digital

VAX/VMS

System Manager's Guide

Order No. AA-D027B-TE

VAX11

March 1980

This document covers the following tasks of VAX/VMS system management: 1) setting up user's accounts, 2) managing public files and volumes, 3) controlling overall system performance, 4) monitoring system activity, and 5) recognizing and dealing with errors and failures.

The aims of this guide are to provide a background for understanding these tasks and to provide rules and guidelines for performing them.

VAX/VMS
System Manager's Guide

Order No. AA-D027B-TE

SUPERSESSION/UPDATE INFORMATION:	This revised document supersedes the VAX/VMS System Manager's Guide (Order No. AA-D027A-TE)
OPERATING SYSTEM AND VERSION:	VAX/VMS V02
SOFTWARE VERSION:	VAX/VMS V02

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard, massachusetts

First Printing, August 1978
Revised, March 1980

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

CONTENTS

CONTENTS

CONTENTS

CONTENTS

Page

CONTENTS

FIGURES

CONTENTS

Page

TABLES

PREFACE

The VAX/VMS System Manager's Guide is both a useful adjunct to a formal course in VAX/VMS system management and an indispensable reference book for every VAX/VMS system manager.

MANUAL OBJECTIVES

The objectives of this guide are twofold. The first objective is to give the reader an understanding of the reasons for performing the tasks of VAX/VMS system management. The second objective is to show the reader how to perform the tasks of VAX/VMS system management.

INTENDED AUDIENCE

This guide is intended for any person who has overall responsibility for controlling the operations of a VAX/VMS installation. The reader of this guide is most likely a data processing generalist, not necessarily a programmer and probably not a systems programmer.

STRUCTURE OF THIS DOCUMENT

This system manager's guide consists of three parts, as follows:

Part I: User Accounts

Understanding the user authorization file (UAF); using the AUTHORIZE utility; understanding groups, limits, priorities, privileges, and the accounting log file

Part II: Operational Control

Initializing and mounting public volumes, backing up files on public volumes, establishing disk quotas, installing known images, maintaining the start-up command procedures, setting up spooling, creating and controlling batch queues and print queues, logging and reporting errors

Part III: System Configuration

Understanding system tuning considerations, understanding the system parameters, using the SYSGEN utility, monitoring the system

ASSOCIATED DOCUMENTS

The VAX-11 Information Directory and Index lists and describes all the documents the system manager may need to refer to in the course of performing system management tasks.

For general background information about the system, see the VAX/VMS Summary Description and Glossary and the VAX/VMS Primer.

The following documents may also be useful:

- VAX/VMS Command Language User's Guide
- VAX/VMS Operator's Guide
- VAX/VMS Release Notes
- VAX-11 Software Installation Guide
- VAX/VMS System Messages and Recovery Procedures Manual

SUMMARY OF TECHNICAL CHANGES

Modifications to this manual reflect the following changes to VAX/VMS for Version 2.0:

- UAF format -- The format of the UAF has changed from sequential to indexed. A conversion procedure (Section 2.5) updates VAX/VMS Version 1.0 UAFs to the new format. No longer is a TMP file used, and no longer is a new UAF version created after an update.

- Limits -- PGFLQUOTA is specified in pages rather than 256-page blocks. CPUTIME has been implemented. Former deductible limits are now pooled. Only CPUTIME is a deductible limit.

- Log-in flags -- LOCKPWD, a new log-in flag, prevents users from issuing the new SET PASSWORD command.

- Privileges -- Privileges are now categorized by level of danger: none, normal, group, devour, systems, file, all. Five new privileges exist: BYPASS (bypass UIC protection), SHMEM (multiport memory), EXQUOTA (exceed disk quotas), PFNMAP (map to physical pages), and SYSPRV (assume system UIC status). Privileges on the DEFAULT record include only TMPMBX and NETMBX; GROUP was deleted.

- Log-in without a UAF -- Any name and password work if no UAF exists; this permits an open system. Any name and password work on the console if the UAF cannot be accessed; this permits the system manager or operator to bypass a file problem.

- AUTHORIZE commands -- The LIST and SHOW commands in AUTHORIZE now take a userspec argument, plus /FULL and /BRIEF qualifiers. The display format has changed.

- User data on UAF -- Users can place up to 256 bytes of their own data on each UAF record using VAX-11 RMS but must specify the offset where this data begins and observe several other conventions.

- Volume sets -- The /BIND qualifier to the MOUNT command (see the VAX/VMS Command Language User's Guide) joins volumes into a volume set.

- File system caches -- The system will now cache extensive amounts of information concerning a volume's free space, file identifications, quota file entries, and file headers. The system manager, however, must take precautions to ensure that the cached information is written back to disk.

- RMSSHARE utility -- The RMSSHARE utility enables VAX-11 RMS file sharing and sets a maximum count on the number of pages that can be allocated from the paged dynamic pool to support this facility.

- DISKQUOTA utility -- The system manager can establish disk quotas with the new DISKQUOTA utility. The system automatically records disk usage and enforces quotas during file operations. The system now requires the SYSPRV privilege to change the owner UIC of a file. Relative volume 1 of a volume set must be online at all times.

- Shareable images -- Shareable images linked into executable images without private copies no longer need to be installed to be executed, as long as they do not contain writeable non-CRF sections. The system will create private sections for them.

- /PROTECT qualifier to INSTALL -- The INSTALL utility contains a new qualifier to designate images containing protected code -- code that runs in kernel or executive mode -- that may be called by a user-level image.

- INSTALL utility displays -- The INSTALL utility displays have been expanded to show protected images and to show global sections in multiport memory units.

- System symbol GBL$file-name -- This new symbol means all global sections beginning with file-name. The symbol is useful for installing shared images in multiport memory units.

- Site-independent start-up command procedure -- Changes include: minor changes to housekeeping chores and known images; deletion of commands to mount diskettes; addition of logical names for VAX-11 COBOL-74 and VAX-11 PASCAL; deletion of logical names for VAX-11 FORTRAN; and addition of commands to enable VAX-11 RMS file sharing. FORTRAN logical name assignments are now implicit.

- Model site-specific start-up command procedure -- Changes include commands to run the system dump analyzer, to connect multiport memory units, and to install secondary paging and swapping files.

- Batch queues -- The system manager can specify (in addition to a job limit, priority, and swapping mode) a working set default, working set quota, CPU time default, and CPU time maximum as qualifiers to the INITIALIZE and START commands. These options permit working set sizes and CPU time limits for batch jobs to be independent of those for interactive users. Users can also specify working set sizes and CPU time limits on the $JOB card or as qualifiers to the SUBMIT command.

- Printer characteristics and forms types -- The system manager can set printer characteristics and forms types as qualifiers to INITIALIZE and START commands. Print jobs submitted with characteristics not belonging to the print queue or with a different forms type are put on a hold status.

- System parameters -- New SYS parameters include PFRATL, PFRATH, WSINC, AWSMIN, AWSMAX, AWSTIME (all for automatic working set adjustments), EXTRACPU, XMAXRATE, LAMAPREGS, REALTIME_SPTS, CLISYMTBL, TTY_TYPAHDSZ, TTY_PROT, and TTY_OWNER. New ACP parameters include ACP_QUOCACHE, ACP_WRITEBACK, and ACP_DATACHECK; in addition, ACP_EXTCACHE, ACP_EXTLIMIT, and ACP_FIDCACHE are now used. The system now adjusts working set limits automatically according to the page fault rates for the working sets and the values of the automatic working set parameters. The new and newly implemented ACP parameters mostly specify cache sizes. Certain parameters are now designated as generative (affecting structures at system generation) and dynamic (able to be modified during system operation).

- SYSGEN utility -- The WRITE ACTIVE option modifies the dynamic parameters during system operation. The WRITE CURRENT option modifies the system image on disk (SYS$SYSTEM:SYS.EXE), which contains the initial parameter values for the next bootstrap operation. SYSGEN now initializes and connects multiport memory units. The CREATE command can now create non-contiguous secondary paging and swapping files on any volume or volume set. The new INSTALL command augments SYS$SYSTEM:PAGEFILE.SYS or SWAPFILE.SYS with a secondary paging or swapping file.

CHAPTER 1

INTRODUCTION

In a nutshell, system management of a VAX/VMS installation entails two main responsibilities: system performance and system operation. The system manager:

- Makes decisions that relate to optimizing the overall performance and efficiency of the system

- Performs procedures that relate to the overall management and control of the system

To make good decisions, the system manager must understand both the needs of users and the capabilities of the VAX/VMS operating system. To perform system management procedures well, the system manager must have a working knowledge of the functions of the VAX/VMS operating system.

The pattern followed throughout this guide is to discuss the issues and the facts that will help the system manager make decisions and then to give general rules and guidelines for performing the procedures of system management.

It is not possible to prescribe a precise set of formulas for setting up and running a VAX/VMS system. System management cannot be done by rote or from a cookbook. Rather, the system manager must -- by combining an understanding of users' needs and system capabilities with a working knowledge of the functions of VAX/VMS -- work out a coherent strategy for effective system management.

1.1 WHAT IS SYSTEM MANAGEMENT?

In its most abstract sense, system management means the overall control of the operations of a computer system for the benefit of the users of the system. System management is a function that can be exercised by a single system manager, assisted by system operators, or it is a function that can be shared by several persons, some or all of whom may also serve as system operators.

A computer installation exists to serve its users. Ideally, then, it should be operated to provide service to all users with efficiency and economy. This is the challenge of system management.

1.2 TASKS THAT THE SYSTEM MANAGER PERFORMS

For practical purposes, system management is best defined in terms of the tasks that the system manager typically performs. The tasks of system management fall into the following four categories:

- Getting the system up and running
- Setting up users' accounts
- Controlling the operation of the system
- Configuring the system for good performance

The first topic (getting the system up and running) is the subject of the VAX-11 Software Installation Guide. The remaining three topics are the subjects of this system manager's guide.

1.3 COMPONENTS OF THE VAX/VMS OPERATING SYSTEM

Among the first and most important responsibilities of the system manager is getting the VAX/VMS system up and running. At a minimum, this means the bootstrap loading of the operating system distributed by DIGITAL.

Usually, in addition to bootstrapping the VAX/VMS operating system, getting a system up and running means both customizing some of the parameters of the operating system and incorporating optional software into the system. This optional software, which runs under control of the VAX/VMS operating system, includes VAX-11 FORTRAN, VAX-11 BASIC, VAX-11 COBOL-74, and other optional products.

For an explanation of the steps that the system manager may have to take in getting a VAX/VMS system up and running, see the VAX-11 Software Installation Guide.

The components of the VAX/VMS operating system are cataloged in nine directories on the system distribution medium. The names of these directories and brief descriptions of their contents follow:

- [SYSLIB]
  This directory contains various macro and object libraries.
- [SYSMSG]
  This directory contains system message files.
- [SYSMGR]
  This directory contains files used in managing the operating system.
- [SYSHLP]
  This directory contains text files and help libraries for the HELP utility.
- [SYSERR]
  This is the directory for the error log file (ERRLOG.SYS).
- [SYSTEST]
  This directory contains files used in testing the functions of the operating system.

- [SYSMAINT]
  This directory contains system diagnostic programs.

- [SYSUPD]
  This directory contains files used in applying system updates.

- [SYSUPD.EXAMPLES]
  This directory contains sample driver programs, user-written system services, and other source programs.

- [SYSEXE]
  This directory contains the executable images of most of the functions of the operating system.

For a complete list of the files contained on the distribution medium, see the VAX-11 Software Installation Guide.

PART I

USER ACCOUNTS

The system manager is responsible for setting up and maintaining a meaningful system of accounts for:

- Defining the users to the system
- Defining relationships among the users of the system for file protection, interprocess communication, and accounting
- Granting some users the privileges to perform certain sensitive system functions, and restricting other users from performing these functions
- Setting limits on the use of reusable system resources
- Giving users priorities in using the system

Part I consists of Chapters 2 through 4:

- Chapter 2: User Authorization File -- Discusses the make-up of the user authorization file, provides operating instructions for the AUTHORIZE utility
- Chapter 3: Protection and Control -- Discusses the organization of users for protection, communication, and accounting functions
- Chapter 4: Resource Control -- Defines the limits and privileges available to VAX/VMS users, explains priorities and the accounting log file

CHAPTER 2

USER AUTHORIZATION FILE

The system manager names users who may log in to the system and places controls on their activities by maintaining a record for each user in the user authorization file (UAF). The full name of the UAF is SYS$SYSTEM:SYSUAF.DAT. Its primary maintenance vehicle is the AUTHORIZE utility.

2.1 STRUCTURE OF THE UAF

Each record in the UAF includes the following information:

- Name and password -- Identifies a user to the system at login time
- User identification code (UIC) -- Identifies a user by a group number and a member number (see Chapter 3 for more detailed information)
- Default file specification -- Provides default device and directory names for file access
- Default command language -- Names the default command interpreter as DCL or MCR
- Login command file -- Names a command procedure to be executed automatically at login time
- Login flags -- Allows the system manager to inhibit the use of the CTRL/Y function, restrict users to their default command interpreters, and/or lock user passwords
- Priority -- Specifies the base priority of the process created for the user at login time (see Chapter 4 for more detailed information)
- Resources -- Limits the system resources the user may consume (see Chapter 4 for more detailed information)
- Privileges -- Limits activities the user may perform (see Chapter 4 for more detailed information)

The software distribution kit provided with a new VAX/VMS system contains a UAF of four records:

- DEFAULT -- Serves as a template in creating user records in the UAF. A new user record is assigned the values of the DEFAULT record except where the system manager explicitly overrides those values. The DEFAULT record can be modified but cannot be deleted from the UAF.

AUTHORIZE

- SYSTEM -- Provides a means for the system manager to log in with full privileges. The SYSTEM record can be modified but cannot be deleted from the UAF.
- FIELD -- Permits DIGITAL field service personnel to check out a new system. The FIELD record can be deleted once the system is installed.
- SYSTEST -- Provides an appropriate environment for running the User Environment Test Package (UETP). The SYSTEST record can be deleted once the system is installed.

Typically, the system manager adds, deletes, and modifies records in the UAF provided with the distribution kit. The system manager can, however, delete the UAF with the DCL command DELETE (or rename the UAF with the DCL command RENAME) and create a new UAF with AUTHORIZE. A new UAF created in this manner contains just two records -- DEFAULT and SYSTEST.

2.2 AUTHORIZE UTILITY

The system manager runs the AUTHORIZE utility to modify the existing UAF or to create a new UAF. Table 2-1 summarizes the AUTHORIZE commands. The ADD, DEFAULT, and MODIFY commands act upon individual fields of UAF records through the specification of appropriate qualifiers. Table 2-2 lists the qualifiers, describes the corresponding fields, and specifies the defaults (as provided in the DEFAULT record in the software distribution kit). Table 2-3 specifies the values of the qualifiers in the SYSTEM, FIELD, and SYSTEST records (as provided with the software distribution kit).

Table 2-1
AUTHORIZE Command Summary

Format	Function
ADD username [qualifier ...]	Adds a record to the UAF
DEFAULT qualifier [...]	Modifies the DEFAULT record
EXIT	Returns the user to DCL command level
HELP	Lists the AUTHORIZE commands
LIST [userspec] [/FULL]	Creates a listing file of UAF records
MODIFY username qualifier [...]	Modifies a record in the UAF
REMOVE username	Deletes a record from the UAF
SHOW userspec [/BRIEF]	Displays UAF records

AUTHORIZE (Cont.)

Table 2-2
UAF Field Qualifiers

Qualifier	Default	Description
/ACCOUNT=account-name	blanks	Default account name (for example, a billing name or number) -- 1-8 characters
/ASTLM=value	10	AST queue limit (total asynchronous system trap (AST) operations and scheduled wake-up requests that can be outstanding at one time) -- integer; should be minimum of 2
/BIOLM=value	6	Buffered I/O count limit (total buffered (for example, terminal) I/O operations that can be outstanding at one time) -- integer; should be minimum of 2
/BYTLM=value	4096	Buffered I/O byte limit (total bytes that can be specified for transfer in outstanding buffered I/O operations) -- integer; should be minimum of 1024
/CLI=cli-name	DCL	Name of the default command interpreter -- 1-9 characters; should be DCL or MCR
/CPUTIME=delta-time	0	Maximum CPU time that a user's process can take per session -- unit of time in delta format; 0 means INFINITE
/DEVICE=device-name	blanks	Name of default device (must be a direct-access device) -- 1-15 alphanumeric characters; colon automatically added if omitted; at log-in time, a blank value is interpreted as the equivalent of SYS$SYSDISK
/DIOLM=value	6	Direct I/O count limit (total direct (usually disk) I/O operations that can be outstanding at one time) -- integer; should be minimum of 2

(continued on next page)

AUTHORIZE (Cont.)

Table 2-2 (Cont.)
UAF Field Qualifiers

Qualifier	Default	Description
/DIRECTORY= directory-name	[USER]	Name of default directory -- 1-31 characters; brackets must be typed to be included
/ENQLM=value	0	Unimplemented
/FILLM=value	20	Open file limit (total files that can be open at one time, including active network logical links) -- integer; should be minimum of 2
/FLAGS= ([[NO]DISCTLY] [,[NO]DEFCLI] [,[NO]LOCKPWD])	blanks	Log-in flags to disable CTRL/Y interrupts (if the intent of DISCTLY is only to force execution of the login command procedure, that procedure should issue a SET CONTROL_Y command before exiting), restrict the user to the default command interpreter (prohibit use of the /CLI qualifier at log-in time; however, the MCR command can still be used), and lock the user's password (prohibit use of the SET PASSWORD command) -- DISCTLY, DEFCLI, and LOCKPWD; NO in front of a flag name clears the flag; parentheses may be omitted for a single flag name; multiple flag names must be separated by commas
/LGICMD=filespec	blanks	Name of login command file -- standard file specification with defaults as follows: device is as specified for /DEVICE, directory is as specified for /DIRECTORY, file type is COM (if command interpreter is DCL) or CMD (if command interpreter is MCR)
/OWNER=owner-name	blanks	Name of owner (for billing purposes, for example) -- 1-32 characters

(continued on next page)

AUTHORIZE (Cont.)

Table 2-2 (Cont.)
UAF Field Qualifiers

Qualifier	Default	Description
/PASSWORD=password	USER	Password for logging in -- 1-31 characters; must be alphanumeric characters, dollar signs, or underlines
/PBYTLM=value	0	Unimplemented
/PGFLQUOTA=value	10000	Paging file limit (total pages that a user process can use in the system paging file) -- integer; should be minimum of 256
/PRCLM=value	2	Subprocess creation limit (total subprocesses that can exist at one time) -- integer
/PRIO=value	4	Default base priority for user processes -- integer; should be in the range of 0-31; 4 is the norm for timesharing users
/PRIVILEGES= ([NO]privname[,...])	NETMBX TMPMBX	Privileges allotted user -- names of privileges as detailed in Chapter 4 (see Table 4-1 for quick reference); NO in front of a privname removes the privilege; parentheses may be omitted for a single privname; multiple privnames must be separated by commas
/SHRFILLM=value	0	Unimplemented
/TQELM=value	10	Timer queue entry limit (total entries in the timer queue plus the number of temporary common event flag clusters that the user's process can have at one time) -- integer
/UIC=uic	[200,200]	User identification code as explained in Chapter 3 -- group number and member number separated by a comma and enclosed in brackets; each number must be octal in the range of 0-377

(continued on next page)

AUTHORIZE (Cont.)

Table 2-2 (Cont.)
UAF Field Qualifiers

Qualifier	Default	Description
/WSDEFAULT=value	150	Default working set limit (initial limit of a working set for the user process) -- integer; should be minimum of 50
/WSQUOTA=value	200	Default working set quota (maximum to which the user's process may raise working set limit) -- integer; should be minimum of 50

Table 2-3
Initial Values of SYSTEM, FIELD, and SYSTEST Records

Qualifier	SYSTEM	FIELD	SYSTEST
/ACCOUNT	SYSTEM	FIELD	SYSTEST
/ASTLM	20	10	10
/BIOLM	12	12	12
/BYTLM	20480	10240	20480
/CLI	DCL	DCL	DCL
/CPUTIME	0	0	0
/DEVICE	blanks	blanks	blanks
/DIOLM	12	12	12
/DIRECTORY	[SYSMGR]	[SYSMAINT]	[SYSTEST]
/ENQLM	0	0	0
/FILLM	20	20	20
/FLAGS	blanks	blanks	blanks
/LGICMD	blanks	blanks	blanks
/OWNER	SYSTEM MANAGER	FIELD SERVICE	SYSTEST-UETP

(Continued on next page)

AUTHORIZE (Cont.)

Table 2-3 (Cont.)
Initial Values of SYSTEM, FIELD, and SYSTEST Records

Qualifier	SYSTEM	FIELD	SYSTEST
/PASSWORD	MANAGER	SERVICE	UETP
/PBYTLM	0	0	0
/PGFLQUOTA	10000	10000	10000
/PRCLM	10	2	8
/PRIO	4	4	4
/PRIVILEGES	all	GRPNAM ALLSPOOL GROUP DIAGNOSE PRMCEB LOG_IO SETPRV TMPMBX PRMMBX PHY_IO NETMBX	CMKRNL CMEXEC SYSNAM GRPNAM DETACH LOG_IO GROUP PRMCEB PRMMBX TMPMBX NETMBX VOLPRO SYSPRV PHY_IO DIAGNOSE
/SHRFILLM	0	0	0
/TQELM	20	20	20
/UIC	[001,004]	[001,010]	[001,007]
/USRDATOFF	0	0	0
/WSDEFAULT	150	150	150
/WSQUOTA	1024	1024	1024

Typically, the system manager issues the following commands in setting up a new UAF:

- MODIFYs -- Modifies the passwords in the SYSTEM, FIELD, and SYSTEST records (FIELD and SYSTEST can be deleted if no longer needed)
- DEFAULT -- Modifies values in the DEFAULT record where the DIGITAL-provided values are not appropriate
- ADDs -- Adds new users

Modification of passwords is essential to prevent unauthorized access to system files.

AUTHORIZE (Cont.)

2.2.1 Invoking AUTHORIZE

The following commands invoke the utility:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
```

It is essential that the system manager's default device and directory be that of the system to access the system UAF. (Users may create and maintain UAF files in their own directories, using AUTHORIZE. Such files, however, have no effect on system operation.) As long as a UAF exists, the system responds with the following prompt:

```
UAF>
```

The system manager can then enter any of the commands listed in Table 2-1. These commands follow the standard rules of grammar as specified in the VAX/VMS Command Language User's Guide.

If no UAF exists (that is, the system manager has deleted it), the system issues an error message and the following prompt:

```
Do you want to create a new file?
```

A response of YES (or Y) results in creation of a new UAF containing a DEFAULT record and a SYSTEM record. These records are initialized with the values shown in Tables 2-2 and 2-3.

Use of the AUTHORIZE utility requires write access to SYS$SYSTEM:SYSUAF.DAT, which is normally restricted to users in the same group as the owner, or users with a system UIC or the SYSPRV privilege. The system manager can further restrict access to the UAF (to just himself, for example) with the DCL command SET PROTECTION. The system manager should not, however, expand access rights to the UAF.

2.2.2 ADD Command

ADD adds a user record to the UAF. It takes the form:

```
ADD username [qualifier ...]
```

username
: A string of 1 through 12 alphanumeric characters and underlines (dollar signs are permissible, but usually reserved for system names)

qualifier
: A qualifier as listed in Table 2-2

AUTHORIZE (Cont.)

Qualifiers not specified take their values from the DEFAULT record except that the default password is always USER (no matter what the password in the DEFAULT record is). Typically, the system manager takes defaults on the limits, priority, privileges, command interpreter, and sometimes device; as a result, the system manager types only the password, UIC, directory, owner, account, and sometimes device. The following example illustrates a typical ADD command:

```
UAF>ADD ROBIN /PASSWORD=SP0152 /UIC=[014,006]-
_/DEVICE=DB1 /DIRECTORY=[ROBIN]-
_/OWNER=CHRISTOPHER ROBIN /ACCOUNT=VMS
```

After the UAF record is successfully added, the system manager should create a first-level directory for the new user, specifying the device name, directory name, and UIC of the UAF record. Protection for the "ordinary" user is normally read, write, execute, and delete access for system, owner, and group processes, and read and execute access for world processes. The following command creates a first-level directory for user Robin:

```
$ CREATE DB1:[ROBIN] /DIRECTORY /OWNER_UIC=[014,006]
```

2.2.3 DEFAULT Command

DEFAULT modifies the DEFAULT record. It takes the form:

```
DEFAULT qualifier [...]
```

qualifier
    A qualifier as listed in Table 2-2

Qualifiers not specified in the command remain unchanged.

The system manager typically modifies the DEFAULT record when qualifiers normally assigned to a new user differ from the DIGITAL-supplied values. The qualifiers most often needing modification are as follows:

- /CLI -- If the command interpreter is MCR
- /DEVICE -- If most users have the same default device
- /LGICMD -- Where automation of initial housekeeping chores at login time is desired
- /PRIVILEGES -- Where users are normally given different privileges than the DIGITAL-supplied ones

The system manager almost always specifies a login command file. At some installations, the system manager simply provides each user directory with its own login command file by specifying /LGICMD as follows:

```
/LGICMD=LOGIN
```

AUTHORIZE (Cont.)

(The name of the file does not have to be LOGIN, although this convention predominates.) Login protocol, then, becomes the responsibility of each user.

Another approach provides a common login command file in a system directory:

```
/LGICMD=SYS$SYSDISK:[SYSMGR]SYLOGIN
```

The system manager maintains a login command file on behalf of all users. This command file may, at the end, contain a call to the user's login file:

```
$ @LOGIN
```

(Of course, in this case, all users must name their login command files LOGIN.COM.) In this way, the log-in protocol consists of a common command procedure followed by a user-specific command procedure.

The following example illustrates the DEFAULT command:

```
UAF>DEFAULT /DEVICE=DB1 /LGICMD=SYS$SYSDISK:[SYSMGR]SYLOGIN-
_/PRIVILEGES=(TMPMBX,GRPNAM,GROUP)
```

After modifying the default value record, the system manager should pencil in the changes in the middle column of Table 2-2.

2.2.4 EXIT Command

EXIT returns the system manager to DCL command level. It takes the form:

```
EXIT
```

The system manager can also return to DCL command level by typing <CTRL/Z>.

2.2.5 HELP Command

HELP lists and explains the AUTHORIZE commands. It takes the form:

```
HELP
```

2.2.6 LIST Command

LIST creates a listing file of reports for selected UAF records. It takes the form:

```
LIST [userspec] [/FULL]
```

userspec
: A user name, a wild card character (*), or a UIC; defaults to wild card

/FULL
: Request for a full report; omission results in a brief report

The name of the listing file is SYSUAF.LIS and can be printed with the DCL command PRINT.

Specification of a user name results in a one-user report. The following example lists a full report for user ROBIN:

```
UAF>LIST ROBIN /FULL
```

Specification of a wild card character results in reports for all users in ascending sequence by user name:

```
UAF>LIST *
```

Specification of a UIC results in reports for all users with the UIC. Wild card characters (*) can be used in specifying the UIC, as illustrated in the following examples:

Command	Description
LIST [014,006] /FULL	Lists a full report for the user (or users) with member number 006 in group 014
LIST [014,*]	Lists a brief report for all users in group 014, in ascending sequence by member number
LIST [*,006]	Lists a brief report for all users with a member number of 006
LIST [*,*]	Lists a brief report for all users, in ascending sequence by UIC

Users with the same UIC are listed in the order that they were added to the UAF. Full reports include the details of the limits, privileges, login flags, and command interpreter. Brief reports do not include the limits, login flags, and command interpreter, and summarize the privileges. The password is never listed. See Figure 2-1 under the SHOW command for examples.

2.2.7 MODIFY Command

MODIFY changes values in a user record. It takes the form:

```
MODIFY username qualifier [...]
```

username
: Name of user in the UAF

qualifier
: A qualifier as listed in Table 2-2

AUTHORIZE (Cont.)

Values not specified in the command remain unchanged. The following example changes the password for user ROBIN:

```
UAF>MODIFY ROBIN /PASSWORD=SP0172
```

Modifications to UAF records do not affect users logged in. The modifications take effect the next time the user logs in.

2.2.8 **REMOVE Command**

REMOVE deletes a user record from the UAF. It takes the form:

```
REMOVE username
```

username
    Name of a user in the UAF

The DEFAULT and SYSTEM records cannot be deleted. The following example deletes user ROBIN:

```
UAF>REMOVE ROBIN
```

2.2.9 **SHOW Command**

SHOW displays reports for selected UAF records. It takes the following form:

```
SHOW userspec [/BRIEF]
```

userspec
    A user name, wild card character (*), or UIC

/BRIEF
    Request for a brief report; omission results in a full report

Specification of a user name results in a 1-user report. The following example displays a full report for user ROBIN:

```
UAF>SHOW ROBIN
```

Specification of a wild card character results in reports for all users in ascending sequence by user name:

```
UAF>SHOW * /BRIEF
```

AUTHORIZE (Cont.)

Specification of a UIC results in reports for all users with the UIC. Wild card characters (*) can be used in specifying the UIC, as illustrated in the following examples:

Command	Description
SHOW [014,006]	Displays a full report for the user (or users) with member number 006 in group 014
SHOW [014,*] /BRIEF	Displays a brief report for all users in group 014, in ascending sequence by member number
SHOW [*,006] /BRIEF	Displays a brief report for all users with a member number of 006
SHOW [*,*] /BRIEF	Displays a brief report for all users, in ascending sequence by UIC

Users with the same UIC are listed in the order that they were added to the UAF. Full reports include the details of the limits, privileges, login flags, and command interpreter. Brief reports do not include the limits, login flags, and command interpreter, and summarize the privileges. The password is never listed. Figure 2-1 illustrates the display.

```
UAF>SHOW HUME /FULL

Username:  ROBIN          Owner:  CHRISTOPHER ROBIN
Account:   VMS            UIC:    [014,006]
CLI:       DCL            LGICMD: SYS$SYSDISK:[SYSMGR]SYLOGIN
Default Directory:  DB1:[ROBIN]
Login Flags:
PRIO:      4  BYTLM:        4096  BIOLM:          6
PRCLM:     2  PBYTLM:          0  DIOLM:          6
ASTLM:    10  WSDEFAULT:     150  FILLM:         20
ENQLM:     0  WSQUOTA:       200  SHRFILLM:       0
TQELM:    10  PGFLQUOTA:   10000  CPU:     no limit
Privileges:
  TMPMBX NETMBX
UAF>SHOW [014,*] /BRIEF
      OWNER              USERNAME         UIC      Account   Privs Pri Default Directory
CHRISTOPHER ROBIN      ROBIN          [014,006]  VMS       Normal  4 DB1:[ROBIN]
DAVID J. JONES         JONES          [014,011]  VMS       Normal  4 DB1:[JONES]
BOBBY SCHMAUTZ         SCHMAUTZ       [014,112]  VMS       Normal  4 DB1:[SCHMAUTZ]
```

Figure 2-1 Display Produced By SHOW Command

AUTHORIZE (Cont.)

2.2.10 **Messages**

Table 2-4 lists the messages issued by the AUTHORIZE utility (in alphabetical order).

Table 2-4
AUTHORIZE Error Messages

Message	Remedy
command is not unique	Enter command name in its entirety.
connect error	Respond to VAX-11 RMS message that follows.
device name too long to add trailing ":"	Restrict device name to 15 characters.
do you want to create a new file?	Respond YES (or Y) or NO (or N). See Section 2.2.1.
error during show	Respond to VAX-11 RMS message that follows.
error generating listing file	Respond to VAX-11 RMS message that follows.
error in UIC specification	Enter as [group,member] where group and member are octal values in the range 000-377, inclusive, or the wild card character (*)
error in user specification	Enter a user name, wild card character (*), or UIC. See LIST or SHOW specifications above.
error in value specification	Enter the value according to the specifications of Tables 2-1 and 2-2.
get error	Respond to VAX-11 RMS message that follows.
invalid command	Enter a valid command name as shown in Table 2-1.
invalid option name	Enter a valid qualifier as shown in Table 2-2.
invalid password syntax	Enter 1-31 alphanumeric, dollar sign ($), or underline (_) characters.
invalid privilege name	Enter a valid privilege name as shown in Table 4-1 (Chapter 4).

(continued on next page)

AUTHORIZE (Cont.)

Table 2-4 (Cont.)
AUTHORIZE Error Messages

Message	Remedy
invalid response	Respond to "Do you ..." message that follows.
invalid username; user-name already exists	Enter a unique user name.
listing file SYSUAF.LIS complete	Do not respond -- informational message.
missing argument for option	Enter a qualifier as listed in Table 2-2.
missing username	Enter a user name as listed in Table 2-1.
missing user specification	Enter a user specification as listed in Table 2-1.
no modifications made	Do not respond -- informational message.
no user matched specification	Enter a user specification that matches at least one existing user account.
option name not unique	Enter at least the first four characters of a qualifier.
privilege name not unique	Enter at least the first four characters of a privilege name.
put error	Respond to the VAX-11 RMS message that follows.
quoted string missing end quote	Enter the command with both quotes.
string too long for field	Restrict character string to length specified in Table 2-2.
superfluous parameter detected	Do not enter the superfluous text.
The DEFAULT record may not be removed	Do not respond -- informational message.
The SYSTEM record may not be removed	Do not respond -- informational message.
UAF modified	Do not respond -- informational message.
unable to add user record	Respond to the VAX-11 RMS message that follows.

(continued on next page)

AUTHORIZE (Cont.)

Table 2-4 (Cont.)
AUTHORIZE Error Messages

Message	Remedy
unable to create SYSUAF.DAT	Respond to the VAX-11 RMS message that follows.
unable to delete record	Respond to the VAX-11 RMS message that follows.
unable to obtain DEFAULT record	Respond to the VAX-11 RMS message that follows. If DEFAULT record cannot be accessed, AUTHORIZE cannot be used to modify the UAF.
unable to open SYSUAF.DAT	Respond to the VAX-11 RMS message that follows.
unable to update record	Respond to the VAX-11 RMS message that follows.
username does not exist	Enter an existing user name.
username syntax error	Enter alphanumeric, underline, and dollar sign characters only.
username too long	Enter a user name no longer than 12 characters.
user record removed	Do not respond -- informational message.
user record successfully added	Do not respond -- informational message.
user record updated	Do not respond -- informational message.
value too large for field	Enter a smaller value for the qualifier, as specified in Table 2-2.
Warning: ADD command does not use DEFAULT record password	Do not respond -- informational message.
writing listing file	Do not respond -- informational message.

For explanations of the VAX-11 RMS messages, see the VAX/VMS System Messages and Recovery Procedures Manual.

2.3 OPERATIONS

During normal log-in activities, the UAF (that is, the file named SYS$SYSTEM:SYSUAF.DAT) controls user access of the system:

- User validation -- To use the system at all, a user must state (1) a name for which a record exists in the UAF and (2) the password associated with that name. Otherwise, the following message appears:

  user validation error

- Level of Access -- A user who successfully logs in receives UIC access rights, a default file specification, a default command interpreter, login flags, a default base priority, resource limits, and privileges as stated in the UAF.

AUTHORIZE can run concurrently with user login operations as long as VAX-11 RMS file sharing is in effect (that is, RMSSHARE has been run). A slight chance exists that a user login operation might fail on a record lock, but the user need only try again to remedy the situation.

The system manager can also implement the following alternatives to normal login activities:

- Open system -- The system manager implements an open system by deleting (or renaming) the UAF. When no UAF exists, a user may log in with any name and password. The system assigns the following values to such a user:

  - Name -- JOB_CONTROL
  - UIC -- [010,040]
  - Command interpreter -- DCL
  - Login flags -- None
  - Priority -- Value of SYSGEN parameter DEFPRI
  - Resources -- Values of PQL SYSGEN parameters
  - Privileges -- All

  The process name is normally the name of the device on which the user logged in.

- Inaccessible UAF -- If the UAF is locked or disabled, the system manager -- or operator -- can log in on the console terminal with any name and password.

- Alternate UAF -- At bootstrap time, the system manager can select an alternate UAF named SYS$SYSTEM:SYSUAFALT.DAT by using a system generation file in which the UAFALTERNATE parameter has a value of 1, or by changing this value to 1 during a conversational bootstrap operation. (See Chapter 13 for creating system generation files.) Naturally, a file named SYSUAFALT in UAF format must exist at bootstrap time. The system manager can create or modify such a file with the following commands:

  ```
  $ SET DEFAULT SYS$SYSTEM
  $ ASSIGN SYSUAFALT SYSUAF
  $ RUN AUTHORIZE
  ```

  Any time after booting the system with an alternate UAF, the system manager can switch to the standard UAF (that is, SYS$SYSTEM:SYSUAF.DAT) with the following command:

  ```
  $ DEASSIGN SYSUAF /SYSTEM
  ```

  (The system initially changes to the alternate UAF by assigning SYSUAF as a logical name for SYSUAFALT.) If the UAFALTERNATE parameter is set at bootstrap time and no alternate UAF exists, the system behaves as an open system.

  The system manager can also switch to the alternate UAF after bootstrap time with the following command:

  ```
  $ ASSIGN SYSUAFALT SYSUAF /SYSTEM
  ```

  (For this mode of operation, the alternate UAF can have any name in the range of 1 to 9 characters.)

2.4 USER DATA AREAS

Users can access UAF records by means of VAX-11 RMS for storage and retrieval of up to 255 bytes of user data. The format of a UAF record is defined in the module $UAFDEF in SYS$LIBRARY:LIB.MLB. The records can be accessed sequentially or through the following keys:

- User name -- The primary key is the user name (as specified to AUTHORIZE), a character field of size UAF$S_USERNAME located at relative offset UAF$T_USERNAME.

- UIC -- The secondary key is the UIC, located at relative offset UAF$L_UIC, consisting of two binary subfields of one word each. The subfields are named UAF$W_GRP (high-order word) and UAF$W_MEM (low-order word).

- Member -- The tertiary key is the member field of the UIC, located at relative offset UAF$W_MEM, consisting of one binary word.

AUTHORIZE uses the alternate keys to display UAF records in UIC order. The tertiary key may be dropped in future releases.

To place data in a UAF record, the user should take the following steps, which are designed to protect programs against future changes to the format of the UAF:

1. Read the UAF record.

2. Check the value of UAF$W_USRDATOFF. If it is zero, insert the current size of the record, as found in the VAX-11 RMS record access block (RAB), into UAF$W_USRDATOFF. (In VAX/VMS Version 2.0, the system initializes this field to zero. Inserting the current size of the record has the effect of placing the user data at the end of the record. However, future changes to the UAF might require the system to fix the location of the user data. In this event, the system would initialize UAF$W_USRDATOFF and the user must not change its value.)

3. Insert the user data at the relative offset pointed to by UAF$W_USRDATOFF. The data must take the form of a counted string -- the first byte must specify the number of bytes that follow and the total bytes must not exceed 256.

4. Modify the size of the record in the RAB to reflect the addition of the user data area, unless the current size of the record exceeds the end of the user data area (that is, the contents of UAF$W_USRDATOFF plus the length of the user data). (In Version 2.0, the user data area, when specified as outlined above, resides at the end of the record, so that modification of the RAB is essential. It is possible, however, that a future version of AUTHORIZE might fix the size (at 256 bytes) and location of the user data area, and locate additional system data after the user data. In this event, the user must not modify the size of the record.)

5. Update the record.

The user can now access the counted string by referring to UAF$W_USRDATOFF. For additional updates, the user does not modify UAF$W_USRDATOFF (the user should not modify UAF$W_USRDATOFF if its value is not zero), but does modify the first byte of the counted string and the record size in the RAB (withstanding the considerations for future changes) if the record size changes. The user should lock records while accessing them.

The format of the UAF record and the way in which the system modifies it is subject to change without notice. (For format changes, however, a utility will be provided to update old UAFs.) Adherance to the above guidelines will minimize reprogramming in the event of system changes.
pg

2.5 UAF VERSION 2.0 UPGRADE

The format of the UAF has changed for VAX/VMS Version 2.0 from sequential to indexed organization. The installation upgrade procedure (see the VAX-11 Software Installation Guide) automatically upgrades SYS$SYSTEM:SYSUAF.DAT to the proper format. The system manager can upgrade other UAFs by invoking the command procedure SYS$SYSTEM:[SYSUPD]CVTUAF as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ @SYS$SYSDISK:[SYSUPD]CVTUAF
```

The old UAF must be accessible to the command procedure. If the old system volume cannot be mounted on a second disk drive (for example, in a single-disk system), the system manager must be sure to include the old UAF with the user files that are saved on another medium at installation time.

In response to a prompt from the command procedure, the system manager types the name of the device on which the old UAF resides.

Execution of the command procedure requires read access to the old UAF and write access to SYS$SYSTEM.

In Version 2.0, the UAF is accessed with file sharing enabled, and updates to the UAF are made on a per-record basis, obviating the need for a temporary UAF and a new version of the UAF after each AUTHORIZE run. Updates become effective as soon as AUTHORIZE commands are entered, not after the termination of AUTHORIZE. (For this reason, the system manager should not enter temporary values with the intent of fixing them later in the run.)

CHAPTER 3

PROTECTION AND CONTROL

The system manager assists in data protection and interprocess control primarily in the following ways:

- User identification code (UIC) -- By assigning each user a UIC that reflects the user's relationships with other users
- Passwords -- By changing the passwords of the SYSTEM, FIELD, and SYSTEST records in the UAF, and by assigning confidential passwords to users of the system
- Privileges -- By strictly limiting privileges to those who need them
- System data and devices -- By applying proper protection to system data files, interactive terminals, and card readers

3.1 USER IDENTIFICATION CODE

A UIC consists of two numbers in the range of 0 through 377 (without the radix indicators), separated by a comma and enclosed in brackets. The first number is the group number of the UIC, and the second number is the member number. For example, [320,101] is a UIC whose group number is 320 octal and whose member number is 101 octal.

The system manager explicitly assigns a UIC to each user in the UAF. The system implicitly assigns this UIC to the detached process created for the user at login time. User processes pass on this UIC to any subprocesses they create. Detached processes created by user processes require explict assignment of a UIC.

Processes can further assign UICs to files, file-structured volumes, devices, mailboxes, shared pages in memory (global sections), common event flags, and logical name tables. For most users, the assignment of UICs to these structures is automatic and transparent. Suppose, for example, that a user with UIC [320,101] logs in and issues the following command:

```
$ CREATE SCRATCH.RNO
```

The system assigns a UIC of [320,101] to SCRATCH.RNO in the user's default directory.

3.2 PROTECTION OF DATA STRUCTURES AND DEVICES

For purposes of protection, four different categories of user exist with respect to data structures and devices:

- Owner -- A user whose UIC is identical with the UIC of the data structure or device. The owner of a file is ordinarily the creator of the file.
- Group -- A user whose group number is identical with the group number of the data structure or device.
- System -- A user whose group number is between 0 and the value of the MAXSYSGROUP system parameter (usually 10 octal), inclusive.
- World -- Any user. This category includes users who are not owners of the data structure or device in question, are not members of the same group as the owner, and are not system users.

Figure 3-1 illustrates the relationships of these four types of user to a file whose UIC is [100,100].

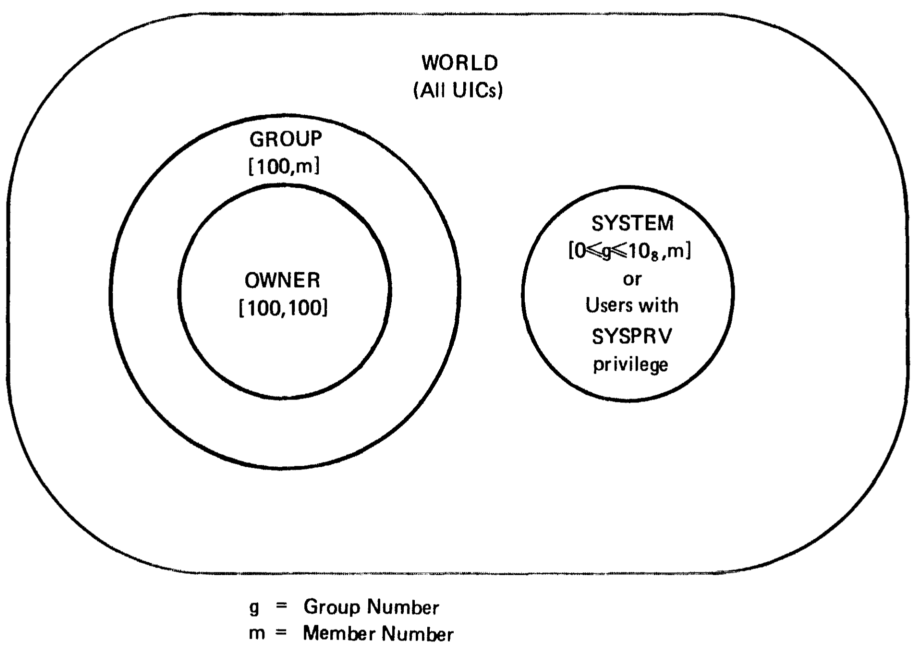

Figure 3-1 Classification of Users with Respect to a File with a UIC of [100,100]

Users are permitted four types of access to protected data structures and devices:

- Read
- Write
- Execute (or physical I/O for shareable non-file-structured devices)
- Delete (or logical I/O for shareable non-file-structured devices)

Each category of user is permitted or denied each type of access through association of the data structure or device with a 16-bit protection mask of the following form:

	WORLD	GROUP	OWNER	SYSTEM
DELETE	15	11	7	3
EXECUTE	14	10	6	2
WRITE	13	9	5	1
READ	12	8	4	0

When a bit is on, that category of user is denied that type of access to the protected data structure or device. For example, if bits 12, 13, and 15 of a file's protection mask are on, world users are denied read, write, and delete access to that file.

DCL commands permit the user to set values for protection masks in the following form:

```
(user-category:access-categories[,...])
```

user-category
    OWNER, GROUP, SYSTEM, or WORLD

access-categories
    Any combination of R (read), W (write), E (execute; or P for physical I/O), or D (delete; or L for logical I/O)

R, W, E, and D represent the access categories permitted the user category (that is, the categories for which the mask bits are set off). In the following example, a user creates a file that denies read, write, and delete access to the world:

```
$ CREATE SCRATCH.RNO -
_/PROTECTION=(SYSTEM:RWED, -
_OWNER:RWED,GROUP:RWED,WORLD:E)
_
```

System and I/O services require specification of the mask as four hexadecimal digits. The low-order digit specifies system access, the next digit specifies owner access, the next digit specifies group access, and the high-order digit specifies world access. In the following example, a user creates a mailbox that denies read and write access to the world:

```
$CREMBX_S PRMFLG=#0,CHAN-MBXCHAN,MAXMSG=MBUFLEN-
     BUFQUO=#384,PROMSK=#^X3000,LOGNAM=MBLOGNAM
```

The protection mask is specified as hexadecimal 3000 -- the two low-order bits of the high-order digit are on.

The following sections describe the protection provided for each of the data structures and devices.

3.2.1 Files and File-Structured Volumes

The default Files-11 protection is to deny write and delete access to the world and to allow all other categories of access. A user can reset the default Files-11 protection:

```
$ SET PROTECTION-(WORLD:E)  /DEFAULT
```

Default protection for the world permits execute access only (while the other user categories do not change).

The default protection is applied to all files created by the user unless explicitly overridden:

```
$ CREATE SCRATCH.RNO /PROTECTION-(SYSTEM:RWED,-
_OWNER:RWED,GROUP:RWED,WORLD:RWE)
```

On an explicit specification, all categories must be included.

The protection on a file can also be changed after the file is created:

```
$ SET PROTECTION=(WORLD=R) SCRATCH.RNO
```

The world has only read access to SCRATCH.RNO (while the other user categories do not change).

Figure 3-2 illustrates how protection works to control access to a file that has a UIC of [100,100] and default Files-11 protection.

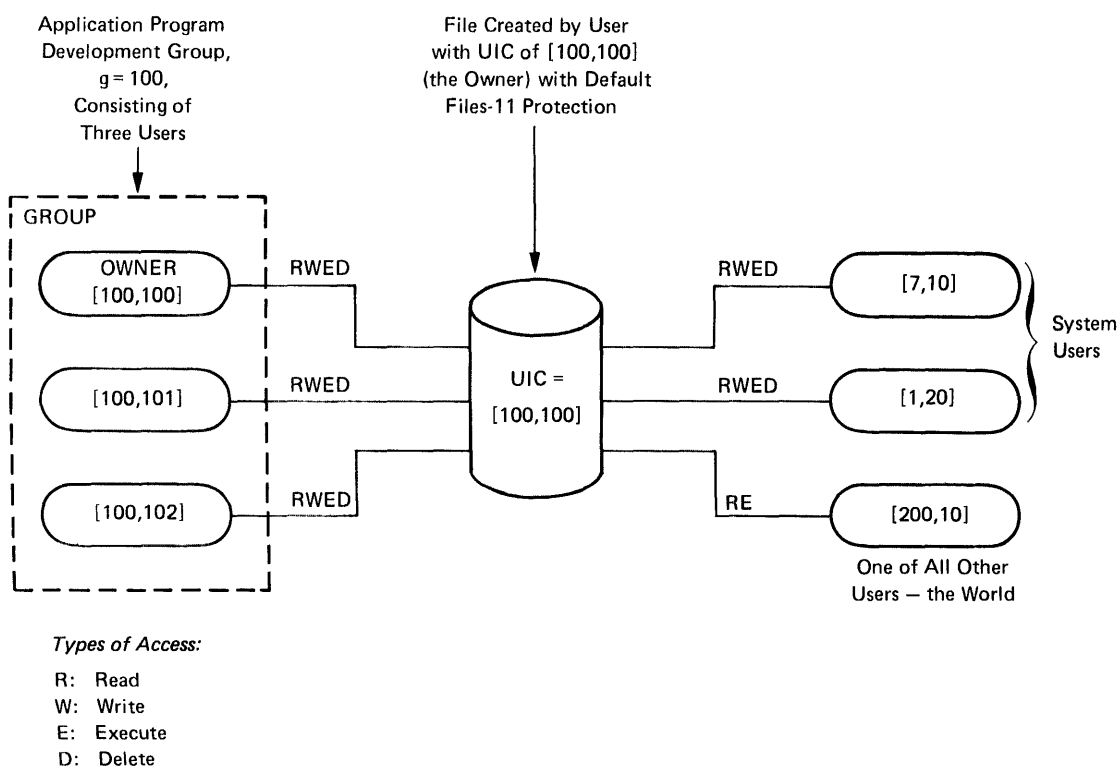

Figure 3-2 Controlling Access to a File

The scheme for protecting file-structured volumes is similar to the one for protecting files, except that execute access to a volume gives a user the right to create files on the volume.

Users who have been assigned the VOLPRO privilege can override the protection of file-structured volumes and users who have been assigned the BYPASS privilege can bypass the UIC protection mechanism for all files. In addition, the SYSPRV privilege grants users the same access rights to files and file-structured volumes as system users. (See Chapter 4 for descriptions of all privileges.)

3.2.2 Nonshareable Non-File-Structured Devices

Nonshareable devices not used for storing files (for example, terminals and card readers) can be protected from allocation by setting or not setting the read bits in the protection masks associated with the devices. (The other bits have no meaning in protection masks for nonshareable non-file-structured devices.) Protection can be applied to all terminals at bootstrap time with the TTY_PROT and TTY_OWNER system parameters (see Chapter 12) and on a device-by-device basis during system operation with the SET PROTECTION /DEVICE command (see the VAX/VMS Operator's Guide). Setting protection on a terminal does not inhibit logins, but restricts the processes that can accept and display data on the termianl.

Users who have been assigned the BYPASS privilege can bypass the UIC protection mechanism for all devices. In addition, users with the SYSPRV privilege have the same access rights to devices as system users.

3.2.3 Mailboxes

Mailboxes (both temporary and permanent) are protected by a code, or mask, that is similar to the code used in protecting files. As with files, four types of user (defined by UIC) can gain access to a mailbox: owner, group, system, and world. However, only two types of access -- read access and write access -- are meaningful to users of a mailbox. Thus, when creating a mailbox, the user can specify read access and write access to the mailbox separately for each type of user.

Because temporary mailboxes are customarily used for interprocess communication between cooperating processes that have the same group number, they are of special interest here. The logical names of these mailboxes are entered in the group logical name table. Temporary mailboxes thus have two layers of protection. First, easy access to the logical names of temporary mailboxes is granted only to users who have the same group number as the creator of the mailbox; other users have no such easy access. Second, by use of the protection mask, the creator of the temporary mailbox grants additional security to the mailbox. As a rule, users who are not in the same group as the creator are totally excluded from using the mailbox.

Furthermore, the creator of a temporary mailbox can discriminate between owners and other group members in granting read access and write access to a temporary mailbox. For example, owners may be allowed only read access or only write access to the mailbox, but other members of the group may be allowed both read access and write access to the mailbox.

3.2.4 Shared Pages in Memory (Global Sections)

All shared pages in memory (global sections) are protected by a code, or mask, that is the same as the one used to protect other files. Thus, the creator of a global section can specify read access, write access, execute access, and delete access separately for each of the four types of user.

Of special interest are group global sections; such a section can be shared only by processes that have the same group number as the global section. This group number is that of the creator of the global section. Thus, group global sections have two layers of protection. First, if a process is to map to a global section, the group numbers of the process and of the global section must be identical. Second, by using the protection mask, the creator of a group global section grants additional security to that global section.

3.2.5 Common Event Flags

Common event flags, which are used in establishing communication and synchronization among cooperating processes in the same group, are protected by UIC.

When a common event flag cluster is created, it takes on the UIC of the process that requested its creation and, thus, the UIC of the user whose process created the cluster. To control access to a common event flag cluster, a protection indicator is used with the UIC. If the value of this indicator is 0 (the default), both read access and write access to the cluster are extended to all processes and users in the owner's group. If the value of this indicator is 1, both read access and write access to the cluster are restricted to owners of the cluster and to subprocesses that have been created on behalf of these owners.

3.2.6 Group Logical Name Table

Entries in the group logical name table are protected by group number.

When a logical name and its equivalence name are entered into the group logical name table, the group number of the process that created the logical name is associated with the entry. Access to an entry in the group logical name table is thus restricted to users and processes that have the same group number as the entry.

3.3 FORMING GROUPS

In setting up a group, the system manager aims toward two goals:

- Sharing -- Users who typically share data and/or control one another's processes should be assigned to the same group.
- Protection -- Users who should not have access to each other's data or control over each other's processes should be assigned to separate groups.

In addition to group number, user privilege determines the ability of one process to control another process or access its data:

- No privilege -- A process needs no special privilege to access its own data structures or control its own subprocesses.
- GROUP privilege -- A process needs GROUP privilege to control processes belonging to other members of the same group.
- WORLD privilege -- A process needs WORLD privilege to control processes belonging to users outside one's group.

The importance of properly setting up groups of users should not be underestimated. As the system is used more and more and relationships among users, their processes, and their data structures become increasingly complex, redefining groups becomes more difficult.

Normally, the system manage should not assign the same UIC to more than one user, and the status of system user should be limited to the SYSTEM, FIELD, and SYSTEST accounts.

3.4 PASSWORDS

It is essential that the system manager change the passwords for the SYSTEM, FIELD, and SYSTEST records to values that are reasonably obscure, are difficult to deduce, and exceed six characters. Similarly, the system manager should afford users of the system individual protection by assigning obscure passwords and suggesting that users change their passwords from time to time.

3.5 PRIVILEGES

Chapter 4 discusses privileges in detail. The system manager should not assign privileges above the category of normal except where (1) the user demonstrates a need for the privilege and (2) the user is not likely to misuse the privilege. In addition, the system manager should not install user's images with amplified privileges (see Chapter 7) unless the above conditions are met.

3.6 SECURING SYSTEM DATA AND DEVICES

The system manager (UIC [1,4]) should own all system files. Protection on all system executable and library files should permit only read and execute access to the world. Protection on the following files should permit no access to the world:

```
SYS$SYSTEM:PAGEFILE.SYS
SYS$SYSTEM:SWAPFILE.SYS
SYS$SYSTEM:SYSDUMP.DMP
SYS$SYSTEM:SYSUAF.DAT
SYS$SYSTEM:RMTNODE.DAT
```

These files contain sensitive data, allowing a user with read access to them a degree of control over the system.

The software distribution kit provides the required protection for these files. However, the system manager could easily compromise a sensitive system file by copying it without applying the proper protection. To prevent such a situation, the system manager should always:

- Copy utilities -- Use the VAX-11 utilities DSC1 and DSC2. These utilities preserve the protection assigned to files. If another utility (RMS-11 BACKUP or RESTORE, for example) is used, the system manager should immediately check the protection on the copied file (DIRECTORY /PROTECTION command) and apply the proper protection (SET PROTECTION command) as needed.

- COPY command -- Log in as the system manager (UIC [1,4]) and set the protection explicitly when copying system files (that is, use the /PROTECTION qualifier).

These procedures require a conscious effort on the part of the system manager to (1) realize that a sensitive system file is being copied and (2) actively ensure that the copy is protected the same as the original.

The system manager should protect interactive terminals and card readers from allocation by processes other than system processes. If general access to these devices is permitted, a user's process could tie up another user's terminal, or -- worse -- monitor another user's input to pick up such sensitive data as the user's password. The system manager can restrict the allocation of all terminals to system processes through the following system parameter specifications:

- TTY_PROT -- Specify the value of TTY_PROT as %X1110. This protection mask allows only system processes to allocate terminals.

- TTY_OWNER -- Specify the value of TTY_OWNER as %X10004, the UIC of the system manager.

Protection on noninteractive terminals can be reset individually with SET PROTECTION /DEVICE commands. (These commands should be placed in the site-specific start-up command procedure; see Chapter 8.) The SET PROTECTION /DEVICE command should also be used to protect card readers from allocation by nonsystem processes.

CHAPTER 4

RESOURCE CONTROL

This chapter contains detailed descriptions of the resource control attributes that the system manager assigns to a user when creating a record in the UAF:

- Limits on the use of reusable system resources
- Base priority used in scheduling the process that the system creates for the user
- Privileges of using restricted and sensitive system functions

In addition, this chapter discusses the accounting log file.

4.1 LIMITS

Limits are set on system resources that can be reused, for example, the amount of memory that a process can have in use for queued I/O requests. Each user of the system is limited in the use of system resources. The system manager sets up these limits when the user is defined to the system. Most limits restrict the use of system memory.

Limits can be either pooled, deductible, or nondeductible in the way in which a process shares its allotment of a resource with the subprocesses that it creates. If the limit on the use of a resource is pooled, a process and created subprocesses share the total limit on a first-come first-served basis. If the limit on the use of a resource is deductible, a subprocess is allotted a portion of the total limit; the portion given to the subprocess is deducted from the total limit. If the limit is nondeductible, the subprocess is allotted the total limit of the creating process; there is no deduction from the allotment of the creating process.

In creating a UAF record, the system manager assigns values to the following limits:

- AST queue limit (ASTLM)
- Buffered I/O count limit (BIOLM)
- Buffered I/O byte count limit (BYTLM)
- CPU time limit (CPUTIME)
- Direct I/O count limit (DIOLM)
- Open file limit (FILLM)

- Paging file limit (PGFLQUOTA)
- Subprocess creation limit (PRCLM)
- Timer queue entry limit (TQELM)
- Default working set size (WSDEFAULT)
- Working set quota (WSQUOTA)

These limits are described in the following sections. Usually, the system manager simply assigns the default values for these limits. However, see Chapter 11 for a discussion of WSDEFAULT and WSQUOTA.

4.1.1 AST Queue Limit (ASTLM)

The AST queue limit (ASTLM) limits the sum of the following:

- The number of asynchronous system trap (AST) requests that a user's process can have outstanding at one time
- The number of scheduled wake-up requests that a user's process can have outstanding at one time

This limit affects not only all system services that accept an AST address as an argument, but also the Schedule Wakeup system service.

ASTLM is a nondeductible limit with a suggested minimum value of 2.

4.1.2 Buffered I/O Count Limit (BIOLM)

The buffered I/O count limit (BIOLM) limits the number of outstanding buffered I/O operations permitted a user's process.

A buffered I/O operation is an I/O operation in which the data transfer takes place from an intermediate buffer in the system pool, not from a process-specified buffer. Buffered operations for a user process include terminal I/O, card reader input, and unspooled printer output. During a buffered I/O operation, the pages containing the process-specified buffer that is given as an argument to the Queue I/O Request system service need not be locked in memory.

BIOLM is a nondeductible limit with a suggested minimum value of 2.

4.1.3 Buffered I/O Byte Count Limit (BYTLM)

The buffered I/O byte count limit (BYTLM) limits the amount of buffer space that a user's process can use.

This buffer space is used for buffered I/O operations and for the creation of temporary mailboxes.

BYTLM is a pooled limit with a suggested minimum value of 1024.

4.1.4 **CPU Time Limit (CPUTIME)**

The CPU time limit (CPUTIME) limits the amount of CPU time that a user's process can use per interactive session or batch job.

The time must be specified in delta format -- hh:mm:ss.ss.

CPUTIME is a deductible limit with a suggested minimum value of 10 milliseconds (but not across sessions or jobs).

4.1.5 **Direct I/O Count Limit (DIOLM)**

The direct I/O count limit (DIOLM) limits the number of outstanding direct I/O operations permitted a user's process.

A direct I/O operation is an I/O operation in which the data transfer takes place directly from a process-specified buffer. Direct I/O operations for a user process typically include disk and tape I/O. The pages containing this buffer are locked in memory by the operating system during the direct I/O operation.

DIOLM is a nondeductible limit with a suggested minimum value of 2.

4.1.6 **Open File Limit (FILLM)**

The open file limit (FILLM) limits the number of files that a user's process can have open at one time. This limit includes the number of network logical links that can be active at the same time.

FILLM is a pooled limit with a suggested minimum value of 2.

4.1.7 **Paging File Limit (PGFLQUOTA)**

The paging file limit (PGFLQUOTA) limits the number of pages that the user's process can use in the system paging file.

The paging file provides temporary disk storage for pages forced out of memory by a memory management operation.

PGFLQUOTA is a pooled limit with a suggested minimum value of 256.

4.1.8 **Subprocess Creation Limit (PRCLM)**

The subprocess creation limit (PRCLM) limits the number of subprocesses a user's process can create.

The process that is created when a user logs in to the system can in turn create subprocesses. These subprocesses are all accountable to the user and share the resources allotted to the initial process.

PRCLM is a pooled limit with a suggested minimum value of 0.

4.1.9 **Timer Queue Entry Limit (TQELM)**

The timer queue entry limit (TQELM) limits the sum of the following:

- The number of entries that a user's process can have in the timer queue
- The number of temporary common event flag clusters that a user's process can have

This limit does not govern the creation of permanent event flag clusters.

Timer queue entries are used in time-dependent scheduling; common event flags are used in synchronizing activities among groups of cooperating processes.

TQELM is a pooled limit with a suggested minimum value of 0.

4.1.10 **Working Set Default (WSDEFAULT)**

The working set default (WSDEFAULT) sets the initial working set limit for a user's process.

WSDEFAULT is a nondeductible limit with a suggested minimum value of 50. If the value specified exceeds the value of WSQUOTA (below), the lesser value is used.

4.1.11 **Working Set Quota (WSQUOTA)**

The working set quota (WSQUOTA) restricts the size to which the working set limit of a user's process can be expanded. This enlargement of the working set limit (the initial size of which is determined by WSDEFAULT) is accomplished by use of the Adjust Working Set Limit system service or by the SET WORKING_SET command.

WSQUOTA is a nondeductible limit with a suggested minimum value of 50. If the value specified exceeds the value of the WSMAX system generation parameter, the lesser value is used.

4.2 **PRIORITY**

A user's priority is the base priority used in scheduling the process that the system creates for the user. There are 32 levels of software priority in the VAX/VMS system, 0 through 31. The highest priority is 31; the lowest is 0. The range of priorities for timesharing processes is 1 through 15; the range for real-time processes is 16 through 31.

Processes with real-time priorities are scheduled strictly according to base priority; in other words, the executable real-time process with the highest base priority is executed first. Processes with timesharing priorities are scheduled according to a slightly different principle to promote overlapping of computation and I/O activities.

In the user's account record of the UAF, the default value of a user's priority is 4; for practical purposes, the minimum value is 1. The priority for timesharing users should remain at the default. Attempting to give some users an edge over other users by varying priorities usually results in ragged performance, as the system reacts sharply to even small priority differences. The system manager should never specify a value over 31 (system operation will be unpredictable).

4.3 PRIVILEGES

Privileges restrict the performance of certain system activities to certain users. These restrictions protect the integrity of performance of the operating system and thus the integrity of service provided users. The system manager should grant privileges to each user on the basis of two factors: (1) whether the user has the skill and experience to use the privilege without disrupting the system and (2) whether the user has a legitimate need for the privilege.

Privileges fall into seven categories according to the damage that the user possessing them could cause the system:

- None - No privileges
- Normal - Minimum privileges to effectively use the system
- Group - Potential to interfere with members of the same group
- Devour - Potential to devour noncritical system-wide resources
- System - Potential to interfere with normal system operation
- File - Potential to compromise file security
- All - Potential to control the system

A user cannot execute an image that requires a privilege the user does not possess unless the image is installed as a known image with the privilege in question. (See Chapter 7 for instructions on installing known images.) Execution of a known image with privileges temporarily (for the duration of the image's execution) grants those privileges to the user process executing the image.

A user's privileges are recorded in the user's UAF record in a 64-bit privilege vector. When a user logs in to the system, the user's privilege vector is stored in the header of the user's process. In this way, the user's privileges are passed on to the process created for the user. A user with the SETPRV privilege can modify the privilege vector with the SET PROCESS /PRIVILEGES command.

Table 4-1 lists the privileges by category and gives brief, general definitions of them. The following sections describe each privilege in detail in alphabetical order.

Table 4-1
VAX/VMS Privileges

Category	Privilege	Activity Permitted
None	None	None requiring privileges
Normal	MOUNT	Execute mount volume QIO
	NETMBX	Create network connections
	TMPMBX	Create temporary mailbox
Group	GROUP	Control processes in the same group
Devour	ACNT	Disable accounting
	ALLSPOOL	Allocate spooled devices
	BUGCHK	Make bugcheck error log entries
	EXQUOTA	Exceed disk quotas
	GRPNAM	Insert group logical names in the name table
	PRMCEB	Create/delete permanent common event flag clusters
	PRMGBL	Create permanent global sections
	PRMMBX	Create permanent mailboxes
	SHMEM	Create/delete structures in shared memory
System	ALTPRI	Set base priority higher than allotment
	OPER	Perform operator functions
	PSWAPM	Change process swap mode
	WORLD	Control any process
Files	DIAGNOSE	Diagnose devices
	SYSGBL	Create system-wide global sections
	VOLPRO	Override volume protection
All	BYPASS	Disregard UIC protection
	CMEXEC	Change to executive mode
	CMKRNL	Change to kernel mode

(continued on next page)

Table 4-1 (Cont.)
VAX/VMS Privileges

Category	Privilege	Activity Permitted
All (Cont.)	DETACH	Create detached processes
	LOG_IO	Issue logical I/O requests
	PFNMAP	Map to specific physical pages
	PHY_IO	Issue physical I/O requests
	SETPRV	Enable any privilege
	SYSNAM	Insert system logical names in the name table
	SYSPRV	Attain system user status

4.3.1 ACNT Privilege

Only a user who has the ACNT privilege can create subprocesses or detached processes in which accounting is disabled. Thus, only such a privileged user can issue the RUN command with the /NOACCOUNTING qualifier or inhibit accounting in the $CREPRC system service.

4.3.2 ALLSPOOL Privilege

The ALLSPOOL privilege allows the user's process to allocate a spooled device by executing the Allocate Device system service, or the user is allowed to allocate a spooled device by using the ALLOCATE command.

The Allocate Device system service lets a process allocate, or reserve, a device for its exclusive use. A shareable mounted device cannot be allocated.

The system manager should grant this privilege only to users who need to perform logical or physical I/O operations to a spooled device. Ordinarily, the privilege of allocating a spooled device is granted only to symbionts.

4.3.3 ALTPRI Privilege

The ALTPRI privilege allows the user's process to (1) increase its own base priority and (2) set the base priority of another process to a value higher than its own base priority.

The base priority is increased by executing the Set Priority system service or the SET PROCESS/PRIORITY command. As a rule, this system service lets a process set its own base priority or the base priority of another process. However, one process can only set the priority of a second process if (1) the second process is a subprocess of the first or (2) the first process has process control privilege (GROUP or WORLD) over the second.

With the same privilege a process can create a process with a base priority higher than its own. Such a process is created by using an optional argument to the Create Process system service or to the RUN command.

The system manager should not grant this privilege widely; if many users have the unrestricted ability to set base priorities, the fair and orderly scheduling of processes for execution can easily be disrupted.

4.3.4 BUGCHK Privilege

The use of this privilege should be restricted to system software supplied by DIGITAL that uses the VAX/VMS Bugcheck Facility.

4.3.5 BYPASS Privilege

The BYPASS privilege allows the user's process read, write, execute, and delete access to all files, bypassing UIC protection.

The system manager should grant this privilege with extreme caution, as it suspends all file protection. It should be reserved for use by well tested, reliable programs and command procedures. SYSPRV (see below) is adequate for interactive users, as it ultimately grants access to all files, while still providing access checks.

4.3.6 CMEXEC Privilege

The CMEXEC privilege allows the user's process to execute the Change Mode to Executive system service.

This system service lets a process change its access mode to executive, execute a specified routine, and then return to the access mode that was in effect before the system service was called. While in executive mode, the process is allowed to execute the Change Mode to Kernel system service.

The system manager should grant this privilege only to users who need to gain access to protected and sensitive data structures and internal functions of the operating system. If many users have unrestricted access to sensitive data structures and functions, the operating system and service to other users can easily be disrupted. Such disruptions can include failure of the system, destruction of the data base, and exposure of confidential information to unauthorized persons.

4.3.7 CMKRNL Privilege

The CMKRNL privilege allows the user's process to execute the Change Mode to Kernel system service.

This system service lets a process change its access mode to kernel, execute a specified routine, and then return to the access mode that was in effect before the system service was called.

The system manager should grant this privilege only to users who need to execute privileged instructions or who need to gain access to the most protected and sensitive data structures and functions of the operating system. If many users have unrestricted use of privileged instructions and unrestricted access to sensitive data structures and functions, the operating system and service to other users can easily be disrupted. Such disruptions can include failure of the system, destruction of the data base, and exposure of confidential information to unauthorized persons.

4.3.8 DETACH Privilege

The DETACH privilege allows the user's process to create detached processes by executing the Create Process system service. Detached processes remain in existence even after the user who created them has logged off the system.

An example of a detached process is the process created by the system for a user when the user logs in to the system.

There is no restriction on the UIC that can be specified for a detached process. Thus, there are no restrictions on the files and directories to which a detached process can gain access.

4.3.9 DIAGNOSE Privilege

The DIAGNOSE privilege allows the user to run online diagnostic programs and to intercept and copy all messages that are written to the error log file.

4.3.10 EXQUOTA Privilege

The EXQUOTA privilege allows the space taken by the user's files on given disk volumes to exceed any usage quotas set for the user (as determined by UIC) on those volumes.

4.3.11 GROUP Privilege

The GROUP privilege allows the user's process to affect other processes in its own group by executing the following process control system services: Suspend Process, Resume Process, Delete Process, Set Priority, Wake, Schedule Wakeup, Cancel Wakeup, and Force Exit. The user's process is also allowed to examine other processes in its own group by executing the Get Job/Process Information system service. The user with the GROUP privilege can issue SET QUEUE, DELETE/ENTRY, STOP/ENTRY, and SET PROCESS commands for other processes in its group.

The GROUP privilege is not needed for a process to exercise control over, or to examine, subprocesses that it created. The system manager should, however, grant this privilege to users who need to share data and whose processes need to cooperate.

4.3.12 GRPNAM Privilege

The GRPNAM privilege allows the user's process to insert names into the logical name table of the group to which the process belongs and to delete names from that table by the use of the following logical name system services: Create Logical Name and Delete Logical Name.

In addition, the privileged user can use the ASSIGN and DEFINE commands to add names to the group logical name table, the DEASSIGN command to delete names from the table, and the /GROUP qualifier of the MOUNT command to share volumes among group members.

This privilege should not be granted to all users of the system because it allows the user to create an unlimited number of group logical names. When many users have the unrestricted ability to create group logical names, excessive use of system dynamic memory can degrade system performance.

4.3.13 LOG_IO Privilege

The LOG_IO privilege allows the user's process to execute the Queue I/O Request system service to perform logical-level I/O operations.

Usually, user I/O requests are handled indirectly by use of an I/O package such as VAX-11 Record Management Services. However, to increase their control over I/O operations and to improve the efficiency of I/O operations, skilled users sometimes prefer to handle directly the interface between their process and a system I/O driver program. They can do this by executing the Queue I/O Request system service; in many instances, the operation called for is a logical-level I/O operation.

The system manager should grant this privilege only to users who need it. If this privilege is given to many users who have no need for it, the operating system and service to other users can easily be disrupted. Such disruptions can include the destruction of information on the system device, the destruction of user data, and the exposure of confidential information to unauthorized persons.

4.3.14 MOUNT Privilege

The MOUNT privilege allows the user's process to execute the mount volume QIO function. The use of this function should be restricted to system software supplied by DIGITAL.

4.3.15 NETMBX Privilege

The NETMBX privilege allows the user to perform functions related to a DECnet computer network.

4.3.16 OPER Privilege

The OPER privilege allows the user to use the Operator Communication Manager (OPCOM) process, as follows: to reply to users' requests, to broadcast messages to all terminals logged in, to designate terminals as operators' terminals and specify the types of messages to be displayed on these operators' terminals, and to initialize and control the log file of operators' messages. In addition, this privilege lets the user set devices spooled, create and control both batch queues and print queues, and initialize and mount public volumes.

The system manager should grant this privilege only to special users -- the operators of the system. These are the users who respond to the requests of ordinary users, who tend to the needs of the system's peripheral devices (mounting reels of tape and changing printer forms), and who attend to all the other day-to-day chores of system operation. (A non-privileged user can log in on the console terminal to respond to operator requests, for example, to mount a tape.)

4.3.17 PFNMAP Privilege

The PFNMAP privilege allows the user's process to map to specific pages of physical memory or I/O device registers, no matter who is using the pages or registers.

The system manager should exercise caution in granting this privilege. If many users have unrestricted access to physical memory, the operating system and service to other users can easily be disrupted. Such disruptions can include failure of the system, destruction of the data base, and exposure of confidential information to unauthorized persons.

4.3.18 PHY_IO Privilege

The PHY_IO privilege allows the user's process to execute the Queue I/O Request system service to perform physical-level I/O operations.

Usually, users' I/O requests are handled indirectly by use of an I/O package such as VAX-11 Record Management Services. However, to increase their control over I/O operations and to improve the efficiency of their applications, skilled users sometimes prefer to handle directly the interface between their process and a system I/O driver program. They can do this by executing the Queue I/O Request system service; in many instances, the operation called for is a physical-level I/O operation.

The system manager should grant the PHY_IO privilege only to users who need it; in fact, this privilege should be granted even more carefully than the LOG_IO privilege (see Section 4.3.13). If this privilege is given to many users who have no need for it, the operating system and service to other users can easily be disrupted. Such disruptions can include the destruction of information on the system device, the destruction of user data, and the exposure of confidential information to unauthorized persons.

4.3.19 PRMCEB Privilege

The PRMCEB privilege allows the user's process to create or delete a permanent common event flag cluster by executing the Associate Common Event Flag Cluster or Delete Common Event Flag Cluster system service.

Common event flag clusters enable cooperating processes to communicate with each other and thus provide the means of synchronizing their execution.

This privilege should not be granted to all users of the system because it allows the user to create an unlimited number of permanent common event flag clusters. A permanent cluster remains in the system even after the creating process has been terminated and continues to use up a portion of system dynamic memory. When many users have the unrestricted ability to create permanent common event flag clusters, the excessive use of system dynamic memory can degrade system performance.

4.3.20 PRMGBL Privilege

The PRMGBL privilege allows the user's process to create global sections by executing the Create and Map Section system service. In addition, the user with this privilege (plus the CMKRNL and SYSGBL privileges) can use the INSTALL utility.

Global sections are shared structures which can be mapped simultaneously in the virtual address space of many processes. All processes see the same code or data. Global sections are used for reentrant subroutines or data buffers.

The system manager should grant this privilege with care. If permanent global sections are not explicitly deleted, they tie up space in the global section and global page tables, which are limited resources.

4.3.21 PRMMBX Privilege

The PRMMBX privilege allows the user's process to create or delete a permanent mailbox by executing the Create Mailbox and Assign Channel system service or the Delete Mailbox system service.

Mailboxes are buffers in virtual memory that are treated as if they were record-oriented I/O devices. A mailbox is used for general interprocess communication.

The PRMMBX privilege should not be granted to all users of the system. Permanent mailboxes are not automatically deleted when the creating processes are deleted and, thus, continue to use up a portion of system dynamic memory.

4.3.22 PSWAPM Privilege

The PSWAPM privilege allows the user's process to control whether it can be swapped out of the balance set by executing the Set Process Swap Mode system service. Not only must a process have this privilege to lock itself in the balance set (that is, to disable swapping), but also to unlock itself (that is, to enable swapping).

With the same privilege, a process can create a process that is locked in the balance set (process swap mode disabled) by using an optional argument to the Create Process system service or, when the RUN command is used to create a process, by using a qualifier of the RUN command.

The system manager should grant this privilege only to users who need to lock a process in memory for performance reasons. Typically, this will be a real-time process. If many users have the unrestricted ability to lock processes in the balance set, physical memory can be held unnecessarily and thereby degrade system performance.

4.3.23 SETPRV Privilege

The SETPRV privilege allows the user's process to create processes whose privileges are greater than its own, by executing the Create Process system service with an optional argument or by issuing a RUN command to create a process. A user with this privilege can also execute the SET PROCESS /PRIVILEGES command to grant itself any desired privilege.

The system manager should exercise the same caution in granting the SETPRV privilege as in granting the CMEXEL or CMKRNL privilege.

4.3.24 SHMEM Privilege

The SHMEM privilege allows the user's process to create global sections and mailboxes (permanent and temporary) in multiport memory, if the process also has appropriate PRMGBL, PRMMBX, SYSGBL, and TPMBX privileges. Just as in local memory, the space required for a multiport memory temporary mailbox counts against the buffered I/O byte count limit (BYTLM) of the process.

4.3.25 SYSGBL Privilege

The SYSGBL privilege allows the user's process to create system global sections by executing the Create and Map Section system service. In addition, the user with this privilege (plus the CMKRNL and PRMGBL privileges) can use the INSTALL utility.

The system manager should exercise caution in granting this privilege. System global sections require space in the global section and global page tables, which are limited resources.

4.3.26 SYSNAM Privilege

The SYSNAM privilege allows the user's process to insert names into the system logical name table and to delete names from that table by using the following logical name system services: Create Logical Name and Delete Logical Name.

In addition, the user with this privilege can use the ASSIGN and DEFINE commands to add names to the system logical name table, and can use the DEASSIGN command to delete names from the table.

The system manager should grant this privilege only to the system operator or to system programmers who need to define system logical names (such as names for user devices, library directories, and the system directory). For example, to mount a system volume, which entails defining a system logical name, the system operator must have the SYSNAM privilege.

4.3.27 **SYSPRV Privilege**

The SYSPRV privilege allows the user to assume the file access rights of a system user, and to change the owner UIC and protection of a file. Even if a file is protected against system access, the user with the SYSPRV privilege can simply change its protection to gain access.

The system manager should exercise caution in granting this privilege. If many users have system access rights, the operating system and service to others can easily be disrupted. Such disruptions can include failure of the system, destruction of the data base, and exposure of confidential information to unauthorized persons.

4.3.28 **TMPMBX Privilege**

The TMPMBX privilege allows the user's process to create a temporary mailbox by executing the Create Mailbox and Assign Channel system service.

Mailboxes are buffers in virtual memory that are treated as if they were record-oriented I/O devices. A mailbox is used for general interprocess communication. Unlike a permanent mailbox, which must be explicitly deleted, a temporary mailbox is deleted automatically when no longer allocated by any process.

The system manager should usually grant this privilege to all users of the system to facilitate interprocess communication. System performance is not likely to be degraded by permitting the creation of temporary mailboxes, because their number is controlled by limits on the use of system dynamic memory.

4.3.29 **VOLPRO Privilege**

The VOLPRO privilege allows the user to (1) initialize a previously used volume with an owner UIC different from the user's own UIC; (2) override the expiration date on a non-owned tape or disk volume; (3) mount a non-owned Files-11 volume with the /FOREIGN qualifier; and (4) override the owner UIC protection of a volume. The VOLPRO privilege only permits control over volumes the user can mount or initialize. Volumes mounted with the /SYSTEM qualifier are safe from the user with the VOLPRO privilege as long as the user does not also have the SYSNAM privilege.

The system manager should exercise extreme caution in granting the VOLPRO privilege. If many users can override volume protection, the operating system and service to others can be disrupted. Such disruptions can include failure of the system, destruction of the data base, and exposure of confidential information to unauthorized persons.

4.3.30 **WORLD Privilege**

The WORLD privilege allows the user's process to affect other processes both inside and outside its group by executing the following process control system services: Suspend Process, Resume Process, Delete Process, Set Priority, Wake, Schedule Wakeup, Cancel Wakeup, and Force Exit. The user's process is also allowed to examine processes outside its own group by executing the Get Job/Process Information system service. The user with the WORLD privilege can issue SET QUEUE, DELETE /ENTRY, STOP /ENTRY, and SET PROCESS commands for all other processes.

To exercise control over subprocesses that it created or to examine these subprocesses, a process needs no special privilege. To affect or to examine other processes inside its own group, a process needs only the GROUP privilege. But to affect or examine processes outside its own group, a process needs the WORLD privilege.

4.4 ACCOUNTING FOR THE USE OF SYSTEM RESOURCES

For accounting purposes, the VAX/VMS system keeps records of the use of system resources. These records are kept in the accounting log file SYS$SYSDISK:[SYSMGR]ACCOUNTNG.DAT, which is updated each time an accountable process terminates, each time a print job is completed, and each time a login failure occurs. In addition, users can send messages to be inserted into the accounting log file.

Accounting records contain cumulative accounts of the resources used either by processes or subprocesses set up for users or by print symbionts that print out files for users. Each accounting record contains two fields -- user name and account name -- that identify the user and establish the connection between the accounting record and a user of the system. These fields correspond to similar fields of the user's account record in the user authorization file (UAF).

Using the detailed accounting records provided by the system, the system manager or a system programmer can devise programs for reporting on the use of system resources and for billing for their use.

Because the users of system resources are identified in two ways, reports on the use of system resources and bills for the use of system resources can be prepared in either of two ways: by user name or by account name.

The accounting log file is created and opened automatically when the operating system is initialized. Accounting records are arranged chronologically in this file. The following list summarizes the characteristics of the accounting log file:

- File name: ACCOUNTNG.DAT (this file is not an ASCII file; hence, it must be formatted before it is printed)
- Residence: the system device
- Directory: [SYSMGR]
- File organization: sequential
- Record length: variable length
- Record types: six

The six types of records correspond to the six conditions that cause records to be written to the file. These record types and their corresponding codes as defined in the macro $ACCDEF in SYS$LIBRARY:LIB.MLB (in parentheses) are as follows:

1. Records written when interactive processes terminate (ACC$K_INTTRM)

2. Records written when batch processes terminate (ACC$K_BATTRM)

3. Records written when subprocesses or detached processes terminate (ACC$K_PRCTRM)

4. Records written when print jobs are completed (ACC$K_PRTJOB)

5. Records written when login failures occur (ACC$K_LOGTRM)

6. Records written when users' messages are sent to the accounting log file (ACC$K_INSMSG)

For a detailed description of the records of the accounting log file, see the discussion of the Send Message to Accounting Manager ($SNDACC) system service in the VAX/VMS System Services Reference Manual.

To suppress the accounting function and thus avoid accounting for the use of system resources requires privilege. Only a user who has the ACNT privilege can create subprocesses or detached processes in which accounting is disabled. The /NOACCOUNTING qualifier of the RUN command is used to disable all accounting in a created process.

A user with OPER privilege can selectively disable various kinds of accounting system-wide by using the /DISABLE qualifier of the SET ACCOUNTING command. Usually, this is a task of the system operator. See the VAX/VMS Operator's Guide for a full description of the SET ACCOUNTING command.

As records are entered in the accounting log file, all but print job completion records are immediately flushed to disk. This precaution guarantees the integrity of the file and the completeness of accounting data even if the system fails.

Normally, the accounting log file is closed at the end of a billing period. The current version of the accounting log file is closed and a new version of the file is created and opened. As a rule, this is the system operator's job, done by use of the SET ACCOUNTING command.

If an attempt to write to the accounting log file results in an error, the file is closed automatically and a new copy is created and opened.

PART II

OPERATIONAL CONTROL

The system manager is responsible for the following aspects of operational control over the system:

- Initializing and mounting public volumes
- Enabling VAX-11 RMS file sharing
- Backing up public volumes
- Establishing usage quotas for individual users on public volumes
- Installing frequently used, shareable, and executable images as known images
- Modifying the site-independent start-up command procedure
- Creating a site-specific start-up command procedure
- Establishing spooled devices
- Controlling batch and print queues
- Logging errors and reporting software problems

Part II consists of Chapters 5 through 10:

- Chapter 5: Maintaining Public Files and Volumes -- Discusses procedures for handling public volumes and provides operating instructions for the RMSSHARE utility.
- Chapter 6: Disk Quotas -- Defines capabilities to limit disk usage and provides operating instructions for the DISKQUOTA utility.
- Chapter 7: Installing Known Images -- Discusses installation of selected executable and shareable images as known images, and the creation of permanent, system global sections for certain known images. Provides operating instructions for the INSTALL utility.
- Chapter 8: Start-up Command Procedures -- Defines the content of the site-independent start-up command procedure supplied by DIGITAL, and suggests the content for the site-dependent start-up command procedure.
- Chapter 9: Batch and Print Queues -- Discusses the use of spooled devices, and common procedures for creating (initializing), deleting, starting, and stopping device and batch queues.
- Chapter 10: Errors and Other System Events -- Discusses error logging procedures and provides operating instructions for the SYE utility. Outlines the software performance report.

CHAPTER 5

MAINTAINING PUBLIC FILES AND VOLUMES

Public volumes, also called system volumes, are file-structured disk volumes that contain public files. Public files are files that must be available to most, if not all, users. Public volumes can also contain files that users create for their own private use or for general use. Thus, as long as UIC-based file protection permits it, all users have access to public volumes and to the files on them.

Public volumes can contain the following kinds of public files supplied by DIGITAL:

- The operating system itself in executable form and files related to the operating system
- Utility programs in executable form
- Diagnostic and test programs in executable form and files related to these programs
- Various system libraries such as macro libraries, object module libraries, shared run-time libraries, and error message libraries
- Text files such as help files
- Optional software in executable form, plus related libraries and other files

In addition, the system manager can include on public volumes files that are unique to an installation. These typically are files that must be accessible to many if not all users of the installation.

Finally, the system manager can permit any user to create and store files on a public volume. Depending on their file protection, these files can be of general or restricted accessibility. This use of a public volume, however, is subject to limitation: a user is free to create, catalog, and store files on a public volume only if volume protection permits, if the user has write access to a directory on the volume, and if disk quotas permit. As a rule, the system manager creates a default disk file directory on a public volume for each user authorized to use the system (see Chapter 2).

Knowing how to initialize and mount public volumes is a prerequisite to managing a system of public files and volumes.

5.1 FILES-11 DISK STRUCTURE

The VAX/VMS system recognizes two disk file structures: Files-11 Structure Level 1 and Files-11 Structure Level 2. Files-11 Structure Level 2 is the default disk structure of the VAX/VMS system, and Files-11 Structure Level 1 is a structure used by DIGITAL's RSX-11M, RSX-11D, RSX-11M-PLUS, and IAS operating systems.

Nine files control the structure of a Files-11 Structure Level 2 volume. Only five of these files are used for a Files-11 Structure Level 1 volume. These files, which are referred to as reserved files, are as follows:

1. Index file (levels 1 and 2)
2. Storage bit map file (levels 1 and 2)
3. Bad block file (levels 1 and 2)
4. Master file directory (levels 1 and 2)
5. Core image file (levels 1 and 2)
6. Volume set list file (level 2 only)
7. Continuation file (level 2 only)
8. Back-up log file (level 2 only)
9. Pending bad block log file (level 2 only)

All of the files listed above are cataloged in the master file directory, [0,0].

5.1.1 Index File

Every Files-11 volume has an index file, which is created when the volume is initialized. This index file identifies the volume to the operating system as a Files-11 structure and contains the access data for all files on the volume. The index file, which is listed in the master file directory as INDEXF.SYS;1, contains the following information:

- Bootstrap block -- The volume's bootstrap block is virtual block number 1 of the index file. If the volume is a system volume, this block contains a bootstrap program that loads the operating system into memory. If the volume is not a system volume, this block contains a program that displays a message that the volume is not the system device but a device that contains users' files only.
- Home block -- The home block establishes the specific identity of the volume, providing such information as the volume name and protection, the maximum number of files allowed on the volume, and the volume ownership information. The home block is virtual block number 2 of the index file.
- Back-up home block -- The back-up home block is a copy of the home block. It permits the volume to be used even if the primary home block is destroyed.

- Back-up index file header -- The back-up index file header permits recovery of data on the volume if the index file header goes bad.
- Index file bit map -- The index file bit map controls the allocation of file headers and thus the number of files on the volume. The bit map contains a bit for each file header that is allowed on the volume. If the value of a bit for a given file header is 0, a file can be created with this file header. If the value is 1, the file header is already in use.
- File headers -- The largest part of the index file is made up of file headers. Each file on the volume has a file header, which describes such properties of the file as file ownership, creation date and time, file protection, and location of the file's blocks. The file header contains all the information needed for gaining access to the file.

5.1.2 Storage Bit Map File

The storage bit map file controls the available space on a volume; this file is listed in the master file directory as BITMAP.SYS;1. It contains a storage control block, which consists of summary information intended to optimize the Files-11 space allocation, and the bit map itself, which lists the availability of individual blocks.

5.1.3 Bad Block File

The bad block file, which is listed in the master file directory as BADBLK.SYS;1, contains all the bad blocks on the volume. The system detects bad disk blocks dynamically and prevents their reuse once the files to which they are allocated have been deleted.

5.1.4 Master File Directory

The master file directory (MFD) itself is listed in the MFD as 000000.DIR;1. The MFD, which is the root of the volume's directory structure, lists the reserved files that control the volume structure and may list both users' files and users' file directories. Usually, however, the MFD is used to list the reserved files and users' file directories; users seldom enter files in the MFD, even on private volumes. In fact, on a private volume, it is most convenient for a user to create a directory that has the same name as the user's default directory on a system disk. For an explanation of users' file directories and file specifications, see the VAX/VMS Command Language User's Guide.

5.1.5 Core Image File

The core image file is listed in the MFD as CORIMG.SYS;1. It is not used by VAX/VMS.

5.1.6 Volume Set List File

The volume set list file is listed in the MFD as VOLSET.SYS;1. This file is used only on relative volume 1 of a volume set. The file contains a list of the labels of all the volumes in the set.

5.1.7 Continuation File

The continuation file is listed in the MFD as CONTIN.SYS;1. This file is used as the extension file identifier when a file crosses from one volume to another volume of a loosely coupled volume set. This file is reserved for future use.

5.1.8 Back-Up Log File

The back-up log file is listed in the MFD as BACKUP.SYS;1. This file contains a history of the back-ups done to the volume. This file is reserved for future use.

5.1.9 Pending Bad Block Log File

The pending bad block log file is listed in the MFD as BADLOG.SYS;1. This file contains a list of suspected bad blocks on the volume that are not listed in the bad block file.

5.1.10 Files-11 Structure Level 1 Versus Structure Level 2

Files-11 Structure Level 2, a compatible superset of Structure Level 1, is the preferred disk structure on VAX/VMS for reasons of performance and reliability. At volume initialization time (see the INITIALIZE command in the VAX/VMS Command Language User's Guide), Structure Level 2 is the default. Structure Level 1 should be specified only for volumes that must be transportable to RSX-11M, RSX-11D, RSX-11M-PLUS, and IAS systems, as these systems support only that structure level. Additionally, the system manager may be required to handle Structure Level 1 volumes transported to VAX/VMS from one of the above systems.

Where Structure Level 1 volumes are in use on the system, the system manager should bear in mind the following limitations on them:

- Directories -- No hierarchies of directories and subdirectories, and no ordering of directory entries (that is, the file names) in any way (RSX-11M, RSX-11D, RSX-11M-PLUS, and IAS systems do not support subdirectories and alphabetical directory entries)
- Wild cards -- Wild card characters only for complete fields of file specifications (for example, SYS$SYSTEM:*USER.PAR is illegal, while SYS$SYSTEM:*.PAR is legal)
- Disk quotas -- Not supported
- Multi-volume files -- Not supported

- Placement control -- Not supported
- Caches -- No caching of file header blocks, file identification slots, or extent entries
- System disk -- Cannot be a Structure Level 1 volume
- Clustered allocation -- Not supported
- Back-up home block -- Not supported
- Protection code E -- Meaning of extend, rather than execute
- File versions -- Limited to 32,767, rather than being set by the /VERSION_LIMIT qualifier of CREATE /DIRECTORY, which defaults to approximately 60

Future enhancements to VAX/VMS will be based primarily on Structure Level 2, so that further restrictions on Structure Level 1 volumes may be incurred.

As explained in the VAX-11 Utilities Reference Manual, the system manager must use DSC1 to back up Structure Level 1 volumes and DSC2 to back up Structure Level 2 volumes, and VFY1 to verify Structure Level volumes and VFY2 to verify Structure Level 2 volumes.

5.2 INITIALIZING PUBLIC VOLUMES

The purpose of initializing a disk volume is to delete all old information from the volume and to impart to the volume a Files-11 structure that the operating system recognizes. This structure prepares a volume to receive data and permits the operating system to locate data stored on the volume.

The VAX/VMS Command Language User's Guide contains information about initializing volumes; see (1) the description of the INITIALIZE command in Part II and (2) Chapter 3, "Disk and Tape Volumes."

The following guidelines for initializing public volumes supplement information presented in the VAX/VMS Command Language User's Guide.

In initializing a public volume (by using the qualifier /SYSTEM), the system manager may need to use one or all of the following qualifiers of the DCL command INITIALIZE:

- /ACCESSED=n
- /CLUSTER_SIZE=n
- /EXTENSION=n
- /HEADERS=n
- /INDEX=position
- /MAXIMUM_FILES=n
- /WINDOW=n

As described below, selecting appropriate values for n and selecting the appropriate position for the /INDEX qualifier often involve making trade-offs.

5.2.1 /ACCESSED=n Qualifier

The /ACCESSED=n qualifier provides an estimate of the number of directories expected to be in use concurrently on a volume. The file system keeps this number of directory file control blocks in system space for ready access on the basis of which directories were most recently used. The result is a substantial reduction of overhead in directory operations. For volumes mounted /SYSTEM, the SYSGEN parameter ACP_SYSACC overrides this value.

5.2.2 /CLUSTER_SIZE=n Qualifier

The /CLUSTER_SIZE=n qualifier specifies the fundamental unit of allocation (expressed in blocks) on a volume. In selecting the cluster size, wasted space at the end of files is traded off against the size of the volume storage bit map, which must contain one bit for each cluster on the volume (or one block for each 4096 clusters).

5.2.3 /EXTENSION=n Qualifier

The /EXTENSION=n qualifier specifies the default number of blocks allocated for extending files on a volume. This value is less important on the VAX/VMS system than on the RSX-11M, RSX-11D, RSX-11M-PLUS, and IAS systems, because VAX-11 Record Management Services use an adaptive algorithm maximized against /EXTENSION. The value of this qualifier should be an even multiple of /CLUSTER_SIZE.

5.2.4 /HEADERS=n Qualifier

The /HEADERS=n qualifier specifies the number of file headers to be allocated initially to the index file. The primary advantage of preallocating file headers is that they will then be located near the storage map file (usually in the middle of the disk). This placement of file headers helps reduce head motion during file manipulation. This value should be estimated conservatively, because space allocated to headers cannot later be made available for file storage.

5.2.5 /INDEX=position Qualifier

The /INDEX=position qualifier specifies the location of the index file on a volume. The default position (MIDDLE) results in minimum head motion during file processing. The position BEGINNING should be used if the disk is to contain only one or a few very large contiguous files. (The Disk Save and Compress (DSC) Utility positions the index at BEGINNING.)

5.2.6 /MAXIMUM_FILES=n Qualifier

The /MAXIMUM_FILES=n qualifier specifies the maximum number of files that a volume can contain. The default value is fairly liberal. A closer estimate of it helps optimize the dynamic allocation of the index file; once set, however, the maximum number of files for a volume cannot be increased. Note that each directory and each extension header of a multiheader file counts as a file against this maximum value.

5.2.7 /WINDOW=n Qualifier

The /WINDOW=n qualifier specifies the number of map pointers in a default file access window. This value is the number of extents of a file to which access can be gained without the cost of file system overhead.

5.3 MOUNTING PUBLIC VOLUMES

The purpose of mounting a volume or volume set is to establish a relationship between the volume or volume set, the device(s) on which the volume is physically mounted, and one or more processes that can gain access to the volume.

The VAX/VMS Command Language User's Guide contains information about mounting volumes; see (1) the description of the MOUNT command in Part II and (2) Chapter 3, "Disk and Tape Volumes."

The following guidelines for mounting disk volumes for public use supplement information presented in the VAX/VMS Command Language User's Guide.

In mounting a public volume (by using the qualifier /SYSTEM), the system manager may need to use one or all of the qualifiers /ACCESSED, /EXTENSION, and /WINDOW (described in Section 5.2), or the qualifiers /BIND=volume-set-name and /PROCESSOR=option (described below).

5.3.1 /BIND=volume-set-name Qualifier

The /BIND=volume-set-name qualifier combines two or more volumes into a volume set, or adds one or more volumes to an existing volume set. A volume set makes two or more disk volumes appear to the user as one volume on one device.

5.3.2 /PROCESSOR=option Qualifier

The /PROCESSOR=option qualifier specifies the number of ancillary control processes (ACPs) to be used in controlling various public volumes. Selecting an appropriate option for the /PROCESSOR qualifier involves making a trade-off. If the system manager specifies the option SAME, file system parallelism and performance may be sacrificed for the sake of saving system space. Conversely, if the system manager specifies the option UNIQUE, system space is sacrificed for the sake of file system parallelism and performance.

RMSSHARE

5.4 MAINTAINING VOLUME INTEGRITY

To enhance performance, the system caches in memory information concerning a volume's free space, file identifications, quota file entries, and file headers. The system manager determines the degree of caching with the ACP cache system parameters (see Chapter 12) and individual users can alter cache sizes on their volumes with qualifiers to the MOUNT command (see the VAX/VMS Command Language User's Guide). The system writes the information in the caches to the disk when the disk is dismounted or the system is shut down. Naturally, removal of a disk before the caches are written back loses any changes made to the information in the caches. Therefore, the system manager and individual user should:

- Not write-lock a volume while it is mounted
- Not remove a volume from a drive until it has been dismounted
- Not halt the system without performing the orderly shut-down procedure (see the VAX/VMS Operator's Guide)

If the user write-locks a volume at mount time, the system additionally applies a software write lock: the user must dismount and remount the volume to write-enable it.

At mount time, if the system detects that the caches were not written back the last time the volume was used, the system automatically rebuilds the file information by scanning the contents of the volume. However, file headers for open files may be partially or wholly lost (see Section 11.5.1).

5.5 RMSSHARE UTILITY

The RMSSHARE utility performs the following functions:

- Enables the VAX-11 Record Management Services (VAX-11 RMS) file sharing capability by initializing file sharing structures in paged dynamic memory (system S0 space), and sets the maximum number of pages that the structures can occupy. The system imposes an absolute maximum of one-half the total space in paged dynamic memory. The VAX-11 RMS file sharing capability must be enabled each time the operating system is booted.
- If VAX-11 RMS file sharing has already been enabled, displays figures on allowable and actual usage, and permits the resetting of the maximum number of pages that the file sharing structures can occupy.

The system manager, or any user with the CMEXEC privilege, invokes the utility with the following command:

```
$ RUN SYS$SYSTEM:RMSSHARE
```

RMSSHARE (Cont.)

If VAX-11 RMS file sharing is not enabled, RMSSHARE displays the following message:

```
RMS file sharing is not currently enabled...
  Maximum allocation allowed: n
```

If VAX-11 RMS file sharing is already in effect, RMSSHARE displays the following message:

```
RMS file sharing is currently enabled...
  Maximum allocation allowed: n
  Number of pages allocated: n
  Max pages used: n
  Current number of pages in use: n
```

In either case, RMSSHARE then prompts for a maximum page count:

```
Enter max pages:
```

The system manager can either:

- Enter the maximum number of pages that file sharing structures can occupy as a positive integer not less than the current number of pages in use and not greater than the maximum number of pages allowed

- Enter the word EXIT (or simply press the return key) to terminate the utility

RMSSHARE continues to display and prompt as shown above until terminated.

If an invalid value is entered for the maximum number of pages that file structures can occupy, the following message appears:

```
Invalid size parameter, set to maximum value:  n
```

RMSSHARE allocates the maximum allowable number of pages in lieu of the invalid value.

The system manager should use the following guidelines to estimate the maximum number of pages required for file sharing structures:

- System base requirement -- 2 pages

- Per sequential file being shared -- 1 page for the first three sharers and 1 page for each additional four sharers

- Per relative file being shared -- 1 page for the first three sharers and 1 page for each additional four sharers

- Per indexed file being shared -- 1 page for the first two sharers and 1 page for each additional two sharers

The system manager totals the above pages.

RMSSHARE (Cont.)

If more than one locked record per sharer is anticipated, the system manager should add 16 bytes for each additional locked record. If the multibuffer count exceeds 1 for a sequential or relative file, or 2 for an indexed file, the system manager should add 36 bytes for each additional buffer. The grand total is then rounded up to the next page.

The site-independent start-up command procedure (see Chapter 8) runs RMSSHARE with a specification of 20 for the maximum number of pages for file sharing structures. This number should be adjusted according to the above guidelines. The system manager should also adjust the size of the paged dynamic pool (PAGEDYN system parameter; see Chapter 12).

5.6 BACKING UP PUBLIC VOLUMES

To prevent the inadvertent loss or destruction of valuable information stored on public volumes, the system manager usually establishes a policy and a schedule for regularly backing up files on public volumes. Once such a policy is established, the system operator usually is responsible for putting it into effect. The VAX/VMS Operator's Guide therefore provides the operating procedures for backing up both selected files and entire volumes.

Just as preserving information on public volumes by backing it up is usually considered desirable, preserving files on private volumes is also considered desirable. However, responsibility for backing up the files on private volumes usually is left to the individual owners of those files and volumes.

There are two kinds of back-ups of public disk files and volumes: (1) selective, or partial, back-ups and (2) system, or all-inclusive, back-ups. The back-up medium, in either case, can be disk or tape.

Selective back-ups of files chosen by users of the system can be accomplished with the VAX-11 RMS utility BACKUP. System back-ups, on the other hand, are usually accomplished with one of the Disk Save and Compress (DSC) utility programs (DSC1 or DSC2), or with the COPY command.

As explained in the VAX/VMS Operator's Guide and in the description of the Disk Save and Compress utility in the VAX-11 Utilities Reference Manual, the difference between the utilities is in the level of the Files-11 disk file structure that they write to a new disk. DSC1 (which writes disks with Files-11 Structure Level 1) is used in backing up disks that have been initialized with Files-11 Structure Level 1; and DSC2 (which writes disks with Files-11 Structure Level 2) is used in backing up disks that have been initialized with Files-11 Structure Level 2.

As a rule, selective back-ups are done more frequently than system back-ups. Normally, the system manager, after consulting with users of the system, decides how frequently to back up files and volumes and how long to retain back-up files and volumes.

The following schedule for backing up public disk volumes on magnetic tape affords adequate protection of data for many installations:

- Daily -- A selective back-up retained for seven days. This schedule requires seven daily tapes that are rotated once a week.
- Weekly -- An all-inclusive back-up retained for four weeks. This schedule requires four weekly tapes that are rotated once every four weeks.
- Monthly -- An all-inclusive back-up retained for a year. This schedule requires twelve monthly tapes that are rotated once a year.

Despite all precautions, there is always the risk of losing a file. Longer retention periods reduce this risk.

CHAPTER 6

DISK QUOTAS

The system manager limits the amount of space available to individual users on public volumes (or volume sets) by creating and maintaining quota files on those volumes. Individual users can similarly restrict usage on private volumes. Quotas are maintained and enforced on a per volume basis. Each volume or volume set has its own quota file; a volume on which quotas are not maintained has no quota file; on a volume set, volume 1 contains the quota file. Each entry in a quota file includes the following information:

- UIC -- UIC of a user entitled to maintain files on the volume

- Usage - Number of blocks on the volume taken up by the user's files

- Quota -- Maximum number of blocks on the volume that the user's files may take up before an error message is issued

- Overdraft -- Number of blocks over the quota that the user's files may take up

The absolute maximum number of blocks permitted a user on a volume is the sum of the quota and the overdraft.

The system manager (or user maintaining the volume) identifies UICs and assigns quotas and overdrafts with the DISKQUOTA utility (Section 6.1). Usage counts are maintained automatically by the system during normal file activities (Section 6.2).

The name of the quota file is [0,0]QUOTA.SYS on the applicable volume.

A quota file is initialized with an entry for UIC [0,0]. The usage count for this UIC should not change from 0 -- the UIC should own no files. Its quota and overdraft, however, serve as defaults in certain situations, and so should be set to values most likely to be assigned as quotas and overdrafts to other UICs.

A quota file requires one block of secondary storage for each 16 entries.

6.1 DISKQUOTA UTILITY

The system manager (or any user maintaining a volume) can run the DISKQUOTA utility to control the usage of disk volumes. Table 6-1 summarizes the DISKQUOTA commands by format and function.

DISKQUOTA

Table 6-1
DISKQUOTA Command Summary

Format	Function
ADD uic [/PERMQUOTA=quota] [/OVERDRAFT=quota-plus]	Adds an entry to the quota file
CREATE	Creates a quota file for a volume that does not currently contain one
DISABLE	Suspends quota operations on a volume
ENABLE	Resumes quota operations on a volume
EXIT	Returns the user to DCL command level
HELP	Lists the DISKQUOTA commands
MODIFY uic [/PERMQUOTA=quota] [/OVERDRAFT=quota-plus]	Changes an entry in the quota file
REBUILD	Reconstructs the usage counts for all entries
REMOVE uic	Deletes an entry from the quota file
SHOW uic	Displays quotas and usage counts
USE device	Specifies the volume to be acted upon

The sequence of commands for creating a new quota file typically includes:

1. USE -- To name the volume (can be omitted if the volume is the user's default device)

2. CREATE -- To create the quota file

3. MODIFY -- To set the quota and overdraft for UIC [0,0]

4. REBUILD -- To add entries for existing files (can be omitted on an empty volume)

5. ADD -- To add entries (one ADD command per entry) for UICs not automatically added during Step 4

The sequence of commands for modifying an existing quota file typically includes (1) a USE command followed by (2) an ADD, MODIFY, or REMOVE command for each entry being changed.

DISKQUOTA (Cont.)

6.1.1 Invoking DISKQUOTA

The following command invokes the utility:

```
$ RUN SYS$SYSTEM:DISKQUOTA
```

The system responds with the following prompt:

```
DISKQ>
```

The user can then enter any of the commands listed in Table 6-1. These commands follow the standard rules of grammar as specified in the VAX/VMS Command Language User's Guide.

6.1.2 ADD Command

ADD adds an entry to the quota file. It takes the form:

```
ADD uic [/PERMQUOTA=quota]
        [/OVERDRAFT=quota-plus]
```

uic
: A valid UIC

quota
: A positive integer that specifies a quota for the specified UIC; defaults to the value of quota in the entry for [0,0]

quota-plus
: A positive integer that specifies an overdraft for the specified UIC; defaults to the value of quota-plus in the entry for [0,0]

The usage count for a new entry is initialized to 0.

In the following example, the user sets the quota for UIC [300,211] to 200 and the overdraft to 50 (for an absolute limit of 250 blocks):

```
DISKQ>ADD [300,211] /PERMQUOTA=200 /OVERDRAFT=50
```

The ADD command requires write access to the quota file.

6.1.3 CREATE Command

CREATE creates a new quota file. It takes the form:

```
CREATE
```

A quota file must not already be present on the volume being used. The CREATE command should be immediately followed by a MODIFY command for UIC [0,0] with /PERMQUOTA and /OVERDRAFT set to reasonable default values.

The CREATE command requires write access to the volume's MFD; and either the SYSPRV privilege, a system UIC, or ownership of the volume.

DISKQUOTA (Cont.)

6.1.4 DISABLE Command

DISABLE suspends the maintenance and enforcement of quotas on a volume. It takes the form:

```
DISABLE
```

The DISABLE command requires the SYSPRV privilege, a system UIC, or ownership of the volume.

6.1.5 ENABLE Command

ENABLE resumes quota operations on a volume. It takes the form:

```
ENABLE
```

The ENABLE command requires the SYSPRV privilege, a system UIC, or ownership of the volume.

6.1.6 EXIT Command

EXIT returns the user to DCL command level. It takes the form:

```
EXIT
```

The system manager can also return to DCL command level by typing <CTRL/Z>.

6.1.7 HELP Command

HELP lists and explains the DISKQUOTA commands. It takes the form:

```
HELP [command [qualifier]]
```

command
: Name of a DISKQUOTA command

qualifier
: Name of a DISKQUOTA command qualifier

6.1.8 MODIFY Command

MODIFY changes an entry in the quota file. It takes the form:

```
MODIFY uic [/PERMQUOTA=quota]
           [/OVERDRAFT=quota-plus]
```

uic
: A UIC with an entry on the quota file

quota
: A positive integer that specifies a quota for the specified UIC

quota-plus
A positive integer that specifies an overdraft for the specified UIC

If quota is less than the current usage count, a warning message is issued. (The new quota does take effect, however.)

In the following example, the user sets the permanent quota for UIC [300,211] to 300 blocks, while making no change to the overdraft:

```
DISKQ>MODIFY [300,211] /PERMQUOTA=300
```

The MODIFY command requires write access to the quota file.

6.1.9 REBUILD Command

REBUILD reconstructs the usage counts for all entries on the volume. It takes the form:

```
REBUILD
```

The REBUILD command automatically adds entries for files owned by UICs with no entries in the quota file, setting their quotas and overdrafts to the values of the defaults in UIC [0,0]. During the time that the REBUILD command is executing, file activity on the volume is frozen -- no files can be created, deleted, extended, or truncated. For this reason, the command should be used judiciously, normally in the following situations:

- Established files -- When a quota file is created on a volume with existing files, the REBUILD command should be run before the ADD commands. REBUILD adds entries and records the existing usage for all UICs with detected usage.
- Usage not updated -- If usage counts were not updated for some period due to suppression or suspension, REBUILD should be run to correct the usage.

The REBUILD command requires write access to the quota file; and either the SYSPRV privilege, a system UIC, or ownership of the volume.

6.1.10 REMOVE Command

REMOVE deletes an entry from the quota file. It takes the form:

```
REMOVE uic
```

uic
A UIC with an entry in the quota file

If the usage count for the UIC is not 0, a warning message is issued. (The UIC is removed, however.) UIC [0,0] cannot be removed.

The following example deletes UIC [300,211] from the quota file:

```
DISKQ>REMOVE [300,211]
```

The REMOVE command requires write access to the quota file.

DISKQUOTA (Cont.)

6.1.11 SHOW Command

SHOW displays quotas, overdrafts, and usage counts. It takes the form:

SHOW uic

uic
A UIC with an entry in the quota file

A wild card character (*) can be used in specifying the UIC, as illustrated in the following examples:

Command	Description
SHOW [300,211]	Show user 211 in group 300
SHOW [300,*]	Show all users in group 300
SHOW [*,211]	Show all users with a member number of 211
SHOW [*,*]	Show all users

The SHOW command requires no privileges to show one's own quota, overdraft, and usage count, but otherwise requires read access to the quota file.

6.1.12 USE Command

USE specifies the volume to be acted upon. It takes the form:

USE device

device
A physical device name or a logical name equated to a physical device name

If USE is not specified, DISKQUOTA uses the user's default device. USE can be specified more than once during a session to work with quota files on more than one volume.

Any volume in a volume set can be specified. The volume actually operated on, however, is relative volume 1.

The following examples specify the volume to be acted upon as (1) a volume on a physical device and (2) a volume on the physical device equated to a logical name:

1. DISKQ>USE DMA2:
2. DISKQ>USE X2_RESEARCH_DATA

DISKQUOTA (Cont.)

6.1.13 **Error Messages**

Table 6-2 lists the error messages issued by the DISKQUOTA utility (in alphabetical order).

Table 6-2
DISKQUOTA Error Messages

Message	Remedy
ambiguous command	Spell out the command name more completely.
ambiguous qualifier	Spell out the qualifier name more completely.
cannot allocate sufficient memory	Increase the VIRTUALPAGCNT system parameter (with the SYSGEN utility) so that the REBUILD command has sufficient space to build the usage table.
cannot examine quota file entry	Respond to VAX-11 RMS message that follows.
command syntax error	Format the command according to the specifications of Table 6-1.
device is not a local device	Enter a device name (or logical name that translates to a device name) that does not contain a node name. DISKQUOTA operates only on local devices.
error closing quota file	Respond to VAX-11 RMS message that follows.
error creating quota file	Respond to VAX-11 RMS message that follows.
error initializing quota file	Respond to VAX-11 RMS message that follows.
failed help library index init	Respond to VAX-11 RMS message that follows.
failed to access index file on relative volume n	Respond to VAX-11 RMS message that follows.
failed to access help text	Respond to VAX-11 RMS message that follows.

(continued on next page)

DISKQUOTA (Cont.)

Table 6-2 (Cont.)
DISKQUOTA Error Messages

Message	Remedy
failed to access quota file	Respond to VAX-11 RMS message that follows.
failed to add quota file entry	Respond to VAX-11 RMS message that follows.
failed to disable quota file	Respond to VAX-11 RMS message that follows.
failed to enable quota file	Respond to VAX-11 RMS message that follows
failed to lock volume	Respond to VAX-11 RMS message that follows.
failed to open help library	Respond to VAX-11 RMS message that follows.
failed to read home block on relative volume n	Respond to VAX-11 RMS message that follows.
failed to remove quota file entry	Respond to VAX-11 RMS message that follows.
failed to unlock volume	Respond to VAX-11 RMS message that follows.
invalid UIC	Enter a correctly formatted UIC.
I/O error reading commands	Respond to VAX-11 RMS message that follows.
I/O error reading file header k on relative volume n	Respond to VAX-11 RMS message that follows. The value of k is the file number of the file in error.
I/O error reading index file bitmap on relative volume n	Respond to VAX-11 RMS message that follows.
I/O error reading quota file	Respond to VAX-11 RMS message that follows.
logical name is recursively defined	Enter the actual name or a logical name that can be translated within 10 iterations.

(continued on next page)

DISKQUOTA (Cont.)

Table 6-2 (Cont.)
DISKQUOTA Error Messages

Message	Remedy
no device currently selected	Enter another USE command specifying a valid device name.
uic has n blocks in use	Do not respond -- informational message. The specified UIC is over quota due to the MODIFY or DELETE operation just performed.
unrecognized command	Enter a valid command.
unrecognized qualifier	Enter a valid qualifier.
volume set has too many volumes to handle	Reduce the number of volumes in the volume set to 255 or less, or do not use disk quotas.

Where correction of an entered value is indicated, the entire command must be reentered. For explanations of the VAX-11 RMS messages, see the VAX/VMS System Messages and Recovery Procedures Manual.

6.2 OPERATIONS

During normal use of a volume with a quota file, the system automatically updates the usage counts as users create, delete, extend, and truncate files. Users without entries in the quota file are not allowed to create files or allocate space on the volume, unless they have the EXQUOTA privilege.

6.2.1 Exceeding the Quota

If an operation to add a new file or expand a current file will put a user's usage count over the quota, the system prohibits the operation and issues the following message:

    disk quota exceeded

If the rejected operation is an extension of a file opened for write, a user with an overdraft can perform the operation by retrying it. Operations to extend the file will succeed until usage exceeds the sum of the quota and the overdraft. At this point, the system reissues the above message and prohibits further extensions to the file.

To create new files, a user's usage must be below quota (not overdraft).

Quota restrictions are not enforced for users with the EXQUOTA privilege. However, their usage counts are maintained.

6.2.2 Suspending Quotas

The DISABLE command in DISKQUOTA (Section 6.1.4) suspends quota operations on a volume. The ENABLE command (Section 6.1.5) lifts the suspension. In addition, quota operations on a volume can be suspended at mount time by specifying the /NOQUOTA qualifier in the DCL command MOUNT.

To discontinue quota operations on a volume, the user executes the DISABLE command, exits from DISKQUOTA, and deletes the QUOTA.SYS file.

6.2.3 REBUILD on Mount

When a volume is mounted with quota operations enabled and that volume was not properly dismounted the last time it was used, the system performs an automatic REBUILD operation. This action ensures that the quota file accurately reflects usage of the disk in the event that quota operations were suspended, the system failed, the volume was physically removed before being dismounted, or the WRITE PROTECT button was pushed.

6.2.4 Restrictions on Other System Operations

The following restrictions must be observed whether or not disk quotas are being used:

- File ownership -- Because a change in file ownership consumes the resources of another user, changing the owner UIC of a file requires the SYSPRV privilege.

- Volume sets -- Relative volume 1 must be online at all times.

CHAPTER 7

INSTALLING KNOWN IMAGES

The system manager enhances the performance of selected executable and shareable images by installing them as known images with the INSTALL utility. Known images can be assigned the following attributes:

- Permanently open -- Directory information on the image file remains permanently resident, eliminating the usual directory search required to locate a file. The cost of keeping an image file permanently open is 192 bytes of nonpaged dynamic memory.

- Header resident -- The header of the image file (native images only) remains permanently resident, saving one disk I/O operation per file access, at a cost of less than one page of paged dynamic memory. The image must also be declared permanently open.

- Privileged -- Amplified privileges are temporarily assigned to any process running the image (executable images only), permitting the process to bypass its UAF privilege restrictions during execution of the image. In this way, "normal" users can run programs that require higher-than-normal privileges.

- Protected -- A shareable image contains protected code, that is, code that runs in kernel or executive mode but that can be called by a user-level image.

- Shared -- More than one user can access the read-only and non-copy-on-reference (non-CRF) read/write sections of the image concurrently, so that only one copy of those sections ever need be in physical memory. (CRF sections always require a separate copy for each process.) The image must also be declared permanently open.

- Writeable -- When a shared non-CRF writeable section is removed from physical memory (for paging reasons or because no processes are referencing it), it is written back to the image file. Any updates made by using processes, therefore, are preserved (while the initial values are lost). The image must also be declared shared.

7.1 EXECUTABLE AND SHAREABLE IMAGES

Most images installed as known images are likely to be executable images. An executable image is one linked with the /EXECUTABLE qualifier.

The system manager can also install shareable images as known images. A shareable image is one linked with the /SHAREABLE qualifier. A shareable image must subsequently be linked into an executable image to be used.

Shareable images must not be confused with known images installed with the shared qualifier:

- Shareable images -- A shareable image is not copied into the executable images that link with it (as long as the options file does not specify /SHAREABLE=COPY or GSMATCH=NEVER). Thus, only one copy of the shareable image need be on disk, no matter how many executable images have linked with it.

- Shared images -- The shared attribute can be assigned to, or withheld from, any known image -- shareable or executable. Its assignment results in the creation of permanent, system global sections. Execution of non-CRF global sections requires only one copy per section to be in physical memory, no matter how many processes are running the image to which the sections belong. Global sections are created for CRF sections, but the sections are not shared in memory.

When an image is not installed, or is installed without the shared attribute, each process running the image requires private sections in memory. (As of Version 2.0, a shareable image linked to an executable image without specification of the /SHAREABLE=COPY qualifier or GSMATCH=NEVER in the options file need not be installed to be executed. At image execution time, the system will create private sections from the shareable image. The only exception is that a shareable image containing a writeable non-CRF section must be installed as a known image with the shared and writeable attributes.)

The number of images that can be installed with the shared qualifier is restricted by the GBLPAGES and GBLSECTIONS system parameters (see Chapter 12).

7.2 KNOWN FILE LISTS

The system defines known images on internal data structures called known file lists. Each known file list contains entries for all known images whose device, directory, and file type are identical. For example, all known images with the file name DBA1:[MAIN]file-name.EXE would be on one known file list, while all known images with the file name DBA1:[TEST]file-name.EXE would be on another known file list.

The number of known file lists is restricted by the KFILSTCNT system generation parameter (see Chapter 12).

7.3 INSTALL UTILITY

The system manager runs the INSTALL utility to install and maintain known images. Table 7-1 summarizes the INSTALL commands.

Table 7-1
INSTALL Command Summary

Format	Function
file-spec [/OPEN] [/HEADER_RESIDENT] [/PRIVILEGED[=(priv-name,...)]] [/PROTECTED] [/SHARED] [/WRITEABLE]	Installs an image as a known image with qualifiers as specified
file-spec /DELETE	Deletes an image as a known image
file-spec /REPLACE	Associates a known image with the latest version of the image file
[file-spec] /LIST [file-spec] /FULL /GLOBAL	Displays descriptions of known images and global sections

7.3.1 Invoking INSTALL

The following command invokes the utility:

```
$ RUN SYS$SYSTEM:INSTALL
```

The system responds with the following prompt:

```
INSTALL>
```

The system manager can then enter any of the commands listed in Table 7-1. These commands follow the standard rules of grammar as specified in the VAX/VMS Command Language User's Guide.

The file specification must name an existing executable or shareable image. Defaults are applied as follows:

- Device and directory -- SYS$SYSTEM
- File type -- EXE

INSTALL (Cont.)

A version number must not be specified. The highest existing version is always used. Specification of a version number produces unpredictable results (because file look-ups for known images ignore version numbers).

Use of the INSTALL utility requires the CMKRNL, PRMGBL, and SYSGBL privileges.

7.3.2 Installing Known Images

The system manager installs a known image with the following command:

```
file-spec [/OPEN]
          [/HEADER_RESIDENT]
          [/PRIVILEGED[=(priv-name,...)]]
          [/PROTECTED]
          [/SHARED]
          [/WRITEABLE]
```

file-spec
: File specification of the image being installed

/OPEN
: For installation of a permanently open known image

/HEADER_RESIDENT
: For installation of a known image with a permanently resident header (native mode images only); the image is made permanently open even if /OPEN is not specified

/PRIVILEGED[=(priv-name,...)]
: For installation of a known image with privileges (executable images only); the image is made permanently open even if /OPEN is not specified; privilege names are specified as shown in Table 7-2 (Section 7.3.6); if no privilege names are specified, all privileges are assigned; parentheses can be omitted for specification of one privilege

/PROTECTED
: For installation of a known image with protected code

/SHARED
: For installation of a shared known image; causes creation of global sections for the image; the image is made permanently open even if /OPEN is not specified

/WRITEABLE
: For installation of a writeable known image; if /SHARED is not also specified, this qualifier is ignored

The following example installs a permanently open, shared known image:

```
INSTALL> DBA1:[MAIN]STATSHR /OPEN /SHARED
```

INSTALL (Cont.)

The next example installs a permanently open known image with privileges:

```
INSTALL> GRPCOMM /OPEN /PRIVILEGES=(GROUP,GRPNAM)
```

Any process running GRPCOMM receives the GROUP and GRPNAM privileges for the duration of the execution of GRPCOMM. The full name of GRPCOMM is assumed to be SYS$SYSTEM:GRPCOMM.EXE.

7.3.3 Deleting Known Images

The system manager deletes a known image with the following command:

```
file-spec /DELETE
```

file-spec
    File specification of an image installed as a known image

The image's entry on the known file list and any global sections created for the image are deleted. The image itself (that is, the image file) remains unaffected. Writeable non-CRF global sections are written back to disk upon deletion of their known images.

The following example deletes a known image:

```
INSTALL> DBA1:[MAIN]STATSHR /DELETE
```

7.3.4 Replacing Known Images

The system manager replaces a known image with the following command:

```
file-spec /REPLACE
```

file-spec
    File specification of an image installed as a known image

The image's entry on the known file list becomes associated with the latest version of the image file.

The replace function cannot be used to assign additional attributes to a known image. If an image is installed without a resident header, for example, a specification of /REPLACE /HEADER_RESIDENT does not add this attribute. The system manager must delete the known image and reinstall it with the desired attributes.

The following example replaces a known image:

```
INSTALL> GRPCOMM /REPLACE
```

The full name of the file specification is assumed to be SYS$SYSTEM:GRPCOMM.EXE.

INSTALL (Cont.)

7.3.5 Listing Known Images

The system manager displays information on known images and global sections with the following commands:

- [file-spec] /LIST
- [file-spec] /FULL
- /GLOBAL

file-spec
: File specification of an image installed as a known image

/LIST
: For display of a 1-line description of the specified known image or (if no file specification is made) all known images; see Figure 7-1

```
$ RUN SYS$SYSTEM:INSTALL
INSTALL>/LIST
                    ❶           ❷    ❸          ❺             ❽
DBA0:[SYSEXE]COPY.EXE;1      Shar Head      Open          Shm
DBA0:[SYSEXE]BLISS32.EXE;8   Shar Head  ❹   Open          Shm
DBA0:[SYSEXE]SET.EXE;2       Shar Head Priv Open          Shm
                               .
                               .
                               .                 ❻
DBA0:[SYSEXE]SOS.EXE;1       Shar           Open Compat
DBA0:[SYSEXE]PIP.EXE;1                      Open Compat
                               .
                               .
                               .                       ❼       ❾
DBA1:[MAIN]STATSHR.EXE;3     Shar           Open       Pro     Noexe
```

❶ File specification of known image

❷ Known image is shared

❸ Known image has permanently resident header

❹ Known image has privileges

❺ Known image is permanently open

❻ Known image runs in compatibility mode

❼ Known image contains protected code

❽ Known image is installed in multiport memory

❾ Known image is not executable -- that is, it is a shareable image

Figure 7-1 Known Image Display -- Brief Description

INSTALL (Cont.)

/FULL
For display of a multiline description of the specified known image or (if no file specification is made) all known images; see Figure 7-2

/GLOBAL
For display of all global sections (whether or not the section was created as a result of INSTALL); see Figure 7-3

```
$     SYS$SYSTEM:INSTALL
INSTALL>USERS /FULL
DBA0:[SYSEXE]USERS.EXE;1         Head Priv Open ❶
    # of times run   = 30 ❷
    Privilege bits   = 00010100,00000000 ❸
    Entry adr/size   = 8006EBF0/64  ❹
    Window adr/size  = 800A17A0/48  ❺
    Header adr/size  = 8006A8FC/368 ❻
```

❶ Same as for brief description (see Figure 7-1)

❷ Number of times image has been run

❸ Privileges allotted to known image; see Table 7-2 for translation; appears only if image installed with privileges

❹ Address (in hexadecimal) of image's entry on known file list and the size of the entry in bytes

❺ Address (in hexadecimal) of image's directory window and the size of the window in bytes; appears only if image installed permanently open

❻ Address (in hexadecimal) of image's resident header and the size of the header in bytes; appears only if image installed with permanently open resident header

❼ Not shown in example - if image runs in compatibility mode, second line reads: Compatibility type = 0000

Figure 7-2 Known Image Display -- Full Description

INSTALL (Cont.)

```
$ RUN SYS$SYSTEM:INSTALL
INSTALL>/GLOBAL
                                     ❶
                       System Global Sections
   ❷            ❸   ❹   ❺        ❻   ❼               ❽
LBRSHR_004    (01000001) WRT CRF      PRM SYS     Pagcnt/Refcnt=1/0
CRFSHR_003    (010003E8) WRT CRF      PRM SYS     Pagcnt/Refcnt=1/1
                         .
                         .
                         .        ❾
27 Global Sections Used, 828 Global Pages Used, 3268 Global Pages Unused
                                   ❿
              Global Sections in Multiport Memory "SHM"
                                                          ⓫
VMSRTL_001      (010003FC)             PRM SYS    Creator Port=0
        Basepfn/Pagcnt=00000000/7 ⓬
        Port 0 PTE Refcnt=35 ⓭
                         .
                         .
                         .
659 Global Pages Used, 1329 Global Pages Unused
15 Global Sections Used, 17 Global Sections Unused ⓮
```

❶ Display of global sections in local memory

❷ Name of global section

❸ Version number (in hexadecimal) of global section; high-order byte (01 in CRFSHR_003) contains major identification and low-order bytes (0003E8 in CRFSHR_003) contain minor identification

❹ The global section is writeable

❺ The global section is copy-on-reference

❻ The global section is permanent; TMP indicates a temporary global section

❼ The global section is system-wide; GRP and a group number indicate a group-wide section

❽ Number of pages in the section and number of times pages from the section have been referenced

❾ Number of global sections created, number of global pages used, and number of global pages unused in local memory

❿ Display of global sections in shared multiport memory unit; one such display appears for each attached multiport memory unit; the multiport memory unit 2 in the example is named SHM

⓫ Number of the port from which the global section was created

⓬ Base page frame number (in hexadecimal) and number of pages in the section

Figure 7-3 Global Sections Display

INSTALL (Cont.)

13	Page table entry (PTE) reference count; one appears for each port where the reference count is not 0
14	Number of global pages used, number of global pages unused, number of global sections used, and number of global sections unused in shared memory

Figure 7-3 (Cont.) Global Sections Display

7.3.6 Specifying and Reading Privileges

Table 7-2 names the 30 currently available privileges and identifies the bit in the privilege vector with which each is associated. The names are used with the /PRIVILEGED qualifier to install known images with privileges. The bit locations are used to interpret the privilege bits displayed (in hexadecimal) by the /FULL qualifier. The table lists the bits from low order to high order as they exist in the first longword shown in the display. (Two bits of the first longword and the entire second longword are currently unused.) A display of 00000001,00000000 means the known image has the CMKRNL privilege; a display of 00000002 means the known image has the CMEXEC privilege; a display of 00000003 means the known image has the CMKRNL and CMEXEC privileges; and so on.

Table 7-2
Privilege Names and Bit Locations

Bit	Name	Bit	Name	Bit	Name	Bit	Name
0	CMKRNL	8	GROUP	16	WORLD	24	PRMGBL
1	CMEXEC	9	ACNT	17	MOUNT	25	SYSGBL
2	SYSNAM	10	PRMCEB	18	OPER	26	PFNMAP
3	GRPNAM	11	PRMMBX	19	EXQUOTA	27	SHMEM
4	ALLSPOOL	12	PSWAPM	20	NETMBX	28	SYSPRV
5	DETACH	13	ALTPRI	21	VOLPRO	29	BYPASS
6	DIAGNOSE	14	SETPRV	22	PHY_IO	30	unused
7	LOG_IO	15	TMPMBX	23	BUGCHK	31	unused

7.3.7 Terminating INSTALL

The system manager terminates INSTALL and returns to DCL command level by typing <CTRL/Z>.

7.3.8 Error Messages

Table 7-3 lists the messages issued by the INSTALL utility (in alphabetical order).

INSTALL (Cont.)

Table 7-3
INSTALL Error Messages

Message	Remedy
ambiguous qualifier or privilege name	Enter the name in its entirety.
command line too long	Shorten the command line or continue it with a hyphen.
error creating global sections	Submit an SPR. This message indicates an internal error.
invalid qualifier or privilege name	Enter a valid name as shown in Table 7-1.
known file not found	Check for the correct file specification. The LIST function can help.
name already in use	Either use another file specification; use the REPLACE function (if updating an existing image); or delete the existing image with the name to be used.
no room to add a new known file list	Either delete existing known images, or increase the KFILSTCNT and/or NPAGEDYN system parameters (and reboot the system).
unable to make image header	Increase the size of the PAGEDYN system parameter (and reboot the system) if a lack of space in the paged dynamic pool is suspected. However, image headers exceeding one block in size cannot be made resident, and this could be the problem. (Large headers result from linking many shareable images.)

The INSTALL user also receives error messages from the Create and Map Section ($CRMPSC) system service and the image activator ($IMGACT).

7.4 OPERATIONS

Certain operational considerations come into play when known images are installed and used.

7.4.1 Start-Up Procedures

Known file lists only last while the system is up. If the system is shut down or fails for any reason, all known images must be reinstalled after the system is booted. For this reason, the site-independent start-up command procedure, SYS$SYSTEM:STARTUP.COM, includes an INSTALL run that installs certain system programs as known images. The system manager is encouraged to include in the site-specific start-up command procedure, SYS$SYSDISK:[SYSMGR]SYSTARTUP.COM, an INSTALL run to install additional images that are run frequently, that are usually run concurrently by several processes, or that require special privileges. (See Chapter 8 for information on the start-up command procedures.)

7.4.2 Order of Installation

In local memory, installing less frequently used images first and more frequently used images last (on each known file list) enhances run-time performance. In multiport memory, installing the more frequently used images first enhances run-time performance.

7.4.3 Privileges

Images to be installed with privileges should be linked with the /NODEBUG and /NOTRACE qualifiers to maintain system integrity and security.

Installing a shareable image with privileges does not assign those privileges to executable images linked with it. The executable images must be installed with privileges.

7.4.4 Deleting Known Images and Dismounting Volumes

System operations are affected by two characteristics of known images:

- Deletion -- A known image is not deleted as soon as the /DELETE qualifier is applied. The deletion occurs only after all processes using the image have released it.

- Dismounting -- A volume cannot be dismounted while any known file lists associated with it contain entries.

To dismount a volume, then, the system manager must not only delete all known images associated with it, but must wait for all processes using those images to release them and for the system to write writeable images back to their files.

7.4.5 **Shareable Image Files**

At execution time, a shareable image must reside in the directory SYS$SHARE (which is [SYSLIB] on the system disk), or the file specification of the shareable image must be assigned to the file name. For example, if the file specification of a shareable image is DBA1:[TEST]STATSHR.EXE, the user must assign that file specification to the logical name STATSHR before running any executable image that calls STATSHR:

```
$ DEFINE STATSHR DBA1:[TEST]STATSHR
```

The file type defaults to EXE. If the file specification for STATSHR were SYS$SHARE:STATSHR.EXE, no assignment statement would be necessary.

Likewise, one shareable image can be substituted for another without requiring the calling executable image to relink. The user simply assigns the file specification of the new shareable image to the file name of the old shareable image. The following statement assigns DBA1:[MAIN]STATSHR.EXE as the shareable image for executable images calling STATSHR:

```
$ DEFINE STATSHR DBA1:[MAIN]STATSHR
```

Again the file type defaults to EXE.

If the new image is installed with the shared qualifier, executable images linked against the old image will be mapped to global sections for the new image. Otherwise, they will be mapped to private sections for the new image.

As demonstrated in the example, the old and new images can have the same name, but must reside in different directories. The system manager should not substitute one version of a file for another in the same directory.

7.4.6 **Multiport Memory**

To install a shared image so that the global sections will reside in a multiport memory unit, the system manager issues a DEFINE command (an ASSIGN command could also be used) in the format:

```
DEFINE GBL$file-name shmem-name:file-name
```

The following example ensures that any global sections created for an image whose file name is STATSHR reside in the multiport memory unit whose logical name is SHRMEM1:

```
$ DEFINE GBL$STATSHR SHRMEM1:STATSHR
$ RUN SYS$SYSTEM:INSTALL
INSTALL> STATSHR /OPEN /SHARED
```

CHAPTER 8

START-UP COMMAND PROCEDURES

The software distribution kit contains two start-up command procedures:

- SYS$SYSTEM:STARTUP.COM -- Commands that, in general, must be executed at initialization time for any VAX/VMS system to run properly
- SYS$SYSDISK:[SYSMGR]:SYSTARTUP.COM -- An empty file, called by STARTUP.COM, that the system manager can load with site-specific initialization commands

The system manager can tailor portions of the site-independent command procedure (STARTUP.COM) and should load the site-specific command procedure (SYSTARTUP.COM). These tasks can be accomplished with any text editor. The system manager should generally put site-specific commands in SYSTARTUP.COM rather than modifying STARTUP.COM, as a new site-independent start-up command file is provided with each major release of VAX/VMS.

8.1 SITE-INDEPENDENT START-UP COMMAND PROCEDURE

STARTUP.COM is automatically executed immediately after the operating system has been booted. The command procedure includes commands for performing housekeeping chores, assigning system-wide logical names, installing known images, building the I/O data base and loading the I/O drivers, enabling VAX-11 RMS file sharing, calling the site-specific start-up command procedure, and logging out.

8.1.1 Housekeeping Chores

The first two commands in STARTUP.COM ensure that execution of the command procedure occurs without the commands being echoed on the terminal and without interruption on an error condition:

```
$ VERIFY = 'F$VERIFY(0)
$ SET NOON
```

The third command sets the default directory to the location of the system executable images:

```
$ SET DEFAULT [SYSEXE]
```

8.1.2 Logical Name Assignments

STARTUP.COM makes system-wide logical name assignments for various system components. The system manager can edit STARTUP.COM to remove the logical name assignments for components not being used at the site or to add logical name assignments for other components (such as VAX-11 BLISS-32 and VAX-11 BASIC).

NOTE

> The logical name assignments for FOR005, FOR$ACCEPT, and FOR$READ to SYS$INPUT, and FOR006, FOR$PRINT, and FOR$TYPE to SYS$OUTPUT are embedded in VAX-11 FORTRAN, and need not be stated explicitly.

8.1.2.1 **Symbolic Debugger** - The VAX-11 Symbolic Debugger requires the following logical name assignments:

```
$ ASSIGN/SYSTEM SYS$INPUT: DBG$INPUT:
$ ASSIGN/SYSTEM SYS$OUTPUT: DBG$OUTPUT:
```

8.1.2.2 **COBOL Programs** - VAX-11 COBOL-74 programs require the following logical name assignments:

```
$ ASSIGN/SYSTEM SYS$INPUT:  COB$INPUT
$ ASSIGN/SYSTEM SYS$OUTPUT:  COB$OUTPUT
$ ASSIGN/SYSTEM SYS$ERROR:  COB$CONSOLE
$ ASSIGN/SYSTEM SYS$INPUT:  COB$CARDREADER
$ ASSIGN/SYSTEM SYS$INPUT:  COB$PAPERTAPEREADER
$ ASSIGN/SYSTEM SYS$OUTPUT:  COB$LINEPRINTER
$ ASSIGN/SYSTEM SYS$OUTPUT:  COB$PAPERTAPEPUNCH
```

8.1.2.3 **PASCAL Programs** - VAX-11 PASCAL programs require the following logical name assignments:

```
$ ASSIGN/SYSTEM SYS$INPUT:  PAS$INPUT
$ ASSIGN/SYSTEM SYS$OUTPUT:  PAS$OUTPUT
```

8.1.2.4 **RSX-11M Programs** - RSX-11M compatibility mode programs (such as BAD, SOS, and PIP) require the following logical name assignments:

```
$ ASSIGN/SYSTEM 'F$LOG("SYS$DISK") LB:
$ ASSIGN/SYSTEM 'F$LOG("SYS$DISK") LB0:
$ ASSIGN/SYSTEM 'F$LOG("SYS$DISK") WK:
$ ASSIGN/SYSTEM 'F$LOG("SYS$DISK") WK0:
$ ASSIGN/SYSTEM 'F$LOG("SYS$DISK") SP:
$ ASSIGN/SYSTEM 'F$LOG("SYS$DISK") SP0:
```

8.1.2.5 **System Processors** - The language processors, the VAX-11 Linker, the image activator, and the help processor require the following logical name assignments:

```
$ ASSIGN/SYSTEM 'F$LOG("SYS$DISK")[SYSLIB] SYS$LIBRARY:
$ ASSIGN/SYSTEM 'F$LOG("SYS$DISK")[SYSMSG] SYS$MESSAGE:
$ ASSIGN/SYSTEM 'F$LOG("SYS$DISK")[SYSHLP] SYS$HELP:
```

8.1.3 **Known Images**

STARTUP.COM installs certain system executable images that are run frequently, that are usually run concurrently by several processes, or that require special privileges:

```
$ RUN SYS$SYSTEM:INSTALL
SETP0     /PRIVILEGED=(CMKRNL)
PRTSMB    /OPEN /SHARED
INIT      /PRIVILEGED=(CMKRNL,PHY_IO,SYSPRV)
VMOUNT    /PRIVILEGED=(CMKRNL,DETACH,BYPASS,SETPRV,-
          ALTPRI,TMPMBX,WORLD,GROUP,EXQUOTA,ACNT,-
          PHY_IO,BUGCHK,MOUNT)
DISMOUNT /PRIVILEGED=(CMKRNL,EXQUOTA,BUGCHK)
SUBMIT    /PRIVILEGED=(TMPMBX)
REQUEST   /PRIVILEGED=(TMPMBX)
MAIL      /PRIVILEGED=(SYSPRV,OPER,GROUP,WORLD) /OPEN
DISPLAY   /PRIVILEGED=(CMKRNL,CMEXEC) /OPEN-
          /HEADER_RESIDENT /SHARED
LOGINOUT /PRIVILEGED=(CMKRNL,CMEXEC,TMPMBX,EXQUOTA,SYSPRV)
SYS$SHARE:DCLTABLES /OPEN /HEADER_RESIDENT /SHARED
SYS$SHARE:DCLINTPRT /OPEN /HEADER_RESIDENT /SHARED
DCL       /OPEN /HEADER_RESIDENT /SHARED
SET       /PRIVILEGED=(CMKRNL,SYSPRV,TMPMBX) /OPEN-
          /HEADER_RESIDENT /SHARED
SHOW      /PRIVILEGED=(CMKRNL,CMEXEC,TMPMBX,NETMBX,-
          OPER) /OPEN /HEADER /SHARED
SOS       /OPEN /SHARED

SYS$SHARE:RSXSHR /OPEN /SHARED /HEADER_RESIDENT
SYS$SHARE:RSXUSR /OPEN /SHARED /HEADER_RESIDENT
RSX       /OPEN /HEADER_RESIDENT /SHARED
BACKTRANS /OPEN /HEADER_RESIDENT /SHARED

SYS$LIBRARY:DEBUG /OPEN /SHARED

SYS$LIBRARY:TRACE /OPEN /SHARED

SYS$SHARE:VMSRTL /OPEN /SHARED /HEADER_RESIDENT

SYS$SHARE:LBRSHR /OPEN /SHARED /HEADER_RESIDENT

SYS$SHARE:CRFSHR /OPEN /SHARED /HEADER_RESIDENT

SYS$SHARE:SUMSHR /OPEN /SHARED /HEADER_RESIDENT
```

The system manager should review the above installed images and eliminate any that are not used or are used infrequently. If necessary, the executable images should be rearranged in reverse order of frequency of use -- that is, the most frequently used images should be installed last.

8.1.4 I/O Devices and Drivers

STARTUP.COM automatically connects devices physically attached to the system and loads their I/O drivers:

```
$ RUN SYS$SYSTEM:SYSGEN
AUTOCONFIGURE ALL
```

For installations that require permanent residence of the console block storage device, the system manager must explicitly connect the device by including the following command in the SYSGEN utility run:

```
CONNECT CONSOLE [/DRIVERNAME=device-driver]
```

See Chapter 13 for a detailed discussion of the SYSGEN utility.

The console block storage device should also be mounted with a command of the form:

```
$ MOUNT /SYSTEM /PROT=(SYSTEM:RWLP)-
CSA1:  CONSOLE
```

Failure to mount the console block storage device with appropriate protection permits random users to mount and access it, as the device is in RT-11 format and has no file protection.

8.1.5 VAX-11 RMS File Sharing

STARTUP.COM enables VAX-11 RMS file sharing with a maximum page count of 20:

```
$ RUN SYS$SYSTEM:RMSSHARE
20
```

If the system manager estimates a different maximum page count (see Section 5.4), that figure should replace 20. In particular the system manager should ensure that this value does not exceed half the size of paged dynamic memory.

8.1.6 Termination of the Procedure

STARTUP.COM calls the site-specific start-up command procedure, sets the number of users that can log in at one time, and logs the site-independent command procedure off:

```
$ @[SYSMGR]SYSTARTUP.COM
$ SET LOGIN /INTERACTIVE=64
$ LOGOUT/BRIEF
```

If necessary, the system manager can replace 64 with a user limit that is more reasonable for the installation.

8.2 SITE-SPECIFIC START-UP COMMAND PROCEDURE

SYSTARTUP.COM, called from STARTUP.COM, performs any commands the system manager wants to place there. Typically, these commands mount system disks, assign logical names, set the characteristics of terminals and other devices, establish and start queues, install known images, run the System Dump Analyzer, purge the operator's log file, submit batch jobs that are run at the time the system is initialized and that are periodically resubmitted, manually connect devices and multiport memory units, install secondary paging and swapping files, and announce that the system is up and running. The commands shown in the following sections are provided as models; they should not be copied line for line.

8.2.1 System Disks

The following commands are typical of those used in mounting system disks:

```
$ MOUNT /SYSTEM DBA1:  BUILDATA
$ MOUNT /SYSTEM DBA2:  BILLING
$ MOUNT /SYSTEM DBB2:  FORTRAN
```

See Chapter 5 for more information on mounting system disks.

8.2.2 Logical Names

The system manager can assign system-wide logical names in addition to the logical names assigned in the site-independent start-up command procedure:

```
$ ASSIGN/SYSTEM 'F$LOG("SYS$SYSDISK") SYSDSK
$ ASSIGN/SYSTEM SYSD$:[SYSMGR] SYSMGR
```

The lexical function 'F$LOG("SYS$SYSDISK") returns the name of the system device. See Chapter 2 of the VAX/VMS Command Language User's Guide for more detailed information on logical name assignments.

8.2.3 Device Characteristics

The system manager uses a series of SET commands to establish the characteristics of the terminals and other devices on the system:

```
$ SET TERMINAL TTC2:  /SPEED=300/LA36/PERM
$ SET TERMINAL TTD1:  /SPEED=9600/PERM
$ SET TERMINAL TTD4:  /SPEED=1200/PERM
$ SET PRINTER  LPA0:  /LOWER /NOCR
$ SET DEVICE   LPA0:  /SPOOLED
```

Note that the /SPEED qualifier sets both transmission and reception speeds to the same value. Printer characteristics (SET PRINTER and SET DEVICE above) must be set prior to establishing queues for the printers.

8.2.4 Queues

The first time the system is booted, batch and print queues must be initialized and started. Whenever the system is rebooted, the queues must just be started. The following commands provide for both contingencies by testing $STATUS:

```
$               START /QUEUE LPA0
$               IF $STATUS THEN GOTO ENDLPA0
$               INITIALIZE /QUEUE /FLAG LPA0
$               START /QUEUE LPA0
$ ENDLPA0:      START /QUEUE LPB0
$               IF $STATUS THEN GOTO ENDLPB0
$               INITIALIZE /QUEUE /FLAG LPB0
$               START /QUEUE LPB0
$ ENDLPB0:
```

See Chapter 9 for more information on initializing and starting queues.

8.2.5 Known Images

The system manager often installs user and system programs (in addition to the ones installed in the site-independent start-up command procedure) so that they can be located quickly, shared, or provided privileges:

```
$ RUN SYS$SYSTEM:INSTALL
BLISS32 /OPEN /SHARED /HEADER_RESIDENT
COPY /OPEN
LINK /OPEN
USERS /PRIV=WORLD
MACRO32 /OPEN /SHARED
TALK /PRIV=(OPER,SYSNAM,WORLD,GROUP,PRMMBX)
DIRECTORY /OPEN
```

The most frequently used images should be installed last in local memory, but first in shared memory. See Chapter 7 for more information on installing known images.

8.2.6 System Dump Analyzer

Each time the system is booted, the system manager should run the System Dump Analyzer (in case the system failed the last time it was running):

```
$ RUN SYS$SYSTEM:SDA
SYS$SYSTEM:SYSDUMP.DMP
COPY [SYSERR]SYSDUMP.DMP
SET OUTPUT LPA0:SYSDUMP.LIS
SHOW CRASH
SHOW STACK/ALL
SHOW SUMMARY
SHOW PROCESS /PCB /PHD /REGISTERS
SHOW SYMBOL/ALL
EXIT
```

If further information is required, the system manager can invoke the System Dump Analyzer for an interactive session upon completion of startup. See the VAX/VMS System Dump Analyzer Reference Manual.

8.2.7 Operator's Log File

Each time the system is booted, the system manager should purge all but the last two or three versions of the operator's log file:

```
$ PURGE /KEEP=2 [SYSMGR]OPERATOR.LOG
```

8.2.8 Standard Batch Jobs

Some sites may have batch jobs that are submitted at system start-up time and that resubmit themselves to run at intervals as long as the system is running. For such jobs, the SUBMIT command is used in the start-up file:

```
$ SUBMIT SYS$SYSTEM:LOGJOBS
```

8.2.9 Manually Connected Devices and Multiport Memory Units

The system manager runs the SYSGEN utility to connect devices not automatically connected in STARTUP.COM and to initialize or connect multiport memory units:

```
$ RUN SYS$SYSTEM:SYSGEN
CONNECT NET /NOADAPTER /DRIVER=NETDRIVER
SHARE MPM1 SHR_MEM_1 /INIT
```

8.2.10 Secondary Paging and Swapping Files

The system manager runs the SYSGEN utility to install secondary paging and swapping files:

```
$ RUN      SYS$SYSTEM:SYSGEN
INSTALL  DRA5:[SYSTEM]PAGEFILE.SYS /PAGEFILE
INSTALL  DRA5:[SYSTEM]SWAPFILE.SYS /SWAPFILE
```

8.2.11 Announcement

The last command in SYSTARTUP.COM typically announces to all terminals that the system is up and running:

```
$ REPLY /ALL /BELL "VAX/VMS System Initialized"
```

CHAPTER 9

BATCH AND PRINT JOBS

System performance can be greatly influenced by how the system manager establishes spooled devices, creates and controls input and output queues, and controls batch and print jobs. Typically, the system manager of a VAX/VMS operating system is responsible for performing the following four closely related functions:

- Establishing spooling of input and output -- The VAX/VMS operating system supports input spooling of batch job files from card readers and transparent spooling of output files for line printers and terminals. Using DCL commands, the system manager specifies which output devices are to be spooled. Section 9.1 describes spooling and the use of DCL commands to establish spooled devices.

- Controlling batch jobs -- Section 9.2 describes batch processing and the use of DCL commands to control batch jobs.

- Controlling print jobs -- Section 9.3 describes queuing output to line printers and the use of DCL commands to control print jobs.

- Creating and controlling terminal queues -- See Section 9.4.

The system manager need not learn all the inner workings of spooling and queuing. However, a working knowledge of how to establish spooled devices and how to create and control queues is essential for the system manager to keep the system running efficiently. The kind of working knowledge that the system manager needs presupposes a familiarity with the DCL commands listed in Table 9-1. The use of these commands is restricted to users who have operator privilege (OPER), typically, the system manager and system operators. The VAX/VMS Operator's Guide fully describes these commands.

In addition, three other DCL commands play a role in the control of batch and print jobs:

- SHOW QUEUE -- Displays information about a file (or files) queued for batch execution or for output. No privilege is needed to use this command.

- SET QUEUE -- Changes the attributes of a file (or files) queued for batch execution or for output.

  Ordinarily, no privilege is needed to use this command; operator privilege (OPER) is needed, however, to use the command to:

  - Modify queued jobs entered by a member of another group

  - Increase the queue priority of a job

- DELETE/ENTRY -- Deletes jobs from queues.

  No privilege is needed to delete entries queued by oneself; world (WORLD) or operator privilege (OPER) is needed, however, to use this command to delete a queued job entered by a member of another group, and group privilege (GROUP) is needed to delete a queued job entered by another member of the same group.

  These commands are described fully in the VAX/VMS Command Language User's Guide.

Table 9-1
Operator Commands Used in Regulating Spooling and Queuing

Command	Function
SET DEVICE/SPOOLED	Establishes spooled printers or terminals, and assigns queues to them
SET DEVICE/NOSPOOLED	Turns off spooling of printers or terminals
INITIALIZE/QUEUE	Creates queues
DELETE/QUEUE	Deletes queues
START/QUEUE	Starts queues
STOP/QUEUE	Stops queues
ASSIGN/QUEUE	Assigns queues to devices
DEASSIGN/QUEUE	Deassigns queues from devices
ASSIGN/MERGE	Empties queues of jobs and places them in other queues
STOP/ABORT	Aborts printing of files currently being printed
STOP/REQUEUE	Stops the printing of jobs currently being printed and requeues them at the end

9.1 SPOOLING

Spooling is the technique of using secondary storage to buffer data passing between slow I/O devices (such as line printers and card readers) and physical memory. The slow devices, which can be either the ultimate sources or the ultimate destinations of buffered I/O data, are called spooled devices; the secondary storage devices are called intermediate devices.

The system manager establishes the spooled devices; to all other users, and their programs, the mechanism of spooling is transparent.

Input spooling makes input from a spooled device (such as a card reader) available for processing by placing it into a file on an intermediate device (such as a disk). Input spooling is used principally to create, from card reader input, batch input files on disk. After they are spooled to disk, batch jobs are queued for processing according to their priority.

Output spooling makes output from the processor available for transmission to a spooled device (such as a line printer) by placing it into files on an intermediate device (such as a disk). Output spooling is used principally to create printer output files on disk. After they are spooled to disk, print jobs are queued for printing according to their priority.

As a rule, programs demand input and produce output at irregular intervals during their execution. If programs were allowed to read directly from slow devices and to write directly to slow devices, the execute time of programs would be limited by the speed of the slow devices. If a process output directly to a printer, the process would be tied up for the time it took to print the listing. Also, other processes needing the printer would have to wait.

The actual transfer of inputs from a spooled device to an intermediate device or the transfer of outputs from an intermediate device to a spooled device is carried out by processes called symbionts.

Input symbionts read input at the speed of the input spooled device and buffer it in a file on the intermediate device. Later, when the input is needed, it is read directly from the file on the intermediate device rather than from the spooled device. While one set of input data is being processed, the input symbiont is free to read another set of input data into another file on the intermediate device.

Output symbionts read data from an intermediate device and write the data to an output spooled device at the speed of that output device. The data on the intermediate device is generated by programs that produce outputs directly into files on the intermediate device. The I/O waiting time of programs is thus minimized. When an output file is complete, it is queued for printing by an output symbiont. As with input symbionts, there is an overlapping here. While an output symbiont is printing a file stored temporarily on an intermediate device, another program can be producing another output file on the intermediate device.

9.1.1 Establishing Spooled Devices

Card readers are spooled by default. To use a card reader without spooling, users must allocate the reader before making it ready to read a card deck. By default, also, the queue SYS$BATCH is used to queue spooled jobs. Thus, no special command is needed to establish card readers as spooled devices.

On the other hand, the operator command SET DEVICE must be used to establish a line printer or a terminal as a spooled device. The use of this command is restricted to users with the OPER privilege. The VAX/VMS Operator's Guide describes the SET DEVICE command in detail.

Typically, the system manager must decide which devices to include in the system's basic complement of spooled devices. Often, the system manager sets up devices for spooling by making entries in the system start-up command procedure.

At a minimum, the system manager should see that at least one line printer is set spooled when the system is started up. In a system with only one line printer, this is the default system printer. The system manager need not set a card reader spooled, because card readers are spooled by default.

Depending on system configuration and anticipated operational needs, more spooled devices can be established at start-up. Moreover, in the course of normal operations (to meet special operational needs), the system manager or the operator can define still other devices as spooled devices without having to reboot the system. Normally, all line printers should be spooled.

Finally, and most important, on a system with both spooled input devices and spooled output devices, the system manager must create and start at least one batch queue to handle spooled input and one output queue to handle output for each spooled output device.

9.1.2 Turning Off Spooling

The system manager or operator can, as necessary, turn off spooling to spooled printers and terminals by use of the SET DEVICE command.

9.2 BATCH JOBS

Batch jobs can be submitted to the VAX/VMS system and queued for execution in two ways:

- As batch job files submitted by use of the $JOB command (see the <u>VAX/VMS Command Language User's Guide</u>) from a card reader. These batch job files are spooled onto disk by an input symbiont and placed in a batch queue according to their priority. Unless the $JOB card specifies otherwise, the name of this batch queue is SYS$BATCH (by default). From the batch queue, batch jobs are selected for execution.

- As command procedure disk files submitted by use of the SUBMIT command (see the <u>VAX/VMS Command Language User's Guide</u>). These files are also placed in a batch queue and selected for execution according to their priority. Again, by default, the name of this batch queue is SYS$BATCH.

Batch jobs cannot be executed unless at least one batch queue has been created on the system and unless that queue has been started. By default, this is the batch queue SYS$BATCH.

In the VAX/VMS system, many jobs, or streams, can be executed at the same time from each of several batch queues. Thus, the system manager can create and start several batch queues at once and can specify the number of jobs, or streams, that can be executed at the same time from each queue.

Among the jobs in a batch queue that has been started, the one with the highest priority is the first candidate for execution. Whether or not that job is actually started up, however, depends on an evaluation of the following limits and conditions:

- The maximum number of batch jobs allowed to be executed from the queue at the same time. The system manager specifies this limit with either the INITIALIZE /QUEUE command or the START /QUEUE command.
- The maximum number of all jobs allowed to be executed in the system at the same time.
- The number of jobs currently being executed in the system.

Hence, the highest priority batch job in a queue is started up only if both of the following conditions are satisfied:

- Fewer than the maximum number of batch jobs allowed are currently running from the queue.
- The system is not saturated with other jobs.

9.2.1 Creating Batch Queues

The operator command INITIALIZE /QUEUE is used in creating, or initializing, a batch queue. The use of this command is restricted to users with the OPER privilege to execute operator functions. The VAX/VMS Operator's Guide describes the INITIALIZE /QUEUE command in detail.

Typically, the system manager must decide on the number of batch queues for an installation, on the job limit of each queue, on the priority of each queue, and on the swap mode of each queue. Often, the system manager creates batch queues by making entries in the system start-up command procedure.

Setting up batch queues is not restricted to start-up time. In the course of normal operations, the system manager or operator can create batch queues as operational needs dictate.

9.2.2 Starting Batch Queues

The execution of batch jobs from a batch queue (dequeuing) can only take place if the batch queue has been started. The operator command START /QUEUE starts a batch queue. The use of this command is restricted to users with the OPER privilege. The VAX/VMS Operator's Guide describes the START /QUEUE command in detail.

Typically, the system manager must see that batch queues created by use of the INITIALIZE /QUEUE command are started. Often, the system manager starts batch queues by making entries in the system start-up command procedure.

Starting batch queues is not restricted to start-up time. In the course of normal operation, the system manager or operator can start queues as operational needs dictate.

9.2.3 Stopping Batch Queues

The system manager or operator can, as necessary, abort a job in a batch queue or disable all processing from the queue until the queue is restarted by use of the START /QUEUE command. The STOP /QUEUE command is used to stop batch queues. The VAX/VMS Operator's Guide describes this command in detail.

9.2.4 Deleting Batch Queues

The system manager or operator can delete batch queues, as necessary, by use of the DELETE /QUEUE command. The VAX/VMS Operator's Guide describes this command in detail.

9.2.5 Batch Versus Interactive Jobs

The system manager should normally encourage users to submit large jobs (such as compiling and linking large programs) as batch jobs and reserve interactive use of the system for jobs that do not require extensive resources. A technique toward this end is to (1) restrict the working set size of interactive jobs by providing small (for example, 150) WSDEFAULT and WSQUOTA values in the UAF records and (2) expand the working set size of batch jobs by providing large (for example, 512) WSDEFAULT and WSQUOTA values in the START /QUEUE and INITIALIZE /QUEUE commands. The system manager can likewise restrict and expand time limits on jobs by setting the CPU values.

9.2.6 Guides to Setting Up Batch Queues

The following rules of thumb are useful in setting up a batch queue for a system that is predominantly interactive:

1. Set up one batch queue named SYS$BATCH, the name of the default batch queue.

2. Give SYS$BATCH the following characteristics:

   a. Job limit -- 1 to 4

   b. Priority -- 3 or 4

   c. Swapping mode -- swapping enabled (by default)

   d. Working set default -- 150 to 1024

   e. Working set quota -- 150 to 1024

   f. CPU default -- INFINITE

   g. CPU maximum -- INFINITE

The system manager executes the following command procedure to create and start this queue:

```
$ START/QUEUE SYS$BATCH
$ IF $STATUS THEN GO TO BATCH_DONE
$ INITIALIZE/QUEUE/BATCH/JOB_LIMIT=1/PRIORITY=3/WSDEFAULT=700-
  /WSQUOTA=700/CPUDEFAULT=INFINITE/CPUMAXIMUM=INFINITE SYS$BATCH
$ START/QUEUE SYS$BATCH
$ BATCH_DONE:
```

The first START command and the test on $STATUS ensure that an existing queue is not initialized.

Normally, these commands are contained in the start-up command procedure (see Chapter 8).

The following rules of thumb are useful in setting up batch queues for a system that is predominantly a batch system and in which editing is the principal interactive activity:

1. Set up three batch queues as follows:

   a. SYS$BATCH -- the default batch queue.

   b. FAST -- a high-priority queue for executing high-priority jobs that should not be swapped out of memory.

   c. SLOW -- a low-priority background queue for processing low-priority jobs. Typically, these are large jobs with large requirements for physical memory. Usually, it is uneconomical to swap such jobs out of memory. The system operator can adjust the system workload by stopping and restarting background queues as needed.

2. Give SYS$BATCH the following characteristics:

   a. Job limit -- 6 to 10

   b. Priority -- 4 (by default)

   c. Swapping mode -- swapping enabled (by default)

3. Give FAST the following characteristics:

   a. Job limit -- 1 (by default)

   b. Priority -- high (5, for example)

   c. Swapping mode -- swapping disabled

4. Give SLOW the following characteristics:

   a. Job limit -- 1 (by default)

   b. Priority -- low (3, for example)

   c. Swapping mode -- swapping disabled

NOTE

This configuration should not be attempted on a small system (under 1MB). Additionally, the system manager should add up the pages required for the batch working sets and insure that enough fluid memory remains for interactive jobs. Otherwise, the system manager must reduce the number of batch jobs or make the FAST and SLOW jobs swappable. In particular, the system manager must not let fluid memory drop below the value of the WSMAX system parameter, or a deadlock could result.

The system manager executes the following command procedure to create and start these queues:

```
$ START/QUEUE SYS$BATCH
$ IF $STATUS THEN GOTO BATCH_DONE
$ INITIALIZE/QUEUE/BATCH/JOB_LIMIT=6  SYS$BATCH
$ START/QUEUE SYS$BATCH
$ BATCH_DONE:
$ START/QUEUE FAST
$ IF $STATUS THEN GOTO FAST_DONE
$ INITIALIZE/QUEUE/BATCH/PRIORITY=10/DISABLE_SWAPPING  FAST
$ START/QUEUE FAST
$ FAST_DONE:
$ START/QUEUE SLOW
$ IF $STATUS THEN GOTO SLOW_DONE
$ INITIALIZE/QUEUE/BATCH/PRIORITY=3/DISABLE_SWAPPING  SLOW
$ START/QUEUE SLOW
$ SLOW_DONE:
```

Normally, these commands are contained in the site-specific start-up command procedure (see Chapter 8).

9.3 PRINT QUEUES

Unless a line printer is associated with a physical queue (a queue that has the same name as the line printer) and unless that queue has been started, no queued output can occur on that line printer.

Print jobs are queued for processing in one of two ways: without the direct intervention of a user (that is, implicitly) or with the direct intervention of a user (that is, explicitly).

An implicitly spooled file is created when a program or DCL command sends its output to a spooled printer. When an implicitly spooled print file destined for a spooled printer is closed, the file is placed in a print queue. Both the spooling of the output file to an intermediate device and the subsequent queuing of a job consisting of this file occur without the direct intervention of a user.

By use of the PRINT command, a user can explicitly queue a disk file or several files for printing. The VAX/VMS Command Language User's Guide describes the PRINT command in detail. The disk file or files specified in the PRINT command are queued as a print job; if several files make up a print job, they will be printed together.

Print jobs are placed in queues according to their priority. These queues can be any one of the following:

- Physical device queues -- Queues associated with (that is, named for) a specific line printer.
- Generic queues -- Queues from which files can be printed out on any available line printer that has correctly matching characteristics.
- Named, or logical, queues -- Queues that are not associated, even indirectly, with any device. To obtain printed output from a named, or logical, queue, the system manager or operator must explicitly assign the queue to a printer. The command ASSIGN /QUEUE is used for this purpose.

From these queues, print jobs are selected for initiation. Among the jobs in a print queue for a particular printer at any given time, the job with the highest priority is the one chosen for printing.

By default, print jobs queued by use of the PRINT command are placed in the queue named SYS$PRINT. Thus, to use the default version of the PRINT command in a system with only one line printer, the name SYS$PRINT is equated with the name of the physical line printer. To use the default version of the PRINT command in a system with several line printers of matching characteristics, SYS$PRINT is normally established as the name of a generic queue.

The maximum number of physical device queues that can be printing at one time is restricted to the value of the MAXPRINTSYMB system parameter. In particular, system managers of systems configured using the 8user and 16USER standard system parameter files should note that the default value of this parameter is 1. See Chapter 12 for further information on system parameters.

9.3.1 Creating Print Queues

The operator command INITIALIZE /QUEUE is used in creating, or initializing, a print queue. The use of this command is restricted to users with the OPER privilege to execute operator functions. The VAX/VMS Operator's Guide describes the INITIALIZE /QUEUE command in detail.

Typically, the system manager must decide on the number of print queues for an installation and on their attributes. Often, the system manager creates print queues by making entries in the site-specific start-up command procedure.

Setting up print queues is not restricted to start-up time. In the course of normal operations, the system manager or operator can create print queues as operational needs dictate.

9.3.2 **Starting Print Queues**

The initiation of print jobs from a print queue (dequeuing) can only take place if the print queue has been started. The operator command START /QUEUE starts a print queue. The use of this command is restricted to users who have the OPER privilege to execute operator functions. The VAX/VMS Operator's Guide describes the START /QUEUE command in detail. All options that can be specified in the INITIALIZE /QUEUE command can also be specified in the START /QUEUE command.

Typically, the system manager must see that print queues created by use of the INITIALIZE /QUEUE command are started. Often, the system manager starts print queues by making entries in the site-specific start-up command procedure.

Starting print queues is not restricted to start-up time. In the course of normal operations, the system manager or operator can start queues as operational needs dictate.

9.3.3 **Stopping Print Queues**

The system manager or operator can abort a job in a print queue, suspend the printing of a job currently being printed, or disable processing from the queue entirely until the queue is restarted by use of the START /QUEUE command. The STOP /QUEUE command is used to stop print queues and to suspend printing of jobs. The VAX/VMS Operator's Guide describes this command in detail.

9.3.4 **Deleting Print Queues**

The system manager or operator can delete print queues, as necessary, by use of the DELETE /QUEUE command. The VAX/VMS Operator's Guide describes this command in detail.

9.3.5 **Assigning a Named, or Logical, Print Queue to a Printer**

The operator command ASSIGN /QUEUE is used in assigning, or redirecting, a named, or logical, print queue to a printer. The use of this command is restricted to authorized users with the OPER privilege to execute operator functions. The VAX/VMS Operator's Guide describes the ASSIGN /QUEUE command in detail.

To produce printer output, a logical queue must first be assigned to a printer and then started.

Typically, the print files of a group of low-priority users can be placed in a logical queue and held there until off-peak hours. Then, to print the files, the system operator can assign the queue to a line printer and start the queue.

9.3.6 Deassigning a Named, or Logical, Print Queue from a Printer

The operator command DEASSIGN /QUEUE is used in deassigning a named, or logical, print queue from a printer. The use of this command is restricted to users with the OPER privilege. The VAX/VMS Operator's Guide describes the DEASSIGN /QUEUE command in detail.

9.3.7 Vertical Page Size

By default, the various system utilities (including the editors, compilers, and linker) produce listings with a vertical page size of 66 lines. The system manager may change the vertical page size for listings produced by the native mode utilities by specifying a numeric value, an integer in the range of 30 to 99, inclusive, for the system logical name SYS$LP_LINES. The following example changes the vertical page size for all users to 60 lines per page:

```
$ DEFINE /SYSTEM SYS$LP_LINES 60
```

Individual users may change the vertical page size on a group or process basis.

A vertical page size of less than that specified by SET PRINTER /PAGE (which defaults to 64) causes a skip to the head of the next form each time the specified line count is reached. A greater vertical page size causes the excess lines to overflow to the next form and then a skip to head of form when the count is reached.

9.3.8 Forms Control

The system manager or operator specifies the forms type of a print queue with the /FORMS_TYPE qualifier of the INITIALIZE /QUEUE or START /QUEUE command. If a user enters a print job with a forms value (/FORMS qualifier of the PRINT command) different from that of the queue, the job is placed on a hold status until the forms value of the queue is set equal to the forms value of the job. (The operator should stop the queue, physically change the forms, and start the queue specifying the new value for the /FORMS_TYPE qualifier.)

The forms type can be specified as a number or an alphabetic code. Alphabetic codes must be defined in the file SYS$SYSDISK:[SYSMGR]:FORMSTYPE.DAT, one code per line, in the following format:

```
% code number comments
```

The system manager can include lines of text in the file if desired. Only lines beginning with a percent sign are taken as code-number definitions. The following example defines the code NORMAL as the number 0:

```
% NORMAL 0 NORMAL LINE PRINTER PAPER
```

A specification of /FORMS=NORMAL (or /FORMS=N, /FORMS=NO, and so on) is interpreted to mean /FORMS=0. Note, however, that the first match found in FORMSTYPE.DAT prevails with no ambiguity checks made, so that potentially conflicting names (that is, names starting with the same letter) should be avoided.

9.3.9 **Printer Characteristics**

The system manager or operator specifies the printer characteristics of a print queue with the /CHARACTERISTICS qualifier of the INITIALIZE /QUEUE or START /QUEUE command. If a user enters a print job with a characteristic (/CHARACTERISTICS qualifier of the PRINT command) not included in those for the queue, the job is placed on a hold status until the characteristics of the queue are set to include all the characteristics of the job. (The operator should stop the queue, physically change the characteristics of the printer, and start the queue specifying the new values for the /CHARACTERISTIC qualifier.)

A characteristic can be specified as a number or an alphabetic code. Alphabetic codes must be defined in the file SYS$SYSDISK:[SYSMGR]CHARTYPE.DAT, one code per line, in the following format:

```
% code number comments
```

The system manager can include lines of text in the file if desired. Only lines beginning with a percent sign are taken as code-number definitions. The following example defines the code REDINK as the number 3:

```
% REDINK 3
```

Subsequent INITIALIZE /QUEUE or /START /QUEUE and PRINT commands can use the number 3 or the alphabetic code REDINK to describe one printer characteristic. Note, however, that the first match found in CHARTYPE.DAT prevails with no ambiquity checks made, so that potentially conflicting names (that is, names starting with the same letter) should be avoided.

9.3.10 **Guides to Setting Up Print Queues and Spooled Line Printers**

The following rules of thumb are useful in setting up and regulating print queues and spooled line printers:

1. Normally, set all line printers spooled.

2. To produce output on a spooled line printer, initialize a print queue with the same name as the spooled printer and start that queue.

3. If more than one line printer is on the system, enable generic printing from as many print queues as possible, and make at least one print queue (SYS$PRINT) a generic queue. Queues for line printers that are in remote locations, that use special forms, or that possess unique printer characteristics should not be enabled for generic printing.

4. To use special forms or apply unique printer characteristics to a general-purpose queue, stop the queue, physically change the forms or apply the printer characteristics, and start the queue with appropriate /FORMS_TYPE or /CHARACTERISTICS qualifiers. After the special jobs are printed, stop the queue, physically reset the forms or printer characteristics, and start the queue with the original /FORMS or /CHARACTERISTICS values.

Figures 9-1 through 9-4 illustrate some of the most common arrangements of spooled line printers and print queues. These figures can be used, with the rules of thumb listed above, as guidelines in setting up spooled line printers and print queues.

NOTE

The commands shown in the examples assume manual entry at run time. Command procedures -- especially start-up command procedures -- containing INITIALIZE and START commands should contain logic to ensure that an existing queue is not initialized. See Chapter 8. If an existing queue is initialized, any jobs in that queue are lost.

Figure 9-1 illustrates a typical spooling and queuing configuration for a system with one line printer. The commands listed in this figure produce the following results:

1. The line printer LPA0 is set spooled.
2. System wide, the logical name SYS$PRINT is equated with the name LPA0. The equivalence of these names is recorded in the system logical name table.
3. The print queue LPA0 is initialized and started.
4. All print jobs explicitly directed to the printer LPA0 are placed in the queue LPA0 and are printed from that queue.
5. All print jobs that normally would be placed by default in a queue named SYS$PRINT (if that queue existed) are actually placed in the queue LPA0 (in this case, the system default print queue) and are printed from that queue.

COMMANDS

```
$ SET DEVICE/SPOOLED=LPA0 LPA0
$ ASSIGN/SYSTEM LPA0  SYS$PRINT
$ INITIALIZE/QUEUE/FLAG  LPA0
$ START/QUEUE  LPA0
```

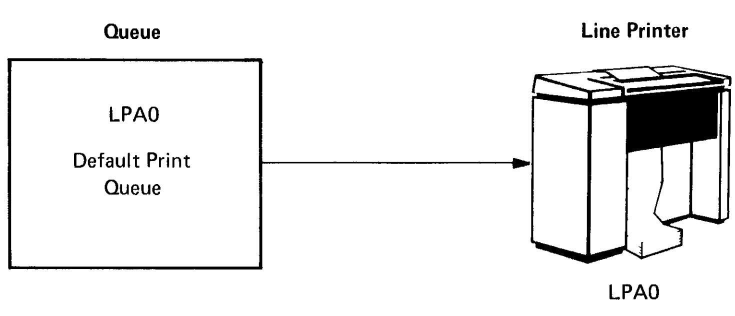

Figure 9-1 Setting Up a Spooled Printer and a Print Queue on a System with One Line Printer

Figure 9-2 illustrates a typical spooling and queuing configuration for a system with two line printers that have the same characteristics. The commands listed in this figure produce the following results:

1. The line printer LPA0 is set spooled.

2. The line printer LPB0 is set spooled.

3. The generic queue SYS$PRINT is initialized and started.

4. Physical queues LPA0 and LPB0 are initialized and started, with generic printing enabled by default.

5. All print jobs explicitly directed by use of the PRINT command to one of the two printers are placed in the queue associated with the specified printer.

6. Print jobs queued by use of the PRINT command without device specification are placed by default in the generic queue SYS$PRINT. From the generic queue, jobs are printed on whichever printer is free, by way of either of the two physical queues, LPA0 or LPB0.

7. Spooled print files destined either for LPA0 or for LPB0 are placed in the generic queue SYS$PRINT, which was associated with both these printers. From the generic queue, these jobs are printed on whichever printer is free.

Figure 9-3 illustrates a typical spooling and queuing configuration for a system with three line printers: two that have the same characteristics and one that uses special forms, has unique printer characteristics, or is in a remote location. The configuration shown in Figure 9-3 is basically the same as the one in Figure 9-2, with the addition of the spooled printer LPC0 and the creation and starting of the queue LPC0. Because of the special forms, unique characteristics, or the remote location, printer LPC0 is not suited for general printing.

COMMANDS

```
$ SET DEVICE/SPOOLED LPA0
$ SET DEVICE/SPOOLED LPB0
$ INITIALIZE/QUEUE/FLAG/GENERIC SYS$PRINT
$ START/QUEUE SYS$PRINT
$ INITIALIZE/QUEUE/FLAG LPA0
$ START/QUEUE LPA0
$ INITIALIZE/QUEUE/FLAG LPB0
$ START/QUEUE LPB0
```

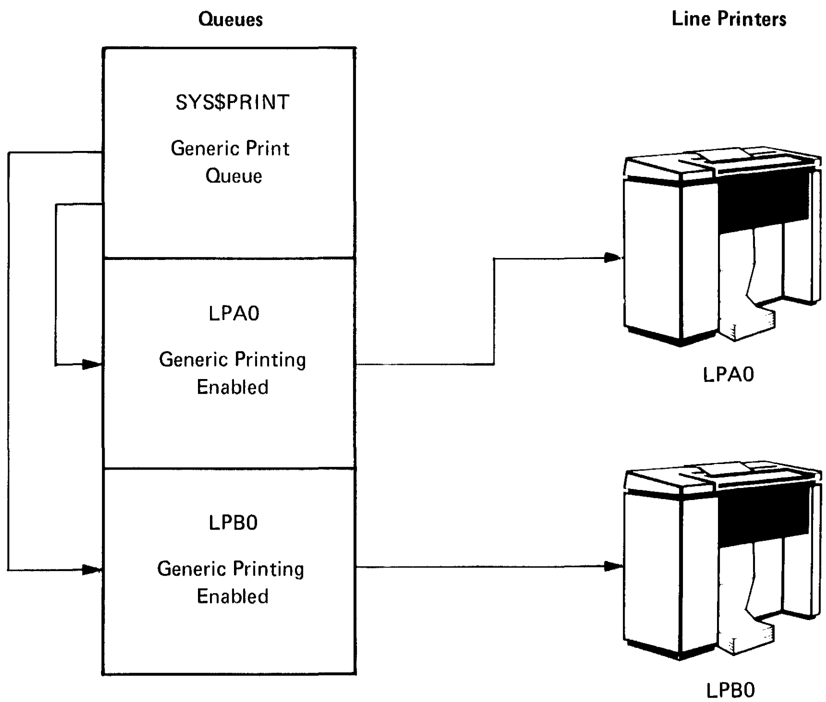

Figure 9-2 Setting Up Spooled Printers and Print Queues on a System with Two Line Printers with the Same Characteristics

COMMANDS

```
$ SET DEVICE/SPOOLED LPAO
$ SET DEVICE/SPOOLED LPBO
$ SET DEVICE/SPOOLED=LPCO LPCO
$ INITIALIZE/QUEUE/FLAG/GENERIC SYS$PRINT
$ START/QUEUE SYS$PRINT
$ INITIALIZE/QUEUE/FLAG LPAO
$ START/QUEUE LPAO
$ INITIALIZE/QUEUE/FLAG LPBO
$ START/QUEUE LPBO
$ INITIALIZE/QUEUE/FLAG/NOENABLE_GENERIC_PRINTING LPCO
$ START/QUEUE LPCO
```

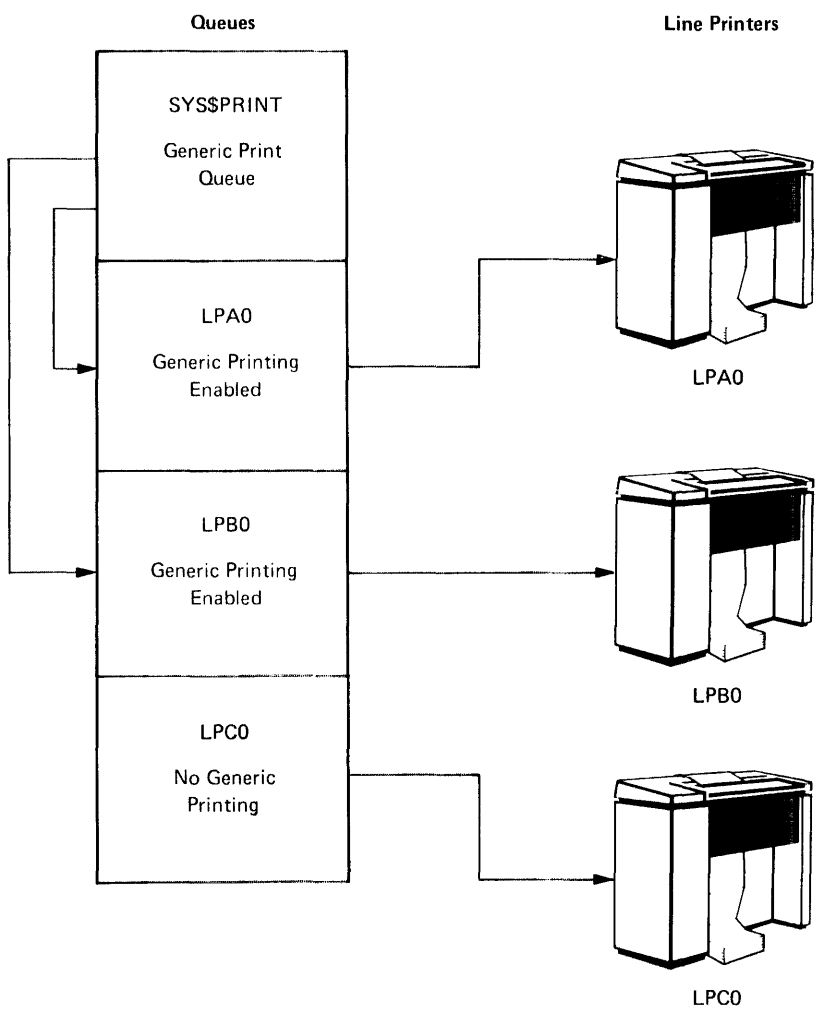

Figure 9-3 Setting Up Spooled Printers and Print Queues on a System with Three Line Printers; Two with the Same Characteristics and One with Special Characteristics or in a Remote Location

Thus, with the following exceptions, the commands listed in Figure 9-3 produce the same results as the commands listed in Figure 9-2:

1. The line printer LPC0 is set spooled.

2. The physical queue LPC0 is initialized and started with generic printing disabled.

3. Only print jobs explicitly directed to the printer LPC0 are ever placed in the queue LPC0; no generic printing is ever done on printer LPC0 by way of the queue SYS$PRINT.

Figure 9-4 adds still another feature to the configuration illustrated in Figure 9-3. This is a logical queue, which is a named, nongeneric queue that is not directly associated with any line printer. When a logical queue is assigned to a line printer and started, however, output to a line printer can occur.

Logical queues can be used, for example, to hold print jobs of low-priority users for printing during off-peak hours. To channel the print jobs of these users into a logical queue, the name of the logical queue (HOLD, for example) must be assigned to the name of their default print queue (SYS$PRINT).

As shown in Figure 9-4, the INITIALIZE /QUEUE command is used to initialize the logical queue HOLD. This queue is initialized with generic printing disabled. When the queue HOLD is assigned to the printer LPB0 and started, the jobs in the queue HOLD are printed on the line printer LPB0 by way of the physical queue LPB0.

COMMANDS

```
$ SET DEVICE/SPOOLED  LPAO
$ SET DEVICE/SPOOLED  LPBO
$ SET DEVICE/SPOOLED=LPCO  LPCO
$ INITIALIZE/QUEUE/FLAG/GENERIC  SYS$PRINT
$ START/QUEUE  SYS$PRINT
$ INITIALIZE/QUEUE/FLAG  LPAO
$ START/QUEUE  LPAO
$ INITIALIZE/QUEUE/FLAG  LPBO
$ START/QUEUE  LPBO
$ INITIALIZE/QUEUE/FLAG/NOENABLE_GENERIC_PRINTING  LPCO
$ START/QUEUE  LPCO
$ INITIALIZE/QUEUE/FLAG/NOENABLE_GENERIC_PRINTING  HOLD
$ ASSIGN/QUEUE  LPBO  HOLD
$ START/QUEUE  HOLD
```

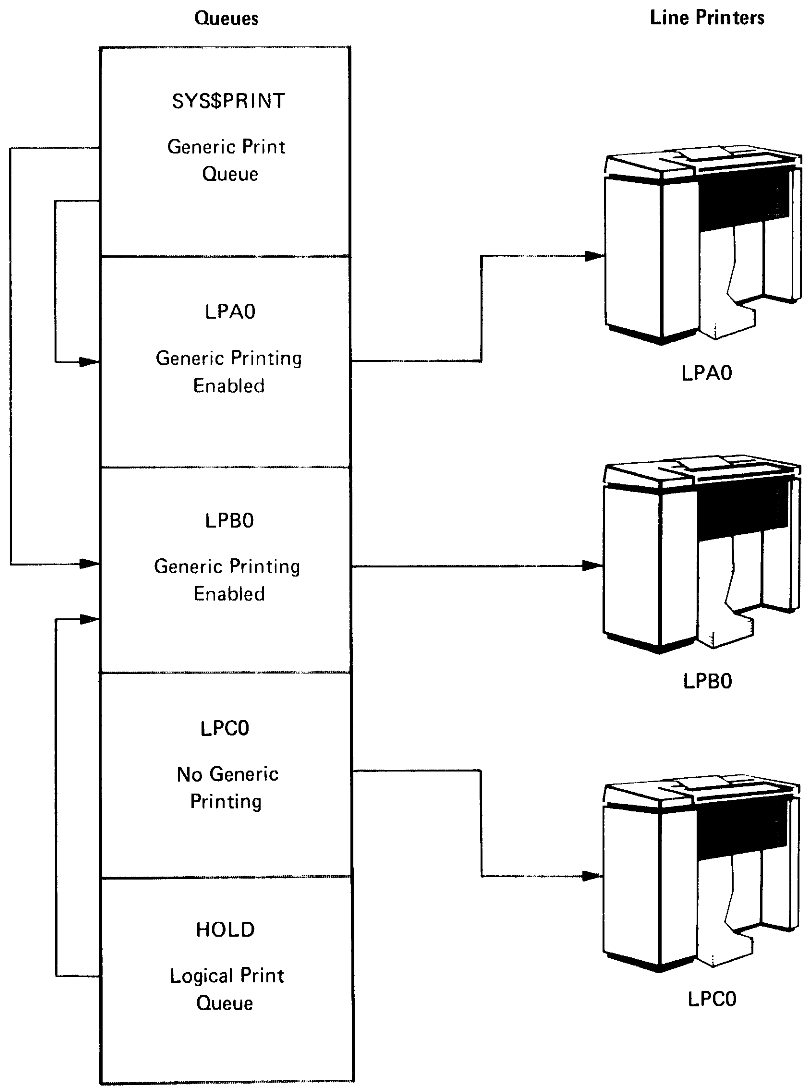

Figure 9-4 Setting Up Spooled Printers and Print Queues -- Adding a Logical Queue to the System with Three Line Printers

9.4 TERMINAL QUEUES

Terminal queues are print queues destined for terminal devices (which are probably being used as remote printers, never interactively). They are created and controlled by use of the same commands that are used in creating and controlling regular print queues. Print jobs in generic print queues are not dequeued to a terminal device queue unless the generic queue is initialized or started with the /TERMINAL qualifier.

CHAPTER 10

ERRORS AND OTHER SYSTEM EVENTS

The system provides several tools for recording and reporting errors and other system events. These tools include facilities for logging and reporting system events, logging operator messages, and reporting problems to DIGITAL. The system also provides facilities for dumping itself on a failure and reporting the contents of the dump -- see the <u>VAX/VMS System Dump Analyzer Reference Manual</u>.

10.1 ERROR LOG

The system automatically writes messages to the latest version of a file named SYS$SYSDISK:[SYSERR]ERRLOG.SYS as the following events occur:

- Errors -- Device errors, machine checks, bus errors, synchronous backplane interconnect (SBI) alerts, soft error correcting code (ECC) errors, asynchronous write errors, hard ECC errors

- Configuration changes -- Volume mounts and dismounts

- System events -- Cold start-up, warm start-up, crash start-up, message from Send Message to Error Logger ($SNDERR) system service, time stamp

The system manager reports the information in the error log by running the SYE utility. Since the system continues to log messages to ERRLOG.SYS and creates a new version of the file if the current one is locked, the user should rename the file before running SYE against it. Recommended as the new name is ERRLOG.OLD since SYE uses this name as a default.

10.1.1 SYE Utility

The system manager runs SYE to selectively report the contents of an error log file. The reports are as follows:

- Roll-up -- Reports totals in each category of errors.

- Cryptic -- For device errors, reports the contents of the device registers at the time of each error. The figures are in hexadecimal without explanation.

- Brief -- Briefly reports the nature of each error, configuration change, and system event.

- Standard -- Fully reports the nature of each error, configuration change, and system event.

SYE

The reports, which are 80 columns wide, can be displayed at the terminal.

The following command invokes the SYE utility:

```
$ RUN SYS$SYSTEM:SYE
```

SYE displays a series of prompts to which the system manager responds with a value or, to take the default, a carriage return:

```
                   Prompt                           Valid Response

_INPUT FILE   [ [SYSERR]ERRLOG.OLD] ?     input-file-spec
_OUTPUT FILE  [SYS$OUTPUT]          ?     output-file-spec
_OPTIONS      [ROLL-UP]             ?     report-option
_DEVICE NAME  [ <CR> ]              ?     category-type
_AFTER DATE   [FIRST ENTRY]         ?     after-date-spec
_BEFORE DATE  [LAST ENTRY]          ?     before-date-spec
```

input-file-spec
: Input log file; defaults to the highest version of SYS$SYSDISK:[SYSERR]ERRLOG.OLD; omitted fields default to those for VAX-11 FORTRAN unit 1 -- the highest version of SYS$SYSDISK:[default-directory]FOR001.DAT (SYE is written in FORTRAN)

output-file-spec
: Output report file; defaults to SYS$OUTPUT, which is usually the user's terminal; omitted fields default to those for VAX-11 FORTRAN unit 2 -- the highest version of SYS$SYSDISK:[default-directory]FOR002.DAT

report-option
: R (roll-up), C (cryptic), B (brief), or S (standard); defaults to R

category-type
: CP (errors other than for devices), CO (configuration changes), SY (system events), or the name of a device (errors and configuration changes on a specific device); defaults to all category types; a device name can be generic to specify a set of devices (for example, DB to specify all devices whose names begin with DB); the colon need not be included on the device name; a minus sign can precede the category type to indicate all category types except the specified one (for example, -DB1 to specify all categories except the device whose name is DB1)

after-date-spec
: Time on error log after which reporting begins, in absolute date/time format; defaults to beginning of log

before-date-spec
: Time on error log before which reporting ends, in absolute date/time format; defaults to end of log

SYE does not issue error messages on incorrect input, but simply reprompts. If an error message appears in the report, the system manager should rerun SYE to eliminate the error. Error messages are reported as VAX-11 FORTRAN messages.

Table 13-2 (Chapter 13) lists and describes the valid device names.

10.1.2 Operations

The error logging facility consists of three parts:

- A set of executive routines that detects errors and events and writes relevant information into error log buffers in memory

- A process called ERRFMT that periodically empties the error log buffers, transforms the descriptions of the errors into standard formats, and stores the formatted information in a file on the system disk

- The SYE utility

The executive routines and the ERRFMT process operate continuously without user intervention. The routines fill up the error log buffers in memory with raw data on every detected error and event. When one of the available buffers becomes full, or when a time allotment expires, ERRFMT automatically writes the buffers to ERRLOG.SYS. Sometimes a burst of errors can cause the buffers to fill up before ERRFMT can empty them. In this case, the system merely assigns a sequence number to the errors and events that occur, without preserving further information. As soon as ERRFMT frees the buffer space, the executive routines resume preserving error information in the buffers.

The ERRFMT process displays an error message and deletes itself if it encounters excessive errors while writing the error log file. The system manager can restart ERRFMT by invoking the ERFSTART command procedure in the system manager's directory:

```
$ SET DEFAULT SYS$SYSDISK:[SYSMGR]
$ @ERFSTART
```

The command procedure must be invoked from the system manager's account (UIC [1,4]).

10.1.2.1 **Using Error Reports** - The error reports generated by SYE are useful tools in two basic ways:

- Reports aid preventive maintenance by identifying areas within the system that show potential for failure.

- Reports speed the diagnosis of a failure by documenting the errors and events that led up to the failure.

The detailed contents of the reports are most meaningful to DIGITAL field service personnel. However, the system manager can use the reports as an important indicator of the system's reliability. For example, when a report shows that a particular device is producing a relatively high number of errors, the system manager can consult DIGITAL field service. By running a diagnostic program to investigate the device, field service can attempt to isolate the source of the errors. Once identified, the source of the errors can possibly be eliminated and a failure averted.

In the event that a system component does fail, a field service representative can study error reports of system activity leading up to and including the failure. For example, if a device fails, the system manager can generate error reports immediately after the failure. One report might describe in detail all the errors associated with the failed device and occurring within the last 24 hours; another report might summarize all types of errors that occurred within the same time period. The summary report can put the device errors into a system-wide context. The field service representative can then run the appropriate diagnostic program for a thorough analysis of the failed device. Using the combined error logging and diagnostic information, the field service representative can proceed to correct the device.

The information made available by the error logging facility is essential to efficient maintenance of a VAX/VMS system. Error reports allow the system manager to track system performance and to anticipate failures. In turn, field service personnel rely on the reports as an aid to both preventive and corrective maintenance. Overall, effective use of the error logging facility, in conjunction with diagnostic programs, can significantly reduce the amount of system downtime.

Because the file ERRLOG.SYS can be renamed and the renamed error log file can then be copied to a removable volume, error reports can be generated either at the site where the errors occurred or at any other VAX/VMS installation. For example, a field service representative can rename and copy the error log file to take back to the field service office, where another VAX/VMS system can be used to generate error reports. Alternatively, a system manager can rename and copy to a disk file a version of the error log file that covers a crucial period of system activity. When a field service representative arrives on site, he or she can generate one or more reports from the copied file as well as from the current version of ERRLOG.SYS.

10.1.2.2 **Maintaining the Error Log Files** - While SYE is accessing ERRLOG.SYS, ERRFMT cannot write any error information into it. Therefore, if SYE is accessing the highest version of ERRLOG.SYS when ERRFMT needs to log an error, ERRFMT creates a new version of the file. The new version picks up logging errors where the previous version left off. All the versions of the ERRLOG.SYS file remain on the system disk until a user explicitly manipulates them in some way.

The fewer the log files, the simpler and more efficient it is to generate log reports. The system manager or operator can take steps to minimize or control the number of versions created.

For example, when generating several reports from the current error log file, a user should first rename the error log file to ERRLOG.OLD and then use the renamed file as input to SYE. In this way, only one new error log file is created, and SYE does not prevent ERRFMT from accessing the new file. In addition, the user ensures that SYE is accessing the same error log file for each report.

A responsibility of the system operator or system manager is to devise a plan for maintaining the versions of the error log file. One way to do this is to rename the highest version of ERRLOG.SYS on a daily basis. This action causes a new error log file to be created and allows the old file (which was renamed) to be copied to a back-up volume where it can be kept as long as needed. For example, an operator could rename the current copy of ERRLOG.SYS every morning at 9:00 to ERRLOG.OLD. To free space on the system disk, the operator could then back up the renamed version of the error log file on a different volume and delete the renamed file from the system disk. Note that caution should be taken to ensure that error log files are not deleted inadvertently.

10.2 OPERATOR'S LOG FILE

The operator's log file (SYS$SYSDISK:[SYSMGR]OPERATOR.LOG) is a system management tool that is useful in anticipating and preventing hardware and software failures. By regularly examining the operator's log file, the system manager can often detect tendencies, or trends, toward failures and can thereby see that corrective action is taken before these failures occur.

The system operator should, therefore, print out copies of the operator's log file regularly, and the system manager should retain these copies for reference. See the VAX/VMS Operator's Guide for descriptions of the messages that appear in this file and for instructions on printing copies of it. Figure 10-1 shows some typical messages of the operator's log file.

```
Opcom, 13-APR-1978 20:09:43.07, Logfile initialized, operator=_OPA0:
Opcom, 06:57:53.70, Device offline, LPA0:
Opcom, 06:58:25.70, Device offline, LPA0:
Opcom, 06:58:57.70, Device offline, LPA0:
Opcom, 06:59:29.70, Device offline, LPA0:
Opcom, 07:00:01.70, Device offline, LPA0:
Opcom, 07:00:33.70, Device offline, LPA0:
Opcom, 07:01:05.70, Device offline, LPA0:

Opcom, 11:30:47.70, Device offline, LPA0:
Opcom, 11:31:19.70, Device offline, LPA0:
Opcom, 11:31:51.70, Device offline, LPA0:
Opcom, 14-APR-1978 13:59:30.27, Terminal enabled, operator=_TTC3:
    Opcom, 13:59:41.88, ROGERP        Accnt=VMS
Opcom, TTC3:, 'TEST

Opcom, 15:26:42.73, Device offline, CRA0:
```

Figure 10-1 Operator's Log File

10.3 REPORTING SOFTWARE PROBLEMS

To inform DIGITAL about problems with the VAX/VMS operating system or about errors in VAX/VMS software documents, the system manager should use the Software Performance Report (SPR), which is illustrated in Figure 10-2. Complete directions for completing the SPR form accompany the form itself.

A supply of SPR forms is included in the VAX/VMS software distribution kit; more forms can be obtained from an SPR center. The addresses of these centers are listed on the backs of the forms.

On a fatal bugcheck, the system manager should (if possible) copy the file SYS$SYSTEM:SYSDUMP.DMP to tape, and send the tape to DIGITAL with the SPR.

digital

SOFTWARE PERFORMANCE REPORT

FIELD NO.: | SPR NO.:

SOFTWARE SERVICES NO.:

229433

PAGE ______ OF ______

OPERATING SYSTEM	VERSION	SYSTEM PROGRAM OR DOCUMENT TITLE	VERSION OR DOCUMENT PART NO.	DATE

(SEE EXAMPLE IN INSTRUCTIONS)

NAME:

FIRM:

ADDRESS:

DEC OFFICE

DO YOU HAVE SOURCES? YES ☐ NO ☐

REPORT TYPE/PRIORITY

☐ PROBLEM/ERROR
☐ SUGGESTED ENHANCEMENT
☐ OTHER

☐ 5.
☐ 4.
☐ 3.
☐ 2.
☐ 1.

SUBMITTED BY: PHONE:

CAN THE PROBLEM BE REPRODUCED AT WILL? YES ☐ NO ☐

ATTACHMENTS

MAG TAPE ☐ FLOPPY DISKS ☐ LISTING ☐
DECTAPE ☐ OTHER ______

COULD THIS SPR HAVE BEEN PREVENTED BY BETTER OR MORE DOCUMENTATION? PLEASE EXPLAIN IN PROVIDED SPACE BELOW. YES ☐ NO ☐

CPU TYPE	SERIAL NO.	MEMORY SIZE	DISTRIBUTION MEDIUM	SYSTEM DEVICE	DO NOT PUBLISH ☐

ALL SUBMISSIONS BECOME THE PROPERTY OF DIGITAL EQUIPMENT CORPORATION

DATE RECEIVED	DATE TO MAINTAINER	DATE LOGGED OFF
LOGGED ON	DATE RECEIVED FROM MAINTAINER	DATE ANSWER SENT

EN-1044G-07 (35C) R0878

ADMINISTRATIVE SERVICES GROUP, SWS

Figure 10-2 Software Performance Report (SPR)

PART III

SYSTEM CONFIGURATION

The system manager is responsible for maintaining an adequate system configuration by:

- Generating a system that meets the user's resources and workload
- Monitoring system activity to determine adequate performance levels
- Adjusting system parameters and other values to improve performance

Part III consists of Chapters 11 through 14:

- Chapter 11: Tuning Considerations -- Discusses the factors involved in generating, monitoring, and adjusting the system for proper performance
- Chapter 12: System Parameters -- Defines the system parameters and the standard system parameter files supplied by DIGITAL
- Chapter 13: System Generation -- Provides operating instructions for the SYSGEN utility
- Chapter 14: DISPLAY Utility -- Provides operating instructions for the DISPLAY utility and describes the displays

CHAPTER 11

TUNING CONSIDERATIONS

Hardware resources -- mainly physical memory and secondary storage -- constitute the primary constraint on system performance. Adequate hardware resources for the workload generally provide adequate performance with little need for tuning, although a certain amount of tuning will enhance system performance. Inadequate hardware resources for the workload generally provide inadequate performance regardless of the tuning effort. Only in the middle ground of just-adequate or just-inadequate resources does tuning become a significant factor. This resource level normally manifests itself in situations where the user is trying to make do with a small system, or where the user's workload has been increasing over a period of time with no corresponding addition of resources. For practical purposes, tunable general-purpose systems can be considered in two categories:

- Small systems -- Small in this context means a system with tight resources. Normally, this would be a system of 1MB or less, although conceivably a large system with an enormous workload might fit this category. The primary emphasis in tuning a small system lies in optimizing the use of physical memory and achieving a balance between the use of physical memory and disk I/O.

- Large systems -- Large in this context means a system with plentiful resources for the workload. Normally, this would be a system with physical memory in excess of 1MB, although conceivably a small system with a light workload might fit this category. The primary emphasis in tuning a large system is on decreasing disk I/O.

Before undertaking a major tuning effort, the manager of a small system should weigh the time and effort involved in the venture against the alternative solution: the purchase of additional physical memory or faster disks.

11.1 PRETUNING CONSIDERATIONS

Before starting the tuning effort, the system manager should ensure that hardware resources are adequate for the workload, that the workload is distributed as equably as possible, and that frequently used code is shared.

11.1.1 Hardware Resources

VAX/VMS hardware configurations are designed to handle the following workloads:

Workload	Memory	Disk Storage (Capacity) (Peak Transfer Rate)
1-8 users	512KB	2 RK07 (56MB) (538KB per second)
9-16 users	512KB	2 RM03 (134MB) (1200KB per second)
17-32 users	1024KB	2 RP06 (352MB) (806KB per second)
33-48 users	1536KB	2-4 RP06 (352-704MB) (806KB per second)
49-64 users	2048KB	2-4 RP06 (352-704MB) (806KB per second)

These figures are approximate. The precise number of users for a memory configuration depends on the size and make-up of the programs those users are running. If the processing requirements of all users consist of editing (with an SOS-type editor), compiling, linking, and running small programs, the system will probably support around the maximum number of users -- 32 on the 1MB system, for example. If, on the other hand, processing consists mainly of developing very large programs, sorting, and/or using keypad-type editors, the system will probably support around the minimum number of users. For user production programs, the system manager can lean toward the maximum figure for programs that contain little data or use VAX-11 RMS to manipulate data on a per-record basis, and toward the minimum figure for programs that contain very large data structures. The trend for native VAX/VMS software products (assemblers, compilers, and so on) is to require larger amounts of memory, mainly through the use of larger data structures to reduce I/O activity (resulting in shorter response times and faster throughput).

The system manager can add physical memory up to a total of 8MB of local memory plus 8MB of multiport memory. Adding memory will most certainly enhance system performance (as discussed below under "Large Systems"), but will not necessarily permit more than 64 concurrent users. Depending on the workload, the system manager should expect to overload the processor in the range of 64 to 128 users.

Actual disk capacity is a function of user file requirements. However, the system should ideally contain at least two disk drives and two disk drive controllers. The system manager should make an effort to place the disk drives containing the execution files (system and user image files, paging file, swapping file, and dump file) and the disk drives containing user data on separate controllers. This procedure will cut down on the contention between system I/O and user I/O activities.

11.1.2 Workload Distribution

The system manager is reminded (although the subject may seem elementary) to distribute the workload as evenly as possible over the time the computer is running. While scheduling interactive users evenly is normally not possible due to the weight of convention on standard working and sleeping hours, either or both of the following techniques are workable:

- Run large jobs as batch jobs -- Force the submission of large jobs on a batch basis; regulate the number of batch streams so that batch usage is high when interactive usage is low. Chapter 9 discusses batch jobs.

- Restrict system use -- Do not permit more users to log in at one time than the system can support with an adequate response time. The system manager can restrict the number of interactive users with the IJOBLIM system parameter (see Chapter 12) and the DCL command SET LOGINS (see the VAX/VMS Operator's Guide).

11.1.3 Sharing Code

The site-independent start-up command procedure creates permanent global sections for system programs and subroutines by installing them as known images with the shared attribute. The system manager should add to these, in the site-specific start-up command procedure, user programs and subroutines that have reached a production status or are in general use. Users should, of course, be encouraged to write shareable code. Chapter 8 of this manual covers start-up command procedures, while Chapter 7 discusses known images.

11.2 BACKGROUND DISCUSSION

Figure 11-1 illustrates the physical memory and secondary storage requirements for supporting process contexts on a VAX/VMS system.

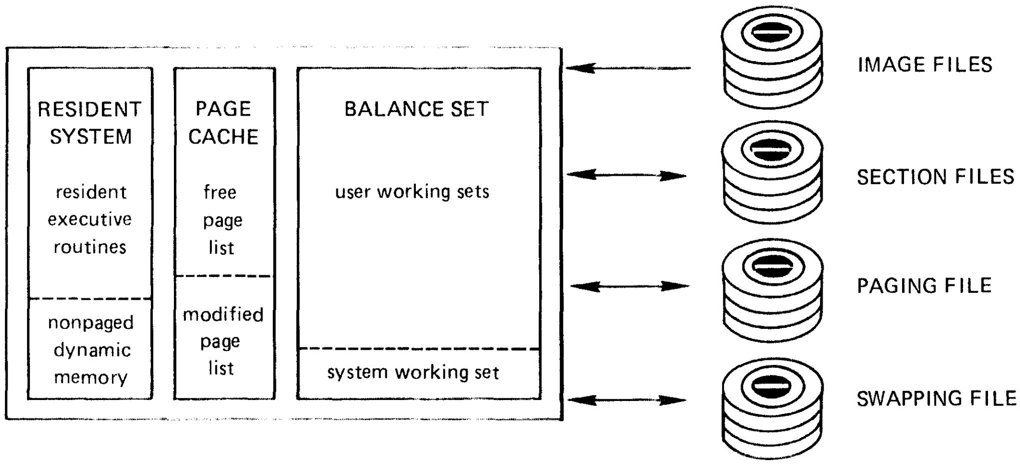

Figure 11-1 VAX/VMS Memory Configuration

Each active process is associated with a working set that contains the pages of code and data being used in the execution of that process. Initially, the working set expands to accommodate clusters of image and section (that is, code or data explicitly mapped by a process to its virtual address space) pages from their disk files as they are referenced. The size of the working set (unless altered by certain system services) equals the cumulative number of pages brought in from the image and section files, until the working set limit is reached. At that point, newly referenced pages are accommodated by pushing current pages out of the working set, on a first-in, first-out (FIFO) basis.

The system automatically adjusts the working set limit by deducting pages if the rate of page faults is low, and adding pages if the rate of page faults is high (unless the system manager turns off the automatic working set adjustment feature). A page fault occurs whenever a referenced page does not exist in the working set.

If sufficient space exists, working sets reside in real memory. If sufficient space does not exist, one or more of the working sets must reside on the swapping file. The system determines which working sets to retain in physical memory and which to outswap to disk according to each process's state and priority, and the quantum:

- Current state -- The currently executing process resides in physical memory.
- Compute state -- Processes that can execute immediately if given processor time should reside in physical memory; they are the last candidates for outswapping.
- Waiting state -- Processes that are waiting for a local event flag, are hibernating, or are suspended constitute the primary candidates for outswapping.
- Priority -- Among processes in the same state, those with the lowest priorities are outswapped.
- Quantum -- A process cannot be outswapped until it has received services for at least one SYSGEN-specified period of time called a quantum.

The system reevaluates physical memory versus disk residency requirements whenever a process's state changes. For example, if a local event flag is posted changing the state of a swapped-out process from wait to compute, the system will (if possible) place the process's working set in physical memory, even if this means outswapping the working set of another process.

One working set is reserved for those portions of the system executive that are pageable. In addition, the system requires user working sets for the system processes ERRFMT, OPCOM, and JOB_CONTROL, the Files-11 and magnetic tape ACPs, and the print symbionts, although these working sets are also swapped.

Pages pushed out of a working set that has reached its limit initially remain in physical memory by going to the page cache: to the free page list if they were not written on, or to the modified page list if they were written on. The free page list expands to accommodate additional pages being pushed out of working sets. The free page list contracts as pages are allocated to processes requesting additional memory; pages leave the free list on a FIFO basis. However, the system cannot reduce the size of the free page list below a SYSGEN-specified lower limit.

The modified page list expands to accommodate additional pages being pushed out of working sets, but only to a SYSGEN-specified upper limit. At that point, pages are written back to their section files or (in the case of demand-zero and copy-on-reference pages) to the paging file, then moved to the free list, until the size of the modified page list drops below a SYSGEN-specified lower limit. If a page that was pushed out of a working set is referred to again, a page fault occurs. The system must return the page to the working set from either:

- The page cache -- If the page is still in physical memory (that is, on the free page list or the modified page list), a "cheap" page fault occurs as no disk I/O is required.
- An image, section, or paging file -- If the page has been overwritten, an "expensive" page fault occurs as the page must be reread from disk.

Although pages move through the working sets and page cache on a FIFO basis, the return of referred-to-again pages from the page cache to the working sets produces a net effect of deleting the least frequently used pages and retaining the most frequently used pages.

When an image terminates, all section pages that were modified are written back to the section files. All pages associated with the program in physical memory are released. (Shared pages, however, are not released if currently in use by another process.)

11.3 TOOLS

The system manager controls system performance overall with the system parameters, and on a per-user basis with the entries in the UAF. Certain DCL commands also aid in improving system performance. The effects of tuning can be monitored with the DISPLAY utility.

11.3.1 System Parameters

In lieu of the standard system parameter file appropriate for the hardware configuration (as shown in Table 12-7, Chapter 12), the system manager can create a tailored system parameter file with the SYSGEN utility. In tailoring a system parameter file for performance, the system manager should concentrate on the following parameters:

- WSMAX -- Maximum working set limit for all users
- PFRATH, PFRATL, WSINC, WSDEC, AWSMIN, AWSMAX, and AWSTIME -- Settings for automatic adjustment of working set limits
- QUANTUM -- Minimum amount of time in the balance set a process will receive before being outswapped; also, maximum amount of service a process will receive before losing control in a round-robin situation
- SWPRATE -- Interval between inswaps of compute-bound processes
- FREELIM -- Minimum number of pages that must be on the free page list
- PFCDEFAULT -- Number of pages read into a working set during a paging operation involving disk I/O
- MPW_HILIM, MPW_LOLIMIT, and MPW_WRTCLUSTER -- Point at which (expressed in number of pages) pages will be written from the modified page list to the paging file, point at which the writing will stop, and the number of pages written in each I/O operation
- NPAGEDYN and IRPCOUNT -- Size of nonpaged dynamic pool
- SYSMWCNT -- Number of pages in system working set
- RMS_DFMBC -- Blocking factor for VAX-11 RMS I/O operations
- ACP_MAPCACHE, ACP_HDRCACHE, ACP_DIRCACHE, ACP_FIDCACHE, ACP_EXTCACHE, ACP_QUOCACHE, ACP_WRITEBACK, ACP_SYSACC, and ACP_WINDOW -- Space used by Files-11 ACP processes for data structures containing directory and file information

- ACP_MULTIPLE, ACP_SHARE, ACP_SWAPFLAGS, and ACP_WORKSET -- Characteristics of ACP processes: whether concurrent I/O operations use the same or separate copies of the same ACP (ACPs are single threaded); whether concurrent ACPs share code; whether ACP working sets are locked in memory or are swapped; the working set limit for ACP processes

Chapter 12 explains these and the other system parameters. Chapter 13 explains how to use the SYSGEN utility to modify the parameters. Although some of the other system parameters affect performance, they should be considered only in a fine-tuning effort. The system manager should leave them at the standard settings, or reset them to reasonable values dictated by other factors.

11.3.2 User Authorization File

The system manager can affect performance on a per-user basis by resetting the following UAF entries:

- WSQUOTA -- Maximum working set limit for the user's processes; cannot exceed the system parameter WSMAX
- WSDEFAULT -- Working set limit for the user's processes; cannot exceed WSQUOTA
- PRIORITY -- Base priority for the user's processes
- CPUTIME -- CPU time permitted the user's processes

Chapter 2 explains these and the other UAF entries. The other UAF entries do not affect system performance.

11.3.3 DCL Commands

The following DCL commands affect system performance:

Commands	Function
SET WORKING_SET /QUOTA RUN /MAXIMUM_WORKING_SET	Reset working set quota for the user's processes; cannot exceed UAF quota
SET WORKING_SET /LIMIT RUN /WORKING_SET	Reset the working set limit for the user's processes; cannot exceed current quota
INITIALIZE QUEUE /BATCH /WSQUOTA* START QUEUE /BATCH /WSQUOTA* SUBMIT/WSQUOTA $JOB /WSQUOTA	Set working set quota for jobs submitted to a batch queue
INITIALIZE QUEUE /BATCH /WSDEFAULT* START QUEUE /BATCH /WSDEFAULT* SUBMIT /WSDEFAULT $JOB /WSDEFAULT	Set working set limit for jobs submitted to a batch queue

Commands	Function
INITIALIZE QUEUE /BATCH /CPUMAXIMUM /CPUDEFAULT* START QUEUE /BATCH /CPUMAXIMUM /CPUDEFAULT* SUBMIT /CPUTIME $JOB /CPUTIME	Set the CPU time permitted batch jobs
SET PROCESS /[NO]SWAPPING	Enables or disables the swapping in and out of memory of a process's working set
SET /PRIORITY RUN /PRIORITY	Reset the base priority of a process
SET RMS_DEFAULT /BLOCK_COUNT	Resets the blocking factor for VAX-11 RMS I/O operations

The VAX/VMS Command Language User's Guide and the VAX/VMS Operator's Guide (those commands with an asterisk) explain the DCL commands.

11.3.4 DISPLAY Utility

With the DISPLAY utility, the system manager can monitor activities indicative of system performance. Useful displays in this regard include:

- VAX/VMS PROCESSES (also called the USERS display) -- For each current process, lists the current state and priority of the process, the physical memory being used in terms of shareable pages and total pages, the cumulative direct I/O operations, the cumulative page faults, and the cumulative CPU time
- I/O SYSTEM RATES -- Provides the sizes of the free and modified page lists, plus the following information, expressed in cumulative units per interval, per second during the preceding interval, and per second since the start of the display: direct I/O operations, buffered I/O operations, page faults, pages read, read I/O operations, pages written, write I/O operations, and total pages inswapped
- PAGE MANAGEMENT STATISTICS -- Substantially the same as I/O SYSTEM RATES, plus the page faults accumulated by the system working set
- TIME IN PROCESSOR MODES -- Percentage of time spent in each processor mode, plus idle time
- NONPAGED POOL STATISTICS -- Amount of unused space in the nonpaged dynamic pool
- NUMBER OF PROCESSES IN SCHEDULER STATES -- Provides a snapshot of the state of the system

Chapter 14 provides operating instructions and detailed descriptions for the DISPLAY utility, and examples of its use.

The following DCL commands complement the DISPLAY utility:

- SHOW SYSTEM -- Displays information on the processes currently executing, including state, priority, elapsed CPU time, number of page faults, and physical memory occupied
- SHOW PROCESS /ACCOUNTING -- Displays information about the executing process, including the cumulative number of page faults, the peak working set size, the peak virtual size, elapsed total time, and elapsed CPU time
- SHOW STATUS -- Displays information about the executing process, including the working set limit, the amount of physical memory being used, the elapsed CPU time, and the number of page faults
- SHOW WORKING_SET -- Displays a process's current limit and quota, and authorized quota
- LOGOUT /FULL -- Displays information about the terminating process, including the number of page faults, the peak working set size, the peak virtual size, elapsed total time, and elapsed CPU time

The VAX/VMS Command Language User's Guide provides full explanations of these commands.

11.4 SMALL SYSTEMS

As the background discussion indicates, the factors affecting efficient performance are interrelated and somewhat complex. The major factors for small systems include working set size, page cache characteristics, priority, quantum, compute swapping rate, and system use of the system.

11.4.1 Working Set Size

Decreasing working set limits increases the number of working sets that can occupy real memory at one time, reducing the need for swapping, but increasing the need (in most cases) for paging. Normally, excessive paging is preferable to excessive swapping for the following reasons:

- Swapping involves the output and input of an entire working set, while paging usually involves a subset of the working set.
- Paging does not necessarily involve disk I/O operations due to the page cache. In fact, the ratio of disk I/O operations to page faults may be quite low.

A process can, then, show larger and larger numbers of page faults with little effect upon performance. Nevertheless, a point does occur where the reduction of the working set limit significantly cramps a process's performance. The optimal working set limit, then, would be just above the point where performance drops sharply. A 2-phase strategy is recommended for setting working sets at their optimal sizes:

1. Figure initial working set limits for different types of processing on a rule-of-thumb basis.

2. Adjust working set limits based on observed behavior.

11.4.1.1 **Initial Working Set Limits** - For processing involving system components, the following working set limits are suggested:

- Small (60 to 200 pages) -- For editing, and for compiling and linking small programs ("typical" interactive processing)

- Large (200 to 500 pages) -- For compiling and linking large programs, and for sorting ("typical" batch processing)

Working set limits for user programs depend on the code-to-data ratio of the program and the amount of data in the program. Programs that are mostly code -- that include a small or moderate amount of data, or use VAX-11 RMS to process data on a per-record basis -- should require only a small working set. The amount of code should not matter as long as it is reasonably linear. Programs that manipulate large amounts of data internally require large working sets.

The following guidelines are suggested for initially setting working sets limits:

- System generation parameters -- Set WSMAX at the highest number of pages required by any program. At the "typical" installation where use of the system ranges from small interactive tasks to large batch streams, the initial setting for WSMAX would be in the range of 200 to 500. (However, WSMAX has other effects on the system -- see Section 12.1.8.)

- UAF options -- For each user, set WSQUOTA at the largest number of pages required by a program that the user will run interactively. Set WSDEFAULT at the median number of pages required by a program that the user will run interactively.

- Batch queues -- For each batch queue, set WSQUOTA (using the INITIALIZE or START command) at the largest number of pages required by a job that a user will submit. Set WSDEFAULT at the median number of pages required by a job that a user will submit.

This arrangement effectively forces users to submit large jobs for batch processing, as the jobs will not run efficiently interactively. The system manager can further restrict the user who attempts to run a large job interactively by imposing CPU time limits in the UAF.

11.4.1.2 **Adjustments** - The system automatically adjusts working set limits based on observed behavior and the values of the PFRATL, PFRATH, WSINC, WSDEC, AWSMIN, AWSMAX, and AWSTIME system parameters. The example below demonstrates the resultant working set limits as the system automatically adjusts a working set with a default limit of 100 pages and system parameter values of PFRATL=2, PFRATH=512, WSINC=21, and WSDEC=5:

Interval	Page Fault Rate	Adjustment	Working Set Limit
0	-	-	100
1	0	Deduct 5 pages	95
2	0	Deduct 5 pages	90
3	1	Deduct 5 pages	85
4	0	Deduct 5 pages	80
5	2	Deduct 5 pages	75
6	17	None	75
7	189	None	75
8	692	Add 21 pages	96
9	1147	Add 21 pages	117
10	1538	Add 21 pages	138
11	953	Add 21 pages	159
12	492	None	159

WSINC is a much larger value than WSDEC because quickly increasing the working set limit may be critical to performance (for example, if a new module is being referenced), while gradually lowering the limit from the last increase tends to fine-tune upward adjustments.

The system manager can monitor system performance and change system parameter values in the following instances:

- Initial limit -- If the system typically increases or decreases a working set by a significant amount immediately after its process starts, the default working set limit for affected users and batch streams should be changed accordingly. For example, if a user's working set limit is 100, but its processes typically show a working set limit of 50 to 60 for the first few minutes of processing, the user's default working set value in the UAF should be decreased to 60.

- Excessive paging I/O activity -- If paging appears to be generating excessive I/O activity, the system manager might decrease the value of PFRATH to make the working set limit more sensitive to paging activity. Additionally or alternatively, the system manager might decrease the value of PFRATL, increase the value of WSINC, or decrease the value of WSDEC.

- Excessive swapping I/O activity -- If swapping I/O operations appear high, the system manager might increase the value of PFRATH to make the working set less sensitive to paging activity. Working sets will tend to remain at smaller sizes, hopefully increasing the number of working sets that will fit in the balance set. Additionally or alternatively, the system manager might increase the value of PFRATL, decrease the value of WSINC, or increase the value of WSDEC.

The system manager should ensure that the values for WSDEC and WSINC are prime in relation to one another. Otherwise, working set limits will be restricted to a subset of integers. For example, if WSINC is set to 27 and WSDEC is set to 6, working set limits will always be multiples of 3 above and below the default working set limit.

Monitoring aids for adjusting working set limits include:

- VAX/VMS PROCESSES Display -- The size column (which displays shareable/total pages being used by each process) indicates what portion of its working set each process requires. The state column pinpoints processes having scheduling problems: a repeated state of PAGE FAULT WAIT, or PFW, may mean that the working set is too small; repeated states of COMPUTE (OUT OF BALANCE SET), or COMO, by several processes usually means that there are too many users for the available resources. See Section 14.13.

- I/O SYSTEM RATES Display -- As working set sizes are decreased, the swapping rate should decrease. As working set sizes are increased, the swapping rate should increase. A very high swapping rate usually means that there are too many users for available resources. See Section 14.7.

- SHOW STATUS DCL Command -- To obtain a per-image page fault rate, execute the command SHOW STATUS before and after running the image (preferably as a command procedure). Subtract the before statistics from the after statistics to obtain the net page faults and CPU time. Divide the page faults by the CPU time to obtain the page fault rate during execution of the image.

The system manager can inhibit automatic working set adjustments for a process by setting the user's working set limit and default equal. Setting the system parameter WSINC to a value of zero inhibits automatic working set adjustments for all users.

11.4.2 Page Cache

The page cache, consisting of the free page list and modified page list, recirculates pages pushed out of the working set when they are needed again. Frequently used pages tend to stay in physical memory -- either in the working sets (unless outswapped) or in the page cache -- while less frequently used pages tend to be deleted. Size requirements for the free page list depend mainly on the amount of code being pushed out of the working set. Size requirements for the modified page list depend mainly on the amount of data being pushed out of the working sets. The cluster size for pages being read from disk to the working sets also affects the efficiency of the page cache.

11.4.2.1 **Cache Sizes** - The FREELIM system generation parameter sets the minimum size of the free page list. The list may expand as space permits. The FREELIM values listed in Table 12-7 are normally adequate. A high percentage of page faults resulting in read I/O operations (I/O SYSTEM RATES display) indicates that the value of FREELIM might be increased, while a small percentage indicates that the value of FREELIM is correct or might be smaller. A high swapping rate (I/O SYSTEM RATES display) indicates that the value of FREELIM might be reduced to give more space to the balance set. (However, paging and swapping I/O might be better regulated by adjusting working set limits; see Section 11.4.1.2 above.)

The following system parameters regulate the size of the modified page list:

- MPW_HILIM -- Largest size the modified page list is permitted to reach before pages are written to disk
- MPW_LOLIM -- Point below which writing of pages stops
- MPW_WRTCLUSTER -- Number of pages the system is permitted to write in one I/O operation

The parameter values in Table 12-7 are normally adequate. The system manager can monitor the effects of the the size of the modified page list by watching the write I/O operation and swapping rates on the I/O SYSTEM RATES display. Increasing the upper and lower limits on the modified page list normally lowers the write I/O operation rate, but may increase the swapping rate. The system manager can raise the limits in small increments to determine whether the changes significantly improve the write I/O operation rate without significantly degrading the swapping rate.

The system manager should not assign arbitrarily high values to FREELIM and MPW_LOLIM, as these values reduce the fluid pages in physical memory. The system calculates fluid pages by subtracting the resident system, the system working set, and the minimum size of the page cache from available physical memory. If the value of WSMAX as specified by the system manager exceeds this value, it is decreased.

11.4.2.2 **Cluster Size** - The system generation parameter PFCDEFAULT controls the number of pages being read from disk to the working sets during one I/O operation. The values suggested in Table 12-7 are normally adequate. As a rule of thumb, the system manager can specify a value of 32 where the median working set limit is 100 pages, and a value of 64 or more where the median working set limit exceeds 200 pages.

A potential problem with a large cluster size is that the pages being transferred into a working set that has reached its limit push out an equal number of pages, causing large turn-overs of pages in the system. However, the actual cluster size during any one I/O operation is limited by other factors, such as the size of the image or section being transferred. Only in a few instances do large cluster sizes adversely affect system performance, but these instances are more than made up for by the savings in disk I/O that the larger cluster sizes promote.

For special cases, users can adjust cluster sizes on an individual basis with the PFC parameter of the CLUSTER option in the linker (for pages being read from image and section files, not the paging file).

11.4.3 Priority, Quantum, and Compute-Bound Swapping Rate

Certain workloads may give processes a disproportionate share of service. The problem can be alleviated somewhat by adjustments to base priorities, the quantum, and the swapping rate for compute-bound processes.

11.4.3.1 **Very Large Working Sets** - Running several working sets that are very large for the amount of physical memory often causes serious performance degradation. An example might be two batch streams using (on the average) 350-page working sets in a 512KB system. Both batch streams cannot fit in physical memory at the same time, requiring an inswap and outswap of 350 pages each time one or the other is granted control. The high swapping rate not only slows batch throughput, but increases interactive response time, as these users are also held up by the contention on the system disk.

When this situation exists, the system manager normally notices a high swapping rate (I/O SYSTEM RATES display) and an abundance of processes in the COMO state (VAX/VMS PROCESSES display).

The obvious solution to the above problem is to run one batch stream. However, if two batch streams are essential -- or perhaps the problem revolves around two very large production programs that must run concurrently -- the system manager can improve the situation as follows:

- Priority -- Decreasing the base priorities for the batch streams from 4 to 3 can result in a better interactive response time, but will slow batch throughput even more.

- Quantum -- Increasing the quantum in this situation normally ensures better service all around. The system manager may find that a rather large quantum -- 1, 2, or even 3 seconds -- is in order. The larger quantum means that the large working sets will be swapped less frequently. While interactive users must sometimes wait in blocks of several seconds to be serviced, this may be an improvement over the previous response time.

11.4.3.2 **Compute-Bound Programs** - Contention among compute-bound processes can be alleviated by leaving the quantum at its default setting and increasing the value of the SWPRATE system parameter -- to as much as 3 to 5 seconds. The compute-bound swapping rate sets the minimum interval (in real time, not processor time) between inswaps of compute-bound processes. (The system defines a compute-bound process as one whose current priority is the same as the default base priority.) A high SWPRATE value under such circumstances should reduce swapping I/O on the system.

Real-time processes are not affected by the value of the SWPRATE system parameter and interactive processes rarely are. They will be inswapped and outswapped according to priority and quantum. If a compute-bound process must make way for an interactive process, it will be outswapped even if the time specified by its SWPRATE value has not passed (provided its quantum has passed).

11.4.4 System Requirements

The system manager can tune system requirements for time and space by adjusting the nonpaged dynamic pool, system working set, and VAX-11 RMS buffers.

11.4.4.1 **Nonpaged Dynamic Pool** - System executive code takes up 150 to 200 pages of physical memory. The nonpaged dynamic pool takes up another 100 to 180 pages, depending a great deal on the size of the system. The NPAGEDYN system parameter determines the size of the nonpaged dynamic pool. Chapter 12 explains how to calculate the values of these parameters. Some space can be saved by shaving the calculations closely, but this is not recommended. The system manager should use reasonable figures in the calculations and, if anything, err on the upper side.

During the day-to-day running of the system, the system manager should monitor the NONPAGED POOL STATISTICS display. If a large amount of unused space continually exists in the pool, the system manager can reduce NPAGEDYN. If very little unused space exists in the pool or the pool is fragmented, the system manager should increase NPAGEDYN. The pool should contain 200 to 300 bytes of unused space per user.

Running out of space in the nonpaged dynamic pool degrades service more than using some extra space to ensure that this condition does not occur. If the nonpaged dynamic pool is depleted, processes needing space from the pool will be placed in a miscellaneous resource wait (MWAIT) state until sufficient memory is returned to the pool by other processes or by the completion of I/O services. If many processes enter an MWAIT state, the system manager may be forced to reboot the system.

11.4.4.2 **System Working Set** - The system also requires a working set for the pageable portions of the executive, the paged dynamic pool, VAX-11 RMS, and the resident portion of the system message file. The SYSMWCNT system parameter controls the upper limit (quota) of this working set. Although the system manager usually would not reduce the fluid pages available to the system, SYSMWCNT should be set at a reasonable value. If programs start waiting on the completion of system services, performance degradation is likely to be sharp. During run-time activities, the system automatically decreases the limit if the page fault rate is low, depending on the values of the PFRATL, PFRATH, WSINC, WSDEC, AWSMIN, AWSMAX, and AWSTIME system parameters.

An optimal quota for SYSMWCNT varies depending on how the user programs are using the system. Factors dictating a larger size include high use of VAX-11 RMS (especially ISAM services), a high number of logical names, and extensive placement of record locks. As a rule of thumb, the system manager can start with a value of 120. If performance is poor, the system manager should immediately increase SYSMWCNT. An upper limit of 180 or more may be required. If performance is good, the system manager can try lowering SYSMWCNT.

The system manager can monitor the system page fault rate by observing the rate of system page faults on the PAGE MANAGEMENT STATISTICS display. The system page fault rate should be kept low (one fault per second or less).

11.4.4.3 **VAX-11 RMS Buffers** - Where VAX-11 RMS operations are mainly random accesses of single small records, a process can decrease its VAX-11 RMS multiblock count to 1 with the SET /RMS_DEFAULT command. This action reduces the size of the VAX-11 RMS buffer required in memory. However, the value of the RMS_DEFAULT system parameter should be left at 4.

11.5 LARGE SYSTEMS

The major strategy in tuning a large system lies in trading more memory for less I/O. On an individual level, programs should either maintain larger data structures mapped into memory from section files (in place of per-record processing by means of VAX-11 RMS), or increase the multi-buffer count and specify read-ahead and write-behind processing for VAX-11 RMS operations. The system manager should install as many programs as possible with the shared and header-resident attributes. On a system-wide basis, the system manager should allot as much memory as practical to the Files-11 ACP data structures, primarily the header, directory, map, file identification, extent, and quota caches. The sections that follow provide guidelines for setting values for the ACP system parameters (see also Chapter 12).

The system parameters, UAF entries, and DCL commands affecting working set size, the page cache, priority, the quantum, the compute swapping rate, the nonpaged dynamic pool, the system working set, and the VAX-11 RMS buffers should, for the most part, be left at their standard values. Any changes should favor increasing the amount of physical memory available to system and user processes. In particular, the system manager should provide working set default and quota values, and PFRATL, PFRATH, WSINC, WSDEC, AWSMIN, AWSMAX, and AWSTIME system parameter values that minimize paging I/O, although limit and quota values should not be set arbitrarily or unreasonably high. On a large system, the system manager should be able to minimize paging I/O without increasing swapping I/O. (If swapping I/O does increase, however, the system manager should adjust working set limits as described in Section 11.4.1)

11.5.1 Header Cache

Because a file header, once accessed, is likely to require more accesses as the file continues to be processed, maintaining header blocks (one block per page) in memory can save a significant amount of I/O. The following system parameters control the header cache:

- ACP_WRITEBACK -- This switch should be set to a value of 1 to enable the deferred writing of file header blocks, which inhibitS the write I/O operation for the file header that otherwise occurs each time a file is extended. The header is always written when the file is closed. Headers for relative and indexed files are always written when the file is extended. A risk is taken in that the file may be lost if a header has never been written and the sytem fails. This loss is usually not serious, since the file is probably in an inconsistent state.

- ACP_HDRCACHE -- The number of header blocks should equal or exceed the number of files that are likely to be open concurrently. The system manager can calculate this value as the product of the number of concurrent users and the median number of files each user accesses concurrently. Reasonable values for a system configured for 48 to 64 users (64USER system) fall in the range of 80 to 200.

Even on a small system, the system manager should try to allot extra pages to the header cache, as the expenditure in memory is well worth the I/O operations saved.

11.5.2 Directory Cache

Because a directory block, once accessed, is likely to require more accesses as files continue to be processed, maintaining directory blocks (one block per page) in memory can save a significant amount of I/O.

The ACP_DIRCACHE system parameter specifies the size of the directory cache. As with ACP_HDRCACHE, the system manager should set a value that equals or exceeds the number of directories that are likely to be accessed concurrently.

The size of the directory cache stands next to the size of the header cache in importance in reducing Files-11 ACP I/O activity.

11.5.3 Quota Cache

If disk quotas are being enforced, the system manager should cache one quota entry per active user. The ACP_QUOCACHE system parameter specifies the number of entries cached. One page holds 16 quota entries, so that, for example, a specification of 64 requires 4 pages of storage.

11.5.4 Map, File Identification, and Extent Caches

The system manager should cache a reasonable number of bit map blocks, file identification slots, and extent entries. Suggested values on a system configured for 48 to 64 users (64USER system) are as follows:

- ACP_MAPCACHE -- 20 pages
- ACP_FIDCACHE -- 8 to 16 entries
- ACP_EXTCACHE -- 16 to 32 entries

ACP_EXTLIMIT can be set to a fairly high value (for example, 200 to 400), since free space lost during a system failure is automatically recovered at volume mount time.

11.5.5 System Directory Cache

The ACP_SYSACC parameter specifies the number of directories for which to save access data on a volume mounted with the /SYSTEM qualifier. Its value can be overridden with the /ACCESS qualifier in the DCL command MOUNT. It should be set to the number of directories expected to be in active use on a system volume at one time. Each unit requires 96 bytes of nonpaged pool. However, too low an ACP_SYSACCESS or /ACCESS value partially negates the benefit of the ACP_DIRCACHE parameter.

11.5.6 Multiple ACPs

Multiple ACPs (that is, duplicate Files-11 ACPs) are almost never worthwhile. The memory spent on the extra ACP would be better spent on increasing the cache sizes of the first ACP. The value of ACP_MULTIPLE should be 0 unless all the following conditions exist: physical memory not less than 2MB; disks attached on separate controllers; high direct I/O activity even with large ACP caches.

CHAPTER 12

SYSTEM PARAMETERS

The system parameters configure a system in terms of resource requirements and limits. Values for the parameters exist in several forms:

- Current values -- The system image on disk (SYS$SYSTEM:SYS.EXE) contains a set of system parameter values.

- Active values -- The active system in memory contains a set of parameter values.

- Parameter files -- System parameter values can be stored in specially structured files.

The VAX/VMS software distribution kit contains seven standard system parameter files, each geared toward a particular hardware configuration:

File Name	**Hardware Configuration**
MINIMUM	512KB memory Console terminal Console floppy diskettes Bootstrap disk
8USER	512KB memory 2 RK07 disks 8 DZ11 lines
16USER	768KB memory 2 RM03 disks 16 DZ11 lines
32USER	1024-1536KB memory 2 RP06 disks 32 DZ11 lines
48USER	1536-2048KB memory 2-4 RP06 disks 48 DZ11 lines
64USER	2048-3072KB memory 2-4 RP06 disks 64 DZ11 lines
VIRT32MB	1024KB memory 2 RP06 disks 16 DZ11 lines Support of 32MB virtual address space

The file specification for each file is SYS$SYSTEM:file-name.PAR.

During the bootstrap process, the operator selects either the current values (that is, the values in SYS$SYSTEM:SYS.EXE) or the values in one of the parameter files, and optionally modifies them. The selected values, as modified, become the active values which actually configure the system, and also become the new current values.

During system operation, the system manager can run the SYSGEN utility (Chapter 13) to modify the current values or the values of any parameter file, or create a new parameter file, for use in subsequent bootstrap operations. The SYSGEN utility also enables the system manager to dynamically alter the run-time system configuration by modifying a subset of the active values.

The system parameters fall into nine general categories:

- MAJOR -- Parameters most likely to require modification; a subset of SYS
- SYS -- Parameters that affect overall system operation; includes the major parameters
- JOB -- Job control parameters
- ACP -- Parameters associated with Files-11 ancillary control processes (ACPs)
- RMS -- Parameters associated with VAX-11 RMS
- PQL -- Parameters associated with process creation limits and quotas
- GEN -- Parameters that affect the creation and initialization of data structures at bootstrap time; a subset of SYS, plus all ACP and PQL parameters
- SPECIAL -- Special parameters used by DIGITAL
- DYNAMIC -- Parameters whose active values can be modified; a subset of SYS and ACP, plus all JOB, RMS, and PQL parameters

Each parameter has associated with it default, minimum, and maximum values that define the scope of allowable values. Tables 12-1 through 12-6 briefly describe the parameters and list the default, minimum, and maximum values. (The SPECIAL parameters are not documented; they should be used only by DIGITAL personnel.) Table 12-7 lists the values of the parameters in the standard system parameter files. The remaining sections of the chapter describe the parameters in more detail and provide guidelines for their modification.

Table 12-1
MAJOR Parameters

Parameter Name	Description	Default	Minimum	Maximum[3]
PFCDEFAULT[2]	Default page fault cluster size in pages	16	0	127
GBLSECTIONS[1]	Number of global section descriptors	40	20	-1
GBLPAGES[1]	Number of global page table entries	2048	512	-1
MAXPROCESSCNT[1]	Maximum number of processes	64	12	256
SYSMWCNT[1]	Maximum size of system working set in pages	100	20	16384
BALSETCNT[1]	Maximum number of resident working sets	24	4	1024
IRPCOUNT[1]	Number of preallocated I/O request packets	250	0	32768
WSMAX[1]	Maximum size of any working set in pages	256	60	16384
NPAGEDYN[1]	Size of nonpaged dynamic pool in bytes (but rounded down to an integral number of pages by the system)	64000	16384	-1
PAGEDYN[1]	Size of paged dynamic pool in bytes (but rounded down to an integral number of pages by the system)	24000	8192	-1
VIRTUALPAGECNT[1]	Maximum virtual space per process in pages	8192	512	262144
QUANTUM[2]	Maximum time a process can use at once and minimum service a process must receive before being outswapped, in 10ms units	30	2	32767
MPW_WRTCLUSTER[1]	Number of pages written per I/O from the modified page list	32	0	127
MPW_HILIM	Maximum size of modified page list in pages	64	0	16384
MPW_LOLIM[1]	Minimum size of modified page list in pages	33	0	16384

1. Also a GEN parameter
2. Also a DYNAMIC parameter
3. The value -1 means no limit.

Table 12-2
SYS Parameters
(excluding MAJOR parameters)

Parameter Name	Description	Default	Minimum[3]	Maximum[3]
KFILSTCNT[1]	Number of known file list heads	4	2	-1
PROCSECTCNT[1]	Number of process sections	20	5	1024
MINWSCNT[1]	Minimum number of fluid pages in any working set	20	10	-1
INTSTKPAGES[1]	Size of interrupt stack in pages	2	1	-1
SPTREQ[1]	Number of additional system page table entries	700	-1	-1
PFRATL[2]	Page fault rate low limit in faults per 10 seconds of processor time	2	0	-1
PFRATH[2]	Page fault rate high limit in faults per 10 seconds of processor time	512	0	-1
WSINC[2]	Working set increment in pages	21	0	-1
WSDEC[2]	Working set decrement in pages	5	0	-1
AWSMIN[2]	Automatic working set minimum in pages	80	0	-1
AWSMAX[3]	Automatic working set maximum in pages	460	0	-1
AWSTIME[2]	Automatic working set time for collecting sample in 10ms units	20	1	-1
EXTRACPU[2]	Extra CPU time added to process after CPU time is expired in 10ms units	1000	0	-1
MAXSYSGROUP[2]	Highest system UIC	8	1	32768
MAXBUF[2]	Maximum number of bytes that can be transferred in one buffered I/O	1024	512	64000
DEFMBXBUFQUO[2]	Default mailbox buffer quota in bytes	1024	256	64000
DEFMBXMXMSG[2]	Default mailbox maximum message size in bytes	256	64	64000
DEFMBXNUMMSG[2]	Not implemented	16	1	-1
FREELIM[1]	Lower limit of free page list in bytes	10	0	-1
XFMAXRATE[2]	Maximum rate of transfer for DR32 devices	236	0	255
LAMAPREGS	Number of map registers allocated to an LPA11 device driver	0	0	255

(continued on next page)

1. Also a GEN parameter
2. Also a DYNAMIC parameter
3. The value -1 means no limit.

Table 12-2 (Cont.)
SYS Parameters
(excluding MAJOR parameters)

Parameter Name	Description	Default	Minimum[3]	Maximum[3]
REALTIME_SPTS	Number of system page table entries reserved for connect-to-interrupt processes	0	0	-1
CLISYMTBL1[2]	Size of command interpreter symbol table in bytes	20	10	128
BUGREBOOT[2]	Automatic reboot on fatal bugcheck; switch	1	0	1
CRDENABLE[1]	Detection and logging of memory-corrected read errors; switch	1	0	1
DUMPBUG[2]	Writing of dump file on fatal bugcheck; switch	1	0	1
BUGCHECKFATAL[2]	All bugchecks fatal; switch	0	0	1
POOLPAGING[1]	Paging of paged dynamic pool; switch	1	0	1
SBIERRENABLE[1]	Internal bus error logging; switch	1	0	1
SETTIME[1]	Time-of-day prompt at boot time; switch	0	0	1
SYSPAGING[1]	Paging of executive code; switch	1	0	1
UAFALTERNATE[1]	Use of alternate UAF; switch	0	0	1
TTYSCANDELTA[1]	Terminal dial-up/hang-up scan interval in increments of 100ns	10000000	100000	-1
TTY_SPEED[1]	Default speed for terminal; code	11	0	15
TTY_BUF[1]	Default line width for terminal	80	0	65535
TTY_DEFCHAR[1]	Default terminal characteristics	403706528	0	-1
TTY_TYPAHDSZ[1]	Size of terminal type-ahead buffer in bytes	78	0	-1
TTY_PROT[1]	Default protection for all terminals	0	0	-1
TTY_OWNER[1]	Owner UIC for TTY_PROT specification	0	0	-1

1. Also a GEN parameter
2. Also a DYNAMIC parameter
3. The value -1 means no limit.

Table 12-3
JOB Parameters

Parameter Name	Description	Default	Minimum	Maximum
MAXPRINTSYMB	Maximum number of print symbionts	8	1	255
DEFPRI	Default priority	4	1	31
IJOBLIM	Maximum number of interactive jobs	64	1	1024
BJOBLIM	Maximum number of batch jobs	16	0	1024
NJOBLIM	Maximum number of network jobs	16	0	1024

Table 12-4
ACP Parameters

Parameter Name	Description	Default	Minimum	Maximum[3]
ACP_MULTIPLE[1]	Multiple ACPs per volume; switch	0	0	1
ACP_SHARE	Sharing of ACP code; switch	1	0	1
ACP_MAPCACHE[2]	Size of bit map cache in pages	4	1	-1
ACP_HDRCACHE[2]	Size of file header cache in pages	4	2	-1
ACP_DIRCACHE[2]	Size of directory cache in pages	16	2	-1
ACP_WORKSET[2]	Working set default for ACP	0	0	-1
ACP_FIDCACHE[1]	Size of file identification cache in pages	8	0	-1
ACP_EXTCACHE[1]	Size of extent cache in pages	16	0	-1
ACP_EXTLIMIT[1]	Maximum amount of free space in extent cache in tenths of a percent of available free space	100	0	1000
ACP_QUOCACHE[1]	Number of entries in quota cache	16	0	-1
ACP_SYSACC[1]	Size of directory access cache in pages	8	0	-1
ACP_MAXREAD[2]	Maximum directory blocks to read	32	1	64
ACP_WINDOW[2]	Default number of window pointers	7	1	-1
ACP_WRITEBACK[2]	Caching of file headers; switch	1	0	1
ACP_DATACHECK[2]	Data verification on ACP I/O operations; 3-way switch	2	0	3
ACP_BASEPRIO[2]	Base priority for ACP processes	6	4	31
ACP_SWAPFLGS[2]	Swapping of ACP working sets; code	15	0	15

1. Also a DYNAMIC parameter, but effective only when a volume is mounted.
2. Also a DYNAMIC parameter, but effective only when an ACP is active.
3. The value -1 means no limit.

Table 12-5
RMS Parameters

Parameter Name	Description	Default	Minimum	Maximum
RMS_DFMBC	Default multiblock count	4	1	127
RMS_DFMBFSDK	Default multiblock count for sequential disk operations	0	0	127
RMS_DFMBFSMT	Default multiblock count for magnetic tape operations	0	0	127
RMS_DFMBFSUR	Default multiblock count for unit record operations	0	0	127
RMS_DFMBFREL	Default multiblock count for relative disk operations	0	0	127
RMS_DFMBFIDX	Default multiblock count for indexed sequential disk operations	0	0	127
RMS_DFMBFHSH	Not implemented	0	0	127

Table 12-6
PQL Parameters

Parameter Name	Description	Default	Minimum[1]	Maximum[1]
PQL_DASTLM	Default number of pending ASTs	6	-1	-1
PQL_MASTLM	Minimum number of pending ASTs	2	-1	-1
PQL_DBIOLM	Default buffered I/O limit	6	-1	-1
PQL_MBIOLM	Minimum buffered I/O limit	2	-1	-1
PQL_DBYTLM	Default buffered I/O byte limit	8192	-1	-1
PQL_MBYTLM	Minimum buffered I/O byte limit	1024	-1	-1
PQL_DCPULM	Default CPU time limit in increments of 10ms	0	-1	-1
PQL_MCPULM	Minimum CPU time limit in increments of 10ms	0	-1	-1
PQL_DDIOLM	Default direct I/O limit	6	-1	-1
PQL_MDIOLM	Minimum direct I/O limit	2	-1	-1
PQL_DFILLM	Default open file limit	10	-1	-1
PQL_MFILLM	Minimum open file limit	2	-1	-1
PQL_DPGFLQUOTA	Default paging file quota	2048	-1	-1
PQL_MPGFLQUOTA	Minimum paging file quota	256	-1	-1
PQL_DPRCLM	Default subprocess limit	8	-1	-1
PQL_MPRCLM	Minimum subprocess limit	0	-1	-1
PQL_DTQELM	Default time queue entries	8	-1	-1
PQL_MTQELM	Minimum timer queue entries	0	-1	-1
PQL_DWSDEFAULT	Default working set size	100	-1	-1
PQL_MWSDEFAULT	Minimum default working set size	10	-1	-1
PQL_DWSQUOTA	Default working set quota	120	-1	-1
PQL_MWSQUOTA	Minimum working set quota	10	-1	-1

1. The value -1 means no limit.

Table 12-7
Standard System Parameter Files

Parameter Name	Parameter File Name						
	MINIMUM	8USER	16USER	32USER	48USER	64USER	VIRT32MB
PFCDEFAULT	16	16	32	64	127	127	127
GBLSECTIONS	20	30	32	48	80	80	32
GBLPAGES	1024	2048	2048	3072	3072	3072	2048
MAXPROCESSCNT	12	20	28	48	68	84	28
SYSMWCNT	80	100	120	140	160	180	120
BALSETCNT	5	12	20	34	52	68	8
IRPCOUNT	0	200	275	530	650	750	300
WSMAX	256	256	256	512	700	1024	1024
NPAGEDYN	32128	51200	74240	131072	159744	179712	92160
PAGEDYN	16384	24576	28672	40960	49152	65536	32768
VIRTUALPAGECNT	4096	4096	4096	8192	8192	8192	65536
QUANTUM	60	60	60	60	60	60	60
MPW_WRTCLUSTER	8	16	32	64	127	127	127
MPW_HILIM	10	24	44	92	220	320	220
MPW_LOLIM	4	12	16	32	100	200	96
SPTREQ	700	650	700	700	700	700	700
WSINC	0	21	21	21	21	21	21
FREELIM	10	10	10	16	16	16	10
BUGCHECKFATAL	0	0	0	0	0	0	0
TTY_DEFCHAR	268440224	268440224	268440224	268440224	268440224	268440224	268440224
MAXPRINTSYMB	1	1	1	4	8	8	2
DEFPRI	4	4	4	4	4	4	4
IJOBLIM	4	9	17	33	49	65	17
BJOBLIM	1	1	1	4	8	8	4
NJOBLIM	16	16	16	16	16	16	16
ACP_SHARE	0	0	0	1	1	1	1
ACP_MAPCACHE	1	2	4	12	20	20	4
ACP_HDRCACHE	4	6	12	40	80	120	12
ACP_DIRCACHE	4	8	16	50	100	150	16
ACP_FIDCACH	8	8	8	12	16	16	8
ACP_EXTCACHE	8	8	16	16	32	32	16
ACP_EXTLIMIT	200	200	200	200	200	200	200
ACP_QUOCACHE	4	8	16	32	48	64	12
ACP_SWAPFLGS	15	15	15	14	14	14	15

Note: All other SYS and ACP parameters (not listed in this table) take defaults. All RMS and PQL parameters take defaults.

12.1 MAJOR PARAMETERS

The MAJOR parameters, a subset of the SYS parameters, most commonly require adjustment.

12.1.1 PFCDEFAULT Parameter

During execution of programs, PFCDEFAULT controls the number of image pages read from disk per I/O operation when a page fault occurs. The read I/O operations may take place from an image file or from the paging file. The actual size of the cluster may be less than PFCDEFAULT depending on the size of image sections and the pattern of page references.

The value of this parameter should not be less than 16 to ensure adequate I/O performance, nor greater than one-fourth the default size of the average working set to prevent a single I/O operation from displacing a major portion of a working set. The average working set default size can be estimated as the median WSDEFAULT value in the UAF (see Chapter 2).

The values in the standard parameter files increase with the size of the system. System managers of 32USER and larger systems especially, should ensure that the PFCDEFAULT value is not too high for the average working set size.

PFCDEFAULT can be overridden on an image-by-image basis with the CLUSTER option of the VAX-11 Linker.

12.1.2 GBLSECTIONS Parameter

GBLSECTIONS sets the number of global section descriptors allocated in the system header at bootstrap time. Each global section requires one descriptor. Each descriptor takes 32 bytes of permanently resident memory.

The values found in the standard system parameter files are more than adequate for the images normally installed as shared in the system start-up command procedures (see Chapters 7 and 8). The system manager may, once the system is running and all global sections are created, examine the actual requirements with the /GLOBAL option of the INSTALL utility (Chapter 7) and reduce the value of GBLSECTIONS accordingly. However, the value of this parameter should not be cut too closely -- each descriptor requires only 32 bytes. If the system manager plans to install many user images as shared or if user programs are likely to create many global sections, the value of this parameter must be increased.

12.1.3 GBLPAGES Parameter

GBLPAGES sets the number of global page table entries allocated at bootstrap time. Each global section requires one global page table entry per section page, plus two entries, with the total rounded up to an even number. Every 128 global page table entries add 4 bytes to permanently resident memory in the form of a system page table entry.

The values found in the standard system parameter files are more than adequate for the images normally installed as shared (see Chapters 7 and 8) in the system start-up command procedures. The system manager may, once the system is running and all global sections are created, examine the actual requirements with the /GLOBAL option of the INSTALL utility (Chapter 7) and reduce the value of GBLPAGES accordingly. However, the value of this parameter should not be cut too closely -- the page table entries are not expensive in terms of permanently resident memory. If the system manager plans to install many user images as shared or if user programs are likely to create many global sections, the value of this parameter must be increased.

12.1.4 MAXPROCESSCNT Parameter

MAXPROCESSCNT sets the number of process entry slots allocated at bootstrap time. One slot is required for each concurrent process on the system. Each slot requires 6 bytes of permanently resident memory.

Total slot requirements can be figured as follows: 1 per interactive user; 1 per batch job; 1 per ACP (including the network ACP); 1 per print symbiont; 1 for the input symbiont (where used); and 3 for ERRFMT, OPCOM, and JOB_CONTROL. The values in the standard system parameter files cover these requirements. The system manager should reduce the value only where interactive use of the system is much lower than anticipated by the standard system parameter file -- for example, if only 20 users are on a 32USER system.

The system automatically reduces the active value of MAXPROCESSCNT if that value times the value of WSMAX exceeds the size of the swapping file in pages. For this reason, the system manager should ensure reasonable values for MAXPROCESSCNT and WSMAX, and an adequate number of pages for the swapping file. (See Chapter 13 for guidelines on creating the swapping file.) The system automatically increases the active value of MAXPROCESSCNT when a larger, secondary swapping file is installed.

12.1.5 **SYSMWCNT Parameter**

SYSMWCNT sets the quota for the size of the system working set, which contains the pageable portions of the system, the paged dynamic pool, VAX-11 RMS, and the resident portion of the system message file.

While a high value takes space away from user working sets, a low value may seriously impair system performance. Appropriate values vary depending on the level of system use. The system manager should -- when the system is running at full load -- check the rate of system faults with the PAGE command in the DISPLAY utility (see Chapter 14). If the system page fault rate is high, and especially if the system seems to be slow, the value of SYSMWCNT should be increased.

12.1.6 **BALSETCNT Parameter**

BALSETCNT sets the number of balance set slots in the system page table. Each memory-resident working set requires one balance set slot. Each balance set slot requires 4 bytes of permanently resident memory per 128 virtual pages (as specified in the VIRTUALPAGECNT parameter; see Section 12.1.11). The requirements for the standard system parameter files are as follows:

	MINIMUM	**8USER**	**16USER**	**32USER**	**48USER**	**64USER**	**VIRT32MB**
VIRTUALPAGECNT	2048	4096	4096	8192	8192	8192	65536
Bytes per Slot	64	128	128	256	256	256	2048
BALSETCNT	5	12	20	34	52	68	8
Total Bytes	320	1536	2560	8704	13312	17408	16384

The system manager can monitor the active system with the DCL command SHOW SYSTEM or with the USERS command of the DISPLAY utility to ascertain the actual maximum number of working sets in memory. If this number is significantly lower than the value of BALSETCNT, this parameter value may be lowered. If all balance set slots are being used, the value of BALSETCNT should be raised.

Lowering the value of BALSETCNT has the effect of increasing the size of the free page list. The system manager might take this tack if the page fault rate is excessive. However, the swapping rate is apt to increase and must be monitored closely.

12.1.7 **IRPCOUNT Parameter**

IRPCOUNT sets the number of preallocated I/O request packets. Each packet requires 96 bytes of permanently resident memory from the nonpaged dynamic pool. If the amount of space required by this number exceeds half the space allocated to the nonpaged dynamic pool (see the NPAGEDYN parameter below), the value of IRPCOUNT is automatically reduced to half that space. The value of IRPCOUNT is automatically reduced to 0 for systems configured with the MINIMUM parameter file.

Preallocated I/O request packets (which are also used to fill requests for nonpaged dynamic memory where the amount of memory requested is equal to or less than the size of a packet) reduce system overhead. When all preallocated packets are in use, I/O packets must be allocated dynamically from the nonpaged dynamic pool.

12.1.8 **WSMAX Parameter**

WSMAX sets the maximum number of pages for any one working set. The value of WSMAX also affects the allocation of permanently resident memory for the swapper map and system page table (and for this reason should not be set at an arbitrarily high figure):

- Swapper map -- 4 bytes for each page of WSMAX
- System page table -- 4 bytes for each 128 pages of WSMAX, times BALSETCNT

Memory requirements for the standard system parameter files are as follows:

	MINIMUM	8USER	16USER	32USER	48USER	64USER	VIRT32MB
WSMAX	256	256	256	512	700	1024	1024
Swapper Map Bytes	1024	1024	1024	2048	2800	4096	4096
BALSETCNT	5	12	20	34	52	68	8
Page Table Bytes	40	96	160	544	1096	2176	256

WSMAX also affects the size of the swapping file in that the file must be large enough to hold WSMAX times MAXPROCESSCNT pages. If the swapping file is not large enough when the system is initialized, the value of MAXPROCESSCNT is reduced until WSMAX times MAXPROCESSCNT pages will fit. (The number of process slots allocated, however, is still based on the original value of MAXPROCESSCNT.) In performing these calculations, the system determines the space available for the balance set (number of pages in the system reduced by the number of nonpageable pages, and the values of SYSMWCNT and FREELIM) and reduces WSMAX to this figure if WSMAX exceeds the figure.

Generally, the system manager should use a reasonable value for WSMAX; that is, whatever the size of the largest working set will be. On the smaller configurations, the system manager may raise the standard values by as much as a few hundred pages. On the larger configurations, the system manager may lower the standard values as much as desired.

12.1.9 **NPAGEDYN Parameter**

NPAGEDYN sets the size of the nonpaged dynamic pool in bytes. This figure is rounded down to an integral number of pages.

Probably the simplest and best approach to take in setting a value for this parameter is to initially use the value found in the appropriate standard system parameter file, then monitor the amount of space actually used with the POOL command of the DISPLAY utility. The value may be decreased if a lot of space is being wasted and should be increased if little space is unused or if all I/O packets are used.

Insufficient space for the nonpaged dynamic pool rapidly degrades system performance, possibly to the point that the system does not function.

To approximate the size of the nonpaged dynamic pool, first add the following:

- MAXPROCESSCNT * 518
- WSMAX * 4
- IRPCNT * 96
- BALSETCNT * 4
- MPW_WRTCLUSTER * 6
- KFILSTCNT * 68
- Size of paging file (in pages) divided by 8
- Size of network device (DMC11) receive buffers

All totals are in bytes. Next add the space required for loadable drivers being used and their data base structures. The sizes of the standard drivers and associated data bases are as follows:

Driver Name	Driver Size	Controller Structures	Unit Structures
CRDRIVER	1328	288	128
DXDRIVER	1008	288	320
DBDRIVER	1776	288	160
DLDRIVER	2128	288	192
DMDRIVER	2288	288	208
DRDRIVER	2080	288	176
DYDRIVER	1776	288	192
LADRIVER	2688	288	400
LPDRIVER	1008	288	128
MBXDRIVER	1776	288	144
TSDRIVER	3232	288	176
TMDRIVER	4256	384	160
NETDRIVER	592	288	144
RTTDRIVER	1136	288	128
DZDRIVER	6544	288	192
XFDRIVER	1504	288	144
XMDRIVER	3232	288	240

All figures are in bytes. Calculate a total for each driver by first multiplying the next-to-the-last column by the number of controllers and the last column by the number of units, then totaling all columns. A DZDRIVER supporting 16 terminals on 2 controllers, for example, requires 6544 + (288 x 2) + (192 x 16) = 10,192 bytes. An LPDRIVER supporting 2 printers requires 1008 + (288 x 2) + (128 x 2) = 1840 bytes (printers require a controller for each unit). In most cases, the controller structures mean the device data block (DDB), channel request block (CRB), and interrupt data block (IDB), which require one I/O request packet each (96 bytes each for a total of 288 bytes). The controller structures for TMDRIVER, however, require 4 I/O request packets. The unit structures refer to the unit control blocks (UCBs), which vary in size.

No matter how many units are in the configuration, only one copy of each driver is required.

12.1.10 **PAGEDYN Parameter**

PAGEDYN sets the size of the paged dynamic pool in bytes. The specified value is rounded down to an integral number of pages. Each 512 bytes of paged dynamic pool adds 4 bytes of permanently resident memory to the system page table; the paged dynamic pool has no other direct memory requirements.

The paged dynamic pool is used to allocate storage for system and group logical names, resident image headers, known file list entries, and VAX-11 RMS file sharing structures. Substantial amounts of space for the pool can be overallocated with little effect on system performance.

12.1.11 **VIRTUALPAGECNT Parameter**

VIRTUALPAGECNT sets the maximum number of virtual pages that can be mapped for any one process. Every 128 virtual pages requires 4 bytes of permanently resident memory in the system page table (as discussed under the BALSETCNT parameter, Section 12.1.6), 1 page table page (which is pageable), and 8 bytes in the process header (which is not pageable, but is swapped). The total number of virtual pages may be divided between the P0 and P1 tables in any proportion except that the P1 table must be large enough to map 320 pages.

The standard system parameter files provide virtual address spaces of 1 million (MINIMUM), 2 million (8USER, and 16USER), and 4 million (32USER, 48USER, and 64USER) bytes, which should be sufficient for normal use. The VIRT32MB file provides an address space of 32 million bytes for a limited number of programs. The system manager should use the values in the standard parameter system files unless virtual-full errors occur.

When the System Dump Analyzer is used, the system manager must insure that the value of VIRTUALPAGECNT is at least the size of the dump file plus 1024 pages.

12.1.12 QUANTUM Parameter

QUANTUM serves two purposes:

- Processor time -- Maximum amount of processor time a process can receive before control passes to another process of equal priority that is ready to compute

- Balance set residency -- Minimum amount of service (includes processor time and time waiting for completion of I/O services) a compute-state process must receive before being outswapped to secondary storage

The standard system parameter value of 30 (300 milliseconds) should be adequate for most applications. On a system where the swapping rate is high (as shown by the IORATES command of the DISPLAY utility), a high value -- even up to 300 (3 seconds) -- may improve overall performance by reducing the disk I/O to the swapping file.

12.1.13 Modified Page Writer Parameters

The Modified Page Writer parameters consist of MPW_WRTCLUSTER, MPW_HILIM, and MPW_LOLIM.

MPW_WRTCLUSTER sets the number of pages to be written from the modified page list during one I/O operation to the paging file or a section file. (The actual size of the cluster may be limited by the number of pages available for the I/O operation.) Each page in the cluster requires 6 bytes of permanently resident memory.

MPW_HILIM sets an upper limit for the modified page list. When the list accumulates the number of pages specified by this limit, writing of the list commences. (The pages that are written are then transferred to the free page list.)

MPW_LOLIM sets a lower limit for the modified page list. When writing of the list causes the number of pages on the list to drop to or below this limit, writing stops.

The system manager may want to adjust the standard system parameter files so that the lower limit is 12 to 16 pages above the MPW_WRTCLUSTER value. This action ensures that the modified page list is not depleted by a write I/O operation. Increasing the value of the lower limit tends to reduce the write I/O operations from the modified page list, but may increase the swapping rate. Both write I/O operationss and the swapping rate can be monitored with the IORATES command of the DISPLAY utility.

12.2 SYS PARAMETERS

The remaining (nonmajor) SYS parameters should be checked for reasonableness, as described in the sections that follow.

12.2.1 **KFILSTCNT Parameter**

KFILSCNT sets the number of known file list heads. Each active list requires 68 bytes of permanently resident memory in the nonpaged dynamic pool. Each extra, never-used list head requires 4 bytes. The actual known file entries are allocated from the paged dynamic pool.

One known file head is required for each set of installed images with a different combination of device name, directory name, and file type. (See Chapter 7.)

12.2.2 **PROCSECTCNT Parameter**

PROCSECTCNT sets the number of section descriptors that a process can contain. Each section descriptor increases the fixed portion of the process header by 32 bytes.

The system manager should set a value greater than the maximum number of image sections in any section to be run, as indicated by the linkage memory allocation map for the image.

12.2.3 **MINWSCNT Parameter**

MINWSCNT sets the minimum number of fluid pages -- that is, pages not locked in the working set -- required for the execution of a process. This value plus the size of the process header establishes the minimum working set size.

The value of MINWSCNT must provide sufficient space to execute any VAX-11 instruction. Theoretically, the worst-case instruction requires 52 pages; however, all VAX/VMS code can run with 20 fluid pages. An insufficient value may inhibit system performance or even put a process into an infinite loop on some instructions.

12.2.4 **INTSTKPAGES Parameter**

INTSTKPAGES sets the size of the interrupt stack in pages. Each page on the interrupt stack requires a page of permanently resident memory.

The default value of 2 should be used unless interrupt-stack-not-valid exceptions occur. These may be caused by either an unusually large number of devices or a driver that requires a very large amount of stack space.

12.2.5 **SPTREQ Parameter**

SPTREQ sets the number of system page table (SPT) entries required for mapping the following components:

Component	**SPT Entries**
Executive image	185
VAX-11 RMS image	126
SYSMSG.MPF file	252
Multiport memory structures	variable
Each MASSBUSS adapter	7
Each UNIBUS adapter	24
Each DR32 adapter	4

The number of system page table entries required for all other purposes is automatically computed and added to the value of SPTREQ to yield the actual size of the system page table.

12.2.6 **PFRATL Parameter**

PFRATL specifies the page fault rate below which the limit of a working set will be automatically decreased. The unit of measure is faults per 10 seconds of processor time. At the default setting, for example, the system will automatically decrease the limit of a working set if it is faulting less than 2 pages every 10 seconds.

Increasing the value of this parameter tends to decrease the limits of the working sets, while decreasing its value tends to increase their limits.

12.2.7 **PFRATH Parameter**

PFRATH specifies the page fault rate above which the limit of a working set will be automatically increased. The unit of measure is faults per 10 seconds of processor time. At the default setting, for example, the system will automatically increase the limit of a working set if it is faulting more than 512 pages per 10 seconds.

Decreasing the value of this parameter tends to increase the limits of the working sets, while increasing its value tends to decrease their limits.

12.2.8 **WSINC Parameter**

WSINC specifies the number of pages by which the limit of a working set is automatically increased at each adjustment interval (which is quantum end). At the default setting, for example, the system will increase the limit of a working set by 21 pages each time an increase is required.

Decreasing the value of this parameter tends to reduce the speed with which working set limits are increased when the need arises. Normally, this parameter should be kept at a high value because a rapid increase in limit is often critical to performance. (The need for an increase in limit typically occurs when new images or sections are being introduced to the working set, so that the working set must suddenly accommodate many more pages than before.) In addition, the value of this parameter should be prime relative to WSDEC so that working set limits will vary as they are increased and decreased.

A value of 0 for WSINC disables the automatic adjustment of working set limits for all processes. Limits stay at their base values. The automatic adjustment of working set limits can be disabled on a per-process basis by setting the limit and quota equal.

12.2.9 **WSDEC Parameter**

WSDEC specifies the number of pages by which the limit of a working set is automatically decreased at each adjustment interval (which is quantum end). At the default setting, for example, the system will decrease the limit of a working set by 5 pages each time a decrease is required.

Increasing the value of this parameter tends to increase the speed with which working set limits are decreased when the need arises. Normally, this parameter should be kept at a low value because a sudden drop in the working set limit may seriously impair performance. In addition, the value of this parameter should be prime relative to WSINC so that working set limits will vary as they are increased and decreased.

12.2.10 **AWSMIN Parameter**

AWSMIN establishes the lowest number of pages to which a working set limit can be decreased automatically.

12.2.11 **AWSMAX Parameter**

AWSMAX establishes the highest number of pages to which a working set limit can be increased automatically. However, the system will never increase the limit of a working set above the quota of its process.

A setting equal to the value of the WSMAX parameter permits all working set limits to increase automatically to the quotas of their processes.

12.2.12 **AWSTIME Parameter**

AWSTIME specifies the minimum amount of processor time that must elapse for the system to collect a significant sample of a working set's page fault rate. The time is expressed in units of 10 milliseconds. The default value of 20, for example, is 200 milliseconds. However, the practical value of this parameter never exceeds the quantum, as page fault rates are sampled at quantum end.

12.2.13 **EXTRACPU Parameter**

EXTRACPU sets the time, in units of 10 milliseconds, allotted to each of a process's exit handlers (for each access mode) after the process times out (that is, reaches its CPU time limit).

12.2.14 **MAXSYSGROUP Parameter**

MAXSYSGROUP sets the highest value that a group number can have and still be classified as a system UIC group number. Note that the specification is not in octal unless preceded by the %O radix indicator. This parameter is normally left at 8 (10 octal).

12.2.15 **MAXBUF Parameter**

MAXBUF sets the maximum size of a buffered I/O transfer (card readers, console floppy diskettes, line printers, mailboxes, and terminals). The space for a buffered I/O operation is allocated from the permanently resident nonpaged dynamic pool. Note that the system adds approximately 16 bytes to a buffer at allocation time for header information, so that the default (1024) -- for most devices -- means that a buffer no larger than 1008 bytes can be allocated.

12.2.16 **DEFMBX Parameters**

The DEFMBX parameters set defaults for the mailbox buffer quota and message size when these values are not specified in a Create Mailbox ($CREMBX) system service call.

12.2.17 **FREELIM Parameter**

FREELIM sets the minimum number of pages that must be on the free page list. The system will write pages from the modified page list or outswap working sets to maintain the minimum count, if necessary.

While the larger free page list generally means less paging I/O, it also means less space for the balance set, which tends to result in more swapping I/O. The system manager can monitor the size of the free page list, the amount of paging, and the amount of swapping with the IORATES command of the DISPLAY utility.

12.2.18 **XFMAXRATE Parameter**

XFMAXRATE limits the data transfer rate that can be set for DR32 devices. On some hardware configurations (especially those without interleaved memory), a high DR32 transfer rate could cause a machine check (CPU timeout).

12.2.19 **LAMAPREGS Parameter**

LAMAPREGS sets the number of UNIBUS map registers allocated to an LPA11 driver when the driver is loaded, and limits the registers for the driver to that number. A value of 0 permits dynamic allocation of an unlimited number of registers.

12.2.20 **REALTIME_SPTS Parameter**

REALTIME_SPTS reserves a number of system page table entries for mapping connect-to-interrupt processes into system space. This value should normally remain at the default (0) in an environment that is not real-time. Where connect-to-interrupt processes do use the system, this value should represent the maximum number of pages all concurrent connect-to-interrupt processes must map into system space. See the VAX/VMS Real-Time User's Guide for details.

12.2.21 **CLISYMTBL Parameter**

CLISYMTBL sets the size of the command interpreter symbol table. For generation of RSX-11S systems, the value of this parameter should be increased to 35. Otherwise, the default value is adequate.

12.2.22 **BUGREBOOT Parameter**

BUGREBOOT enables or disables automatic rebooting of the system if a fatal bugcheck occurs. This switch should normally be off (0) only when the executive is being debugged.

12.2.23 **CRDENABLE Parameter**

CRDENABLE enables or disables detection and logging of memory-corrected read data (ECC) errors. This switch should normally be on (1).

12.2.24 **DUMPBUG Parameter**

DUMPBUG enables or disables the writing of error log buffers and memory contents to SYS$SYSTEM:SYSDUMP.DMP when a fatal bugcheck occurs. This switch should be off (0) only when the executive is being debugged.

12.2.25 **BUGCHECKFATAL Parameter**

BUGCHECKFATAL enables or disables the making of all bugchecks fatal bugchecks. The system must be rebooted on a fatal bugcheck. A nonfatal bugcheck just places an entry in the error log. This switch should normally be on (1).

12.2.26 **POOL_PAGING Parameter**

POOL_PAGING enables or disables paging of the pageable dynamic pool. When paging is disabled, the pageable dynamic pool is allocated as permanently resident memory, just like the nonpaged dynamic pool. This switch should normally be on (1) to conserve physical memory.

12.2.27 **SBIERRENABLE Parameter**

SBIERRENABLE enables or disables error logging for certain classes of SBI errors. This switch should be off only under extraordinary debugging situations.

12.2.28 SETTIME Parameter

SETTIME enables or disables solicitation of the time of day each time the system is booted. This switch should normally be off (0), so that the system sets the time of day at boot time to the value of the processor time-of-day register. The system manager can reset the time after the system is up with the SET TIME command (see the VAX/VMS Operator's Guide).

12.2.29 SYSPAGING Parameter

SYSPAGING enables or disables the paging of code in the system working set. This parameter may be off (0) for a system with sufficient memory and a stringent real-time response requirement, or in a debugging situation, but should normally be on (1). Setting this switch off does not disable the paging of VAX-11 RMS code.

12.2.30 UAFALTERNATE Parameter

UAFALTERNATE enables or disables the assignment of SYSUAF as the logical name for SYSUAFALT, causing all references to the user authorization file (SYSUAF) to be translated to SYS$SYSTEM:SYSUAFALT. Use of the normal user authorization file (SYS$SYSTEM:SYSUAF) can be restored by deassigning the system logical name SYSUAF. This switch should be on (1) only where the system is being used by a restricted set of users. The system manager must create a user authorization file named SYSUAFALT prior to its use (see Chapter 2).

12.2.31 RESALLOC Parameter

This parameter is not in use currently.

12.2.32 TTYSCANDELTA Parameter

TTYSCANDELTA sets the interval for polling terminals for dial-up and hang-up events. Shorter intervals use more processor time; longer intervals may result in missing a hang-up event.

12.2.33 TTY_SPEED Parameter

TTY_SPEED sets the default speed for terminals, using the following codes:

Code	Baud Rate	Code	Baud Rate
1	50	8	1200
2	75	9	1800
3	110	10	2000
4	134.5	11	2400
5	150	12	3600
6	300	13	4800
7	600	14	7200
		15	9600

12.2.34 TTY_BUF Parameter

TTY_BUF sets the default line width for terminals.

12.2.35 TTY_DEFCHAR Parameter

TTY_DEFCHAR sets the default characteristics for terminals, using a code derived by summing the following values:

Characteristic	Value in Hexadecimal
page length	(page length)*1,000,000
if SCOPE	1,000
if WRAPAROUND	200
if MECH_TAB	100
if LOWER_CASE	80
if TERMINAL_SYNCH	20

Where a condition is false, the value is 0. Common codes include:

Terminal	Value in Hexadecimal	Decimal Equivalent
LA36	80002A0	134218400
VT52	180012A0	402657952
VT100	180012A0	402657952

12.2.36 TTY_TYPAHDSZ Parameter

TTY_TYPAHDSZ sets the size of the terminal type-ahead buffer.

12.2.37 TTY_PROT Parameter

TTY_PROT sets the default protection for all terminals in relation to the UIC specified for the TTY_OWNER parameter below. The specification must represent the value of a standard 16-bit protection mask as described in Chapter 3. However, only read bits are meaningful. When a read bit is on, that category of user is prohibited from allocating terminals.

The default (0) provides for no protection on terminals. The system manager should normally set TTY_PROT to a value of hexadecimal 1110, which permits only system processes to allocate terminals. The system manager can change protection on a per-terminal basis with the SET PROTECTION /TERMINAL command to permit user allocation of remote and application terminals.

12.2.38 TTY_OWNER Parameter

TTY_OWNER specifies the owner UIC against which terminal protection is set. The specification must represent the value of a standard 32-bit UIC -- group number in the high-order word and member number in the low-order word. The system manager should normally set TTY_OWNER to a value of hexadecimal 10004, which is UIC [1,4].

12.3 JOB PARAMETERS

The JOB parameters should be checked for reasonableness, as described in the sections that follow.

12.3.1 MAXPRINTSYMB Parameter

MAXPRINTSYMB sets the maximum number of print symbionts that can be created. (However, the maximum number of print symbionts is further restricted by the value of the PQL_DPRCLM parameter -- the default process creation limit.)

12.3.2 DEFPRI Parameter

DEFPRI sets the base default priority for processes. Its value should normally be left at 4.

12.3.3 JOBLIM Parameters

The JOBLIM parameters set the maximum number of interactive (IJOBLIM), batch (BJOBLIM), and network (NJOBLIM) jobs that can be on the system concurrently. The maximum number of processes (MAXPROCESSCNT) may actually reduce any or all of the job limits.

12.4 ACP PARAMETERS

Varying circumstances make changes in the ACP parameters worthwhile, as described in the sections that follow.

12.4.1 ACP_MULTIPLE Parameter

ACP_MULTIPLE enables or disables the default creation of a separate disk ACP process for each volume mounted on a different device type. Even where memory is not at a premium, this switch should normally be off (0) to queue all requests through the same ACP process. Performance is better enhanced by increasing cache sizes (see Section 12.4.3) than by adding another ACP. The switch can be overridden on an individual-volume basis with the DCL command MOUNT.

12.4.2 ACP_SHARE Parameter

ACP_SHARE enables or disables the creation of a global section for the first ACP used, so that succeeding ACPs may share its code. This switch should be set off (0) when ACP_MULTIPLE is off.

12.4.3 CACHE Parameters

The CACHE parameters set the number of pages that each Files-11 ACP can allocate for caching control information rather than going to disk each time such information is needed or is modified. Cache sizes should generally be maximized to reduce ACP I/O activity. They can be overridden on an individual volume basis in the DCL command Mount.

12.4.3.1 **ACP_MAPCACHE Parameter** - ACP_MAPCACHE sets the number of pages for caching bit map blocks.

12.4.3.2 **ACP_HDRCACHE Parameter** - ACP_HDRCACHE sets the number of pages for caching file header blocks.

12.4.3.3 **ACP_DIRCACHE Parameter** - ACP_DIRCACHE sets the number of pages for caching directory blocks.

12.4.3.4 **ACP_FIDCACHE Parameter** - ACP_FIDCACHE sets the number of file identification slots cached. A specification of 1 means no cache.

12.4.3.5 **ACP_EXTCACHE Parameter** - ACP_EXTCACHE sets the number of entries in the extent cache. Each entry points to one contiguous area of free space on disk. A specification of 0 means no cache.

12.4.3.6 **ACP_EXTLIMIT Parameter** - ACP_EXTLIMIT specifies the maximum amount of free space to which the extent cache can point, expressed in thousandths of the currently available free blocks on the disk. For example, if available free space on the disk is 20,000 blocks, a specification of 10 limits the extent cache to 200 blocks. The reason for this parameter is to limit the amount of free space that might be lost in the event of a system failure. However, lost free space on a volume is normally recovered at mount time.

12.4.3.7 **ACP_QUOCACHE Parameter** - ACP_QUOCACHE sets the number of quota file entries cached. A specification of 0 means no cache.

12.4.3.8 **ACP_WRITEBACK Parameter** - ACP_WRITEBACK enables the deferred writing of file headers. A specification of 0 causes all modifications of file headers to be written to disk immediately.

12.4.3.9 **ACP_SYSACC Parameter** - ACP_SYSACC sets the number of directory file control blocks (FCBs) that will be cached for disks mounted with the /SYSTEM qualifier. Each directory FCB contains a 16-byte array containing the first letter of the last entry in each block of the directory (or group of blocks if the directory exceeds 16 blocks). Since entries in a directory are alphabetical, the cached FCB provides quick access to a required directory block.

This parameter value should be roughly equivalent to the number of directories that will be in use concurrently on each system volume. It may be overridden on a per-volume basis with the /ACCESSED qualifier of the MOUNT command. The value should be kept down in small systems, as the FCBs require a significant amount of space in the nonpaged dynamic pool.

12.4.4 ACP_WORKSET Parameter

ACP_WORKSET sets the default size of a working set for an ACP. A value of zero permits the ACP to calculate the size. This value should be nonzero only on small systems where memory is tight.

12.4.5 ACP_MAXREAD Parameter

ACP_MAXREAD sets the maximum number of directory blocks read in one I/O operation. This parameter does not affect user file I/O.

12.4.6 ACP_WINDOW Parameter

ACP_WINDOW sets the default number of window pointers to be allocated in a window for a default file access, for disks mounted with the /SYSTEM qualifier.

12.4.7 ACP_DATACHECK Parameter

ACP_DATACHECK enables verification of reading and/or writing of file structure data (for example, directories and file headers): a specification of 3 means read and write checks; 2 means write check only; 1 means read check only; 0 means no checks. On a read check, the ACP information is read twice and compared. On a write check, the ACP information is written, then read and compared.

12.4.8 ACP_BASEPRIO Parameter

ACP_BASEPRIO sets the base priority for all ACPs. The SET PROCESS /PRIORITY command can be used to reset the base priorities of individual ACPs.

12.4.9 ACP_SWAPFLGS Parameter

ACP_SWAPFLGS enables or disables swapping for four classes of ACPs through the value of a 4-bit number:

Bit	Class of ACP
0	Disks mounted /SYSTEM
1	Disks mounted /GROUP
2	Private disks
3	Magnetic tape ACP

If the value of the bit is 1, the corresponding class of ACPs can be swapped. The default value of hexadecimal F (all bits on) enables swapping for all classes of ACP. A value of hexadecimal E would disable swapping for ACPs for volumes mounted with the /SYSTEM qualifier. Normally, swapping is enabled for all ACPs except those for system volumes.

12.5 RMS PARAMETERS

The RMS parameters set default blocking factors for all VAX-11 RMS I/O operations (RMS_DFMBC) or selected VAX-11 RMS I/O operations (the remaining RMS parameters). For small systems, the RMS parameters are normally left at their default values (RMS_DFMBC equals 4 and the remaining RMS parameters equal 0). For large systems, the system manager should increase the value of RMS_DFMBC or selected RMS parameters affecting operations involving many contingent blocks of data.

12.6 PQL PARAMETERS

The PQL parameters set default limits for processes created by the Create Process ($CREPRC) system service and the RUN (Process) command. They are normally left at their default values.

CHAPTER 13

SYSTEM GENERATION

The system manager performs various functions to ensure proper configuration of the system:

- System parameters -- Creates and modifies the standard system parameter files for use in subsequent conversational bootstrap operations; modifies the parameter values in the current system image (SYS$SYSTEM:SYS.EXE) for use in subsequent bootstrap operations; dynamically modifies the parameter values of the active system (applies only to the dynamic system parameters)
- Devices and device drivers -- Connects devices and loads their device drivers (most of this work is automatic)
- System files -- Creates swapping, paging, and dump files
- Start-up command procedure -- Names the current site-independent start-up command procedure
- Multiport (shared) memory -- Initializes multiport memory units

13.1 SYSGEN UTILITY

The system manager performs the system generation functions with the SYSGEN utility. Table 13-1 summarizes the SYSGEN commands by format and function.

Table 13-1
SYSGEN Command Summary

Format	Function
AUTOCONFIGURE nexus [/SELECT=(device,...)]	Automatically connects devices physically attached to the system and loads their drivers
CONNECT device /ADAPTER=nexus [/CSR=csr-addr] [/DRIVERNAME=driver] [/MAXUNITS=max-unit-cnt] [/NUMVEC=vector-cnt] [/VECTOR=vector-addr]	Connects hardware devices and loads their drivers (if they are not already loaded)

(continued on next page)

SYSGEN

Table 13-1 (Cont.)
SYSGEN Command Summary

Format	Function
CONNECT device /NOADAPTER [/DRIVERNAME=driver]	Connects software devices and loads their drivers (if they are not already loaded)
CONNECT CONSOLE	Connects the console block storage device and loads its driver
CREATE file-spec /SIZE=block-count [/[NO]CONTIGUOUS]	Creates a paging, swapping, or dump file
DISABLE CHECKS	Disables range checks
ENABLE CHECKS	Enables range checks
EXIT	Terminates SYSGEN
HELP [command [qualifier]]	Lists the SYSGEN commands
INSTALL file-spec {/PAGEFILE | /SWAPFILE}	Activates a secondary paging or swapping file
LOAD file-spec	Loads a device driver
RELOAD file-spec	Loads a new version of an existing device driver
SET parameter-name value	Modifies the value of a system generation parameter in the SYSGEN work area
SET /STARTUP file-spec	Names the current site-independent start-up command procedure
SHARE MPMn mpm-name /INIT [/GBLSECTIONS=gbl] [/MAILBOXES=mail] [/CEFCLUSTERS=cef] [/MAXGBLSECTIONS=max-gbl] [/MAXMAILBOXES=max-mail] [/MAXCEFCLUSTERS=max-cef] [/POOLBCNT=block-cnt] [/POOLBSIZE=block-size] [/PRQCOUNT=prq-cnt] [/START=start]	Initializes a multiport memory unit
SHARE MPMn mpm-name [/GBLSECTIONS=gbl] [/MAILBOXES=mail] [/CEFCLUSTERS=cef]	Connects a multiport memory unit

(continued on next page)

SYSGEN (Cont.)

Table 13-1 (Cont.)
SYSGEN Command Summary

Format	Function
SHOW { parameter-name / /ACP / /ALL / /DYNAMIC / /GEN / /JOB / /MAJOR / /NAMES / /PQL / /RMS / /SYS }	Displays the values of system parameters in the SYSGEN work area, plus the default, minimum, and maximum values of the parameters, and their units of measure
SHOW /DEVICE[=driver-name]	Displays information on connected devices and loaded drivers
SHOW /STARTUP	Displays the name of the current site-independent start-up command procedure
USE { file-spec / CURRENT / ACTIVE / DEFAULT }	Initializes the SYSGEN work area with system parameter values from a parameter file, the current system image, the active system, or the default list
WRITE { file-spec / CURRENT / ACTIVE }	Writes the system parameter values from the SYSGEN work area to a parameter file, the current system image, or the active system

Many of the SYSGEN commands are specific to particular functions:

- System parameters -- USE, SET, ENABLE, DISABLE, WRITE, and SHOW
- Devices and device drivers -- AUTOCONFIGURE, CONNECT, LOAD, RELOAD, and SHOW /DEVICES
- System files -- CREATE and INSTALL
- Start-up command procedure -- SET /STARTUP and SHOW /STARTUP
- Multiport memory -- SHARE

A subset of the SYSGEN commands can be used during bootstrap operations; see the VAX-11 Software Installation Guide.

SYSGEN (Cont.)

13.1.1 Invoking SYSGEN

The following command invokes the utility:

```
$ RUN SYS$SYSTEM:SYSGEN
```

The system responds with the following prompt:

```
SYSGEN>
```

The system manager can then enter any of the commands listed in Table 13-1. These commands follow the standard rules of grammar as specified in the VAX/VMS Command Language User's Guide.

13.1.2 AUTOCONFIGURE Command

AUTOCONFIGURE automatically connects devices that are physically attached to the system and loads their drivers. It takes the form:

```
AUTOCONFIGURE nexus [/SELECT=(device,...)]
```

nexus
: Nexus number of a UNIBUS or MASSBUS adapter, or the keyword ALL

device
: Standard device name as listed in Table 13-2 (see Section 13.2.2) and (optionally) a controller designation; if the controller designation is omitted, device generically specifies all devices of the specified type; parentheses can be omitted for a single device; defaults to all devices on the adapter

The autoconfigure operation assigns standard device names and assumes standard driver names, as listed in Table 13-2 (see Section 13.2.2).

The following example automatically configures all standard devices:

```
SYSGEN> AUTOCONFIGURE ALL
```

The next example automatically configures all terminals, all magnetic tape units on controller A, all RM03 disks, and all line printers:

```
SYSGEN> AUTOCONFIGURE ALL /SELECT=(TT,MTA,DR,LP)
```

Use of the AUTOCONFIGURE command requires the CMEXEC and CMKRNL privileges.

13.1.3 CONNECT (Hardware) Command

CONNECT (hardware) connects a hardware device and loads its driver, if the driver is not already loaded. It takes the form:

```
CONNECT device /ADAPTER=nexus
               [/CSR=csr-addr]
               [/DRIVERNAME=driver]
               [/MAXUNITS=max-unit-cnt]
               [/NUMVEC=vector-cnt]
               [/VECTOR=vector-addr]
```

device
: Name of the device; should take the form device type, controller, unit -- for example, LPA0 for unit 0 on controller A of device type LP

nexus
: Nexus number of the UNIBUS or MASSBUS adapter to which the device is attached

csr-addr
: UNIBUS address of the controller status register for the device; must be specified for UNIBUS devices

driver
: Name of the driver as recorded in the prologue table; if the driver has not been loaded, the system assumes that the driver name is also the name of an executable (file type of EXE) image in SYS$SYSTEM, and loads the driver; DIGITAL standard drivers are named in Table 13-2 (see Section 13.2.2); defaults to the first two characters of the device name plus DRIVER

max-unit-cnt
: Highest number unit attached to the device controller plus 1; defaults to the number specified in the prologue table of the driver, or 8 (if not explicitly specified in the prologue table)

vector-cnt
: Number of interrupt vectors for the device; defaults to 1

vector-addr
: UNIBUS address of the interrupt vector for the device, or the lowest vector if there is more than one; must be specified for UNIBUS devices

The system manager typically uses this command to connect devices that are not physically attached or whose drivers are not standard. The following example connects the device named LPA0 to the driver named LPDRIVER (the default driver name), and loads the driver if it is not already loaded:

```
SYSGEN>  CONNECT LPA0 /ADAPTER=3 /CSR=%O777514-
_/VECTOR=%O200
```

The next example performs the same operation, but uses a nonstandard driver named LP2DRIVER:

```
SYSGEN>  CONNECT LPA0 /ADAPTER=3 /CSR=%O777514-
_/DRIVERNAME=LP2DRIVER/VECTOR=%O200
```

SYSGEN (Cont.)

The system manager should exercise extreme caution when using the CONNECT command, as the system does little error checking. An incorrect vector address or misspelled device name, for example, will damage the I/O data base and very likely cause the system to fail.

The VAX/VMS Guide to Writing a Device Driver contains more detailed information on loading device drivers and connecting devices.

Use of the CONNECT command requires the CMEXEC and CMKRNL privileges.

13.1.4 **CONNECT (Software) Command**

CONNECT (software) connects a software device and loads its driver if it is not already loaded. It takes the form:

```
CONNECT device /NOADAPTER
            [/DRIVERNAME=driver]
```

device
    Name of the device

driver
    Name of the driver as recorded in the prologue table; if the driver has not been loaded, the system assumes that the driver name is also the name of an executable (file type of EXE) image in SYS$SYSTEM, and loads the driver; DIGITAL standard drivers are named in Table 13-2 (see Section 13.2.2); defaults to the first two characters of the device name plus DEVICE

The following example connects the device NET to the driver NETDRIVER, and loads the driver if it is not already loaded:

```
SYSGEN>  CONNECT NET /NOADAPTER /DRIVER=NETDRIVER
```

Use of the CONNECT command requires the CMEXEC and CMKRNL privileges.

13.1.5 **CONNECT (Console) Command**

CONNECT (console) connects the console block storage device and loads its driver. It takes the form:

```
CONNECT CONSOLE
```

The name of the console block storage device is CSA1.

Use of the CONNECT command requires the CMEXEC and CMKRNL privileges.

13.1.6 **CREATE Command**

CREATE creates a file that can be used as a primary paging, swapping, or dump file, or a secondary paging or swapping file. It takes the form:

```
CREATE file-spec /SIZE=block-count
            [/[NO]CONTIGUOUS]
```

file-spec
Name of the file; primary files must be named SYS$SYSTEM:PAGEFILE.SYS, SWAPFILE.SYS, and SYSDUMP.DMP

block-count
Number of blocks to be allocated to the file

Primary files must be contiguous. Secondary files can be noncontiguous and can span volumes in a volume set. The default specification is /CONTIGUOUS.

The following example creates a primary paging file of 98,304 blocks:

```
SYSGEN> CREATE SYS$SYSTEM:PAGEFILE.SYS /SIZE=98304
```

The next example creates a secondary non-contiguous paging file:

```
SYSGEN> CREATE DRA5:[SYSTEM]PAGEFILE.SYS /SIZE=98304
_/NOCONTIGUOUS
```

A secondary file must be installed to take effect (see Section 13.1.11)

The system manager does not have to issue this command directly for the creation of primary files, but can use the paging, swapping, and dump files provided with the software distribution kit, or issue the command procedure SYS$SYSDISK:[SYSUPD]SWAPFILES to create new primary files.

13.1.7 **DISABLE Command**

DISABLE inhibits range checks on parameter values specified in SET commands. It takes the form:

```
DISABLE CHECKS
```

Initially, checks are enabled. If the system manager specifies a value above the maximum, the system sets the parameter to the maximum and issues an error message. If the system manager specifies a value below the minimum, the system sets the parameter to the minimum and issues an error message. In the following example, the system manager, foiled in the initial attempt to set WSMAX below the minimum, disables range checks:

```
SYSGEN>  SET WSMAX 20
%SYSGEN-W-Value set to minimum
SYSGEN>  DISABLE CHECKS
SYSGEN>  SET WSMAX 20
```

13.1.8 **ENABLE Command**

ENABLE ensures that range checks are in effect. It takes the form:

```
ENABLE CHECKS
```

Initially, range checks are enabled, so that this command need be used only when a DISABLE command occurs earlier in the session.

SYSGEN (Cont.)

13.1.9 EXIT Command

EXIT returns the system manager to DCL command level. It takes the form:

```
EXIT
```

The system manager can also return to DCL command level by typing <CTRL/Z>.

13.1.10 HELP Command

HELP lists and explains the SYSGEN commands. It takes the form:

```
HELP [command-name [qualifier]]
```

command-name
    Name of a SYSGEN command

qualifier
    Name of a SYSGEN qualifier

If no command name is specified, HELP displays general information on all commands. Specification of command names and qualifiers obtains more detailed information.

13.1.11 INSTALL Command

INSTALL activates a secondary paging or swapping file. It takes the form:

```
INSTALL file-spec {/PAGEFILE}
                  {/SWAPFILE}
```

file-spec
    Name of a secondary paging or swapping file created with a CREATE command

Either /PAGEFILE or /SWAPFILE must be specified to indicate the type file.

All processes started after entry of the INSTALL command use the new paging or swapping file. Current processes continue using the old file.

The following example installs a secondary file:

```
SYSGEN> INSTALL DRA5:[SYSTEM]PAGEFILE.SYS /PAGEFILE
```

The new paging or swapping file is effective until system shutdown.

Use of the INSTALL command requires the CMKRNL privilege.

13.1.12 LOAD Command

LOAD loads an I/O driver. It takes the form:

LOAD file-spec

file-spec
File specification of the driver image; file type defaults to EXE

If the file specification is the same as a driver already loaded, no load takes place. If the file name is the same as a driver already loaded (but the file specification is different), the specified driver replaces the existing driver.

DIGITAL supplies drivers for the standard devices, as listed in Table 13-2 (see Section 13.2.2).

The following example loads the standard driver for a remote terminal:

```
SYSGEN>  LOAD SYS$SYSTEM:RTTDRIVER
```

The VAX/VMS Guide to Writing a Device Driver contains more detailed information on loading a driver.

Use of the LOAD command requires the CMEXEC and CMKRNL privileges.

13.1.13 RELOAD Command

RELOAD replaces a loaded device driver with a new version. It takes the form:

RELOAD file-spec

file-spec
File specification of the new driver image; file type defaults to EXE

The specified image is loaded and replaces any existing driver with the same file specification. The following example reloads the remote terminal driver:

```
SYSGEN>  RELOAD SYS$SYSTEM:RTTDRIVER
```

Use of the RELOAD command requires the CMEXEC and CMKRNL privileges.

SYSGEN (Cont.)

13.1.14 SET (Parameter) Command

SET (parameter) assigns a value to a system parameter as it exists in the SYSGEN work area. The command takes the form:

```
SET parameter-name value
```

parameter-name
: Name of a system parameter as specified in Tables 12-1 through 12-6 (see Chapter 12)

value
: An integer or the keyword DEFAULT; integer values must be within the minimum and maximum values unless DISABLE CHECKS was specified; DEFAULT means the default value for the parameter

This command does not modify parameter files, the current system image on disk, or the active system. See the WRITE command below. The following example assigns a value of 20 to the PFCDEFAULT parameter:

```
SYSGEN>  SET PFCDEFAULT 20
```

The next example assigns the default value (40) to the GBLSECTIONS parameter:

```
SYSGEN>  SET GBLSECTIONS DEFAULT
```

13.1.15 SET (Start-Up Command Procedure) Command

SET (start-up command procedure) names the site-independent start-up command procedure to be associated with a parameter file for subsequent bootstrap operations. It takes the form:

```
SET /STARTUP file-spec
```

file-spec
: File specification of a start-up command procedure on the system disk (maximum of 31 characters)

The SET command does not modify parameter files or the current system image on disk. See the WRITE command below.

The following example assigns SYS$SYSTEM:XSTARTUP.COM as the current site-independent start-up command procedure:

```
SYSGEN>  SET /STARTUP SYS$SYSTEM:XSTARTUP.COM
```

The initial (as named in the software distribution kit) site-independent start-up command procedure is SYS$SYSTEM:STARTUP.COM.

13.1.16 SHARE (Initialize) Command

SHARE (initialize) initializes a multiport memory unit, if it is not already initialized, and connects it to the processor on which SYSGEN is running. The command takes the form:

```
SHARE MPMn mpm-name /INIT
                    [/GBLSECTIONS=gbl]
                    [/MAILBOXES=mail]
                    [/CEFCLUSTERS=cef]
                    [/MAXGBLSECTIONS=max-gbl]
                    [/MAXMAILBOXES=max-mail]
                    [/MAXCEFCLUSTERS=max-cef]
                    [/POOLBCNT=block-cnt]
                    [/POOLBSIZE=block-size]
                    [/PRQCOUNT=prq-cnt]
                    [/START=start]
```

n
: Number on the front panel of the multiport memory unit being initialized

mpm-name
: Name by which the multiport memory unit is to be known to systems using it; consists of 1 to 15 alphanumeric characters, dollar signs, or underlines

gbl
: Total global sections permitted in the multiport memory unit; defaults to 32

mail
: Total mailboxes permitted in the multiport memory unit; defaults to 32

cef
: Total common event flag clusters permitted in the multiport memory unit; defaults to 32

max-gbl
: Maximum global sections that this processor can create in the multiport memory unit; defaults to no limit

max-mail
: Maximum mailboxes that this processor can create in the multiport memory unit; defaults to no limit

max-cef
: Maximum common event flag clusters that this processor can create in the multiport memory unit; defaults to no limit

block-cnt
: Number of blocks allocated to the multiport memory unit's dynamic pool; defaults to 128 blocks

block-size
: Size of each block in the dynamic pool; defaults to 128 bytes

SYSGEN (Cont.)

prq-cnt
: Number of interprocess or request blocks (PRQs) allocated; defaults to 64

start
: Relative page number of first page in the multiport memory unit for use by VMS data structures; defaults to 1

The dynamic pool is used for AST control blocks, message buffers, and other small dynamic structures. The start specification permits space to be left at the front of the multiport memory unit for the dedicated use of one processor or for non-VAX/VMS use.

If the specified multiport memory unit is already initialized and connected to other active processors, the gbl, mail, cef, block-cnt, block-size, prq-cnt, and start parameter values are ignored, and the unit is simply connected to the processor.

The following example initializes a multiport memory unit with defaults on all but the gbl, mail, and cef parameters:

```
SYSGEN> SHARE MPM1 SHR_MEM1 /INIT-
_/GBLSECTIONS=128 /MAILBOXES=64 /CEFCLUSTERS=0
```

The number of the multiport memory unit as it appears on the front panel is 1. The system manager names the unit SHR_MEM_1.

Use of the SHARE command requires the CMKRNL and CMEXEC privileges.

13.1.17 SHARE (Connect) Command

SHARE (connect) connects a processor to a multiport memory unit already initialized by this or another processor. The command takes the form:

```
SHARE MPMn mpm-name [/MAXGBLSECTIONS=max-gbl]
                    [/MAXMAILBOXES=max-mail]
                    [/MAXCEFCLUSTERS=max-cef]
```

n
: Number on the front panel of the multiport memory unit being connected

mpm-name
: Name of the multiport memory unit as specified in a previous SHARE (initialize) command

max-gbl
: Maximum global sections that this processor can create in the multiport memory unit; defaults to no limit

max-mail
: Maximum mailboxes that this processor can create in the multiport memory unit; defaults to no limit

max-cef
: Maximum common event flag clusters that this processor can create in the multiport memory unit; defaults to no limit

The number and name of the multiport memory unit must match those of an initialized unit or an error condition results.

The following example connects a multiport memory unit with defaults on the parameters:

```
SYSGEN>  SHARE MPM1 SHR_MEM_1
```

The unit with a 1 on the front panel must be initialized with the name SHR_MEM_1 for the command to work.

Use of the SHARE command requires the CMKRNL and CMEXEC privileges.

13.1.18 SHOW (Parameter) Command

SHOW (parameter) displays the values of system parameters in the SYSGEN work area, plus the default, minimum, and maximum values of the parameters, and their units of measure. It takes the form:

```
        ⎧ parameter-name ⎫
        ⎪ /ACP           ⎪
        ⎪ /ALL           ⎪
        ⎪ /DYNAMIC       ⎪
        ⎪ /GEN           ⎪
SHOW   <  /JOB            >
        ⎪ /MAJOR         ⎪
        ⎪ /NAMES         ⎪
        ⎪ /PQL           ⎪
        ⎪ /RMS           ⎪
        ⎩ /SYS           ⎭
```

parameter-name
: Name of a system parameter as specified in Tables 12-1 through 12-6 (see Chapter 12)

The system manager can specify only one parameter name or one of the qualifiers. Specification of a parameter name displays the values of that parameter. Specification of a qualifier displays the values of all ACP parameters (/ACP), all parameters (/ALL), all DYNAMIC parameters (/DYNAMIC), all GEN parameters (/GEN), all JOB parameters (/JOB), all MAJOR parameters (/MAJOR), all PQL parameters (/PQL), all RMS parameters (/RMS), all SYS parameters (/SYS), or just the names of all parameters (/NAMES).

NOTE

> When parameter names are abbreviated, the first parameter matching the abbreviation is selected for display. No ambiguity checks are made. For example, a specification of SHOW GBL displays the GBLSECTIONS parameter. A display of the GBLPAGES parameter requires a specification of (at least) SHOW GBLP.

SYSGEN (Cont.)

Figure 13-1 illustrates the SHOW command.

```
SYSGEN>  SHOW /MAJOR

Parameter Name       Current   Default   Minimum   Maximum  Unit
--------------       -------   -------   -------   -------  ----
PFCDEFAULT                16        16         0       127  PAGES
GBLSECTIONS               40        40        20        -1  SECTIONS
GBLPAGES                2048      2048       512        -1  PAGES
MAXPROCESSCNT             64        64        12       256  PROCESSES
SYSMWCNT                  48        48        20     16384  PAGES
BALSETCNT                 24        24         4      1024  SLOTS
IRPCOUNT                  80        80         0     32768  PACKETS
WSMAX                    256       256        60     16384  PAGES
NPAGEDYN               40448     40448     16384        -1  BYTES
PAGEDYN                 8192      8192      8192        -1  BYTES
VIRTUALPAGECNT          8192      8192       512     65536  PAGES
QUANTUM                   30        30         2     32767  10MS
MPW_WRTCLUSTER            16        16         0       127  PAGES
MPW_HILIM                 24        24         0     16384  PAGES
MPW_LOLIM                 12        12         0     16384  PAGES
```

Figure 13-1 Display Produced by SHOW Command of SYSGEN

13.1.19 SHOW (Device) Command

SHOW (device) displays information on device drivers loaded into the system, the devices connected to them, and their I/O data bases. It takes the form:

SHOW /DEVICE[=driver-name]

driver-name
: Name of the driver; defaults to all device drivers loaded into the system

The specific information displayed is as follows:

- Driver -- Name of the driver
- Start -- Starting address of the driver
- End -- Ending address of the driver
- Dev -- Name of each device connected to the driver
- DDB -- Address of the device's device data block
- CRB -- Address of the device's channel request block
- IDB -- Address of the device's interrupt dispatch block
- Unit -- Number of each unit on the device
- UCB -- Address of each unit's unit control block

SYSGEN (Cont.)

All addresses are in hexadecimal and are virtual. Figure 13-2 illustrates the SHOW /DEVICE command.

Use of the SHOW /DEVICE command requires the CMEXEC privilege.

```
SYSGEN>  SHOW /DEVICE=DBDRIVER
 Driver      Start     End    Dev   DDB      CRB      IDB      Unit  UCB
DBDRIVER   80082390 80082A7E
                              DBA 80000848 800988C0 80098920
                                                                0 8000087C
                                                                1 8008A4F0
                                                                2 8008A590
                                                                5 8008A630
                                                                7 8008A6D0
```

Figure 13-2 Display Produced by SHOW /DEVICE Command of SYSGEN

13.1.20 **SHOW (Start-Up) Command**

SHOW (start-up) displays the name of the current site-independent start-up command procedure. It takes the form:

SHOW /STARTUP

Figure 13-3 illustrates the SHOW (start-up) command.

```
SYSGEN>  SHOW /STARTUP

 Startup command file = SYS$SYSTEM:STARTUP.COM
```

Figure 13-3 Display Produced by SHOW /STARTUP Command of SYSGEN

13.1.21 **USE Command**

USE initializes the SYSGEN work area with system parameter values and the name of the site-independent start-up command procedure, from a parameter file, the current system image on disk, the active system in memory, or the default list. It takes the form:

```
     ⎧ file-spec ⎫
USE  ⎨ CURRENT   ⎬
     ⎪ ACTIVE    ⎪
     ⎩ DEFAULT   ⎭
```

file-spec
    File specification of a system parameter file; file type defaults to PAR

SYSGEN (Cont.)

The system manager specifies one parameter file or one of the keywords. The parameter file must be either one of the standard system parameter files supplied by DIGITAL (see Chapter 12, Table 12-7) or a file created by the system manager with the WRITE command of SYSGEN. Existing values in the SYSGEN work area are overwritten. The following example initializes the SYSGEN work area with the parameter values of the 64USER standard system parameter file:

```
SYSGEN> USE SYS$SYSTEM:64USER
```

The initial values of the SYSGEN work area when the utility is invoked are the default values.

13.1.22 WRITE Command

WRITE writes the system parameter values and the name of the site-independent start-up command procedure from the SYSGEN work area to a parameter file, the current system image on disk, or the active system in memory. (However, only the dynamic parameter values are written to the active system.) The command takes the form:

```
        { file-spec }
WRITE   {  CURRENT  }
        {  ACTIVE   }
```

file-spec
    File specification; file type defaults to PAR

Specification of a file results in the creation of a new parameter file, as illustrated in the following example:

```
SYSGEN> WRITE SYS$SYSTEM:SPECIAL
```

The next example modifies the current system image on disk (SYS$SYSTEM:SYS.EXE):

```
SYSGEN> WRITE CURRENT
```

Use of the WRITE ACTIVE command requires the CMKRNL privilege.

13.1.23 Error Messages

See the VAX/VMS System Messages and Recovery Procedures Manual for the messages issued by the SYSGEN utility.

13.2 OPERATIONS

The system manager uses SYSGEN differently depending on the type of function being performed.

13.2.1 System Parameters

The bootstrap process initializes the active system parameter values in memory from the current system image on disk (that is, the starting parameter values are those in SYS$SYSTEM:SYS.EXE). In a conversational bootstrap operation, the operator can modify these values by reinitializing the active parameter values from any parameter file or the default list, and by setting new parameter values on an individual basis. The operator performs these activities with a subset of the SYSGEN commands, as explained in the VAX-11 Software Installation Guide. At the end of the bootstrap operation, the system image is modified to conform to the active parameter values.

After the system is booted and running, the system manager can run SYSGEN to create or modify parameter files, modify the current system image, and modify the dynamic parameter values of the active system. The system manager typically uses the following sequence of commands:

- Invokes SYSGEN -- Invoking SYSGEN initializes a work area to the default parameter values.
- Optionally issues a USE command -- The system manager can reinitialize the work area to the values of a parameter file, the current system image, or the active system, if the default values do not provide a suitable base for subsequent operations.
- Issues SET commands -- The system manager modifies parameters on an individual basis. These modifications have no effect outside the SYSGEN work area.
- Issues a WRITE command -- The system manager creates a parameter file, modifies the current system image on disk, or modifies the active system on disk.

During these operations, the system manager can use the SHOW command to examine the parameter values in the SYSGEN work area.

13.2.1.1 **Create a Parameter File** - The creation of a new parameter file does not immediately affect system performance. At subsequent conversational bootstrap operations, however, the operator can initialize the active system with the values of the new file. In the following example, the system manager creates a new version of the 64USER standard system parameter file with a new value for the PFCDEFAULT parameter:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN SYSGEN
SYSGEN>  USE 64USER
SYSGEN>  SET PFCDEFAULT 52
SYSGEN>  WRITE 64USER
SYSGEN>  EXIT
```

The next example creates a user file named SYS$SYSTEM:52USER.PAR, using the 64USER file as a base:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN SYSGEN
SYSGEN>  USE 64USER
SYSGEN>  SET PFCDEFAULT 52
SYSGEN>  WRITE 52USER
SYSGEN>  EXIT
```

13.2.1.2 **Modify the System Image** - The modification of the current system image also does not immediately affect system performance. At subsequent bootstrap operations, however, the active system is initialized with the new values. A conversational bootstrap operation permits the operator to modify these values further, while a nonstop bootstrap operation makes the new values the values of the active system. The following example modifies the PFCDEFAULT parameter value in the system image:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  USE CURRENT
SYSGEN>  SET PFCDEFAULT 52
SYSGEN>  WRITE CURRENT
SYSGEN>  EXIT
```

13.2.1.3 **Modify the Active System** - Modification of the active system immediately affects that subset of the system parameters called the dynamic parameters by changing their values in the active system in memory. Chapter 12 lists the dynamic parameters (as does the SHOW /DYNAMIC command). The other parameters cannot be changed immediately because they regulate structures that cannot be changed once the system is running. The following example modifies the active value of the PFCDEFAULT parameter:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  USE ACTIVE
SYSGEN>  SET PFCDEFAULT 52
SYSGEN>  WRITE ACTIVE
SYSGEN>  EXIT
```

Modification of the active system does not affect the current system image on disk. If, for example, the system manager sets new active parameter values (WRITE ACTIVE) and later wants to use these values for subsequent bootstrap operations, the values must be explicitly written to the current system image on disk:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  USE ACTIVE
SYSGEN>  WRITE CURRENT
SYSGEN>  EXIT
```

13.2.2 Devices and Device Drivers

Normally, the system manager issues the AUTOCONFIGURE command to automatically connect all devices physically attached to the system and load their device drivers, saving a great deal of effort and reducing the possibility of error. Devices not attached to the system and devices with nonstandard names can be connected and their device drivers loaded with explicit CONNECT (or CONNECT and LOAD) commands. Great care must be exercised in issuing CONNECT and LOAD commands -- see the VAX/VMS Guide to Writing a Device Driver.

Devices not connected automatically by AUTOCONFIGURE include the network communications logical device and the console block storage device. Connecting the network communications logical device requires the following explicit CONNECT command:

```
CONNECT NET /NOADAPTER /DRIVER=NETDRIVER
```

Connecting the console block storage device requires the following explicit CONNECT command:

```
CONNECT CONSOLE
```

The following example autoconfigures the devices physically attached to the system and explicitly connects the network software device and the console block storage device:

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

Normally, the SYSGEN commands for connecting devices and loading device drivers are included in the site-independent start-up command procedure (See Chapter 8).

Table 13-2 names the standard device types and their device drivers as supplied by DIGITAL. The drivers reside in files named SYS$SYSTEM:file-name.EXE, except for DZDRIVER which resides in SYS$SYSTEM:TTDRIVER.EXE.

Another DIGITAL-supplied driver named CONINTERR (which resides in SYS$SYSTEM:CONINTERR.EXE) permits real-time processes to connect to interrupt vectors for quick response to and special handling of real-time events. The driver is not associated with any one device type. See the VAX/VMS Real-Time User's Guide for further information.

Table 13-2
Standard Device Types and Drivers

Name	Device Type	Driver
CR	Card reader	CRDRIVER
CS	Console block storage device	DXDRIVER
DB	RP05, RP06 disk	DBDRIVER
DL	RL02 cartridge disk	DLDRIVER
DM	RK06, RK07 cartridge disk	DMDRIVER
DR	RM03	DRDRIVER
DY	RX02 floppy diskette	DYDRIVER
LA	LPA11-K laboratory peripheral accelerator	LADRIVER
LP	Line printer	LPDRIVER
MB	Shared memory mailbox	MBXDRIVER
MS	TS11 magnetic tape	TSDRIVER
MT	TE16, TU45, TU77 magnetic tape	TMDRIVER
NET	Network communications logical device	NETDRIVER
OP	Operator's console	OPERATOR
RT	Remote terminal	RTTDRIVER
TT	Interactive terminal	DZDRIVER
XF	DR32 interface adapter	XFDRIVER
XJ	DUP11 synchronous communications line	*
XM	DMC11 synchronous communications line	XMDRIVER

* Available under separate license

13.2.3 System Files

Table 13-3 provides suggested sizes for the paging, swapping, and dump files for each hardware configuration. The full file specification of each file is SYS$SYSTEM:file-name.type. Sizes are expressed in pages.

Table 13-3
Suggested Sizes for System Files

System File	MINIMUM	8USER	16USER	32USER	48USER	64USER	VIRT32MB
PAGEFILE.SYS	8192	8192	16384	32768	61440	98304	98304
SWAPFILE.SYS	3072	5120	7168	24576	47600	86016	28672
SYSDUMP.DMP	516	1028	2052	3076	4100	6198	2052

The VAX/VMS software distribution kit creates system files suitable for systems configured with the MINIMUM and 8USER parameter files. For the other configurations, or for variant configurations, the system manager must specify the file sizes with the CREATE command of the SYSGEN utility or (to create primary files only) with a DIGITAL command procedure called SYS$SYSDISK:[SYSUPD]SWAPFILES.COM. The following example uses the CREATE command to create files suitable for a system configured with the 16USER parameter file:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN SYSGEN
SYSGEN>  CREATE PAGEFILE.SYS /SIZE=16384
SYSGEN>  CREATE SWAPFILE.SYS /SIZE=7168
SYSGEN>  CREATE SYSDUMP.DMP /SIZE=2052
SYSGEN>  EXIT
```

The next example creates the same files using the command procedure:

```
$ @SYS$SYSDISK:[SYSUPD]SWAPFILES
To leave a file size at its current value type a carriage  return
<CR> in response to its file size.  Current file sizes are:

DIRECTORY DB0:[SYSEXE]
22-JAN-80 09:59

PAGEFILE.SYS;1     8192.   C    21-JAN-80 09:23
SWAPFILE.SYS;1     3072.   C    21-JAN-80 09:23
SYSDUMP.SYS;1      516.    C    21-JAN-80 09:23

TOTAL OF 11788./11788. BLOCKS IN 3. FILES

Enter new size for paging file: 16384
Enter new size for swapping file: 7168
Enter new size for system dump file: 2052
```

Use of the command procedure or the CREATE command results in a new version of the system file. The old version must be deleted explicitly (but not until after the next bootstrap operation). In the case of a primary file (PAGEFILE.SYS, SWAPFILE.SYS, or SYSDUMP.DMP), the new file does not become effective until the system is shut down and restarted. In the case of a secondary file, the new file becomes effective when it is installed. No more than two versions of the primary paging file (PAGEFILE.SYS) can exist at one time.

The system manager can verify the suggested sizes through the following calculations:

- Paging file -- Size of average program (in pages) times the maximum number of processes (MAXPROCESSCNT system parameter). Program size statistics can be gathered by looking at the peak virtual sizes when users log off with the LOGOUT /FULL command, or by examining user accounting records. User program sizes can be limited with the /PGFLQUOTA qualifier of the ADD and MODIFY commands in the AUTHORIZE utility (see Chapter 2). The suggested file sizes assume an average program size of 1024 pages. The system manager should not reduce the value of /PGFLQUOTA below 1024, nor reduce the size of the paging file below the suggested value for the configuration, as sufficient space in the paging file is critical to system performance. Size requirements can vary depending on the user applications, as the paging file is used only for copy-on-reference and demand-zero pages.

- Swapping file -- Maximum number of processes (MAXPROCESSCNT system parameter) times maximum working set size (WSMAX system parameter). The system manager should not specify a file size less than this value, as the system will reduce the value of MAXPROCESSCNT until MAXPROCESSCNT times WSMAX is equal to or less than the size of the swapping file.

- Dump file -- Size of physical memory in pages (to save the contents of memory if the system fails) plus 4 pages (to provide continuity of the error log when the system is shutdown or if the system fails).

At bootstrap time, the system activates the latest versions of SYS$SYSTEM:PAGEFILE.SYS, SWAPFILE.SYS, and SYSDUMP.DMP. After bootstrap, the system manager can replace or augment the primary paging or swapping file with a secondary file (previously created with a CREATE command) by issuing an INSTALL command. The advantages to using a secondary file are that it does not have to be on the system disk and it can span volumes in a volume set. Where secondary files are the norm, the system manager should:

- Use the MINIMUM sizes for PAGEFILE.SYS and SWAPFILE.SYS

- Include SYSGEN INSTALL commands for the secondary files in the site-specific start-up command procedure (see Chapter 8)

The system recalculates the maximum allowable value for the MAXPROCESSCNT system parameter upon installation of a secondary swapping file.

All processes created after installation of the secondary paging file use that file, while all processes created before its installation continue to use the primary paging file. Swapping slots are allocated (at process creation) from whichever swapping file has space, starting with the primary.

Installation of a secondary paging file requires nonpaged dynamic memory for a bit map, just as the primary paging file does, equal to (in bytes) the number of pages in the paging file divided by 8.

13.2.4 **Start-Up Command Procedure**

Chapter 8 describes the site-independent start-up command procedure supplied by DIGITAL -- SYS$SYSTEM:STARTUP.COM. The system manager can create alternate site-independent start-up command procedures; for example, by copying STARTUP.COM to other files of type COM and editing those files.

Following a bootstrap operation, the system executes the current site-independent start-up command procedure. Initially (that is, as supplied in the software distribution kit), the current site-independent start-up command procedure is STARTUP.COM. The system manager can name an alternate site-independent start-up command procedure to be used during subsequent bootstrap operations with the SET /STARTUP command in SYSGEN. The SHOW /STARTUP command displays the current site-independent start-up command procedure. The following example displays the current site-independent start-up command procedure, then specifies an alternate:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE CURRENT
SYSGEN>  SHOW /STARTUP

 Startup command file = SYS$SYSTEM:STARTUP.COM
SYSGEN>  SET /STARTUP SYS$SYSTEM:XSTARTUP.COM
SYSGEN> WRITE CURRENT
SYSGEN>  EXIT
```

13.2.5 Multiport (Shared) Memory

A single processor can attach up to four multiport memory units, each of which may vary in size from 256KB to 2MB. The front panel of each multiport memory unit displays the number of that unit. When the system manager issues a SHARE (initialize) command, SYSGEN polls the processors with ports on the specified multiport memory unit. If no other processors are using the unit, the unit is initialized. If another processor is using the unit, the unit is connected. A SHARE (connect) command for an uninitialized multiport memory unit results in an error condition.

The system manager should power down a multiport memory unit only after first shutting down and rebooting all systems connected to the unit.

A multiport memory unit managed by VAX/VMS contains the following structures with space requirements as indicated:

- Pointer pages -- A common data page plus one page for each active port

- Global section description (GSD) table -- Total global sections, as specified in the SHARE command, times 96 bytes, rounded up to the next full page

- Mailbox table -- Total mailboxes, as specified in the SHARE command, times 48 bytes, rounded up to the next full page

- CEF table -- Total common event flag clusters, as specified in the SHARE command, times 80 bytes, rounded up to the next full page

- PRQ pool -- Total interprocess or request messages, as specified in the SHARE command, times 64 bytes, rounded up to the next full page

- Dynamic pool -- Number of blocks allocated to the pool, as specified in the SHARE command, times the size of each block, as specified in the SHARE command

- Non-VAX/VMS structures -- Relative number of first page in multiport memory for use by shared structures, as specified in the SHARE command, minus 1

- Global section bit map -- One page

- Global sections -- Size of multiport memory unit minus the sum of the above preallocated structures (that is, the remaining space)

Figure 13-4 illustrates the appearance of a multiport memory unit where the SHARE command specifies the default structure values, assuming a 256KB unit with four active ports.

```
SYSGEN> SHARE MPM1 SHR_MEM_1 /INIT-
 /GBLSECTIONS=32    /MAILBOXES=32-
 /CEFCLUSTERS=32    /POOLBCNT=128-
 /POOLBSIZE=128     /PRQCOUNT=64
```

POINTER PAGES	1 + 4	= 5 pages
PRQ POOL	64 x 64	= 8 pages
DYNAMIC POOL	128 x 128	= 32 pages
GSD TABLE	32 x 96	= 1 page
MAILBOX TABLE	32 x 48	= 1 page
CEF TABLE	32 x 8	= 1 page
BIT MAP	1 page	
GLOBAL SECTIONS	512 - 49	= 463 pages

← relative page 1

Figure 13-4 Example of Multiport Memory Structures

The following guidelines are suggested in selecting values for the SHARE qualifiers that regulate the sizes of the preallocated structures:

- /CEFCLUSTERS, /GBLSECTIONS, and /MAILBOXES -- The system manager should simply specify the maximum number of each type of structure required by all processors at any one time. The same structure being used by many processes on one or more processors counts as just one structure.

- /POOLBCNT and /POOLBSIZE -- The primary use of the dynamic pool is to buffer mailbox messages. The size of a message is 28 bytes plus the data in the message. Since space from the pool is always allocated in whole blocks, the recommended block size is the median message size plus 28. A block size that is too small for a message requires extra system overhead to concatenate the message blocks into the user buffer and segment them out of the user buffer. The number of blocks should be sufficient to satisfy all messages that might be outstanding at once. If a mailbox request cannot be satisfied due to insufficient pool space, the requesting process enters a resource wait state or the request fails (if resource wait mode is not enabled), just as if the nonpaged dynamic pool were depleted. For this reason, the system manager should tend to overestimate space requirements in the dynamic pool.

- /PRQCOUNT -- The system uses interprocessor request blocks internally to transfer requests among the VAX/VMS executive routines and mailbox drivers on the different processors. PRQs are allocated and deallocated rapidly, so that a large number should not be needed. The default value normally suffices. If an executive or driver request cannot be satisfied due to depletion of the PRQs, the requesting routine waits until a PRQ becomes available.

The system manager should calculate the space remaining for the global sections and be satisfied that it is sufficient. If the space is insufficient, the system manager might reduce the size of the dynamic pool. However, this condition really suggests the need for a larger or additional multiport memory unit.

Where a multiport memory unit is a normal part of the system configuration, the system manager should include the SYSGEN commands to initialize and connect it in the site-specific start-up command procedure (see Section 8.2.9).

CHAPTER 14

DISPLAY UTILITY

The system manager runs the DISPLAY utility to display information on system performance. Table 14-1 summarizes the DISPLAY commands. Use of the utility is restricted to VT52, VT55, and VT100 terminals. (VT100 terminals automatically operate in VT52 emulation mode.) Print files (for example, assigning a file to SYS$OUTPUT) are not permitted.

Table 14-1
DISPLAY Command Summary

Format	Function
<CTRL/C>	Returns the user to DISPLAY command level
CYCLE	Produces all displays cyclically, except for the FILE PRIMITIVE STATISTICS display
EXIT	Returns the user to DCL command level
FCP	Produces the FILE PRIMITIVE STATISTICS display
HELP	Lists the DISPLAY commands
IORATES	Produces the I/O SYSTEM RATES display
M2	Produces the TIME IN PROCESSOR MODES display
M5	Produces the TIME IN PROCESSOR MODES display as a bar graph (VT55 terminals only)
PAGE	Produces the PAGE MANAGEMENT STATISTICS display
POOL	Produces the NONPAGED POOL STATISTICS display
S2	Produces the NUMBER OF PROCESSES IN SCHEDULER STATES display
S5	Produces the NUMBER OF PROCESSES IN SCHEDULER STATES display as a bar graph (VT55 terminals only)
TOPUSERS	Produces the TOP CPU TIME USERS display
USERS	Produces the VAX/VMS PROCESSES display

DISPLAY

14.1 INVOKING DISPLAY

The following command invokes the utility:

```
$ RUN SYS$SYSTEM:DISPLAY
```

The system responds with the following prompt:

```
DISPLAY NAME?
```

The system manager can then enter any of the commands listed in Table 14-1. The command names can be abbreviated to their first two characters (or their first characters if the abbreviation is unique). If the command is a request for a display (all commands except <CTRL/C>, EXIT, and HELP), the system responds with the following prompt:

```
UPDATE INTERVAL IN SECONDS?
```

The system manager responds with a positive integer to represent the number of seconds over which statistics for the display are gathered. At the end of each interval, the system posts the results to the display. For example, if the system manager specifies an interval of 10 seconds, the system:

1. Displays the headings of the display
2. Collects statistics for 10 seconds
3. Posts the results to the display
4. Collects statistics for another 10 seconds
5. Posts the new results to the display
6. And so on until the system manager types <CTRL/C>

A carriage return results in the default interval of 3 seconds. (The CYCLE command is an exception; see Section 14.1.3).

The top of each display contains the time in the format hours:minutes:seconds that the last statistics were posted.

14.2 CTRL/C COMMAND

Typing <CTRL/C> (that is, pressing the CTRL key and, while holding it down, typing C) terminates the current display and returns the user to DISPLAY command level. The screen is cleared and the prompt DISPLAY NAME? appears.

14.3 CYCLE COMMAND

CYCLE cyclically produces the following displays in the order indicated:

1. TIME IN PROCESSOR MODES (VT55 terminals only)
2. VAX/VMS PROCESSES
3. TOP CPU TIME USERS
4. PAGE MANAGEMENT STATISTICS
5. NUMBER OF PROCESSES IN SCHEDULER STATES (VT55 terminals only)
6. I/O SYSTEM RATES
7. TIME IN PROCESSOR MODES
8. NONPAGED POOL STATISTICS
9. NUMBER OF PROCESSES IN SCHEDULER STATES

The command takes the form:

```
CYCLE
```

In the case of the CYCLE command, the response to the interval prompt means the number of seconds between different displays, and defaults to 30. The update interval is fixed at 3 seconds.

The following example produces the CYCLE displays with 20 seconds between each display:

```
DISPLAY NAME?  CYCLE

INTERVAL BETWEEN DISPLAYS IN SECONDS?  20
```

14.4 EXIT COMMAND

EXIT returns the system manager to DCL command level. It takes the form:

```
EXIT
```

14.5 FCP COMMAND

FCP produces the FILE PRIMITIVE STATISTICS display. It takes the form:

```
FCP
```

DISPLAY (Cont.)

The FILE PRIMITIVE STATISTICS display measures the following activities for all Files-11 ACPs on the system:

- FCP CALLS -- QIO requests received by the system
- ALLOCATIONS -- Calls that caused allocation of disk space
- CREATES -- New files created
- DISK READS -- Read I/O operations from disk by the file system
- DISK WRITES -- Write I/O operations to disk by the file system
- CACHE HITS -- Requested blocks located in the file system cache
- CPU TICS -- CPU time used by the file system in 10 millisecond tics
- WINDOW TURNS -- File mapping window misses
- FILE LOOKUPS -- File name look-up operations in file directories
- FILE OPENS -- Files that were opened

The display provides three measurements for each activity:

- VALUE -- Number of times the activity occurred during the preceding interval
- RATE/SEC -- Average number of times the activity occurred per second during the preceding interval
- AVG RATE -- Average number of times the activity occurred per second since the display began

The following example produces a FILE PRIMITIVE STATISTICS display with an interval of 20 seconds:

```
DISPLAY NAME? FCP

UPDATE INTERVAL IN SECONDS? 20
```

Normally, a lengthy interval provides better results, because the sample is larger. If a short interval is used, the system manager can allow several intervals to pass and watch the AVG RATE statistics.

Figure 14-1 illustrates a FILE PRIMITIVE STATISTICS display. From this display, the system manager might, for example, learn that over the past interval the system performed 4 read operations from disk for an average rate of .97 reads per second during the interval and .85 reads per second since the display started.

FILE PRIMITIVE STATISTICS
14:27:27

NAME	VALUE	RATE /SEC	AVG RATE	NAME	VALUE	RATE /SEC	AVG RATE
FCP CALLS	7	1.70	3.92	CACHE HITS	8	1.94	2.87
ALLOCATIONS	0	0.00	0.04	CPU TICS	10	2.43	2.85
CREATES	0	0.00	0.01	WINDOW TURNS	1	0.24	1.01
DISK READS	4	0.97	0.85	FILE LOOKUPS	7	1.70	2.64
DISK WRITES	0	0.00	0.15	FILE OPENS	1	0.24	0.97

Figure 14-1 FILE PRIMITIVE STATISTICS Display

14.6 HELP COMMAND

HELP lists and explains the DISPLAY commands. It takes the form:

```
HELP
```

14.7 IORATES COMMAND

IORATES produces the I/O SYSTEM RATES display. It takes the form:

```
IORATES
```

The I/O SYSTEM RATES display measures the following activities:

- DIRECT I/Os -- Direct I/O (for example, disk and tape) operations
- BUFFERED I/Os -- Buffered (for example, terminal and line printer) I/O operations
- MAILBOX WRITES -- Write-to-mailbox requests received by the system
- WINDOW TURNS -- File mapping window misses
- LOGNAME TRANS -- Logical name translations

DISPLAY (Cont.)

- FILE OPENS -- Files opened
- PAGE FAULTS -- Page faults for all working sets
- PAGES READ -- Pages read from disk as a result of page faults
- READ I/Os -- Read I/O operations from disk as a result of page faults
- PAGES WRITTEN -- Pages written to the paging file
- WRITE I/Os -- Write I/O operations to the paging file
- TOTAL INSWAPS -- Working sets read into memory from the swapping file

The display provides three measurements for each activity:

- VALUE -- Number of times the activity occurred during the preceding interval
- RATE/SEC -- Average number of times the activity occurred per second during the preceding interval
- AVG RATE -- Average number of times the activity occurred per second since the display began

The top corners of the display contain the number of pages on the free page and modified page lists at post time.

The following example produces an I/O SYSTEM RATES display with an interval of 20 seconds:

```
DISPLAY NAME?  IORATES

UPDATE INTERVAL IN SECONDS?  20
```

Normally, a lengthy interval provides better results, because the sample is larger. If a short interval is used, the system manager can allow several intervals to pass and watch the AVG RATE statistics.

Figure 14-2 illustrates an I/O SYSTEM RATES display. From this display, the system manager might, for example, learn that over the past interval 65 page faults occurred, resulting in 4 pages being read from disk in 2 I/O operations.

The system manager can calculate further statistics from this display:

- Percentage of page faults resulting in a disk I/O operation -- READ I/Os divided by PAGE FAULTS
- Average cluster size for reading pages into working sets -- PAGES READ divided by READ I/Os
- Average cluster size for writing pages from modified page list -- PAGES WRITTEN divided by WRITE I/Os
- Total swapping I/O operations -- TOTAL INSWAPS times 2
- Disk I/O rate -- DIRECT I/Os plus READ I/Os plus WRITE I/Os plus total swapping I/O operations (TOTAL INSWAPS times 2) allowing for any DIRECT I/Os to tape

FREE LIST: 2016 — I/O SYSTEM RATES 16:17:09 — MODIFY LIST: 35

NAME	VALUE	RATE /SEC	AVG RATE	NAME	VALUE	RATE /SEC	AVG RATE
DIRECT I/Os	32	7.30	1.50	PAGE FAULTS	65	14.84	1.83
BUFFERED I/Os	29	6.62	3.24	PAGES READ	4	0.91	0.11
MAILBOX WRITES	0	0.00	0.00	READ I/Os	2	0.45	0.07
WINDOW TURNS	3	0.68	0.14	PAGES WRITTEN	0	0.00	0.00
LOGNAME TRANS	39	8.90	0.98	WRITE I/Os	0	0.00	0.00
FILE OPENS	3	0.68	0.07	TOTAL INSWAPS	0	0.00	0.00

Figure 14-2 I/O SYSTEM RATES Display

14.8 M2 AND M5 COMMANDS

M2 and M5 produce the TIME IN PROCESSOR MODES display. The M5 command produces the display as a bar graph; its use is restricted to VT55 terminals. The M2 and M5 commands take the forms, respectively:

```
M2

M5
```

The display measures the following activities as percentages of the total CPU time used by the system:

- TIME ON INTERRUPT STACK -- Time spent on interrupt stack (INTER time)
- TIME IN KERNEL MODE -- Time spent in kernel mode, but not on interrupt stack (KERNEL time)
- TIME IN EXEC MODE -- Time spent in executive mode (EXEC time)
- TIME IN SUPER MODE -- Time spent in supervisor mode (SUPER time)
- TIME IN USER MODE -- Time spent in user mode, but not executing compatibility-mode images (USER time)

DISPLAY (Cont.)

- TIME IN COMPATIBILITY MODE -- Time spent executing compatibility-mode images (COMPAT time)

- IDLE TIME -- Time spent executing the NULL process

The system manager can calculate the approximate time spent executing system programs by adding the INTER, KERNEL, EXEC, and SUPER percentages; and the approximate time spent executing user programs by adding the USER and COMPAT percentages. User programs include the DIGITAL editors, compilers, linker, and utilities.

The following example produces a TIME IN PROCESSOR MODES display with an interval of 20 seconds:

```
DISPLAY NAME? M2

UPDATE INTERVAL IN SECONDS? 20
```

Normally, a lengthy interval provides better results, because the sample is larger.

Figure 14-3 illustrates a TIME IN PROCESSOR MODES display in standard format. Figure 14-4 illustrates a TIME IN PROCESSOR MODES display as a bar graph (VT55 terminals only). From the standard display, the system manager might, for example, see that system programs used about 38 percent of the system's CPU time and user programs used about 25 percent.

```
                                TIME IN PROCESSOR MODES(%)
                                         14:30:35

                                          0        25        50        75       100
                                          + - - - - + - - - - + - - - - + - - - - +
                                          :         :         :         :         :
     TIME ON INTERRUPT STACK               ▒▒▒▒
                                          :         :         :         :         :
     TIME IN KERNEL MODE                   ▒▒▒▒▒▒
                                          :         :         :         :         :
     TIME IN EXEC MODE                     ▒▒▒▒▒
                                          :         :         :         :         :
     TIME IN SUPER MODE
                                          :         :         :         :         :
     TIME IN USER MODE                     ▒
                                          :         :         :         :         :
     TIME IN COMPATIBILITY MODE            ▒▒▒▒▒▒▒▒▒
                                          :         :         :         :         :
     IDLE TIME                             ▒▒▒▒▒▒▒▒▒▒▒
                                          :         :         :         :         :
                                          :         :         :         :         :
                                          :         :         :         :         :
                                          :         :         :         :         :
                                          + - - - - + - - - - + - - - - + - - - - +
```

Figure 14-3 TIME IN PROCESSOR MODES Display (M2 Command)

DISPLAY (Cont.)

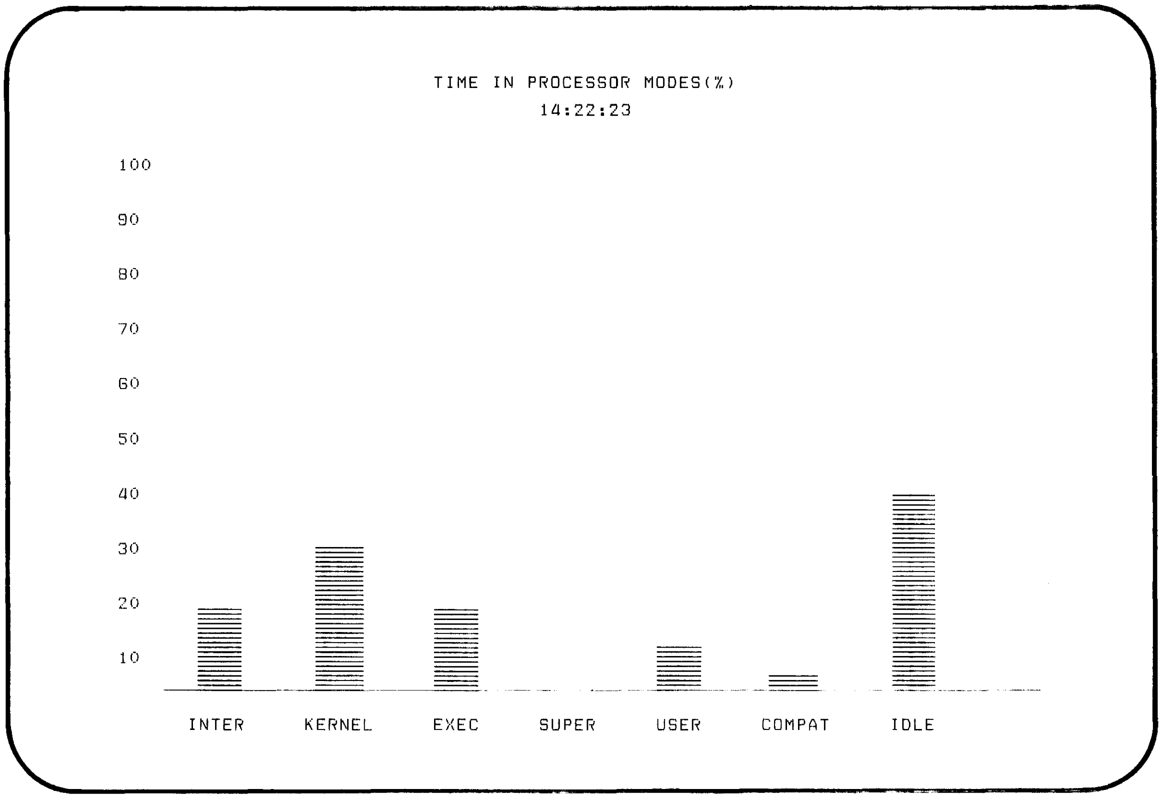

Figure 14-4 TIME IN PROCESSOR MODES Display (M5 Command)

14.9 PAGE COMMAND

PAGE produces the PAGE MANAGEMENT STATISTICS display. It takes the form:

PAGE

The PAGE MANAGEMENT STATISTICS display measures the following activities:

- PAGE FAULTS -- Page faults for all working sets
- PAGES READ -- Pages read from disk as a result of page faults
- READ I/Os -- Read I/O operations from disk as a result of page faults

DISPLAY (Cont.)

- PAGES WRITTEN -- Pages written to the paging file
- WRITE I/Os -- Write I/O operations to the paging file
- FREE LIST -- Pages read from the free page list as a result of page faults
- MODIFIED LIST -- Pages read from the modified page list as a result of page faults
- DEMAND ZERO -- Zero-filled pages allocated as a result of page faults
- WRITE IN PROG -- Pages read that were in the process of being written back to disk, when faulted
- SYSTEM FAULTS -- Page faults for pages in system space

The display provides three measurements for each activity:

- VALUE -- Number of times the activity occurred during the preceding interval
- RATE/SEC -- Average number of times the activity occurred per second during the preceding interval
- AVG RATE -- Average number of times the activity occurred per second since the display began

The top corners of the display contain the number of pages on the free page and modified page lists at post time.

The following example produces a PAGE MANAGEMENT STATISTICS display with an interval of 20 seconds:

```
DISPLAY NAME?  PAGE

UPDATE INTERVAL IN SECONDS?  20
```

Normally, a lengthy interval provides better results, because the sample is larger. If a short interval is used, the system manager can allow several intervals to pass and watch the AVG RATE statistics.

Figure 14-5 illustrates a PAGE MANAGEMENT STATISTICS display. From this display, the system manager might, for example, learn that over the past interval 50 page faults occurred, resulting in 9 pages being read from disk in 3 I/O operations.

DISPLAY (Cont.)

FREE LIST: 181 — PAGE MANAGEMENT STATISTICS 14:25:19 — MODIFY LIST: 19

NAME	VALUE	RATE /SEC	AVG RATE	NAME	VALUE	RATE /SEC	AVG RATE
PAGE FAULTS	50	11.96	8.04	FREE LIST	1	0.23	0.10
PAGES READ	9	2.15	2.08	MODIFIED LIST	3	0.71	1.55
READ I/Os	3	0.71	0.46	DEMAND ZERO	8	1.91	2.44
PAGES WRITTEN	0	0.00	0.42	WRITE IN PROG	0	0.00	0.00
WRITE I/Os	0	0.00	0.02	SYSTEM FAULTS	0	0.00	0.00

Figure 14-5 PAGE MANAGEMENT STATISTICS Display

14.10 POOL COMMAND

POOL produces the NONPAGED POOL STATISTICS display. It takes the form:

POOL

The NONPAGED POOL STATISTICS display measures space allocations in the nonpaged dynamic pool:

- I/O REQUEST PACKETS LEFT -- I/O request packets not in use
- NUMBER OF HOLES IN POOL -- Unused blocks of contiguous space in the dynamically allocated portion of the pool
- TOTAL SPACE LEFT -- Total unused bytes in the dynamically allocated portion of the pool
- LARGEST BLOCK -- Size in bytes of the largest block of unused space in the dynamically allocated portion of the pool
- SMALLEST BLOCK -- Size in bytes of the smallest block of unused space in the dynamically allocated portion of the pool
- NUMBER OF BLOCKS LEQ 32 BYTES -- Blocks less than or equal to 32 bytes in size in the dynamically allocated portion of the pool

DISPLAY (Cont.)

The display provides two measurements for each space allocation:

- VALUE -- Amount of space at the end of the preceding interval
- AVG -- Average amount of space since the display began

The following example produces a NONPAGED POOL STATISTICS display with an interval of 8 seconds:

```
DISPLAY NAME? POOL

UPDATE INTERVAL IN SECONDS? 8
```

The system manager should set the update interval as short as possible without impeding readability to obtain many pictures of the pool at different times.

Figure 14-6 illustrates a NONPAGED POOL STATISTICS display. From this display, the system manager might, for example, learn that the dynamic portion of the pool has 12,048 bytes of unused space in 27 blocks, but that most of the space is in one contiguous 11,136-byte block.

NONPAGED POOL STATISTICS
14:28:26

	VALUE	AVG
I/O REQUEST PACKETS LEFT	64	64.00
NUMBER OF HOLES IN POOL	27	27.00
TOTAL SPACE LEFT	12048	12048.00
LARGEST BLOCK	11136	11136.00
SMALLEST BLOCK	16	16.00
NUMBER OF BLOCKS LEQ 32 BYTES	18	18.00

Figure 14-6 NONPAGED POOL STATISTICS Display

14.11 S2 AND S5 COMMANDS

S2 and S5 produce the NUMBER OF PROCESSES IN SCHEDULER STATES display. The S5 command produces the display as a bar graph; its use is restricted to VT55 terminals. The S2 and S5 commands take the forms, respectively:

```
S2

S5
```

DISPLAY (Cont.)

The display shows the number of processes in each of the 14 scheduler states:

- COLLIDED PAGE WAIT (CPG) -- Waiting for a faulted page in transition
- MUTEX AND MISC. RESOURCE WAIT (MWT) -- Waiting for the availability of a mutual exclusion (mutex) semaphore or a dynamic resource
- COMMON EVENT FLAG WAIT (CEF) -- Waiting for some combination of event flags to be set in a common event block
- PAGE FAULT WAIT (PFW) -- Waiting for a page to be read as a result of a page fault; resident processes
- LOCAL EVENT FLAG WAIT (LEF) -- Waiting for some combination of local event flags to be posted; resident processes
- LOCAL EVENT FLAG (OUT OF BALANCE SET) (LFO) -- Waiting for some combination of local event flags to be posted; outswapped processes
- HIBERNATE WAIT (HIB) -- Waiting for a hibernate request; resident processes
- HIBERNATE WAIT (OUT OF BALANCE SET) (HBO) -- Waiting for a hibernate request; outswapped processes
- SUSPENDED WAIT (SSP) -- Waiting for a suspend request; resident processes
- SUSPENDED WAIT (OUT OF BALANCE SET) (SPO) -- Wating for a suspend request; outswapped processes
- FREE PAGE WAIT (FPG) -- Waiting for a free page of memory
- COMPUTE (COM) -- Ready to use the processor; resident processes
- COMPUTE (OUT OF BALANCE SET) (CMO) -- Ready to use the processor; outswapped processes
- CURRENT PROCESS (CUR) -- Using the processor

The CUR process is the one currently executing. LFO processes normally belong to interactive users who are thinking, although they might be processes waiting for disk I/O on a crowded system. LEF processes are normally interactive users who are thinking or processes waiting on a disk I/O. A state of CMO for any process indicates a very crowded system and one on which performance (interactive response time and batch throughput, for example) is poor.

The following example produces a NUMBER OF PROCESSES IN SCHEDULER STATES display with an interval of 8 seconds:

```
DISPLAY NAME?  S2

UPDATE INTERVAL IN SECONDS?  8
```

DISPLAY (Cont.)

The system manager should set the update interval as short as possible without impeding readability to obtain many pictures of the processes at different times.

Figure 14-7 illustrates a NUMBER OF PROCESSES IN SCHEDULER STATES display in standard format. Figure 14-8 illustrates a NUMBER OF PROCESSES IN SCHEDULER STATES display as a bar graph (VT55 terminals only). From the standard display, the system manager might, for example, learn that 8 resident processes are ready to use the processor.

```
                    NUMBER OF PROCESSES IN SCHEDULER STATES
                                   14:29:40

                                        0         10        20        30        40
                                        + - - - - + - - - - + - - - - + - - - - +
                                        :         :         :         :         :
COLLIDED PAGE WAIT
MUTEX AND MISC. RESOURCE WAIT
COMMON EVENT FLAG WAIT
PAGE FAULT WAIT
LOCAL EVENT FLAG WAIT                    ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
LOCAL EVENT FLAG (OUT OF BALANCE SET)    ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
HIBERNATE WAIT                           ▒▒▒▒▒▒▒▒▒▒▒▒▒▒
HIBERNATE WAIT (OUT OF BALANCE SET)
SUSPENDED WAIT
SUSPENDED WAIT (OUT OF BALANCE SET)
FREE PAGE WAIT
COMPUTE                                  ▒▒▒▒▒▒▒▒
COMPUTE (OUT OF BALANCE SET)
CURRENT PROCESS                          ▒
                                        :         :         :         :         :
                                        :         :         :         :         :
                                        :         :         :         :         :
                                        + - - - - + - - - - + - - - - + - - - - +
```

Figure 14-7 NUMBER OF PROCESSES IN SCHEDULER STATES Display (S2 Command)

DISPLAY (Cont.)

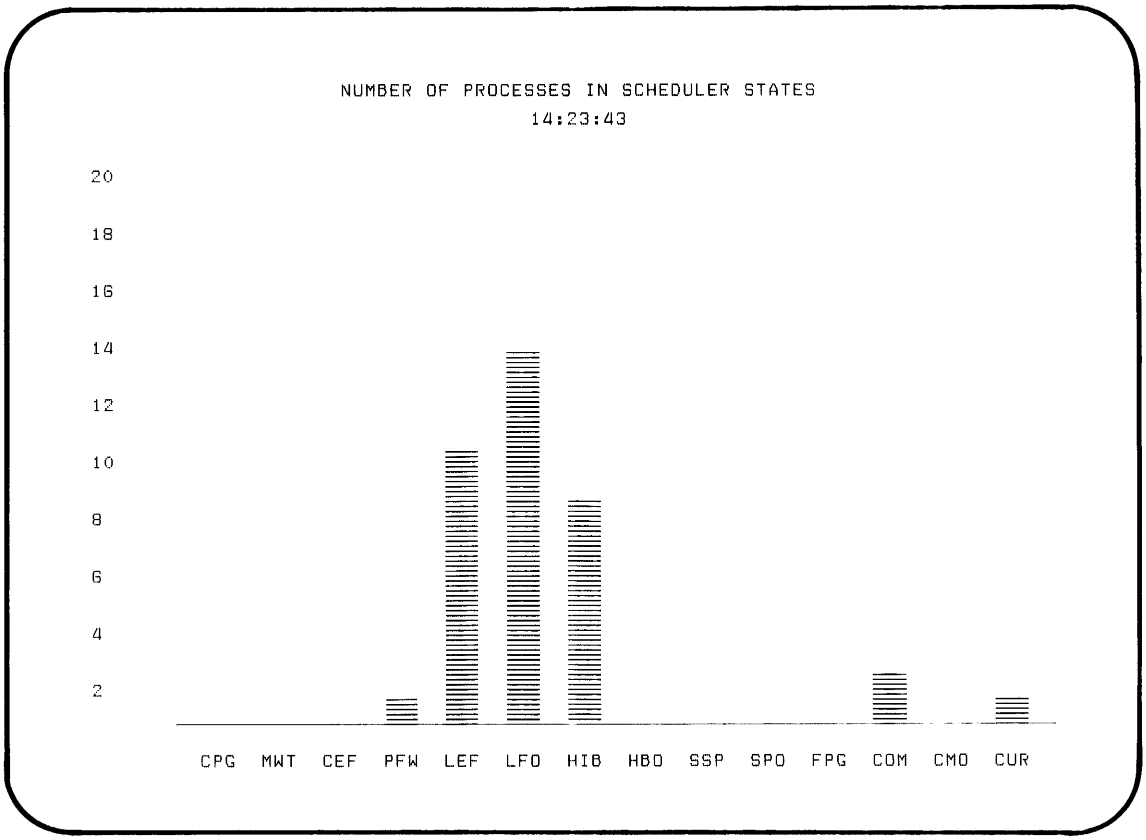

Figure 14-8 NUMBER OF PROCESSES IN SCHEDULER STATES Display (S5 Command)

14.12 TOPUSERS COMMAND

TOPUSERS produces the TOP CPU TIME USERS display. It takes the form:

TOPUSERS

The TOP CPU TIME USERS display measures the percentage of the total CPU time taken by each user during the preceding interval. Users are identifed by UIC and name. Processes not in memory during the interval do not participate in the display.

The following example produces a TOP CPU TIME USERS display with an interval of 8 seconds:

```
DISPLAY NAME? TOPUSERS

UPDATE INTERVAL IN SECONDS? 8
```

The system manager should set the update interval as short as possible without impeding readability to obtain many pictures of the processes at different times.

DISPLAY (Cont.)

Figure 14-9 illustrates a TOP CPU TIME USERS display. From this display, the system manager might, for example, learn that user QA1 used over 25 percent of the total CPU time during the preceding interval.

```
                              TOP CPU TIME USERS(%)
                                    16:15:41

                                     0         25        50        75       100
                                     + - - - - + - - - - + - - - - + - - - - +
                                     :         :         :         :         :
 [320,001]   QA1                      ▒▒▒▒▒▒▒▒▒▒▒
                                     :         :         :         :         :
 [320,001]   _JOB1255                 ▒▒▒▒▒▒▒▒▒
                                     :         :         :         :         :
 [110,010]   ROGER                    ▒▒▒▒▒▒
                                     :         :         :         :         :
 [001,002]   DBA3ACP                  ▒▒▒
                                     :         :         :         :         :
 [222,020]   _JOB1256                 ▒▒▒
                                     :         :         :         :         :
 [001,002]   DBA2ACP                  ▒
                                     :         :         :         :         :
 [007,022]   LINDAP                   ▒
                                     :         :         :         :         :
 [265,010]   DISPLAY
                                     :         :         :         :         :
                                     :         :         :         :         :
                                     + - - - - + - - - - + - - - - + - - - - +
```

Figure 14-9 TOP CPU TIME USERS Display

14.13 **USERS COMMAND**

USERS produces the VAX/VMS PROCESSES display. It takes the form:

    USERS

For each process on the system during the preceding interval, the VAX/VMS PROCESSES display provides the following information:

- PID -- Process identification as assigned by the system, in hexadecimal
- UIC -- User identification code, in octal
- STATE -- Process's scheduler state (some abbreviations are longer than those used in the NUMBER OF PROCESSES IN SCHEDULER STATES display; COMO instead of CMO and LEFO instead of LFO, for example)

DISPLAY (Cont.)

- PRI -- Current (as opposed to base) priority of the process
- NAME -- Process name
- SIZE -- Number of shareable pages and total number of pages in use by the process
- DIOCNT -- Cumulative direct I/O operations performed by the process since its start; not displayed if process is swapped out
- FAULTS -- Cumulative page faults since the process started; not displayed if process is swapped out
- CPU TIME -- Cumulative CPU time in the format hours:minutes:seconds used by the process since its start; not displayed if process is swapped out

The top corners of the display contain the number of processes on the system and the time in days, hours, and minutes since the system was last booted. Processes that are swapped out are so noted.

If more processes are on the system than can be displayed on the terminal screen, the system segments the list of processes and displays each segment at the user-specified interval.

The following example produces a VAX/VMS PROCESSES display with an interval of 10 seconds:

```
DISPLAY NAME?  USERS

UPDATE INTERVAL IN SECONDS?  10
```

The system manager should set an update interval as short as possible without impeding readability to obtain many pictures of the system at different times.

Figure 14-10 illustrates a VAX/VMS PROCESSES display. From this display, the system manager might, for example, learn that user DAVIDP, waiting on a local event flag, has been running for quite a while and doing a fair amount of disk I/O.

DISPLAY (Cont.)

```
     PROCESSES: 37              VAX/VMS PROCESSES              UPTIME:     300:48
                                    14:20:12

   PID        UIC       STATE  PRI    NAME        SIZE    DIOCNT  FAULTS    CPU TIME

001D002C  [101,010]  LEF      8  ANDREW      61/100    1624    3370  00:00:35.10
0017002D  [201,011]  LEFO     4  RMSTST      14/43          SWAPPED OUT
0009002E  [105,020]  COM      5  RICHARDP    29/150    4496    3569  00:02:39.29
0031002F  [011,010]  LEF      5  LINDAP      14/100    6378    4778  00:01:57.99
00130030  [251,020]  LEF      9  JAMESP      18/128   12126    8254  00:03:57.25
00060031  [010,040]  LEFO     6  _TTC6:      0/32           SWAPPED OUT
00170032  [007,007]  LEFO     4  HANKP       20/48          SWAPPED OUT
00010033  [001,002]  HIB      8  DBA2ACP     31/71    99114      47  00:23:04.62
00350034  [360,015]  LEFO     4  JONES       15/44          SWAPPED OUT
00220035  [221,020]  LEF      7  KATHYP      31/102    9044    9734  00:03:36.99
00010036  [007,377]  LEFO     4  SYSTEM      3/35           SWAPPED OUT
00150037  [241,010]  LEF      7  WILLIAM_B   38/170   20771    8904  00:08:15.74
00010038  [001,002]  LEF      9  DBA1ACP     31/71    44381      47  00:05:24.83
00020039  [001,002]  HIB      8  PRTSYMB2    8/31     20470      21  00:11:40.62
000B003A  [010,010]  LEF      7  DAVIDP      37/114  452875  212260  14:51:45.25
0001003B  [001,006]  HIB      7  ERRFMT      0/26       900      15  00:00:08.25
0001003C  [002,001]  LEFO     8  OPERATOR    0/30           SWAPPED OUT
```

Figure 14-10 VAX/VMS PROCESSES Display

14.14 ERROR MESSAGES

Table 14-2 lists the error messages issued by the DISPLAY utility (in alphabetical order).

Table 14-2
DISPLAY Error Messages

Message	Remedy
COMMAND NOT UNIQUE	Spell out the command name more.
INVALID DISPLAY NAME, TYPE "HELP" FOR HELP	Enter a valid command.
THIS DISPLAY FOR VT55's ONLY -- TERMINAL IS NOT A VT55	Informational message -- do not respond.

Where correction of an entered value is indicated, the entire command must be reentered.

INDEX

A

B

C

D

L

M

N

O

P

T

U

V

W

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- [ ] Assembly language programmer
- [ ] Higher-level language programmer
- [ ] Occasional programmer (experienced)
- [ ] User with little programming experience
- [ ] Student programmer
- [ ] Other (please specify)________________

Name________________ Date________________

Organization________________

Street________________

City________________ State________________ Zip Code________________
or
Country

Please cut along this line.

— — — Do Not Tear - Fold Here and Tape — — — — — — — — — — — — — — — — — — —

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

BSSG PUBLICATIONS TW/A14
DIGITAL EQUIPMENT CORPORATION
1925 ANDOVER STREET
TEWKSBURY, MASSACHUSETTS 01876

— — Do Not Tear - Fold Here — — — — — — — — — — — — — — — — — — —