

IAS Performance and Tuning Guide

Order Number: AA-H848C-TC

Operating System Version: IAS Version 3.4

May 1990

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright ©1990 by Digital Equipment Corporation

All Rights Reserved.
Printed in U.S.A.

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

The following are trademarks of Digital Equipment Corporation:

DDIF	IAS	VAX C
DEC	MASSBUS	VAXcluster
DEC/CMS	PDP	VAXstation
DEC/MMS	PDT	VMS
DECnet	RSTS	VR150/160
DECUS	RSX	VT
DECwindows	ULTRIX	
DECwrite	UNIBUS	
DIBOL	VAX	digital ™

This document was prepared using VAX DOCUMENT, Version 1.2

Contents

PREFACE	ix
---------	----

CHAPTER 1 PERFORMANCE AND TUNING CONSIDERATIONS 1-1

1.1	FACTORS AFFECTING PERFORMANCE	1-1
-----	-------------------------------	-----

1.2	SELECTING FEATURES AT SYSTEM GENERATION	1-2
-----	---	-----

1.3	EFFICIENT USE OF MEMORY	1-2
-----	-------------------------	-----

1.4	EFFECTIVE USE OF DISKS	1-2
-----	------------------------	-----

1.5	TAILORING THE FILE SYSTEM	1-2
-----	---------------------------	-----

1.6	TUNING THE IAS SCHEDULER	1-2
-----	--------------------------	-----

1.7	IMPROVING THE PERFORMANCE OF A TIMESHARING OR MULTIUSER SYSTEM	1-2
-----	---	-----

1.8	IMPROVING THE PERFORMANCE OF A REAL-TIME SYSTEM	1-3
-----	---	-----

1.9	MANUAL OVERVIEW	1-3
-----	-----------------	-----

CHAPTER 2 SELECTING FEATURES AT SYSTEM GENERATION 2-1

2.1	INTRODUCTION	2-1
-----	--------------	-----

2.2	MEMORY MANAGEMENT DIRECTIVES	2-1
-----	------------------------------	-----

2.3	TERMINAL HANDLER	2-1
-----	------------------	-----

Contents

2.4	MODIFYING SYSTEM TASKS	2-3
2.5	BATCH	2-4
2.6	SCOM	2-4
2.6.1	Size of the SCOM Node Pool	2-5
2.7	PARTITIONS	2-5
CHAPTER 3 EFFICIENT USE OF MEMORY		3-1
3.1	SYSTEM SIZE	3-1
3.1.1	Memory Size	3-1
3.1.2	Number of Terminals	3-1
3.2	INSTALLING TASKS	3-2
3.2.1	Loading Handlers	3-2
CHAPTER 4 EFFECTIVE USE OF DISKS		4-1
4.1	INTRODUCTION	4-1
4.2	IMPROVING IAS USE OF DISKS	4-1
4.2.1	Spooler Files	4-1
4.2.2	Work Files	4-3
4.3	IMPROVING YOUR INSTALLATION'S USE OF DISKS	4-3
4.3.1	Using Different Disks	4-3
4.3.2	Specifying the Position of Index Files on Disk	4-4
4.3.3	Placing Files on More than One Disk	4-4
4.3.4	Using Memory-Resident Overlays	4-4
4.3.5	Releasing Free Space	4-5
4.4	SWAP FILES	4-5
4.4.1	Using a Fixed-Head Disk For Swapping	4-6

4.4.2	Using a Second Disk For Swapping _____	4-6
4.4.3	Disk Controller and Units _____	4-6
4.4.4	Swap Ratio _____	4-7

4.5	TUNING FOR MSCP DISK AND T/MSCP TAPE CONFIGURATIONS	4-7
-----	---	-----

CHAPTER 5	TAILORING THE FILE SYSTEM	5-1
------------------	----------------------------------	------------

5.1	HOW THE FILE SYSTEM AFFECTS IAS PERFORMANCE	5-1
-----	---	-----

5.2	SELECTING THE APPROPRIATE ACP TASK	5-1
-----	------------------------------------	-----

5.3	USING MULTIPLE ACP TASKS	5-2
-----	--------------------------	-----

5.4	USING THE FILE SYSTEM DATA AREAS	5-4
-----	----------------------------------	-----

5.4.1	Extending the Size of FCPCOM _____	5-4
-------	------------------------------------	-----

5.4.2	Extending the Size of the Data Areas within the ACP Task _____	5-5
-------	--	-----

5.5	REDUCING FILE SYSTEM DISK ACCESSES	5-6
-----	------------------------------------	-----

CHAPTER 6	TUNING THE IAS SCHEDULER	6-1
------------------	---------------------------------	------------

6.1	INTRODUCTION	6-1
-----	--------------	-----

6.2	EXAMINING THE TYPE OF JOBS YOU RUN	6-1
-----	------------------------------------	-----

6.3	IAS SCHEDULING PARAMETERS	6-1
-----	---------------------------	-----

6.3.1	Quantum Parameters _____	6-2
-------	--------------------------	-----

6.3.2	Promotion Time _____	6-3
-------	----------------------	-----

6.4	BATCH SCHEDULING PARAMETERS	6-3
-----	-----------------------------	-----

6.4.1	Batch Quantum _____	6-3
-------	---------------------	-----

6.4.2	Time Between Batch Schedules _____	6-3
-------	------------------------------------	-----

Contents

6.5	DEFAULT SCHEDULING PARAMETERS	6-4
6.6	SYSTEMS WHERE BATCH IS RARELY USED	6-5
6.7	SYSTEMS WHERE BATCH IS FREQUENTLY USED	6-6
6.8	SYSTEMS WHERE MOST TASKS ARE COMPUTE-BOUND	6-6
6.9	SYSTEMS WHERE MOST TASKS ARE I/O-BOUND	6-6
6.10	REAL-TIME SYSTEMS	6-6
6.11	REAL-TIME TASKS IN A TIMESHARING OR MULTIUSER SYSTEM	6-7
CHAPTER 7 EXAMPLE OF TUNING AN IAS SYSTEM		7-1
7.1	INTRODUCTION	7-1
7.2	IMPROVING A SLOW RUNNING SYSTEM	7-1
7.3	CHANGES TO THE SCHEDULING PARAMETERS	7-2
APPENDIX A WORKBOOK		A-1
A.1	WORKBOOK	A-1
INDEX		

FIGURES

1-1	Summary Flowchart _____	1-4
1-2	Summary Flowchart (continued) _____	1-5
5-1	Using a Single F11ACP Task _____	5-2
5-2	Using Multiple F11ACP Tasks _____	5-3
5-3	Picking Nodes for Data Structures _____	5-5

TABLES

4-1	Disk Attributes _____	4-1
5-1	File System Disk Access Parameters _____	5-8
6-1	IAS Scheduling Parameter Default Values _____	6-4
7-1	Effect of Changing the IAS Scheduling Parameters _____	7-1

Preface

Purpose of the Manual

This manual gives advice and general guidelines for improving the overall performance of your IAS system. By tuning (that is, modifying) certain parts of the system, you can increase throughput and/or reduce terminal response time. Note that the manual does not describe how to measure performance.

The reader is assumed to be the system manager with a detailed knowledge of IAS.

Document Structure

Chapter 1 gives an overview of performance and tuning considerations.

Chapter 2 describes how to save memory by sensible selection of features at system generation.

Chapter 3 to Chapter 5 give advice on system housekeeping (use of disks, placement of swap files, and tailoring the file system).

Chapter 6 describes how to tune the IAS scheduler to reflect the type of jobs running at your installation.

Chapter 7 gives an example of tuning the IAS scheduling parameters on one particular IAS system.

Appendix A is a workbook, consisting of a list of system components and their defaults, and blank spaces for you to fill in when you change the values. The workbook is printed three times to enable you to keep records easily.

Associated Documents

The reader should be familiar with the IAS manual set. In particular, the following manuals are essential sources of information:

- *IAS System Management Guide*
- *IAS Installation and System Generation Guide*
- *IAS Executive Facilities Reference Manual*

Refer to the IAS Documentation Directory for a brief description of other related documents in the manual set.

Convention	Meaning
<code>RET</code>	A box symbol with a one- to six-character abbreviation indicates that you press the appropriate key on the terminal, for example, <code>RET</code> .
<code>CTRL/x</code>	The phrase CTRL/x indicates that you must press the key labeled CTRL while you simultaneously press another key, for example, CTRL/C, CTRL/Y, CTRL/O.
<code>PDS>SHOW TIME</code> <code>05-JUN-1990 11:55:22</code>	Command examples show all output lines or prompting characters that the system prints or displays in capital letters.

Preface

Convention	Meaning
PDS>TYPE MYFILE.DAT . . .	Vertical series of periods, or ellipsis, mean either that not all the data that the system would display in response to the particular command is shown or that not all the data a user would enter is shown.
file-spec,...	Horizontal ellipsis (...) indicates that additional parameters, values, or information can be entered.
[logical-name]	Square brackets indicate that the enclosed item is optional. (Square brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks ("). The term apostrophe (') is used to refer to a single quotation mark.

1

Performance and Tuning Considerations

1.1 Factors Affecting Performance

IAS performance varies from one installation to another, depending on many different factors, including but not limited to the following:

- Type of system (timesharing, multiuser, or real-time)
- Number of users
- Workload

However, there are a number of ways you can improve the performance (response time) of your system. You can also speed up task execution time and increase throughput (that is, increase the number of tasks that you complete in a given time). This manual points out the areas of the system that you can tune (that is, modify) to suit your installation's needs, and gives general guidelines on the types of changes you can make.

Before tuning the system, understand what response time and throughput rate you require. You might have to trade off an improved response time against a slower throughput rate, or vice versa. After initial installation of IAS as distributed, examine what sorts of jobs run on your system and what sort of performance is obtained. Study the system over a period of a few weeks to get a true picture. Compare the actual performance with the desired performance and, if there is a disparity, look at the areas of the system described in this manual and try tuning these to improve performance.

Sometimes you must decide whether improving performance is worth the corresponding trade-off. For example, although you can save memory space by removing certain features, this might not benefit your installation in the long run. Similarly, increasing throughput sometimes necessitates allocating a certain amount of memory. You must decide whether it is more economical for your system to have that extra amount of memory available or whether it is better to increase throughput by adjusting a different area.

You can control the performance of an IAS system in different ways, depending upon various factors:

- Number of optional features selected
- Memory usage
- Disk usage
- File system performance
- Number and type of tasks running in the system (I/O-bound or compute-bound)
- Batch processing
- Number of terminal users

The following sections give a brief introduction to these factors and point to chapters in the manual where each is described.

Performance and Tuning Considerations

1.2 Selecting Features at System Generation

At system generation, you decide which features to include in your system. Your decisions will affect performance, because including unnecessary or rarely-used features might increase scheduling overheads or might take up valuable memory that you could allocate for user task processing. Chapter 2 lists features that you can select, exclude, or modify to suit your requirements.

1.3 Efficient Use of Memory

Always pay attention to the use of memory in your system, because wasted space slows down response time by reducing the amount of memory available for processing user tasks. On the other hand, ensure that there are sufficient nodes in the node pool to cope with your system's needs. You can reduce the load on the system node pool by using the data areas in FCPCOM and the ACP tasks.

See Chapter 3 for advice on efficient use of memory, and see Chapter 5 for FCPCOM and ACP tasks.

1.4 Effective Use of Disks

Use of disks is an important performance consideration. Making effective use of disks increases throughput and improves response time. See Chapter 4 for detailed suggestions.

1.5 Tailoring the File System

Most installations make substantial use of the file system; and by modifying a few areas you can improve performance. Chapter 5 describes the necessary changes to suit different types of media and usage.

1.6 Tuning the IAS Scheduler

On a timesharing or multiuser system, the IAS scheduler parameter settings affect performance. Digital supplies default values, but you can change these to suit the needs of your installation. Chapter 6 describes the parameters and suggests changes for different types of system and workload. The example in Chapter 7 shows how tuning the IAS scheduler in one particular installation considerably improved performance.

1.7 Improving the Performance of a Timesharing or Multiuser System

The most important factors to consider when you want to improve the performance of timesharing or multiuser systems are as follows:

- Number of terminal users (see Section 3.1.2)
- Disk usage (see Chapter 4)
- Swapping (see Section 4.4)
- File system (see Chapter 5)
- Number and type of tasks (see Section 6.2, Section 6.8, and Section 6.9)

- IAS scheduler (see Chapter 6)

.8 Improving the Performance of a Real-time System

The most important factors to consider when you want to improve the performance of real-time systems are as follows:

- Disk access (see Chapter 4)
- Checkpointing (see Section 4.4)
- File system (see Chapter 5)
- Task priority (see Section 6.10)

.9 Manual Overview

Figures 1–1 and 1–2 provide a summary of the types of IAS systems, indicate where you can improve performance, and refer to the relevant chapter or section of the manual.

Performance and Tuning Considerations

Figure 1-1 Summary Flowchart

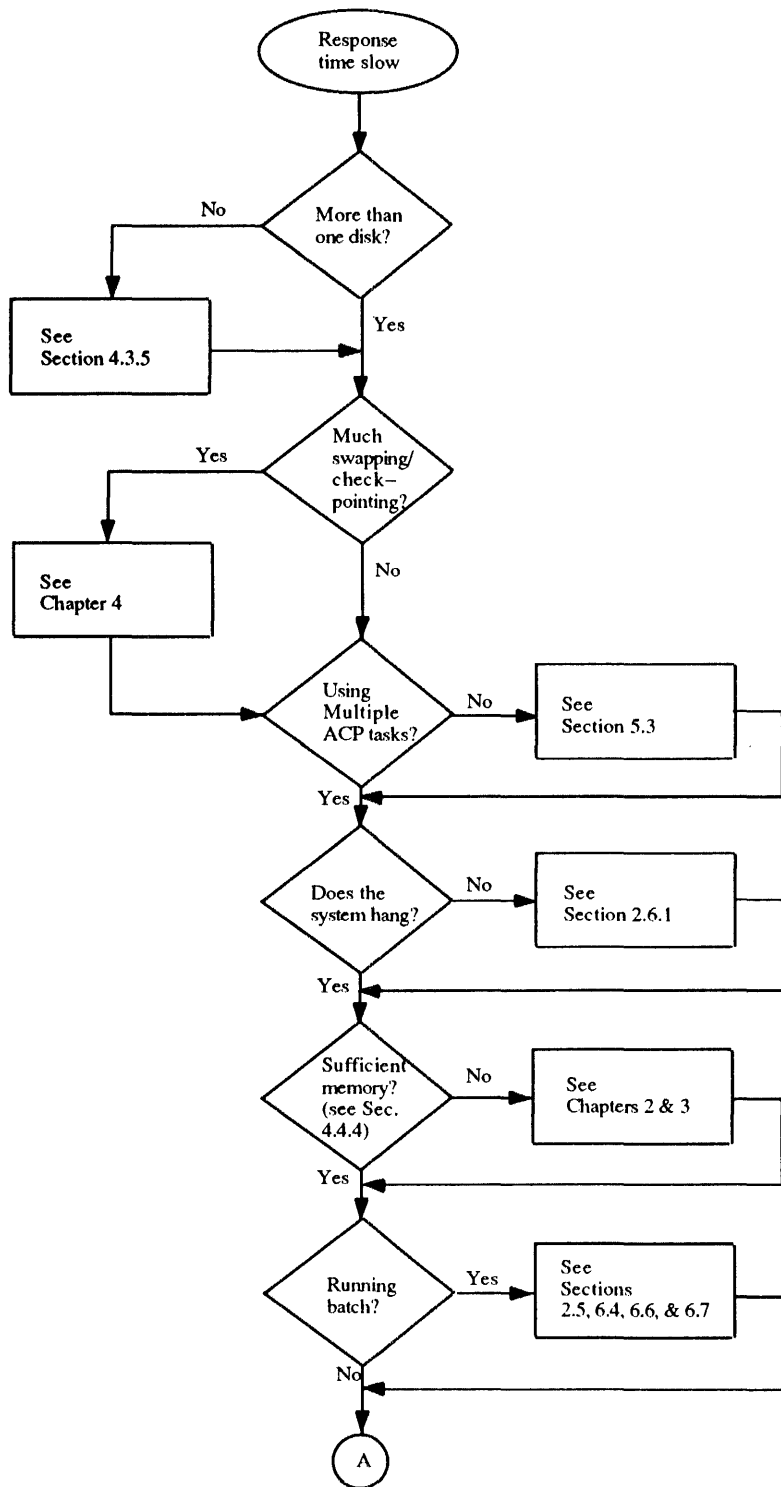
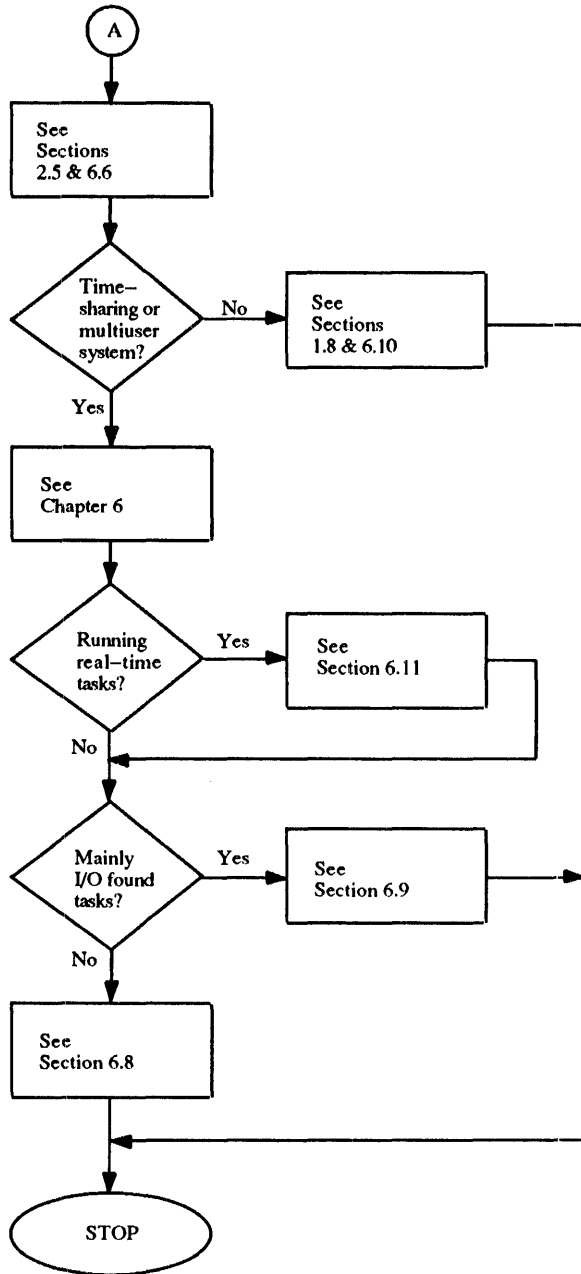


Figure 1-2 Summary Flowchart (continued)



2

Selecting Features at System Generation

2.1 Introduction

At system generation time, you decide on the type and basic characteristics of your system. Many options are available. Select only those that are needed at your installation. You specify most of the options in the question and answer session, which is described in the *IAS Installation and System Generation Guide*.

While pertinent selection of useful features can improve performance, inclusion of unnecessary or rarely-used features can waste memory and slow down the system. The rest of this chapter draws attention to those features (specified at system generation) that you can modify, include, or exclude, depending on the needs of your installation.

2.2 Memory Management Directives

Memory management directives enable a task to have segments totalling more than 32K words resident in memory rather than overlaid on disk. The *IAS System Directives Reference Manual* contains a full description of memory management directives. Memory management can increase a task's execution speed by reducing disk I/O requirements. However, you achieve this increase at the expense of increased memory usage by the task. The executive is also larger because memory management directives take up approximately 1 to 1.5K words of permanently resident code.

To include or omit the memory management directives, answer YES or NO to the following question in the question and answer session at system generation.

DO YOU REQUIRE MEMORY MANAGEMENT DIRECTIVES? [Y/N]:

Note: If your installation uses the FORTRAN-IV virtual array facility or RMS resident overlay libraries, you must include the memory management directives in your system.

2.3 Terminal Handler

IAS supports two versions of the terminal handler:

- 1 The single-terminal handler (TT01), that you specify only on single-user, real-time configurations where space is at a premium.
TT01 is not supported on multiuser or timesharing systems.
- 2 The multiple-terminal handler (TT), which supports multiple terminals on all types of interface, and has many additional features.

If you have more than one terminal, you must use the multiple-terminal handler (TT).

When you configure the terminal handler at system generation, omit the features that your installation does not need. This saves space. The file [311,114]PARAMS.MAC contains a list of terminal handler features that you can edit to suit your installation's needs. The file PARAMS.MAC consists of three parts:

- 1 Group 1 assembly parameters, which you must set up for each individual system.

Selecting Features at System Generation

- 2 Group 2 assembly parameters. Digital assigns values to these parameters in the distributed software, but you can change the values to suit your installation.
- 3 Group 3 assembly parameters that you must not change.

Features you could omit (depending on your installation) are listed below.

NOTE: The figures given for the amount of space saved by omitting each feature are in all cases approximate. The amount of space saved varies depending on which other features are present.

- 1 Bells and whistles (B\$\$AW). This parameter controls the inclusion of a number of useful but space-consuming features (for example, type-ahead, `Ctrl/R`, and scope tab rubout). Omit this parameter to save space. Specify:

- B\$\$AW=1 — to include bells and whistles.
- B\$\$AW=0 — to omit bells and whistles.

Omitting bells and whistles saves 1.2K words.

If you want to include or omit certain bells and whistles features, specify them with the individual parameters (for example, R\$\$BTB=1 to include scope tab rubout). This overrides the parameter setting in B\$\$AW.

- 2 Escape sequence support (E\$\$SEQ). If you specify this parameter, the terminal handler treats escape as the start of an escape sequence rather than as a read terminator (for those terminals that are set for escape sequence recognition). You can use escape sequences to extend the number of terminal control functions and special characters available without increasing the number of character codes.

Specify E\$\$SEQ

- =0 — No escape sequence support.
- =1 — Only DEC sequences supported. This takes up approximately 1K words.
- =2 — Only ANSI sequences supported. This takes up approximately 0.5K words.
- =3 — Both DEC and ANSI sequences supported. This takes up approximately 1K words.
- =4 — Both DEC and ANSI sequences supported but no translation. This takes up approximately 0.5K words.

- 3 Interface definitions (for example, DC11, DH11, DZ11, DHU11). Do not include interfaces that your installation does not use, because each unwanted interface specification takes up 10 (decimal) words. If you leave out unwanted interfaces, the relevant interface-dependent code is omitted from the terminal handler. This provides an additional space saving. Specify the following value to the appropriate interface parameter:

- n — to include support for n such interfaces, where n is a positive number.
- 0 — to omit support for the interface.

Ensure that the file [311,114]CONFIG.MAC reflects the changes you have made to PARAMS.MAC. CONFIG.MAC specifies each interface in detail and is described in the *IAS Installation and System Generation Guide*.

- 4 Dial-up support (D\$\$IAL). If you do not need dial-up or remote lines into the system, omit this feature to save space. Specify:

- D\$\$IAL=1 — to include dial-up support.

- D\$IAL=0 — to omit dial-up support.

Omitting support for dial-up saves 0.4K words.

- 5 Block mode support (B\$LCK). This feature enables a terminal to transmit data a block at a time. A typical application is a text editor, where you edit a page of data on the terminal, without involving the computer. When you have finished editing the page, you transmit it to the computer via an operator command. Specify:

- B\$LCK=1 — to include support for block mode terminals.
- B\$LCK=0 — to omit support for block mode terminals.

Omitting block mode support saves 0.3K words.

For further information on block mode support, see the *IAS Device Handlers Reference Manual*.

- 6 The following features take up less than 100 words apiece:

- T\$APE — remote paper tape
- B\$SP — backspace
- D\$HSF — silo fill
- H\$OLD — terminal hold
- L\$30S — LA30S support
- N\$L — newline terminals
- S\$FF — simulated formfeed
- V\$FIL — VT05 support

Each additional terminal requires approximately 32 words, and each additional interface requires approximately 10 words.

The assembly parameters and the PARAMS.MAC file are described in the *IAS Installation and System Generation Guide*. The terminal handlers are described in the *IAS Device Handlers Reference Manual*.

2.4 Modifying System Tasks

The **EXTEND TASK** (EXTK\$) directive enables a task to modify its size dynamically. Certain system tasks, such as TKB and MAC, use **EXTEND TASK** automatically. These tasks can be installed with a minimal in-core size since they extend themselves dynamically as the need arises.

The tasks CRF and LBR use disk work space if they run out of in-core work space. Installation of these tasks with a larger initial extension increases the task's in-core work space, thus increasing its speed. On systems with ample free memory, increase the extension size to 15000 words for CRF and LBR. You do this when installing the tasks during system generation Phase 2. The command might be, for example,

```
INS [11,1]CRF/INC=15000
```

You can also modify PIP to increase buffer size and speed for disk-to-disk copying. Edit the file [11,5] PIPBLD.CMD. Change 5100 to a larger value in the following line.

```
EXTSCT=$DYB1:5100
```

Selecting Features at System Generation

2.5 Batch

Specify the number of batch streams that your system needs during the question and answer session at system generation. Multiple batch streams enable you to process more than one batch job at the same time. IAS supports a maximum of eight batch streams. Support of each batch stream requires approximately 1K words of memory. If your system does not run much batch, specify only one batch stream. Even if your system runs a lot of batch, experience has shown that there is usually little to be gained by having more than three batch streams because the scheduling overheads involved slows down the system. You do not improve the throughput of the system by having more than three batch streams, because this number of non-interactive jobs is sufficient to keep the disk(s) and CPU busy. Too many batch jobs could increase swapping, and could therefore slow down the system.

2.6 SCOM

Specify the size of the system communication area (SCOM) at system generation by means of the SCOM directive. SCOM comprises the system subroutines, a communication area, the system tables and the node pool.

Determine the optimum SCOM size for your system by adding together the following elements:

- 1 The length in words of the system subroutines and the communication area:
subroutines + communication area = 1184 (decimal)
- 2 The number of tasks installed at any given time, multiplied by 16 words.
- 3 The number of system task directory (STD) entries specified in the SCOM directive (see the *IAS Installation and System Generation Guide*).
- 4 The number of DEV directives (see the *IAS Installation and System Generation Guide*) multiplied by 26 words.
- 5 The number of PAR directives (see Section 2.7) multiplied by 10 words.
- 6 The number of shareable global areas (SGAs) installed at any given time (taking account of task pure areas), multiplied by 24 words.
- 7 The number of swap files specified, multiplied by 16, plus the size of each swap file bitmap. You can calculate the size of each swap file bitmap by dividing the number of blocks in each swap file by 16, and then rounding up to the next whole word.
Use the MCR command SWA /LI or the SCI command SHOW SWAP_FILES to find the number of blocks in the bitmaps.
- 8 The number of scheduling levels (see Chapter 6, Section 6.3), multiplied by 16 words.
- 9 The number of interrupt vectors dealt with by each device handler, multiplied by 16 words.
- 10 The size of the variable-length node pool (see Section 2.1).

The above list details the elements that are present in a general-purpose system used mainly for timesharing or program development. Take into account other variable factors as well. For example, each active real-time task uses 24 words. A reasonable size for SCOM for a general-purpose system is $4K + 200n$ decimal words, where n is the number of terminals.

The maximum size for SCOM is 12K words. On systems where memory is not a constraint (particularly on PDP-11/70 installations), set the SCOM size to its maximum.

2.6.1 Size of the SCOM Node Pool

The node pool is part of SCOM, and each node contains a variable number of 8-word blocks. See the *IAS Executive Facilities Reference Manual* for a description of the use of nodes. Insufficient nodes can cause your system to hang (that is, you cannot obtain a response from the system on any terminal). Ensure that you specify enough space for the node pool.

You establish the size of the node pool when you specify the SCOM directive at system generation (see Section 2.6). The amount of space available for the node pool is as follows:

$$\text{Pool size} = \text{SCOM size} - (\text{Total of list items 1 to 9 in Section 2.6})$$

Specifying more space for SCOM results in a larger node pool. You might need to increase the size of the node pool if you have many real-time tasks in your system. Each active real-time task uses 24 words, and needs more if the task is sending and/or receiving messages.

Once you have specified the size of SCOM, you can dynamically monitor your node usage by means of the MCR command DEMO or by running the POOL node usage program. DEMO displays details of memory usage and task activity on a VDU screen. POOL prints the number of unused nodes in the pool (along with the largest contiguous amount of node space) every two seconds, on either a VDU screen or a hard-copy terminal. With DEMO and POOL you can determine whether the node pool contains any unused nodes. If there are many unused nodes, you can save space by reducing the size of your node pool. See the *IAS MCR User's Guide* for details about the DEMO command and the POOL node usage program.

You can also make use of nodes from the file system's private data areas to reduce the load on the SCOM node pool (see Chapter 5, Section 5.4).

2.7 Partitions

You specify the name, base address, size, and type of each partition in your system at system generation by way of the PAR directive (see the *IAS Installation and System Generation Guide*). To determine the optimum partition size, establish which tasks run in which partition. Determine the size of the relevant tasks, and specify the partition size accordingly. IAS supports three types of partitions:

- 1 User-controlled partitions
- 2 System-controlled partitions
- 3 Timesharing partitions

Only tasks executing in a timesharing partition run under IAS scheduler control, while real-time tasks can execute in any of the three types of partition. The *IAS Executive Facilities Reference Manual* contains a description of the three types of partition and their advantages.

On most timesharing and multiuser systems, timesharing partitions are preferable to system-controlled partitions because shuffling can take place to make free memory contiguous. However, system-controlled partitions are more suitable in the following circumstances:

- 1 When shuffling is undesirable, for example because a group of interacting executive privileged tasks are aware of each others' physical memory locations.
- 2 When the size of the node pool is a limiting factor. Each allocated segment of a timesharing partition takes one node from the pool to describe the segment. System-controlled partitions do not need space from the node pool. Also, active task list (ATL) nodes for tasks run under IAS scheduler control are 8 words (1 node) larger.

Selecting Features at System Generation

Real-time tasks that run on a timesharing or multiuser system can affect the response time for other tasks. However, this depends on the nature of the real-time tasks and the amount of real-time work done. If certain real-time tasks require a rapid response (for example, tasks controlling critical processes) use a separate partition for them (see Chapter 6, Section 6.11). However, for most purposes, a single general-purpose timesharing partition is sufficient.

When you specify partitions, remember that too many rarely-used partitions can result in memory-waste for long periods of time.

3

Efficient Use of Memory

3.1 System Size

Performance varies with your system's memory size, configuration, and workload. This chapter examines areas where more efficient use of memory improves the performance of an IAS system.

3.1.1 Memory Size

The response time of a small system (for example, a system with 96K words of memory) might be slow if the system has many users and a heavy workload. Such a system does not have much memory available for user tasks that have to wait for service, thus increasing response time. Adding more memory in such a case improves the performance, unless you also increase the workload and/or the number of users.

However, your installation (whatever its size) benefits from making the system use memory more efficiently.

NOTE: Remember that many suggestions in this manual for improving IAS performance cost memory. You must decide whether the gain in performance outweighs any possible loss due to increased memory overhead.

3.1.2 Number of Terminals

Adding terminals affects memory usage in two ways:

- 1 More terminals mean more users. The more users accessing the system, the greater the competition for the available user space. To overcome this, either add memory to your system or group the users so they do not all access the system at the same time of day. For example, one group could use the system during the morning, with another group using it in the afternoon. Thus, you can minimize peaks in system load and improve response time by reducing the competition for memory as well as for other system resources.
- 2 Extra terminals can decrease the character-buffering space available per terminal in the terminal handler's node pool. During assembly, the terminal handler allocates a number of nodes per terminal (shown in the value of the N\$\$ODS parameter in the file PARAMS.MAC) using an algorithm defined in PARAMS.MAC. See the *IAS Installation and System Generation Guide* for a description of the file PARAMS.MAC. The algorithm is based on general use, and in special circumstances (for example, if you use block mode terminals) you can override the value of the N\$\$ODS parameter and specify a larger value.

To determine whether you need to increase the N\$\$ODS value, monitor the terminal handler word NODCNT at run-time. You can use the OPE or EXAMINE commands for this purpose. If the value becomes too small, increase the value of N\$\$ODS. However, you cannot increase this value indefinitely, because the task segment containing the terminal handler node pool and tables has a maximum size of 4K words. To determine the maximum amount of space available, examine the APR2 segment in the terminal handler's memory allocation map. When you have used all the available space, no further increase is possible. Adding terminals in this

Efficient Use of Memory

situation results in less buffer space available per terminal, although you should not reach the space limit on a system with 32 or less terminals.

3.2 Installing Tasks

Installing a task causes the system to find and note the physical position of a task held on disk. This enables fast loading into memory. Installing a task causes a system task directory (STD) entry to be made that uses 16 SCOM words.

Do not install tasks that are rarely used or whose use can be predicted, because this wastes SCOM nodes.

3.2.1 Loading Handlers

You cannot use a device unless an appropriate device handler is resident in memory. Device handlers are tasks and you can install and load them into memory when required and unload them after use.

Some installations install and load all the handlers they require and leave them resident in memory. This wastes memory if rarely-used handlers are resident. You can save space by installing and loading rarely-used handlers only when they are required. The handler is loaded in any memory available within the assigned partition. Once loaded, a handler (with the exception of the Batch (BA), Message Output (MO), Null Device (NL), and Timesharing Control Primitives (PI) handlers) cannot be moved around in memory. This might cause the partition to become fragmented. However, if memory fragmentation presents a problem, you might prefer to accept the memory overhead and keep all handlers resident in memory. Also, if you cannot predict when a handler is to be used, or if you require an immediate response, you must install and load that handler permanently.

4 Effective Use of Disks

4.1 Introduction

You can improve IAS performance by making effective use of your disks. This is especially true in commercial and transaction processing applications, which perform a lot of I/O. This chapter describes how you can alter the way IAS uses disks, improve the way your own files are arranged on disk, and determine the best place for swap files.

Table 4–1 lists the disks IAS supports, together with their attributes. The access time and transfer rates of a disk are important performance considerations.

4.2 Improving IAS Use of Disks

By default, IAS places spooler files and work files on the system disk. This might not be the most effective use of available disks. You can improve performance by specifying different disks for these files, as described in Section 4.2.1 and Section 4.2.2.

4.2.1 Spooler Files

IAS places spooler files on the disk designated SP (spooled device). Usually, SP is assigned to the system disk (SY) by default at system generation. However, because the system disk (SY) is normally heavily used, placing the spooler files on SY might reduce performance, especially if users generate large amounts of spooled output. To ensure that there is sufficient free space for all the files that are being spooled at any given time, change the SP to a disk that is used less frequently by redirecting SP via the MCR command REDIRECT (see the *IAS MCR User's Guide*). Make sure that there are no files in the queue before redirecting SP. The current SP disk and the disk to which you are redirecting SP must not be mounted when you are redirecting files using this command.

Table 4–1 Disk Attributes

Disk	IAS Device Mnemonic	Capacity	Average Access Time	Transfer Rate	16Kw Transfer Time	Overlapped Seek
RA90	DUn	1.2 Gbytes	18.5 ms	2.8 Mbytes/second	-	-
RA82	DUn	622 Mbytes	32.3 ms	2.4 Mbytes/second	-	-
RA81	DUn	456 Mbytes	36.3 ms	2.2 Mbytes/second	-	-
RA70	DUn	280 Mbytes	27.0 ms	1.4 Mbytes/second	-	-
RA60	DUn	205 Mbytes	50 ms	1.98 Mbytes/second	-	-

Effective Use of Disks

Table 4-1 (Cont.) Disk Attributes

Disk	IAS Device Mnemonic	Capacity	Average Access Time	Transfer Rate	16Kw Transfer Time	Overlapped Seek
RD54	DUn	159 Mbytes	38.3 ms	625 Kbytes/second	-	-
RD53	DUn	71 Mbytes	38.3 ms	625 Kbytes/second	-	-
RD32	DUn	42 Mbytes	48.3 ms	625 Kbytes/second	-	-
RX33 floppy disk	DUn	1.2 Mbytes	92 ms	500 Kbytes/second	-	-
RX50 floppy disk	DUn	409 Kbytes per diskette	290 ms	250 Kbytes/second	-	-
RF11	DFn	256 Kwords	17 ms	16 micro- seconds/word	262 ms	No
RK05	DKn	1.2 Mwords	70 ms	11.1 micro- seconds/word	181 ms	Yes
RK06	DMn	14 Mbytes	38 ms	4.3 micro- seconds/word	70 ms	Yes
RK07	DMn	28 Mbytes	36.5 ms	4.3 micro- seconds/word	70 ms	Yes
RL01	DLn	5.2 Mbytes	67.5 ms	3.9 micro- seconds/word	63 ms	No
RL02	DLn	10.4 Mbytes	55 ms	3.9 micro- seconds/word	63 ms	No
RM02	DRn	67 Mbytes	42.5 ms	2.48 micro- seconds/word	40 ms	Yes
RM03	DRn	67 Mbytes	38.3 ms	1.65 micro- seconds/word	27 ms	Yes
RM05	DRn	256 Mbytes	38.3 ms	1.65 micro- seconds/word	27 ms	Yes
RP02	DPn	10.2 Mwords	29 ms	7.5 micro- seconds/word	122 ms	Yes
RP03	DPn	20.4 Mwords	29 ms	7.5 micro- seconds/word	122 ms	Yes
RP04/ RP05	DBn	88 Mbytes	36 ms	1.24 micro- seconds/byte	40 ms	Yes
RP06	DBn	176 Mbytes	36 ms	1.24 micro- seconds/byte	40 ms	Yes
RS03	DSn	256 Kwords	8.5 ms	4 micro- seconds/word	65 ms	No
RS04	DSn	512 Kwords	8.5 ms	4 micro- seconds/word	65 ms	No

Table 4–1 (Cont.) Disk Attributes

Disk	IAS Device Mnemonic	Capacity	Average Access Time	Transfer Rate	16Kw Transfer Time	Overlapped Seek
RX01 floppy disk	DSn	256 Kb	483 ms	18 micro-seconds/byte	589 ms	No
RX02 floppy disk	DYn	512 Kb	262 ms	21 micro-seconds/byte	688 ms	No

4.2.2 Work Files

WK is the pseudo-device for the disk where IAS places work files. Work files are temporary files to which certain system tasks (MAC, TKE, and CRF) require fast random access. Usually WK is assigned to the system disk (SY) by default at system generation. However, the system disk is normally heavily used and if delays in accessing work files occur, IAS performance suffers.

To overcome this, move the system work files onto another disk that is used less frequently. This improves the execution times of assemblies and task builds and generally makes more space available for work file processing.

To move the work files onto another disk, redirect WK by means of the MCR command REDIRECT (see the *IAS MCR User's Guide*).

Some user-supplied tasks that run under IAS use work files but do not specify the WK device. These work files are typically placed on the system disk. To speed up access, move these work files to a less frequently used removable disk, or a fixed-head disk.

If WK has been redirected to a disk other than the system disk, modify the task build command files for these tasks to assign the work file logical unit numbers (LUNs) to WK (the work file device).

4.3 Improving Your Installation's Use of Disks

To increase throughput and improve performance, examine the way your installation uses disks. Section 4.3.1 to Section 4.3.5 contain suggestions for improving your use of disks.

4.3.1 Using Different Disks

Using additional disks speeds up access times. If your installation has a fast fixed-head disk, its speed is an advantage (see Table 4–1). Move the files accessed by tasks that you run often, together with heavily-overlaid tasks, onto the fixed-head disk. These tasks involve frequent disk accesses and using a separate fixed-head disk improves their response and throughput.

If your installation does not have a fixed-head disk, the alternative is to move the files accessed by your most frequently-used tasks, together with heavily-overlaid tasks, onto an additional, removable disk. This also improves performance, because moving part of the load to a separate disk speeds up access (see also Section 4.3.3). If the disk drives support overlapped seeks, the system spends less time between data transfers, bringing an additional performance benefit.

Effective Use of Disks

If you are transferring large amounts of data, some removable disks are faster than the fixed-head disks because the transfer time (not the access time) is the most important factor. See the 16KW transfer time column in Table 4-1.

4.3.2 Specifying the Position of Index Files on Disk

The index file on Files-11 disks contains volume information and user file-header blocks so that the system can locate files quickly. See the *IAS I/O Operations Reference Manual* for more details.

To speed up access on a removable disk, put the index file near the most regularly-used files. When you initialize a disk, you specify the position of the index file by means of the MCR command INI or the PDS command INITIALIZE. The relevant qualifier is as follows:

MCR command INI	PDS command INITIALIZE
/INDX=Option	/INDEX:Option
where option is:	where option is:
BEG = beginning of volume	BEGINNING = beginning of volume
MID = middle of volume	MIDDLE = middle of volume
END = end of volume	END = end of volume
BLK:n - where n is the logical block number	n - where n is the logical block number
The default is MID	The default is MIDDLE

Note: When backing up and compressing disks with the DSC utility, you cannot specify the position of the index file. DSC places the index file at the beginning of the volume. See Section 4.3.5 for the advantages of DSC.

To determine the positions of your most regularly-used files, estimate the size of your files and in what order the system created them. If you are unable to do this, place the index file in the middle of the volume (the default).

4.3.3 Placing Files on More than One Disk

If your installation has more than one disk, balance your files so that the load is shared between the disks. This enables quicker access to those files. If a disk is over 75% full, response time suffers because the system takes longer to find and allocate disk space for write operations.

4.3.4 Using Memory-Resident Overlays

Overlaid user tasks can save time if you specify memory-resident overlays as the overlay structure.

Access to memory-resident overlays is faster than to disk-resident overlays because they avoid the need for copying an overlay segment from disk each time it is called. The executive loads memory-resident overlays from disk only the first time they are called; afterwards they reside permanently in memory.

However, memory-resident overlays take up more memory and could slow down the rest of the system if there is not much memory available. You need to decide whether your system can spare the extra memory, and whether the time saved is worth the extra memory involved.

NOTE: To specify memory-resident overlays for your tasks, you must include memory management directives at system generation.

If tasks specify disk-resident overlays, place related parts of the task in one overlay segment to save unnecessary disk accesses.

The *IAS Task Builder Reference Manual* describes both memory-resident and disk-resident overlays.

4.3.5 Releasing Free Space

Full disks slow down response time, as described in Section 4.3.3. You can release unused disk space by truncating files back to their logical end-of-file point with the PDS command TRUNCATE or the PIP truncate switch /TR. You can obtain more disk space by deleting unwanted versions of files using the PDS command DELETE/KEEP or the PIP purge switch /PU. You can specify an optional value *n* to be used in calculating the number of versions of the file to be retained. All existing versions with version numbers greater than the latest version minus *n* are retained. Note that if any versions between the latest and the latest minus *n* have already been deleted, fewer than *n* versions are retained. For both DELETE/KEEP and the PIP Purge switch, the default number of versions to be retained is 1.

You can also improve response time by invoking the disk save and compress (DSC) utility. DSC copies all the files on one disk to another disk for backup purposes. DSC copies only those blocks allocated to files. DSC writes data files consisting of blocks scattered randomly over the disk to contiguous areas on the new volume. This consolidates the free space into one contiguous area, and improves file access time. Using DSC also makes each file more contiguous, reducing the number of headers and retrieval pointers, and also reducing head movement. See the chapter on DSC in the *IAS Utilities Manual* for more information.

NOTE: DSC always places the index file at the beginning of the volume.

4.4 Swap Files

For a description of how IAS swaps tasks in and out of memory, see the *IAS Executive Facilities Reference Manual*. If you organize swapping efficiently, an improvement in performance results.

The attributes of the disk that you use for swapping (or checkpointing on real-time systems) affects performance. (See Table 4–1 for a list of supported disks and their attributes.) Disks with fast access time and fast transfer rates are more suitable for swapping. Section 4.4.1 and Section 4.4.2 contain suggestions for the best placement of swap files.

Determine the amount of swap space your system needs before you decide which disk to use for swapping. Swap space consists of swap files on one or more disks. IAS does not run a timesharing task unless enough swap space exists to record that task. Ensure that you specify sufficient swap space for all tasks that you want to run concurrently. Inadequate swap space limits the number of timesharing tasks that execute in the system at any one time.

IAS never splits a task between two swap files. If a task does not fit into one swap file, the system attempts to place it in the next swap file. If the task is scheduler-controlled and the system cannot place it in a swap file, that task is not run. If the task is not scheduler-controlled (that is, it is a real-time task with a priority of more than 100), and does not fit in a swap file, the system does not checkpoint the task. In this case, the system tries repeatedly to place the task in a swap file.

Effective Use of Disks

IAS also uses swap space for checkpointing real-time tasks, so make allowance for this when specifying the swap space size. Alternatively, you can dedicate a swap file to checkpointing real-time tasks only, by specifying the /RT switch to the MCR command SWA, or the SCI command CREATE/SWAPFILE REALTIME. See the *IAS MCR User's Guide* for details.

You specify the amount of swap space and the swapping disk by means of the MCR command SWA in the file [11,17]SYSBLD.COMD during system generation. Alternatively, you can specify the MCR command SWA or the SCI command CREATE/SWAPFILE during system start-up, or during system operation. See the *IAS MCR User's Guide* and the *IAS System Management Guide* for details of these commands.

4.4.1 Using a Fixed-Head Disk For Swapping

Specifying swap space on a fixed-head disk speeds up the time IAS takes to swap a task to disk. This is because fixed-head disks have better access times than removable disks (see Table 4-1). Fixed-head disks are thus suitable for very fast swapping, unless you are swapping large files. However when IAS swaps large files, the faster transfer rate of removable disks such as the RM02/3 and the RP04/5/6 overcomes their slower access times, and gives a faster transfer time (see the 16K word transfer time column in Table 4-1).

Fixed-head disks are more expensive than removable disks, and also have less capacity per surface. Space might be a problem if you want to use a fixed-head disk for all your swap files. One solution is to put the primary swap file on a fixed-head disk, and the remaining swap files on another, removable disk. IAS swaps onto the fixed-head disk first and uses the removable disk only when necessary.

If you specify a fixed-head disk for swapping, you can dedicate that disk to swap file use. Specify the /DV switch to the MCR command SWA or the /DEDICATED_VOLUME qualifier to the SCI command CREATE/SWAPFILE.

On the other hand, to take advantage of the fast access time of a fixed-head disk you can use some of the disk for swapping and the rest of the disk for work files and files accessed by commonly-used tasks (see Section 4.2). In this case, do not specify the /DV or DEDICATED_VOLUME switches.

4.4.2 Using a Second Disk For Swapping

If your installation does not have a fixed-head disk, put your swap files onto a second removable disk. A second removable disk is cheaper than a fixed-head disk and is almost as fast. Certain removable disks are faster than fixed-head disks if you are transferring large swap files (see Table 4-1).

4.4.3 Disk Controller and Units

Swapping performance is not only dependent on the hardware specifications of the swapping disks (as shown in Table 4-1), but also on the use of disks within the system. The important factors are whether you dedicate a controller and/or one device unit to a disk that is used only for swapping.

If a disk that is used for swapping has neither a dedicated controller nor dedicated units, the swapping performance might be poor.

If you dedicate a unit to a disk used only for swapping, the performance improves. If you dedicate both a controller and a unit to a disk used only for swapping, you obtain the best performance. No delays occur because no other activity is taking place. However, you might be unable to dedicate a controller to a disk used only for swapping if swapping does not take place very often. Obtaining an extra disk is a better alternative in this situation.

NOTE: IAS disk handlers do not support multiple controllers. However, you can modify a disk handler to operate as a multi-user handler. The handler is written for just one controller, and runs a separate copy of the task for each controller to be serviced. The handler does not have to cope with several simultaneous transfers, and each controller can operate at full speed.

See Section 1.7 of the *IAS Guide To Writing A Device Handler Task* for more information.

4.4.4 Swap Ratio

Calculate the swap ratio for a timesharing or multiuser system as follows:

$$\text{Swap ratio} = \frac{\text{Total size of timesharing programs running in your system}}{\text{Free memory available in the timesharing partition}}$$

To determine the total size of timesharing programs running in the system, take samples of system use over a period of time to find out the size of programs running. Use the PDS command SHOW TASKS/TIMESHARING to obtain this information.

Compare your swap ratio to those given below. The figures give a guide to the maximum acceptable swap ratio.

I/O-bound Tasks	CPU-bound Tasks
1.5	3 to 4

If your swap ratio is much higher, your system spends a great deal of time swapping tasks in and out of memory, thus slowing down the system. To improve your swap ratio, consider the following steps:

- 1 Add more memory to your system.
- 2 Limit the number of users who access the system. For example, use the PDS command SUBMIT/AFTER:time to regulate when batch jobs can be run.
- 3 Stop some tasks running.

4.5 Tuning for MSCP disk and T/MSCP tape configurations

Systems configured with 2 or more T/MSCP controllers should take advantage of the MSCP port server UQ\$SSP ([11,1]UQSSP.TSK) and the multiuser versions of the T/MSCP handlers DU.... ([11,1]DUMU.TSK for disk) and MU.... ([11,1]MUMU.TSK for tape). Using these tasks together significantly reduce the impact on system resources such as memory and UMR usage.

Effective Use of Disks

UQ\$SSP creates a region in memory that contains the communications areas for all T/MSCP controllers on the system. Special options at task build time can be used to determine the size of the communications regions so that only the minimum memory required is reserved for the number of controllers in a given configuration. Other options available can be specified to adjust the configuration of each communications area, adjusting the number of command and response buffer allocated, to obtain maximum I/O efficiency with minimum memory impact.

Memory usage is further reduced by putting all T/MSCP initialization code in one task. This eliminates the multiple copies of initialization code that would be necessary for each copy of the single user versions of the T/MSCP handlers.

For UNIBUS PDP-11s, a further reduction in UMR requirements is achieved since the all the communications areas (one for each controller) in the communications region can be mapped by 1 UMR. See the *IAS Version 3.4 Release Notes* for further information on T/MSCP configurations.

Install the following tasks for multiple T/MSCP controller configurations:

- [11,1]DUMU.TSK - MSCP handler for RAxx,RDxx,RC25,Rx50,RX33 disks
- [11,1]MUMU.TSK - T/MSCP handler for TK50 and TU81 tape drives
- [11,1]JUQSSP.TSK - T/MSCP Port Server (Controller initialization)

Install one of the following tasks for single T/MSCP controller configurations:

- [11,1]DU.TSK - MSCP handler (for single MSCP controller)
- [11,1]MU.TSK - T/MSCP handler (for single T/MSCP controller)

Install the following task whenever an MSCP handler is installed:

- [11,1]HIBBR.TSK - MSCP Bad Block Replacement task (HI\$BBR)

Install the following task whenever an MSCP or T/MSCP handler is installed:

- [11,1]ERRDSA.TSK - T/MSCP Error log server (ER\$LOG)

5 Tailoring the File System

5.1 How the File System Affects IAS Performance

The file system is often one of the most heavily used parts of a computer system. On a timesharing system especially, users continually access files via the file system. On a system with many users, it is important that you tailor the file system to the installation's needs. Otherwise the file system is likely to restrict system throughput.

You can tailor the file system in four ways:

- 1 Select the appropriate ACP task for your system (see Section 5.2).
- 2 Use multiple ACP tasks (see Section 5.3).
- 3 Increase the size of the file system's private data areas to reduce the load on the system node pool (see Section 5.4).
- 4 Modify the disk access parameters at volume initialization or mount time, to reduce disk accesses (see Section 5.5).

These methods are described in the rest of the chapter.

5.2 Selecting the Appropriate ACP Task

The IAS file system operates through privileged tasks called Ancillary Control Processors (ACPs). These tasks use the file structure information on the disk to convert file-oriented requests from tasks into the necessary disk accesses.

IAS provides three versions of the disk ACP task (F11ACP):

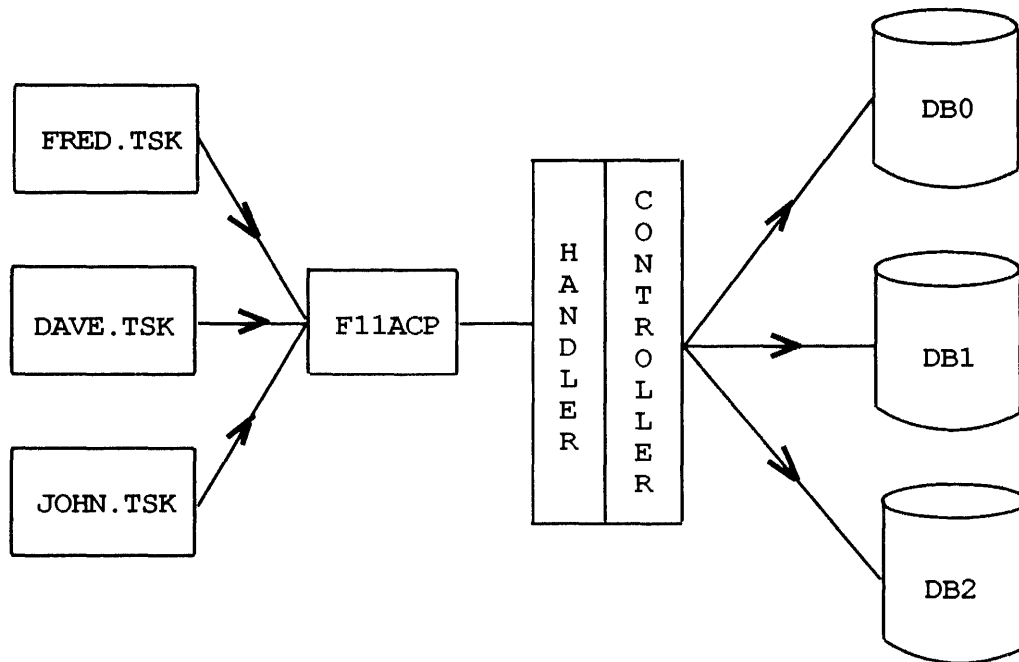
- 1 **FCP.TSK** — a very heavily overlaid, small version.
Use this version if you have a small system where memory is at a premium. If less than 96K words of memory are available for user task processing, use FCP.TSK until you have enough experience of the system to make a more informed decision.
- 2 **BIGFCP.TSK** — a lightly overlaid, larger version.
Use this version if your system has several simultaneous users. You are recommended to use this version on any timesharing or multiuser system that has at least 48K of memory available for user task processing.
- 3 **RESFCP.TSK** — a memory-resident overlaid version.
Use this version to improve the performance of the file system. Access to memory-resident overlays is faster than to disk-resident overlays. However, memory-resident overlays take up more memory. You need to decide whether your system can spare the extra memory, and if the time saved is worth the extra memory involved.

Select the appropriate ACP task at system generation time. See the *IAS Installation and System Generation Guide* for details of the required commands.

5.3 Using Multiple ACP Tasks

Normally Files-11 disks and DECTapes use only one ACP task (F11ACP). This ACP task processes all requests for all such volumes sequentially, one at a time (see Figure 5-1).

Figure 5-1 Using a Single F11ACP Task



Sequential processing can restrict system throughput. In Figure 5-1, task FRED has to finish accessing one of the disks before task DAVE can gain access. Similarly, task JOHN must wait until task DAVE has finished accessing one of the disks. When using a single F11ACP task, you cannot take advantage of overlapped seeks for multiple file accesses. Swapping, overlaying and task loading can overlap a file access.

You can overcome this problem by allocating a separate ACP task to the different Files-11 devices, as follows:

- 1 Specify a separate ACP for DECTape processing if DECTape is regularly used at your installation. If you allocate F11ACP for DECTape as well as for disks, all file processing on the disks stops whenever a DECTape file operation is in progress.

The DECTape ACP task is called DTAACP. You install DTAACP at system generation time, as described in the *IAS Installation and System Generation Guide*.

- 2 Specify a separate copy of F11ACP for each heavily-used disk. In this way, file operations for different disks can occur concurrently, as shown in Figure 5-2.

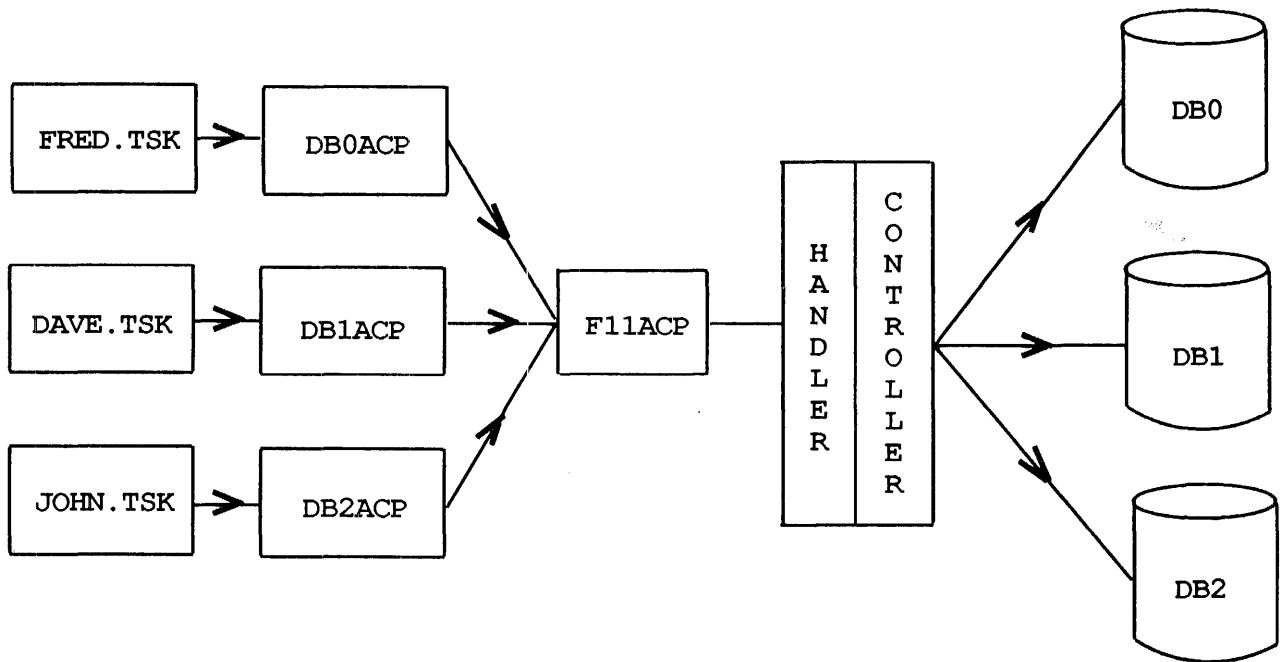
For each disk that is to have its own ACP task, install the ACP task with a different name, of the form:

nnnACP

where nnn is the device name, for example DB1ACP.

Specify the disk that is to use the ACP task either at system generation or when you mount the disk. See the *IAS Installation and System Generation Guide* for details of the commands. If your installation uses more than one type of disk, use a separate ACP for each type to increase throughput.

Figure 5-2 Using Multiple F11ACP Tasks



Even in systems with only one type of disk but more than one drive, dedicating a separate ACP for each drive improves performance. If the handler supports overlapped seek, a separate ACP for each drive enables a drive to seek in preparation for the actual transfer while another drive is transferring data. This is useful for multiple file accesses, swapping, task loading and overlaying.

A separate ACP task exists for magnetic tape handling, called MTAACP. You must select MTAACP if you intend to use ANSI-structured magnetic tape at your installation.

5.4 Using the File System Data Areas

The file system has to maintain a number of data structures to keep track of open files and mounted volumes. Some of these data structures take nodes from the SCOM node pool (see Section 2.6.1). However, to reduce the load on the SCOM node pool you can configure the file system at system generation time so that nodes are taken from private data areas instead. This is particularly valuable for a system with 24 or more users.

The file system data structures are as follows:

- 1 A Volume Control Block (VCB) of 23 words for each mounted volume.
- 2 A File Control Block (FCB) of 22 words for each open file.
- 3 A File Window Block of 16 words (but see Section 5.5) for each file that each task opens.

Note the difference between an FCB and a window block. If more than one task opens the same file simultaneously, only one FCB contains information about the file. However, one window block exists for each task that contains information about the particular access to the file.

- 4 A Locked Block List for each file opened with shared access, using RMS-11. This contains a varying number of entries of eight words each.

The following data areas exist to contain these structures:

- 1 The SCOM node pool.
- 2 The F11ACP common area, FCPCOM. This is a pool of up to 4K words that all Files-11 ACP tasks share.
- 3 A data area private to each ACP task.

A F11ACP task can only pick nodes for some data structures from certain data areas, as described in Figure 5-3.

You can extend the size of FCPCOM and the data areas within an ACP task to accommodate the data structures at system generation time (see Section 5.4.1 and Section 5.4.2).

5.4.1 Extending the Size of FCPCOM

Extend the size of FCPCOM as follows:

- 1 Determine the required size. Because FCPCOM is used mainly for window blocks, its size is $32n$ bytes, where n is the typical number of open files. As a guide, on a timesharing system calculate n as twice the number of terminals and batch streams.

The maximum size is 4K words.

- 2 Edit the command file [11,16]FCPCOMBLD.COMD. Change the line:

```
EXTSCT=FCPCOM:nnnnn
```

so that $nnnnn + 2010$ is the required size (in octal bytes).

Figure 5-3 Picking Nodes for Data Structures

Data Structure Area	Data Structure			
	VCB	FCB	Window Block	Locked Block List
SCOM	1	3	2	1
FCPCOM	-	2	1	-
ACP Task	-	1	-	-

Key	
-	Indicates that a F11ACP task cannot pick nodes for the data structure from that area.
1,2,3,	Indicates that a F11ACP task will pick nodes for the data structure from the area marked 1 if possible, then the area marked 2, and so on.

3 Rebuild FCPCOM with the command:

```
TKB @[11,16]FCPCOMBLD
```

After rebuilding FCPCOM, rebuild all the Files-11 ACP tasks so that they refer to the new version. Perform a system generation to install the new FCPCOM and the new ACP tasks.

5.4.2 Extending the Size of the Data Areas within the ACP Task

Extend the size of the ACP task data areas as follows:

- 1 Determine the required size of the data area. Because this area is used for FCB, its size is $44n$ bytes, where n is the typical number of open files.

The maximum size of the data area is as follows:

Tailoring the File System

F11ACP (using FCP.TSK)	20000 octal bytes
F11ACP (using BIGFCP.TSK)	10000 octal bytes
F11ACP (using RESFCP.TSK)	10000 octal bytes
DTAACP	20000 octal bytes

- 2 Edit the build file for the appropriate FCP task file, either [11,16]FCP.CMD, [11,16]BIGFCP.CMD, [11,16]RESFCP.CMD, or [11,16]DTAACP.CMD. Change the line:

```
EXTSCT=$$AFR1:nnnnn
```

so that nnnnn + 2 is the required size (in octal bytes), as determined above.

- 3 Rebuild the task using the command:

```
TKB @buildfile
```

where *buildfile* is the file selected in Step 2 above. You need to perform a system generation if the ACP task is for the booted disk.

5.5 Reducing File System Disk Accesses

To reduce file accesses to the system disk and thus increase throughput, specify three parameters when you initialize or mount a volume. The appropriate commands are INITIALIZE or MOUNT (PDS), and INJ or MOU (MCR). The three parameters are described below:

- 1 The directory LRU cache. This feature allows directories to be pre-accessed as follows. The first time you perform a directory operation on the MFD or a UFD, you access that directory. A 22-word File Control Block (FCB) is stored in the directory Least Recently Used list (LRU). The LRU list is a thread through the FCB list where the volume control block points.

Pre-accessing directories in this way normally saves three disk accesses for every directory operation. The search of the MFD to identify the UFD is completely by-passed; this eliminates reading the MFD file header and data block. Pre-accessing the UFD also eliminates reading the UFD's file header.

However, including this time-saving feature is at the expense of nodes in FCP's data area because each pre-accessed directory that you specify (for each mounted volume) requires an 22-word LRU entry. For example, if you specify /LRU=3, you require three 22-word LRU entries. Users with limited dynamic memory available might find they cannot include this feature.

NOTE: See Table 5-1 for details of the LRU parameter.

- 2 The default file extension. When you create a sequential file, the file system cannot know in advance how much space the file needs. The file system initially allocates the file a certain amount of space, and subsequently extends the file by the amount necessary. Each such extension requires up to three or more disk accesses. The default value for the file extension size is five blocks. Extending the file several times can increase the number of disk accesses by as much as 50%.

If you have large disks, you can reduce the number of disk accesses by specifying a larger file extension of, for example, 20 blocks. The disadvantage of a large file extension is that a certain amount of space is wasted at the end of each file. On average, this is half the default extension size per file. Therefore, if your installation has many small files rather than a few larger files, you might prefer to accept the overhead rather than wasting disk space.

Alternatively, regular use of the DSC utility (see Section 4.3.5) makes files contiguous and reduce the number of file extensions. You can also use the PIP utility with the /TR switch to truncate files back to their logical end-of-file point.

NOTE: See Table 5-1 for details of the file extension parameter.

- 3 The window block size. The file window block contains retrieval pointers that indicate the location of the blocks of the files on the disk. By default, seven such retrieval pointers are held for each file. Access to a part of the file to which the current set of retrieval pointers do not refer necessitates a single disk access. Particularly in commercial applications that use large files, it can be an advantage to extend the number of retrieval pointers. However, this feature uses extra memory, so you need to consider whether the advantages gained outweigh the loss of memory. Alternatively, use the DSC utility (see Chapter 4, Section 4.3.5) to make your files contiguous, because contiguous files need fewer retrieval pointers.

Ideally, the window block for a file contains sufficient retrieval pointers to cover the whole file. Calculate the number of pointers as follows:

$$\frac{\textit{file size}}{\textit{default extension}}$$

where default extension is as described in list item 2.

NOTE: See Table 5-1 for details of the window block size parameter.

Tailoring the File System

Table 5-1 File System Disk Access Parameters

MCR Command	Parameter	Default	PDS Command	Parameter	Default
INI	/LRU=n To set the directory LRU cache size.	/LRU=3	INITIALIZE	/ACCESSED:n To set the directory LRU cache size.	/ACCESSED:3
	/EXT=n To set the default file extension	/EXT=5		/EXTENSION:n To set the default file extension.	/EXTENSION:5
	/WIN=n To set the default window block size.	/WIN=7		/WINDOW:n To set the default window block size.	/WINDOW:7
MOU	/LRU=n To set the directory LRU cache size.	Set at volume initialization.	MOUNT	/ACCESSED:n To set the directory LRU cache size.	Set at volume initialization.
	/EXT=n To set the default file extension.	Set at volume initialization.		/EXTENSION:n To set the default file extension.	Set at volume initialization.
	/WIN=n To set the default window block size.	Set at volume initialization.		/WINDOW:n To set the default window block size.	

6 Tuning the IAS Scheduler

6.1 Introduction

The IAS scheduler controls the scheduling of tasks in a multiuser or timesharing system. The objective of the scheduler is to find the best compromise between response time and throughput. The IAS scheduler, as distributed, contains default settings for parameters that affect the performance of your system. When you have assessed the requirements of your installation, you can alter these settings to improve performance.

In a real-time system, scheduling is based solely on task priorities (see Section 6.11). The only way you can directly influence the scheduling is by selecting the desired task priority.

Before tuning the scheduler, find out how and in what areas you want to improve performance. After initial installation of a distributed IAS system, study what sort of jobs the installation runs and monitor the system's performance over a period of time. You are probably not in a position to tune the scheduling parameters at initial system generation, until you have had a chance to study how the system performs in your circumstances.

6.2 Examining the Type of Jobs You Run

Examine the type of jobs that users of your system typically run. Are they mostly real-time, interactive or batch tasks, or a mixture of more than one type? At certain times, only one type of task might be active (for example, you might run batch jobs exclusively at night).

Look more closely at the interactive tasks to see what percentage are compute (CPU)-bound, and what percentage are I/O-bound. Compute-bound tasks spend most of their time using the processor, for example, a FORTRAN program performing calculations. I/O-bound tasks spend little time using the processor and most of their time accessing files.

The conclusions you draw after examining the different types of jobs that your installation runs determines how you tune the scheduler to suit your needs.

6.3 IAS Scheduling Parameters

The IAS scheduler places the tasks to be run under its control in a number of different levels. The scheduler places I/O-bound tasks that need short, frequent amounts of time in the higher levels, while compute-bound tasks that need large amounts of time go in the lower levels. The scheduler services tasks in each level in round-robin fashion in priority order of levels, level one being the highest. See the *IAS System Management Guide* for a detailed description of scheduling.

IAS as distributed has three scheduling levels, to contain the following types of tasks:

Interactive Levels: $\left\{ \begin{array}{l} \text{Level 1 - Terminal interactive tasks} \\ \text{Level 2 - I/O-bound tasks} \\ \text{Level 3 - Compute-bound tasks} \end{array} \right\}$

The IAS scheduler automatically promotes or demotes interactive tasks between the first three levels according to their recent execution characteristics. Automatic promotion of tasks to level 1 whenever users complete terminal I/O ensures that terminal interactive tasks obtain the best

Tuning the IAS Scheduler

possible response from the system. Whenever a user types `Ctrl/C` on a terminal (timesharing systems only), the scheduler promotes that user's CLI to level 1.

For timesharing systems only, you can specify batch as the lowest level, Level 4, provided you specified at least one batch stream at system generation. To specify the lowest level as a batch level answer YES to the question:

```
DO YOU REQUIRE AN EXCLUSIVE BATCH LEVEL? [Y/N]
```

in the question and answer session at system generation. Level 4, the lowest level, is for batch tasks only and no promotion from or demotion to that level takes place.

The time the scheduler waits (if there is no task to run) before looking at the next level is specified via the idle time parameter.

You do not improve performance by specifying more or less than three interactive levels. If you run batch tasks, you can specify the batch level, making four scheduling levels altogether.

Promotion or demotion between the levels is controlled by the quantum and promotion time system parameters, described in Section 6.3.1 and Section 6.3.2 below.

6.3.1 Quantum Parameters

A task's quantum determines the amount of CPU time a task can have before it becomes eligible to be swapped out of memory. The scheduler allocates each task its quantum in a series of time slices. The time slice is the maximum continuous CPU time for which a task can execute before the scheduler services another task.

Each scheduling level also has an associated time factor. This value increases with the level number so that tasks at the higher priority levels (levels 1 and 2) that are scheduled more often, receive short amounts of CPU time, while tasks at lower levels receive longer amounts of CPU time when scheduled.

The following formula determines a task's quantum (Q):

$$Q = Ast + C$$

where:

- Q = task's quantum.
- A = allocation factor. This is the number of ticks per memory size. The allocation factor is used to determine the amount of CPU time allocated to a task. The value is given in ticks per memory size, so that the amount of CPU time allocated increases with task size.
- s = task size in 32-word blocks.
- t = time factor associated with each scheduling level. t increases with level number, that is, the lower the level, the higher the value of t.
- C = quantum constant (that is, the minimum guaranteed quantum for the system).

The quantum for a task in a high priority (low-numbered) scheduling level is low, so the task receives short amounts of CPU time because it is a relatively I/O-bound task. The quantum for a task in a low scheduling level is high, so it receives longer amounts of CPU time to reflect its needs as a relatively CPU-bound task. In order not to block higher priority tasks awaiting service, the scheduler allocates the quantum to a task in a series of time slices. At the end of each time slice there is an opportunity for a higher priority task to be scheduled. This ensures fair service to all users.

You can specify a maximum time slice. Do not, however, specify a maximum time slice that is smaller than the quantum for a level one task. To calculate the quantum for a level one task, use the average task size for tasks at that level.

Relate the quantum constant (C) to the type of disk in your installation. Make C equal to the time it takes to swap a task to disk (in ticks). The value of C varies, depending on the type of disk and the average size of tasks being swapped.

6.3.2 Promotion Time

The scheduler automatically promotes or demotes tasks between the interactive scheduling levels. If a task uses up all its quantum without relinquishing control (for example, waiting or exiting), it is demoted to the next lowest interactive level. See the *IAS System Management Guide*.

The promotion time parameter (number of ticks between scheduler promotions) ensures that tasks at lower levels are not starved of CPU time while having to wait for the next schedule. For each level in which no task has been scheduled during this time, the scheduler promotes one task to the next highest level. Specify the promotion time parameter to suit the type of tasks your installation runs. If you are running jobs that alternate between being I/O-bound and compute-bound, specify a lower value than the distributed default (for example, 100).

6.4 Batch Scheduling Parameters

Batch tasks can execute in a background scheduling level known as the batch level, which is the lowest level. Batch tasks timeshare with tasks at higher levels, but are neither promoted nor demoted. The batch level is only available on a timesharing system running PDS batch. The batch level is not available on a multiuser system. See the *IAS System Management Guide* for further information on batch scheduling.

The two parameters that control batch scheduling are the batch quantum and the maximum time between batch schedules, described in Section 6.4.1 and Section 6.4.2. You can change the batch scheduling parameters to suit different sessions. For example, in one session, batch obtains little service while in another session the scheduler allocates most resources to batch.

6.4.1 Batch Quantum

The batch quantum parameter specifies a quantum for batch tasks. The default value for the batch quantum is 25 clock ticks, so that a batch scheduling level is created by default. If you do not want a batch scheduling level, set the batch quantum to 0 (zero) before bringing up timesharing. Batch tasks are then scheduled and have memory allocated as does any other scheduler-controlled task.

WARNING: Do not set the batch quantum to 0 (zero) after system start-up.

To ensure that batch tasks are guaranteed time, set the batch quantum and the time between batch schedules to a non-zero figure.

6.4.2 Time Between Batch Schedules

The time between batch schedules specifies a period of time after which a batch schedule always occurs. Specify this parameter to guarantee that the scheduler services batch tasks.

Tuning the IAS Scheduler

If you want batch tasks to run in soak mode (that is, to soak up any remaining time left after interactive tasks have run), set the time between batch schedules to 0 (zero). This results in the scheduler not guaranteeing any time for batch tasks.

WARNING: Do not set the time between batch schedules to 0 (zero) on systems where scheduler controlled tasks are likely to be compute-bound for long periods of time. Under these conditions, the scheduler does not run any non-memory resident tasks while there is a compute-bound job running and a non-memory resident job to run. This situation persists until the compute-bound job becomes non-compute-bound.

If you intend to run a lot of batch tasks, set the batch quantum to a relatively high figure (for example, 25), and the time between batch schedules to a low figure (for example, 100). However, if your batch tasks are I/O-bound, decrease both these figures. Never set the batch quantum equal to or higher than the time between batch schedules.

6.5 Default Scheduling Parameters

When you have generated your IAS system, the scheduling parameters are set to the IAS default values (see Table 6-1). You can change the parameter values at system start-up, or dynamically during system operation (see the commands listed below). The scheduling parameters, their defaults, and the corresponding MCR UTL command switches, are listed in Table 6-1.

Table 6-1 IAS Scheduling Parameter Default Values

Parameter	Default	UTL Switch
Time between scheduler promotions (ticks)	1000	/SP:n
Time between batch schedules (ticks)	1000	/BS:n
Batch quantum (ticks)	25	/BQ:n
Idle time (ticks)	1	/IT:n
Maximum time slice (ticks)	3	/TS:n
Quantum constant (ticks)	2	/QC:n
Allocation factor memory size (blocks)	128	/AM:n
Allocation factor (ticks per memory size)	1	/AT:n
Number of scheduling levels	3 or 4	/LV:n

Table 6–1 (Cont.) IAS Scheduling Parameter Default Values

Parameter	Default	UTL Switch
Time factor per level	Level 1 = 1 Level 2 = 2 Level 3 = 4 Level 4 = 8	/TF:m:n
Maximum task size:	1030	/MT:n

To change these default values at system start-up, specify the SET BATCH and/or SET QUANTUM start-up commands. SET BATCH enables you to alter default values for the batch quantum and time between batch schedules. SET QUANTUM enables you to alter the time between scheduler promotions, the quantum constant, the allocation factor and the level time factor. The *IAS Installation and System Generation Guide* contains descriptions of both these commands.

To change the default values dynamically, specify the following commands:

- SET ALLOCATION (SCI) specifies the allocation factor.
- SET QUANTUM (SCI) specifies the batch quantum, the quantum constant and the level time factor.
- SET SERVICE (SCI) specifies the time between batch schedules and the time between scheduler promotions.
- UTL (MCR) specifies all the scheduling parameters.

See the *IAS System Management Guide* for details of the SCI commands, and the *IAS MCR User's Guide* for details of the MCR UTL command.

Although Digital provides default values for the scheduling parameters, these values are not suitable for all installations. The following sections give suggestions for altering these values to improve the performance of different types of system.

6.6 Systems Where Batch is Rarely Used

If your installation runs mainly interactive tasks with some non-critical batch jobs, you should allocate most resources to interactive users, and let batch run when there is time left. If your system has peak levels of activity, together with very low levels of activity, set the batch scheduling parameters to invoke batch soak mode (see Section 6.4.2). Batch tasks are not guaranteed time to run, but should be scheduled only when there is no other activity. Set the time between batch schedules to 0 (zero) to invoke soak mode.

For systems with relatively steady load characteristics, set the batch quantum lower than the maximum time slice value. Because a batch job receives all its quantum in one time slice, if the batch quantum is the same as the maximum time slice value, the batch job would receive all the time while small interactive jobs would suffer. Setting the batch quantum lower than the maximum time slice value overcomes this.

You can also set the time between batch schedules to a high value (for example, the default value of 6000 ticks) so that the scheduler does not often service batch tasks.

6.7 Systems Where Batch is Frequently Used

If your installation runs many batch jobs and you want a good turnaround time, tune the scheduling parameters' values to ensure that batch jobs obtain good service. This might result in poorer response for tasks in other levels. Set the time between batch schedules to a low value (for example, 200 ticks) to ensure that the scheduler services batch jobs at frequent intervals. Also set the batch quantum to a non-zero value (for example, 25 ticks) so that the batch level operates and thus guarantees time for batch jobs to run. If the corresponding drop in performance of interactive tasks is unacceptable, consider running separate batch sessions (for example, at night) when the interactive load is at a minimum. In an extreme case, by setting the time between batch schedules to a very low figure (for example, 10 ticks), and by setting the batch quantum to one clock tick less (for example, 9 ticks), you can ensure that batch processing obtains almost exclusive service.

NOTE: Do not set the time between batch schedules to less than 2 clock ticks. Do not set the batch quantum equal to the time between batch schedules.

6.8 Systems Where Most Tasks are Compute-bound

If most tasks in your system are compute-bound, you must keep tasks in memory longer so that they can use the processor without frequently being swapped in and out of memory.

Set the level time factor for level 3 (compute-bound level) higher than the default value. This increases the quantum for compute-bound tasks. However, any I/O-bound tasks running in this environment suffer. To overcome this, set the time slice parameter to a smaller value. As a task receives its quantum in a series of time slices, whenever a time slice expires a reschedule occurs, which allows other tasks to be serviced.

6.9 Systems Where Most Tasks are I/O-Bound

If most tasks in your system are I/O-bound, you must ensure that tasks go in and out of memory faster. Set the time slice parameter to a low value to make sure that reschedules happen at frequent intervals.

Reduce the quantum for level 2 (I/O-bound level), as I/O-bound tasks do not require large quanta. They need short, frequent amounts of time in between I/O operations. Increasing the allocation factor memory size (for example, from 1 tick per 200 octal blocks to 1 tick per 400 octal blocks) reduces the quantum (see Section 6.3.1). Alternatively, specifying a low value for the maximum time slice achieves the same effect.

6.10 Real-time Systems

In IAS real-time systems, scheduling is based on a task's priority. Each real-time task in the system has an associated priority, in the range 1 to 250 (decimal). 250 is the highest, or most urgent priority. IAS real-time scheduling constantly tries to allocate the processor to the highest priority executable task.

Real-time scheduling is under your control in that you can specify a task's priority and ensure that the task runs by assigning it a high value. You can alter a task's priority while it is running by means of the PDS command SET PRIORITY or the MCR command ALT (see the *IAS PDS User's Guide and the IAS MCR User's Guide*).

See the *IAS Executive Facilities Reference Manual* for a full description of real-time scheduling and the range of priorities available for user tasks.

6.11 Real-time Tasks in a Timesharing or Multiuser System

Running a large amount of real-time tasks in a timesharing or multiuser system could slow down the response for interactive users, because real-time tasks typically have higher priorities than timesharing tasks.

You can overcome this by specifying a separate partition for real-time tasks that control critical processes (see Chapter 2, Section 2.7). This improves response time because swapping out timesharing tasks to make room for a real-time task takes longer than checkpointing out a real-time task.

NOTE: Real-time tasks in a timesharing or multiuser system run under the control of the IAS scheduler if they are in the timesharing partition, and have an initial priority that is less than or equal to the scheduler priority.

7

EXAMPLE OF TUNING AN IAS SYSTEM

7.1 Introduction

This chapter contains an example of tuning one IAS system where poor performance was a problem. Tuning the system means, in this example, tuning the scheduling parameters. The results show how the settings of these parameters can affect IAS performance. The results also suggest improvements you can make by studying the type of jobs your installation runs, deciding what performance you want, and tuning the scheduling parameters accordingly.

NOTE: Performance figures given in this chapter apply only to this one example, and will not necessarily be valid for any other installation. The figures are examples of what you can achieve.

7.2 Improving a Slow Running System

The slow running system is a timesharing system running interactive tasks (25K to 28K words in size) with very few batch jobs. Slow running occurs when one user is running the task builder, and two other users are running FORTRAN tasks at the same time.

As described in Chapter 6, the following factors affect performance:

- Settings of the scheduling parameters
- Task size
- Whether the rest of the system is I/O-bound or CPU-bound
- Whether a batch job is running

As the task builder is a task with I/O-bound phases, a typical program was task built four times, each time with a different set of jobs running concurrently (see Table 7-1).

Table 7-1 Effect of Changing the IAS Scheduling Parameters

Set of Jobs Running	Number of Task Builds Running	Number of CPU-bound Tasks Running	Swapping?	Time Before Changes ¹	Time After Changes ¹
First Run	3	—	No	53	48
Second Run	1	2	No	137	66
Third Run	5	—	Yes	190	155
Fourth Run	1	4	No	386	139

¹Time in seconds.

EXAMPLE OF TUNING AN IAS SYSTEM

Level 2	12	ticks
Level 3	22	ticks
Level 4	25	ticks

The disk in this configuration is a half-full, compressed RP06. The typical access time is 15 to 20 ms; thus I/O-bound tasks obtain worse service in a CPU-bound system because the disk access time is much less than the average quantum. I/O-bound tasks would suffer even more when batch jobs run on this system because batch jobs are guaranteed time, as follows:

Initial Parameter Settings

```
SCI> SHO SCHEDULER
SCHEDULER PARAMETERS
SCHEDULER ENABLED IN PARTITION GEN , SPACE 20015
TIMESHARING PRIORITY: 100
TIME BETWEEN SCHEDULER PROMOTIONS: 104 CLOCK TICKS
PROMOTION TABLE SIZE:8
MAXIMUM TIME SLICE: 25 CLOCK TICKS
SYSTEM IDLE TIME: 1 CLOCK TICKS
MAXIMUM TASK SIZE: 1024 32-WORD BLOCKS
BATCH PARAMETERS:
    BATCH QUANTUM: 25 CLOCK TICKS
    TIME BETWEEN BATCH SCHEDULES: 500 CLOCK TICKS
QUANTUM PARAMETERS:
    QUANTUM CONSTANT: 2 CLOCK TICKS
    ALLOCATION FACTUR: 1 TICKS PER 160 MEMORY BLOCKS
    NUMBER OF SCHEDULING LEVELS: 4
    LEVEL: 1  TIME FACTOR: 1 CLOCK TICKS
    LEVEL: 2  TIME FACTOR: 2 CLOCK TICKS
    LEVEL: 3  TIME FACTOR: 4 CLOCK TICKS
    LEVEL: 4  TIME FACTOR: 8 CLOCK TICKS
```

After changing the parameter values, the quantum for each level for a task between 25K and 28K in size is as follows:

Level 1	4	ticks
Level 2	6	ticks
Level 3	10	ticks

No batch level exists because the batch quantum was changed to 0 (zero) at system start-up.

The same program was task built in exactly the same four ways as before the parameters were altered. The results are shown in Table 7-1.

These results show that the throughput of the system has increased. I/O-bound tasks run more quickly than before, especially in a compute-bound environment.

A batch job was run to ensure that it obtained a reasonable response. Batch response had decreased, but was still acceptable.

7.3 Changes to the Scheduling Parameters

The scheduling parameters were changed as follows:

- 1 The allocation factor memory size was changed from 160 blocks to 400 blocks. This decreases the quanta and helps I/O-bound jobs.
- 2 The promotion time parameter (time between scheduler promotions) was changed from 104 to 149 to help I/O-bound jobs.

EXAMPLE OF TUNING AN IAS SYSTEM

- 3 The batch quantum was changed from 25 to 0 (zero), because the system hardly ran any batch tasks and the total number of scheduling levels was reduced to 3. Setting the batch quantum to 0 (zero) means that no batch level exists. Batch tasks compete with interactive tasks for CPU time (see Section 6.4.1).

NOTE: When you alter IAS scheduling parameters, change only one parameter at a time and study the effect on the system. This is the only way to find out which parameter change has the biggest impact on your system's performance.

Revised Parameter Settings

```
SCI> SHO SCHEDULER
      SCHEDULER PARAMETERS
SCHEDULER ENABLED IN PARTITION GEN , SPACE 20015
TIMESHARING PRIORITY: 100
TIME BETWEEN SCHEDULER PROMOTIONS: 149 CLOCK TICKS
PROMOTION TABLE SIZE: 8
MAXIMUM TIME SLICE: 25 CLOCK TICKS
SYSTEM IDLE TIME: 1 CLOCK TICKS
MAXIMUM TASK SIZE: 1024 32-WORD BLOCKS
BATCH PARAMETERS:
    BATCH QUANTUM: 0 CLOCK TICKS
    TIME BETWEEN BATCH SCHEDULES: 500 CLOCK TICKS
QUANTUM PARAMETERS:
    QUANTUM CONSTANT: 2 CLOCK TICKS
    ALLOCATION FACTOR: 1 TICKS PER 400 MEMORY BLOCKS
    NUMBER OF SCHEDULING LEVELS: 3
    LEVEL: 1   TIME FACTOR: 1 CLOCK TICKS
    LEVEL: 2   TIME FACTOR: 2 CLOCK TICKS
    LEVEL: 3   TIME FACTOR: 3 CLOCK TICKS
```


A

Workbook

A.1

Workbook

This appendix contains a workbook to help you tune IAS. The workbook consists of a list of system components, their defaults (if any) and blank spaces for you to fill in the new values when you change them. This will enable you to keep a record of each change you make, together with the initial default.

Workbook

Area of System	Default	Changed To	Date of Change
Memory Management Directives - included?	No		
TERMINAL HANDLER PARAMS.MAC FILE			
B\$\$AW -	0		
B\$\$LCK -	0		
D\$\$C11 -	0		
D\$\$CDU -	0		
D\$\$H11 -	0		
D\$\$IAL -	0		
D\$\$J11 -	0		
D\$\$L11 -	1		
D\$\$LDU -	0		
D\$\$M11 -	0		
D\$\$Z11 -	0		
D\$\$ZDU -	0		
E\$\$SEQ -	0		
Number of batch streams	0		

Area of System	Default	Changed To	Date of Change
SCOM size	5120 words		
PARTITIONS:			
User-controlled (U)			
System-controlled (S)	S(real-time)		
Timesharing (T)	T (multi-user or timesharing)		
SYSTEM TASK EXTENSION SIZES:			
CRF /INC	10000		
LBR /INC	6500		
Number of terminals	1		
Spoiled device (SP)	SY		
Work file device (WK)	SY		
Swapping disk(s)	SY		
Swap ratio	Installation-dependent		
Disk ACP task	FCP.TSK		
MTAACP included?	No		
DTAACP included?	No		
Multiple disk ACP tasks included?	No		

Workbook

Area of System	Default	Changed To	Date of Change
FCPCOM size	516 words		
ACP task data area size	1 word		
INI /EXT or INIT/EXTENSION	5		
INI /LRU or INIT/ACCESSED	3		
INI /WIN or INIT/WINDOW	7		
MOU /EXT or MOUNT/EXTENSION	Initialization default		
MOU /LRU or MOUNT/ACCESSED	Initialization default		
MOU /WIN or MOUNT/WINDOW	Initialization default		
SCHEDULER PARAMETERS:			
Ticks between scheduler promotions	2500		
Idle time (ticks)	1		
Maximum time slice (ticks)	25		
Quantum constant (ticks)	2		
Allocation factor memory size (blocks)	200		
Allocation factor (ticks per memory size)	1		
Number of scheduling levels	4		
Maximum task size	1024		

Area of System	Default	Changed To	Date of Change
TIME FACTOR PER LEVEL:			
Level 1	1		
Level 2	2		
Level 3	4		
Level 4	8		
BATCH SCHEDULING PARAMETERS:			
Ticks between batch schedules	6000		
Batch quantum (ticks)	25		

Index

A

ACP task
 F11ACP • 5-2, 5-4
 MTAACP • 5-3
ACP task data area
 extending • 5-5
ACP tasks • 5-1

B

B\$\$AW • 2-2
B\$\$LCK • 2-3
Batch quantum parameter • 6-3
Batch scheduling level • 6-3
Batch scheduling parameters • 6-3
Batch streams • 2-4
BIGFCP.TSK • 5-1, 5-6

C

CONFIG.MAC
 consistency with PARAMS.MAC • 2-2
Control block
 file • 5-4
 file window • 5-4
 volume • 5-4
Controllers, multiple • 4-7
CREATE/SWAPFILE REALTIME command • 4-6

D

D\$\$IAL • 2-3
Default scheduling parameters • 6-4
DELETE/KEEP command • 4-5
Device handlers
 loading • 3-2
Disk attributes, table • 4-3
Disk configurations
 tuning • 4-7

Disk-resident overlays • 4-4
Disks
 effective use of • 4-1
 index files • 4-4
Disk Save and Compress utility • 4-5
Disk space, releasing unused • 4-5
DSC utility • 4-5

E

E\$\$SEQ • 2-2
EXAMINE command • 3-2
EXTK\$ • 2-3

F

F11ACP • 5-2, 5-4
FCB • 5-4
FCP.TSK • 5-1, 5-6
FCPCOM • 5-4
FCPCOM size
 extending • 5-4
File accesses
 reducing • 5-6
File Control block • 5-4
Files
 index • 4-5
 spooler • 4-1
File system
 disk access parameters • 5-7
 effect on IAS performance • 5-1
 file structures • 5-4
 tailoring • 5-1
File Window block • 5-4
FORTRAN-IV requirements • 2-1
Fragmentation • 3-2
Free space
 releasing • 4-5

I

IAS scheduler • 1-2

Index

IAS scheduler (Cont.)
 parameters • 6-1
 tuning • 6-1
IAS scheduling parameter default values • 6-4
Index file • 4-5
Index files on disks • 4-4
INITIALIZE command • 4-4, 5-6
Installing tasks • 3-2

L

Locked Block List • 5-4

M

Memory management
 directives • 2-1
 memory usage • 2-1
Memory-resident overlays • 4-4
Memory usage • 1-2, 3-1
 memory management • 2-1
 real-time tasks • 2-5
 reducing • 4-8
 terminal handler • 3-2
MOUNT command • 5-6
MSCP disk configurations
 tuning • 4-7
MTAACF • 5-3

N

N\$\$ODS • 3-1

O

OPE command • 3-2
Overlays
 disk-resident • 4-4
 memory-resident • 4-4

P

Parameters

Index-2

Parameters (Cont.)
 quantum • 6-2
PARAMS.MAC • 2-1, 3-1
PARAMS.MAX
 consistency with CONFIG.MAC • 2-2
Partitions • 2-5
Performance factors • 1-1, 1-3
 real-time systems • 1-3
PIPBLD.CMD • 2-3
Promotion time parameter • 6-3

Q

Quantum • 6-2
Quantum constant • 6-3

R

R\$\$BTB • 2-2
Real-time systems
 performance factors for • 1-3
RESFCP.TSK • 5-1, 5-6
RMS resident overlay libraries requirements • 2-1

S

Sceduling parameter default values • 6-4
Scheduler
 parameters • 6-1
 tuning • 6-1
SCOM
 node pool • 5-4
SCOM nodes • 3-2
SCOM size • 2-4
 maximum • 2-4
 node pool • 2-5
SHOW TASKS/TIMESHARING command • 4-7
Spooled devices • 4-1
Spooler files • 4-1
SUBMIT/AFTER command • 4-7
SWA command • 4-6
Swap files • 4-5
Swap ratio • 4-7
SYSBLD.CMD file • 4-6
System generation • 2-1

T

T/MSCP disk configurations

tuning • 4–7

Tasks

batch • 6–3

checkpointing • 4–5

compute bound • 6–6

demote • 6–3

extend • 2–3

I/O bound • 6–6

modifying size • 2–3

promote • 6–3

quantum • 6–2

real-time • 6–6

Terminal handlers • 2–1

maximum memory space • 3–2

multiple-terminal • 2–1

PARAMS.MAC • 2–1

single-terminal • 2–1

Terminals • 3–1

Time slicing • 6–2

TRUNCATE command • 4–5

Tuning the IAS scheduler • 6–1

Tuning the IAS system

example • 7–1

U

UMR usage

UNIBUS PDP-11 • 4–8

UQ\$SSP • 4–7

V

VCB • 5–4

Volume Control block • 5–4

W

Window block size • 5–7

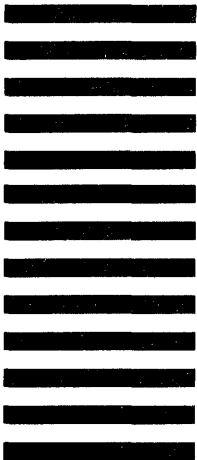
Work files • 4–3

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

IAS Engineering/Documentation
Digital Equipment Corporation
5 Wentworth Drive GSF/L20
Hudson, NH 03051-4929



Do Not Tear - Fold Here