RSX-11D

System Manager's Guide

Order No. DEC-11-OXSMA-C-D

RSX-11D

System Manager's Guide

Order No. DEC-11-OXSMA-C-D

RSX-11D Version 6.2

digital equipment corporation · maynard, massachusetts

First Printing, April 1975
Revised: November 1975
February 1977

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

CONTENTS

FIGURES

TABLES

PREFACE

Ø.1 MANUAL OBJECTIVES AND READER ASSUMPTIONS

The RSX-11D System Manager's Guide is intended for the individual who has overall responsibility for the operation of an RSX-11D system.

The system manager must have knowledge of RSX-11D in order to benefit from this guide. In particular, the system manager should consider the following documents prerequisites:

RSX-11D User's Guide,
RSX-11D Task Builder Reference Manual,
RSX-11D System Generation Reference Manual.

Ø.2 DOCUMENT STRUCTURE

This manual comprises the following chapters:

Chapter 1	Introduction to the manual and the significance of UICs and passwords,
Chapter 2	Operation of and output from task accounting,
Chapter 3	Use of spooling, despooling, and input spooling,
Chapter 4	Checkpointing,
Chapter 5	Global libraries and common areas,
Chapter 6	Core dump analysis,
Chapter 7	Time-scheduled partitions,
Chapter 8	Error logging,
Chapter 9	System management notes,
Chapter 1Ø	Executive debugging aids,
Chapter 11	Parity support,
Appendix A	Sample core dump analyzer output,
Appendix B	TKB15.CMD file,

Appendix C Error logging sample output and data record formats.

Appendix D 11/70 parity error response

Ø.3 ASSOCIATED DOCUMENTS

Refer to the _RSX-11D Documentation Directory_, order number DEC-11-OXUGA-D-D for a description of associated RSX-11D documents and their readerships.

CHAPTER 1

INTRODUCTION

The RSX-11D system manager is responsible for system features that fall into the following three categories:

1. Coordination and accounting of the multiuser environment,

2. Control of system operation,

3. System reliability monitoring.

The coordination of the multiuser environment consists of controlling the assignment and use of user identification codes (UICs) and their associated passwords as described in Section 1.1. UICs and passwords uniquely identify system users and establish their access rights. UICs also provide the basis for the accumulation of accounting statistics on executing tasks. Accounting details system use on a task-by-task basis, as described in Chapter 1.

Control of system operation consists of such features as output spooling and despooling of print files, input spooling of card files, the operation of checkpointing, and the implementation and use of global libraries and common areas.

Output spooling and despooling improves system operation by eliminating individual task contention for printers, teleprinters, and other serial output devices. Input spooling allows tasks to process card files from the disk rather than requiring them to wait for card reader availability. Output spooling and input spooling are detailed in Chapter 3.

Checkpointing is the mechanism used to aid high priority tasks to obtain the memory they require to execute as soon as possible after they are requested. Lower priority tasks are checkpointed to disk to allow higher priority tasks to run. Chapter 4 describes this process.

Global libraries are collections of routines that can be shared by many simultaneously active tasks. Common areas are data areas in memory that can be shared by many simultaneously active tasks. Both global libraries and common areas are loaded into memory dynamically when needed and their memory is released when they are no longer required as described in Chapter 5.

System reliability monitoring includes the core dump analyzer and the hardware error logging facility.

The core dump analyzer provides a means of interpreting the state of the software when a system crash occurred. It is described in Chapter 6 .

Chapter 7 contains a description of the algorithm used in time-scheduled partitions. This information should be taken into consideration when configuring a system that is to contain time-scheduled partitions.

The error logging facility allows the system manager to obtain a detailed and/or summary report of peripheral device errors. Error logging reports can be produced for specified time frames and for specified devices or volumes, as described in Chapter 8.

The system manager also is responsible for optimizing system operation. Chapter 9 contains system management notes for that purpose.

Executive trace and ODT are provided in RSX-11D for the debugging of user-initiated modifications to the Executive. These features are described in Chapter 10.

Chapter 11 details the software support for handling hardware parity errors. This information should be used in conjunction with the error logging description in Chapter 8.

1.1 UICS AND PASSWORD PROTECTION

Under RSX-11D, the key to controlling the multiuser environment is UIC and password assignment. Each system user should have a unique UIC. Each UIC is associated with a UFD on the system volume (SY). The UFD command, described in Chapter 6 of the RSX-11D User's Guide, is used to place UFDs on directoried devices. Any UFD placed on the SY volume also serves as a UIC. A UIC is not valid unless the same UFD exists on SY.

The group code portion of the UIC is used to determine the privileges extended to the user once the user has logged onto the system. Any group code of 10 (octal) or less indicates a privileged UIC. Its owner is granted access to system functions as a privileged user. Group codes greater than 10 (octal) indicate a general user with restricted access to system functions. The terms general user and privileged user are defined in the RSX-11D User's Guide. The following are examples of privileged UICs: [1,12], [5,20] and [3,3].

UICs are specified in the HELLO command that allows a user to log onto the system. The HELLO command is detailed in Chapter 5 of the RSX-11D User's Guide. The system validates UICs typed in the HELLO command to determine that the UIC is valid for the system.

Each UIC can have a password associated with it. When passwords are used, once the user types the HELLO command, a request is printed on the terminal as follows.

```
PASSWORD>
```

The user types the password at this point. The password is not printed on the terminal as it is typed. If the password corresponds to the UIC specified in HELLO, the user gains access to the system.

A UIC created by the UFD command does not have a password associated with it. The PWD command to MCR is used to assign a password if one is desired. The PWD command is detailed in Chapter 6 of the RSX-11D User's Guide.

When assigning UICs to system users, the system manager. should take into consideration the ways that UICs might be used. For example, if accounting is to be performed on tasks executing in the system, the UIC serves as the account number. The system manager should ensure that a meaningful system of assigning UICs is developed.

The RSX-11D system as distributed by DIGITAL contains privileged and nonprivileged UICs. These UICs are not password protected. The system manager can log on under any one of the privileged UICs to perform a system generation. The system manager can use the PWD command to assign passwords to any of the distribution system UICs as desired.

NOTE

All functions described in this manual must be performed by a privileged user.

CHAPTER 2

ACCOUNTING

RSX-11D accounting provides the system manager with a means of knowing how the system resources are being used by executing tasks. Accounting provides statistics on a task-by-task basis and also accumulates totals for each UIC (account). The user specifies at task build time whether accounting is to be performed for the task. Therefore, accounting information is accumulated only when the information is desired.

2.1 INTRODUCTION

Accounting statistics provide the following information for a task:

1. The amount of computer memory used by a task in terms of 32-word blocks,

2. The elapsed execution time; i.e., time in memory,

3. Central processor time used by the task indicated in clock ticks,

4. The number and types of accesses to each peripheral device used by the task.

Summary statistics contain totals for each UIC under which tasks specified for accounting are run. Figure 2-1 provides an example of task statistics and associated statistics.

Elapsed time is total time starting when the task is first made active and ending when the task exits. The time between a suspend request and a resume request is included in the elapsed time. Interrupt and Executive overhead time for the task are included in both central processor time and elapsed time.

Tasks that are requested more than once have a set of statistics for each time they execute. Each version of a multiuser task has a separate set of statistics.

```
ACCOUNTING REPORT FOR 19-FEB-75 TIME:  17:43:58

  ACCOUNTS FOR UIC [5,73]

   TASK:  ...FLX   STARTED AT:  19-FEB-75 17:43:49:22

     CORE USED:                CPU TIME:
        +415                      +2

     ELAPSED TIME:  Ø-Ø-Ø Ø:Ø:1:12

     DEVICE ACCESS SUMMARY:

     DEVICE:  UNIT:  READS:  WRITES:  OTHER:

       DP       Ø       Ø        Ø       1

       TT       7       1        1       3

  SUMMARY FOR UIC [5,73]

     CORE USED:                CPU TIME:
        +415                      +2

     ELAPSED TIME:  Ø-Ø-Ø Ø:Ø:1:12

     DEVICE ACCESS SUMMARY:

        +1 READS     +1 WRITES     +4 OTHERS
```

Figure 2-1 Accounting Output

Accounting provides data that can be used for billing based on computer usage. Because all accounting statistics are accumulated using UICs as account numbers, each account to be billed should be given a specific UIC or a related set of UICs so that statistics related to that account can be identified readily.

In addition, the system manager should be aware of how the system uses UICs for various functions. For example, a task run by a general (nonprivileged) user executes under the UIC that was specified in the HELLO command. On the other hand, if the same task is run by a privileged user, it executes under the UIC for which it was built or the UIC specified using the /UIC= switch with the RUN command. The UIC for which the task was built is the one given in the UIC= command to the task builder or the value to which it defaulted. When using batch, the /ACC= option on the $JOB command dictates the UIC under which tasks within that job execute, as described in the RSX-11D Batch Reference Manual.

Tasks which run at a priority above the priority of ACCLOG should NOT be built accountable.

Privileged tasks should not be made accountable if there is any possibility that they might be terminated for time limit by ACCLOG while modifying system data. It is NOT recommended that any distributed task be built accountable, except for the non-privileged system utilities. A batch file is provided for this purpose (see Section 2.3.4).

2.2 FUNCTIONAL DESCRIPTION

The actual statistics for each task are accumulated by a set of system subroutines and placed in the task's header area. Storing information in the task header means that statistics for a task usually need to be written to a disk file once only; i.e., when the task exits.

Two accounting tasks are required to 1) log the accounting information into a permanent disk file, and 2) analyze the accounting and produce the final output files.

2.2.1 Logging Task -- ACCLOG

The logging task, ACCLOG, takes statistics from the task header when the task exits and writes that information in a disk-resident file.

NOTE

> ACCLOG is a privileged, high-priority task that must be running at a priority of 235 or greater for accounting to take place.

ACCLOG creates two files on SY under UFD [1,5] in which statistics are stored. First, it opens and uses a file named ACCTS1.DAT and writes statistical data into it. When the analysis task runs, as described below, it sets global event flag 61 to indicate that it wants to process ACCTS1.DAT. ACCLOG tests the same flag. When it is set, ACCLOG stops updating ACCTS1.DAT and opens another file named ACCTS2.DAT. ACCLOG also sets global event flag 62 to indicate that analysis can proceed.

When the analysis task runs again to process the second file, it sets a flag to indicate this fact. At that point, ACCLOG creates a file named ACCTS1.DAT. ACCLOG maintains two files: one to store statistics in and one to be processed by the analysis task. ACCLOG alternates between creation of ACCTS1.DAT and ACCTS2.DAT.

2.2.2 Analysis Task -- ACCRPT

The analysis task, ACCRPT, analyzes the data file produced by ACCLOG. ACCRPT can be installed and run or scheduled at the user's discretion; unlike ACCLOG, it need not be resident constantly. When ACCRPT executes, it analyzes all information in the data file.

ACCRPT produces two files: a report file (ACCTS.RPT) that can be listed on a line printer and a data file (ACCTS.TOT) that can be processed by a user-written task, if desired. All files are under UFD [1,5]. Additionally, ACCRPT deletes the raw data file (ACCTS1.DAT or ACCTS2.DAT).

The records in ACCTS.TOT are variable in length and contain octal information providing task and UIC statistics. Figures 2-2 and 2-3 illustrate the fields and their lengths for task-specific and summary records, respectively.

Each time ACCRPT executes, it resumes ACCLOG and sets an event flag to tell ACCLOG to open a new file. When the old file is closed, ACCRPT reopens it and begins processing the data on a UIC and task basis. When finished, ACCRPT deletes the input file (ACCTS1 or ACCTS2).

		WORDS
RECORD TYPE	0	1
	TASK NAME	2
	(RADIX 50)	3
	UIC	4
	MEMORY USED IN 32-WORD BLOCKS	5
ELAPSED TIME	TICKS	6
	SECONDS	7
	MINUTES	8
	HOURS	9
	DAYS	10
	MONTHS	11
	YEARS	12
	PROCESSOR TIME	13
	IN TICKS	14
DEVICE ACCESS INFORMATION	DEVICE NAME (ASCII)	15
	UNIT NUMBER	16
	NUMBER OF READS	17
	NUMBER OF WRITES	18
	NUMBER OF OTHER ACCESSES	19

NOTE

1. The minimum record size is 14 words. The maximum is 110 words.
2. The device access information is repeated once for each device used by the task.
3. Except where noted, all information is in binary.

Figure 2-2 Format of Task Statistics Record (ACCTS.TOT)

		WORDS
RECORD TYPE	1	1
	UIC	2
	MEMORY USED IN	3
	32-WORD BLOCKS	4
ELAPSED TIME	TICKS	5
	SECONDS	6
	MINUTES	7
	HOURS	8
	DAYS	9
	MONTHS	10
	YEARS	11
	PROCESSOR	12
	TIME	13
	IN TICKS	14
DEVICE ACCESS INFORMATION	TOTAL NUMBER	15
	OF READS	16
	TOTAL NUMBER	17
	OF WRITES	18
	TOTAL NUMBER OF	19
	OTHER ACCESSES	20

NOTES

The record containing UIC statistics is 20 words in length.

All information is in binary.

Figure 2-3 Format of UIC Summary Records (ACCTS.TOT)

Each time ACCRPT runs, it creates new output files rather than updating the existing ones, thereby making it easier for the user to keep track of accounting information. It is the system manager's responsibility to delete these files when they are no longer needed.

If a system crash occurs, ACCRPT is not able to process the raw data file (either ACCTS1.DAT or ACCTS2.DAT) that was being used by ACCLOG when the failure occurred. However, the file remains on the disk and can be processed by the user. Figure 2-4 illustrates the record lay-out for the raw data file.

2.3 OPERATIONAL INFORMATION

Operational information describes 1) the procedures for initiating and terminating accounting and 2) the task builder and task installation requirements of the task to be accounted. Optionally, accounting can be performed for utility programs as described in this section.

NOTE

> If the system clock is not running at 60 (decimal) ticks per second, the analysis task (ACCRPT) must be rebuilt before it is used.
>
> To rebuild ACCRPT, first edit the file [11,26]RPTBLD.CMD by locating the following line.
>
> ```
> GBLDEF=TICKS:74
> ```
>
> Change the 74 (octal) to the correct number of ticks per second in octal. Then issue the following string to TKB.
>
> ```
> TKB>@[11,26]RPTBLD.CMD
> ```

		WORDS
	UNIQUE ID # (OCTAL) (See Note 1)	1
	TASK NAME IN RADIX 5Ø	2
		3
	UIC (OCTAL)	4
CPU TIME	3 (See Note 2)	5
	TOTAL TICKS	6
		7
	TIME LAST STARTED: TICKS	8
	SECONDS	9
	MINUTES	1Ø
CORE USED	2 (See Note 2)	11
	NUMBER OF 32-WORD BLOCKS USED	12
	4 (See Note 2)	13
START TIME	TICK	14
	SECOND	15
	MINUTE	16
	HOUR	17
	DAY	18
	MONTH	19
	YEAR	2Ø
DEVICE ACCESS (up to 16 of these fields can be present	1 (See Note 2)	21
	DEVICE NAME (ASCII)	22
	UNIT NUMBER (OCTAL)	23
	READ COUNT	24
	WRITE COUNT	25
	OTHER COUNT	26
END TIME (Always last field of record)	5 (See Note 2)	n
	TICK	n+1
	SECOND	n+2
	MINUTE	n+3
	HOUR	n+4
	DAY	n+5
	MONTH	n+6
	YEAR	n+7

NOTES

1. Uniquely identifies two tasks with the same UIC and name.

2. Field type code.

Figure 2-4 Record Format for Raw Data File

2.3.1 Initiating Accounting

Use the following procedures to initiate accounting.

1. If it has not been done already, mount the system disk using the MOU command and update the time and date using the TIM command.

2. Run ACCLOG from a privileged UIC using the following commands.

   ```
   MCR>RUN ACCLOG
   ```

   At this point, the system is ready to start accumulating statistics. If tasks designated for accounting execute before ACCLOG runs, no statistics accumulate for them.

3. When accounting statistics are desired, install and run ACCRPT using the following commands.

   ```
   MCR>INS[11,1]ACCRPT
   MCR>RUN ACCRPT
   ```

4. When ACCRPT terminates, use PIP to print ACCTS.RPT. Additionally ACCTS.TOT can be printed using the DMP utility program. All files are on UFD [1,5].

2.3.2 Terminating Accounting

Two separate tasks are provided to terminate accounting. The first, ACCABT, allows the operator to abort accounting for the entire system. The second, ACCOFF, allows the operator to terminate accounting in an orderly fashion.

When ACCABT runs, it aborts ACCLOG and sets an internal switch to indicate that no further accounting is to be performed. It makes no attempt to finish the logging of information for tasks that have not terminated nor does it attempt to analyze information in the raw data file.

On the other hand, ACCOFF first runs ACCRPT to preserve all accumulated data and then aborts ACCLOG. Normally, ACCOFF is used to perform an orderly shut down of accounting. ACCABT is used only in emergency situations to abort accounting.

Regardless of which method is used to terminate accounting, it can be restarted at any time by following the procedures in Section 2.3.1.

Use one of the following procedures to terminate accounting.

1. To perform an orderly shut down, install ACCRPT and ACCOFF and run ACCOFF using the following commands.

   ```
   MCR>INS [11,1]ACCRPT
   MCR>INS [11,1]ACCOFF
   MCR>RUN ACCOFF
   ```

2. To abort accounting, type the following commands.

   ```
   MCR>INS [11,1]ACCRPT
   MCR>RUN ACCABT
   ```

Both procedures cause a normal termination message to be printed on the terminal. Do not under any circumstances use ABO to abort ACCLOG, as it causes misinformation to be left in the system data base and could cause serious problems. Always use one of the above first.

2.3.3 Task Requirements (/TA and /TIM)

All tasks for which accounting information is to be accumulated must have accounting specified for them during task building. The /TA switch is used in the command string to specify accounting as described in the RSX-11D Task Builder Reference Manual.

NOTE

> The default for the Task Builder can be changed so that /TA becomes the default. Refer to Chapter 9.

Another option for controlling accounted tasks is the /TIM switch that can be used on the INS command to MCR. The /TIM switch allows the user to specify the amount of central processor time that a task can use. If the specified amount is exceeded, the task is aborted and its accounting information is recorded. The /TIM switch has the following format.

/TIM=nx

n is a decimal number indicating the number of time units (x) of central processor time that the task is allowed.

x indicates the type of time unit and can be one of the following.

H = hour
M = minutes
S = seconds
T = ticks

The amount of time specified must be no greater than 24 hours. For example, 24H, 1440M, and 86400S are the maximum amounts for hours, minutes, and seconds, respectively.

If /TIM is not included and accounting is specified for the task, the maximum amount of processor time allowed is five minutes. If the task is still running after five minutes of processor time has been used and /TIM was not specified, the system aborts the task.

2.3.4 Rebuilding Utilities to be Accounted

The system manager can rebuild all the RSX-11D utility programs, MACRO-11, and the FORTRAN compiler to have their use accounted. To simplify the rebuilding process, a batch command file is included in the system to rebuild all utilities, MACRO-11, and FORTRAN. The file is located under UFD [11,26] and is called UTLACCBLD.BIS. After UTLACCBLD.BIS is processed by batch, the new task images are located under UFD [11,26].

Use the following procedures to rebuild the utilities, MACRO-11, and FORTRAN to be accounted.

1. Redirect SY to the device containing the system object files.

2. Type the following command to request batch and process the file.

    MCR>BAT [11,26]UTLACCBLD

    If some, but not all, are to be accounted, the utility tasks, MACRO-11, and/or the FORTRAN compiler can be rebuilt individually. Table 2-1 lists the individual task build files to make a utility, MACRO-11, or FORTRAN accounted. All of the files are under UFD [11,26].

Table 2-1
Task Build Files for Accounted System Tasks

TASK BUILD FILE	ASSOCIATED MODULE
TKB04ACC.CMD	FLX
TKB05ACC.CMD	PIP
TKB06ACC.CMD	SLIPR
TKB07ACC.CMD	Librarian
TKB10ACC.CMD	MACRO
TKB11ACC.CMD	Task Builder
TKB20ACC.CMD	Editor
TKB41ACC.CMD	FORTRAN

2.3.5 Disk Storage Statistics

The /TB switch for PIP can be used to obtain statistics on the amount of disk storage that is used for a UFD. To determine the amount of storage used by a UFD on a specific device, type the following command to PIP.

    PIP>dev:[ufd]/TB

dev — is the device for which statistics are desired.

[ufd] — is the UFD whose usage is to be determined.

To determine the amount of disk storage used for all UFDs on a device, type the following command to PIP.

    PIP>dev:[*,*]/TB

2.3.6 Suggestions

1. Run ACCRPT only when necessary. Random running of ACCRPT results in many report files (ACCTS.RPT and ACCTS.TOT). Normally, ACCRPT needs to be run only when the system is to be shut down.

2. When the system is to be shut down, run ACCOFF rather than running ACCRPT and ACCOFF individually. ACCOFF automatically calls ACCRPT.

3. Use ACCTS.TOT as a basis for billing or further analysis. ACCTS.TOT is an octal file that can be processed by user-written tasks using FCS.

4. If accounting is to be used for billing purposes, ensure that UICs are assigned to individual accounts in a meaningful and easily processed fashion.

5. If many .TOT files are to be processed to produce one set of output, combine the .TOT files into one file using PIP. See the RSX-11 Utilities Procedures Manual.

2.4 ERROR MESSAGES

ACCLOG, ACCRPT, ACCOFF, and ACCABT issue error messages to the user. All of the accounting tasks assume that the SY handler is loaded, the device is mounted with correct access, and the device is not full. If this is not the case, error messages are printed on the terminal and the user must correct the situation.

2.4.1 ACCLOG Error Messages

ACCLOG -- FAILED TO OPEN NEW FILE, RUN ACCABT.

Use the following procedures to correct the situation:

1. Ensure that UFD [1,5] is present on the system disk,

2. Ensure that ACCLOG was running under a privileged UIC so that it has correct access rights,

3. If neither 1 or 2 above was the cause of the problem, check the device,

4. Run ACCABT,

5. Rerun ACCLOG.

ACCLOG -- IO ERROR; SOME DATA LOST

This message indicates that write errors have occurred and data has been lost for a task's account. Use the following procedures:

1. Check the file and device to see if they are usable,

2. If the errors persist, run ACCOFF to preserve accumulated statistics and turn accounting off until the cause of the problem is fixed.

2.4.2 ACCRPT Error Messages

ACCRPT -- FAILED TO OPEN NEEDED FILE, CANNOT PROCEED

filename

If the filename specified in the message is ACCTS1.DAT or ACCTS2.DAT, use the following procedures:

1. Ensure that ACCTS1 or ACCTS2 exists under UFD [1,5],

2. Ensure that ACCRPT was run under a privileged UIC so that it has the correct access rights,

3. Ensure that the file is closed properly and not locked,

4. If none of the above is the cause of the problem, check the device,

5. Rerun ACCRPT.

If the filename specified in the message is ACCTS.TOT, ACCTS.RPT, or ACCTS.TMP, use the following procedures:

1. Ensure that ACCRPT is running under a privileged UIC so that it has correct access rights,

2. Check the device,

3. Rerun ACCRPT.

If rerunning ACCRPT does not produce the desired results, use the DMP utility program to retrieve all data files except ACCTS.RPT. Use PIP to retrieve ACCTS.RPT or try to reanalyze the file.

ACCRPT -- IO ERROR, SOME DATA LOST

filename

If the filename specified in the message is ACCTS1.DAT, ACCTS2.DAT, or ACCTS.TMP, some input data has been lost. One task's account will be missing from the final analysis. Ensure that the file is not corrupted or the device is not malfunctioning.

If the error persists, stop analysis and print or dump all files under [1,5] to retrieve data and locate the problem.

If the filename specified in the message is ACCTS.TOT or ACCTS.RPT, a write error has occurred. Some data will be missing from the file. Data can be retrieved by comparing the ACCTS.TOT file with ACCTS.RPT unless both files are reporting problems.

If errors persist, print or dump all files under [1,5] to retrieve data and locate the problem.

2.4.3 ACCOFF Error Messages

ACCOFF -- PLEASE RUN ACCRPT TO PRESERVE DATA

The request from ACCOFF to run ACCRPT failed; probably ACCRPT was not installed. ACCOFF does not terminate accounting until all data is preserved. Type the following commands.

```
INS [11,1]ACCRPT
RUN ACCRPT
```

ACCOFF -- ABORT FAILED; ABORT ACCLOG FROM CONSOLE

For some reason ACCOFF was unable to abort ACCLOG. Use the ABO command to terminate ACCLOG, as follows.

```
ABO ACCLOG
```

2.4.4 ACCABT Error Message

ACCABT -- ABORT FAILED; ABORT ACCLOG FROM CONSOLE

For some reason ACCABT was unable to abort ACCLOG. Use the ABO command to terminate ACCLOG, as follows.

```
ABO ACCLOG
```

2.5 ACCOUNTABLE TASKS AND SAVING THE SYSTEM

The SAVE command (see the RSX-11D User's Guide, Section 6.23) preserves on disk a copy of the memory part of an RSX-11D system. The disk can thereafter be bootstrapped to restart the system.

A SAVE may be performed only when the system is quiescent, as defined in the section referred to. This includes, for example, no loading of tasks or peripheral transfers in progress. However, an accountable task can be resident and if a SAVE is performed, the time limit will be recorded and, on the disk being bootstrapped, will continue from where it was before the save. This can result in terminations for time limit for tasks that seem not to have run for very long.

CHAPTER 3

SPOOLING, DESPOOLING AND INPUT SPOOLING

RSX-11D provides three types of file spooling:

1. Automatic output spooling to disk,

2. Automatic despooling of disk files queued for printing,

3. Input spooling from a card reader to disk.

Automatic output spooling eliminates program contention for line printers, teleprinters, and other serial output devices. Because programs do not have to wait to use a serial device, higher system throughput is achieved.

The need to wait for a particular device is removed by temporarily redirecting the output intended for the serial device to a disk file. This temporary redirection is defined as automatic output spooling. The files redirected to the disk are later produced on the original device. This process is defined as despooling.

RSX-11D provides several levels of automatic output spooling/despooling. On the simplest level, the automatic output spooler can queue files for the single device despooler which prints them one at a time on a first-in/first-out basis. Only one output device is used at a time.

On a more sophisticated level, the automatic output spooler and the console user both can queue files for printing. In this case, the multiple device despooler is used. It sends queued files for printing on a priority basis. Up to 26 devices can be used for despooled output simultaneously.

The input spooler eliminates program contention for the card reader and serves as a batch job input device. Card files are read onto a disk and placed under a user-specified UFD.

3.1 AUTOMATIC OUTPUT SPOOLING

Automatic output spooling is implemented for use with programs using RSX-11D FCS. It is not operational with programs using the QIO directive. However, both automatic output spooling and programs issuing QIO directives can coexist when using the same output device. The program using QIO directives should adhere to the following procedure:

1. Attach the output device,

2. Space the form to headmof-form if using a line printer,

3. Issue the QIO directives,

4. Detach the output device.

If the task issuing the QIO directives is executing under a privileged UIC of [1,n], it must interrogate the status returned from the attach QIO tio ascertain whether the device already is attached. If the device already is attached, the calling task should wait for a significant event and then retry the attach.

3.1.1 System Requirements for Automatic Output Spooling

The following three system requirements must be met for automatic output spooling.

1. The SY device must have a UFD of [1,4] with read, write, extend and delete privileges for owner, group, system, and world. The following command is used to put the UIC on SY.

   ```
   MCR>UFD SY:[1,4]/PRO=[RWED,RWED,RWED,RWED]
   ```

2. A PUD for SP (spooler) must be created during system generation. A DEV directive to create a PUD for SP is included in the system generation command files on the distribution medium. The command has the following format.

   ```
   DEV=SP
   ```

3. The SP PUD is normally assigned to SY during Phase 2 of system generation by the redirect command:

   ```
   MCR>RED SY=SP
   ```

   Once the above command has been issued, the system manager has the option of redirecting the SP PUD to any disk device. Redirection can be performed at any time. The device that SP is assigned or redirected to must have a UFD of [1,4] with the same protection as defined above.

4. Both the devices assigned to SY and SP must be write enabled.

3.1.2 Program Conventions For Automatic Output Spooling

The following requirements must be met by the MACRO-11 program producing the file to be spooled.

1. The file storage region must contain at least a 512-byte output area to contain the output for the file that is targeted for the serial device. It is allocated using the FSRSZ$ macro. An open failure results on the output file if less than 512 bytes are supplied.

2. The program must open and close the output file.

3. An open by file identification or an open by filename block must be preceded by a call to .PARSE or by obtaining the directory identification.

4. The program must not attempt to regain access to the file once it is closed.

5. The maximum output record size must not exceed 256 bytes.

The FORTRAN compiler automatically adheres to the above procedures within the FORTRAN OTS. However, FORTRAN programs using the DEFINE FILE statement, must ensure that the maximum output record size of 256 bytes is not exceeded.

3.1.3 Enabling or Disabling Automatic Output Spooling

Automatic output spooling is enabled or disabled on an output device by typing the SET command to MCR. The syntax of the SET command to enable spooling follows with brackets indicating optionality.

```
MCR>SET /SP=dev:[,dev:,...,dev:]
```

/SP= is the command switch for automatic output spooling.

dev: indicates which device(s) are to have files spooled. dev is a 2-character device name and a 1- or 2-digit unit number. The device must be both a carriage-controlled and a record-oriented device, e.g., line printer or teleprinter.

Any attempt to set spooling of files for a device that is not a teleprinter or a line printer results in the printing of the following error message on the console.

```
SET -- SPOOLING NOT SUPPORTED IN THIS RELEASE FOR THIS DEVICE
```

The syntax of the SET command to disable spooling follows.

```
MCR>SET /-SP=dev:[,dev:,...,dev:]
```

/-SP= is the command switch to disable spooling.

dev: indicates those devices whose files are no longer to be spooled.

The SET command can be used at any time to enable or disable spooling. The SYS /DEV command can be typed to MCR to determine whether spooling is enabled or disabled for a particular device. Any device for which spooling is enabled is designated as follows.

```
dev: SPOOLED x
```

x is the designated form type for the device. Refer to Section 3.2.1.5.

3.1.4 Automatic Output Spooling Technique

The RSX-11D system performs special functions for programs writing spooled files when the file is opened, as individual records are written, and when the file is closed.

During open, when the output LUN is assigned to the file, the system determines whether the output device is spooled. If it is, the output device is redirected to SP. Normally, a file is created in UFD [1,4] with a 2-character filename indicating the final output device and a file type of .SPR. The file specification has the following format.

SP:[1,4]dv.SPR;n

dv — is the device on which the file is to be printed, e.g., LP.

n — is the file version. It is one greater than any current file having the same filename and file type in UFD [1,4].

FORTRAN programs differ in that they have the following file specification.

SP:[uic]FORnnn.DAT;n

uic — is the terminal user's UIC.

nnn — corresponds to the logical unit number used to create the file.

During program execution, each successive PUT$ results in placing the contents of the output buffer into the 512-byte block. When the file storage region block is filled, its contents are written into the file space on the disk.

When the file is closed, the file identification information is taken from the file description block (FDB) and sent to the despooler task using a SEND/REQUEST DIRECTIVE.

3.2 DESPOOLING

RSX-11D provides two despooling functions:

1. Single device despooler,
2. Multiple device despooler.

NOTE

The Multiple device despooler must be used if BATCH is to be run.

The single device despooler services SEND/RECEIVE requests on a first-in/first-out (FIFO) basis. The single device despooler is intended for smaller systems not requiring a system manager's operator interface by means of the OPR command to MCR or a terminal interface command by means of the QUE command to MCR.

The single device despooler has a file name of [11,1]PRT and a task name of SPR... .

The multiple device despooler can service up to 26 devices simultaneously. It consists of four tasks as listed in Table 3-1.

Table 3-1
Multiple Device Despooler Tasks

FILE NAME	FUNCTIONS	TASK NAME
[11,1]OPR	System manager's operator interface to despooling	...OPR
[11,1]QUE	User's console interface	...QUE
[11,1]SPR	Despooler queue manager	SPR...
[11,11]SPR2	Multiple device despooler	SPR2..

3.2.1 Multiple Device Despooler Options

The multiple device despooler has the following options that are not available with the single device despooler:

1. QUE and OPR commands to provide user and system manager interface with despooling,
2. Up to 26 output devices,
3. Despooling of any file on a Files-11 directory device.
4. Output by priority,
5. Multiple form types,
6. Capability to obtain multiple copies of a file,
7. Capability to delete or preserve the file after printing,
8. Batch output support.

3.2.1.1 QUE and OPR Commands to MCR - The QUE command provides the user with a terminal interface with despooling. The QUE command allows the user to perform the following:

1. Queue files to be despooled for printing on any teleprinter or printer,
2. List files that are queued for despooling,
3. Delete (kill) entries from the queue,
4. Modify entries in the queue. The following options can be modified for an entry:
   a. The output device to be used for printing,
   b. Priority of the entry,
   c. Form type to be used,
   d. Number of copies to be printed,
   e. Whether the file is to be deleted or preserved after printing.

The despooler handles all files identically regardless of whether they were queued by the automatic output spooler or the QUE command.

The OPR command is the system manager's interface with despooling. It allows the system manager to perform the following:

1. Stop output,
2. Resume output,
3. Resume output from the beginning of the input file,
4. Resume output from the last top-of-page encountered in the input file,
5. Abort output,
6. Begin despooling output,
7. Change the form type,
8. Indicate that a form type change is to occur and set up the printing of files to test form type alignment,
9. Recycle all devices that were being used for despooling prior to a system crash. This option provides a recovery of despooling operations.

The QUE and OPR commands are detailed in the RSX-11D User's Guide.

3.2.1.2 Multiple Output Devices - Files can be despooled to up to 28 output devices simultaneously. Any printer or teleprinter can be specified as the output device. The output device does not have to be specified for spooling in an MCR SET command. Setting a device for spooling only affects automatic output spooling; it has no effect on the despooling operation. That is, a device does not have to be spooled to have files queued to it for despooling.

3.2.1.3 Despooling of any File - Any file on a Files-11 directory device can be designated in a QUE command for despooling. The file to be printed should be other than a binary file, such as a data, source, list, or map file.

An attempt to print a binary file such as an object or task-image file results in a despooler error and an automatic abort of despooling operations for the file.

3.2.1.4 Despooling Priority - Priority determines the order in which files are despooled. The priority can be specified in the QUE command or the PRINT$ macro or it can default to the task's running priority. Priority is a decimal number in the range of 1 through 250.

3.2.1.5 Multiple Form Types - The type of form on which the file is to be printed can be specified by a number in the range of one through seven. The form type value itself is contained in the PUD for the device and can be displayed using the SYS /DEV command. The system is generated with a form type of 0 for every printer and teleprinter.

The PUD form type value can be changed dynamically by the system manager using the OPR command. Two switches to the OPR command are provided for this purpose.

1. /CHG alters the form type of the specified device to form type 7. Form type 7 is reserved for the despooling of test files to verify the form alignment. Test files can be queued by using the /TE option of the QUE command.

2. /FO:n alters the PUD form type to the value n; n is in the range from Ø through 6.

The system manager determines which form types correspond to the numbers zero through six. The OPR command is used to specify the value for the current form type for a particular device by placing that value in the PUD.

The system manager can assign separate values for different form types or have several form type values correspond to the same type of printer paper. However, the following facts should be considered.

1. Files queued to be printed on a particular device with a specified form type are printed only when the PUD for that device has a matching form type value.

2. Form type Ø is the form type initially contained in the PUD for each printer or teleprinter; therefore, it is logical to define the most commonly-used printer form as form type Ø.

3. Despooling of queued files onto a device with a matching form type is automatic. Differing form types require the system manager to alter the PUD form type using the OPR command.

3.2.1.6 Multiple Copies of the Printed File - The user can specify the number of copies of a file that are to be printed. The number of copies is a decimal number in the range of 1 through 31.

3.2.1.7 Delete/Preserve Indicator - The user can specify whether the file is to be deleted after printing or to be preserved for subsequent use.

3.2.1.8 Account Number - The account number is the UIC of the user who queued the despool request. The account number determines access rights of a nonprivileged user to modify or delete a queued request. It also determines which entries are to be listed in response to a Queue List MCR command.

3.2.2 Functions of the Despooler Tasks

QUE, OPR, SPR, and SPR2 are the four tasks used to achieve multiple device despooling. The functions of QUE and OPR are presented in Section 3.2.1.1. SPR and SPR2 are described below.

3.2.2.1 SPR (Queue Manager) - SPR is the queue manager. It receives SEND/REQUEST directives that are directed to the multiple device despooler and creates an entry in the queue. The despooler queue is a file that is maintained on the system disk under UFD [1,4]. The file is created by the queue manager should no file exist. The file specification for the queue follows.

SY:[1,4]SPRQUEUE.SYS

SPRQUEUE.SYS is a dynamically expanding file capable of holding up to 2047 (decimal) queued items. The queue manager selects files (queue entries) to be despooled based on the number of devices the multiple device despooler can service simultaneously. It issues a SEND/REQUEST directive to the multiple device despooler. The queue manager is active only when it is queuing an entry or engaged in sending a new file specification to the multiple device despooler.

The selection algorithm for determining which entry to send to the multiple device despooler is illustrated in Figure 3-1.

3.2.2.2 SPR2 (Multiple Device Despooler) - SPR2 is the multiple device despooler task. It writes the files indicated to it by SPR on the designated printer or teleprinter. SPR2 is AST-driven and handles up to 26 devices simultaneously depending on the size of the task when installed.

SPR2 supports one device automatically. For each additional device to be supported, another 460 words must be added to the size of SPR2 when it is installed. The /INC switch is used in the INS command to MCR to increase the task size. Multiply the number of devices to be supported by 460 to determine the appropriate value to use in the /INC switch. For example, if three devices are to be supported, type the following command when SPR2 is installed.

```
MCR>INS [11,1]SPR2/INC=1380        (3 * 460 = 1380)
```

3.2.3 Interfaces to the Despooling Tasks

RSX-11D provides four interfaces with the despooling tasks:

1. Automatic output spooling triggered by the closing of a file that is assigned to a spooled device,
2. The PRINT$ macro call,
3. The QUE command to MCR,
4. The OPR command to MCR.

Automatic output spooling is operational for both the single and multiple device despoolers.

3.2.3.1 Print$ Macro Call - The PRINT$ macro call is the program interface with the despooler. It is operational with both the single and multiple device despooler. The syntax of the PRINT$ macro is described in the IAS/RSX-11 I/O Operations Reference Manual.

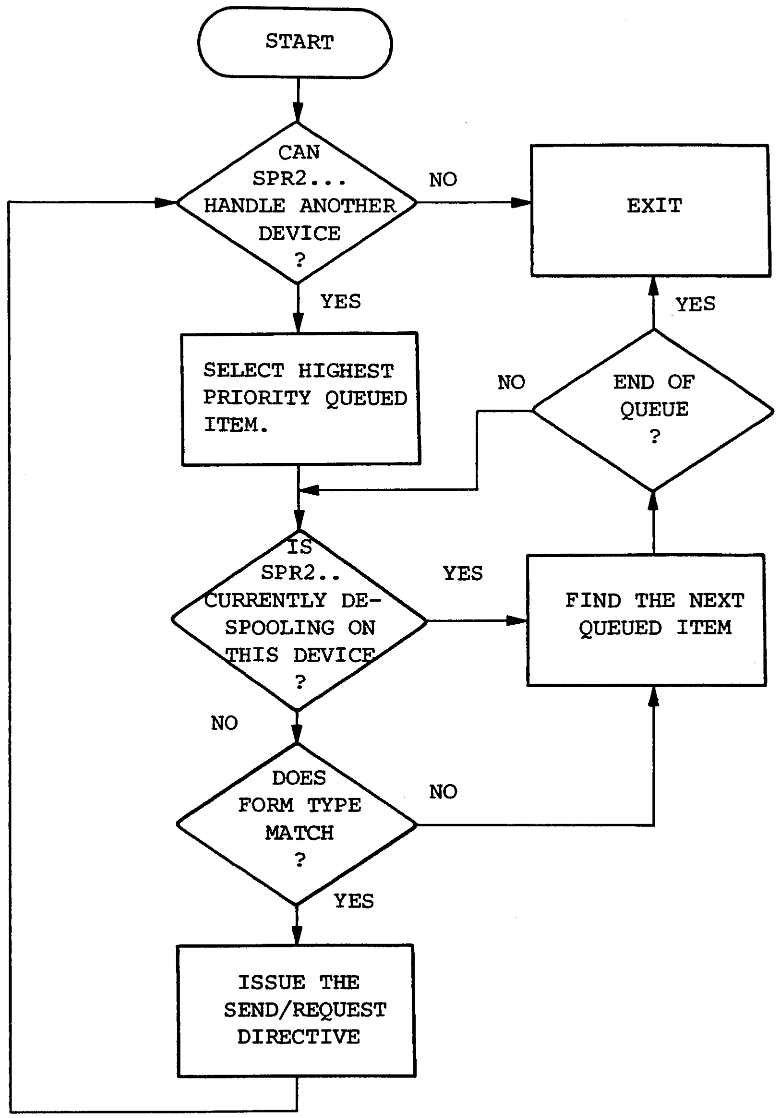

Figure 3-1 Queue Entry Selection Algorithm

The multiple device despooler is capable of using more of the parameters passed to it by PRINT$ than the single device despooler can. Table 3-2 summarizes the differences.

Table 3-2
PRINT$ Capabilities of the Single and Multiple Device Despoolers

PARAMETER	SINGLE DEVICE	MULTIPLE DEVICE
Output device	yes	yes
Input filename	yes	yes
Priority	no	yes
Form type	no	yes
Number of copies	1 only	1 through 31
Delete/preserve the file	delete	either

If parameters that do not apply for the single device despooler are included in a PRINT$ macro call to that despooler, they are ignored.

3.2.3.2 QUE and OPR Commands - The QUE and OPR commands to MCR are the user's and system manager's terminal interface to the despooling tasks. Both commands are summarized in Section 3.2.1.1.

This interface is operational only for the multiple device despooler. The system manager should install these tasks only if the multiple device despooler is installed.

3.2.4 Priorities of the Despooling Task

All of the despooler tasks run in the GEN partition with a priority of 100. If the system manager wishes to change their priorities, he should consider the following paragraphs.

3.2.4.1 OPR - The priority of OPR should be maintained at a high value. The task must be able to communicate instantaneously with SPR2 (multiple device despooler).

3.2.4.2 QUE - The QUE task is a multiuser task. Its running priority is not critical and can be altered to any value. However, ideally, its priority should not exceed that of the queue manager (SPR). Multiple SEND/REQUEST directives to SPR, such as a number of modify or kill QUE commands could cause the pool limit of 100 to be exceeded.

3.2.4.3 SPR (Queue Manager) - The queue manager should operate at a priority of 100. It should process SEND/REQUEST orders in a reasonably short time. SEND/REQUEST orders originate from any task desiring a despooling operation, e.g., MACRO-11, FORTRAN, and the task builder.

It is not recommended that SPR's priority be increased above 100 because the efficiency increases proportionally to the number of

SEND/REQUEST orders queued. Under these conditions, the queue manager performs a minimum number of read operations or write operations of the queued file.

3.2.4.4 SPR2 (Multiple Device Despooler) - The multiple device despooler can operate at any priority from 1 through 240. The system manager should determine the optimum priority for his system. The following points should be considered.

1. The number of queued input files increases if the despooler task is not able to despool files at the same rate that they are queued. An increase in the number of files queued slightly reduces the performance of the queue manager.

2. If the system manager desires more work performed by other processes than the despooler, the despooler can be lowered to a priority that is less than the priority of the other processes.

3. If it is desirable to expedite the output of certain files, the priority of the despooler can be altered dynamically using the MCR Alter Priority command. The change in priority should be lowered after the file has been despooled.

4. The multiple device despooler should not have a priority greater than 210. A priority greater than 210 may cause the despooler to interfere with device handler or F11ACP task execution.

3.2.5 Error Recovery Procedures

Error recovery procedures for device or file errors, internal despooler errors, and queue manager errors are described in the following paragraphs.

3.2.5.1 Device or File Errors - The following procedures should be used if persistent errors occur on a given device.

1. Execute the MCR OPR command to change the form type for the device.

   ```
   MCR>OPR dev:/CHG
   ```

2. Abort the file being processed on the device receiving errors.

   ```
   MCR>OPR dev:/AB
   ```

3. Perform a redirect of the device to an alternate device, if appropriate.

4. Once the redirect has been performed or the device has been repaired, reset the form type to its original value.

   ```
   MCR>OPR dev:/FO:n
   ```

3.2.5.2 Internal Despooler Errors (Task SPR2..) - The following procedures should be used if an internal error occurs in the despooler task.

1. Obtain a list of all queued files.

   MCR>QUE/AL

2. Repeat the MCR OPR change of form type command for all output devices in the queue.

   MCR>OPR dev:/CHG

3. Abort the despooler task.

   MCR>ABO SPR2.../TI=TTØ:

4. Perform an MCR OPR recycle command to make all active orders inactive. The /RE switch is used to recycle.

5. Reset all changed forms to their original value.

3.2.5.3 Queue Manager Errors (Task SPR...) - If persistent queue manager errors such as read/write errors on the queue file occur or if the queue manager enters a loop, the following error recovery procedures should be used.

1. Abort the queue manager if it is active.

   MCR>ABO SPR.../TI=TTØ:

2. If possible, dump the queue file. Use the following commands to disable spooling for the line printer and to dump the file.

   MCR>SET /-SP=LP:
   MCR>DMP LP:=[1,4]SPRQUEUE.SYS

3. Delete the queue file ([1,4]SPRQUEUE).

4. Obtain a directory of [1,4] and if it contains any files of type .SPR requeue these files using the following command.

   MCR>QUE [1,4]*.SPR;*/DE

5. Enable spooling for the line printer, if it was disabled.

   MCR>SET /SP=LP:

3.3 INPUT SPOOLING

RSX-11D provides an input spooling task called [11,1]SPL. Once installed, the task can be called by any task issuing a SEND DATA AND RESUME OR REQUEST directive. Th input spooler is a multiuser task that can be called simultaneously by more than one task. The task reads variable length data from an input device.

To call the input spooler, data sent to the input spooler should have the following format: the first word contains the 2-character ASCII device name and the second word contains the unit number in binary of the input device.

Once the input spooler task is called by the send directive, it receives the same TI as the TI assignment of the task issuing the directive. When invoking several simultaneous versions of the input spooler task, the user should ensure that the tasks issuing the send/request directives have different TI assignments.

The input spooler reads double-buffered, variable-length input data. The data can have a maximum length of eighty characters.

All RSX-11D card reader handlers (CRBR and CRNP) automatically call the input spooler if input spooling is enabled on the card reader device.

During the input spooling of card decks, files are spooled directly into the user's UFD which must be specified on the $CREATE card. For input spooling of BATCH job files the same system requirements must be met as for output spooling (see Section 3.1.1).

3.3.1 Enabling and Disabling Input Spooling

Input spooling is enabled and disabled using the SET command to MCR. Use the following command to enable spooling.

```
MCR>SET /SP=CRn:
```

Use the following command to disable spooling.

```
MCR>SET /-SP=CRn:
```

The DEV command can be issued to MCR to determine whether a card reader is spooled or not. If it is spooled, it has the word SPOOLED printed with the card reader status.

3.3.2 Basic Operations

When files are to be spooled from the card reader, set the device as being spooled, place a deck in the card reader input hopper and depress the READY switch. This action causes the card reader handler to request the input spooler task (SPL...). If the input spooler is not installed, the following message is printed on the terminal.

```
***CRn SPL TASK NOT INSTALLED
```

The console operator should install the input spooler; operation resumes automatically with a maximum delay of one minute.

Once the input spooler is loaded, it reads cards until no more remain in the card reader input hopper.

3.3.3 Format of the Input Deck

Input decks consist of one or more data decks. Each data deck must be peceded by a header card and followed by an EOF card. The CREATE card is the data deck header. It is followed by the actual data cards and an EOF card as illustrated in Figure 3-2. The formats of the header and EOF cards are described below.

```
CREATE filespecification/switches

   data cards

         .
         .
         .
         .
         .
         .

EOF card
```

Figure 3-2 Format of Input Data Deck

3.3.3.1 Format of the CREATE Card (Data Deck Header) - The CREATE card has the following format.

```
col 1

     CREATE dev:[ufd]filename.type;version/switches
```

CREATE must appear in columns 1 through 6. The file specification is the standard RSX-11D file specifier. Table 3-3 lists default fields in the file specifier. Table 3-4 lists and defines the switches that can be used on the CREATE card.

Table 3-3
Defaults in CREATE File Specifier

FIELD	DEFINITIONS	DEFAULT
dev*	The device onto which the file is to be spooled. It must be a directoried disk device.	SP:
[ufd]	A valid UFD that is defined on dev:.	No default. Must be specified.
filename	Name to be used for the file spooled onto the disk.	No default. Must be specified.
type	File type	DAT
ver	File version can be specified or omitted.	If /-NV, ver is 1. If /NV, ver is one greater than the existing version number.

Table 3-4
CREATE Card Switches

SWITCH	DEFINITION	DEFAULT
/BL	Suppress all trailing blanks	/BL
/-BL	Do not suppress trailing blanks	
/SE	Delete card columns 73 through 80 (sequence number)	/-SE
/-SE	Do not delete columns 73 through 80	
/NV	Create a version number that is one greater than the existing version of the file.	/NV
/-NV	Do not create a version number that is one greater than the existing version of the file.	

The input spooler first searches the UFD on the indicated device to determine whether the specified file already exists. If the file exists, a new file is not created unless the /NV switch has been specified. If /NV is not specified, the following message is printed on the terminal.

```
SPL -- FILE ALREADY EXISTS
```

This message is followed by a printout of the header card.

If there are no errors in the file specification, the data cards up to but not including, the EOF card are written onto the disk.

NOTE

Neither the CREATE card nor the EOF card is written in the file.

3.3.3.2 Format of the EOF Card - The EOF card contains a 12, 11, 0, 1, 6, 7, 8, 9 multipunch in card column 1.

3.3.4 Input Spooling Support of BATCH

In addition to processing CREATE cards, the input spooler processes standard RSX-11D batch job control cards. A $JOB card either as the first card in the card reader input hopper or following an EOF card is treated in the same manner as the CREATE card.

3.3.4.1 $JOB card - The format of the $JOB card is identical to the $JOB card described in the RSX-11D Batch Reference Manual. The user should use the /ACCOUNT switch to establish the proper UIC for accounting. The batch file is spooled onto the physical device to which SP is directed, under UFD [1,4]. If the /NAM switch is included, the specified job name becomes the name of the file on disk. In the following example, the file is named BJOB12. The file type defaults to .BIS.

```
$JOB/NAM=BJOB12/LIM=200
```

If the /NAM switch is not included, the default filename is RSX11DBAT, which also is the default batch job name.

3.3.4.2 Processing of Spooled Batch Input - The $JOB card causes a file to be created on disk. The disk file contains the $JOB card followed by data cards, followed by the $EOJ card.

The input spooler closes the output file once it reads the $EOJ card. To combine several jobs in one disk file, the user should omit the $EOJ card from each separate job and terminate only the last job with a $EOJ card.

3.3.4.3 Automatic Calling of Batch - Provided the $EOJ card is followed by two successive EOF cards (3.3.3.2), the input spooler closes the spooled file on disk, then issues a SEND/REQUEST directive to batch (...BAT). If ...BAT is installed, it automatically starts processing the file. The input file created by the input spooler is deleted automatically by batch after it processes the input data. If it is not installed, the operator receives no notification. This approach permits batch job files to be created with or without automatic calling of batch.

3.3.5 End of Input Spooling

When the input spooler reads two successive EOF cards, it prints the following message on the operator's terminal.

```
SPL -- END OF SPOOL INPUT
```

If the two successive end cards are followed by an input hopper empty condition, the card reader handler enters a waiting state and the input spooler exits. If the input hopper is not empty and spooling is still requested for files, the card reader begins reading cards. Input decks can be placed one behind the other in the input hopper or the hopper can be allowed to empty before a new deck is placed in the hopper. Pressing the RESET button again resumes input spooling.

3.3.6 Disk Files Created by Input Spooler

With the exception of batch files, the disk files created by the input spooler are not deleted by it. The files must be deleted under user control. The files cannot be accessed using the CR: input designator; they must be accessed using the full input file specification. This specification is identical to the file specification supplied on the CREATE card or to the $JOB default specification, as described in Section 3.3.4.1.

3.3.7 Aborting a Batch Job

Ordinarily batch jobs should not be aborted. However, if it becomes necessary to abort a batch job, use the following procedure.

1. Abort the task BPR.

   ```
   ABØ BPR.../TI=BPØ
   ```

2. Obtain an active task list.

   ```
   ACT /ALL
   ```

3. Abort any task that has a TI designation of BPØ.

4. Delete all batch files created for the execution of the aborted batch job. If only one batch job is queued or executing under the UFD that contained the batch job, the files should be deleted using the command:

   ```
   PIP RSX11DBAT.*,*/DE
   ```

   If more than one batch job is queued for the particular UFD, wait until the queued batch jobs complete and then deleted the files using the PIP command described above.

5. Set the invoking terminal to a nonspooled state.

   ```
   SET /-SP=TTn:
   ```

6. Rerun the task BPR... to resume processing of other batch jobs within the queue.

   ```
   RUN BPR...
   ```

7. The partial listing output from the aborted batch job is printed at the completion of the next batch job.

CHAPTER 4

CHECKPOINTING

Checkpointing is a mechanism to aid high priority tasks obtain the memory required to execute as soon as possible after they are requested. The process provides for temporarily halting the execution of one or more tasks in an orderly fashion to free memory for execution of a higher priority task. A task image of each halted task is rolled onto disk to free memory for the higher priority task. Checkpointing also is known as rollout/rollin.

Individual tasks can be built at task build time as checkpointable or non-checkpointable (see the RSX-11D Task Builder Reference Manual). The checkpointability of a task can be disabled/enabled using the directives described in the RSX-11D Executive Reference Manual.

Two factors are critical to the operation of checkpointing:

1. Whether any executing tasks have been designated as checkpointable during task building.

2. The priority of tasks that require memory for execution.

To control these factors, the Executive maintains information about which tasks can be checkpointed and maintains the memory required list (MRL) of tasks waiting for execution. Checkpointable task list (CTL) information is threaded through two system tables: the active task list (ATL) and the memory required list (MRL). The layout of the ATL and MRL is provided in Appendix A of the RSX-11D Executive Reference Manual.

4.1 CHECKPOINTABLE TASK LIST (CTL)

Every system-controlled partition has an associated checkpointable task list (CTL). Each CTL is a priority-ordered list of all checkpointable tasks that are active in or queued to run in the partition. All tasks are checkpointable unless specified as not checkpointable (/-CP) during task building. CTL is scanned in reverse order to ensure that the lowest priority tasks are the first candidates for checkpointing.

4.2 MEMORY REQUIRED LIST (MRL)

The memory required list (MRL) is a priority-ordered list of tasks that are waiting for memory to become available in a partition. There is one MRL for each partition. If more than one task is in the MRL, lower priority tasks are not considered for execution until the highest priority task starts execution.

4.3 FUNCTIONAL DESCRIPTION

When a task is listed in the MRL, the following series of events occurs.

1. A routine scans the CTL to determine whether a checkpointable task with a priority lower than the task awaiting memory is executing. The scan occurs wjenever a task is inserted or reinserted in the MRL.

2. If a checkpointable task is running, a routine allocates a checkpoint image for the task on disk and, if successful, performs either of the following:

   a. If the task has no I/O in progress, calls the rollout routine.

   b. If the task has I/O in progress, marks the task for checkpointing. When a task is marked for checkpointing, no I/O requests are dequeued for it. When the I/O in progress is completed, the I/O done function calls the rollout routine.

3. The rollout routine initiates the actual rolling out of the task image and sets its status to suspended for checkpointing.

4. The task remains in the ATL until it is recorded on the disk.

5. Another routine relinks the task´s ATL node to the MRL at the task´s original priority.

6. after the rolled out task´s memory is freed, an attempt is made to activate the task that forced checkpointing. If this routine fails to find sufficient memory, it calls the routine in step 1 to try to roll out another checkpointable task.

7. when a checkpointed task returns to memory, the appropriate handlers are restarted, ASTs are restarted, and the disk area used for the task image is released.

NOTE

When the checkpointed task resumes execution, it does not necessarily occupy the same memory area in which it was executing; i.e., any available memory area of sufficient size in the same partition can be used.

CHAPTER 5

GLOBAL LIBRARIES AND COMMON AREAS

RSX-11D provides a means of using and maintaining global libraries and common areas. Global libraries are collections of routines that can be shared by many simultaneously active tasks. Global common areas are data areas in memory that can be shared by many simultaneously active tasks.

In fact, global libraries and common areas are the same type of entity. Both are treated similarly by the system. The only real difference is in terminology to aid the user in understanding their functions. For this reason, global libraries and common areas are referred to as shared global areas (SGAs). SGAs are completely dynamic; without requiring a system generation. Each task can use a maximum of seven SGAs.

5.1 PURPOSE AND USE OF SGAs

The purpose of a shared global area (SGA) is to allow the system designer to share code and data among simultaneously active tasks. For example, the FORTRAN compiler and the editor share the SGA (library) named SYSRES when they are both active at the same time. Similarly, two or more FORTRAN programs can exchange data through an SGA (common area).

An SGA is loaded into real memory whenever a task that links to it is requested for execution. However, if the SGA already has been loaded into memory by a previous task execution, the in-memory version is used; duplicate copies of SGAs never occur.

An access count is maintained for each SGA. An SGA remains in memory as long as the count is greater than zero. When a task exits or is checkpointed to the disk, the access count for any SGAs that the task refers to is decreased by one. If the count becomes zero, the memory allocated to the SGA is released. In addition, if the SGA was installed in the system as a read/write, as often is the case with common areas, the SGA is written to the disk when the access count becomes zero.

NOTE

> SGAs are not checkpointable and never are checkpointed. However, read/write SGAs are written back onto the disk over

the original file when their access count becomes zero.

Read-only SGAs simply have their memory released when the access count becomes zero.

The access count becomes zero if, for example, all tasks linked to the SGA are checkpointed.

The read-only (pure) area of multiuser tasks is treated in exactly the same manner as the read-only SGAs. The only difference is that the SGA is global to many tasks (i.e., referred to by name) and the pure area of a task is local to the task with which it was built. Multiuser task read-only areas are created, installed, loaded, and removed, automatically by the system.

5.2 IMPLEMENTATION OF SGAs

An SGA is contructed and introduced to the system in three distinct phases, namely:

writing the SGA,

Task building,

Installation and removal.

5.2.1 Writing the SGA

Because SGAs are used to perform a variety of functions, the following generalizations should be considered when writing an SGA.

1. Read-only libraries of routines must be written in MACRO-11 assembly language. The user is responsible for all re-entrancy problems. The user must write his own re-entrant code.

2. Read/write areas can be written in MACRO-11 language or as FORTRAN block data subprograms. Re-entry problems do not exist for read/write SGAs.

   If the SGA is written as a FORTRAN block data subprogram, it is recommended that the data be initialized by the DATA statement.

3. The user is responsible for synchronization of accesses to the data base in read/write areas.

5.2.2 Task Building the SGA

The RSX-11D Task Builder Reference Manual contains a complete description of the task builder and its options. The following descriptions are intended to provide an outline of the procedure for and the main points of task building an SGA.

First, the user is responsible for determining whether an SGA is position independent. Refer to the IAS/RSX-11 MACRO-11 Reference Manual for a discussion of position independent code. Position independency indicates the ability of a bound module (either code or data area) to exist anywhere in the virtual address space of a task. All FORTRAN block data subprograms must be built as being position independent; they are so by definition.

Secondly, the SGA must be task built. The user must be operating under UIC [1,1] when TKB is requested. The output from the task build operation is the following:

1. A task image of the SGA under UFD [1,1],
2. A symbol table file under UFD [1,1],
3. Optionally, a map file.

The following task builder input is mandatory:

1. The no header switch (/-HD) must be used on the task image output file,
2. The input options STACK=0 and UNITS=0 must be specified.

If the SGA is position independent, the /PI switch can be specified. The /PI switch must be included on FORTRAN block data SGAs.

5.3 INSTALLATION AND REMOVAL OF AN SGA

SGAs are installed into the system and removed from it in a manner similar to task installation. The system MCR functions INS and REM are used. See the RSX-11D User's Guide for a complete description of these functions and their options. Below is a description of the main points of the installation and removal process.

1. At installation, the user specifies that the file being installed is an SGA by using either the /LI switch to indicate that the file is a global library or the /CM switch to indicate that it is a common area. INSTALL determines that the entity is an SGA and assumes that the SGA is a common area unless /LI is specified.
2. The owner UIC is specified at installation by means of the /UIC=[n,n] switch. The owner UIC always should be specified when the SGA is installed.
3. The nonowner protection that is applied to the SGA is specified at installation by means of the /ACC= switch. The switch can be specified in the form /ACC=RO for read-only or

/ACC=RW for read/write. The default is no access; be sure to include the switch if tasks running under a nonowner UIC are to have access to the SGA.

5.3.1 Effect of Installation

Within the system, the main effect of installing an SGA is to create a global common directory (GCD) entry. The GCD entry is a control block that serves to identify the SGA to the system. Refer to the RSX-11D Executive Reference Manual for the format of a GCD entry. One important field in each GCD entry is the installed reference count.

5.3.1.1 Installed Reference Count

- The system uses the installed reference count to keep track of the number of tasks that currently are installed and that refer to the SGA.

NOTE

Do not confuse the installed reference count with the active task access count.

An SGA must be installed prior to installation of the tasks that link to it.

The maximum number of tasks that can refer to one SGA is 255 (decimal). This count is set to zero when the SGA is installed. The count is increased by one whenever a task that links to that SGA is installed.

Similarly, whenever a task that links to the SGA is removed, the count is decreased. Therefore, if an attempt is made to remove an SGA using the REM command to MCR, the installed reference count is checked. The removal is not allowed if the count is nonzero, i.e., if installed tasks refer to that SGA. Before an SGA can be removed, all tasks that link to it must be removed.

5.4 SGA OPERATION AT RUN TIME

Once the SGA is installed in the system and a GCD entry exists, it can be loaded into memory for use by tasks that refer to it. A brief description of the basic mechanism for loading and unloading SGAs follows.

5.4.1 Access Count in the GCD

At run time, the most important field of the GCD entry is the access count. This count is increased whenever a task that links to the SGA is loaded successfully into memory. It is decreased whenever the task exits or is checkpointed to the disk. The access count determines when as SGA is to loaded or unloaded.

5.4.2 Loading Principle for an SGA

When a task that links to an SGA is requested, the Executive prepares to load the task and at the same time checks the GCD entries for the SGA to determine their status. If the SGA is not loaded, the Executive prepares to load it as well.

The actual loading of the task and any of its related SGA is attempted only if memory can be found for all of the components (task and SGAs) of the load. If an SGA already is loaded, no attempt is made to load it again. The access count provides the Executive with this information. When the task and its SGAs have been loaded successfully, the access count for each SGA is increased by one.

5.4.3 Task Exit/Checkpointing and Unloading SGAs

When a task relinquishes its memory by exiting or being checkpointed, the access count for each SGA that it refers to is decreased by one. If the Executive determines that the access count is now zero for an SGA, it unloads the SGA by releasing the memory allocated to it. If the SGA is a common area, i.e., if it is read/write, it is written to disk also, and then memory is released. Read/write SGAs are written back to the disk at the physical block where the file was positioned at install time. The writing back will occur even if the file is deleted and may corrupt any subsequent file on the disk. Therefore, do not delete the task image file until the SGA has been removed.

The result of this approach is that a single SGA remains in memory as long as its access count is greater than zero regardless of how many tasks refer to it, exit, or are checkpointed.

5.5 CHARACTERISTICS OF SGAs

The characteristics of SGAs described in the following paragraphs should be kept in mind when designing the use of SGAs.

1. Every load of an SGA requires another disk access if the SGA is not fixed or already loaded. The result is that a task that refers to two SGAs, neither of which is loaded, requires three disk accesses to activate the task, i.e., one for the task and one for each SGA.

   This process applies to multiuser tasks with pure areas as well. These tasks require two disk accesses to load them: one for the read/write area and one for the read-only (pure) area.

   This does not apply to nonmultiuser tasks or to multiuser tasks with no pure area.

Every request of a task and its related SGAs requires one I/O request node to be picked for every component of the load. For example, if a task has a read-only area and links to an SGA, three I/O request nodes are picked at task request time. If the nodes cannot be picked, the request is rejected with the standard error return.

2. SGAs can map over the external page by use of the pseudo-partition name $$$EXT at INSTALL time; however, an actual load of such an SGA does not occur. The simplest procedure for such an SGA is shown in the following example.

   Assembly code example:

```
        .TITLE   EXTPAG
START:  .BLKW    50      ; IGNORE FIRST 50 WORDS
                         ;   OF EXTERNAL PAGE

XX::    .WORD    0       ; AN INTERESTING REGISTER
YY::    .WORD    0       ; ANOTHER INTERESTING REGISTER
                         ; BOTH GLOBALLY DEFINED

        .END     START
```

   Task build procedure example:

```
TKB [1,1]EXTPAG/-HD,LP:/CR,SY:[1,1]EXTPAG=[200,200]EXTPAG
/
STACK = 0
UNITS = 0
BASE  = 12             ; MAP IT UNDER ASR 3
/
TKB TASK,MAP=TASK
/
COMMON=EXTPAG:RN
/
```

3. The original copy of an SGA installed as a common area is updated when the SGA is unloaded. This characteristic should be considered when designing a system restart procedure.

To retain the original task-built version of a common area, use PIP to make a second version of the file. When the SGA is installed, the file with the higher version number is selected for installation. Because the original version is never installed, it is not modified when the common area is written onto the disk. The modified version is the one with the higher version number.

To reinitialize the common area with its original data, remove the common area from the system, delete the copy of the common area with the higher version number, make another copy of the original data file, and reinstall the common area.

5.5.1 Fragmentation and Lockouts

The system manager must avoid lockout conditions due to the unavailability of memory. The following points must be understood.

1. No attempt is made to load a task until memory has been found for all components (task and SGAs) of the load.

2. The necessary memory need not be contiguous for each component; i.e., the memory for an SGA can be discontiguous from the memory required for the task. Memory is allocated contiguously on a component basis, i.e., for the task or SGA level.

3. Attempts to allocate memory for a task are always made for the highest priority task in the MRL.

4. If a lower priority task fits in an area of memory that is too small for the higher priority task, the lower priority task is not loaded.

5. A task is not considered checkpointable unless it is being loaded or is actually running.

6. No moving of currently running tasks is performed to make room for a task desiring memory.

The following rules should be applied when designing the use of SGAs.

1. SGAs should be installed into their own partition or into the same partition as the tasks that use them.

2. Multiuser tasks with read-only areas should be requested into the partition into which they were installed.

5.6 FIXING AN SGA IN MEMORY

An SGA can be fixed in memory by fixing a dummy task that uses the SGA using the FIX command to MCR.

The dummy task should be built with the following options, which are described in the RSX-11D Task Builder Reference Manual:

```
/-CP /-FP /FX
STACK = 10
UNITS = 0
LIBR = sga name     or     COMMON = sga name
```

The dummy task uses 96 words; this includes the task header and minimum stack size, plus one FTL entry. The above approach is the optimal solution to fixing an SGA.

Fixing the dummy task results in the loading and fixing of the SGA that it uses.

The following is an example of a dummy task.

```
       .MCALL      EXIT$C
START:      EXIT$C   ; EXIT$C is included in
       .END  START   ; case the task is
                     ; ever executed
```

CHAPTER 6

CORE DUMP ANALYZER

The Core Dump Analyzer (CDA) program allows the user to analyze the state of the software at the time of a system crash. The term 'crash' refers throughout this chapter either to an actual system crash or to a crash forced by the operator.

When a crash occurs, and provided the Executive has been built with a crash module appropriate to some dumping device, an area of memory is automatically dumped on that device. When the system is again running, the operator can request CDA to analyze the core dump. The same system should be used as the one that crashed. CDA can be run at the same time as other tasks.

NOTE

In order to benefit fully from CDA output, the user must be familiar with the RSX-11D Executive and with the PDP-11 Processor Handbook.

The output from CDA is a listing. The information listed falls into two categories, standard and optional.

The standard information is:

1. An analysis of the system status at the time the core dump was taken and an indication of the cause of the system crash,

2. The content of all the central processing unit registers and memory management (KT11) registers,

3. The content of the SCOM data area and checkpoint bit map.

The optional information is:

1. An analysis of all tasks active at the time of the crash including the interpretation of their headers,

2. An analysis of all system lists,

3. A verification of the system node pool including a comparison of the nodes allocated with those nodes in use in lists,

4. A complete dump of the system node pool in three output formats (octal, ASCII, and Radix 50),

5. A dump of the address space of any task that has memory allocated at the time of the crash.

6. A dump of a named Shareable Global Area (SGA).

Standard information is always listed. Optional information is listed only if the appropriate option switches are included in the command sequence that calls CDA. Standard information is described in Section 6.1. Optional information is described in Section 6.2.

CDA may be called more than once, if desired to analyze the same core dump.

Appendix A contains a sample CDA listing that is referred to for all examples throughout this chapter.

6.1 STANDARD INFORMATION PRODUCED BY CDA

The standard CDA information is system data and is printed at the beginning of the core dump analysis. See pages 1 through 3 of the sample analysis in Appendix A.

6.1.1 Crash Stack

The first page of output from CDA displays the data recorded on the crash stack at the time the dump was taken. The crash stack refers to the data area that is internal to the crash module. It contains the following information.

1. Cause of the crash -- The specific reason for the system crash, e.g.,

   SST EXECUTED IN KERNEL MODE

   If the crash dump was obtained by some means other than an Executive crash module, i.e., by a stand alone program as described in Section 6.5, the following message is printed.

   MANUAL DUMP BY OPERATOR

2. The processor status register (PS), the kernel stack pointer (SP) and the user SP after the crash -- If the crash was forced by the Executive, these registers reflect the content of the respective registers after the IOT (I/O trap) instruction has been executed.

3. The program counter (PC) and PS before the crash -- If the crash was forced by the Executive, the content of these two registers is pushed onto the kernel stack by the IOT instruction. The PC value is equated to a specific global crash code whenever the .CRASH macro appears in the Executive code.

4. RØ through R5.

5. Segmentation status registers Ø through 3. On a PDP-11/4Ø, not all of the registers are present. Refer to the <u>PDP-11 Processor Handbook</u>.

6. All of the segmentation registers for user, supervisor, and kernel modes for both I and D space. On a PDP-11/4Ø, not all of these registers are present. Refer to the <u>PDP-11 Processor Handbook</u>.

The information contained in the crash stack is the key to the operation of CDA. In particular, the contents of the kernel PARs enable CDA to convert virtual memory addresses into the physical block and word addresses of the dump medium.

6.1.2 Kernel Stack

The kernel stack of 128 words is dumped on the second page of the CDA output. The location of the kernel stack is determined from the symbol .SG.TS in the Executive, which is the top of the Executive stack.

6.1.3 System Communication Area (SCOM)

The system communication area (SCOM) data area is dumped on the third page of the CDA output. The contents of the following SCOM items are interpreted.

1. Time of the crash
2. Current task name and ATL address (.CRTSK)
3. PDP-11/45 floating point hardware indicator (.FP45X)
4. Processor type (.PDP11)
5. Size of memory in 32-word blocks (.MSIZE)
6. Size of the checkpoint file (.CKFSZ)
7. Significant event flags (.SERFG)
8. System event flags (.COMEF)
9. The entire SCOM data area from .SG.BC to .SG.EC

6.2 OPTIONAL OUTPUT FROM CDA

Optional output from CDA is obtained by including the appropriate switches in the command string that calls CDA, as described in Section 6.3.

All switches are presented in alphabetic order by switch name for ease of reference. However, anyone unfamiliar with CDA should read all descriptions before using the following information for reference purposes. Also, refer to the tables and lists described in the RSX-11D Executive Reference Manual.

6.2.1 All Lists (/ALL)

When the /ALL switch is included in the command string, CDA includes the following information in the printed output.

1. AST queues
2. Active task list
3. Batch command buffers
4. Clock queue
5. Fixed task list
6. Global common directory
7. Task headers
8. MCR command buffers
9. System node pool
10. Physical unit directories (PUDs)
11. SEND/RECEIVE queues
12. STD entries
13. Task partition directories
14. Verification of the system node pool

Using the /ALL switch is equivalent to including all CDA command switches except /TAS, /DMP and /SGA. These last three are used to dump specified portions of memory.

6.2.2 AST Queues (/ASQ)

A dump of the AST queues is included if the /ASQ switch is used. See page 46 of the sample listing for an example of AST queue output.

The /ASQ switch causes all tasks which have any entries in their AST queue to have such output. Note that the AST queue also includes I/O requests which have been completed but for which the task has not yet been notified.

For each task with ASTs pending, the task name, ATL address (real and virtual) and STD address are output as a heading.

I/O completion entries are output as:

1. AST entry point, if any
2. Event flag number specified in the QIO
3. I/O status block address
4. I/O status block contents (two words)
5. Intermediate buffer address (real and virtual) (RSX-11D intermediate buffered transfer only)
6. User virtual buffer address (RSX-11D intermediate buffered transfers only).

For true AST entries the output is:

1. AST type
2. AST entry point
3. Additional information, if any, which will be pushed on to the user's stack.

6.2.3 Active Task List (/ATL)

A dump of the active task list is included in CDA output if the /ATL switch is used. See page 4 of the sample listing for an example of ATL output.

The /ATL switch causes all entries in the ATL to be listed and provides interpretation of the following items in the order described.

1. ATL address -- virtual and real address nodes
2. STD address -- virtual and real address nodes
3. REQUESTER -- name of the task that requested this task
4. Load address -- real memory address of the task's header
5. Name of the partition in which the task was running
6. The task's run priority
7. I/O in progress count
8. I/O pending count
9. Status -- the same 2-character codes as those used by ...ACT as described in the RSX-11D User's Guide
10. TI assignment -- the terminal from which the task is being run
11. Flag word -- A.TF
12. Flags -- task's 32 event flags (two words)
13. Masks -- task's 64 event flag masks (four words)
14. Entire ATL node dumped with virtual addresses

6.2.4 Batch Command Buffers (/BAT)

A dump of the batch command buffers is included in CDA output if the /BAT switch is used in the command string. On page 44 of the example, the sample listing indicates that no batch command buffers were in use at the time of the crash.

The format of CDA output for the batch command buffer is the same as the format for the MCR command buffers shown on page 43 of the sample listing.

The /BAT switch causes all the batch command buffers to be dumped and provides interpretation of the following items for each buffer.

1. Node address -- virtual address of the node
2. Name of the MCR task requested
3. TI assignment -- terminal initiating the batch job
4. Length of the buffer
5. The first 40 characters in the buffer converted to ASCII

6.2.5 Clock Queue (/CKQ)

A dump of the clock queue is included in CDA output if the /CKQ switch is used. See page 48 of the sample listing for an example of CKQ output.

The /CKQ switch causes all entries in the clock queue to be dumped. The format of the output depends on the type of the clock queue entry. For all types, the first line indicates the type and the time in ticks which must elapse after the previous entry before this entry will be dequeued (the ´Delta´ time).

The types are:

a. Time-slice, for the RSX-11D time-slicing scheduler. No further information is output.

b. Mark time. The following is supplied:

   1. Task name

   2. ATL address - virtual and real node address.

   3. Event flag number

   4. AST entry point address

c. Schedule, that is, a request for a task to run at a specified tiime. The additional information is:

   1. Name of requested task

   2. STD address of requested task - real and virtual addresses.

   3. Task name of requestor.

   4. STD address of requestor - real and virtual addresses.

   5. Partition specified for task to run in, or blank if no task specified.

   6. Priority specified for task, or zero if no priority specified.

   7. UIC specified for task to run under, or [Ø,Ø] if no UIC specified.

   8. TI for which the task will run.

d. Reschedule, that is, a schedule which is to be repeated at a regular interval. The additional information is as for ´schedule´ in c. above, with the addition of the interval between periodic reschedules, in ticks.

6.2.6 Dump of Real Memory (/DMP)

The /DMP switch causes a portion of real memory to be dumped in three notations: octal, ASCII, and radix 5Ø. The /DMP switch has the following syntax.

/DMP:start:end

start — is the starting real memory address. An octal number representing up to 22 bits is used.

end — is the ending real memory address. An octal number representing up to 22 bits is used.

Both the starting and end addresses must be specified. Only one /DMP switch can be specified.

6.2.7 Fixed Task List (/FTL)

A dump of the fixed task list is included in CDA output if the /FTL switch is used. See Page 45 of the sample listing for an example of FTL output.

The /FTL switch causes all entries in the FTL to be listed and provides interpretation of the following items in the order described.

1. FTL address -- virtual and real address nodes
2. STD address -- virtual and real address nodes
3. Load address -- real memory address of the task's header
4. Name of the partition in which the task was running
5. The task's run priority
6. I/O in progress count
7. I/O pending count
8. Status -- the same 2-character codes as those used by ...ACT as described in the RSX-11D User's Guide
9. TI assignment -- the terminal from which the task is being run
1Ø. Flag word -- A.TF
11. Flags -- task's 32 event flags (two words)
12. Masks -- task's 64 event flag masks (four words)
13. Entire FTL node dumped with virtual addresses

6.2.8 Global Common Directory (/GCD)

A dump of the global common directory is included in CDA output if the /GCD switch is included in the command string. See pages 49 and 50 of the sample listing for an example of GCD output.

In addition to the entries in the GCD itself, entries linked to STDs for the pure area of a multiuser task also are dumped. The formats of both types of nodes are identical except that pure area GCD nodes do not contain the name of the task. Instead, an octal 35 (an illegal radix 50 character) is in the first word of the name. The following items are interpreted for each GCD when the /GCD switch is specified.

1. Name of the common block or library, or octal 35 if the node is for the pure area of a task

2. Virtual address of the GCD node

3. Size -- 1/64th of the size of the common block, library, or pure area

4. Name of the partition in which the shared area is installed

5. Flags byte

6. Status of the global area using the same 2-character codes as ...ACT with the addition of NL to indicate not in use or not loaded

7. Active reference count and installed reference count separated by a slash (/)

8. Creation time

9. UIC

6.2.9 Task Headers (/HDR)

A dump of the task headers is included in CDA output if the /HDR switch is included in the command string. See pages 8 and 9 for an example of task header output.

Interpretation of the following items is provided in addition to the dump of the headers.

1. Header address -- real address of the task's header
2. ATL address for the task
3. H.CR1 -- context buffer reference point 1
4. User PDRs
5. User PARs
6. PS, PC, RØ through R5, and SP
7. Initial PS, PC and SP
8. Size of the header in 32-word blocks
9. Directive status word (DSW) -- task's virtual zero
10. Number of LUNs assigned to the task
11. Logical unit table (LUT) -- the device and unit address for each entry in the LUT. If there is a window open for the entry, the following additional items are dumped.

    From the window: W.CTL, W.VBN, W.FCB

    From the FDB: F.FNUM, F.FSEQ, F.STAT, F.NACS, F.NLCK
12. The entire header is dumped.

6.2.1Ø MCR Command Buffers (/MCR)

A dump of the MCR command buffer is included in CDA output if the /MCR switch is used in the command string. See page 43 of the sample listing.

The /MCR switch causes all MCR command buffers to be dumped and provides interpretation of the following items for each buffer.

1. Node address -- virtual address of the node
2. Name of the MCR task requested
3. TI assignment -- terminal initiating the batch job
4. Length of the buffer
5. The first 4Ø characters in the buffer converted to ASCII

6.2.11 Dump of System Node Pool (/POL)

Inclusion of the /POL switch causes the entire system node pool to be dumped with virtual and real addresses (see pages 54 and 55 of the sample listing). Each node (8-word block) is dumped in three formats: octal, ASCII, and radix 5Ø. In addition if the node is in use (allocated), an asterisk (*) is printed at the lefthand margin.

6.2.12 PUDs and I/O Request Queues (/PUD)

Inclusion of the /PUD switch causes all peripheral unit directories (PUDs) to be dumped. If any I/O requests are queued for the device type, the I/O request nodes are dumped after each associated PUD. See pages 31 and 32 of the sample listing.

The following items are interpreted for each PUD entry.

1. Device name and unit number
2. Flag byte
3. Name of the task attaching the device, if any
4. Device to which this one has been redirected
5. Handler task name
6. Count of express requests in the queue (EXP CNT)
7. U.CH -- characteristics word in PUD (CHAR)
8. First three characters of the default ACP task name in radix 5Ø (DACP)
8. Name of the ACP task
9. Terminal flag bytes (TERM)
11. A dump of the entire PUD

The following items are interpreted for the I/O request node.

1. Name of the task generating the I/O request
2. ATL node address of the requester
3. Size of the DPB (DPB SZ)
4. Priority of the request
5. LUN
6. Event flag number (EFN)
7. Function code -- if the function code is one of the more common function codes, it is printed symbolically, e.g., IO.WLB. If it is not a common code, IO.XXX is printed.
8. Request type -- either normal or express
9. A dump of the entire I/O request node

6.2.13 Dump of an SGA (/SGA)

The /SGA switch is used to dump a shareable global area (library or common area). The SGA must have been resident in memory.

The /SGA switch has the following syntax, with brackets indicating the optional portion:

/SGA=sganamel[:start:end]

where:

sganame — is the name of the SGA to be dumped

start — is the starting virtual address in octal of the portion of the SGA to be dumped.

end — is the ending virtual address in octal of the portion of the SGA to be dumped.

Default: if the start and end addresses are omitted the entire SGA is dumped.

For a position-independent SGA, addresses are assumed to start at 0, both in the dump and in reckoning the ´start´ and ´end´ values. For a non-position-independent SGA, addresses correspond to those used when the task was built, e.g. an SGA mapped through APR3 would start at 60000.

This switch may only appear once in a CDA command string. See pages 102 and 103 of the sample listing for an example of the output obtained by using the /SGA switch.

6.2.14 SEND/RECEIVE Queues (/SRQ)

Inclusion of the /SRQ switch (see page 47 of the sample listing) causes all the SEND/RECEIVE queues to be listed and provides interpretation of the following items for each queue.

1. Task name of the receiver
2. Task name of the sender
3. TI -- terminal of receiver or sender task
4. Priority of the send
5. Data block contents

6.2.15 System Task Directory Entries (/STD)

Inclusion of the /STD switch causes all the entries in the system task directory (STD) to be dumped. See pages 22 and 23 of the sample listing. When /STD is used, CDA provides interpretation of the following items for each STD entry.

1. Name of the task
2. Name of the partition in which the task is installed (PAR)
3. STD flag word
4. Priority at which the task is installed (PRI)
5. Disk from which the task is installed (DISK)
6. Size of the task (SIZE) -- S.TZ
7. Number of active versions of the task (AV)
8. Pool limit specified in 8-word blocks (LIM)
9. Pool utilization specified in 8-word blocks (UTIL)
10. SRQ listhead

6.2.16 Dump of Task's Address Space (/TAS)

Inclusion of the /TAS switch causes the address space of a task that has memory allocated to it at the time of the crash to be dumped (see pages 100 and 101 of the sample listing). The task must have been either active or fixed in memory. Each task area is dumped in three notations: octal, ASCII, and radix 50.

The /TAS switch has the following syntax with brackets indicating the optional portion.

/TAS=taskname[:start:end]

taskname	is the name of the task to be dumped
start	is the starting virtual address in octal for the portion of the task to be dumped
end	is the ending virtual address in octal of the portion of the task to be dumped.

If the starting and ending addresses are omitted, the entire task is dumped. When this switch is used to dump a multiuser task, only its impure area is dumped.

NOTE

Only tasks with entries in the STD can be dumped using the /TAS switch. Pure areas of multi-user tasks cannot be dumped.

6.2.17 Task Partition Directory (/TPD)

Inclusion of the /TPD switch causes all entries in the task partition directory to be dumped. Also dumped are any entries in the memory required list (MRL) or the checkpointable task list (CTL) for each partition. An allocation map is printed for each partition showing the unused area, active and fixed tasks, and shared areas in the partition. See pages 19 and 20 of the sample listing.

The following items are interpreted for each partition.

1. Partition name
2. TPD entry address -- virtual and real address of TPD entry
3. Base address -- real starting address of the partition
4. Size of the partition in bytes
5. TPD flag word
6. Unused areas (HOLES) -- if the partition is system-controlled, the address and size of each unused area is printed.
7. Allocation map -- all active or fixed tasks and shared areas in the partition are listed with the real address of the space allocated. One of the following letters is used to indicate the type of task or area.

   A -- active task
   F -- fixed task
   P -- pure area of multiuser task
   L -- library
   C -- common block

If the partition has any entries in the memory required list (MRL), the following items are interpreted.

1. Virtual address of the MRL node (NODE ADR)
2. Name of the task
3. Size of the task (SIZE) -- S.TZ
4. Run priority of the task
5. Checkpoint priority (CP PRI)
6. Status (STS)
7. Checkpoint status (CP STS)
8. I/O node address
9. TI

If the partition has any entries in the CTL, the following items are interpreted.

1. CTL node address
2. Name of the task
3. Run priority
4. Status (STS)
5. TI

6.2.18 Verification of System Node Pool (/VFY)

Inclusion of the /VFY switch causes the system node pool to be verified (see pages A-16 and A-17 of the sample listing). The following procedures are used to verify the node pool.

1. The number of nodes (8-word blocks) in use is determined.

2. The total number of nodes in the pool also is calculated using the following formula.

   Total number of nodes = (Mod32(.STDTA+(.STDTC*2))-.PUDBA/8

   .STDTA is the starting address of the STD alpha table. .STDTC is the capacity of the STD. .PUDBA is the beginning address of the PUD.

   This formula provides the total number of nodes available in the pool, i.e., those that are not allocated permanently to fixed lists such as the PUD, TPD, and STD alpha table.

3. All the dynamic system lists are scanned to find the nodes that have been picked from the pool. The lists that are searched include the following.

   Fixed task list
   Batch command buffer
   I/O request nodes
   Volume control block
   Vector connection (..CINT) nodes
   AST nodes
   Window nodes
   AST nodes linked to task headers
   Active task list
   CKQ (clock queue)
   GCD nodes
   I/O nodes linked to GCD nodes
   MCR command buffer
   Memory required list
   I/O nodes linked to MRL nodes
   AST nodes linked to MRL nodes
   System task directory
   SEND/RECEIVE Queue
   Pure area GCD nodes

   As a result of this search, a bit map is constructed to show which nodes are in use in system lists. Although every list is searched, it is still possible to miss some nodes because any privileged task can pick nodes. For example, the teletype handler uses nodes for read-ahead buffers. Nodes in use for privileged tasks are not found when the system lists are searched.

4. The bit map created in step 3 above is scanned and the number of nodes marked as being in use is printed.

5. The pool is tested for discrepancies. First, the nodes actually allocated, but not found in any list, are dumped. There always are some of these.

   Next, the nodes found in the lists, but not allocated, are dumped. If there are any of these, they should be examined carefully. They indicate that the current pool allocation does not reflect the status of the nodes in the pool accurately.

6. Finally, the bitmaps representing the nodes allocated and the nodes found in the lists are dumped.

6.3 OPERATIONAL INFORMATION

The operational information required for CDA includes the process of linking a crash module into the Executive (Section 6.3.1) and the command string used to call CDA (Sections 6.3.2 and 6.3.3).

6.3.1 Linking a Crash Module into the Executive

In order for a crash module to write the dump that provides CDA with input, a crash module must be linked into the Executive.

The command file [11,15]TKB15.CMD is used as input to the task builder for this purpose. Appendix B provides a listing of the command file.

To link the crash module into the Executive, replace NCRASH.OBJ with one of the following crash modules:

TC11CM or TC11CR to dump onto DECtape,

TU10CM or TU10CR to dump onto TU10 magnetic tape,

TU16CM or TU16CR to dump onto TU16 magnetic tape,

RK05CM or RK05CR to dump onto RK05 disk.

The modules with names ending in the letters CM are the same as those with names ending in CR except that the CM versions cause more detailed messages to be printed on the console at the time of a crash.

If none of the crash modules is needed, but the more detailed console messages are desired, replace NCRASH.OBJ with the module MCRASH.OBJ.

Refer to the listing in Appendix B to determine default unit numbers for the crash modules and how to modify them.

Rebuild the Executive using the TKB15.CMD file and rebuild the bootstraps using [11,17]BOOTSBLD.CMD. Then rebuild CDA using [11,13]CDABLD.CMD.

6.3.2 Calling CDA

Core dump analyzer is a task that can execute online with other tasks. Type the following command to MCR to install CDA.

```
MCR>INS [11,1]CDA
```

The following command syntax is typed to MCR to request CDA.

```
MCR>CDA [option switches]
```

option switches — are those switches described previously in Section 6.2 and summarized in Table 6-1 below. Any combination of switches can be used, or all switches can be omitted.

Table 6-1
Summary of CDA Option Switches

SWITCH	FUNCTION
/ALL	Lists the same information as produced by using all CDA option switches except /TAS, /SGA and /DMP.
/ASQ	Lists and interprets ASTs and I/O requests completed but not yet notified to the task.
/ATL	Interprets and dumps ATL entries.
/BAT	Interprets and dumps batch command buffers.
/CKQ	Interprets and dumps clock queue entries.
/DMP:start:end	Causes a specified portion of memory to be dumped.
/FTL	Interprets and dumps the fixed task list.
/GCD	Interprets and dumps GCD entries for libraries, global common areas, and multi-user task pure areas.
/HDR	Interprets and dumps individual task headers.
/MCR	Interprets and dumps MCR command buffers.
/POL	Dumps the node pool in octal, ASCII, and Radix 50.
/PUD	Interprets and dumps PUD information.

(Continued on next page)

Table 6-1 (Cont.)
Summary of CDA Option Switches

SWITCH	FUNCTION
/SGA=sganame[:start:end]	Dumps all or part of a named SGA
/SRQ	Interprets and dumps SEND/RECEIVE queues.
/STD	Interprets and dumps STD entries.
/TAS=taskname[:start:end]	Dumps all or part of the task´s address space for a task that has memory allocated to it.
/TPD	Interprets and dumps all entries in the task partition directory.
/VFY	Causes verification of the system node pool.

NOTE

The CDA input LUN is assigned to DTØ. To change this assignment, use the REA command as in the following example before calling CDA.

```
MCR>REA ...CDA 3 DK1:
```

6.3.3 Examples of CDA Command Strings

The following are examples of CDA command strings:

Example 1:

```
MCR>CDA
```

Result: Only the system data is printed.

Example 2:

```
MCR>CDA /ALL
```

Result: All the optional output is printed.

Example 3:

```
MCR>CDA /POL
```

Result: The system node pool is dumped.

Example 4:

```
MCR>CDA /ATL/HDR
```

Result: The ATL nodes and task headers are dumped for those tasks that were active when the system crashed.

Example 5:

MCR>CDA /PUD/TAS=TT....

Result: All the PUD´s are dumped followed by a dump of the entire task "TT....".

Example 6:

MCR>CDA /TASK=XYZ:0:2000

Result: If the task XYZ had memory allocated at the time of the crash, its virtual address space from 0 to 2000 is dumped. If no memory was allocated for XYZ a message is printed instead.

Example 7:

MCR>CDA /DMP:4400000:4430000

Result: Real memory from 4400000 to 4430000 is dumped.

6.4 OPERATOR ERROR MESSAGES

The messages listed in Table 6-2 are reported on the console requesting CDA when an error is detected that prevents CDA performing the analysis. 'Input device' is the device on which the memory dump is held and which provides the input to CDA.

Table 6-2
CDA Error Messages

Message	Meaning	Suggested Action
CDA -- DEVICE READ ERROR	Input device is not ready or is otherwise unavailable.	Check status of device. Call field service if problem persists.
CDA -- PLEASE MOUNT INPUT DEVICE AS FOREIGN	Input device not mounted as a foreign volume.	Mount the input device as foreign.
CDA -- DEVICE HANDLER MISSING	The device handler for the input device is not loaded.	Load the handler, then retype the command string.
CDA -- FAILED TO READ MCR COMMAND BUFFER	CDA was not invoked as an MCR function.	Execute CDA as an MCR function, not with RUN.
CDA -- SYNTAX ERROR	There is a mistake in the command string.	Correct the error and retype the command.
CDA -- ILLEGAL SWITCH	An unrecognized switch was found in the command string.	Type in the command string again using the correct switch.
CDA -- DUMP ABORTED - KERNEL PAR'S CLOBBERED	Either the crash clobbered the PAR's or the format of the crash stack on the dump medium is incorrect.	Try to obtain another crash.

6.5 INPUT TO CDA

The usual source for input to CDA is the dump produced by one of the RSX-11D Executive crash modules that can be linked in when the Executive is built (see Section 6.3). If the dump has been made by one of these modules, it is possible for CDA to report the cause of the crash.

6.5.1 Forcing a Crash

It is also possible that the user would like to crash the system in order to study the dump produced.

The executive can crash for any of the following reasons:

1. Odd Address or TRAP 4
2. Undefined System Event
3. AST Node Pick Failure
4. Floating Point Error
5. Memory Parity Error
6. Non-RSX-EMT execution
7. Reserved Instruction
8. SST in Kernel Mode
9. Red or Yellow Stack
10. EMT 377 in Kernel Mode
11. Undefined System Error.

A crash is forced by generating one of the above conditions. The two easiest methods are 1 and 9.

An ´odd address or trap 4´ can be generated by halting the processor, setting the Program Counter (PC) value odd, and continuing.

A ´Red or Yellow Stack´ can be generated by halting the processor, depositing zero into the Kernel Stack Pointer, and continuing.

The Console Switches are used to halt, set an appropriate value and continue. See the section on the Operator´s Console in the Processor Handbook for the model involved.

NOTE

Causing an active, running system to crash may result in serious consequences. It always causes a complete loss of all data in memory. It can cause corruption of disk file structures. Use extreme caution, e.g., write protect disks first.

6.5.2 Dumps from Stand-alone Programs

Occasionally, none of these methods causes a crash; e.g., the system is dead or in a very tight loop. For this reason, various techniques have been developed for dumping a system without using the crash modules.

All of these techniques involve bootstrapping a stand-alone program into memory. The most common technique is to bootstrap a program from DECtape. This program proceeds to write the dump on the DECtape.

CDA can interpret the dumps from stand-alone programs if the following conditions are met.

1. The memory dump begins at block 101 (octal) for disks or DECtape or at physical record 2 for magnetic tape. The memory dump must begin at real memory location 0.

2. Block 100 (octal) or record 1 must be formatted in a specific way and must include at least the kernel PARs in particular locations. For the precise format of this information, refer to Figure 6-1.

NOTE

If the dump was not made by an Executive crash module, the first 2000 (octal) bytes of real memory may be lost because the hardware bootstraps for some devices, notably DECtape, use the first 2000 bytes. In certain cases, losing these 2000 bytes can be critical, especially if the kernel stack extends into that area.

```
THE INTERNAL STACK (AT 'ENDST') CONTAINS THE FOLLOWING INFORMATION:

      (SP+340 IS AT LOCATION 'ENDST')

      SP+334 -- PROGRAM STATUS WORD AFTER BOOT
      SP+332 -- KERNEL SP AFTER BOOT
      SP+330 -- PC BEFORE CRASH IOT (FROM KERNEL STACK)
      SP+326 -- PS BEFORE CRASH IOT (FROM KERNEL STACK)
      SP+324 -- USER SP
      SP+322 -- R0 (REGISTER SET 1)
      SP+320 -- R1
      SP+316 -- R2
      SP+314 -- R3
      SP+312 -- R4
      SP+310 -- R5
      SP+306 -- SEGMENTATION STATUS REGISTER 0
      SP+304 -- SR1
      SP+302 -- SR2
      SP+300 -- SR3
      SP+276 -- USER I SPACE DESCRIPTOR REGISTER 0
      .
      .
      .
      SP+260 -- USER I SPACE DESCRIPTOR REGISTER 7
      SP+256 -- USER D SPACE DESCRIPTOR REGISTER 0
      .
      .
      .
      SP+240 -- USER D SPACE DESCRIPTOR REGISTER 7
      SP+236 -- USER I SPACE ADDRESS REGISTER 0
      .
      .
      .
      SP+220 -- USER I SPACE ADDRESS REGISTER 7
      SP+162 -- USER D SPACE ADDRESS REGISTER 6
      .
      .
      .
      SP+200 -- USER D SPACE ADDRESS REGISTER 7
      SP+176 -- SUPERVISOR I SPACE DESCRIPTOR REGISTER 0
      .
      .
      .
      SP+160 -- SUPERVISOR I SPACE DESCRIPTOR REGISTER 7
      SP+156 -- SUPERVISOR D SPACE DESCRIPTOR REGISTER 0
      .
      .
      .
      SP+140 -- SUPERVISOR D SPACE DESCRIPTOR REGISTER 7
      SP+136 -- SUPERVISOR I SPACE ADDRESS REGISTER 0
      .
      .
      .
      SP+120 -- SUPERVISOR I SPACE ADDRESS REGISTER 7
      SP+116 -- SUPERVISOR D SPACE ADDRESS REGISTER 0
      .
      .
      .
      SP+100 -- SUPERVISOR D SPACE ADDRESS REGISTER 7
      SP+076 -- KERNEL I SPACE DESCRIPTOR REGISTER 0
      .
      .
      .
      SP+060 -- KERNEL I SPACE DESCRIPTOR REGISTER 7
      SP+056 -- KERNEL D SPACE DESCRIPTOR REGISTER 0
      .
      .
      .
      SP+040 -- KERNEL D SPACE DESCRIPTOR REGISTER 7
      SP+036 -- KERNEL I SPACE ADDRESS REGISTER 0
      .
      .
      .
      SP+020 -- KERNEL I SPACE ADDRESS REGISTER 7
      SP+016 -- KERNEL D SPACE ADDRESS REGISTER 0
      .
      .
      .
      SP+000 -- KERNEL D SPACE ADDRESS REGISTER 7

 *****  NOTE:  THE D-SPACE AND SUPERVISOR REGISTER LOCATIONS
               WILL CONTAIN RANDOM GARBAGE WHEN THE DUMP IS
               MADE ON AN 11/40.
```

Figure 6-1 Format of CDA Input (Record 1 or Block 100)

CHAPTER 7

TIME-SCHEDULED PARTITIONS

RSX-11D supports a time-based scheduling algorithm as an aid to program development. The algorithm performs round-robin scheduling on all tasks of which the following are true:

1. The task runs in a time-slice controlled partition, as defined at System Generation.

2. The task, when installed, has a priority lying between limits defined in the system.

The running of time slicing tasks takes place within the priority structure of RSX-11D. Tasks satisfying the above conditions will come under time slicing whether or not they are checkpointable (liable to rollout/rollin).

7.1 TIME-SCHEDULING OF TASKS

Round-robin scheduling has two time parameters:

1. The CPU Burst, which is the length of time (number of clock ticks) that must pass before the current user of the CPU is changed.

2. The Memory Swap Slice, which is the length of time (number of CPU bursts) that must pass before a resident checkpointable task is rolled out of memory, if the MRL is not empty.

Other parameters are those associated with task priority.

The mnemonics for these parameters and the values supplied in the released system are described in Table 7-1.

A time sliceable task is a task which has been installed into a time-slice controlled partition at a priority of between 3 and PR.MAX.

When such a task is loaded the task's priority is set to the default value PR.DFL associated with time slicing. Normal RSX-11D processing then resumes. No time sliceable task will execute unless all tasks, whose installed priority is greater than PR.MAX, are idle. When the last time sliceable task exits, time slicing is turned off.

When a CPU burst has been completed, the time scheduler is again activated. The Active Task List (ATL) is searched for the first task whose priority is equal to the default. This task is deleted from the ATL, and from the Checkpointable Task List (CTL) if it is checkpointable, and then reinserted into these lists as appropriate. This has the effect of 'lowering' the task's priority in the sense that all other equal priority tasks now precede it in the ATL. Another CPU burst is initiated and the scheduler becomes inactive.

By contrast, when a memory swap slice has been completed the priority of each task, whose current MRL priority is between 3 and PR.MAX, is incremented by 1. In addition, the status of any task on the MRL whose priority is 2 is separately checked. The status of such a task can imply that it is now runnable, for example because of one of the following:

1. an awaited event flag has been set

2. the task has been resumed from a suspended state

3. its status has been changed to 'RUN'

4. an AST has been declared for it

5. the task has been aborted

6. an intermediately buffered I/O request has been terminated.

If such a task with priority 2 is now runnable, its priority is raised either to the maximum time-slice value of PR.MAX or to its installed priority, whichever is the greater.

At this point an attempt is made to activate the highest priority task on the MRL, firstly by a normal MRL scan and secondly by trying to force checkpointing for lower priority tasks. The first candidate for swapping out is the task now at the bottom of the list of those sharing the default priority.

These attempts to activate the highest priority task on the MRL can still fail to provide the necessary memory. Normally the next step is to begin another memory swap slice. But if a task with priority PR.MAX remains at the top of the MRL for 3 memory swap times, then this task is 'demoted'. Demotion is accomplished by subtracting PR.DEC from the task priority and reinserting the task, by priority, in the MRL. Demotion can be inhibited by setting PR.DEC=0.

7.2 PRIORITY OF TASKS ON THE MRL

When a task is put on the MRL, either as a new request or as a roll-out, it has one of two priorities. These are:

1. Its installed priority

2. A priority value of 2. A task is given this priority whenever it is rolled out awaiting some external event which blocks additional execution. In this case the task is inhibited from contending for memory until the external event becomes due (see previous section).

A task with a installed priority greater than the maximum time-slice priority will always be rolled in before any task whose priority is less than or equal to this maximum. This is because, during time slicing, MRL priority will never be incremented beyond this maximum.

If, for example, PIP is installed into a time sliced partition with a priority of 55 and TKB is installed into a time sliced partition with a priority of 50, then as long as both PIP and TKB are on the MRL, PIP will be rolled in first. However, once PIP is rolled in, the MRL priority of TKB will continue to be incremented until, at some subsequent rollout of PIP, PIP will appear in the MRL after TKB, whose priority by now has been incremented to 55 or more.

Rollout is not initiated until there is a task on the MRL whose priority is greater than or equal to the default time-slice priority. This ensures that low-priority 'back-ground tasks' do not get rolled in as often as others. As an example, if task X has been installed at priority 30, then since the default priority is 50, 20 memory swap slices occur before task X is even considered as a candidate for memory.

A task will not be rolled out if its executing priority is greater than the MRL priority of the first (highest priority) task on the MRL. If there is not room in memory for the highest priority task on the MRL, no task of lower priority will be rolled in until the demotion algorithm, described above, takes effect.

7.3 TIME SLICING PARAMETERS

The following table summarizes the parameters involved in 7.1 and 7.2 above. The period (.) indicates values in decimal. B indicates a byte value, W a word.

Table 7-1
Time Slicing Parameters

Parameter	V6.2 System Value	Meaning
PR.DFL	50.	The execution priority of any task whose installed priority is between 3 and PR.MAX inclusive. (B)
PR.MAX	75.	The maximum priority for CPU time slicing. If a task is installed with a priority greater than PR.MAX it will always execute at its installed priority. Also, if the task is on the MRL, it will always take precedence over all time sliceable tasks. (B)
SL.BST	2.	The CPU burst which is the number of clock ticks before putting another equal priority task at the top of the 'equal priority ATL subset'. (W)
SL.SWP	15.	The memory swap slice which is the number of SL.BSTs before swapping is initiated. (W)
SL.BLK	3.	The number of successive SL.SWPs before a task is demoted from the top of the MRL. (W)
PR.DEC	0.	The amount by which a task's priority is decremented when the time slicer decides to 'demote' the task. (W)

7.4 LIMITATIONS OF TIME SLICING

RSX-11D Time slicing is designed to cover program development - editing, assembling, linking - by one or more interactive users. In this way better service can be provided than by the standard priority scheduling of RSX-11D. It is not designed as a full timesharing system.

CHAPTER 8

ERROR LOGGING

In RSX-11D the logging of hardware is performed for both memory errors and device errors. The Executive log memory errors, which can be either main memory parity errors or cache memory errors on the PDP-11/70. Device handler tasks developed for RSX-11D log device errors.

A single error log contains items logged by both the Executive and the device handler tasks. The system manager can use the output from the error logging analysis task to determine the reliability of memory and devices used in the RSX-11D system.

Error statistics are accumulated by handlers for the following devices:

- Disks,
- DECtape,
- Magnetic tape.

For further information on the handling of parity errors, refer to Chapter 11 of this manual.

The report produced by error logging can contain itemized error statistics with optional summary information, or it can contain only the summary information. Additionally, the system manager can select the time frame that the report is to encompass and can indicate that the report is to include only memory parity errors or device errors. Further, if a report on device errors is desired, the device type, unit, or volume for which the report is desired can be specified.

Figure 8-1 provides a sample report for a device error. Figure 8-2 provides a sample of summary information. All numbers followed by a period (.) are decimal values. All others are octal. Appendix C contains sample reports for various types of errors.

```
RSX 11 SYSTEM ERROR REPORT COMPILED AT 20-AUG-75 13:46:42        PAGE 1.


*************************************************************
ERROR LOGGING SYSTEM STARTED
AT 12-AUG-75 10:13:45
*************************************************************


*************************************************************
TAPE HARDWARE ERROR
LOGGED AT 12-AUG-75 10:14:54            ERROR NUMBER 1.
*************************************************************

TAPE PARAMETERS
        UNIT NAME                   DT1
        VOLUME LABEL
        VOLUME OWNER UIC            1,1
        DEVICE TYPE                 TU56 UNIT-1 CONTROLLER-0

TAPE REGISTERS AT ERROR TIME
        TCST                004037  SELECTION ERROR
        TCCM                100702
        TCWC                177400
        TCBA                113534
        TCDT                100770

ERROR DIAGNOSIS
RECOVERED
SELECTION ERROR

RETRIES PERFORMED       0.

USER TASK PARAMETERS
        TASK NAME                   F11ACP
        TASK UIC                    2,2
        PHYSICAL START ADDRESS      102200
        USER FUNCTION REQUESTED             READ  (1000)
        FUNCTION INTERPRETED FROM REGISTERS         READ BLOCK NUMBER FORWARD
                PHYSICAL BUFFER ADDRESS START       113534
                TRANSFER SIZE IN BYTES              1000
                LOGICAL BLOCK NUMBER AT IO GO       13
                COUNT OF IO IN PROGRESS             1.
                COUNT OF IO PENDING                 1.

NUMBER OF FUNCTIONS ISSUED TO THIS UNIT             7

VECTORS WITH ACTIVE IO
```

Figure 8-1 Device-Specific Error Log

```
SYSTEM ERROR REPORT SUMMARY  (SYSTEM)

COMMAND LINE USED
LP:=/BREAKDOWN:ALLALL/DETAIL

REPORT FILE ENVIRONMENT
        SOURCE FILE                SY0:[1,6]ERROR.SYS
        OUTPUT FILE                LP:[200,200]ALLALL.LST
        DATE OF FIRST ENTRY        12-AUG-75 10:13:45
        DATE OF LAST ENTRY         12-AUG-75 10:21:01
ENTRIES PROCESSED                            19.
ENTRIES MISSING                               4.
UNKNOWN ENTRIES ENCOUNTERED                   0.
UNKNOWN DEVICES ENCOUNTEED                    0.
FIELD FORMAT ERRORS ENCOUNTERED               0.
DEVICE ERRORS PROCESSED                      14.
UNDEFINED INTERRUPTS PROCESSED                0.
UNDF. INTR. MISSED DUE TO UNDF. INTR. PROCESSING     0.
TRAPS THROUGH LOCATION 0 PROCESSED            0.
DEVICE TIMEOUTS PROCESSED                     0.
MEMORY MANAGEMENT ERRORS ENCOUNTERED
SYSTEM POWER FAILS ENCOUNTERED
SYSTEM PARITY ERRORS ENCOUNTERED              1.
        REPRODUCIBLE
        NON REPRODUCIBLE
SYSTEM LOADS                                  2.

SYSTEM ERROR REPORT SUMMARY  (TAPE)

TU56 UNIT-1 CONTROLLER-0
HARD        0.
SOFT       14.
```

Figure 8-2 Summary Error Log

8.1 FUNCTIONAL DESCRIPTION

Error logging consists of two distinct functions. The first function is the gathering of information pertinent to the errors that occur and the second is error analysis and the creating of a list file. These functions are performed by three tasks: ERRLOG, PSE, and SYE.

ERRLOG gathers volatile information when a device error occurs. It places this information in a temporary file named ERR.TMP under UFD [1,6] on the system device or a user-specified device. To specify a device other than SY, type the following command.

MCR>REA ERRLOG 4 dev: dev: is the device on which ERR.TMP is to be placed.

When a report of errors is desired, the preanalyzer (PSE) and the analyzer (SYE) tasks are run. When PSE starts, it attempts to open the ERROR.TMP file that has the highest version number. If PSE cannot locate an ERROR.TMP file, it requests that ERRLOG close its current log file (ERR.TMP) and rename that file to ERRLOG.TMP.

When the next error occurs, ERRLOG creates a new ERR.TMP file and continues logging errors. The preanalyzer uses the information in ERROR.TMP to produce a formatted file. The default filename and UFD are ERROR.SYS and [1,6], respectively. However, any operator-specified name and UFD can be used. When the analyzer is run, it uses that formatted file to produce a list file as illustrated in Figures 8-1 and 8-2.

ERROR.SYS remains on disk until the system manager deletes it because it contains information that can be reprocessed by the analyzer or processed by user-written tasks to provide a report with different content. Appendix C provides the layout of records in ERROR.SYS.

8.2 OPERATIONAL INFORMATION

This section provides operating procedures for the three error logging and analysis tasks (ERRLOG, PSE, and SYE) and for the task that terminates error logging (ERROFF).

8.2.1 Running ERRLOG

ERRLOG must be running in order for error statistics to be accumulated and for the raw data file to be passed to the preanalyzer (PSE), if an ERROR.TMP file is not ready for PSE to process. Normally, ERRLOG is installed during system generation. To run ERRLOG, type the following command to MCR and press ALTMODE.

```
MCR>RUN ERRLOG
```

The task responds with the following message.

```
INPUT MINIMUM NUMBER OF ERRORS CAPABLE OF BEING
LOGGED IN A 5 SECOND PERIOD "CARRIAGE RETURN."
THIS VALUE SHOULD NOT EXCEED 5.
IF ERROR LOGGING NOT WANTED INPUT "CONTROL Z."
5 SECOND ERROR RATE =
```

At this point, type a value in the range from 1 through 5 and press RETURN. The value indicates the number of 72-word nodes to be assigned permanently to the error log task. The number of nodes allocated determines the minimum number of errors that can be logged by ERRLOG without a disk write.

Errors can occur more rapidly than they can be handled by ERRLOG. In this case, the summary report contains an entry under the heading NUMBER OF ERRORS MISSED. This entry indicates the number of errors that were not logged due to insufficient node space. Because the device-specific reports provide sequential numbers for errors, the user can determine at which point errors occurred but were not logged.

If a large amount of node space is allocated, it may adversely affect the ability of other tasks to acquire enough dynamic memory to run. The number of nodes required is determined by the system manager and varies from installation to installation.

8.2.1.1 Abnormal Termination of ERRLOG - The ERRLOG task terminates automatically in three cases:

1. When the desired number of nodes cannot be obtained,

2. If the error logging device used by ERRLOG becomes full,

3. If an error occurs when writing to the logging device.

NOTE

For procedures to terminate error logging normally, see Section 8.3.

The ERRLOG task terminates at task startup when the task cannot obtain the number of nodes specified by the user. When this situation occurs, the following message is printed on the console.

```
"ERRLOG" TASK FAILED TO PICK LARGE ENOUGH ERROR LOG
NODE BUFFER. "ERRLOG" TASK TERMINATED. IF YOU WANT
TO TRY AGAIN EXECUTE THE FOLLOWING SEQUENCE.

     RUN ERRLOG
```

To attempt to run ERRLOG again, type the request to run ERRLOG again. When the request for the number of nodes is printed respond with a smaller number to 5 SECOND ERROR RATE = .

The second case that causes ERRLOG to terminate is when the error logging device becomes full. The following message is printed on the console.

```
ERROR LOGGING DEVICE device and unit number FULL. "ERRLOG"
TASK TERMINATED. IF YOU WISH TO CONTINUE LOGGING ERRORS
EXECUTE THE FOLLOWING SEQUENCE.

     REA ERRLOG 4 device and unit number
     RUN ERRLOG
```

The REA command is detailed in the RSX-11D User's Guide.

Prior to reassigning the logging device, the new device to which it is to be reassigned must be given a UFD of [1,6] if it does not have one already. The UFD must have the following access rights [RWED,RWED,RWED,RWED].

Type the following command to place the UFD on the disk.

```
UFD dev:[1,6]/PRO=[RWED,RWED,RWED,RWED]
```

The UFD command is detailed in the RSX-11D User's Guide.

After creating the UFD, type the sequence provided in the console printout.

It is suggested that the entire volume be cleared of old files using PIP and VFY (verify). PIP and VFY are described in the RSX-11 Utilities Procedures Manual.

The third case in which ERRLOG terminates occurs if an error occurs while trying to write to the logging device. The following information is printed on the console.

```
ERROR - xx ON ERROR LOGGING DEVICE device name
"ERRLOG" TASK EXITING.
TASK "ERRLOG" TERMINATED
VIA 'EXIT' WITH PENDING I/O
```

xx is the standard system code as defined in the IAS/RSX-11D Device Handlers Reference Manual. If MO is loaded additional error information is printed on the terminal. If it is desirable to continue error logging, reassign the logging device.

8.2.2 Running PSE

The function of the preanalyzer, PSE, is to format the raw data collected by ERRLOG into a file to be processed by SYE. In order to run PSE, PSE must be installed, and the user must be operating under a privileged UIC. In addition, if no ERROR.TMP file is available for PSE to process, the ERRLOG task must be running. ERRLOG can rename the ERR.TMP file and pass it to PSE.

Use the following command to install PSE.

```
MCR>INS [11,1]PSE
```

To initiate PSE after it has been installed, type the following command to MCR.

```
MCR>PSE
```

The preanalyzer responds with the prompt PSE> and waits for the user to type a command line. The format of the PSE command line follows.

```
OUTDEV:[ufd]file.ext=indev:
```

The output file specification is a standard RSX-11D specification.

The input file specification consists only of the input device specification. The file name is always ERROR.TMP and it is under [1,6]. The name is assigned by ERRLOG.

The following defaults are used for omitted portions of the file specifications.

```
outdev    defaults to SY:
ufd       defaults to [1,6]
file.ext  defaults to ERROR.SYS
indev     defaults to SY:
```

If the default values are to be used, press RETURN in response to the PSE prompt. When PSE prompts again, type CTRL Z to return to MCR.

8.2.3 Running SYE

The analyzer produces an error report in the form of a printed listing or a listing file.

Before SYE can run, the user must be operating at a privileged terminal and SYE must be installed. Type the following command to install SYE.

```
MCR>INS [11,1]SYE
```

Once SYE is installed, type the following command to run it.

```
MCR>SYE
```

The analyzer responds with the following prompt SYE> and waits for the user to type a command line. The format of the SYE command line follows.

```
outdev:[ufd]=index:[ufd]file.ext/switch1.../switchn
```

The output filename corresponds to the values specified for xxx and yyy for the /BR: switch described below. The file type is LST. The output filename and type should not be included in the command line.

The input file specification is a standard RSX-11D file specification. The input device, UFD, filename, and extension must be identical with the output file specification used when running PSE. It is the output of PSE that SYE analyzes.

The following switches can be used as part of the input file specification.

/BR:xxxyyy — is the breakout switch that determines what information is to be included in the report.

xxx can have one of the following values:

- ALL — indicates that error statistics for all disk, magnetic tape, and DECtape units are to be included.
- DEV — indicates that only device errors are to be itemized in the report; that is, no memory parity errors are to be included.
- SYS — indicates that only memory parity errors are to be included in the report.
- DSK — indicates that error statistics for all disk units are to be included.
- MAG — indicates that error statistics for all tape devices, both magnetic tape and DECtape, are to be included.

ALL is the default value for xxx.

yyy can have one of the following values:

ALL indicates that both the device-specific and the summary information is to be included in the report.

SUM indicates that only the summary information is to be included in the report.

DSK indicates that only disk errors are to be included in the summary report.

MAG indicates that only magnetic tape errors are to be included in the summary report.

SUM is the default for yyy.

The output filename is formed from the xxxyyy portion of the /BR switch. The output file type is .LST.

/ID:name indicates that the report of errors is to contain only those errors that occurred while a specified volume is mounted. The value name provides the volume identification.

The volume can be mounted using the /OVE switch if the volume identification is not known. The volume information printed on the console gives the volume identification once the device is mounted.

/DV:devn indicates that the report is to contain only those errors that occur on a specified device type or on a specified unit. For example, if devn is specified as DK, error statistics for all RK03 or RK05 units are provided. If devn is specified as DK1, error statistics for RK03 or RK05 unit 1 are provided.

/BG:time:date indicates that only those errors that occur after the specified time and date are to be included in the report. The format of the time and date specification follows.

hh:mm:ss:dd:mmm:yy

All numbers are decimal and all six fields must be specified.

/ED:time:date indicates that only those errors that occurred on or before the specified time and date are to be included in the report. Time and date have the same format as in the /BG switch.

/DE indicates that a more detailed report than that produced using the defaults is desired. When /DE is specified additional information is included if it is available:

- Device error register bit description
- Task and function that caused the entry to be logged
- Statistical information about the device
- Information relating to concurrent activity on the UNIBUS

The default is NODE or -DE.

See Appendix A for examples of the report when /DE is specified.

/-SU indicates that the summary report is not to be produced. The default is /SU.

The following are the default values for the SYE command string.

```
SYØ:[user uic]ALLSUM.LST=SYØ:[1,6]ERROR.SYS/BR:ALLSUM/NODE/SU
```

8.2.4 Terminating Error Logging - Use the following procedures to terminate the error logging task (ERRLOG) in an orderly fashion.

```
MCR>INS [11,1]ERROFF

MCR>RUN ERROFF
```

One of two messages is printed on the terminal when ERROFF has terminated error logging. The following message is printed to indicate that ERROFF executed successfully.

```
ERRLOG TASK TERMINATED
```

The following message is printed to indicate that ERRLOG was not active, and therefore, was not terminated.

```
ERRLOG TASK NOT ACTIVE
```

8.3 ERROR MESSAGES

Both the preanalyzer (PSE) and the analyzer (SYE) issue error messages to inform the user of operational difficulties.

8.3.1 PSE Error Messages

After each error message, PSE prompts again for a new command line.

COMMAND STRING PARSE ERROR

A syntax or semantic error was encountered while examining the input command string to PSE. PSE prompts again for a new command line.

Type a corrected version of the command line.

DELETE ERROR

After the preanalyzer processed the input file ERROR.TMP, it was unable to delete it.

Attempt to delete the file using PIP.

INPUT FILE ERROR

An error was encountered while trying to open or obtain data from the input file ERROR.TMP. ERROR.TMP is closed, processing is terminated, and the input file is not deleted.

PSE prompts again for a new command string. If this fails, delete the file.

NO ERROR FILES FROM SYSTEM

The preanalyzer is unable to locate a file named ERROR.TMP. This message can be caused by one of the following situations.

1. No errors have occurred. Therefore ERRLOG has no raw data file to pass to PSE.

2. ERRLOG is not running and, therefore, cannot rename the ERR.TMP file to ERROR.TMP and pass it to PSE.

3. ERRLOG is not writing to the specified device. Check the LUN assignment of ERRLOG. The logging device is assigned to LUN 4.

If the cause of the message is that ERRLOG is not running, follow the procedures in Section 2.2.1 to run the task.

If the cause of the message is that the ERROR.TMP file resides on a device other than the one specified, use the device assigned to LUN 4 of ERRLOG as the indev for the PSE command line.

OUTPUT FILE ERROR

An error was encountered while working with the PSE output file. Both the input and output files are closed. ERROR.TMP is not deleted.

Try to rerun PSE.

PREANALYZER OUTPUT DEVICE FULL

The output device became full while PSE was writing data to the output file. Both the input and output files are closed. ERROR.TMP is not deleted.

Rerun PSE using a different output volume.

UNABLE TO CLOSE INPUT FILE

PSE is unable to close the file ERROR.TMP. The file is not deleted. File processing is terminated.

Use PIP to delete the file.

UNABLE TO CLOSE OUTPUT FILE

PSE is unable to close the output file.

Rerun PSE or use PIP to delete the file.

8.3.2 SYE Error Messages

SYE--COMMAND STRING ERROR
portion of the string in error

The format convention within a particular portion of the command string is violated. No files remain open. SYE issues a prompt.

Correct the error and type the command.

SYE--COMMAND STRING ERROR
ERROR NUMBER n

The command string interpreter detects an error while attempting to get a command line. n is a CSI error code. Refer to the IAS/RSX-11 I/O Operations Reference Manual to determine the meaning of n. No files remain open. SYE issues a prompt.

Correct the error and type the command.

SYE--COMMAND STRING SYNTAX ERROR
command string typed

The proper format was not used in the command string. No files are open. SYE issues a prompt.

Type the corrected command.

SYE--DEVICE ERROR INPUT FILE
FATAL ERROR - n

SYE was attempting to obtain further information from the input file but could not get the next record. n is an FCS error code. Refer to the IAS/RSX-11 I/O Operations Reference Manual to determine the meaning of n.

Both the input and output files are closed. SYE issues a prompt for the next command.

SYE--DEVICE ERROR OUTPUT FILE
FATAL ERROR - n

SYE was unable to write information in the output file. n is an FCS error code. Refer to the IAS/RSX-11 I/O Operations Reference Manual to determine the meaning of n.

Both the input and output files are closed. SYE issues a prompt for the next command.

SYE--ILLEGAL BREAKOUT SWITCH
/BR:xxxyyy

SYE issues this message when the operator attempts to request a breakout of the input file that is not legal. No files remain open. SYE issus a prompt for another command.

Retype the command with a correct use of /BR:.

SYE-- OPEN FAILURE ON INPUT FILE
FATAL ERROR - n

SYE was unable to open the input file. No files remain open. n is an FCS error code. Refer to the IAS/RSX-11 I/O Operations Reference Manual to determine the meaning of n.

SYE--OPEN FAILURE ON OUTPUT FILE
FATAL ERROR - n

SYE was unable to open the output file. No files remain open. n is an FCS error code. Refer to the IAS/RSX-11 I/O Operations Reference Manual to determine the meaning of n.

SYE--n. PAGES
text

This message does not indicate an error; rather, it indicates the number of pages (n) in the finished report. The value "text" indicates the filename for future reference.

SYE--SUMMARY TABLE OVERFLOW
REPORT CONTINUES WITHOUT SUMMARIES

Summary table overflow occurred because more devices received errors than the analyzer is built to handle.

Use the breakout switches to reduce the number of devices included in the summary.

CHAPTER 9

SYSTEM MANAGEMENT NOTES

This chapter contains information that can be used to optimize system performance and to alter certain system defaults.

9.1 ANCILLARY CONTROL PROCESSORS AND THEIR USE

An ancillary control processor (ACP) is a file processing task that interfaces between a user task and FCS on the one hand and a device handler on the other. The device handler performs basic operations, such as read and write, without aid of an ACP. However, file processing functions, such as create, delete, and access, are performed by an ACP task associated with the device. Not all devices have ancillary control processors associated with them.

The ancillary control processor to be associated with a device is specified either explicitly or by default during system generation as a parameter in the DEV directive. RSX-11D supplies three ACPs:

F11ACP	for Files-11 devices (disk and DECtape),
MTAACP	for magnetic tape handling,
DTAACP	for installations that require significant amounts of DECtape processing.

F11ACP and MTAACP are the default ACPs for Files-11 devices and magnetic tape, respectively. DTAACP is used for DECtape instead of F11ACP if so specified during system generation. If desired, a user-written ACP can be associated with any device during system generation. A user-written ACP can be specified instead of a standard RSX-11D ACP or to perform special functions on a device that does not have a standard ACP associated with it. Only one ACP can be associated with each device.

System generation establishes the default ACP to be associated with a device when it is mounted. However, use of the /ACP switch with the MOUNT command can override the default ACP for a device.

All ACPs must have a 6-character task name that ends with the letters ACP. ACPs must be installed either during system generation or by means of the INSTALL command to MCR.

9.1.1 Files-11 ACPs

Two versions of F11ACP are supplied with RSX-11D: FCP.TSK and BIGFCP.TSK. They appear under different filenames in UFD [11,1].

The file named FCP.TSK is a minimum size, heavily overlaid version occupying approximately 2.4K words of memory. It is recommended for small systems with limited space and where high file processing throughput is not essential.

The file named BIGFCP.TSK is minimally overlaid and has additional buffer space. It incurs almost no overlay switching during normal operations and provides some optimization of disk accesses due to its additional buffer. BIGFCP.TSK occupies approximately 4.5K words of memory.

Both FCP.TSK and BIGFCP.TSK are built with a task name of F11ACP.

The file DTAACP.TSK is a version of F11ACP that is optimized for DECtape operation. It is heavily overlaid, but has the additional buffer. It occupies approximately 2.6K words of memory. DTAACP.TSK is built with a task name of DTAACP.

In large systems with a heavy file processing requirement (for example, a multiuser program development system), multiple versions of an ACP can be installed to achieve some parallelism in file processing. Each ACP task processes all functions in a strictly sequential manner. Having two versions of F11ACP installed in the system allows each version to service requests for a different set of devices.

For example, in a system with two RP04 drives and two RK05 drives, the system manager could specify F11ACP as the ACP for the RP04 drives and RKAACP as the ACP for the RK05 drives during system generation. By installing BIGFCP twice, the system manager could provide each device type with its own ACP. File processing then occurs concurrently on RP04s and RK05s. In addition, the extra buffer space available in the two copies of the file processor, whether it be FCP or BIGFCP, saves a certain percentage of disk accesses.

An installation that does a significant amount of Files-11 DECtape I/O in a multiuser environment should consider using DTAACP for DECtape and F11ACP for disks. Use of DTAACP ensures that the processing of DECtapes does not interfere with disk file processing.

9.1.2 Magnetic Tape ACP

The magnetic tape ACP is named MTAACP and provides support of ANSI magnetic tape file structures. Unlike F11ACP, it is both nonfixable and checkpointable.

MTAACP does not need to be fixed in memory for performance reasons. When a magnetic tape is mounted, MTAACP is loaded into memory if it is not already there. It does not exit from the system after it services a request; the only time MTAACP exits is when no magnetic tape volumes are mounted and no outstanding I/O requests remain to be serviced. The result is that once a magnetic tape volume is mounted, which requires use of MTAACP, the ACP task is effectively locked in memory.

MTAACP is built with a priority of 200. Because it is checkpointable, a deadlock can occur on either a normal I/O request or I/O rundown in the following situation:

1. A task with a higher priority than MTAACP causes it to be checkpointed,

2. That task then needs the services of MTAACP but insufficient memory remains to load MTAACP.

9.1.2.1 File and Record Attribute Support under MTAACP

The ANSI magnetic tape file system (MTAACP) supports the FCS record types and attributes with the exception of symbolic offset FD.BLK in the FDB. FD.BLK is set unconditionally to indicate that records cannot cross block boundaries. The result is that, for files containing variable-length records, the block length (that is, buffer size) must be equal to or greater than the maximum record size plus four. Therefore, the default block size of 512 bytes yields a maximum variable-length record size of 508 bytes.

MTAACP does not support the FCS file characteristics (attributes) in that it treats all files as noncontiguous. If a file on magnetic tape is to be contiguous on disk or DECtape, the following steps must be performed using PIP:

1. Transfer the file from magnetic tape to either a Files-11 disk or DECtape to cause FCS to determine the real size of the file in blocks.

2. Transfer the file from the disk or DECtape to the desired device using the /CO switch in the output file specification to PIP.

9.2 /LRU SWITCH ON INITVOL AND MOUNT

Another optimization feature is the /LRU switch that can be specified on the INITVOL and MOUNT commands to MCR. All versions of F11ACP maintain in memory a list of recently referred to directories. Each mounted volume has an associated list. If a directory is present in the list, the overhead involved in gaining access to the directory is almost completely eliminated. Entries in the list are replaced on a least-frequently-used basis.

The /LRU switch specifies the number of entries that are to be retained in the list for a particular volume while it is mounted. The INITVOL and MOUNT default value is three, which is an appropriate value for a single-user volume. In general, the /LRU value should be a few units greater than the expected number of concurrent users of the volume.

9.3 TAILORING F11ACP

The file primitives (F11ACP) in RSX-11D Version 6B can be tailored to suit the needs of a particular installation in several ways. All involve balancing certain size/performance tradeoffs.

9.3.1 Overlay Structure

The file primitives are almost always run overlayed. Two task builder command files and overlay description files are provided for two different overlay structures. The smallest FCP (FCP.CMD and FCP.ODL) occupies roughly 2.4K words of memory. It is structured with one overlay for each FCP function and is recommended for small systems where space is limited and high file processing throughput is not important. A larger overlay structure is also provided (BIGFCP.CMD and BIGFCP.ODL). It incurs almost no overlay switching during normal file processing; overlays are used only for errors and exceptions.

9.3.2 Additional Directory Buffer

A separate buffer for directories can be allocated in FCP with either overlay structure. The presence of this buffer speeds up operation when repeated references are made to the same directory; for example, such as file purges and wild card operations. The directory buffer is allocated by including the following statement in the task builder command file to build FCP:

```
EXTSCT=$$BUF1:1006
```

9.3.3 Extended Directory Buffer

If a significant percentage of the directories used in a system are large (greater than 30 entries), an additional performance improvement can be realized by increasing the size of the directory buffer. Then fewer physical disk accesses are necessary to search a large directory. The directory buffer should be extended in units of one block, expressed in multiples of 1000 bytes (octal). For example, the following task builder statement:

```
EXTSCT=$$BUF3:2000
```

adds two blocks to the directory buffer, allowing the file system to handle directories in three block units. Extending the directory buffer can be done only if a separate directory buffer has also been allocated as described above in Section 9.3.2.

9.3.4 Additional Storage for FCB's

The file primitives use one file control block (FCB) for each file that is currently open in the system. In addition, each directory LRU list entry occupies one FCB, and one FCB is used for each mounted volume for the index file. Therefore, the total number of FCBs allocated in the system is the sum of the number of mounted volumes plus the sum of all mounted volumes' LRU limits plus the total number of open files. Each FCB is 21 words in length. FCBs are normally allocated in the system node pool, where they occupy 24 words apiece. Pool space is allocated in 8-word units. Clearly, in a large multiuser system FCBs can take up a substantial amount of pool space (several K, in fact). The system manager can allocate space for FCBs within the task body of the file primitives with a task builder command of the following form:

```
EXTSCT=$$FRE1:2000
```

The space is expressed in octal bytes. Each unit of 1000 has space for approximately 21 FCBs. FCP allocates FCBs out of this area until it is full; it then uses the system node pool.

NOTE

> In the system as distributed, FCP has no internal FCB space and BIGFCP has 2000 bytes of internal space.

9.4 REASSIGNMENT OF TASK BUILDING AND CROSS REFERENCE LUNS

Both the Task Builder and the cross reference task (CRF) use dynamic memory for symbol processing. System performance when TKB or CRF execute can be enhanced given appropriate resources.

TKB uses LUN 8 for its dynamic workfile, while CRF uses LUN 7. In the released system, these logical unit numbers are assigned to the device on which the TKB and CRF tasks reside. Both tasks are overlaid so that, in any configuration, it may be advantageous to reassign the logical unit numbers to another device.

Example:

```
MCR>REA CRF... 7 DP1:
```

9.5 COLUMN WIDTH FOR CRF OUTPUT

The released CRF is built with 132-column line output.

The column width for printed output from CRF can be altered by rebuilding the cross reference task. The task build command file for CRF resides under UFD [11,32] on the auxiliary distribution medium and is called CRFBLD.CMD. It contains the following line:

```
EXTSCT=$$RCB0:204
```

This line establishes a 132-column line width. Omission of this option by inserting a semicolon (;) before the line causes the building of a CRF with an 80-column line output.

9.6 CHANGING TASK BUILDER DEFAULTS

The RSX-11D Task Builder contains default values that are used when switches and options are omitted from a command to the Task Builder. These switches and options and their defaults are detailed in the RSX-11D Task Builder Reference Manual.

Some installations may prefer a different set of defaults in TKB: in particular, a default of "task unaccountable" may be desirable. Defaults can be changed by rebuilding the Task Builder with a global patch applied.

To apply the patch, three files must be obtained from UFD [11,11] of the auxiliary distribution medium: TKBBLD.CMD, TKBODL.ODL, and TKB.OLB. These files are the build command file, the overlay-descriptor file, and the relocatable object library of all TKB modules, respectively.

The list below provides the available task image attributes and the bit pattern required to set them. The default in the released version of the Task Builder was achieved by initializing the task image with SW$FX!SW$FP.

```
SW$CP==100000          ; TASK CHECKPOINTABLE (0=YES)
SW$FP==040000          ; TASK USES FLOATING POINT (1=YES)
SW$DA==020000          ; TASK DEBUGGING AID SPECIFIED (1=YES)
SW$PI==001000          ; PIC OUTPUT (1=YES) (ALSO APPLIES TO STB)
SW$PR==000400          ; TASK IS PRIVILEGED (1=YES)
SW$TR==000200          ; SET TRACE BIT IN PS (1=YES)
SW$NH==010000          ; BUILD TASK IMAGE WITHOUT HEADER (1=YES)

SW$FX==004000          ; TASK IS FIXABLE (0=YES)
SW$AB==000020          ; TASK ABORTABLE (0=YES)
SW$DS==000010          ; TASK DISABLABLE (0=YES)
SW$MU==000002          ; +++016 MULTI-USER SWITCH (1=YES)
SW$TA==000001          ; +++016 ACCOUNTING SWITCH (1=YES)
```

Referring to the list above, the defaults produce a pattern of 44000.

To build a TKB that has these defaults (not fixable, floating point, etc.) plus the task accountable default, the following pattern is required:

```
SW$FX!SW$FP!SW$TA = 44001
```

Having calculated the required default bit pattern from the list above, edit the file [11,11]TKBBLD.CMD to insert the option in the following form:

```
GBLPAT=P11OVR:$MRFLG+134:value
```

value = the requisite bit pattern for the desired defaults; for example, 44001 is used to produce the standard released defaults with the addition of the task accountable default.

CHAPTER 10

EXECUTIVE DEBUGGING AIDS

RSX-11D provides two debugging aids that can be incorporated into the Executive:

1. Executive ODT

2. Executive trace.

10.1 EXECUTIVE ODT

A simplified version of ODT can be task built into the Executive if needed for the debugging of user-initiated modifications to the Executive. This version of ODT can perform the following functions:

1. Set and remove breakboints,

2. Examine the content of a location and modify the content,

3. Search for bit patterns.

Executive ODT operates only in Kernel space. For example, breakpoints can be set only in Kernel space and only Kernel locations can be examined.

10.1.1 Building ODT into the Executive

Use the following procedures to build ODT into the Executive and generate the new system.

1. Log onto the system under [1,1].

2. In the file [11,15]TKB15.CMD, locate the line [11,15]NODT and change it to [11,15]ODT.

3. Rebuild the Executive using [11,15]TKD15.CMD as in the following command.

   ```
   TKB>[11,15]TKB15
   ```

4. Rebuild the bootstraps using the command file [11,17]BOOTSBLD.CMD.

5. Perform a system generation using an appropriate command file, examples of which are distributed under UFD[11,17].

   After phase 1 of system generation when the disk is booted, the following message is printed at the terminal.

```
S/T RSX ODT V7.00
*
```

6. Press CTRL/C to initiate phase 2 of system generation.

7. Follow the formal procedures for saving the system at the end of phase 2 as described in the <u>RSX-11D System Generation Reference Manual</u>.

10.1.2 <u>Using Executive ODT</u>

NOTE

It is not possible to use both Executive ODT and user task ODT at the same time.

Once the system generation is completed, the ODT message (S/T RSX EXEC ODT V.07) is printed at the terminal each time the system is bootstrapped or a power failure recovery occurs. When the message is printed, it is followed by the ODT prompt, which is an asterisk (*). The user can perform either of the following in response to the ODT prompt:

1. Interact with ODT,

2. Press CTRL/C to allow normal system operation.

Before CTRL C is depressed, the user can issue the commands to set or remove breakpoints, examine or modify the contents of locations, or search for bit patterns. These commands are detailed in the <u>IAS/RSX-11 ODT Manual</u>. Examples are provided below.

Once the necessary initial set-up for Executive ODT is complete, press CTRL C to obtain MCR. Normal RSX-11D operation begins and continues until an instruction in a breakpoint location is executed. Then ODT prompts with an asterisk. At that point, any of the commands to ODT named in Section 10.1 can be used. When ready to continue normal RSX-11D operation, use the P command as described in the <u>IAS/RSX-11 ODT Manual</u>.

10.1.2.1 <u>Setting a Breakpoint</u> - In order to set a breakpoint, the user needs a listing of [111,15]EXEC.MAP and a listing of each module in which a breakpoint is to be set. This is similar to the process used with task ODT. The following is an example of setting a breakpoint at location 3612.

```
*3612;OB        The asterisk (*) is the ODT prompt.
```

10.1.2.2 <u>Examining and Modifying a Location</u> - The following command causes ODT to print the contents of location 3716 on the terminal and allows the user to modify the content. Location 3716 contains 000001; the user types 4 to indicate that ODT is to change the content at the location to 000004.

```
*3716/000001  4
```

10.1.2.3 Sample Use of ODT - The following example illustrates the use of Executive ODT.

```
S/T RSX ODT V7.00                 ;SYSTEM IS BOOTED
*3612;OB                          ;SET INITIAL BREAKPOINT
*^C                               ;PRESS CTRL/C
     RSX-11D V6.2                 ;NORMAL SYSTEM OPERATION COMMENCES
MCR>MOU DK0:/OVR                  ;MOUNT SYSTEM DISK
MCR>HEL [1,1]                     ;LOG ONTO SYSTEM
MCR>INS TEST                      ;INSTALL TASK NAMED TEST
MCR>RUN TEST$                     ;RUN TEST AND DEPRESS ALTMODE
B0:3612                           ;TEST CAUSED EXECUTIVE TO EXECUTE
                                  ;INSTRUCTION AT LOCATION 3612
*3646;1B                          ;SET ANOTHER BREAKPOINT
*$1/000542                        ;EXAMINE CONTENTS OF REGISTER 1
*P                                ;CONTINUE RSX-11D OPERATION
MCR>
```

10.2 EXECUTIVE TRACE

The Executive trace facility is part of the resident Executive and is useful for tracing ISRs. The Executive trace allows the user to trace ranges within an ISR and to halt at specified locations.

10.2.1 Building Trace into the Executive

The resident Executive in the distributed system does not contain the trace facility. To include trace, the Executive must be rebuilt according to the steps below.

1. Put UFD [11,15] on disk and set it as the current UIC.
2. Use FLX to transfer the [11,15] files from the distribution binary medium.
3. Repeat steps 1 and 2 for UFD [11,17].
4. Modify [11,15]TKB15.CMD by changing the line [11,15]NTRACE to [11,15]TRACE.
5. Task Build the Executive using TKB15.CMD and task build the bootstraps using [11,17]BOOTSBLD.CMD.
6. Change the system generation phase 1 command file to reflect the increased size of the Executive.
7. Generate the new system.

10.2.2 Using Executive Trace

Once the system generation is completed, use the steps listed below to cause trace to execute.

1. Find the address of the global symbol .DBTRA in the EXEC.STB listing which can be obtained from [11,15]EXEC.MAP.

2. Using the OPE command to MCR, set up to four trace ranges in the four word-pairs starting at location .DBTRA-30. Addresses are in Kernel virtual space. The following is an example of using the OPE command to specify one range.

   ```
   MCR>OPE .dbtra-30/KNL      Type in actual address of
   001160 000000/4140         .DBTRA-30
   001162 000000/4147
   001164 000000/$            Depress ALTMODE to return
                              to MCR.
   ```

   Alternatively, set console switch 0 up to trace the entire ISR.

3. If desired, set up to five halt addresses in the four word-pairs starting at .DBTRA-40 or set one halt address in console switches 15 through 1. Use the OPE command as in step 2 to set the halt addresses.

4. Change Kernel address 14 to contain the address of .DBTRA as follows:

   ```
   MCR>OPE 14/KNL
   000014 nnnnnn/xxxxxx    nnnnnn = present contents
                                    of 14.
                           xxxxxx = address of .DBTRA
                                    as specified by
                                    the user.
   ```

5. Set the T-bit (bit 4) in the second word of the interrupt vector for the interrupt service routine to be traced; for example in the DP ISR, set bit 4 in vector 254.

The trace printed on CL is essentially in the same format as that for the task trace package; only the relative PC is omitted. Refer to the task trace chapter of the IAS/RSX-11 ODT Manual.

If one of the requested halt addresses is encountered, the computer halts with the requested address in the data lights. Execution can be resumed by pressing CONT.

CHAPTER 11

PARITY SUPPORT

Under RSX-11D, the handling of errors is divided into three categories:

1. Main memory parity support,

2. Cache memory parity support on a PDP-11/70,

3. Additional PDP-11/70 hardware error support.

For a description of cache memory, refer to the PDP-11/70 Processor Handbook.

11.1 MAIN MEMORY PARITY SUPPORT

Main memory parity errors are an indication of a memory malfunction. When a parity error occurs, it causes a trap to the RSX-11D Executive. The Executive determines whether the error was fatal (nonrecoverable) or not and then takes appropriate action. In the PDP-11/40, 11/45, and 11/50, all parity errors are fatal and are treated as main memory errors.

The PDP-11/70 hardware recognizes two types of main memory errors: address and data. The Executive handles both types of errors identically, although errors are logged as being one of the two types; refer to Chapter 8 for information on the logging of errors.

When a fatal main memory parity error occurs, the Executive terminates the task executing where the error occurred unless that task has a hardware priority greater than three. The aborted task remains in memory and no further access to or execution of the task is allowed. Locking the task in memory also makes the entire area of memory allocated to the task inaccessible to the system or other tasks. When the Executive terminates a task, an error message is printed on CO and on the terminal that initiated the task.

If the Executive encounters any of the following conditions, it halts system operation; that is, a system crash occurs:

1. The hardware priority of the task executing where the error occurred is greater than three,

2. No task is executing currently,

3. The error occurred in Kernel mode.

The only difference between parity handling on the PDP-11/70 and other PDP-11´s is that the 11/70 provides the Executive with an indication of whether the error was fatal. Fatal errors are handled as described above. Nonfatal errors are transparent to the task executing where the error occurred, but are logged by the Executive as described in Chapter 8.

11.2 CACHE PARITY SUPPORT (PDP-11/70 ONLY)

Because cache contains a copy of information also maintained in main memory, it can recover from cache parity errors by obtaining the information from main memory. Fatal parity errors are never returned for cache. Rather, the Executive performs the following sequence:

1. Determines the cache group in which the error occurred,

2. Determines whether the group has exceeded the number of errors allowed in a 1-minute period,

3. Turns the group off if the limit has been exceeded,

4. Logs the error.

This process is transparent to the task using the cache group in which the error occurs. However, program execution speed may be affected.

The default number of errors allowed in a 1-minute period is 50 (decimal). This number can be changed by the system manager as described in Section 11.4.

The PDP-11/70 hardware recognizes two types of cache errors: address and data. The Executive handles both types of errors identically, although errors are logged as being of the two types; refer to Chapter 8 for information on error logging.

11.3 ADDITIONAL PDP-11/70 HARDWARE ERROR SUPPORT

The RSX-11D Executive handles three other types of hardware errors that the PDP-11/70 reports:

Nonexistent memory,

UNIBUS error,

CPU/UNIBUS abort.

11.3.1 Nonexistent Memory

A nonexistent memory error implies that faulty mapping or a bad address has caused the central processing unit to refer to a nonexistent memory location. This type of error is treated in the same manner as a main memory parity error.

11.3.2 UNIBUS Error

A UNIBUS error occurs when bad memory is referred to during a direct memory access (DMA) transfer. Since the task currently executing is probably not the task for which the I/O is being done, the Executive makes no attempt to determine which task issued the I/O request or to locate the bad memory. The error is only logged. Some UNIBUS device controllers detect the memory error and report it at the end of the transfer.

11.3.3 CPU/UNIBUS Abort

The CPU/UNIBUS abort occurs when bad data is transferred during a non-DMA I/O transfer or a read of UNIBUS I/O page registers. The Executive terminates the task currently executing because it is the task that issued the I/O request. The error is logged.

11.4 SYSTEM MANAGER'S CONTROL OF PARITY HANDLING

Two MCR functions, SET and MEM, provide the system manager with a means of controlling the effects of parity error handling. Both of these functions are detailed in the RSX-11D User's Guide.

11.4.1 SET Functions for Controlling Parity

The SET command is used in conjunction with cache parity. It allows the system manager to turn cache groups on and off and to change the default number of errors that are allowed in a 1-minute period.

Examples:

MCR>SET /CAC=75	This command indicates that 75 cache parity errors are to be allowed in a 1-minute period before the Executive turns the group off. The system default is 5Ø.
MCR>SET /CAC=ON1	This command turns on cache group 1.
MCR>SET /CAC=OFØ	This command turns off cache group Ø.

The system manager can determine which groups are receiving errors by consulting the error log as described in Chapter 8.

11.4.2 MEM Function for Freeing Memory

The MEM command to MCR allows the system manager to free a task locked in memory due to main memory parity errors and to free the memory allocated to it. That task then can be run again; however, the memory can also be used for subsequent task execution and can result in further task terminations.

Examples:

MCR>MEM PROG1	This command frees the first task encountered that has the name PROG1.
MCR>MEM PROG1/TI=TT3	This command frees the task PROG1 that was initiated from terminal 3.
MCR>MEM PROG1/AL	This command frees all locked versions of PROG1.

11.5 TASK CONTROL OF PARITY HANDLING

The user can avoid having a task terminated by the Executive when a parity error occurs by declaring an SST (synchronous system trap) in the task. SSTs are described in the <u>RSX-11D Executive Reference Manual</u>.

APPENDIX A

CDA SAMPLE LISTING

```
RSX-11D CORE-DUMP ANALYZER       V06.2        5-OCT-76    14:25          PAGE 1
CRASH STACK


CAUSE OF CRASH:  NON-TASK LEVEL SEGMENT FAULT OR TRAP FOUR


AFTER CRASH:  PS=004344   SP(K)=001454   SP(U)=170542

BEFORE CRASH:  PC=004456   PS=004301

R0=013173    R1=000001    R2=000001    R3=006000    R4=000774    R5=012667

SR0=000003    SR1=000000    SR2=035576    SR3=000000

                                  U S E R

                    I    S P A C E              D    S P A C E
                    PDR        PAR              PDR        PAR
                   000406     000021           000000     000377
                   000000     000000           000000     007717
                   000000     000000           000000     002437
                   000000     000000           000000     047733
                   000000     000000           000000     007660
                   000000     000000           000000     127400
                   000000     000000           000000     007624
                   000000     000000           000000     006000

                           S U P E R V I S O R

                    I    S P A C E              D    S P A C E
                    PDR        PAR              PDR        PAR
                   000000     000000           000000     000000
                   000000     000000           000000     000017
                   000010     000000           000010     000217
                   000014     000000           000000     000017
                   000000     000000           000000     000000
                   000016     000000           000007     000017
                   000010     000000           000000     000000
                   000017     000000           000000     000000

                               K E R N E L

                    I    S P A C E              D    S P A C E
                    PDR        PAR              PDR        PAR
                   077506     000004           060017     125252
                   077106     000204           044017     125252
                   000000     000000           010015     125252
                   077406     000021           077415     125252
                   000116     000402           030013     125252
                   077406     000602           047010     165542
                   077506     001002           010011     000000
                   077506     177600           000000     000000
```

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76     14:25          PAGE 2
KERNEL STACK


  KERNEL STACK:

    001000      004216  004340  000000  000000  000000  000000  000000  000000
    001020      000000  000000  000000  000000  000000  000000  000000  000000
    001040      000000  000000  000000  000000  000000  000000  000000  000000
    001060      000000  000000  000000  000000  000000  000000  000000  000000
    001100      000000  000000  000000  000000  000000  000000  000000  000000
    001120      000000  000000  000000  000000  000000  000000  000000  000000
    001140      000000  000000  000000  000000  000000  000000  000000  000000
    001160      000000  000000  000000  000000  000000  000000  000000  000000
    001200      000000  000000  000000  000000  000000  000000  000000  000000
    001220      000000  000000  000000  000000  000000  000000  000000  000000
    001240      000000  000000  000000  000000  000000  000000  000000  000000
    001260      000000  000000  000000  000000  000000  000000  000000  000000
    001300      000000  000000  000000  000000  000000  000000  000000  000000
    001320      000000  000000  000000  000000  000000  000000  000000  000000
    001340      000000  000000  000000  002352  034110  142040  062352  004240
    001360      060166  000012  040142  071102  004240  040140  000001  000400
    001400      062574  004250  061230  000400  061312  000400  103360  141740
    001420      150076  034040  000002  147364  034151  034111  145170  147554
    001440      110254  034111  000000  015474  035266  035122  000006  004456
    001460      004301  004320  004300  140017  013171  034300  000074  174011
```

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76     14:25          PAGE 3
SYSTEM COMMUNICATIONS AREA


CRASH OCCURRED AT 14:24:03   5-OCT-76


.CRTSK = NULL TASK

.FP45X=1     .PDP11=70.     .MSIZE=8192.=256K     .CKFSZ=515.

.SERFG=000000     .COMEF=000000 040000



CHECKPOINT BITMAP:

    037500      177777  177777  177777  177777  177777  177777  177777  177777


SCOM DATA AREA:

    146100      145740  000001  000106  020000  000000  000000  000400  000062
    146120      070100  177777  170001  177777  000002  105721  041104  000062
    146140      000017  000071  000044  000012  000005  000012  000114  177546
    146160      000100  000001  000000  000000  040000  051522  030530  042061
    146200      000454  026226  000000  004401  060146  060146  000000  000000
    146220      000374  100100  000022  000000  000000  000000  104760  103220
    146240      144724  000000  000000  000000  144724  000000  000000  000000
    146260      000000  000000  000000  000000  000000  005111  025167  003310
    146300      000000  000000  000000  000000  146316  000000  000003  104760
    146320      144724  177777  000402  000602  001002  177600  000016  077406
    146340      077406  077406  144642  000002  070624  145100  145174  142240
    146360      142140  142300  145070  145200  000210  000200  145600  145740
    146400      106060  106320  107440  107440  111060  111060  111720  111720
    146420      101160  000000  100000  000000  037500  037520  001003  000002
    146440      000002  000017  045462  000000  000003  000050  000001  000500
    146460      001051  000001  145150  041104  000000  014554  000172  000005
```

```
RSX-11D CORE-DUMP ANALYZER        V06.2          5-OCT-76     14:25           PAGE 4
ATL SCAN


DB....
------
  ATL ADDRESS = 145600V/106000R     STD ADDRESS = 142100V/102300R
  REQUESTOR = DB....           LOAD ADDRESS = 00120200
  PAR=SYDISK    PRI=250.    I/O IN PROG=0.    I/O PEND=0.    STATUS=W0    TI=TT0
  EVENT FLAGS=000200 000000    MASKS=000007 000000 000000 000000
  TASK FLAGS: 000000

    145600     100260  146374  142100  142300  000372  000000  000000  001202
    145620     000016  142100  000200  000000  000007  000000  000000  000000
    145640     145100  145642  145642  000000  000000  000000  145654  145654


TT....
------
  ATL ADDRESS = 100260V/040460R     STD ADDRESS = 141740V/102140R
  REQUESTOR = TT....           LOAD ADDRESS = 00150100
  PAR=GEN       PRI=248.    I/O IN PROG=0.    I/O PEND=0.    STATUS=W0    TI=TT2
  EVENT FLAGS=000000 040000    MASKS=007377 000000 000000 000000
  TASK FLAGS: 000000

    100260     101040  145600  141740  144332  000370  000000  000001  001501
    100300     000016  141740  000000  040000  007377  000000  000000  000000
    100320     145150  100322  100322  000002  027733  000000  100334  100334


LP....
------
  ATL ADDRESS = 101040V/041240R     STD ADDRESS = 104360V/044560R
  REQUESTOR = ...MFT           LOAD ADDRESS = 00225600
  PAR=GEN       PRI=248.    I/O IN PROG=0.    I/O PEND=0.    STATUS=W0    TI=TT0
  EVENT FLAGS=000000 000000    MASKS=000003 006401 000000 000000
  TASK FLAGS: 000000

    101040     106460  100260  102360  142300  000370  000000  000000  002256
    101060     000016  104360  000000  000000  000003  006401  000000  000000
    101100     145150  101102  101102  000001  141022  000000  101114  101114


DK....
------
  ATL ADDRESS = 106460V/046660R     STD ADDRESS = 103420V/043620R
  REQUESTOR = ...MFT           LOAD ADDRESS = 00251000
  PAR=GEN       PRI=248.    I/O IN PROG=0.    I/O PEND=0.    STATUS=W0    TI=TT0
  EVENT FLAGS=000000 000000    MASKS=000003 006401 000000 000000
  TASK FLAGS: 000000

    106460     106720  101040  102360  142300  000370  000000  000000  002510
    106500     000016  103420  000000  000000  000003  006401  000000  000000
    106520     145150  106522  106522  000001  000066  000000  106534  106534
```

•
•
•

```
RSX-11D CORE-DUMP ANALYZER       V06.2        5-OCT-76    14:25          PAGE 8
TASK HEADERS


DB....
------
  HEADER ADDRESS = 00120200    ATL ADDRESS = 145600V/106000R
  H.CR1=000000
  USER PDR'S: 035606  000000  000000  000000  000016  077406  077406  077406
  USER PAR'S: 001206  000000  000000  000000  000402  000602  001002  177600
  PS=174000     PC=000412
  R0=007034  R1=112060  R2=000000  R3=000001  R4=100200  R5=145600  SP=000356
  INITIAL PS=174017    INITIAL PC=000100    INITIAL SP=000100
  HEADER SIZE=000003    DSW=000001    NUMBER OF LUNS=6.


  LOGICAL UNIT TABLE:

  #  DEV  WINDOW    W.CTL     W.VBN     W.FCB  F.FNUM  F.FSEQ  F.STAT  NAC  NLCK

  1  SY0  000000
  2  SY0  000000
  3  SY0  000000
  4  SY0  000000
  5  TI0  000000
  6  CL0  000000

  HEADER:

    000000      000000  035606  000000  000000  000000  000016  077406  077406
    000020      077406  001205  000000  000000  000000  000402  000602  001002
    000040      177600  000003  000000  000000  000000  000402  000602  001002
    000060      177600  174000  000412  007034  112060  000000  000001  100200
    000100      145600  000356  174017  000100  000100  000000  000000  000000
    000120      100000  000000  000401  000401  000003  000010  000000  000000
    000140      137665  000000  000000  060146  060146  000000  000000  000006
    000160      145006  000000  145006  000000  145006  000000  145006  000000
    000200      000001  000000  142526  000000  000000  000000  000000  000000
    000220      000000  000000  000000  000000  000000  000000  000000  000000
    000240      000000  000000  000000  000000  000000  000000  000000  000000
    000260      000000  000000  000000  000000  000000  000000  000000  000000


TT....
------
  HEADER ADDRESS = 00150100    ATL ADDRESS = 100260V/040460R
  H.CR1=000000
  USER PDR'S: 061206  077406  023106  065106  000016  077406  077406  077406
  USER PAR'S: 001503  004227  002034  002103  000402  000602  001002  177600
  PS=174000     PC=004114
  R0=004010  R1=000044  R2=000000  R3=000001  R4=177777  R5=040040  SP=000172
  INITIAL PS=174017    INITIAL PC=000200    INITIAL SP=000200
  HEADER SIZE=000002    DSW=000001    NUMBER OF LUNS=3.


  LOGICAL UNIT TABLE:
```

```
RSX-11D CORE-DUMP ANALYZER       V06.2        5-OCT-76     14:25          PAGE 9
TASK HEADERS

  #  DEV   WINDOW    W.CTL     W.VBN     W.FCB  F.FNUM  F.FSEQ  F.STAT   NAC   NLCK

  1  CO0   000000
  2  SY0   000000
  3  DB0   000000

  HEADER:

   000000      000000  061206  077406  023106  065106  000016  077406  077406
   000020      077406  001503  004227  002034  002103  000402  000602  001002
   000040      177600  000002  000003  000001  000002  000402  000602  001002
   000060      177600  174000  004114  004010  000044  000000  000001  177777
   000100      040040  000172  174017  000200  000200  000000  002736  001000
   000120      101000  000000  000401  000401  000002  000023  000000  000000
   000140      027733  000000  000000  110720  110720  000000  000000  000003
   000160      142444  000000  145006  000000  144642  000000  000000  000000


LP....
------
  HEADER ADDRESS = 00225600    ATL ADDRESS = 101040V/041240R
  H.CR1=000000
  USER PDR'S: 003206  011602  000000  000000  000016  077406  077406  077406
  USER PAR'S: 002261  002270  000000  000000  000402  000602  001002  177600
  PS=174000     PC=020006
  R0=000370  R1=101040  R2=000000  R3=000001  R4=000000  R5=000406  SP=000100
  INITIAL PS=174017    INITIAL PC=000106    INITIAL SP=000100
  HEADER SIZE=000003    DSW=000001    NUMBER OF LUNS=6.


  LOGICAL UNIT TABLE:

  #  DEV   WINDOW    W.CTL     W.VBN     W.FCB  F.FNUM  F.FSEQ  F.STAT   NAC   NLCK

  1  SY0   000000
  2  CO0   000000
  3  SY0   000000
  4  SY0   000000
  5  TI0   000000
  6  CL0   000000

  HEADER:

   000000      000000  003206  011602  000000  000000  000016  077406  077406
   000020      077406  002261  002270  000000  000000  000402  000602  001002
   000040      177600  000003  000007  000000  000000  000402  000602  001002
   000060      177600  174000  020006  000370  101040  000000  000001  000000
   000100      000406  000100  174017  000106  000100  000000  000000  000000
   000120      101640  000000  000401  000401  000003  000004  000000  000224
   000140      141022  000000  000000  060146  060146  000000  000000  000006
   000160      145006  000000  142444  000000  145006  000000  145006  000000
   000200      000001  000000  142526  000000  000000  000000  000000  000000
   000220      000000  000000  000000  000000  000000  000000  000000  000000
   000240      000000  000000  000000  000000  000000  000000  000000  000000
```

•
•
•

```
RSX-11D CORE-DUMP ANALYZER       V06.2        5-OCT-76    14:25          PAGE 19
TPD SCAN

SYDISK
------

  TPD ENTRY ADDRESS = 145100V/105300R

  BASE ADDRESS = 00120200    SIZE = 00007700    FLAGS = 000003

  ALLOCATION MAP:

      SPACE ALLOCATED       OCCUPIED BY

    00120200-00130100       DB.... (A)

  *****  NO ENTRIES IN MRL  *****

  *****  NO ENTRIES IN CTL  *****


MCR
------

  TPD ENTRY ADDRESS = 145124V/105324R

  BASE ADDRESS = 00130100    SIZE = 00020000    FLAGS = 000000

  HOLES:

      ADDRESS      SIZE

    00132300    00015600

  ALLOCATION MAP:

      SPACE ALLOCATED       OCCUPIED BY

    00130100-00131100       ...MCR (A)
    00131100-00132300       ...MCR (P)
    00132300-00150100

  *****  NO ENTRIES IN MRL  *****

  *****  NO ENTRIES IN CTL  *****


GEN
```

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76    14:25          PAGE 20
TPD SCAN


------

 TPD ENTRY ADDRESS = 145150V/105350R

 BASE ADDRESS = 00150100    SIZE = 01627700    FLAGS = 031014

 HOLES:

     ADDRESS       SIZE

    00236700    00003400
    00244400    00004400
    00323500    00011700
    00436500    01341300

 ALLOCATION MAP:

     SPACE ALLOCATED       OCCUPIED BY

    00150100-00164600       TT.... (A)
    00164600-00203400       SYSRES (L)
    00203400-00210300       TTLIBA (L)
    00210300-00225600       TTLIBB (L)
    00225600-00231400       LP.... (A)
    00231400-00235300       MO.... (A)
    00235300-00236700       ...VFY (A)
    00236700-00242300
    00242300-00244400       USRTSK (A)
    00244400-00251000
    00251000-00254700       DK.... (A)
    00254700-00261500       DT.... (A)
    00261500-00270600       MM.... (A)
    00270600-00274300       DS.... (A)
    00274300-00310400       DTAACP (F)
    00310400-00323500       BPR... (A)
    00323500-00335400
    00335400-00365200       F11ACP (A)
    00365200-00421400       ...LBR (A)
    00421400-00436500       ...DEM (A)
    00436500-02000000


 *****  NO ENTRIES IN MRL  *****

 CTL:

   NODE ADR    TASK     PRI    STS    TI

    105034    ...LBR    50.    W1     TT10

   104760    106540  111440  102420  143656  000462  000001  000000  003652
```

•
•
•

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76     14:25          PAGE 22
STD SCAN

ALPHA TABLE STARTING ADDRESS = 145200V/105400R     NO. OF ENTRIES = 69.


 TASK      PAR      FLAGS     PRI    DISK    SIZE     AV   LIM   UTIL   SRQ LISTHEAD
 ----      ---      -----     ---    ----    ----     --   ---   ----   ------------

ACCABT    GEN      060200    50.   DB0    000010    0.   40.    0.   102642 102642

    102620    003273  003244  145150  060200  014462  000010  000010  024000
    102640    000000  102642  102642  000001  104125  000000  000000  000000

ACCLOG    GEN      060200    235.   DB0    000032    0.   40.    0.   102502 102502

    102460    003273  046537  145150  060200  014753  000032  000032  024000
    102500    000000  102502  102502  000001  104043  142140  000000  000000

ACCOFF    GEN      060200    50.   DB0    000012    0.   40.    0.   102602 102602

    102560    003273  057266  145150  060200  014462  000012  000012  024000
    102600    000000  102602  102602  000001  104114  000000  000000  000000

ACCRPT    GEN      020000    50.   DB0    000167    0.   40.    0.   102542 102542

    102520    003273  071424  145150  020000  014462  000167  000167  024000
    102540    000000  102542  102542  000001  104135  142140  000000  000000

BPR...    GEN      040200    50.   DB0    000131    1.   40.    13.   102122 102122

    102100    007422  131574  145150  040200  014462  000072  000131  024001
    102120    000015  102122  102122  000002  113407  142140  000000  000000

CRF...    GEN      020100    50.   DB0    001067    0.   40.    0.   103602 103602

    103560    012626  131574  145150  020100  014462  000145  001067  024000
    103600    000000  103602  103602  000000  154672  103320  142140  000000

DB....    SYDISK   064200    248.   DB0    000077    1.   40.    9.   142122 142122

    142100    014554  131574  145100  064200  014770  000077  000077  024001
    142120    000011  142122  142122  000001  137665  000000  000005  000000

DK....    GEN      064200    248.   DB0    000037    1.   40.    4.   103442 103442

    103420    015324  131574  145150  064200  014770  000037  000037  024001
    103440    000004  103442  103442  000001  000066  000000  000000  000000
```

CDA SAMPLE LISTING

```
RSX-11D CORE-DUMP ANALYZER       V06.2        5-OCT-76    14:25          PAGE 23
STD SCAN

 TASK      PAR      FLAGS    PRI   DISK   SIZE     AV  LIM  UTIL  SRQ LISTHEAD
 ----      ---      -----    ---   ----   ----     --  ---  ----  ------------

DS....    GEN      064200   248.  DB0   000035    1.  40.   4.  107242 107242

    107220    016024  131574  145150  064200  014770  000035  000035  024001
    107240    000004  107242  107242  000001  137716  000000  000000  000000

DTAACP    GEN      040202   210.  DB0   000141    0. 255.   0.  103042 103042

    103020    016041  003310  145150  040202  014722  000072  000141  177400
    103040    000000  103042  103042  000001  147772  000000  000000  000000

DT....    GEN      064200   248.  DB0   000046    1.  40.   4.  103702 103702

    103660    016074  131574  145150  064200  014770  000046  000046  024001
    103700    000004  103702  103702  000001  141001  000000  000000  000000

ERRLOG    GEN      044200   200.  DB0   000104    0.  40.   0.  104042 104042

    104020    021042  046537  145150  044200  014710  000104  000104  024000
    104040    000000  104042  104042  000001  123515  142140  000000  000000

F11ACP    GEN      040202   220.  DB0   000276    1. 255.   0.  101142 101142

    101120    025167  003310  145150  040202  014734  000210  000276  177401
    101140    000000  101142  101142  000002  052604  000000  000000  000000

F11MSG    GEN      020000   200.  DB0   000053    0.  40.   0.  104142 104142

    104120    025167  052077  145150  020000  014710  000045  000053  024000
    104140    000000  104142  104142  000001  045466  000000  000000  000000

LP....    GEN      064200   248.  DB0   000036    1.  40.   4.  104402 104402

    104360    046634  131574  145150  064200  014770  000036  000036  024001
    104400    000004  104402  104402  000001  141022  000000  000000  000000

MCRERR    GEN      060300    75.  DB0   000006    0.  20.   0.  102342 102342

    102320    050712  021042  145150  060300  014513  000006  000006  012000
    102340    000000  102342  102342  000001  156004  102260  000000  000000
```

•
•
•

```
RSX-11D CORE-DUMP ANALYZER       V06.2         5-OCT-76    14:25         PAGE 31
PUD SCAN


TT0
----

  FLAGS    ATTACH    RED     HANDLER    EXP CNT    CHAR      DACP      ACP       TERM

  000210             TT0      TT....       0.     000200                        000005

    142300     052124  104000  000007  000000  000000  001000  000000  142300
    142320     100260  000000  002756  142324  000060  000200  043140  000000
    142340     000401  000200  000000  000000  000000  000005  000000  000000
    142360     000000

  I/O REQUEST NODES:

      REQUESTOR    ATL NODE    DPB SZ    PRI    LUN    EFN    FUNC CODE    REQ TYPE

        USRTSK       110200       8.      50.    1.     0.      IO.RLB       NORMAL

    110720     060146  060146  105520  110200  004062  000001  001000  000000
    110740     000000  000401  001336  000120  000000  000401  000400  000000
    110760     143740  000000  000000  000000  000001  000001  000000  000000



CI0
----

  FLAGS    ATTACH    RED     HANDLER    EXP CNT    CHAR      DACP      ACP       TERM

  000000             TT0                   0.     000000                        000000

    142362     044503  000000  000000  000000  000000  000000  000000  142300
    142402     000000  000000  000000  000000  000000  000000  000000  000000
    142422     000000  000000  000000  000000  000000  000000  000000  000000
    142442     000000

        *****  I/O  REQUEST  QUEUE  EMPTY  *****


CO0
----

  FLAGS    ATTACH    RED     HANDLER    EXP CNT    CHAR      DACP      ACP       TERM

  000000             TT0                   0.     000000                        000000

    142444     047503  000000  000000  000000  000000  000000  000000  142300
    142464     000000  000000  000000  000000  000000  000000  000000  000000
```

```
RSX-11D CORE-DUMP ANALYZER       V06.2        5-OCT-76    14:25         PAGE 32
PUD SCAN

    142504    000000  000000  000000  000000  000000  000000  000000  000000
    142524    000000

        ***** I/O REQUEST QUEUE EMPTY *****


CL0
----

  FLAGS   ATTACH   RED    HANDLER   EXP CNT   CHAR    DACP    ACP     TERM

  000000           LP0                 0.    000000                  000000

    142526    046103  000000  000000  000000  000000  000000  000000  143264
    142546    000000  000000  000000  000000  000000  000000  000000  000000
    142566    000000  000000  000000  000000  000000  000000  000000  000000
    142606    000000

        ***** I/O REQUEST QUEUE EMPTY *****


BP0
----

  FLAGS   ATTACH   RED    HANDLER   EXP CNT   CHAR    DACP    ACP     TERM

  000000           TT4                 0.    000200                  000001

    142610    050102  000000  004003  000000  000000  000000  000000  144166
    142630    000000  000000  142634  142634  000000  000000  000000  000000
    142650    000401  000200  000000  000000  000000  000001  000000  000000
    142670    000000

        ***** I/O REQUEST QUEUE EMPTY *****


SP0
----

  FLAGS   ATTACH   RED    HANDLER   EXP CNT   CHAR    DACP    ACP     TERM

  000000           SY0                 0.    000200                  000000

    142672    050123  000000  000000  000000  000000  000000  000000  145006
    142712    000000  000000  142716  142716  000000  000000  000000  000000
```

•
•
•

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76    14:25          PAGE 43
MCR COMMAND BUFFERS


NODE ADDRESS    TASK     TI    LENGTH    FIRST 40 CHARACTERS IN BUFFER
------------    ----     --    ------    -----------------------------

   111060      ...LBR   TT10    24.      LBR CDAOLB,LP:/LI=CDACKQ



RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76    14:25          PAGE 44
BATCH COMMAND BUFFERS


NODE ADDRESS    TASK     TI    LENGTH    FIRST 40 CHARACTERS IN BUFFER
------------    ----     --    ------    -----------------------------

   111720      ...VFY   BP0     10.      VFY SY:/LO



RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76    14:25          PAGE 45
FTL SCAN


DTAACP
------
  FTL ADDRESS = 107440V/047640R     STD ADDRESS = 103020V/043220R
  LOAD ADDRESS = 00274300
  PAR=GEN        PRI=210,    I/O IN PROG=0,    I/O PEND=0,    STATUS=LS    TI=TT7
  FLAGS WORD=002000   FLAGS=000000 000000    MASKS=144642 006401 000000 000000

    107440     146404  146404  102360  143740  000322  000000  000000  002743
    107460     000004  103020  000000  000000  144642  006401  000000  000000
    107500     145150  107502  107502  000001  147772  002000  107514  107514



RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76    14:25          PAGE 46
AST QUEUE SCAN


  TASK = USRTSK    ATL ADDRESS = 110200V/050400R    STD ADDRESS = 105520V/050422R

    TYPE = MARK TIME          ENTRY POINT = 001000
    ADDITIONAL INFORMATION:    000000

    110440     110500  110242  105520  000400  001000  000000  000000  000000
    110460     000000  000000  001000  042300  176740  000013  000323  000001

    TYPE = I/O DONE           ENTRY POINT = 001330
    EVENT FLAG = 0,     IOSB ADR = 000000     IOSB CONTENTS = 006401   000003

    110500     110340  110440  105520  000401  004062  000001  001000  000000
    110520     001330  000401  020041  000120  006401  000003  000000  000000
    110540     143740  000000  002000  002441  000120  001000  000000  000001

    TYPE = TERMINAL           ENTRY POINT = 001330
    NO ADDITIONAL INFORMATION

    110340     110242  110500  105520  000010  001330  000000  001000  000000
    110360     110260  000000  000171  130100  177400  005422  000443  000000
```

```
RSX-11D CORE-DUMP ANALYZER        V06.2         5-OCT-76     14:25           PAGE 47
SRQ SCAN

RECEIVER  SENDER  TI   PRI                    DATA  BLOCK
--------  ------  --   ---                    -----------

 SPR...   ...QUE  TT2  100.    050114  031000  040001  041104  000000
                               000172  000001  000000  004477  052140
                               000000  047014  000003



RSX-11D CORE-DUMP ANALYZER        V06.2         5-OCT-76     14:25           PAGE 48
CLOCK QUEUE SCAN

  TYPE = TIME-SLICE       DELTA = 1. TICKS

    106060     104320  146400  142100  145600  000000  000001  000400  145150
    106100     000000  106102  106102  000000  153252  142140  000000  000000

  TYPE = MARK TIME        DELTA = 22. TICKS
  TASK = ...DEM         STD ADDRESS = 103360V/044524R
  ATL ADDRESS = 111360V/044526R     EVENT FLAG = 2.      AST ADDRESS = 000000

    104320     100640  106060  103360  111360  000000  000026  000000  000002
    104340     111404  000002  000000  000000  164434  142140  000000  000000

  TYPE = MARK TIME        DELTA = 8. TICKS
  TASK = TT....         STD ADDRESS = 141740V/041044R
  ATL ADDRESS = 100260V/041046R     EVENT FLAG = 6.      AST ADDRESS = 000000

    100640     110400  104320  141740  100260  000000  000010  000000  000040
    100660     100304  000006  000000  000000  000000  000000  000000  000000

  TYPE = MARK TIME        DELTA = 22595. TICKS
  TASK = BPR...         STD ADDRESS = 102100V/050604R
  ATL ADDRESS = 111440V/050606R     EVENT FLAG = 0.      AST ADDRESS = 002124

    110400     106320  100640  102100  111440  000000  054103  000000  000000
    110420     000000  000000  002124  000000  000000  000000  000000  000000

  TYPE = MARK TIME        DELTA = 509673. TICKS
  TASK = USRTSK         STD ADDRESS = 105520V/046524R
  ATL ADDRESS = 110200V/046526R     EVENT FLAG = 1.      AST ADDRESS = 001000

    106320     146400  110400  105520  110200  000007  143351  000000  000001
    106340     110224  000001  001000  014402  000000  000000  000000  127030
```

```
RSX-11D CORE-DUMP ANALYZER       V06.2       5-OCT-76    14:25         PAGE 49
GCD SCAN

 NAME    ADDRESS    SIZE    PAR      FLAGS  STS  ACT/INS   CREATED      UIC
 ----    -------    ----    ---      -----  ---  -------   -------      ---

TTLIBB  002103   000153  GEN     000002  LS    1./1.    4-OCT-76   [1,1]

    142240    142200  146356  100054  034222  002103  000153  000114  002012
    142260    001404  000401  145150  014402  000401  000000  000002  027451

TTLIBA  002034   000047  GEN     000002  LS    1./1.    4-OCT-76   [1,1]

    142200    142140  142240  100054  034221  002034  000047  000114  002012
    142220    001004  000401  145150  014402  000401  000000  000002  024645

SYSRES  001646   000166  GEN     000026  LS    4./28.   28-SEP-76   [1,1]

    142140    146356  142200  075273  070533  001646  000166  000114  016011
    142160    000004  000401  145150  014426  016004  000000  000000  110006


         ***** THE FOLLOWING ARE GCD NODES FOR PURE AREAS *****

CRF...  000000   000001  GEN     000002  NL    0./1.   28-SEP-76   [1,1]

    103320    000000  000000  000035  000000  000000  000001  000114  016011
    103340    000400  000401  145150  014402  000400  000000  000000  154707

MCRERR  000000   000023  GEN     000002  NL    0./1.   30-SEP-76   [1,1]

    102260    000000  000000  000035  000000  000000  000023  000114  017011
    102300    000400  000401  145150  014402  000400  000000  000001  156005

...ACT  000000   000007  GEN     000002  NL    0./1.   30-SEP-76   [2,2]

    102660    000000  000000  000035  000000  000000  000007  000114  017011
    102700    001000  001002  145150  014402  000400  000000  000001  155524

...DMP  000000   000106  GEN     000002  NL    0./1.   28-SEP-76   [0,0]

    103520    000000  000000  000035  000000  000000  000106  000114  016011
    103540    001000  000000  145150  014402  000400  000000  000000  126572

...EDI  000000   000142  GEN     000002  NL    0./1.    5-OCT-76   [0,0]

    101440    000000  000000  000035  000000  000000  000142  000114  002412
    101460    000400  000000  145150  014402  000400  000000  000001  151641
```

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76    14:25          PAGE 50
GCD SCAN


...MAC  000000    000130   GEN       000002   NL      0./1.     5-OCT-76   [0,0]

    101340    000000   000000   000035   000000   000000   000130   000114   002412
    101360    000400   000000   145150   014402   000400   000000   000000   103273

...MCR  001311    000012   MCR       000002   LS      1./1.    30-SEP-76   [1,1]

    102160    000000   000000   000035   000000   001311   000012   000114   017011
    102200    000404   000401   145124   014402   000401   000000   000001   156053

...MFT  000000    000023   MCR       000002   NL      0./1.    30-SEP-76   [2,2]

    102220    000000   000000   000035   000000   000000   000023   000114   017011
    102240    000400   001002   145124   014402   000400   000000   000001   156105

...PIP  000000    000063   GEN       000002   NL      0./1.    28-SEP-76   [0,0]

    104620    000000   000000   000035   000000   000000   000063   000114   016011
    104640    000400   000000   145150   014402   000400   000000   000000   126107

...PRE  000000    000215   GEN       000002   NL      0./1.    29-SEP-76   [1,1]

    104720    000000   000000   000035   000000   000000   000215   000114   016411
    104740    002000   000401   145150   014402   000400   000000   000001   131676

...QUE  000000    000050   GEN       000002   NL      0./1.    28-SEP-76   [200,200]

    105160    000000   000000   000035   000000   000000   000050   000114   016011
    105200    000400   100200   145150   014402   000400   000000   000000   167001
```

```
RSX-11D CORE-DUMP ANALYZER       V06.2        5-OCT-76    14:25         PAGE 51
POOL VERIFICATION


CONSISTENCY CHECK OF POOL BITMAP AND SYSTEM LISTS:

    BITMAP INDICATES 347. NODES IN USE OUT OF 1112.

    LISTS  INDICATE  307. NODES IN USE


                    ************  IMPORTANT NOTE  ************
                    *                                        *
                    *  THERE  WILL  ALWAYS  BE  SOME  NODES  *
                    *  THAT ARE ALLOCATED IN THE BITMAP BUT  *
                    *  NOT FOUND IN THE  LISTS,  SINCE  ANY  *
                    *  PRIVILEGED  TASK  CAN  PICK   NODES.  *
                    *                                        *
                    ******************************************


NODES ALLOCATED IN BITMAP BUT NOT FOUND IN LISTS:

    100140    100140  100140  100144  100144  100150  100150  100154  100154

    100160    100160  100160  100164  100164  100170  100170  100174  100174

    100200    100200  100200  000000  001270  000000  000000  000000  000000

    100220    000000  000000  000000  000000  000000  000000  000000  000000

    100240    000000  000000  000000  000000  000000  000000  000000  000000

    107100    000000  000647  000000  000000  000000  000000  000000  000000

    107120    000000  000000  000000  000000  000000  000000  000000  000000

    107140    000000  000000  000000  000000  000000  000000  000000  000000

    107320    000000  000000  000000  000000  000000  000000  000000  000000

    107340    000000  000000  000000  000000  000000  000000  000000  000000

    107560    000000  000000  000000  000000  000000  000000  000000  000000

    107600    000000  000000  000000  000000  000000  000000  000000  000000

    110040    110040  110040  110044  110044  000000  011022  000000  000000

    110060    142000  031560  000000  000000  000000  000000  000000  000000

    110100    000000  000000  000000  000000  000000  000000  000000  000000

    110260    060146  110720  110120  104760  006062  016021  001030  026542
```

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76     14:25          PAGE 52
POOL VERIFICATION

    110300    000000  000401  020046  000001  000000  000000  000000  000137
    110320    143656  000000  002000  004142  000001  011022  000000  110334
    110560    060146  060146  105520  110200  004062  000001  001000  000000
    110600    000000  000401  020036  000120  000000  000000  000000  000000
    110620    143740  000000  002000  002441  000120  000000  000000  000000
    110640    110720  060146  105520  110200  004062  000001  001000  000000
    110660    000000  000401  001336  000120  000000  000000  000000  000000
    110700    143740  000000  000000  122164  000001  001000  000000  000001
    111300    110720  060146  102420  111000  005746  000401  004600  000370
    111320    000000  000401  020074  000120  000037  021112  000006  000000
    111340    144332  000000  002000  000522  000120  011022  000000  145170
    111560    000000  000000  000000  000000  000000  000000  000000  000000
    111600    000000  000000  000000  000000  000000  000000  000000  000000
    111660    000000  000000  000000  000000  000000  000000  000000  000000
    111700    000000  000000  000000  000000  000000  000000  000000  000000
    112060    112220  027420  102420  000000  144332  000000  001000  000000
    112100    111000  000000  000171  130100  177400  005422  000443  000000
    112120    144642  000000  004000  000016  000000  011022  000000  000000
    112140    142000  027500  103360  000401  000000  000401  000410  000000
    112160    000000  000401  020221  000000  000001  000131  000000  000000
    112200    143574  000000  002400  004227  000131  000044  000000  112214
    146020    000000  000000  004767  166244  012600  000207  004767  165316
    146040    005002  012700  007552  004767  165624  016702  156022  004767
    146060    177666  005046  016705  156006  072527  000006  012701  001340


NODES FOUND IN LISTS BUT NOT ALLOCATED IN BITMAP:
```

•
•
•

```
RSX-11D CORE-DUMP ANALYZER        V06.2          5-OCT-76    14:25          PAGE 54
NODE POOL DUMP

                     S Y S T E M    N O D E    P O O L

* = NODE ALLOCATED

   VIRT     REAL                 CONTENTS                  ASCII          RAD50

* 100000   040200     145642  145642  142100  000003   "K"K@D      2WJ 2WJ 1P     C
* 100010   040210     003212  000000  000000  000000               AA4
* 100020   040220     000000  000000  000000  000000
* 100030   040230     000000  000000  000000  000000
* 100040   040240     100100  041540  142100  013746   @  C@D      TT2 J1H 1P  C28
* 100050   040250     172346  042737  004000  100104     _E  D     9GN KGG AKH TT6
* 100060   040260     012737  001205  172346  000137   _      _    CS9  PE 9GN  BO
* 100070   040270     064102  000000  000000  000000   B           PSJ
* 100100   040300     000000  000000  000000  000000
* 100110   040310     000000  000000  000000  000000
* 100120   040320     000000  000000  000000  000000
* 100130   040330     000000  000000  000000  000000
* 100140   040340     100140  100140  100144  100144               TUX TUX TU. TU.
* 100150   040350     100150  100150  100154  100154               TU2 TU2 TU6 TU6
* 100160   040360     100160  100160  100164  100164               TV  TV  TVD TVD
* 100170   040370     100170  100170  100174  100174               TVH TVH TVL TVL
* 100200   040400     100200  100200  000000  001270          8    TVP TVP      QP
* 100210   040410     000000  000000  000000  000000
* 100220   040420     000000  000000  000000  000000
* 100230   040430     000000  000000  000000  000000
* 100240   040440     000000  000000  000000  000000
* 100250   040450     000000  000000  000000  000000
* 100260   040460     101040  145600  141740  144332      K CZH    T22 2VP 1MX 2ER
* 100270   040470     000370  000000  000001  001501          A     FH       A  T3
* 100300   040500     000016  141740  000000  040000     C    @     N 1MX      JIX
* 100310   040510     007377  000000  000000  000000               BO9
* 100320   040520     145150  100322  100322  000002    JR R       2OP TXR TXR   B
* 100330   040530     027733  000000  100334  100334   [/   \ \    G<K     TX. TX.
* 100340   040540     100400  000240  141740  013746         C     TYX  D  1MX C28
* 100350   040550     172346  042737  010000  100100      _E  @    9GN KGG BVP TT2
* 100360   040560     012737  002103  172346  000137   _ C    _    CS9  $K 9GN  BO
* 100370   040570     060174  004040  000526  000000       V       OQT AL   HV
* 100400   040600     100440  000200  141740  013746         C     TZP  CH 1MX C28
* 100410   040610     172346  042737  020000  100100      _E  @    9GN KGG ED2 TT2
* 100420   040620     012737  002103  172346  000137   _ C    _    CS9  $K 9GN  BO
* 100430   040630     060446  000000  001000  000000   &           OU0      L2
* 100440   040640     100500  000140  141740  013746   @     C     T$H  BP 1MX C28
* 100450   040650     172346  042737  000400  100106      _E  F    9GN KGG  FP TT8
* 100460   040660     012737  002103  172346  000261   _ C    1    CS9  $K 9GN  DQ
* 100470   040670     000137  060174  000000  000000   _            BO OQT
* 100500   040700     100540  000100  141740  013746      @  C     T.   AX 1MX C28
* 100510   040710     172346  042737  001000  100106      _E  F    9GN KGG  L2 TT8
* 100520   040720     012737  002103  172346  000261   _ C    1    CS9  $K 9GN  DQ
* 100530   040730     000137  060446  000526  000001   _ & V        BO OU0  HV   A
* 100540   040740     100600  000040  141740  013746         C     T.2   2 1MX C28
* 100550   040750     172346  042737  000001  100106      _E  F    9GN KGG   A TT8
* 100560   040760     012737  002103  172346  000137   _ C    _    CS9  $K 9GN  BO
```

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76    14:25          PAGE 55
NODE POOL DUMP

* 100570  040770   060302 000000 001000 000000  B              OSJ      L2
* 100600  041000   100700 000040 141740 013746  @     C        T0P    2 1MX C28
* 100610  041010   172346 042737 000002 100106    _E   F       9GN KGG    B TT8
* 100620  041020   012737 002103 172346 000137  _ C    _       CS9  SK 9GN  BO
* 100630  041030   060502 011022 000000 000000  B              OVR B5Z
* 100640  041040   110400 104320 141740 100260    P   C0       WH  U5X 1MX TWX
* 100650  041050   000000 000010 000000 000040                       H       2
* 100660  041060   100304 000006 000000 000000  D              TXD   F
* 100670  041070   000000 000000 000000 000000
* 100700  041100   100740 040700 141740 013746     @A C        T1H JT2 1MX C28
* 100710  041110   172346 042737 000004 100106    _E   F       9GN KGG    D TT8
* 100720  041120   012737 002103 172346 000261  _ C    1       CS9  SK 9GN  DQ
* 100730  041130   000137 060302 000000 000000  _ B             BO OSJ
* 100740  041140   101000 040640 141740 013746      A C        T2  JT  1MX C28
* 100750  041150   172346 042737 000010 100106    _E   F       9GN KGG    H TT8
* 100760  041160   012737 002103 172346 000261  _ C    1       CS9  SK 9GN  DQ
* 100770  041170   000137 060502 000000 000000  _ B             BO OVR
* 101000  041200   100322 100322 141740 000003  R R   C        TXR TXR 1MX   C
* 101010  041210   002616 000000 000000 000000                  5V
* 101020  041220   000000 000000 000000 000000
* 101030  041230   000000 000000 000000 000000
* 101040  041240   106460 100260 102360 142300  0 0     @D     VWX TWX UJ2 1SH
* 101050  041250   000370 000000 000000 002256          .       FH         /8
* 101060  041260   000016 104360 000000 000000                    N U6P
* 101070  041270   000003 006401 000000 000000                    C BCI
* 101100  041300   145150 101102 101102 000001   JB B          20P T3Z T3Z   A
* 101110  041310   141022 000000 101114 101114   B  L L        1BB     T36 T36
* 101120  041320   025167 003310 145150 040202   *H   J @      F11 ACP 20P JL4
* 101130  041330   014734 000210 000276 177401  \   >          DET  CP  D0  2A
* 101140  041340   000000 101142 101142 000002                     T4R T4R   B
* 101150  041350   052604 000000 000000 000000   U             MSL
  101160  041360   101500 000020 142100 142300  @    @D@D      U     P 1P  1SH
  101170  041370   001772 111220 144642 010000       "I         YR WR  2JR BVP
* 101200  041400   131053 077514 145150 040200  +2L   J @      .TK TN. 20P JL2
* 101210  041410   014771 000031 000031 024000          (      DFI   Y   Y FP
* 101220  041420   000000 101222 101222 000001                     T5Z T5Z   A
* 101230  041430   156260 000000 000000 000000  0\             5LP
* 101240  041440   131574 073376 145150 040200   3    J @      ... SAV 20P JL2
* 101250  041450   014735 000151 000151 024000  ]       (      DEU  BY  BY FP
* 101260  041460   000000 101262 101262 000001    2 2              T6R T6R   A
* 101270  041470   136775 000000 000000 000000   .             007
* 101300  041500   113000 000000 145150 040200        J @      X       20P JL2
* 101310  041510   014734 000210 000276 177400  \   >          DET  CP  D0  2
* 101320  041520   000000 101322 101322 000002    R R              T7J T7J   B
* 101330  041530   052604 000000 000000 000000   U             MSL
* 101340  041540   000000 000000 000035 000000                          /
* 101350  041550   000000 000130 000114 002412    X L               BH  A6  2J
* 101360  041560   000400 000000 145150 014402        J         FP     20P D B
* 101370  041570   000400 000000 000000 103273         ;        FP         UVK
* 101400  041600   131574 050553 145150 020100   3 Q J@        ... MAC 20P EFP
* 101410  041610   014462 000131 001166 024000  2 Y     (      DAJ  BI  00 FP
* 101420  041620   000000 101422 101422 000000                     T84 T84
* 101430  041630   103257 101340 142140 000000  /     D        UU9 T7X 1P2
* 101440  041640   000000 000000 000035 000000                          /
```

•
•
•

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76     14:25          PAGE 100
TASK DUMP


TASK = ...VFY
=============


    000000    000002  000000  000000  000000                        B
    000010    000000  000000  000000  000000
    000020    000000  000000  000000  000000
    000030    000000  000000  000000  000000
    000040    000000  000000  000000  000000
    000050    000000  000000  000000  000000
    000060    000000  000000  000000  000000
    000070    000000  000000  000000  000000
    000100    000000  000000  000000  000000
    000110    000000  000000  000000  000000
    000120    000000  000000  000000  000000
    000130    000000  000000  000000  000000
    000140    000000  000000  000000  000000
    000150    000000  000000  000000  000000
    000160    000000  000000  000000  000000
    000170    000000  000000  000000  000000
    000200    000000  000000  000000  000000
    000210    000000  000000  000000  000000
    000220    000000  000000  000000  000000
    000230    000000  000000  000000  000000
    000240    000000  000000  000000  000000
    000250    000000  000000  000000  000000
    000260    000000  000000  000000  000000
    000270    000000  000000  000000  000000
    000300    000000  000000  000000  000000
    000310    000000  000000  000000  000000
    000320    000000  000000  000000  000000
    000330    000000  000000  000000  000000
    000340    000000  000000  000000  000000
    000350    000000  000000  000000  000000
    000360    000000  000000  000000  000000
    000370    000000  000000  000000  000000
    000400    000000  000000  000000  000000
    000410    000000  000000  000000  000000
    000420    000000  000000  000000  000000
    000430    000000  000000  000000  000000
    000440    000000  000000  000000  000000
    000450    000000  000000  000000  000000
    000460    000000  000000  000000  000000
    000470    000000  000000  000000  000000
    000500    000000  000000  000000  000000
    000510    000000  000000  000000  000000
    000520    000000  000000  000000  000000
    000530    000000  000000  000000  000000
    000540    000000  000000  000000  000000
    000550    000000  000000  000000  000000
    000560    000000  000000  000000  000000
    000570    000000  000000  000000  000000
    000600    000000  000000  000000  000000
    000610    000000  000000  000000  000000
```

CDA SAMPLE LISTING

```
RSX-11D CORE-DUMP ANALYZER        V06.2        5-OCT-76     14:25          PAGE 101
TASK DUMP

    000620    000000  000000  000000  000000
    000630    000000  000000  000000  000000
    000640    000000  000000  000000  000000
    000650    000000  000000  000000  000000
    000660    000000  000000  000000  000000
    000670    000000  000000  000000  000000
    000700    000000  000000  000000  000000
    000710    000000  000000  000000  000000
    000720    000000  000000  000000  000000
    000730    000000  000000  000000  000000
    000740    000000  000000  000000  000000
    000750    000000  000000  000000  000000
    000760    000000  000000  000000  000000
    000770    000000  000000  000000  000455          -                   GU
    001000    012746  000455  104377  000000      -           CTF   GU U61
    001010    000000  000000  000000  000000
    001020    000000  000000  000000  000000
    001030    000000  000000  000000  000000
    001040    000000  000000  000000  000000
    001050    000000  000000  000000  000000
    001060    000000  000000  000000  000000
    001070    000000  000000  000000  000000
    001100    002444  000034  000000  000000    $              26    .
    001110    077406  000016  077406  077406                  TL8    N TL8 TL8
    001120    077406  002372  000000  000000                  TL8   14
    001130    004365  000402  000602  001002                  AQM   FR   IZ  L4
    001140    177600  000003  000000  000000                   5H    C
    001150    000000  000402  000602  001002                        FR   IZ  L4
    001160    177600  174000  000656  111000        .          5H  9SA   J0 WNP
    001170    000000  000064  000000  001277      4   ?              AL      QW
    001200    060104  000100  174017  000100    D .   .       OPD   AX 9SW  AX
    001210    000100  000000  000000  000000    .              AX
    001220    000000  000000  001002  000401                             L4  FQ
    001230    000003  000004  000000  000026                    C    D        V
    001240    156260  000000  000000  060146    0\            5LP            OP8
    001250    060146  000000  000000  000006                  OP8              F
    001260    142444  000000  000001  000000    $E            1U.          A
    001270    145006  000000  145006  000000     J   J        2L8         2L8
    001300    000001  000000  142526  000000        VE          A         1V8
    001310    000000  000000  000000  000000
    001320    000000  000000  000000  000000
    001330    000000  000000  000000  000000
    001340    000000  000000  000000  000000
    001350    000000  000000  000000  000000
    001360    000000  000000  000000  000000
    001370    000000  000000  000000  000000
    001400    000001  000000  000000  000000                    A
```

SGA DUMP

```
SGA = SYSRES
------------

000000   017646 000000 032736 010000   &   ^5     EBV     HX0 BVP
000010   001003 062716 000002 013607     NE        L5 PKV   B C0W
000020   004567 016320 010246 016605   W P &      ATW DXP BZV D.7
000030   000012 012502 011566 000012     B V        J CPB CDV   J
000040   004767 000224 005005 005004   W          AW1  C. AXE AXD
000050   005003 005762 000004 001004     R        AXC A6R   D  L6
000060   042762 010000 000000 000403   RE         KGZ BVP      FS
000070   004767 000300 103412 016202   W @        AW1  D2 UXJ DVR
000100   000010 001405 032762 010000       R5       H  SM HYJ BVP
000110   000000 001354 000770 012602     L X           R.  LX CQZ
000120   000207 013703 000006 004773     C   [     CO C2C   F AW5
000130   000010 000744 122710 000003     D H%       H  LD ZU    C
000140   000261 003051 004567 016174   1 ) W \     DQ  9Q ATW DVL
000150   010246 013702 000006 016202   & B        BZV C2B   F DVR
000160   000002 005712 000261 001436     J 1        B A5R  DQ  S8
000170   026260 000014 000002 001004   0,         GF    L   B  L6
000200   026260 000016 000004 001403   0,         GF    N   D  SK
000210   062702 000020 000762 005005   BE  R      PKJ   P  LR AXE
000220   005004 121027 000004 103407      "       AXD Y7O   D UXG
000230   016004 000006 121027 000005        "     DSL   F Y7O   E
000240   103402 016005 000010 005003              UXB DSM   H AXC
000250   156003 000001 004767 000010    \  W      5HC   A AW1   H
000260   004767 000110 012602 000207   W H        AW1  A2 CQZ  CO
000270   010246 032762 010000 000000   & R5       BZV HYJ BVP
000300   001005 016202 000010 001371         Y     L7 DVR   H  SA
000310   012602 000207 005762 000010       R      CQZ  CO A6R   H
000320   001773 010246 010246 016202   [ & &       YS BZV BZV DVR
000330   000012 020216 001414 032762         R5     J EHN  ST HYJ
000340   010000 000000 001370 052762       X RU   BVP       S M0B
000350   010000 000000 016216 000006              BVP     DV0   F
000360   001402 011602 000764 005726       T V     SJ CD4  LT A50
000370   012602 000743 005046 016246     C & &    CQZ  LC AX8 DWN
000400   000000 042716 170000 005046     NE P&        KF0 8P  AX8
000410   005046 016246 000004 016246   & &   &    AX8 DWN   D DWN
000420   000002 010546 013705 000006     F E        B B1N C2E   F
000430   010446 001006 105703 001004   &   C      B/0  L8 VNS  L6
000440   010504 062704 000004 010416   D DE       B0T PKL   D B/F
000450   005046 150316 001002 012716   & NP  N    AX8 3PN  L4 CSV
000460   000037 016546 000000 012746     F   F      1 D.F     CTF
000470   001010 012746 006001 104377     F   ▲      M CTF A63 U61
000500   103420 042762 010000 000000     RE       UXP KGZ BVP
000510   105703 001013 012746 000037   C   F      VNS  MC CTF   1
000520   012746 001051 104377 105714   F ) ▲ L    CTF  M3 U61 VN.
000530   100004 052762 010000 000000     RU       TSL M0B BVP
000540   000261 000207 004567 005244   1   W $     DQ  CO ATW A.D
000550   012760 000001 000052 016001   P   *      CTP   A  AB DSI
000560   000070 001002 000167 000612   8   W       AP  L4  B9  I4
000570   132760 000001 000016 001063   P5    3    /DP   A   N  NC
000600   132760 000001 000051 001432   P5  )      /DP   A  AA  S4
000610   004767 006020 132760 000040   W   P5     AW1 A7H /DP   2
```

```
RSX-11D CORE-DUMP ANALYZER        V06.2         5-OCT-76      14:25          PAGE 103
SGA DUMP

    000620    000017  001416  122760  000001         P%         O   SV  ZV    A
    000630    000000  001407  004767  005656         W .           SO  AW1  A40
    000640    103007  012770  177777  000030      X __        UQ1 CTX   80    X
    000650    000403  016002  000030  160102            B@     FS DSJ    X  55J
    000660    004767  015012  016001  000070    W       8     AW1 DFZ  DSI   AP
    000670    004767  011532  116002  000052    W Z     *     AW1 CC4  X8R   AB
    000700    005005  156005  000055  000412       \-         AXE 5HE   AE   FZ
    000710    004767  011512  103007  005702    W J     B     AW1 CCR  UQ1  A5J
    000720    002405  132760  000002  000056      P5    .      2E /DP    B   AF
    000730    001401  111102  016101  000012      B A          SI WPB  DT3    J
    000740    077515  110260  000052  116046    M_0 *   &     TN/ WF    AB  X9N
    000750    000052  004767  007252  005025    * W *          AB AW1  BM4  AXU
    000760    010604  005046  132760  000010      & P5        B2D AX8  /DP    H
    000770    000043  001426  132760  000041    #    P5|       5  S0  /DP    3
    001000    000017  001022  132760  000020         P5         O  MJ  /DP    P
    001010    000051  001405  010046  062716    )     & NE     AA  SM  BWN  PKV
    001020    000120  012746  001007  013702    P F     B      B  CTF   L9  C2B
    001030    000002  062702  000010  005712      BE    J      B  PKJ    H  A5R
    001040    001403  010246  012746  001001      & F          SK BZV  CTF   L3
    001050    132760  000001  000043  001003    P5    #       /DP   A    5   L5
    001060    010046  012746  007004  010625    & F           BWN CTF  BI.  B2U
    001070    010446  012704  010000  005760    & D     P     B/0 CSL  BVP  A6P
    001100    000076  100006  062705  000004    >     EE       AV TSN  PKM    D
    001110    016015  000076  042715  100000      >   ME      DSU  AV  KF/  TSH
    001120    004767  011262  012606  112601    W 2           AW1 B9Z  CQ0  W63
    001130    003002  110160  000052  132760      P * P5       8R WDP   AB  /DP
    001140    000200  000051  001516  162706      ) N FE       CH  AA   UF  60V
    001150    000040  010602  010246  012746        & F         2 B2B  BZV  CTF
    001160    001077  104377  016203  000014    ? _            NO U61  DVS    L
    001170    062706  000040  012746  131574    FE    F \3    PKN   2  CTF  ...
    001200    012746  074522  010601  010002    F RY          CTF SPR  B2A  BVR
    001210    062702  000102  016246  000016    BEB &         PKJ  AZ  DWN    N
    001220    016246  000014  016246  000012    &     &       DWN   L  DWN    J
    001230    016246  000010  016246  000006    &     &       DWN   H  DWN    F
    001240    016246  000030  016246  000026    &     &       DWN   X  DWN    V
    001250    016246  000024  016246  000034    &     &       DWN   T  DWN    .
    001260    016246  000032  012746  000001    &     F       DWN   Z  CTF    A
    001270    116046  000074  110366  000001    & < V         X9N  AT  WG0    A
    001300    016046  000072  010602  005046    & !     &     DIF  AR  B2B  AX8
    001310    010246  005046  005046  005046    & & & &       BZV AX8  AX8  AX8
    001320    005046  016146  000002  011146    & F     F     AX8 DU0    B  B70
    001330    012746  004511  104377  062706    F I   _ FE    CTF ASQ  U61  PKN
    001340    000036  116001  000042  016002          W         0 X8Q    4  DSJ
    001350    000072  116003  000074  010346    !     < F      AR X8S   AT  B.F
    001360    010246  010146  012746  002007    & F F         BZV BX8  CTF   YI
    001370    104377  005060  000072  105060    _ 0 : 0       U61 AYH   AR  VDP
    001400    000074  004767  011142  000167    < W B W        AT AW1  B7Z   B9
    001410    006226  004567  004376  004767      W A W       B V ATW  AQV  AW1
    001420    007452  103443  132760  000001    * # P5        BQB UX5  /DP    A
    001430    000017  001441  132760  000010      | P5          O  TA  /DP    H
    001440    000051  001404  112760  177766    )     P V_     AA  SL  W9X   8F
    001450    000052  000427  004767  011360    *     W P      AB  F9  AW1  CAH
    001460    016001  000070  016103  000002      8 C         DSI  AP  DT6    B
    001470    016001  000030  004767  000402        W         DSI   X  AW1   FR
```

•
•
•

APPENDIX B

TKB15.CMD FILE

See Section 6.3.1

```
;
;       [ 1 1 , 1 5 ] T K B 1 5 . C M D
;
;
;  INDIRECT TKB COMMAND FILE FOR THE RSX-11D EXEC
;
;
[11,17]EXEC/-HD/-FP,[11,15]EXEC/CR/-SP,[1,1]EXEC.STB=
;
; IF A MAP IS NOT REQUIRED PUT A ';' AT THE BEGINNING OF THE ABOVE LINE
; AND REMOVE THE ';' FROM THE BEGINNING OF THE FOLLOWING LINE
;
;[11,17]EXEC/-HD/-FP,,[1,1]EXEC.STB=
;
[11,15]EXEC/LB:EM00
[11,15]EXEC/LB:EM01
[11,15]EXEC/LB:EM02
[11,15]EXEC/LB:EM03
[11,15]EXEC/LB:EM04
[11,15]EXEC/LB:EM05
[11,15]EXEC/LB:EM06
[11,15]EXEC/LB:EM07
[11,15]EXEC/LB:EM08
[11,15]EXEC/LB:EM09
[11,15]EXEC/LB:EM10
[11,15]EXEC/LB:ACCTET
[11,15]EXEC/LB:DMGCL
[11,15]EXEC/LB:DMQIO
[11,15]EXEC/LB:DMGLI
[11,15]EXEC/LB:DMASS
[11,15]EXEC/LB:DMALP
[11,15]EXEC/LB:DMREQ
[11,15]EXEC/LB:DMSCH
[11,15]EXEC/LB:DMMKT
[11,15]EXEC/LB:DMCSR
[11,15]EXEC/LB:DMCMT
[11,15]EXEC/LB:DMSED
[11,15]EXEC/LB:DMSUS
[11,15]EXEC/LB:DMEXT
[11,15]EXEC/LB:DMGTP
[11,15]EXEC/LB:DMGPP
[11,15]EXEC/LB:DMGMP
[11,15]EXEC/LB:DMGCP
[11,15]EXEC/LB:DMSAR
[11,15]EXEC/LB:DMABO
[11,15]EXEC/LB:DMFIX
[11,15]EXEC/LB:DMDST
[11,15]EXEC/LB:DMDCP
[11,15]EXEC/LB:DMIAR
```

```
[11,15]EXEC/LB:DMSDV
[11,15]EXEC/LB:DMAST
[11,15]EXEC/LB:DMATX
[11,15]EXEC/LB:DMGSS
[11,15]EXEC/LB:TSCH
;
[11,15]EXEC/LB:NTRACE
;
; THE EXEC IS NORMALLY BUILT WITHOUT ODT. TO INCLUDE ODT IN THE EXEC
; CHANGE THE NEXT SPECIFICATION TO BE:  [11,15]EXEC/LB:ODT
;
; THE SYSTEM IS RESUMED FROM ODT BY KEYING "CONTROL C"
;
; THE SYSTEM IS RESUMED FROM A BREAKPOINT BY KEYING "P"
;
[11,15]EXEC/LB:NODT
;
;
; THE EXEC IS BUILT INCLUDING A CRASH MODULE WHICH CAN OPTIONALLY WRITE
; MEMORY ONTO A DUMP MEDIUM AND/OR DISPLAY A CRASH MESSAGE.  IF MESSAGES
; ARE SELECTED ONE OF THE FOLLOWING WILL BE PRINTED ON THE SYSTEM
; CONSOLE IF THE SYSTEM CRASHES.
;
;       ODD ADDRESS OR TRAP 4
;       UNDEFINED SYSTEM EVENT
;       AST NODE PICK FAILURE
;       FLOATING POINT ERROR
;       MEMORY PARITY ERROR
;       NON-RSX EMT EXECUTION
;       RESERVED INSTRUCTION
;       SST IN KERNEL MODE
;       RED OR YELLOW STACK
;       EMT 377 IN KERNEL MODE
;       UNDEFINED SYSTEM ERROR
;
; THE OPTIONS AVAILABLE WITH EACH CRASH MODULE ARE SHOWN BELOW
;
;       MODULE          MESSAGES        MEMORY DUMPING
;       ------          --------        --------------
;
;       NCRASH          NO              NO
;       MCRASH          YES             NO
;       TC11CR          NO              YES -- TC11 DECTAPE   **
;       TC11CM          YES             YES -- TC11 DECTAPE   **
;       TU10CR          NO              YES -- TU10 MAGTAPE
;       TU10CM          YES             YES -- TU10 MAGTAPE
;       TU16CR          NO              YES -- TU16 MAGTAPE
;       TU16CM          YES             YES -- TU16 MAGTAPE
;       RK05CR          NO              YES -- RK05 DISK   ++
;       RK05CM          YES             YES -- RK05 DISK   ++
;
;  **   TC11 DECTAPE CAN NOT DUMP MORE THAN 124K WORDS
;  ++   RK05 DISK CAN NOT DUMP MORE THAN 1152K WORDS
;
;
; THE DEFAULT UNIT NUMBERS FOR THE RESPECTIVE DEVICES ARE:
;
;       TC11 -- #7
;       TU10 -- #7
;       TU16 -- #0
;       RK05 -- #1
;
; THE DEFAULT MAY BE CHANGED BY USING THE TKB GLOBAL PATCH OPTION
; TO MODIFY THE CRASH UNIT NUMBER (.CRSUN) AS DESIRED.  FOR EXAMPLE,
; TO CHANGE THE UNIT NUMBER TO 2, INCLUDE THE FOLLOWING LINE:
;
;       GBLPAT=EXEC:.CRSUN:2
;
[11,15]EXEC/LB:NCRASH
;
[11,15]EXEC/LB:LVEND
[11,15]EXEC/LB:SCOMM
```

TKB15.CMD FILE

```
/
STACK=0
UNITS=0
EXTSCT=:130000
;GBLPAT=EXEC:.CRSUN:0
/
;
;                          ** NOTE **
;
; WHENEVER THE EXECUTIVE IS REBUILT USING THE COMMAND FILE [11,15]TKB15.CMD
; THE BOOTSTRAPS MUST ALSO BE REBUILT USING [11,17]BOOTSBLD.CMD BEFORE
; PERFORMING A SYSTEM GENERATION TO INCLUDE THE NEW EXECUTIVE IN THE STSTEM.
;
```

APPENDIX C

ERROR LOGGING SAMPLE OUTPUT AND DATA RECORD FORMATS

This appendix contains sample output from error logging and the format of records in the ERROR.SYS file produced by the preanalyzer (PRE).

C.1 SAMPLE ERROR LOGGING OUTPUT

```
RSX 11 SYSTEM ERROR REPORT COMPILED AT 20-AUG-75 13:48:53        PAGE 15


************************************************************
TAPE HARDWARE ERROR
LOGGED AT 12-AUG-75 10:19:06            ERROR NUMBER 6.
************************************************************

TAPE PARAMETERS
        UNIT NAME               DT1
        VOLUME LABEL
        VOLUME OWNER UIC        1,1
        DEVICE TYPE             TU56 UNIT-1 CONTROLLER-0

TAPE REGISTERS AT ERROR TIME
        TCST            04034 SELECTION ERROR[1]
        TCCM            104702
        TCWC            177400
        TCBA            113534
        TCDT            000007

ERROR DIAGNOSIS
RECOVERED
SELECTION ERROR[1]

RETRIES PERFORMED       0.

USER TASK PARAMETERS
        TASK NAME                F11ACP
        TASK UIC                 2,2
        PHYSICAL START ADDRESS   102200
        USER FUNCTION REQUESTED         WRITE  (400)
        FUNCTION INTERPRETED FROM REGISTERS    READ BLOCK NUMBER REVERSE
                PHYSICAL BUFFER ADDRESS START  113534
                TRANSFER SIZE IN BYTES         1000
                LOGICAL LOCK NUMBER AT IO GO   2
                COUNT OF IO IN PROGRESS        1.
                COUNT OF IO PENDING            1.

NUMBER OF FUNCTIONS ISSUED TO THIS UNIT        66

VECTORS WITH ACTIVE IO


************************************************************[2]
PRE ANALYZER RUN
AT 12-AUG-75 10:21:01
************************************************************
```

[1]Included only if /DE switch is specified.
[2]System information.

Figure C-1 Tape Error Report

```
RSX 11 SYSTEM ERROR REPORT COMPILED AT 20-AUG-75 13:48:04     PAGE 9.

************************************************************
MEMORY PARITY ERROR
LOGGED AT 12-AUG-75 10:16:27           ERROR NUMBER 13.
************************************************************

MEMORY REGISTERS AT ERROR TIME
        PARITY CSR ADDRESS              172124
        PARITY CSR CONTENTS             000000
PROCESSOR STATUS WORD PRIOR TO THIS ERROR     174000
PHYSICAL START ADDRESS OF ERRING TASK         171500
PROGRAM COUNTER PRIOR TO THIS ERROR           254

ERROR DIAGNOSIS
11/45 MEMORY PARITY ERROR
AT LOCATION 0XX
WITH MEMORY MANAGEMENT REGISTER 0 ACTIVE
IN USER MODE


************************************************************[1]
ERROR LOGGING SYSTEM STOPPED
AT 12-AUG-75 10:17:06
************************************************************


************************************************************[1]
ERROR LOGGING SYSTEM STARTED
AT 12-AUG-75 10:18:35
************************************************************
```

[1]System information.

Figure C-2 Memory Parity Error (No /DE Switch)

```
RSX 11 SYSTEM ERROR REPORT COMPILED AT 20-AUG-75 13:48:04        PAGE 9.


************************************************************
MEMORY PARITY ERROR
LOGGED AT 12-AUG-75 10:16:27            ERROR NUMBER 13.
************************************************************

MEMORY REGISTERS AT ERROR TIME
        PARITY CSR ADDRESS                172124
        PARITY CSR CONTENTS               000000
PROCESSOR STATUS WORD PRIOR TO THIS ERROR        174000
PHYSICAL START ADDRESS OF ERRING TASK            171500
PROGRAM COUNTER PRIOR TO THIS ERROR              254

ERROR DIAGNOSIS
11/45 MEMORY PARITY ERROR
AT LOCATION 0XX
WITH MEMORY MANAGEMENT REGISTER 0 ACTIVE
IN USER MODE


************************************************************
ERROR LOGGING SYSTEM STOPED
AT 12-AUG-75 10:17:06
************************************************************


************************************************************
ERROR LOGGING SYSTEM STARTED
AT 12-AUG-75 10:18:35
************************************************************
```

[1]Included only if /DE switch is specified.

Figure C-3 Memory Parity Error (/DE Switch Specified)

C.2 FORMAT OF RECORDS IN ERROR.SYS FILE

HEADER FIELD (HF)
ALL RECORDS CONTAIN A HEADER FIELD. WORDS Ø-7 OF THE HEADER FIELD ARE STANDARD FOR ALL HEADER FIELDS.

ERROR HEADER FIELD

WD	HIGH BYTE	LOW BYTE
Ø	SIZE OF HEADER FIELD	
1	FORMAT = Ø	ENTRY TYPE (See 1)
2	YEAR	
3	MONTH	
4	DAY	
5	HOUR	
6	MIN	
7	SEC	
8-N	VARIABLE ACCORDING TO ENTRY TYPE	

THE FOLLOWING IS A DESCRIPTION OF WORD 8-N OF THE HEADER FIELDS WHICH ARE PRESENTLY USED.

ENTRY TYPE Ø4Ø ERRLOG STARTUP, TYPE 41 ERRLOG SHUTDOWN, TYPE 43 SOURCE FILE RE-INITIALIZE

8	NEXT UNIVERSAL ERROR SEQUENCE NUMBER
9	POINTER TO INITIALIZE DESCRIPTOR FIELD (IDF) IF ZERO FIELD NOT PRESENT. CURRENTLY = Ø
1Ø	POINTER TO INITIALIZE COMMENTS FIELD (ICF) IF ZERO FIELD NOT PRESENT. CURRENTLY = Ø
11	POINTER TO SYSTEM HARDWARE HISTORY FIELD (SHF) IF ZERO FIELD NOT PRESENT. CURRENTLY = Ø

ENTRY TYPE 1, DEVICE ERROR BIT SET

8	UNIVERSAL ERROR SEQUENCE NUMBER	
9	DEVICE TYPE (see 3)	DEVICE CLASS (see 2)
1Ø	PHYSICAL UNIT NUMBER (THIS BYTE NOT USED)	
11	TWO BYTE ASCII OF THIS DEVICE MNEMONIC WITHIN THE SYSTEM	
12	UNIT NUMBER AS SEEN BY THE SYSTEM	
13	POINTER TO DEVICE REGISTER FIELD	
14	POINTER TO PROGRAM FIELD	
15	POINTER TO VOLUME FIELD	
16	POINTER TO STATISTICS FIELD	
17	POINTER TO BUS ACTIVITY FIELD	

ENTRY TYPE 2, MEMORY PARITY ERROR

8	UNIVERSAL ERROR SEQUENCE NUMBER	
9	DEVICE TYPE (see 3)	DEVICE CLASS (see 2)
1Ø	THIS WORD NOT USED	
11	PROCESSOR STATUS WORD BEFORE PARITY ERROR	
12	PROGRAM COUNTER AT TIME OF PARITY ERROR	
13	1/64 PHYSICAL BASE ADDRESS OF TASK WHICH HAD PARITY ERROR	
14	POINTER TO DEVICE REGISTER FIELD	

DEVICE REGISTER FIELD (DRF)
THIS FIELD IS A COPY OF THE DEVICE REGISTERS WHICH REPRESENT THE STATE OF THE DEVICE AND CONTROLLER AT INTERRUPT TIME. THIS FIELD IS PRESENT WITH MOST DEVICE RELATED ERRORS.

DEVICE REGISTER FIELD

WD	HIGH BYTE	LOW BYTE
Ø	SIZE OF DEVICE REGISTER FIELD	
1	FORMAT = Ø	
2-N	DEVICE AND CONTROLLER REGISTERS AS SEEN ON THE UNIBUS	

PROGRAM FIELD (PGF)
THIS PROGRAM FIELD IS USED FOR DEVICE ERROR ENTRIES.
THIS FIELD IS PRESENT WITH ERRORS WHICH CAN BE TRACED TO A PROGRAM

PROGRAM FIELD

WD	HIGH BYTE	LOW BYTE
Ø	SIZE OF PROGRAM FIELD	
1	2 = 18 BIT ADDRESS SY STEM 4 = 22 BIT ADDRESS SY STEM	FORMAT = 1
2-3	TASK NAME (6 CHARACTER RAD5Ø)	
4	TASK uic PROGRAMMER GROUP CODE	PROGRAMMER CODE
5	1/64 LOAD IMAGE BASE ADDRESS	
6	USER FUNCTION CODE	
7-12	6 USER FUNCTION PARAMETER WORDS (DEVICE SPECIFIC)	
13	MAXIMUM # OF RETRIES	# OF RETRIES REMAINING TO BE DONE
14	COUNT OF THIS TASK'S I/O IN PROGRESS	
15	COUNT OF THIS TASK'S I/O WAITING IN A QUEUE	

VOLUME FIELD (VLF)
THIS FIELD IS USED TO SOFTWARE IDENTIFY THE PHYSICAL MEDIA WHICH IS MOUNTED ON A DEVICE. WHICH PACK OF MULTIPLE PACKS WHICH WAS ON THE DEVICE WHICH ERRED.

VOLUME FIELD

WD	HIGH BYTE	LOW BYTE
Ø	SIZE OF VOLUME FIELD	
1	(NOT USED)	FORMAT = Ø
2-7	12 BYTE ASCII VOLUME NAME	
8	VOLUME OWNER uic GROUP NUMBER	VOLUME OWNER uic PROGRAMMER NUMBER

STATISTICS FIELD (STF)
THIS FIELD REPRESENTS THE RATE OF ERRORS ON A DEVICE OR THE USAGE OF A DEVICE. THIS FIELD IS PRESENT FOR DEVICE ERRORS

STATISTICS FIELD

WD	HIGH BYTE	LOW BYTE
Ø	SIZE	
1	(NOT USED)	FORMAT = 2
2-3	NUMBER OF TIMES THE GO BIT WAS SET FOR THIS UNIT	

BUS ACTIVITY FIELD (BAF)
THIS FIELD REPRESENTS THE ACTIVITY WHICH WAS CONCURRENT ON THE UNIBUS AT THE TIME OF ERROR. THIS FIELD IS PRESENT TO ASSIST IN THE DETECTION OF BUS INTERACTION PROBLEMS.

	WD	HIGH BYTE	LOW BYTE
BUS ACTIVITY FIELD	Ø	SIZE	
	1	(NOT USED)	FORMAT = 1
	2-9	A BIT SET IN EACH WORD REPRESENTS ACTIVITY ON THAT VECTOR I.E., WORD 1 BIT Ø = VECTOR 1ØØ.	

NOTE 1

ENTRY TYPES ARE DIVIDED INTO MAJOR CATEGORIES. THESE MAJOR CATEGORIES CORRESPOND TO A DIVISION WHICH DETECTED THE ERROR.

THE FOLLOWING MAJOR CATEGORIES ARE DEFINED WITH THE RANGE OF ENTRY CODE NUMBERS.

ØØØ UNDEFINED
ØØ1 - Ø37 HARDWARE ERRORS
Ø4Ø - Ø77 SYSTEM

THE FOLLOWING EXPLICIT ENTRY TYPE CODES ARE DEFINED

HARDWARE ERRORS
ØØ1 DEVICE ERROR BIT SET
ØØ2 MEMORY PARITY ERRORS

SYSTEM
Ø4Ø ERRLOG START-UP
Ø41 ERRLOG SHUT-DOWN
Ø43 SOURCE FILE RE-INITIALIZATION

NOTE 2

DEVICE CLASSES ARE DIVIDED INTO MAJOR CATEGORIES ACCORDING TO THEIR FUNCTION WITHIN THE OPERATING SYSTEM.

THE FOLLOWING MAJOR CATEGORIES AND THE CODE NUMBER RANGES ARE DEFINED

ØØØ UNDEFINED
ØØ1 - Ø37 MASS STORAGE
2ØØ - 237 CPU OPTIONS/SYSTEMS

THE FOLLOWING EXPLICIT DEVICE CLASS NUMBERS ARE DEFINED

MASS STORAGE
ØØ1 DISKS
ØØ2 TAPES

CPU OPTIONS/SYSTEMS
2ØØ PROCESSOR

NOTE 3

DEVICE TYPES ARE FURTHER DEFINITION OF THE DEVICE CLASSES. THEREFORE A WHOLE SERIES OF DEVICE TYPE NUMBERS EXIST WITH EACH DEVICE CLASS.

THE FOLLOWING DEVICE TYPES ARE PRESENTLY DEFINED BY DEVICE CLASS

DEVICE TYPE CODES FOR DEVICE CLASS ØØ1 (DISK)

ØØØ	UNDEFINED
ØØ1	RKØ5/RKØ3
ØØ2	RPØ3
ØØ3	RF11
ØØ4	RSØ4
ØØ5	RSØ3
ØØ6	RPØ4
ØØ7	RPØ2 ON RP11C CONTROLLER

DEVICE TYPE CODES FOR DEVICE CLASS ØØ2 (TAPES)

ØØØ	UNDEFINED
ØØ1	TU56
ØØ2	TU1Ø
ØØ3	TU16

DEVICE TYPE CODES FOR DEVICE CLASS 2ØØ (PROCESSOR)

ØØØ	UNDEFINED
ØØ1	11/45
ØØ2	11/7Ø

APPENDIX D

11/70 PARITY ERROR RESPONSE

SOURCE OF CYCLE	CYCLE TYPE	TYPE OF ERROR	ACTION* ALL TRAPS ENABLED	DISABLE WARNING TRAPS (CR00 1)	DISABLE ALL TRAPS (CR 01:00 3)	BITS SET IN ERROR REGISTER
	DATI/P	MAIN BUS TIMEOUT	ABORT	ABORT	ABORT	15 11 8
		MAIN BUS PARITY	ABORT	ABORT	ABORT	15 11 1
		MAIN MEM WANTED WD	ABORT	ABORT	ABORT	15 11 2 or 3
		MAIN MEM OTHER WD	TRAP	NO ACTION	NO ACTION	11 2 or 3
CPU to CACHE		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	11 4 or 5
		FAST MEM DATA	TRAP	NO ACTION	NO ACTION	11 6 or 7
	DATO/B	MAIN BUS TIMEOUT	ABORT	ABORT	ABORT	15 11 0
		MAIN BUS PARITY	ABORT	ABORT	ABORT	15 11 1
	DATI/P	MAIN BUS TIMEOUT	UNIBUS TIMEOUT	UNIBUS TIMEOUT	UNIBUS TIMEOUT	NONE
		MAIN BUS PARITY	UNIBUS TIMEOUT	UNIBUS TIMEOUT	UNIBUS TIMEOUT	10 1
		MAIN MEM WANTED WD	UNIBUS PB TRAP	UNIBUS PB, TRAP	UNIBUS PB	13 10 2 or 3
UNIBUS (thru MAP) to CACHE		MAIN MEM OTHER WD	TRAP	NO ACTION	NO ACTION	10 2 or 3
		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	10 4 or 5
		FAST MEM DATA	TRAP	NO ACTION	NO ACTION	10 6 or 7
	DATO/B	MAIN BUS TIMEOUT	UNIBUS TIMEOUT TRAP	UNIBUS TIMEOUT	UNIBUS TIMEOUT	10 8
		MAIN BUS PARITY	UNIBUS TIMEOUT TRAP	UNIBUS TIMEOUT	UNIBUS TIMEOUT	10 1
		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	10 4 or 5
	DATI/P	MAIN BUS TIMEOUT	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT	NONE
CPU to UNIBUS thru MAP to CACHE		MAIN BUS PARITY	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIME-OUT, ABORT TO 4	10 1
		MAIN MEM WANTED WD	UNIBUS PARITY ABORT TO 114	UNIBUS PARITY ABORT TO 114	UNIBUS PARITY ABORT TO 114	13 10 9 2 or 3
		MAIN MEM OTHER WD	TRAP	NO ACTION	NO ACTION	10 2 or 3
		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	10 4 or 5
		FAST MEM DATA	TRAP	NO ACTION	NO ACTION	10 6 or 7

11/70 PARITY ERROR RESPONSE

SOURCE OF CYCLE	CYCLE TYPE	TYPE OF ERROR	ALL TRAPS ENABLED	DISABLE WARNING TRAPS (CR00 1)	DISABLE ALL TRAPS (CR 01:00 3)	BITS SET IN ERROR REGISTER
	DATAO/B	MAIN BUS TIMEOUT	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT ABORT TO 4	
		MAIN BUS PARITY	UNIBUS TIMEOUT ABORT TO 4 TRAP	UNIBUS TIMEOUT ABORT TO 4	UNIBUS TIMEOUT, ABORT TO 4	10 1
		FAST MEM ADDRESS	TRAP	NO ACTION	NO ACTION	10 4 or 5

*ALL TRAPS VECTOR TO 114

NOTES: 1. These tables only cover single errors.

2. The processor also will abort to 114 when doing DATI/P cycles on the UNIBUS and the device asserts PB. In these cases, bit 9 sets in the error register.

3. Errors on MASSBUS cycles are handled by the MASSBUS controls, and have an effect on the error register.

INDEX

RSX-11D
System Manager's Guide
DEC-11-OXSMA-C-D

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

Did you find errors in this manual? If so, specify by page.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Non-programmer interested in computer concepts and capabilities

Name ______________________ Date ______________

Organization ______________________________________

Street ____

City ______________ State ______________ Zip Code or Country ______________

Please cut along this line.

Fold Here

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
146 Main Street ML5-5/E39
Maynard, Massachusetts 01754

digital equipment corporation

Printed in U.S.A.