up devices
  - Use of console device common
- Example: Stand-alone BACKUP
  - Packaged on two RX01 diskettes or TU58 cartridges
  - Started up from console device
  - Command to issue at console

VAX-11/780	BOOT CS1
VAX-11/750 (CCL)	B DDA0
VAX-11/750 (BOOT58)	BOOT CSA1
VAX-11/730	B CS1

CONSOLE COMMANDS

- Each VAX processor has its own CCL

Table 6-9 Typical Console Commands

Function of Command	VAX-11/780 CCL	VAX-11/750 CCL	VAX-11/750 BOOT58	VAX-11/730 CCL
Starting up VMS or a stand-alone utility or diagnostic	BOOT	B	BOOT	B
Examining a processor register or memory location	EXAMINE	E	EXAMINE	E
Placing data in a processor register or memory location	DEPOSIT	D	DEPOSIT	D
Initializing state of VAX processor	INITIALIZE	I	----	I
Executing commands in a file	@file-spec	----	@file-spec	@file-spec
Reloading and restarting the console processor	REBOOT	----	----	----

Issuing Console Commands at System Start-Up

- For the start-up methods covered in this module, can issue CCL commands at the end of step 2, in response to >>> prompt
- In the case of starting up the VAX-11/750 using the console volume, can issue BOOT58 commands at the end of step 2, in response to BOOT58> prompt
- Console terminal serves as CCL command terminal at start-up

Issuing Console Commands While VMS Is Running

- Console terminal used
  - Program mode - Interactive VMS terminal
  - Console mode - CCL command terminal
- When the console terminal is in console mode:
  - VAX-11/750 and VAX-11/730 processors halt
  - VAX-11/780 processor continues to run

Table 6-10 Issuing Console Commands While VMS Is Running

Function of Command	VAX-11/780 CCL	VAX-11/750 CCL	VAX-11/730 CCL
Changing mode of console terminal from program mode to console mode	<CTRL/P> (does not halt VAX processor)	<CTRL/P> (halts VAX processor)	<CTRL/P> (halts VAX processor)
Changing mode of console terminal from console mode to program mode	SET TERMINAL PROGRAM	C (restarts halted VAX processor)	C (restarts halted VAX processor)
Halting VAX processor	HALT	----	----
Restarting halted VAX processor without taking console terminal out of console mode	CONTINUE	----	----

- Disabling the use of console mode

Table 6-11 Disabling the Use of Console Mode

Function of Keyswitch Setting	VAX-11/780	VAX-11/750	VAX-11/730
Enables <CTRL/P>	LOCAL	LOCAL	LOCAL
Disables <CTRL/P>	LOCAL DISABLE	SECURE	LOC DSBL

Customizing Start-Up and Automatic Restart

- On VAX-11/750, default system device specified by setting BOOT DEVICE switch
- On VAX-11/780 and VAX-11/730, default system device specified by customizing CCL command file DEFBOO.CMD on console volume
- On VAX-11/750 started up using console volume, default system device specified by customizing BOOT58 command file DEFBOO.CMD on console volume
- DIGITAL supplies many command files on console volume that can be chosen for DEFBOO.CMD, using command procedure SYS$UPDATE:SETDEFBOO.COM

Table 6-12 CCL Command Files Used at Start-up

Device Code Used at System Start-up (See Table 6-8)	Command Files on Console Volume Executed by BOOT Command
DB	DBnBOO.CMD (n = 0,1,...,7)
DM	DMnBOO.CMD (n = 0,1,2,3)
DL	DLnBOO.CMD (n = 0,1,2,3)
DQ	DQnBOO.CMD (n = 0,1,2,3)
CS	CS1BOO.CMD

```
!
!       DB0 BOOT COMMAND FILE - DB0B00.CMD
!
HALT                        ! HALT PROCESSOR
UNJAM                       ! UNJAM SBI
INIT                        ! INIT PROCESSOR
DEPOSIT/I 11 20003800       ! SET UP SCBB
DEPOSIT R0 0                ! DISK PACK DEVICE TYPE
DEPOSIT R1 8                ! MBA TR=8
DEPOSIT R2 0                ! ADAPTER UNIT = 0
DEPOSIT R3 0                ! CONTROLLER UNIT = 0
DEPOSIT R4 0                ! BOOT BLOCK LBN (UNUSED)
DEPOSIT R5 0                ! SOFTWARE BOOT FLAGS
DEPOSIT FP 0                ! SET NO MACHINE CHECK EXPECTED
START 20003000              ! START ROM PROGRAM
WAIT DONE                   ! WAIT FOR COMPLETION
                            !
EXAMINE SP                  ! SHOW ADDRESS OF WORKING MEMORY+^X200
LOAD VMB.EXE/START:@        ! LOAD PRIMARY BOOTSTRAP
START @                     ! AND START IT
```

Example 6-8 File DB0B00.CMD from a VAX-11/780 Console Volume

```
$ @SYS$UPDATE:SETDEFBOO

        Command file to set default boot command file.

%SYSTEM-F-DEVNOTMOUNT, device is not mounted
Please place the system console medium in the console drive.
Is the console medium ready to be mounted? (Y/N): Y
%MOUNT-I-MOUNTED, CONSOLE       mounted on _CSA1:
Enter name of default boot command file: DBOBOO.CMD
Default boot command file now replaced with DBOBOO.CMD.

Are you satisfied with this file as the default boot command file? (Y/N): Y

$
```

Example 6-9 Customizing the Default System Device for Start-Up

- On VAX-11/780, additional CCL command file RESTAR.CMD used for powerfail restart.

```
!
!       RESTART COMMMAND FILE - RESTAR.CMD
!
!       THIS COMMAND FILE IS INVOKED IN THE EVENT OF POWER RECOVERY AND
!       OTHER CONSOLE DETECTED RESTART CONDITIONS IF THE AUTO RESTART SWITCH
!       IS SET.  IT CAN ALSO BE INVOKED MANUALLY WITH THE COMMAND:
!                               @RESTAR.CMD
!
HALT                        ! HALT PROCESSOR
UNJAM                       ! UNJAM SBI
INIT                        ! INITIALIZE PROCESSOR
DEPOSIT/I 11 20003800       ! SET ADDRESS OF SCB BASE
DEPOSIT R0 0                ! CLEAR UNUSED REGISTERS
DEPOSIT R1 3                ! UBA TR=3
DEPOSIT R2 0                ! CLEAR UNUSED REGISTER
DEPOSIT R3 0                ! CLEAR UNUSED REGISTER
DEPOSIT R4 0                ! CLEAR UNUSED REGISTER
DEPOSIT R5 0                ! CLEAR UNUSED REGISTER
DEPOSIT FP 0                ! NO MACHINE CHECK EXPECTED
START 20003004              ! START RESTART REFEREE
```

Example 6-10 File RESTAR.CMD from a VAX-11/780 Console Volume

SYSTEM PARAMETERS

- Stored in SYS$SYSTEM:SYS.EXE
- Used to build data structures, configure peripheral devices, and customize user interface each time the system starts up
- If modifications are required, usually accomplished by SYSGEN utility and SYS$UPDATE:AUTOGEN.COM command procedure, while VMS is running
- To make modifications effective at start-up, use SYSBOOT program during conversational start-up

Using SYSBOOT to Modify System Parameters

Table 6-13 Using SYSBOOT During Conversational Start-Up

Function	Command Format and Examples
Examining a system parameter or group of parameters	SYSBOOT> SHOW parameter SYSBOOT> SHOW /parameter-group SYSBOOT> SHOW MAXPROCESSCNT SYSBOOT> SHOW /ALL
Modifying a system parameter	SYSBOOT> USE parameter value SYSBOOT> USE UAFALTERNATE 1
Modifying a group of system parameters (.PAR files should be in SYS$SYSTEM)	SYSBOOT> USE parameter-file.PAR SYSBOOT> USE CURRENT SYSBOOT> USE DEFAULT SYSBOOT> USE ALTPARAM.PAR
Using an alternate DCL start-up file	SYSBOOT> SET/STARTUP SYS$SYSTEM:startup-file SYSBOOT> SET/STARTUP SYS$SYSTEM:ALTSTART
Exiting from SYSBOOT	CONTINUE EXIT

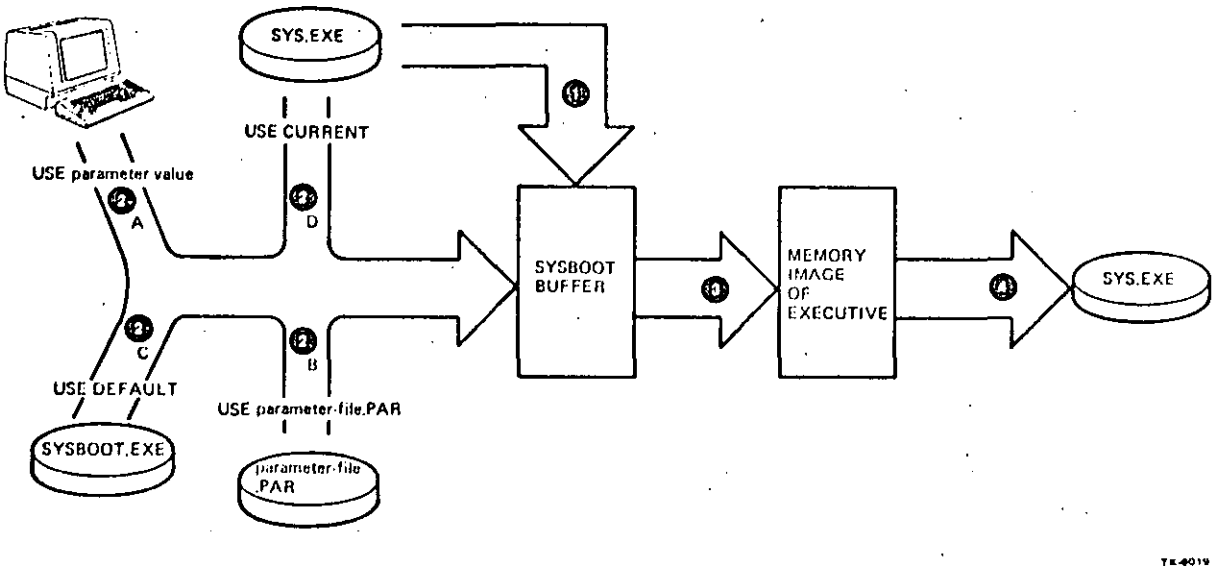


Figure 6-3 Effect of VMS Start-Up on System Parameters

Table 6-14 Starting Up a VAX/VMS System Conversationally

	VAX-11/780	VAX-11/750 No Console Volume	VAX-11/750 Using Console Volume	VAX-11/730
Step 1	Set AUTO RESTART switch to OFF. Place console volume in console device.	Set POWER ON ACTION switch to HALT. Set BOOT DEVICE switch to default setting for system device (B, C, or D).	Set POWER ON ACTION switch to HALT. Place console volume in console device. Set BOOT DEVICE switch to setting for console device, A.	Set AUTO RESTART/ BOOT switch to OFF. (Internal console volume is always in internal console device.)
Step 2	Set key switch to LOCAL.	Set key switch to LOCAL.	Set key switch to LOCAL.	Set key switch to LOCAL.
Step 3	If >>> prompt is not displayed on console terminal, obtain it by typing <CTRL/P>. Type @dduGEN<RET>, where ddu is the start-up code for the system device.	If >>> prompt is not displayed on console terminal, obtain it by typing <CTRL/P>. Type B/1 <RET>.	Set POWER ON ACTION switch to BOOT. Press INITIALIZE button. When the BOOT58> prompt appears on console terminal, type D/G 3 u <RET> @ddcGEN <RET> where ddcu is the start-up code for the system device.	If >>> prompt is not displayed on console terminal, obtain it by typing <CTRL/P>. Type @dduGEN<RET>, where ddu is the start-up code for the system device.
Step 4	Set key switch from LOCAL to LOCAL DISABLE. Set AUTO RESTART switch to ON.	Set key switch from LOCAL to SECURE. Set POWER ON ACTION switch to RESTART/ BOOT.	Set key switch from LOCAL to SECURE. Set BOOT DEVICE switch to default setting for system device (B, C, or D), and set POWER ON ACTION switch to RESTART/BOOT.	Set key switch from LOCAL to LOC DSBL. Set AUTO RESTART/BOOT switch to ON.

```
>>>@DBOGEN
!
!       DBO CONVERSATIONAL BOOT COMMAND FILE - DBOGEN.
!       BOOT FROM DBO AND STOP IN SYSBOOT TO ALTER PARAMETERS
!
HALT                    ! HALT PROCESSOR
        CPU HALTED
UNJAM                   ! UNJAM SBI
INIT                    ! INIT PROCESSOR
        INIT SEQ DONE
DEPOSIT/I 11 20003800   ! SET UP SCBB
DEPOSIT R0 0            ! DISK PACK DEVICE TYPE
DEPOSIT R1 8            ! MBA TR=8
DEPOSIT R2 0            ! ADAPTER UNIT = 0
DEPOSIT R3 0            ! CONTROLLER UNIT = 0
DEPOSIT R4 0            ! BOOT BLOCK LBN (UNUSED)
DEPOSIT R5 1            ! SOFTWARE BOOT FLAGS (CONVERSATIONAL BOOT)
DEPOSIT FP 0            ! SET NO MACHINE CHECK EXPECTED
START 20003000          ! START ROM PROGRAM
WAIT DONE               ! WAIT FOR COMPLETION
        HALT INST EXECUTED
        HALTED AT 200034F9
                        !
EXAMINE SP              ! SHOW ADDRESS OF WORKING MEMORY+^X200
        0  0000000E  00000200
LOAD VMB.EXE/START:0    ! LOAD PRIMARY BOOTSTRAP
        LOAD DONE, 00004200 BYTES LOADED
START 0                 ! AND START IT
<@EOF>
<@EXIT>
SYSBOOT>  USE ALTPARAM.PAR
SYSBOOT>  SET UAFALTERNATE 1
SYSBOOT>  SET/STARTUP SYS$SYSTEM:ALTSTART
SYSBOOT>  CONTINUE

  VAX/VMS Version V3.0 26-APR-1982 16:21

%OPCOM, 27-JUN-1982 19:06:13.68, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
  Login quotas - Interactive limit=64, Current interactive value=0
  SYSTEM       Job terminated at 27-JUN-1982 19:06:32.72
```

Example 6-11 Typical Conversational Start-Up

SPECIFYING THE SYSTEM CONFIGURATION

- At each start-up:
  - The system device is selected
  - VMS data base is constructed according to system parameters
  - System environment created by DCL commands
- Files determine the default system configuration; must be customized

Table 6-15 Customizing the System Configuration

Customize	Contents of File	Location of File	Utility Used for Customization
DEFBOO.CMD	CCL (or BOOT58) commands	Console volume	Editor
SYS.EXE	System parameters	SYS$SYSTEM	SYSBOOT or SYSGEN
SYSTARTUP.COM	DCL commands	SYS$MANAGER	Editor

VAX-11/782 MULTIPROCESSING SYSTEM

Introduction

- Attached processor system
  - Two VAX-11/780 processors connected by shared memory
- Primary processor
  - Runs VMS
  - Attached to all peripheral devices
- Attached processor
  - Used by primary processor to increase computing power only
  - Has no peripheral devices of its own
  - Operates under the control of the primary processor

Starting Up the VAX-11/782

- Customization of start-up files on VAX-11/782
  - Special multiprocessing console volume created for the primary processor
  - Special multiprocessing console volume created for the attached processor
  - CCL command files customized for multiprocessor operation to load all software into shared memory
  - SYS$MANAGER:SYSTARTUP.COM is customized for VAX-11/782
- Start-up of VAX-11/782
  - Start-up of primary processor same as for single VAX-11/780
  - Wait for message on primary processor console terminal
  - Start-up of attached processor same as DEFAULT start-up of single VAX-11/780

    Cannot specify alternative system device

    Cannot have conversational start-up

Controlling Multiprocessing

- Primary processor controls and monitors attached processor with three commands

Table 6-16 Controlling Multiprocessing on the VAX-11/782

Function	Command (Requires CMKRNL Privilege)
Start the attached processor	$ START/CPU
Stop the attached processor	$ STOP/CPU
Display current state of the attached processor	$ SHOW/CPU

SHUTDOWN

- Three methods for system shutdown
  - Orderly shutdown
  - Emergency shutdown
    - From DCL
    - From CCL

- Orderly shutdown
  - Preferred method

    Warns users of approaching shutdown

    Preserves all system and user data
  - Command procedures used

    SYS$SYSTEM:SHUTDOWN.COM (site-independent)

    SYS$MANAGER:SYSHUTDWN.COM (site-specific)
  - Steps to follow

    Log in as SYSTEM

    Issue @SYS$SYSTEM:SHUTDOWN

    Respond to prompting messages

    When message SYSTEM SHUTDOWN COMPLETE is displayed on console terminal, use that terminal to type:

    <CTRL/P>

    >>> H
  - Canceling shutdown

    <CTRL/Y> cancels remaining shutdown functions

    Shutdown functions already accomplished must be manually reversed

- Functions of SHUTDOWN.COM
  - Notifies users
  - Prevents new logins (except with OPER privilege)
  - Shuts down DECnet
  - Stops batch and print queues
  - Executes SYS$MANGER:SYSHUTDWN.COM
  - Stops user processes
  - Removes installed images
  - Dismounts mounted volumes (other than system disk)
  - Closes operator's log file
  - Writes error log entries to disk
  - Records file changes in memory on disk
  - Writes crash dump file
  - Dismounts system disk

```
Username: SYSTEM
Password:
        Welcome to VAX/VMS version V3.0 on node HARDY

$ @SYS$SYSTEM:SHUTDOWN

        System shutdown command procedure.

How many minutes until shutdown [0]? 10
Reason? PREVENTATIVE MAINTENANCE
Do you want to spin down the disks [No]?
Expected uptime (<RET> if not known)? AT 21:00
Enable automatic reboot [No]? N
%OPCOM, 27-JUN-1982 19:34:36.12, operator status for operator OPA0
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, OPER1, OPER2,
OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10, OPER11,
OPER12

--OPA0:,SYSTEM       19:34:36.40
System shutdown in 10 minutes; UP AT 21:00.
PREVENTATIVE MAINTENANCE

--OPA0:,SYSTEM       19:39:39.17
System shutdown in 5 minutes; UP AT 21:00.
PREVENTATIVE MAINTENANCE
  Login quotas - Interactive limit=0, Current interactive value=3
        Non-operator logins are disabled.

--OPA0:,SYSTEM       19:40:43.22
Batch and device queues have been stopped.

--OPA0:,SYSTEM       19:40:44.11
System shutdown in 4 minutes; UP AT 21:00.  Logins are disabled; please log out.
PREVENTATIVE MAINTENANCE

--OPA0:,SYSTEM       19:42:45.63
System shutdown in 2 minutes; UP AT 21:00.  Logins are disabled; please log out.
PREVENTATIVE MAINTENANCE

--OPA0:,SYSTEM       19:43:47.13
System shutdown in 1 minute; UP AT 21:00.  Logins are disabled; please log out.
PREVENTATIVE MAINTENANCE

--OPA0:,SYSTEM       19:44:48.62
System shutdown in 0 minutes; UP AT 21:00.  Logins are disabled; please log out.
PREVENTATIVE MAINTENANCE
        Invoke installation dependent shutdown procedure.
        Stop all user processes.
        Remove installed images.
        Dismount all mounted volumes.
%OPCOM, 27-JUN-1982 19:45:07.97, message from user SYSTEM
--OPA0:, Operator requested shutdown
%OPCOM, 27-JUN-1982 19:45:08.97, logfile closed by operator OPA0
        logfile was SYS$MANAGER:OPERATOR.LOG
        SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM

>>>H

        HALTED AT 80007D3C
>>>
```

Example 6-12 Performing an Orderly System Shutdown

```
__OPA0:,SYSTEM        19:34:36.40
System shutdown in 10 minutes; UP AT 21:00.
PREVENTATIVE MAINTENANCE
$

__OPA0:,SYSTEM        19:39:39.17
System shutdown in 5 minutes; UP AT 21:00.
PREVENTATIVE MAINTENANCE
$

__OPA0:,SYSTEM        19:40:43.22
Batch and device queues have been stopped.
$

__OPA0:,SYSTEM        19:40:44.11
System shutdown in 4 minutes; UP AT 21:00.  Logins are disabled; Please log out.
PREVENTATIVE MAINTENANCE
$

__OPA0:,SYSTEM        19:42:45.63
System shutdown in 2 minutes; UP AT 21:00.  Logins are disabled; Please log out.
PREVENTATIVE MAINTENANCE
$

__OPA0:,SYSTEM        19:43:47.13
System shutdown in 1 minute; UP AT 21:00.  Logins are disabled; Please log out.
PREVENTATIVE MAINTENANCE
$

__OPA0:,SYSTEM        19:44:48.62
System shutdown in 0 minutes; UP AT 21:00.  Logins are disabled; Please log out.
PREVENTATIVE MAINTENANCE
$
```

Example 6-13 User View of Orderly System Shutdown

- Emergency shutdown from DCL
  - Use if orderly shutdown fails
  - Use in emergency for speed
  - For speed when no emergency exists, usually suffices to use orderly shutdown, specifying the number of minutes until shutdown as Ø
  - Steps to follow

    Log in as SYSTEM

    Issue $ RUN SYS$SYSTEM:OPCCRASH

    When message SYSTEM SHUTDOWN COMPLETE is displayed on console terminal, use that terminal to type:

    <CTRL/P>

    >>> H

  - Functions of OPCCRASH (last four SHUTDOWN functions)

    Writes error log entries to disk

    Records file changes in memory on disk

    Writes crash dump file

    Dismounts system disk

- Emergency shutdown from CCL
  - Use if no system response to interactive terminal input
  - Console terminal only
  - Functions

    Writes crash dump file

    Displays register contents on console terminal

Table 6-17 Emergency Shutdown From CCL

VAX-11/780	VAX-11/750	VAX-11/730
<CTRL/P> >>>HALT >>>@CRASH	<CTRL/P> >>>E/G F >>>E P >>>E/I 0 >>>E/I 1 >>>E/I 2 >>>E/I 3 >>>E/I 4 >>>D/G F FFFFFFFF >>>D P 1F0000 >>>C	<CTRL/P> >>>@CRASH

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" Y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO, D.F.

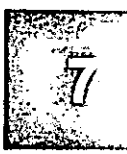
SYSTEM INSTALLATION AND UPDATE

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985

SYSTEM INSTALLATION AND UPDATE



INTRODUCTION

The methods of system start-up in Module 6 assume that the system you are starting up already has a VMS system volume customized for your site. The process of creating a VMS version 3.0 system volume and starting up VMS for the first time is called installing the VMS version 3.0 system.

If you are already running VMS version 2.5 on your VAX-11/780 or VAX-11/750 system, you have already customized a version 2.5 system disk. In creating your version 3.0 system disk, you can preserve many files you have already customized for version 2.5, such as your user authorization file and site-specific start-up file. The process of creating a VMS version 3.0 system volume and starting up VMS for the first time, while preserving as much version 2.5 customization as possible, is called upgrading VMS from version 2.5 to version 3.0.

After system installation or upgrade is complete, you should run the User Environment Test Package (UETP) to confirm that the system volume was successfully built.

If you want to run optional DIGITAL software, such as compilers and data base managers, you must transfer this software from distribution kits to your system volume. This process is called installing VMS optional software.

Maintenance updates are distributed periodically to correct VMS problems that have been discovered. Applying these maintenance updates to the system volume is called updating VMS. For example, VMS version 2.0 was updated five times, the final update resulting in version 2.5. The first update of version 3.0 will produce version 3.1.

OBJECTIVES

To install and test a system, a system manager must:

- Install the base system
- Install updates, optional software, and single node DECnet
- Use the User Environment Test Package (UETP) to test the hardware and software

RESOURCES

1. VAX-11/xxx Software Installation Guide
   (/xxx = /780, /750, or /730)
2. VAX-11/782 User's Guide
3. VAX/VMS UETP User's Guide

INSTALLING OR UPGRADING VMS

To create VMS version 3.Ø system volume:

- Installation and upgrade procedures highly automated
- VAX-11/xxx Software Installation Guide (SIG)
  (/xxx = /78Ø, /75Ø, or /73Ø)

  VAX-11/782 User's Guide
  - Step by step instructions

Overview

1. Build a VMS system volume
2. Customize system volume and console volume to your site

Building a System Volume on a VAX-11/780 or VAX-11/750

1. Get tools

   - Two scratch volumes
     - System volume
     - System volume backup
   - Distribution kit
     - VMS version 3.Ø software (tape volume or RKØ7)
     - Two volumes of console medium (RXØ1 or TU58) for stand-alone BACKUP

       For tape kits stand-alone BACKUP is provided

       For RKØ7 kits stand-alone BACKUP must be built
     - VMS version 3.Ø console volume (RXØ1 or TU58)
     - Mandatory security update (console medium RXØ1 or TU58)

2. Physically mount VMS distribution volume and start up stand-alone BACKUP

   - Magnetic tape kit
     - Use console device
     - Two BACKUP volumes in distribution kit
   - RKØ7 kit
     - Start up stand-alone BACKUP from RKØ7
   - Procedure in SIG, Chapter 2

3. Physically mount scratch volume and create kernel VMS system

   - Use stand-alone BACKUP
   - Restore save-sets from distribution volume
     - Required save-set
     - Library save-set
     - Optional save-set
   - Procedure in SIG, Chapter 2

4. At DCL prompt ($), halt processor

   <CTRL/P>

   >>>H

5. Start up kernel VMS system

   - Make sure console volume is mounted
   - Procedure in SIG, Chapter 2, or Table 6-7

6. Respond to prompts for:

   - Time
   - Drive holding VMS distribution volume

7. At message SYSTEM SHUTDOWN COMPLETE, halt processor:

   <CTRL/P>

   >>>H

```
        CPU HALTED,SOMM CLEAR,STEP=NONE,CLOCK=NORM
        RAD=HEX,ADD=PHYS,DAT=LONG,FILL=00,REL=00000000
        INIT SEQ DONE
        HALTED AT 00000000

        (RELOADING WCS)
        LOAD DONE, 00003200 BYTES LOADED
        VER: PCS=01 WCS=0D-13 FPLA=0D CON=V04-03-L
>>>B CSI

        CPU HALTED
        INIT SEQ DONE
        LOAD DONE, 00004200 BYTES LOADED
Please mount first standalone system diskette or cartridge and press return


  VAX/VMS Version V3.0 26-APR-1982 16:21

Please mount second standalone system diskette or cartridge and press return
SYSGENZW-NOSUCHFILE, file not found
DYDRIVER.EXE
SYSGENZW-NOSUCHFILE, file not found
DYDRIVER.EXE
SYSGENZW-NOSUCHFILE, file not found
DDDRIVER.EXE
SYSGENZW-NOSUCHFILE, file not found
DDDRIVER.EXE
$ BACKUP/VERIFY MTA0:REQUIRED/SAVE DRA0:
%BACKUP-I-READYREAD, mount volume 1 on _MTA0: for reading
Press return when ready:
%BACKUP-I-STARTVERIFY, starting verification pass
$
>>>H

        HALTED AT 80007B06
```

Example 7-1 Installing VMS (Sheet 1 of 2)

```
>>>B DB0

        CPU HALTED
        INIT SEQ DONE
        HALT INST EXECUTED
        HALTED AT 200034F9

        G  0000000E  00000200
        LOAD DONE, 00004200 BYTES LOADED

  VAX/VMS Version V3.0 26-APR-1982 16:21


%OPCOM, 27-JUN-1982 15:43:30.49, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
  Login quotas - Interactive limit=64, Current interactive value=0
PLEASE ENTER DATE AND TIME (DD-MMM-YYYY HH:MM) 27-JUN-1982 15:45

Enter drive holding distribution kit (DDCU): MTA0
%MOUNT-I-MOUNTED, VMS300 mounted on _MTA0:

Restoring library saveset.
%BACKUP-I-STARTVERIFY, starting verification pass

Restoring optional saveset.
%BACKUP-I-STARTVERIFY, starting verification pass
Running AUTOGEN -- Please wait.
%SYSGEN-I-EXTENDED, SYS$SYSROOT:[SYSEXE]PAGEFILE.SYS;1 extended
%SYSGEN-I-CREATED, SYS$SYSROOT:[SYSEXE]SWAPFILE.SYS;1 created
%SYSGEN-I-CREATED, SYS$SYSROOT:[SYSEXE]SYSDUMP.DMP;1 created
%OPCOM, 27-JUN-1982 15:55:00.80, message from user SYSTEM
%SYSGEN-I-WRITECUR, CURRENT system parameters modified by process ID 00010045 into file SYS$SYSROOT:[SYSEXE]SYS.EXE;1

System shutting down to allow boot of complete V3.0 system,
with generated site-specific parameters and installed images.
You must manually reboot the system after it halts.

        System shutdown command procedure.

  Login quotas - Interactive limit=0, Current interactive value=0
        Non-operator logins are disabled.
%JBC-E-SYMBDSAB, symbiont manager is disabled

Batch,SYSTEM        15:55:08.15
Batch and device queues have been stopped.

Batch,SYSTEM        15:55:09.02
System shutdown in 0 minutes! up soon.  Logins are disabled! please log out.
Reboot system with generated parameters
        Stop all user processes.
        Remove installed images.
        Dismount all mounted volumes.
%OPCOM, 27-JUN-1982 15:55:28.14, message from user SYSTEM
%STARTUP, Operator requested shutdown
        SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM

>>>H

        HALTED AT 80007D3C
>>>
```

Example 7-1 Installing VMS (Sheet 2 of 2)

Customizing the System Volume and Console Volume on a VAX-11/780 or VAX-11/750

1. Start up complete VMS system
   - See Table 6-7
2. Log in as SYSTEM (password MANAGER)
3. Install mandatory security update
   - @VMSUPDATE
     - See SIG, Appendix C, and Example 7-3
4. Customize default bootstrap file
   - @SETDEFBOO
     - See SIG, Chapter 3, and Example 6-9
5. Customize site-specific files on system disk
   - See SIG, Chapter 3
   - Start-up file (Module 5)
   - UAF and UFDs (Module 2)
6. If system parameter values must be changed
   - @AUTOGEN
   - See SIG, Chapter 3, and Table 9-10
7. Shut down the system
   - @SHUTDOWN
   - When system comes back up, changes in effect

```
>>>B DB0

        CPU HALTED
        INIT SEQ DONE
        HALT INST EXECUTED
        HALTED AT 200034F9

        @  0000000E  00000200
        LOAD DONE, 00004200 BYTES LOADED

  VAX/VMS Version V3.0 26-APR-1982 16:21


%OPCOM, 27-JUN-1982 16:31:26.63, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
  Login quotas - Interactive limit=64, Current interactive value=0
  SYSTEM       Job terminated at 27-JUN-1982 16:31:45.47

Username: SYSTEM
Password:
        Welcome to VAX/VMS version V3.0
$ @SYS$UPDATE:VMSUPDATE

                    VMS Update Procedure

 This command procedure performs VAX/VMS software updates and
 optional software installations for VAX/VMS Release 3.
 During this sequence, the standard console medium will not
 be present in the console drive. Therefore, the system may
 be vulnerable to a power failure or other fatal crash. If a
 system crash should occur during this period the update
 sequence can be restarted at the beginning of the first
 incomplete update.

 Dismount the current console medium.

 Please place the first volume in the console drive

Are you ready to continue?: Y
%MOUNT-I-MOUNTED, VMSV300      mounted on _CSA1:

        VAX/VMS Version 3.00 Update

Do you want all the updates applied? (Y/N, YES recommended): Y
Do you want an explanation of each update displayed during the update? (Y/N): Y
Do you want previous versions of updated files purged? (Y/N): Y

%PATCH-I-NOLCL, image does not contain local symbols
%PATCH-I-NOGBL, some or all global symbols not accessible
%PATCH-I-WRTFIL, updating image file SYS$SYSROOT:[SYSEXE]LOGINOUT.EXE;2
%PATCH-I-WRTFIL, updating image file SYS$SYSROOT:[SYSEXE]LOGINOUT.EXE;2
%PATCH-I-NOLCL, image does not contain local symbols
%PATCH-I-NOGBL, some or all global symbols not accessible
%PATCH-I-WRTFIL, updating image file SYS$SYSROOT:[SYSEXE]SYS.EXE;2
%PATCH-I-WRTFIL, updating image file SYS$SYSROOT:[SYSEXE]SYS.EXE;2

Are there more kits to process?: N

 Please place the system console medium in the console drive.
```

Example 7-2 Installing VMS (Sheet 1 of 2)

```
Are you ready to continue?: Y
%MOUNT-I-MOUNTED, CONSOLE       mounted on _CSA1:
 Requested update sequence is complete.
$ @SYS$UPDATE:SETDEFBOO

        Command file to set default boot command file.

Please place the system console medium in the console drive.
Is the console medium ready to be mounted? (Y/N): Y
%MOUNT-I-MOUNTED, CONSOLE       mounted on _CSA1:
Enter name of default boot command file: DB0BOO.CMD
Default boot command file now replaced with DB0BOO.CMD.

Are you satisfied with this file as the default boot command file? (Y/N): Y

$ @SYS$SYSTEM:SHUTDOWN

        System shutdown command procedure.

How many minutes until shutdown [0]? 0
Reason?,
Do you want to spin down the disks [No]? N
Expected uptime (<RET> if not known)?
Enable automatic reboot [No]? Y
%OPCOM, 27-JUN-1982 16:52:17.71, operator status for operator OPA0
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, OPER1, OPER2,
OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10, OPER11,
OPER12
  Login quotas - Interactive limit=0, Current interactive value=1
        Non-operator logins are disabled.

__OPA0:,SYSTEM        16:52:21.84
Batch and device queues have been stopped.

__OPA0:,SYSTEM        16:52:22.69
System shutdown in 0 minutes.  Logins are disabled! Please log out.
        Invoke installation dependent shutdown procedure.
        Stop all user processes.
        Remove installed images.
        Dismount all mounted volumes.
%OPCOM, 27-JUN-1982 16:52:41.63, message from user SYSTEM
__OPA0:, Operator requested shutdown
%OPCOM, 27-JUN-1982 16:52:42.61, logfile closed by operator OPA0
        logfile was SYS$MANAGER:OPERATOR.LOG
        HALT INST EXECUTED
        HALTED AT 800039ED

        (BOOTING)
        CPU HALTED
        INIT SEQ DONE
        HALT INST EXECUTED
        HALTED AT 200034F9

        G  0000000E  00000200
        LOAD DONE, 00004200 BYTES LOADED

  VAX/VMS Version V3.0 26-APR-1982 16:21


%OPCOM, 27-JUN-1982 16:53:42.50, logfile initialized by operator OPA0
        logfile is SYS$MANAGER:OPERATOR.LOG
  Login quotas - Interactive limit=64, Current interactive value=0
  SYSTEM        Job terminated at 27-JUN-1982 16:54:01.33
```

Example 7-2 Installing VMS (Sheet 2 of 2)

UPGRADING VMS

- Upgrading from VMS version 2.5 to version 3.Ø
- @SYS$UPDATE:VMSINSTAL
- See SIG, Chapter 4

Additional Steps After Installation or Update

1. Install DECnet
   - Single node
   - Multinode
2. Run user environment test package (UETP)
3. For RKØ7 distribution kit, build standalone BACKUP
   - @STABACKIT
   - Use scratch console medium (RXØ1 or TU58)
   - See SIG, Chapter 6
4. Back up new system disk
   - See SIG, Chapter 3
5. Back up new console volume
   - See SIG, Chapter 6

INSTALLING VMS ON A VAX-11/730

- Distribution medium is RLØ2 disk volume
- Distribution volume contains only REQUIRED and LIBRARY save-sets
- Must purchase optional additional distribution volume to obtain OPTIONAL save set and EDTCAI computer-based course

Systems With an R80 System Disk

- All of VMS fits on system volume
- Kernel system automatically restores LIBRARY save-set to system volume; optionally restores OPTIONAL save-set as well

Systems With an RL02 System Disk

- REQUIRED save-set and some of LIBRARY save-set fits on system volume
- With full contents of LIBRARY save-set available if needed on second RLØ2, have subset of VMS suitable for most applications
- If second RLØ2 needed for user applications, can tailor the subset of the LIBRARY subset available on system volume
- Can record alternative tailored configurations and move between them as needed
- Tailoring controlled by command procedure SYS$UPDATE:VMSTAILOR. See SIG, Chapter 6, Dual RLØ2 Operations

INSTALLING VMS ON A VAX-11/782 MULTIPROCESSING SYSTEM

1. Follow standard VMS installation or upgrade on primary processor
   - Treat as independent VAX-11/780
2. Log in as SYSTEM
3. Create mutliprocessor console volumes
   - @SYS$UPDATE:BOOTBLDR
   - See the VAX-11/782 User's Guide
4. Shut down primary processor
5. Load new multiprocessing console volumes
6. Start up primary processor
7. Log in as SYSTEM
8. Issue DCL command START/CPU
9. Start attached processor
10. Add the following lines to SYS$MANAGER:SYSTARTUP.COM

```
$ START/CPU
$ WRITE SYS$OUTPUT "You can start up"
$ WRITE SYS$OUTPUT "the attached processor now."
```

11. For more information see the VAX-11/782 User's Guide, Chapter 2

ACTIVATING SINGLE-NODE DECnet ON VMS

- After standard installation or upgrade, single-node DECnet is present but not active

- To activate DECnet:

1. Choose node name (up to six characters)

2. Log in as SYSTEM

3. Use AUTHORIZE to create UAF records NETNONPRIV and NETPRIV:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF>ADD NETNONPRIV /PASS=nonpriv-passwd -
_/DEVICE=user-device: /DIRECTORY=[NETUSER] -
_/UIC=[200,200] /PRIV=(TMPMBX,NETMBX) -
_/FLAGS=(CAPTIVE,DISCTLY,LOCKPWD) /BYTLM=6000 -
_/LGICMD=SYS$SYSTEM:NETLOGIN
UAF>COPY NETNONPRIV NETPRIV /PASS=priv-passwd -
_/PRIV=(TMPMBX,NETMBX,OPER,DIAGNOSE)
UAF>EXIT
```

4. Create the directory referred to in these UAF records:

```
$ CREATE/DIRECTORY/OWNER_UIC=[200,200] -
$_user-disk:[NETUSER]
```

5. Create the file SYS$SYSTEM:NETLOGIN.COM referred to in these UAF records:

```
$ IF F$MODE() .NES. "NETWORK" THEN LOGOUT
$ EXIT
```

6. Load the network driver:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT NET/NOADAPTER/DRIVER=NETDRIVER
SYSGEN> EXIT
```

7. Modify the DECnet configuration data base:

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE EXECUTOR NAME node-name -
_ NONPRIVILEGED USER NETNONPRIV PASSWORD nonpriv-passwd -
_ PRIVILEGED USER NETPRIV PASSWORD priv-passwd -
NCP>DEFINE EXECUTOR ADDRESS 1
NCP>EXIT
```

8. Add the following command to your site-specific start-up file, SYS$MANAGER:SYSTARTUP.COM:

```
$ @STARTNET
```

- Above steps configure DECnet for single node communications
- STARTNET command procedure activates DECnet on the system at start-up

RUNNING UETP AFTER INSTALLING VMS

- User Environment Test Package (UETP) tests
  - All standard peripheral devices
  - Various native mode functions
  - Most compatibility mode utilities
  - System multiuser capability
  - DECnet VAX
- To use UETP
  - Log in with user name SYSTEST
  - Prepare devices for testing

    See VAX/VMS UETP User's Guide, Chapters 1 and 2
  - Execute UETP:

    $ @UETP
  - At UETP prompts, press <RET>
  - If UETP completes successfully, run SYE utility (Module 8)
  - If UETP does not complete successfully, see VAX/VMS UETP User's Guide, Chapter 3

INSTALLING OPTIONAL SOFTWARE ON VMS AND UPDATING VMS

- Software installation done after installing VMS and running UETP
  - Uses either VMSINSTAL or VMSUPDATE command procedure in SYS$UPDATE
  - Consult the optional product's software installation guide for instructions
- System update performed periodically to correct problems
  - Uses either VMSINSTAL or VMSUPDATE command procedure in SYS$UPDATE
  - Consult the VMS Release Notes of the maintenance update for instructions

```
$ SET DEFAULT SYS$UPDATE
$ @VMSUPDATE

                    VMS Update Procedure

 This command procedure performs VAX/VMS software updates and
 optional software installations for VAX/VMS Release 3.
 During this sequence, the standard console medium will not
 be present in the console drive. Therefore, the system may
 be vulnerable to a power failure or other fatal crash. If a
 system crash should occur during this period the update
 sequence can be restarted at the beginning of the first
 incomplete update.

 Dismount the current console medium.

 Please place the first volume in the console drive

Are you ready to continue?: Y
%MOUNT-I-MOUNTED, DECNETLIC     mounted on _CSA1:

    DECnet-VAX software license installation

%PATCH-I-NOLCL, image does not contain local symbols
%PATCH-I-NOGBL, some or all global symbols not accessible
%PATCH-I-WRTFIL, updating image file SYS$SYSROOT:[SYSEXE]NETACP.EXE;2

Are there more kits to process?: N

 Please place the system console medium in the console drive.

Are you ready to continue?: Y
%MOUNT-I-MOUNTED, CONSOLE       mounted on _CSA1:
 Requested update sequence is complete.
$
```

Example 7-3 Installing the DECnet License

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" Y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO, D.F.

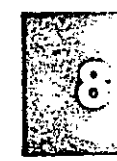
SYSTEM PROBLEMS

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985

SYSTEM PROBLEMS



INTRODUCTION

Many error reporting and analysis tools contribute to the reliability and maintainability of VAX/VMS. These tools are used extensively by DIGITAL personnel to prevent, diagnose, and solve system problems. Although a complete explanation of these tools is beyond the scope of this course, this module introduces you to tools you can use to:

- Display data on system failures
- Record and display data on device errors
- Format a disk volume

Additional tools for monitoring a VAX/VMS system are covered in Module 9, Performance.

A system manager is not usually expected to identify and solve every VAX/VMS system problem that may arise. Many problems require outside assistance, such as is provided by DIGITAL Field Service and Software Services.

DIGITAL Field Service and Software Services organizations provide services that can identify and solve system problems. In particular, many hardware maintenance agreements include DIGITAL's remote diagnosis service, which enables Field Service to examine VAX hardware from a DIGITAL Diagnosis Center without sending personnel on site. If remote diagnosis indicates that on-site repairs are necessary, personnel and parts are dispatched.

Similarly, if a customer has purchased telephone support from Software Product Services, or if a software warranty is still in force, software problems can be handled by telephone from a Telephone Support Center. Software "bugs" that are discovered should be reported to Software Engineering on a Software Performance Report (SPR).

OBJECTIVES

To collect information useful for troubleshooting, a system manager must:

- Run the system dump analyzer utility after a system failure
- Enable error logging and generate an error log report
- Initiate a remote diagnosis session
- Fill out a Software Performance Report (SPR)

RESOURCES

1. VAX/VMS Command Language User's Guide
2. VAX/VMS System Management and Operations Guide
3. VAX-11 Utilities Reference Manual
4. VAX/VMS System Dump Analyzer Reference Manual

SYSTEM DUMPS

System Dump File

- Name: SYS$SYSTEM:SYSDUMP.DMP
- Created: At system initialization
- Use: To store a copy of physical memory, written by VMS each time the system fails

Analyzing the System Dump File

- Tool: SDA utility
- Requirement: Read access to the system dump file
- Result: Reports Describing
  - Processes
  - Devices
  - Memory management data structures
  - I/O data structures
- Steps in running SDA utility

Table 8-1 Analyzing a System Dump

Steps in Running SDA Utility	Comments
1. $ ANALYZE/CRASH_DUMP - $_SYS$SYSTEM:SYSDUMP	Requires: - Read access to dump file - Read access to system symbol table - Value of system parameter VIRTUALPGCNT at least 1000 pages larger then size of dump file - If executed in site-specific start-up file, value of system parameter PQL_DPGFLQUOTA at least 1000 pages larger than size of dump file - If executed elsewhere, value of PGFLQUOTA field in UAF record at least 1000 pages larger than size of dump file
2. SDA > command	Table 8-2 shows typical SDA commands that produce reports about the system dump.
3. SDA > EXIT	

- Types of SDA reports

Table 8-2 Reports Generated by the System Dump Analyzer

Subject of Report	Command Format
Reason system failed, processor registers	SHOW CRASH
Active processes	SHOW SUMMARY
Current process	SHOW PROCESS /ALL
Process in balance set	SHOW PROCESS/ALL[/INDEX=nn][process-name]
System device	SHOW DEVICE device-name

```
Time of system crash:  4-MAY-1982 15:59:31.86

Version of system: VAX/VMS VERSION V3.0

Reason for BUGCHECK exception: MACHINECHK, Machine check while in kernel mode

Process currently executing: HLYNKA

Current image file name: SYS$SYSROOT:[SYSEXE]EDT.EXE;

Current IPL: 31  (decimal)


General registers:

        R0  = 00000030   R1  = 0000000C   R2  = 000039D0   R3  = 00C00001
        R4  = 800EC860   R5  = 00003CA8   R6  = 7FFD51C0   R7  = 00004524
        R8  = 0000452C   R9  = FFD00000   R10 = 00003C58   R11 = 00000003
        AP  = 7FFB5C50   FP  = 7FFEADE4   SP  = 80127DCC   PC  = 800A8E21
        PSL = 041F0008


Processor registers:

        P0BR   = 802A1E00     PCBB   = 00162C74     ACCS   = 00000000
        P0LR   = 000001F0     SCBB   = 003F0C00     SBIFS  = 00040000
        P1BR   = 7FAB5800     ASTLVL = 00000004     SBISC  = 00000000
        P1LR   = 001FFD9B     SISR   = 00180000     SBIMT  = 00200000
        SBR    = 003F2600     ICCS   = 800000C1     SBIER  = 00008002
        SLR    = 00003680     ICR    = FFFFFB4B     SBITA  = 00000000
                              TODR   = 4FAFAAFB     SBIS   = 20CB0001

        ISP    = 80127DCC
        KSP    = 7FFEADE4
        ESP    = 7FFEBE00
        SSP    = 7FFED56C
        USP    = 7FFB5C14
```

Example 8-1 Report Produced by SDA Command SHOW CRASH

PID	Process name	Image name	State	Pri	UIC	WKSET
00010000	NULL		COM	0	[000,000]	0
00010001	SWAPPER		HIB	16	[000,000]	0
000F003B	MELVIN	DISK$USER:[DRAW]DRAW.EXE;2	LEF	7	[011,240]	200
000C003C	FRIEDMAN		LEF	6	[011,040]	81
0017003E	HUNT	-- IMAGE NAME NOT AVAILABLE --	PFW	7	[011,140]	95
0008003F	HLYNKA	SYS$SYSROOT:[SYSEXE]EDT.EXE;	CUR	7	[011,120]	167
000E0040	TOWLE		LEF	4	[011,340]	87
00070041	SIGMON	SYS$SYSROOT:[SYSEXE]MAIL.EXE;	LEF	9	[011,330]	140
00140045	OU	SYS$SYSROOT:[SYSEXE]EDT.EXE;	COM	7	[011,310]	95
00120046	_TTG1:		LEF	5	[011,320]	76
00010047	REMACP	-- IMAGE NAME NOT AVAILABLE --	HIB	9	[001,003]	27
00010048	EVL	-- IMAGE NAME NOT AVAILABLE --	HIB	4	[001,004]	20
00010049	NETACP	-- IMAGE NAME NOT AVAILABLE --	HIB	10	[001,004]	145
0001004A	PRTSYMB1	-- IMAGE NAME NOT AVAILABLE --	HIB	8	[001,004]	16
0001004B	OPCOM	SYS$SYSROOT:[SYSEXE]OPCOM.EXE;	LEF	8	[001,004]	73
0001004C	JOB_CONTROL	-- IMAGE NAME NOT AVAILABLE --	HIB	9	[001,004]	117
0001004E	DRAOBACP	-- IMAGE NAME NOT AVAILABLE --	HIB	10	[001,003]	173
0002004F	ERRFMT	SYS$SYSROOT:[SYSEXE]ERRFMT.EXE	HIB	7	[001,006]	44

Example 8-2 Report Produced by SDA Command SHOW SUMMARY

Backing Up the System Dump File

- Reason: Overwritten each time the system fails
- When to back it up: Between failures, to preserve information
- Automatic backup: Place SDA commands in site-specific start-up command file

```
$ !
$ ! If system failed, back up system dump file and print reports
$ !
$ ANALYZE/CRASH_DUMP SYS$SYSTEM:SYSDUMP.DMP
COPY DUMP_DISK:SAVEDUMP.DMP     !Save dump file on DUMP_DISK volume
SET OUTPUT LPA0:SYSDUMP.LIS     !Print dump reports on line printer LPA0:
SHOW CRASH                      !Display crash information
SHOW STACK                      !Show current stack
SHOW SUMMARY                    !List all active processes
SHOW PROCESS/PCB/PHD/REG        !Display current process
SHOW SYMBOL/ALL                 !Print system symbol table
EXIT
```

Example 8-3 Backing Up the Dump File at Start-Up

DEVICE ERRORS

Error Logging

- The system error log file
  - Name: SYS$ERRORLOG:ERRLOG.SYS
  - Records: Hardware problems and other activity on disk and tape drives
  - Entries: Made by ERRFMT process
  - Note: Not all entries represent device or volume failure
- Controlling input to the file
  - Enable/Disable: On individual disk and tape devices
  - Default: Enabled on all disk and tape devices

Table 8-3 Enabling and Disabling Error Logging

	Command Format and Examples (Requires OPER Privilege)
Enabling error logging on a device	$ SET DEVICE/ERROR_LOGGING device-name $ SET DEVICE/ERROR_LOGGING DBB2:
Disabling error logging on a device	$ SET DEVICE/NOERROR_LOGGING device-name $ SET DEVICE/NOERROR_LOGGING DBB2:

- Analyzing the system error log file
  - Tool: SYE utility
  - Requirement: Read access to the file
  - Result: Report generated
  - Reports are used:

    By system managers: To find out when a device or volume may need attention

    By DIGITAL Field Service: To isolate and solve device problems

Table 8-4 Analyzing the Error Log

Steps in Running the SYE Utility	Comments
1. $ SET DEFAULT SYS$ERRORLOG	
2. $ RENAME ERRLOG.SYS ERRLOG.OLD	Requires W access to SYS$ERRORLOG After you rename the current error log file, the ERRFMT process opens a new one, and error logging continues. If you rename the error log file according to a schedule (for example, daily), you can skip steps 1 and 2.
3. $ RUN SYS$SYSTEM:SYE SYE VERSION 3.5 _INPUT FILE [SYS$ERRORLOG:ERRLOG.OLD] ? _OUTPUT FILE [SYS$OUTPUT] ? _OPTIONS [ROLL-UP] ? _DEVICE NAME [<CR>] ? _AFTER DATE [FIRST ENTRY] ? _BEFORE DATE [LAST ENTRY] ?	Requires R access to error log file. Your responses to the SYE prompts determine the type of reports generated. For typical report types, see Table 8-5. After SYE generates the report, it exits.

Table 8-5 Typical Error Reports

	Type of Report			
SYE Prompt	Error Summary (Totals Only)	Brief Description of Each Error	Mounts and Dismounts	Error Summary Specific Device
_INPUT FILE	<RET>	<RET>	<RET>	<RET>
_OUTPUT FILE	SUMMARY	BRIEF	CONFIG	DEVICE
_OPTIONS	<RET>	BRIEF	BRIEF	<RET>
_DEVICE NAME	<RET>	<RET>	*/CONFIG	device-name
_AFTER DATE	<RET>	<RET>	<RET>	<RET>
_BEFORE DATE	<RET>	<RET>	<RET>	<RET>

```
$ run sys$system:sve

SYE VERSION 3.5

_INPUT FILE   [SYS$ERRORLOG:ERRLOG.OLD]   ?
_OUTPUT FILE     [SYS$OUTPUT]             ?
_OPTIONS         [ROLL-UP]                ?
_DEVICE NAME     [<CR>]                   ?
_AFTER DATE      [FIRST ENTRY]            ?
_BEFORE DATE     [LAST ENTRY]             ?


DEVICE ROLLUP LOGGED BY SID 01380137

DEVICE          ERROR BITS SET          QIO TIMEOUT      ERRORS THIS SESSION   QIOS THIS SESSION
                [HARD] [SOFT]     [HARD] [SOFT]

_DBA2:                  1.                                      1.                 3251.

_DBA6:                130.                                    389.               872876.

_MTA0:                  1.                                      1.                  532.

_MTA1:                  2.                                     17.                  544.

VOLUME LABEL(S) LOGGED BY SID 01380137

                                                  QIO(S)      ERROR(S)    MOUNT(S)

      LABEL --

      _DBA6:                                     158077.          130.           5.

      LABEL -- 12APR.

      _MTA0:                                        227.            0.           1.

      LABEL -- BCK1

      _MTA0:                                        627.            1.           1.
      _MTA1:                                        859.            1.           1.

      LABEL -- BILL

      _MTA1:                                        205.            0.           1.

      LABEL -- IHFS

      _DMA0:                                         59.            0.           1.

      LABEL -- JALBAK

      _DMA2:                                         40.            0.           5.
```

Example 8-4 Error Log Summary Report (Sheet 1 of 2)

```
SUMMARY OF ALL ENTRIES LOGGED BY SID 01380137

      DEVICE ERROR BIT SET                134.
      ERRLOG.SYS CREATED                    1.
      TIME-STAMP                           30.
      SYSTEM BUGCHECK                       1.
      VOLUME MOUNT                         15.
      VOLUME DISMOUNT                      15.

      DATE OF EARLIEST ENTRY          19-APR-1982 09:55:35.27
      DATE OF LATEST ENTRY            20-APR-1982 08:05:35.28

PROCESSED ENTRIES HOUR-OF-DAY HISTOGRAM LOGGED BY SID 01380137

      00:00      7. *******
      01:00     16. ****************
      02:00     20. ********************
      03:00     17. *****************
      04:00     32. ********************************
      05:00     42. ******************************************
      06:00     13. *************
      07:00      7. *******
      08:00      1. *
      09:00      1. *
      10:00      2. **
      11:00      0.
      12:00      0.
      13:00      0.
      14:00      6. ******
      15:00      8. ********
      16:00      5. *****
      17:00      0.
      18:00      0.
      19:00      7. *******
      20:00      0.
      21:00      0.
      22:00     10. **********
      23:00      2. **

      RECORDS PROCESSED             196.
      RECORDS READ                  196.

PROGRAM RUNTIME STATISTICS

      TIMES IN SECONDS        PAGE    DIRECT     BUFFERED
          CPU  ELAPSED       FAULTS       I/O          I/O

          5.6     40.8          182         4          103


%SYE-I-SUCCOM, SUCCESSFUL COMPLETION
$
```

Example 8-4 Error Log Summary Report (Sheet 2 of 2)

Hardware Diagnostics

- Used by: DIGITAL Field Service
- Can run:
  - In the device hardware itself
  - In the VAX CPU, stand-alone
  - In the VAX CPU, on-line, under the control of VMS
- Hardware diagnostics are beyond the scope of this course
  - Introduction to on-line diagnostics in the following section to enable you to format a disk volume

On-Line Diagnostics

- Not considered part of VMS
- Not supplied when VMS is installed
- Must be copied into SYS$MAINTENANCE by DIGITAL Field Service
- Run while logged in as FIELD
- Run under the control of the Diagnostic Supervisor program appropriate to the VAX processor type

Table 8-6 Diagnostic Supervisor Programs

VAX Processor	Diagnostic Supervisor Program
VAX-11/780	ESSAA
VAX-11/750	ECSAA
VAX-11/730	ENSAA

- Can be used to format a disk volume
  - DIGITAL disk volumes are factory formatted
  - Formatting a volume includes:

    Locating bad blocks and recording locations

    Recording timing information required for VAX/VMS to use volume
  - Bad blocks can be located on formatted volumes by BAD utility

```
$
$
$ ALLOCATE DMA0:
  _DMA0: allocated
$ SET DEFAULT SYS$MAINTENANCE
$ RUN ESSAA
DIAGNOSTIC SUPERVISOR,  ZZ-ESSAA-6.4-513  22-APR-1982 14:22:52
DS> ATTACH DW780 SBI DW0 3 4
DS> ATTACH RK611 DW0 DMA 777440 210 5
DS> ATTACH RK07 DMA DMA0
DS> SELECT DMA0
DS> RUN EVRAC/SEC:PACKINIT
.. PROGRAM: EVRAC VAX DISK FORMATTER, REV 5.4, 11 TESTS, AT 14:23:59.78.
TESTING: _DMA0

DISK TYPE= RK07  FOR PHYSICAL DEVICE _DMA0
PACK STRUCTURE= ODS2    PACK LABEL= 'SCRATCH

  ***********  WARNING  **********
  BAD SECTOR FILE WILL BE REFORMATTED

DO YOU WISH TO CONTINUE? [(NO), YES] YES

FOR PHYSICAL DEVICE _DMA0
ENTER PACK SERIAL NUMBER [(1), 1-2147483647(D)]

CONTENTS OF BAD SECTOR FILE
PACK SERIAL NUMBER=          1(D) FOR DEVICE _DMA0
NO BAD SECTORS IN FILE
FORMAT STARTED     AT 14:24:25.21 FOR DEVICE _DMA0
FORMAT COMPLETED   AT 14:26:42.44 FOR DEVICE _DMA0
VERIFY STARTED     AT 14:26:42.88 FOR DEVICE _DMA0
BAD SECTOR FILE UPDATED BY     1(D) ENTRIES

CONTENTS OF BAD SECTOR FILE
PACK SERIAL NUMBER=          1(D) FOR DEVICE _DMA0
LOGICAL BLOCK      CYLINDER    TRACK    SECTOR
-----------------------------------------------
     47058(D)        713(D)      0(D)      0(D)
VERIFY COMPLETED   AT 14:40:10.95 FOR DEVICE _DMA0
FOR PHYSICAL DEVICE _DMA0 DISK TYPE=RK07
PACK WAS LABELED 'SCRATCH      '
PACK SERIAL NUMBER=          1(D)
NUMBER OF BAD SECTORS ADDED TO BAD SECTOR FILE=     1(D)
.. End of run.  0 errors detected.  Pass count: 1.  Time: 22-APR-1982 14:40:12.84
DS> ^Z
$ DEALLOCATE DMA0:
$
```

Example 8-5 Formatting an RK07 Volume

```
$ ALLOCATE DRA0:
  _DRA0: allocated
$ SET DEFAULT SYS$MAINTENANCE
$ RUN ESSAA
DIAGNOSTIC SUPERVISOR,  ZZ-ESSAA-6.4-513  23-APR-1982 12:35:15
DS> ATTACH RH780 SBI RH0 8 5
DS> ATTACH RM03 RH0 DRA0
DS> SELECT DRA0
DS> RUN EVRAC/SEC:PACKINIT
        .
        .
        .
```

Example 8-6 Formatting an RM03 Volume

```
$ ALLOCATE DBA0:
  _DBA0: allocated
$ SET DEFAULT SYS$MAINTENANCE
$ RUN ESSAA
DIAGNOSTIC SUPERVISOR,  ZZ-ESSAA-6.4-513  23-APR-1982 11:50:00
DS> ATTACH RH780 SBI RH0 8 5
DS> ATTACH RP06 RH0 DBA0
DS> SELECT DBA0
DS> RUN EVRAC/SEC:PACKINIT
        .
        .
        .
```

Example 8-7 Formatting an RP06 Volume

```
$ ALLOCATE DMA0:
  _DMA0: allocated
$ SET DEFAULT SYS$MAINTENANCE
$ RUN ESSAA
DIAGNOSTIC SUPERVISOR,  ZZ-ESSAA-6.4-513  22-APR-1982 14:22:52
DS> @CONFIG
DS> !CONFIGURATION FILE FOR SYSTEM TYPE SV-AXCWA RP06/TU45 PACKAGED SYSTEM
DS> ! CONFIG.COM;2          VERSION:1.3          22-JUN-79
DS> !
DS> ! DEFINE PROCESSOR...
DS> ATTACH KA780 SBI KA0 NO NO 0 0
DS> !
DS> ! DEFINE MEMORY...
DS> ATTACH MS780 SBI MS0 1
DS> !
DS> ! DEFINE MASSBUS ADAPTERS...
DS> ATTACH RH780 SBI RH0 8 5
DS> ATTACH RH780 SBI RH1 9 5
DS> !
DS> ! DEFINE UNIBUS ADAPTERS...
DS> ATTACH DW780 SBI DW0 3 4
DS> !
DS> !DEFINE UNIBUS DISKS
DS> ATTACH RK611 DW0 DMA 777440 210 5
DS> ATTACH RK07 DMA DMA0
DS> ATTACH RK07 DMA DMA1
DS> !
DS> ! DEFINE MASSBUS DISKS...
DS> ATTACH RP06 RH0 DBA0
DS> !
DS> ! DEFINE MAGTAPES...
DS> ATTACH TU45 RH1 MTA0
DS> !
DS> ! DEFINE TERMINALS...
DS> ATTACH DZ11 DW0 TTB 760130 330 5 EIA
DS> !
DS> ATTACH DMC11 DW0 XMA 760070 300 5
DS> ATTACH DMC11 DW0 XMB 760100 310 5
DS> ATTACH DMC11 DW0 XMC 760110 320 5
DS> !
DS> ATTACH VT100 TTB TTB0
DS> ATTACH VT100 TTB TTB1
DS> ATTACH VT100 TTB TTB2
DS> ATTACH VT100 TTB TTB3
DS> ATTACH VT100 TTB TTB4
DS> ATTACH VT100 TTB TTB5
DS> ATTACH VT100 TTB TTB6
DS> ATTACH VT100 TTB TTB7
DS> SELECT DMA0
DS> RUN EVRAC/SEC:PACKINIT
        .
        .
        .
```

Example 8-8 Formatting an RK07 Volume, Using a Configuration Command File

Remote Diagnosis

- Performed at the DIGITAL Diagnosis Center
  - If VMS is running, tools are on-line diagnostics, SDA, and SYE
  - If VMS is not running, tools are stand-alone diagnostics and console commands
- Remote diagnosis access controlled by keyswitch on console

Table 8-7 Enabling Remote Diagnosis

Function that Can Be Performed from Remote Location	Keyswitch Setting		
	VAX-11/780	VAX-11/750	VAX-11/730
Given authorized user name and password, can log in	REMOTE DISABLE	REMOTE SECURE	REM DSBL
Given authorized user name and password, can log in; also, can type <CTRL/P> and exercise console functions	REMOTE	REMOTE	REMOTE

UETP

- Run after system installation to demonstrate that VAX/VMS is working correctly (Module 7)
- Can use as additional tool to uncover hardware problems
  - Device-testing phase
  - Does not identify cause of problem

SOFTWARE PERFORMANCE REPORTS

Reporting Software Problems

- When software problem appears to be the result of a bug, submit SPR to DIGITAL Software Engineering
- Problem should be reproducible and well documented
- Observe the following rules:
  - Include listings that reveal the state of entire system
  - Explain how to reproduce the problem as simply as possible
  - Provide information in machine-readable form
  - Provide information about system environment not apparent from listings
  - Include any clues and analyses of your own

CURSOS: "INTRODUCCION AL SISTEMA VAX - 11/780" Y
"OPERACION Y ADMINISTRACION DEL SISTEMA VAX /VMS"
DEL 29 DE ABRIL AL 13 DE MAYO
MEXICO, D.F.

PERFORMANCE

PROFESORES:

ING. EDUARDO S. JALLATH CORIA
ING. ALEJANDRO JIMENEZ GARCIA
ING. SOCRATES A. MUÑIZ ZAFRA
ING. HUMBERTO SANCHEZ SANDOVAL

MAYO DE 1985

PERFORMANCE

INTRODUCTION

VMS provides many tools for monitoring system behavior. Using these tools, you can examine the activity of VMS processes and the utilization of system resources. If user demands on system resources become too great, system performance will become unsatisfactory.

The primary factor limiting VAX/VMS performance is hardware resources -- especially physical memory and disk storage. Hardware resources that are inadequate for the workload generally provide inadequate performance, regardless of any other efforts to improve performance.

When VMS is installed and started up for the first time, the system determines the amount of physical memory and disk storage available. Then, it uses this information to compute sizes for its data structures and work files that make effective use of available resources. If additional physical resources are added to the system to improve its performance, you can have VMS recompute these sizes to take advantage of the new resources.

The topic of system performance is covered in detail in another course, VAX/VMS System Performance Management.

OBJECTIVES

A system manager must:

- Monitor the system to isolate a performance problem
- Modify system parameters to reflect a change is system resources

RESOURCES

1. VAX/VMS Command Language User's Guide
2. VAX/VMS System Management and Operations Guide
3. VAX-11 Utilities Reference Manual

OVERVIEW

- System performance is determined by
  - Hardware resources
    - Physical memory
    - Disk storage
  - Application design
  - System performance management
    - System parameters tuning
- Duties of the system manager
  - Isolate performance problems by monitoring
  - Improve performance problems by modifying:
    - System resources
    - System parameters

MONITORING SYSTEM ACTIVITY

Overview

Table 9-1 System, Process, and Device Monitoring

Information Displayed	Command or Utility
General System Information	
Overview of the processes on the system	$ SHOW SYSTEM
Overview of print queues	$ SHOW QUEUE/DEVICES/ALL
Overview of batch queues	$ SHOW QUEUE/BATCH/ALL
Overview of mounted disk and tape volumes	$ SHOW DEVICES/MOUNTED
Overview of system memory resources	$ SHOW MEMORY
Demands on system resources	$ MONITOR
Specific Information (Process or Device)	
Interactive users, terminal codes, and process IDs	$ SHOW USERS
Information on what a process is currently doing	$ SHOW PROCESS/CONTINUOUS
Information on what a process is allowed to do	$ RUN SYS$SYSTEM:AUTHORIZE
Information on how much disk space a user is allowed	$ RUN SYS$SYSTEM:DISKQUOTA $ SHOW QUOTA/USER=[uic]
Information on devices and volumes	$ SHOW DEVICE device

Monitoring Active Processes

- SHOW SYSTEM
- Process scheduling states
  - Current process
  - Process in wait states
    - Computable processes
    - Other processes

```
VAX/VMS  Processes on          9-MAR-1981 17:40:50.05      Uptime  4 03:34:43

  Pid     Process Name     UIC  State Pri  Dir. I/O     CPU       Page flts Ph.Mem

00010000 NULL          (1) 000,000 COM    0        0 11:08:14.89        0     0
00010001 SWAPPER           000,000 HIB   16        0 00:02:45.65        0     0
000F001E FRIEDMAN          363,002 LEF    9      297 00:01:12.90     7384    44
002A0023 MARSH         (2) 363,010 CUR    7     2180 00:03:27.68    29011   104
001B0029 TOWLE             363,020 LEF    4      144 00:00:18.17     2256    47
00340036 MASORS        (3) 363,005 LEF    9       29 00:00:02.37      618   128
00020038 REMACP            001,003 HIB   10        1 00:07:12.45       36    30
00010039 NETACP            001,004 HIB   10    11874 00:11:33.65   193528   120
003F003C NATASHA       (4) 366,010 CEF    4        1 00:00:00.07       23    52
0001003D PRTSYMB2          001,004 HIB    8     1891 00:03:57.92       27    46 S
0001003F ERRFMT            001,006 HIB    8     1340 00:00:16.15       30    32
00010040 OPCOM             001,004 LEF    7      458 00:00:05.32       34    42
00010041 JOB_CONTROL   (5) 001,004 HIB    9     4543 00:01:08.37      347   130
00010042 DBA0ACP           001,003 HIB    8   179208 01:25:50.87   130922   225
00020043 PRTSYMB1          001,004 HIB    8     3453 00:07:36.01       45    44 S
```

Example 9-1 Output from the SHOW SYSTEM Command

NOTE

The numbers in Table 9-2 correspond to the numbered lines in Example 9-1.

Table 9-2 System States

	Name of State and Abbreviation	OK	Probably OK	Possible Problem	Definition of State
1	Computable COM/COMO		X		Available to use the processor
4	Common Event Flag CEF/CEFO		X		Waiting for something to happen
*	Collided Page COLP			X	Waiting for the system
2	Current CUR	X			Executing on the processor
*	Free Page Wait FPG			X	Waiting for physical memory
3	Local Event Flag LEF/LEFO		X		Waiting for something to happen
5	Hibernate HIB/HIBO		X		Intentionally doing nothing**
*	Mutex/Misc Resource MWAIT			X	Waiting for a resource
*	Suspended SUSP/SUSPO		X		Intentionally doing nothing***
*	Page Fault Wait PFW			X	Waiting for the system

* No example of this state is in Example 9-1.

** This state is normally used by system processes when they have no task to perform. This means that they are available when needed, but are not competing with computable processes for the processor.

*** This state should be used for short periods only.

Monitoring System Processes

Table 9-3 Definition of System Processes

System Process	Normal State(s)	Purpose
NULL	COM	Keeps the CPU busy when no other processes are computable.
SWAPPER	HIB,COM	Transfers processes between physical memory and disk.
ERRFMT	HIB,COM	Records device errors in a file on the system disk.
xxxACP	HIB,COM	Manages transmission of data to and from disks, tapes, and interconnected computers.
PRTSYMBn	HIB,COM	Prints files on line printers.
OPCOM	HIB,COM	Transmits and records user requests and operator replies.
JOB_CONTROL	HIB,COM	Manages print jobs, batch jobs, interactive processes, and process accounting.

Table 9-4 Results of System Process Deletion

System Process	Result of Process Deletion
SWAPPER	Cannot be deleted
ERRFMT	No device errors logged*
xxxACP	System slows down, possibly stops completely
OPCOM	No user/operator communication*
JOB_CONTROL	Users cannot log in and queues do not function*
PRTSYMBn	Printers may not print
NULL	Cannot be deleted

* To restart the ERRFMT, OPCOM, or JOB_CONTROL process, log in as SYSTEM and run the command procedure STARTUP.COM, as shown below. Instead of process-image, type either ERRFMT, OPCOM, or JOBCTL.

```
$ @SYS$SYSTEM:STARTUP process-image
```

Obtaining Information About a Device

- SHOW DEVICES

Table 9-5 SHOW DEVICES Command

Information Displayed	Command
List of all devices on the system	$ SHOW DEVICES
Amount of unused space on a mounted disk volume	$ SHOW DEVICES/MOUNTED
Characteristics or error count on a specific device	$ SHOW DEVICES devices-name
Owner of an allocated device	$ SHOW DEVICES/FULL device-name
Names of all open files on a volume	$ SHOW DEVICES/FILES device-name

Monitoring Memory Resources

- System performance strongly dependent on amount of physical memory
- SHOW MEMORY

```
                System Memory Resources on 23-MAR-1982 15:53:10.83

Physical Memory Usage (pages):      Total       Free    In Use  Modified
Main Memory (4.00Mb)                 8192       4524      3631        37

Slot Usage (slots):                 Total       Free  Resident   Swapped
  Process Entry Slots                  75         53        22         0
  Balance Set Slots                    70         50        20         0

Fixed-size Pool Areas (packets):    Total       Free    In Use      Size
  Small Packet (SRP) List              272         66       206        96
  I/O Request Packet (IRP) List        300       2184       116       160
  Large Packet (LRP) List               13          7         6       640

Dynamic Memory Usage (bytes):       Total       Free    In Use   Largest
  Nonpaged Dynamic Memory           179712      97504     82208     96960
  Paged Dynamic Memory               65536      45312     20224     45232

Paging File Usage (pages):                      Free    In Use     Total
  DBA0:[SYS0.SYSEXE]SWAPFILE.SYS               11352      3648     15000
  DBA0:[SYS0.SYSEXE]PAGEFILE.SYS               28870      1130     30000

Of the physical pages in use, 1303 pages are permanently allocated to VMS.
```

Example 9-2 Sample SHOW MEMORY Output

Monitoring Print and Batch Queues

- SHOW QUEUE/DEVICES/ALL
- SHOW QUEUE/BATCH/ALL

MONITOR Utility

- MONITOR class-name/class-name-qualifier [,...]

Table 9-6 MONITOR Class Names

Class Description	Class Name
DECnet-VAX statistics	DECNET
File system ACP statistics	FCP
System I/O statistics	IO
Lock management statistics	LOCK
Time spent in each of the processor modes	MODES
Page management statistics	PAGE
Statistics on space allocation in the nonpaged dynamic pool	POOL
Statistics on all processes	PROCESSES
Number of processes in each of the scheduler states	STATES

Table 9-7 MONITOR PROCESSES Class Qualifiers

Qualifier Description	Qualifier
Show the top buffer I/O users	/TOPBIO
Show the top CPU users	/TOPCPU
Show the top direct I/O users	/TOPDIO
Show the top page fault users	/TOPFAULT

Unlike most utilities that display system information, MONITOR can:

- Display several classes of information alternately
- Summarize statistics over a long period of time
- Record information in a disk file
- Play back information that it has recorded

VAX/VMS Monitor Utility
NONPAGED POOL STATISTICS
19-MAR-1982
15:55:46

	CUR	AVG	MIN	MAX
SRPs Left	48.00	56.72	8.00	99.00
IRPs Left	151.00	163.61	140.00	181.00
LRPs Left	31.00	30.89	27.00	31.00
Total Space Left	27216.00	28031.91	25632.00	29536.00
Holes In Pool	16.00	15.51	13.00	19.00
Largest Block	25536.00	26533.13	24512.00	27344.00
Smallest Block	16.00	16.00	16.00	16.00
Blocks LE 32 Bytes	5.00	4.47	3.00	7.00

Example 9-3 MONITOR Screen Display of the POOL Class

NOTE
Each value updated every few seconds.

```
                    VAX/VMS Monitor Utility
                     TOP CPU TIME PROCESSES
                          24-MAR-1982
                            10:11:45

                              0         25        50        75       100
                              + - - - - + - - - - + - - - - + - - - - +
[000,000] NULL          66    **************************
                              |         |         |         |         |
[011,210] LUCAS         16    ******
                              |         |         |         |         |
[011,130] HOUSSELL       4    *
                              |         |         |         |         |
[011,010] BUFORD         3    *
                              |         |         |         |         |
[000,000] SWAPPER        3    *
                              |         |         |         |         |
[001,003] DRAOACP        1
                              |         |         |         |         |
[021,111] KERMIT
                              |         |         |         |         |
                              + - - - - + - - - - + - - - - + - - - - +
```

Example 9-4 MONITOR PROCESSES/TOPCPU Screen Display

NOTE

Bar graph updated every few seconds.

Information on a Specific Process

- SHOW PROCESS/CONTINUOUS [/ID=proc-id] [proc-name]
- SHOW PROCESS/CONTINUOUS/ID=process_id
  - Requirement: WORLD Privilege
  - Exit: "E"

```
                    Process MARSH
State                   CUR           Working Set       125
Cur/base priority       6/4           Virtual Pages     881
UIC                     [360,010]     CPU Time          0:01:45.75
PID                     002A0023      Direct I/O        25
Stored PC               80000328      Buffered I/O      330
Stored PSL              03C00000      Page Faults       534
Stored SP               7FFC77D4      Event flags       CC000013
                                                        80000000
SYS$SYSROOT:[SYSEXE]SHOW.EXE;1
```

Example 9-5 Output from SHOW PROCESS/CONTINUOUS

Interactive Users

- SHOW USERS

```
VAX/VMS Interactive Users - Total = 5
     26-MAY-1981 16:13:26.29

    TTA1:     MARSH            002A0023
    TTA5:     FRIEDMAN         000F001E
    TTA7:     TOWLE            001B0029
    TTB3:     MASORS           00340036
    TTB4:     NATASHA          00BF003C
```

Example 9-6 Output from SHOW USERS

IMPROVING SYSTEM PERFORMANCE

Introduction

- Optimum system performance requires match of system resources with:
  - Size of VMS data structures
  - Number of installed image files
  - Size of system files
- System files (sizes customized at system installation)
  - Paging file
  - Swapping file
  - Dump file

Table 9-8 System Files

File	File Specification	Function
Paging file	SYS$SYSTEM:PAGEFILE.SYS	Used by VMS to create and manage virtual memory
Swapping file	SYS$SYSTEM:SWAPFILE.SYS*	Used by VMS to manage physical memory use
Dump file	SYS$SYSTEM:SYSDUMP.DMP	Used by VMS to save a copy of physical memory when the system fails

*If the storage capacity of the system disk is less than 25,000 blocks, the file specification is SYS$SYSTEM:SWAPFILE1.SYS.

- Change in physical resources should be accompanied by change in:
  - System parameters
  - File sizes
- Tools to use
  - SYS$UPDATE:AUTOGEN.COM - Command procedure (recommended method)

    Determines hardware resources

    Performs computations

    Changes system parameters

    Creates list of image files to install

    Creates new system files
  - SYS$SYSTEM:SYSGEN.EXE - Utility

    Changes system parameters

    Creates new system files
  - SYSBOOT - Conversational start-up utility

    Changes system parameters

Reconfiguring the System with AUTOGEN

- Running AUTOGEN
  - When system physical resources change

Table 9-9 Running AUTOGEN When System Physical Resources Change

Action		AUTOGEN Function
Step 1:	Log in, using the user name SYSTEM.	
Step 2:	$ SET DEFAULT SYS$UPDATE	
Step 3:	$ @AUTOGEN CONFIG	Determines hardware resources, records results in SYS$SYSTEM:CONFIG.DAT.
Step 4:	$ @AUTOGEN GENPAR	Uses CONFIG.DAT to generate certain system parameter values, records results in SYS$SYSTEM:PARAMS.DAT. This ASCII file can be edited to set the values of system parameters and sizes of system files manually.
Step 5:	$ @AUTOGEN	Uses PARAMS.DAT to compute values for system parameters, records the results in SYS$SYSTEM:AUTOGEN.PAR and in VMS executive image SYS$SYSTEM:SYS.EXE. Also, uses PARAMS.DAT to compute sizes of system files, creates them.
	or	
	$ @AUTOGEN SHUTDOWN	Also shuts down system for manual restart.
	or	
	$ @AUTOGEN REBOOT	Also shuts down system for automatic restart.

- To specify a system parameter value or file size manually

Table 9-10 Running AUTOGEN to Specify a System Parameter Value or File Size Manually

Action		Comments
Step 1:	Log in, using the user name SYSTEM.	
Step 2:	Edit the file SYS$SYSTEM:PARAMS.DAT to specify the system parameter value or system file size.	Parameter values appear as the full name of the parameter followed by an equal sign and the value of the parameter. File sizes are specified in blocks, using the keywords PAGEFILE, SWAPFILE, and DUMPFILE.
Step 3:	$ @SYS$UPDATE:AUTOGEN	AUTOGEN uses PARAMS.DAT to compute values for system parameters, records the results in SYS$SYSTEM:AUTOGEN.PAR and SYS$SYSTEM:SYS.EXE. Also uses PARAMS.DAT to compute sizes of system files, creates them.
	or	
	$ @SYS$UPDATE:AUTOGEN SHUTDOWN	Also shuts down system for manual restart.
	or	
	$ @SYS$UPDATE:AUTOGEN REBOOT	Also shuts down system for automatic restart.

Reconfiguring the System with SYSGEN

- Use SYSGEN to:
  - Specify alternate system start-up file or UAF
  - Back up system parameters in parameter file
  - Perform functions not available from AUTOGEN
- System parameters
  - Current
  - Active
  - Dynamic
- Principal SYSGEN commands and functions in Table 9-11

Performance Tuning

- Memory
  - Add more memory
  - Use shared code
  - Invoke automatic working set adjustment for interactive processes
  - Use batch streams for large/slow jobs
- CPU
  - Process CPU utilization is a function of:
    - User code
    - Run-time library
    - System procedures
    - System overhead (most responsive to tuning)
      - Page faults
      - ACP I/O
      - Swapping
      - "Unnecessary" I/O

- I/O Subsystems
  - Share code to reduce paging and swapping I/O
  - Use load balancing

    Place system and user files on separate disks/controllers
  - ACPs

    One FILES-11 ACP per system except for very slow devices

    Use ACP caching

    Bit map blocks

    File headers

    Directory blocks

    File IDs

    Extend blocks

    Quota file blocks
  - File access

    Use contiguous files to reduce ACP window turning (BACKUP/IMAGE)

    Use multi-block I/O to reduce RMS operations

    Use multi-buffer ISAM I/O to overlap I/O operations