DataGeneral
Data General Corporation, Westboro, Massachusetts 01580

DG/UX™ Real-Time Performance Monitor (UX/RPM) Reference Manual

093-000874-02

Systems Evaluation and Performance Analysis Center

Data General

SEPAC Documentation

DG/UX Real-time

Performance Monitor (UX/RPM)

Reference Manual

093-000874-02

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice supplied with the software.

Revision 02, July 1994

NOTICE

DATA GENERAL CORPORATION
4400 Computer Drive
Westboro, MA 01580

UX/RPM Reference Manual
Order No. 093-000874-02

Revision History:

Original Release - September 1992
Rev 01 - December 1993
Rev 02 - July 1994

Effective with:

UX/RPM Rev 2.00
UX/RPM Rev 2.10
UX/RPM Rev 3.00

A vertical bar in the margin of a page indicates substantive technical change from the previous revision.

Contents

Preface

Purpose

This document describes software designed and developed by Data General's Systems Evaluation and Performance Analysis Center (SEPAC) personnel. The software was designed to monitor system performance for Data General DG/UX operating systems.

Who Should Read This Manual

This document is intended for System Administrators who are responsible for monitoring the performance of their computer systems and assumes that such individuals have expert knowledge of DG/UX.

Scope

This document describes the features of the UX/RPM product, provides instructions for using the product, and describes data fields monitored by the product. This document does not include installation instructions. Please refer to the software release notice you received with the software for details about installation and for possible enhancements and changes made to the product since this document was published.

Related Documentation

This document describes occasions when you may need to set environment variables for UX/RPM to effectively monitor your system. If you need instructions for setting system variables, refer to:

Customizing the DG/UX System, DG Order No. 093-701101

In addition, when analyzing performance data, you may find it helpful to refer to the following document that is shipped with each production version of UX/RPM:

Analyzing DG/UX System Performance, DG Order No. 093-701129

Organization

This document is organized into ten chapters:

- Chapter 1 - Introduction to UX/RPM describes the purpose, features, and components of the UX/RPM product.
- Chapter 2 - Using UX/RPM tells you how to start the software, describes command line switches, and describes general product functionality, such as how to access different screens and how to scroll screens.
- Chapters 3 through 10 detail each type of UX/RPM screen and tells you specifically how to access each screen. These chapters also define screen fields for each screen.

The appendixes provide definitions of fields that are output to a DIF file as a result of file conversions and quick-reference tables for files, command keys, and switches.

Reader, Please Note

The following conventions have been used in this document:

command *required* [optional] ... (↵)

where:

command	indicates a character or characters you must type exactly as shown
required	indicates a variable value that you (and sometimes UX/RPM) supply
[ ]	indicates you have the option of entering this command or argument
...	indicates you may repeat the preceding entry or entries
(↵)	indicates you must press the ENTER key

Joining Our Users Group

Please consider joining the largest independent organization of Data General users, the North American Data General Users Group (NADGUG). In addition to making valuable contacts, members receive FOCUS monthly magazine, a conference discount, access to the Software Library and Electronic Bulletin Board, an annual Member Directory, Regional and Special Interest Groups, and much more. For more information about membership in the North American Data General Users Group, call 1-800-932-6693 or 1-508-443-3330.

- End of Preface -

Chapter 1

Introduction to UX/RPM

UX/RPM is a system performance monitor for Data General's DG/UX operating system. It enables a DG/UX System Administrator to view and measure various performance statistics, including CPU and memory utilization, disk, block, and character I/O, process, network, and virtual memory statistics, and system call rates.

UX/RPM provides four basic services:

Logging, which allows you to write all gathered data to a disk file,

Playback, which allows you to examine logfiles at your convenience,

Interactive use, which allows you to view statistical updates as they occur, and

Conversion of log files to Data Interchange Format (DIF) files.

Features

Easy to use, UX/RPM provides a fast and efficient way to identify user processes that impact DG/UX system performance. System Administrators use UX/RPM to:

- Highlight key performance statistics on a single DG/UX system
- Help quickly identify performance bottlenecks
- Display performance data over short or long periods of time
- Show maximum and minimum performance values and when they occurred
- Display process information, sorted by CPU, memory, and priority
- Save all screen images to disk for later evaluation
- Collect performance data to disk for later playback or conversion
- Convert collected data to DIF (Data Interchange Format) for use by spreadsheets and databases

UX/RPM also offers the following features:

- On-line Help screens with information about using UX/RPM
- The ability to fork a DG/UX shell from UX/RPM
- The ability to interactively save a screen dump of the current screen to a disk file for future reference
- Optional automatic execution at any time, day or night

Environment

UX/RPM runs on any AViiON computer system and is designed to run on most display devices. UX/RPM uses a general purpose library to handle the screen and recognize function keys on your terminal or emulator. The TERM environment variable must be set correctly in order for UX/RPM to operate correctly on your display. An incorrectly set TERM variable may result in garbage being written to your display or may impact the ability to use wide-screen display.

UX/RPM requires DG/UX revision 5.4 release 2 or later. Many performance statistics are available only with DG/UX revision 5.4 release 3 or later. These statistics will display as "n/a" when UX/RPM is run on older releases of DG/UX.

Program Control

UX/RPM was designed for a non-graphical environment. This allows it to run on a wide variety of display hardware, from X-Windows graphical displays to simple ASCII terminals and terminal emulators. You execute UX/RPM either with or without switches, depending on your information needs. Chapter 2 provides a complete list of valid switches and switch arguments. The quick reference card provided with this manual provides a summary of the command line switches and has several examples showing proper use of switches to perform different tasks.

All UX/RPM functions can be accessed through single ASCII key strokes or through function keys on terminals that support function key use.

Command Keys

Command keys are one-character keystrokes, such as **w** or **m**. It is not necessary to press the ENTER key after pressing the command key.

Some command keys cause UX/RPM to switch screens. Others modify UX/RPM's behavior in certain screens. The **a** and **A** keys, for instance, toggle the Process Screens' display to show only active processes or all processes.

Function Keys

Function keys are supported on devices and emulators which provide function key codes.

"Soft" function key labels appear across the bottom of all screens and correspond to the first eight function keys of an IBM PC-AT Enhanced keyboard layout. The labels serve as a reminder or "soft template" for those devices that support function keys. The labels also show the single-keystroke command letters (in square brackets) to use with devices that do not support function keys. The labels adapt to the current screen being displayed to indicate functionality appropriate to that screen.

Cycles

UX/RPM collects and updates information in "cycles." Every cycle, information is gathered from DG/UX from a variety of performance-related metering points. This information is then arranged and stored in UX/RPM for comparison against information gathered previously. Certain statistics are compared against historical minimums and maximums to determine whether a new minimum or maximum value has been observed.

The default cycle time is ten seconds. The cycle time can be changed either when UX/RPM is started or at any time during its execution.

Screens

UX/RPM arranges its information into screens. Each screen contains information that is similar in nature or logically related. For instance, all information related to virtual memory management is contained on one screen, while information on DG/UX's process scheduling is shown on another screen. The two most general screens are:

- An "overview" screen that provides a concise look at several important areas of performance at one time, and
- A menu screen that lists available screens as well as the command letters used to access them.

Screen Organization

All UX/RPM screens have a header containing:

- the name of the product
- the current screen name (and its command key)
- the current time
- the time that UX/RPM was started (or the values were reset)
- the selected cycle time
- the actual duration of the last cycle

Screen Labels

Screens labeled "Incremental" or "Current" show the amount of change or activity that has occurred since the previous cycle. Screens labeled "Accumulated" or "Average" show the amount of change that has occurred since UX/RPM was started. "History" screens show minimum and maximum values or rates observed since UX/RPM was started. The time that the minimum or maximum value occurred is shown as well. It is possible to reset the accumulated and history values, effectively "restarting" UX/RPM.

Display Intensity

UX/RPM initially displays information in "bright" or "highlighted" screen intensity, if such a mode is available on the current display hardware. If a value or statistic does not change over

the next cycle, UX/RPM redisplays the field in "dim" intensity (again, if such a mode is available). Thereafter the field is not updated unless it changes. This helps to minimize the I/O performed, thereby lowering the impact the program has on a system being monitored.

Screen Descriptions

UX/RPM displays nine basic types of screens, which are generally described below. (For complete descriptions of screens and their fields, refer to subsequent chapters in this manual.)

System Overview Screen

The UX/RPM System Overview Screen provides an at-a-glance summary of overall system performance conditions. By default, this screen is the initial screen displayed by UX/RPM. It includes data for seven major areas:

- CPU Utilization - Current and average
- Memory Utilization
- I/O Utilization - Current and average
- Load Averaging
- Memory faults - Current and average
- Processes - Current and average
- System calls - Current and average

Menu Screen

This screen displays a list of command keys you use interactively with UX/RPM.

I/O Screens

The "I/O Screen" shows current and average rates of system activities related to I/O as well as the number of activities that have occurred since system boot. I/O activities monitored include iget calls, pathname searches, block READs and WRITEs, physical READ and WRITE requests, TTY input and output characters, total characters read and written, and various interrupts. A display of the current CPU utilization is also provided at the edge of the screen. The "I/O History Screen" shows the minimum and maximum rates observed for these activities.

Disk Screens

Two screens are available for information about physical disk drives. "Disk Screen - Incremental" shows activity since the last cycle. "Disk Screen - Accumulated" shows the amount of activity since UX/RPM started. The information shown for each physical disk includes the disk name and number of requests, percentage of total disk I/O on each drive as well as the percentage of time that a drive was busy, average queue length, blocks read and written, and average service time and response time.

Virtual Disk Screens

Two screens are available for information about virtual disk drives. "Virtual Disk Screen - Incremental" shows activity since the last cycle. "Virtual Disk Screen - Accumulated" shows the amount of activity since UX/RPM started. The information shown for each virtual disk includes the disk name and number of requests, percentage of total disk I/O on each drive as well as the percentage of time that a drive was busy, average queue length, blocks read and

written, and average service time and response time. Note that virtual disks require DG/UX 5.4R3.0 or later.

Network Screens

These screens display information pertaining to the network server and client activities. The "Network Screen" shows current and average rates of Remote Procedure Call and Network Filing System activities, as well as the number of times certain activities have occurred since system boot. A display of the current CPU utilization is also provided at the edge of the screen. The "Network History Screen" shows the minimum and maximum rates observed for these activities.

Process Screens

These screens display information about processes that are currently running on the system. "Process Screen - Incremental" shows activity since the previous cycle. "Process Screen-Accumulated" shows activity since UX/RPM started. A toggle allows either all processes to be displayed or just the processes that were active in the previous cycle. By default, only active processes are shown. It is possible to specify a sort order for the displayed processes. The process screens support a feature whereby it is possible to select one of many possible display formats. These are referred to as windows. The available windows are named based on the type of information they provide. The process scheduling and process virtual memory windows provide information that is available only from DG/UX 5.4R3.0 or later.

Scheduling Screens

Two screens show information related to DG/UX's process scheduling, such as the number of bound, unbound, and eligible processes, bind and unbind rates, system call rates, and switch rates. The "Scheduling Screen" shows current and average values and rates for various process management parameters. It also displays CPU utilization on the side of the screen. The "Scheduling History Screen" indicates maximums and minimums observed along with the times they were observed.

Virtual Memory Screens

The "Virtual Memory Screen" shows information related to memory management, such as operations rates, purge rates, faults, swap space, and table sizes. A CPU utilization display is included on the side of the screen. The "Virtual Memory History Screen" shows the maximum and minimum rates observed by UX/RPM, along with the times they occurred. The virtual memory screens support a feature whereby it is possible to select one of many possible display formats. These are referred to as windows. The windows other than the standard window display information available only from DG/UX 5.4R3.0 or later.

Logging and Playback

UX/RPM enables System Administrators to record system activity to log files. You can specify when, how often, and what kind of data to record, and can specify the name of the log file.

After data is logged, System Administrators can view the log file and observe changes in data as though the changes were happening at that moment. This "playback" mode enables you to analyze previously gathered data.

DIF Files

UX/RPM can be executed with a switch that causes gathered data to be converted and written to DIF files that are usable by most spreadsheet programs and databases.

- End of Chapter -

Chapter 2

Using UX/RPM

UX/RPM is easy to use. It has been designed to give a System Administrator access to relevant DG/UX performance information in an easily understood format.With a single keystroke, you can display any of the available information.

This chapter teaches you how to:

- Prepare certain environment variables for UX/RPM, including the program path, cursor movement, function keys, wide-screen display, and type of shell to provide
- Start and exit UX/RPM
- Display help information
- Use command line switches to:
  - Control the start-up defaults for cycle time, initial screen display, and disks to monitor
  - Set process TTY display
  - Specify logging (including background logging)
  - View logged data
  - Set graphic line displays for your screen
  - Convert log files to a format usable by spreadsheet programs or databases
- Navigate screens, including how to:
  - View a list of command keys
  - Move between screens
  - Toggle screens
  - Scroll displays that cannot fit on a single screen
- Change the cycle time while UX/RPM is running
- Search for text strings on scrollable screens
- Print a copy of the current screen to a disk file
- Reset accumulated and history values
- Fork a shell under UX/RPM
- Set UX/RPM to execute automatically from a script

Preparing Environmental Variables

Before you use UX/RPM, specify the program's pathname, the key set your system uses, whether your device supports function keys and wide displays, and the type of shell that should be provided when you need to access the shell prompt.

To Specify the Program Path

Check that **/usr/sbin** is included in your path. If **/usr/sbin** is not already included in your path, you may add it with the following shell command:

PATH=/usr/sbin:$PATH;export PATH (↵)

To Specify the Cursor Movement Key Set

If your environment does not support function keys, you may need to specify the key set your system uses. The command keys used to scroll UX/RPM screens are specified by the environment variable EDITOR. This allows you to use scrolling or cursor movement keys that are familiar to you. Currently, UX/RPM supports two sets of scroll keys - vi and emacs (see Table 1). If the EDITOR variable is not set, UX/RPM defaults to the vi control key set.

Table 1: UX/RPM Scroll Keys

Function	Function Key	vi keys	emacs keys
Down one line	Cursor Down Arrow	j	CTRL-n
Up one line	Cursor Up Arrow	k	CTRL-p
Down one screen	Page Down	CTRL-f	CTRL-v
Up one screen	Page Up	CTRL-b	ESC v
Go to top of list	Home	1G	ESC <
Go to end of list	End	G	ESC >

To Specify Function Key Support

When using a device that supports function keys, be sure to set the TERM environment variable to accurately reflect that device's abilities. Likewise, when using a terminal emulator (like X-term), be sure to set TERM to an entry that reflects the emulator's function key support (xterm-fk in the case of the X-term emulator).

To Specify Wide-screen Display

Some UX/RPM displays use the full width of the screen to display information. If you are using a resizeable display, such as an X-term window, you must resize the window to the desired width BEFORE beginning UX/RPM to utilize the extra width properly.

In addition, the TERM definition you are using must be able to make use of the extended width. UX/RPM places function key labels at the bottom of the screen. The right-most label can be used to immediately tell if the TERM environment variable is set properly to take

advantage of the full width of the display. If it is not, the LINES and COLUMNS environment variables may need to be set to the correct size of the display.

To Specify a Shell

UX/RPM allows you to create a shell process without exiting UX/RPM. The type of shell created (Bourne, C, Korn, etc.) depends on the setting of the SHELL environment variable. Set SHELL appropriately.

Starting UX/RPM

Once **/usr/sbin** is on your path, you can start UX/RPM from the system prompt by typing, at a minimum:

rpm (↵)

Displaying Help Information

If you want to see a list of UX/RPM switches and the switch format, type the following start-up command:

rpm -h(↵)

UX/RPM executes for only a moment or two, then displays the list of switches and terminates.

To display help for a UX/RPM screen you are viewing, type **?** or **h**, or press function key two (F2) where supported. The Help Screen provides on-line instructions for using UX/RPM and is in the form of a scrollable screen. Press **q** to exit Help and return to the previous screen.

Exiting UX/RPM

To terminate UX/RPM, press **q** (or function key eight (F8), if your system supports function keys) at any time.

If you wish to externally terminate UX/RPM, use the following command line:

dg_kill -TERM rpm (↵)

Be careful that only one copy of UX/RPM is running before using this command. This dg_kill command terminates *all* processes with the rpm name.

Using Command Line Switches

UX/RPM accepts command line switches and arguments to control its initialization. Not all switches require or permit you to specify a value. The format for command line switches is:

rpm *[-switch switch_value]..* (↵)

Refer to specific topics in this section for instructions on how to use switches to control UX/RPM execution. Refer to Appendix B for a concise listing of valid switches and their arguments.

To Set Cycle Time

Use the following switch on the rpm start-up command line to specify how long you want a cycle to be:

-c *n* Sets the cycle time to *n* seconds. *n* must be a positive integer between 0 and 999, inclusive. If you specify a cycle time of 0, you must manually advance UX/RPM to a new cycle by pressing any key that is NOT reserved as a UX/RPM command key (such as the space bar).

To Specify a Start-up Screen

Use the following switch on the rpm start-up command line to specify the screen you want displayed when UX/RPM starts:

-s *screen* *screen* must be one of the following letter codes:

d	Disk screen - Incremental
D	Disk screen - Accumulated
h	Help screen
i	I/O screen
I	I/O history screen
m	Menu screen
n	NFS Screen
N	NFS history screen
o	System overview screen
p	Process screen - Incremental
P	Process screen - Accumulated
s	Scheduling screen

S Scheduling history screen

v Virtual memory screen

V Virtual memory history screen

x Virtual disk screen - Incremental

X Virtual disk screen - Accumulated

These letter codes are also used to switch between screens while UX/RPM is running.

To Exclude Specific Disks from Data Collection

First, create a text file and list the numbers of any physical disks you want to exclude. The disk numbers must be listed one per text line and coincide with the numeric entries in **/dev/pdsk.** Then use the following switch on the rpm start-up command line to exclude those disks from data collection:

-d *file* *file* must be the name of the text file you created.

This switch is useful if disks have been included in your system build, but either do not physically exist or are off-line for some reason. Users with CLARiiONs should note that drives controlled by another system will be inaccessible and should be listed in the disk exclusion file. Users with CD-ROM drives may also wish to exclude these drives, since you will not be able to unmount and eject media from the CD-ROM drive while UX/RPM is collecting performance data on the drive.

To Retrieve TTY Names

By default, UX/RPM shows the major and minor device numbers for each process' TTY. Use the following switch on the rpm start-up command line to resolve device names for process screens:

-t Causes UX/RPM to scan all of the devices in **/dev** and create a table for translating the device numbers into device names. The scan takes extra time during UX/RPM initialization. Omitting this switch causes the program to start faster, especially on large systems with a many TTYs and pseudo-TTYs.

To Specify Logging

Use these switches to control logging. (See *To View Logfiles in Playback Mode* later in this section for details on viewing logged data.)

-l *log_level* Specifies running UX/RPM with logging. *log_level* indicates the kind of information to be recorded. Valid options for *log_level* are:

0 - No logging is performed.

1 - Record system information only. Do not record any disk or process data.

2 - Record system and disk information. Do not record any process data.

3 - Record all available information.

By default, UX/RPM creates a log file named *rpm.<mm>.<dd>*, where <mm> is the current month and <dd> is the current day. Hours, minutes, and seconds are appended to the filename, if needed, to produce a unique filename. The time continues to be incremented until a unique name is found.

-L Causes UX/RPM to output all screens to disk upon termination. (You can accomplish the same effect by pressing the **L** command key immediately followed by the **q** command key during an interactive UX/RPM session.) The output filename is always "rpm.lst" and cannot be changed.

-n *samples* Specifies how many data samples UX/RPM should gather before terminating. *samples* must be a positive integer. This switch is effective only when used in conjunction with logging.

If this switch is not used, UX/RPM continues running until it receives a **q** command or receives a termination signal.

-f *filename* Functionality depends on the presence of the **-l** switch. If you have used **-l**, this switch specifies an output file and allows you to override the default logfile name with *filename*.

NOTE: For descriptions of additional uses of the **-f** switch, see *To View Logfiles in Playback Mode* and *To Specify Logfile Conversions* later in this section.

Each UX/RPM log actually consists of two separate files. One file contains the data gathered by UX/RPM during each snapshot. The second file contains indexes into the data file. By default, the files are named:

rpm.<*datestamp*>.idx (contains indexes)
rpm.<*datestamp*>.log (contains data)

You can override the default filename "rpm" by starting UX/RPM with the **-l** switch (to specify logging) and the **-f** switch (to rename the logfile).

The following sample command line collects system and disk information every 30 seconds until 240 snapshots have been collected, logs the data to a file called "for_sepac," and dumps the last set of screens to the "rpm.lst" ASCII text file:

rpm -l 2 -c 30 -n 240 -f for_sepac -L (↵)

To View Logfiles in Playback Mode

If you do NOT use the **-l** switch (which specifes logging), the **-f** switch starts UX/RPM in playback mode. To view existing logfiles of collected data, start UX/RPM with the following minimum start-up command line:

rpm -f *filename* (↵)

where *filename* is the name of the logfile to read for playback

Do not include the .idx or the .log extension for *filename*.

To View Specific Snapshots

While viewing a logfile in playback mode, press **t** to specify a particular snapshot to examine. UX/RPM displays a pop-up window (Figure 2.1) that shows the minimum and maximum time values found in the log and prompts for the time stamp of the snapshot you want to display. You may enter any time or portion of a time:

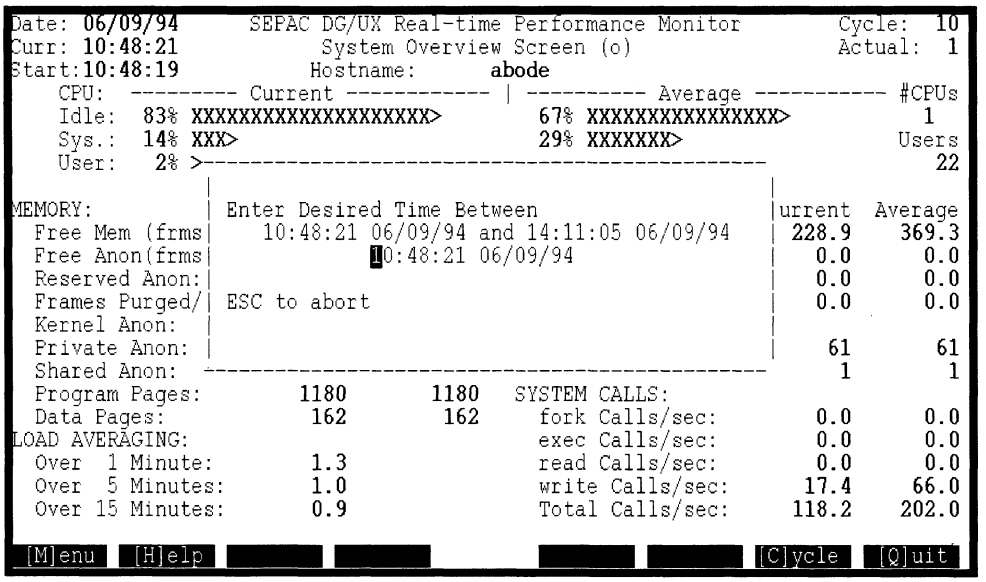

Figure 2.1 - Specifying a Logfile Snapshot to View

If you do not enter an exact snapshot time, UX/RPM displays the snapshot with the nearest time prior to the time you entered.

You can examine system performance during specific time periods by using the **t** command (to move to the beginning of the period), then the **b** command key.

To Maintain Cumulative Values

If process data is collected or displayed, cumulative values must always be recalculated when you specify a new snapshot time to view. However, if process data is not collected or displayed, UX/RPM prompts whether cumulative values should be recalculated based on the snapshot time you specify. A "yes" response to this prompt affects how quickly UX/RPM is

able to update the display. Cumulative values are calculated from the log's start time to the current snapshot time. However, if the snapshot time you specify is previous to the currently displayed snapshot, cumulative values must be recalculated from the beginning of UX/RPM data collection.

To maintain cumulative values, UX/RPM must read all log records between the log's start time and the snapshot you specify. (Start time is defined as the currently displayed snapshot if the snapshot you specify is later than the currently displayed snapshot. If the snapshot you specify is previous to the currently displayed snapshot, start time is defined as the beginning of the log.) Reading many records in a large log file can take awhile. When the snapshot time you specify is more than a few snapshots away from the current time, UX/RPM displays a pop-up menu with a moving wheel to indicate work is in progress.

If you decline cumulative values, then UX/RPM goes directly to the snapshot you specify in the log file. This process is fast, no matter how large the log file is. In this case, all cumulative, average, and history values are reset to zero. This is equivalent to using the **b** command key.

To Reset Cumulative and History Values

The hyphen key (-) displays information for the previous snapshot in the log (relative to the snapshot currently being displayed). By default, the (-) command resets the cumulative and history values, since the specified snapshot time is prior to the currently displayed snapshot.

To Run UX/RPM as a Background Process

While logging, UX/RPM can be run as a background process; this avoids having to dedicate a terminal to UX/RPM and, in fact, uses fewer CPU resources. To run UX/RPM as a background process, simply add an ampersand (&) to the command line. For example:

```
rpm -l 2 -c 30 -n 240 -f for_sepac -L & (↵)
```

To Specify Graphic Screen Displays

Use the following switch on the rpm start-up command line to specify that line drawing graphics should be used in your screen displays:

-g *drawing* *drawing* must be one of the following:

- **l** Limited line drawing graphics. Use line drawing characters for all pop-up boxes, such as those that display warning messages.
- **f** Full line drawing graphics. Use features specified by the **l** argument plus display CPU utilization on the Overview Screen, using bar graphics instead of X characters.

To Specify Logfile Conversions

Use these switches on the rpm start-up command line to convert log files to a format usable by spreadsheet programs or databases.

-C *format* Specifies that a log file is to be converted, using the format indicated. Currently, since only DIF output is available, the only valid value for *format* is **d**.

Even if **-C** is not used, but one or more of **-b, -e, -E,** and **-F** are used when starting UX/RPM, the conversion format defaults to **d**.

-b *snapshot* Specifies the first snapshot to use when converting a log file. If this switch is not used, conversion begins with the first snapshot of data.

snapshot must be entered as either a simple sequence number or as a time in the format of HH:MM:SS MM/DD/YY. Elements after the minutes field are optional; if you use any part of MM/DD/YY, *snapshot* must be enclosed in quotes. For example,

rpm -C -b "12:30:25 12/05/93" (↵)

This example specifies that UX/RPM should convert the default logfile (which is named rpm.<mm>.<dd>) starting with the snapshot specified.

-e *snapshot* Specifies the last snapshot to use when converting a log file. If this switch is not used, conversion ends with the last snapshot of data.

snapshot must be entered as either a simple sequence number or as a time in the format of HH:MM:SS MM/DD/YY. Elements after the minutes field are optional; if you use any part of MM/DD/YY, *snapshot* must be enclosed in quotes.

-E Specifies that an express conversion of a log file to the format specified by the -C switch is to be done. This causes UX/RPM to read the logfile (using the specified beginning and ending shapshots if given) and convert the file to the selected output format. No user interaction is required or permitted during this process. No screens are displayed. This is the recommended way to convert log files.

CAUTION: If you are converting a logfile and do not use this switch, UX/RPM executes in playback mode and performs conversion each time UX/RPM examines a new snapshot. If you use the **t** or "-" command keys, records may be duplicated in or missing from the resulting output file. See *To View Specific Snapshots* and *To Reset Cumulative and History Values* earlier in this chapter for more information on the **t** and "-" command keys.

-f *filename* Specifies an input file to convert. Valid only if you have used one of the conversion switches.

-F *prefix* Specifies the filename prefix to use for a series of files output as a result of file conversions. If the prefix you type is five or fewer characters, filenames for the generated files will conform to DOS filename limits. (UX/RPM automatically appends three characters to the filename.) This simplifies the steps required to import the generated DIF data into a DOS/Windows spreadsheet.

The default prefix is "uxrpm." (See Appendix B for more information on filenames.)

For example, the following command line converts the file "rpm.12.01" and names the output file "log1":

rpm -E -f rpm.12.01 -F log1 (↵)

Even without the **-C** switch, the output file is automatically formatted as a DIF file (because of the use of **-E** and/or **-F**). The input file referred to in the above example is actually two files: rpm.12.01.log and rpm.12.01.idx. Both files are required to be present. (See *To Specify Logging* earlier in this chapter for more details on filename conventions.) The output files are log1hdr.dif, log1sin.dif, log1sum.dif, log1din.dif, log1dum.dif, log1vin.dif, and log1vum.dif.

Navigating UX/RPM Screens

Interactive command keys enable you to move between UX/RPM screens, to perform global actions of one sort or another, and to scroll screens when applicable. After you start UX/RPM, you can access a menu screen to see a list of valid command keys.

To View a List of Interactive Command Keys

To access the menu screen, press the **m** key or function key one (F1). One column on the menu screen lists the key you should press to perform a desired function. The opposite column shows a brief description of the function assigned to each key.

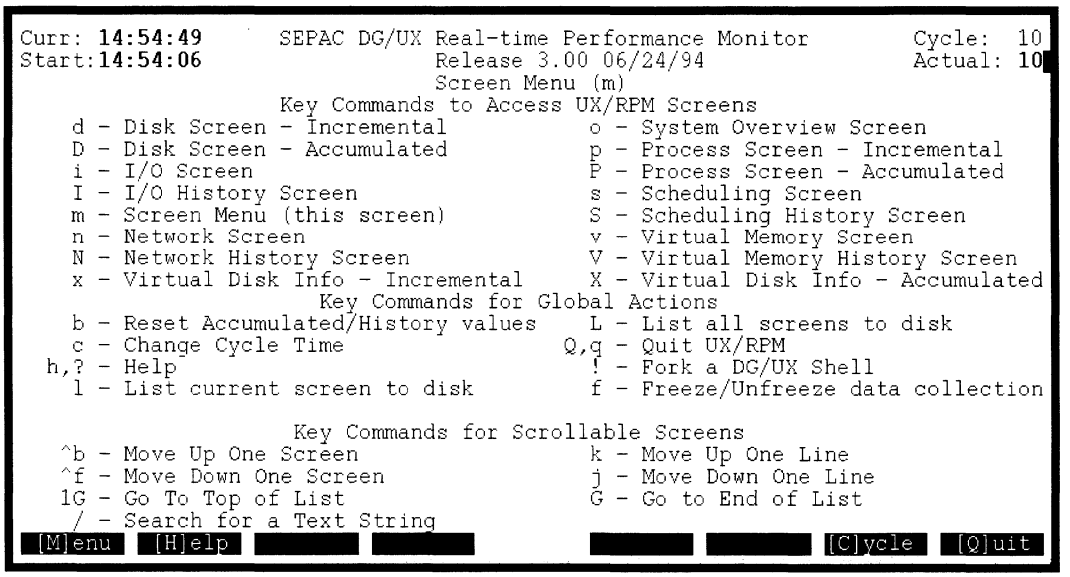

Figure 2.2 - Screen Menu

To View UX/RPM Screens

While UX/RPM is running, to view the screen of your choice press a valid command key (shown in Figure 2.2). Most UX/RPM screens allow you to toggle between two sets of information; some screens present more data than can be displayed all at once and, therefore, can be scrolled.

To Toggle Screens

You can switch screens using the letter characters shown in Figure 2.2. In addition, when viewing screens that can toggle, such as the I/O Screen and the I/O History Screen, if your environment supports function keys you can press function key three (F3) to toggle between the screens. The Process screens, in addition to displaying incremental and accumulated data, **also** allow you to display data for either all or only active processes. You can press function key four (F4) to toggle between the "all" and "only active" displays.

To Scroll Screens

When viewing Help screens or the Process or Disk Screens, there may be more data to be viewed than will fit on the screen. The presence of additional data either above or below the current screen is indicated by first or last lines reading "^^^ more ^^^," or "vvv more vvv."

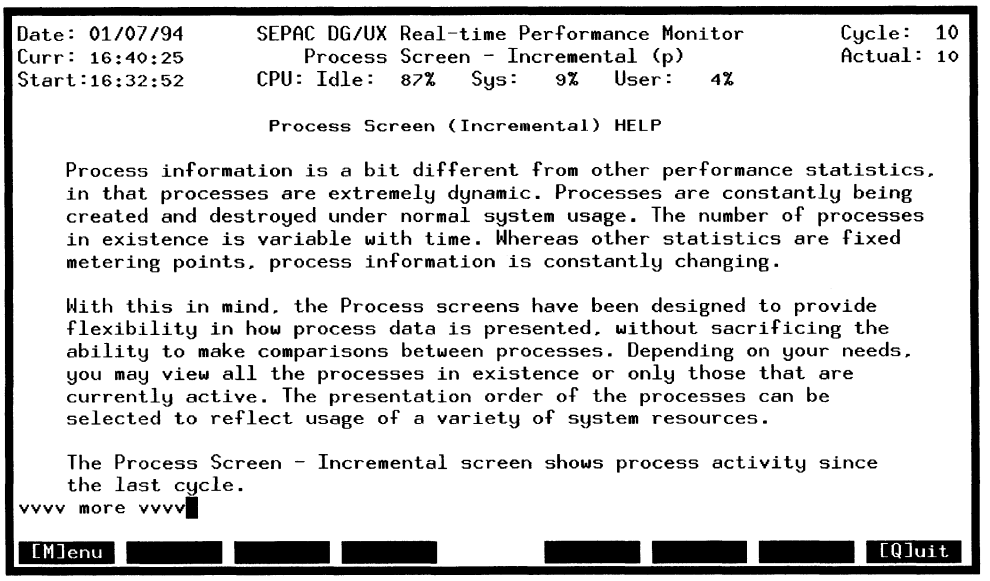

Figure 2.3 - Scrollable Screen

These screens may be scrolled one line at a time or an entire screen at a time. As with the rest of UX/RPM, scrolling can be controlled with either command keys or function keys (if available). Use the set of keys defined by your EDITOR variable or use function keys.

Changing Cycle Time

To change the amount of time between data collection cycles (in realtime mode) or between data sample displays (in playback mode), press the **c** key or function key seven (F7). A pop-up window appears (Figure 2.4) requesting the new cycle time. Enter the number of seconds desired, then press ENTER. The new cycle time must be between 0 and 999 seconds.

During realtime mode, the *Cycle* field displays the number of seconds UX/RPM is supposed to wait between data sampling. The *Actual* field shows the actual time in seconds of the last cycle. Due to system performance or user control, *Actual* may be different from *Cycle*.

In playback mode, *Cycle* displays the number of seconds UX/RPM is supposed to wait to read the log file and update the display. *Actual* displays the actual time (in seconds) of one data sampling cycle, not time elapsed since the last screen update. For example, *Cycle* may be set to 1 second for playback, but if the data was gathered every 10 seconds, then *Actual* displays 10.

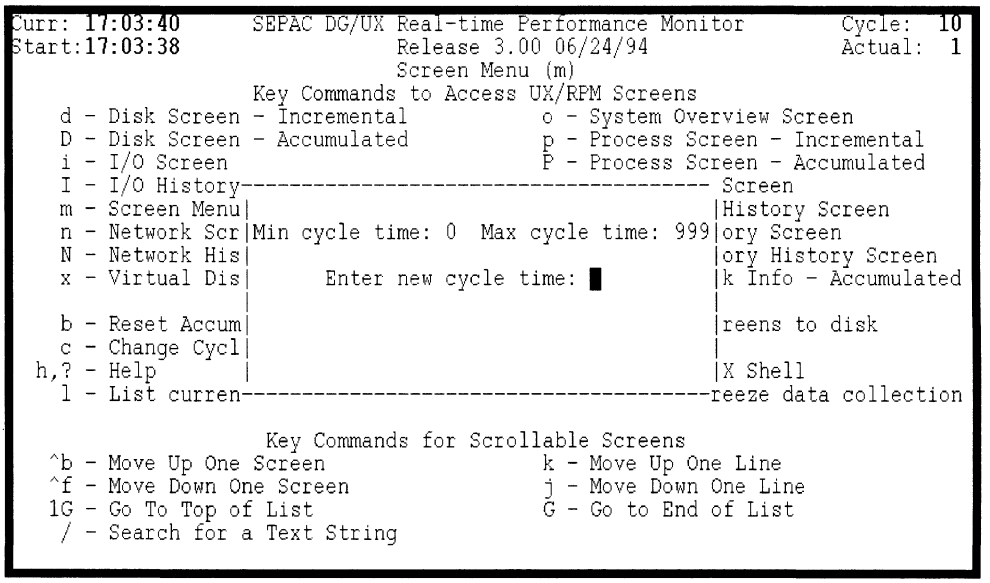

Figure 2.4 - Changing Cycle Time

Searching For Text

In any scrollable screen (Process, Disk, or Help), it is possible to search for a text string.

Press the **/** (backslash) key; a pop-up window appears. Type the text string you wish to search for and press ENTER. UX/RPM searches (starting from the current screen position) for the requested text. The search is case-insensitive. If the text is not found, an error message appears. If the text is found, the line containing the string is positioned at the top of the screen.

To search for further occurrences of the same string, press the backslash key again, then press ENTER.

Interactively Printing Screens to a Disk File

If you start UX/RPM without the **-L** switch, which dumps screens to a disk file, you can interactively dump screens while viewing them. Press the **l** key to send a copy of the current screen to a disk file named rpm.lst. If this file already exists, the screen print is appended to the end of the file. To send a copy of *all* screens to rpm.lst, press **L**. Pressing **l** or **L** is effective even if you started UX/RPM with the **-L** switch.

Resetting Accumulating and History Values

When UX/RPM is started, it stores information needed to show changes in certain values since the monitor was started. This information is used to calculate "accumulated" and "average" values for the various screens. To clear these values, you can either exit and restart UX/RPM or use a command key while UX/RPM is running.

Press the **b** key to reset the accumulated or average values. The **b** key also clears all historical information on maximum and minimum rates observed. To prevent accidental clearing of needed information, no function key is assigned to this command key.

Accessing the Shell from UX/RPM

Often you may need to execute some shell commands or programs without leaving UX/RPM. One reason for not wanting to terminate UX/RPM is that, since it keeps statistics covering the period of time since it was started, exiting would lose those statistics.

To create a shell as a UX/RPM child process (via the fork system call) and pass control to that process, press the **!** key. The type of shell created depends on the setting of the SHELL environment variable.

To exit the shell and return to UX/RPM, type:

exit (↵)

UX/RPM returns execution to the last screen that was on display when the shell was forked.

Setting UX/RPM to Execute Automatically

UX/RPM is delivered with a script that enables you to automatically execute and terminate UX/RPM at any time of the night or day, any day of the week, without impacting any interactive UX/RPM session you might start. The script, called rpm_setup, uses the standard UNIX® cron utility. Cron accepts values from a file called "crontab." The rpm_setup script

enables you to easily set crontab parameters. While this manual tells you how to run rpm_setup to set crontab parameters, cron and crontab are beyond the scope of this manual. For on-line information, type "man cron" or "man crontab" at the shell prompt.

To use rpm_setup, become user **root** (if you aren't already) and go to the **usr/opt/rpm/sbin** directory. Then type the following command line at the shell prompt to display the Script Menu (Figure 2.5):

./rpm_setup (↵)

```
        UX/RPM Automatic collection control

1.      Add a collection
2.      List the collections
3.      Edit/View a collection
4.      Delete a collection
5.      Manage log file policies
6.      Exit updating the collection schedule
x.      Exit without updating the collection schedule

h       for help
Please enter your selection: [6] ▌
```

Figure 2.5 - Script Menu

To Create a Data Collection Profile

Type **1** at the selection prompt on the Script Menu screen to create a collection profile. The system displays the Profile Definition screen (Figure 2.6):

```
Enter collection parameters

        1.      [*]     Days of month
        2.      [*]     Month
        3.      [1-5]   Days of week
        4.      [0,8,16]        Hours of day
        5.      [0]     Minute of hour
        6.      [ON]    Summary flag
        7.      [30]    Cycle time
        8.      [960]   Cycle count
        9.      [2]     Logging level
        0.      done

       Enter your choice 0-9 [0] ▌
```

Figure 2.6 - Profile Definition Screen

The first five values control cron execution. Default values set data collection to occur as follows:

- Every day of the month. The asterisk (*) means "any value." Other values are 1-31.
- Every month of the year. The asterisk (*) means "any value." Other values are 1-12.
- Only on days 1-5 (Monday through Friday), instead of a possible 0-7.
- Three times a day, beginning at 0 hour (midnight), then at 8 hour (8 a.m.), then again at 16 hour (4 p.m.). All time is specified on a 24-hour clock basis.
- On the hour.

The next four values control UX/RPM execution.

- The Summary Flag allows you to specify that all UX/RPM information screens be dumped to a disk file (rmp.lst) when UX/RPM terminates. If the flag is ON, screen dumps are performed. This is equivalent to using the **-L** switch on the rpm start-up command line. Default is ON.
- Cycle Time allows you to specify the time (in seconds) between snapshots (this is the same as the **-c** switch). Default is 30 seconds.
- Cycle Count allows you to specify how many snapshots to collect (same as the **-n** switch). Default is 960 snapshots.
- Logging Level allows you to specify the level of logging to be done (same as the **-l** switch). Default is Level 2 logging.

To exit the Profile Definition screen and return to the Script Menu, type **0** (for Done) at the selection prompt.

To List, Maintain, or Delete a Data Collection Profile

To view a list of existing data collection profiles, select Option 2 from the Script Menu.

To maintain or delete a data collection profile, select Option 3 or 4, respectively, from the Script Menu and when prompted, type the name of the profile you want.

- If you are changing a profile, the system displays an explanation of current settings, then displays the Profile Definition screen and allows you to change information.
- If you are deleting a profile, you are prompted for confirmation to perform the deletion.

To Manage Logfile Retention

Select Option 5 from the Script Menu to specify how long to keep logfiles, where to keep them, and where to keep report files generated when the Summary Flag is set to ON. By default, rpm_setup writes logfiles and report files to /var/spool/rpm.

CAUTION: You can redirect the logfiles to any directory of your choice. However, rpm_setup includes a cleanup routine that deletes files older than the retention period you specify, and the cleanup routine examines ONLY the directory you specify for writing logfiles to. This means it is possible UX/RPM will delete files you don't intend to be deleted. Also, old logfiles in other directories will need to be manually deleted, since the cleanup routine won't examine those directories.

To Implement Changes to Collection Profiles

Select Option 6 from the Script Menu to exit rpm_setup and make changes to collection profiles effective. This updates crontab to cause UX/RPM to run at the scheduled time.

To Save Collection Profile Changes Without Making Them Effective

Type **x** at the Script Menu selection prompt if you want to exit rpm_setup without implementing the changes you made. The new values you specified are saved in a file used by rpm_setup, but do not become effective until you return to rpm_setup and select Option 6.

- End of Chapter -

Chapter 3

System Overview Screen

The System Overview Screen provides a concise summary of several major areas. This summary is often useful in isolating specific areas of concern when tuning the performance of a DG/UX system.

Accessing the Overview Screen

Access the Overview Screen by pressing the **o** key.

Each of the screen areas (except Load Averaging) show both current and average statistics. Current statistics show the amount (or rate) of activity that was observed since the previous cycle. Average statistics indicate activity since UX/RPM was started.

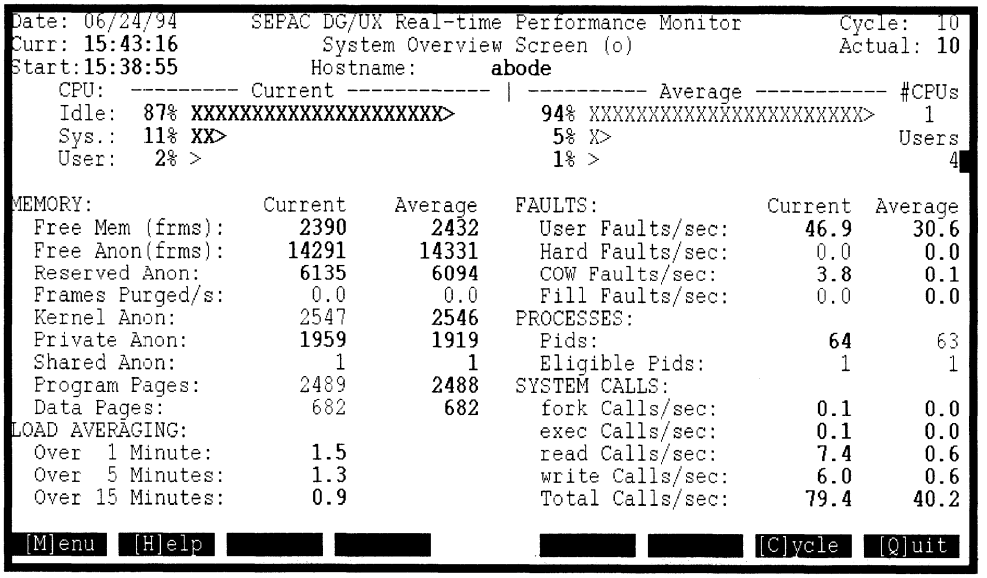

Figure 3.1 - System Overview Screen

Screen Fields

The Overview Screen displays data for seven main categories, beginning with CPU usage statistics in the upper part of the display. Each field is described below.

CPU Usage

CPU usage is shown in two forms: as percentages of total available CPU time and as bar charts. The bar chart is a graphic representation of the numeric percentage where each "X" represents 4%. Both current and average usage figures are shown. Note that available CPU time is greater than the actual real time when more than one processor is present. Two processors together can do 20 seconds of work in 10 seconds of real time.

CPU Idle Time

Idle time indicates how much of the total available CPU time was spent in an idle state. Idle time is time accumulated by a special kernel idle process that runs when there is nothing else for the system to do. Idle time represents unused CPU time.

CPU System Time

System time is time spent executing instructions in the DG/UX kernel, excluding time spent handling interrupts in user processes.

CPU User Time

User time is the time spent executing instructions in user programs.

CPUs

The number of physical CPUs on the host machine is shown to the right of the CPU usage.

Users

The number of TTYs with processes attached to them is shown to the right of the CPU usage. Typically, this is the number of users logged on to the system. Note that this is done differently from the *who* command.

Memory

The memory section of the Overview Screen indicates current and average usage figures for the system's memory. Memory usage is expressed in frames. In current AViiON systems, the frame size is 4K bytes.

Free Memory

Free memory is the number of free memory frames in the free memory pool.

Free Anon

Free anon is the number of unallocated virtual memory frames. On DG/UX systems prior to 5.4R3.0, this number corresponds to free swap as reported by earlier versions of UX/RPM.

Reserved Anonymous

This value represents the amount of virtual memory which has been reserved to some process.

Frames Purged/Sec
This value represents the number of frames purged (reclaims) per second.

Kernel Anon
This is the number of kernel anonymous frames. This value is reported only for DG/UX 5.4R3.0 or later.

User Private
This is the number of user private anonymous frames. This value is reported only for DG/UX 5.4R3.0 or later.

User Shared
This is the number of user shared anonymous frames. This value is reported only for DG/UX 5.4R3.0 or later.

Program Files
This is the number of program file buffering frames. This value is reported only for DG/UX 5.4R3.0 or later.

Data Files
This is the number of data file buffering frames. This value is reported only for DG/UX 5.4R3.0 or later.

Load Averaging

Load Averaging is a method for summarizing system performance. It is defined as the average number of processes in the kernel's eligible queue over a given period of time. Load averages are shown over the previous one-minute, five-minute and fifteen-minute intervals. By comparing the averages over the three intervals, it is possible to note whether the load is climbing or falling. If the five-minute average is greater than the fifteen-minute average, and the one-minute average is greater still, the load would generally be considered to be increasing. A falling average would usually mean that the load is decreasing. The load average is calculated by DG/UX and not by UX/RPM. Because the load average may not have been calculated at the same time as other statistics shown, there may be apparent differences.

Faults

Faults occur when a memory page is needed that is not in main memory.

User Faults/sec
This value shows the current and average rate at which page faults are being taken by processes running in user space.

Hard Faults/sec
This indicates the current and average rates at which page faults are being taken in user space that require disk I/O.

COW Faults/sec
COW (Copy On Write) faults rates are shown for both current and average. For more information on COW faults, see Chapter 9.

Fill Faults/sec

This statistic shows the current and average rates at which page faults are being satisfied by page-in operations from a file or a file system.

Processes

Brief information on running processes is displayed in this section.

Pids

This shows the current and average number of user processes on the system.

Eligible Pids

This indicates the number of eligible bound runnable processes currently running, along with the average number seen since UX/RPM started.

System Calls

A system call causes a process to switch modes - from running in user space to running in kernel space, then back to user space. Since this entails some overhead, the number and type of system calls being performed is a performance concern. The current and average rate of fork(2), exec(2), read(2) and write(2) system calls are shown. Also shown are the current and average rates for all system calls (including those listed above) as well as those not separated out.

- End of Chapter -

Chapter 4

I/O Screens

UX/RPM provides two screens with information relating to I/O. The I/O Screen shows the current and average per-second rates for I/O operations. The I/O History Screen shows the highest and lowest per-second rates that have been observed since UX/RPM was started, along with the time that those rates were observed.

Accessing the I/O Screen

Access the I/O Screen by pressing the **i** key. The I/O Screen shows:

- Current per-second rates
- Average per-second rates
- Total operations since system boot
- Current CPU utilization

```
Date: 12/13/93      SEPAC DG/UX Real-time Performance Monitor      Cycle:  10
Curr: 11:50:19                  I/O Screen (i)                    Actual: 10
Start:11:47:58

                        Current(/s)  Average(/s)   Since Boot       CPU
Iget Calls:                     0.0          0.1        34991
Pathname Searches:             10.9          2.6       171486       idl
Directory Block Reads:          0.5          0.2        46115        45%
Phys. Block Reads:              0.0          0.0          376
Phys. Block Writes:             0.0          0.2        65002       sys
Log. Block Reads:               0.5          0.4       177251        44%
Log. Block Writes:              0.0          0.2       127974
Phys. Read Requests:            0.0          0.0            0       usr
Phys. Write Requests:           0.0          0.0            0        11%
TTY Raw Input Characters:       0.0          0.0         8634
TTY Canonical Input Chars:      0.0          0.0           73
TTY Output Characters:         70.7         50.0       929159
Total Characters Read:       2230.9       1867.7    176509771
Total Characters Written:     191.9        620.2     47387332
Receive Interrupts:             0.0          0.0            0
Transmit Interrupts:            0.0          0.0            0
Modem Interrupts:               0.0          0.0            0


[M]enu [H]elp Hist [I]                                  [C]ycle [Q]uit
```

Figure 4.1 - I/O Screen

Current rates reflect activity observed since the last cycle. Average rates reflect activity observed since UX/RPM started running. The "Since Boot" column displays accumulating

figures that begin at zero when the system is booted and constantly increment. Be aware that these numbers can "wrap," that is, increment past the maximum value that will fit in its allotted storage space. When this happens, the value resets to zero and continues accumulating again. Therefore, if a system has been running for a long period of time, the "Since Boot" figures may not be a completely accurate reflection of all that has happened since the system was booted.

Accessing the I/O History Screen

Access the I/O History Screen by pressing the **I** key. If your environment supports function keys, function key three (F3) toggles the I/O Screen and the I/O History Screen.

The I/O History Screen shows maximum and minimum rates that have been observed by UX/RPM since it started execution. The times at which those "limits" were observed are also shown.

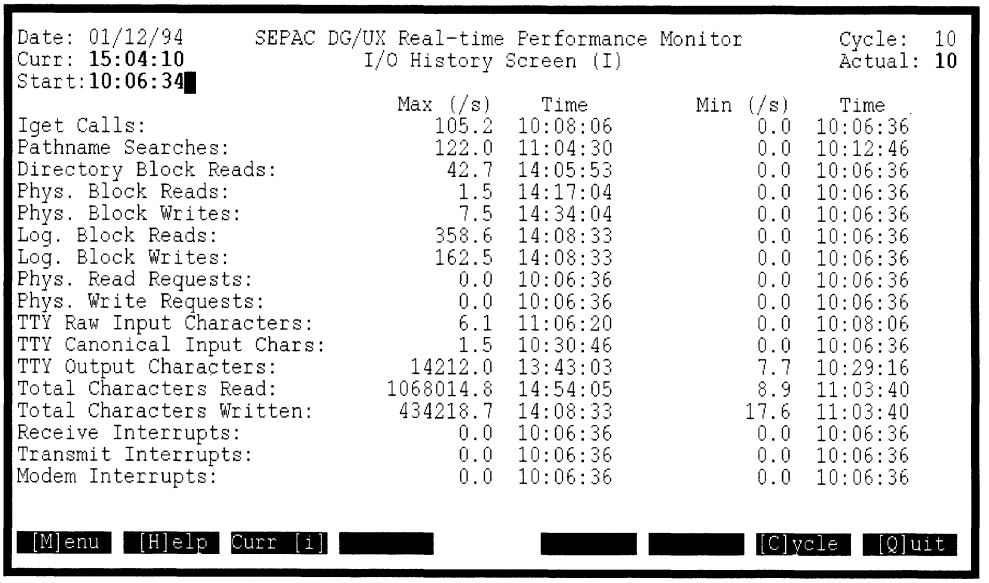

Figure 4.2 - I/O History Screen

Screen Fields

The same data items are shown on both the I/O Screen and the I/O History Screen. Descriptions of each item are given in the following sections.

Iget Calls

This field indicates the per-second rate at which inodes had to be initialized.

Pathname Searches
This indicates how often a pathname had to be resolved.

Directory Block Reads
Directory Block Reads are reads associated with the buffering portion of a directory file.

System I/O Operations
DG/UX reports information about the operating system's usage of I/O. The kind of data being read from and written to disk by the system generally falls under that category of "metadata," that is, data about data. These statistics do not include user I/O.

Physical Block Reads & Physical Block Writes
Physical reads and writes indicate I/O operations that resulted in data actually being transferred from or to a physical device. The Physical Block Reads and Physical Block Writes on the I/O screens give a measurement in blocks per second of those operations.

Logical Block Reads & Logical Block Writes
Logical reads and writes are I/O operations that were satisfied from the system buffers and did not result in actual physical I/O. The Logical Block Reads and Logical Block Writes on the I/O screens give a measurement in blocks per second of those operations.

Physical Read Requests and Physical Write Requests
Physical read and write requests indicate how many times data transfers were requested, whereas the block reads and writes indicate the number of blocks that were requested. By comparing the request rates with the block rates, the average number of blocks per request can be determined.

TTY I/O

TTY Raw Input Characters
Raw input means that no special processing is done and that the characters are readable immediately, without waiting for a line to be assembled. The number of raw characters read per second is shown here.

TTY Canonical Input Characters
In canonical input mode, input characters are queued until a line is complete, as indicated by a line terminator character of some sort (like the **ENTER** key). This statistic shows the rate per second that characters are being read in canonical mode.

TTY Output Characters
This set of fields shows the rate at which characters are being written to terminal output devices.

Character I/O Rates

The next two fields show total character I/O statistics. The current, average and since-boot values are displayed.

Total Characters Read

This field shows the per second rates of all characters read via the read(2) system call.

Total Characters Written

The per second rates of all characters written via the write(2) system call are shown here.

Interrupt Rates

Receive Interrupts and Transmit Interrupts

Receive and transmit interrupts are generated in processing terminal I/O services. Receive interrupts represent interrupts from terminal devices, whereas transmit interrupts are interrupts to those devices. The rate per second for both of these types of interrupts are shown.

Modem Interrupts

These are interrupts for modem related operations such as carrier detection. The rates per second are shown.

- End of Chapter -

Chapter 5

Disk Screens

UX/RPM presents two screens that contain information for activity on physical disk units and displays data for up to fifteen disks on a standard 23-line screen. If more disks are present, a "scrollable indicator" appears above and/or below the lines of disk information. (For information on how to scroll the screen, see Chapter 2.)

While the layout of each screen and the type of information are identical, different actual values are displayed. One screen shows incremental information; that is, information on activity that has occurred since the previous cycle. The other screen shows accumulated data. Accumulated data indicates activity that has occurred since UX/RPM was started.

Accessing the Disk Screens

Access the Incremental Disk Screen by pressing the **d** key. To see the Accumulated Disk Screen, press the **D** key. If your environment supports function keys, function key three (F3) toggles the Incremental Disk Screen and the Accumulated Disk Screen.

```
Date: 06/28/94      SEPAC DG/UX Real-time Performance Monitor        Cycle:  60
Curr: 16:56:31           Disk Screen - Incremental (d)               Actual: 66
Start:16:53:23      CPU: Idle:  31%   Sys:  55%   User:  14%
disk                  # of    %     %  aver      blocks     blocks  aver  aver
name               requests  tot  busy  queue       read    written  serv  resp

sd(dgsc(0,7),0,A)       168    5     5    1.4         68       1066  18.5  25.6
sd(dgsc(0,7),0,B)         0    0     0    0.0          0          0   0.0   0.0
sd(dgsc(0,7),0,C)         0    0     0    0.0          0          0   0.0   0.0
sd(dgsc(0,7),0,D)         0    0     0    0.0          0          0   0.0   0.0
sd(dgsc(0,7),0,E)         0    0     0    0.0          0          0   0.0   0.0
sd(dgsc(0,7),0,F)       967   28    19    6.3       3298       3333  12.7  80.4
sd(dgsc(0,7),0,10)        2    0     0    1.6          0         32  25.0  40.0
sd(dgsc(0,7),0,11)       79    2     2    1.6        146        795  13.4  22.0
sd(dgsc(0,7),0,12)     2207   64    47    7.1       1507      27295  14.1 100.0
sd(dgsc(0,7),0,13)        1    0     0    1.0          8          0  20.0  20.0
sd(dgsc(0,7),0,1F)       25    1     1    1.3          0        128  34.8  45.2

[M]enu [H]elp Accum[D]                                   [C]ycle [Q]uit
```

Figure 5.1 - Incremental Disk Screen

When UX/RPM is initialized, it scans the contents of **/dev/pdsk** to find all of the disk drives that were included in the system when the kernel was built. If your system has disks built into the kernel that do not really exist, or are off-line for some reason, UX/RPM displays a warning message saying that disk statistics for that device are unavailable. To avoid messages of this sort, you should exclude them from the collection by creating a "disk exclusion" file. (See the description of the **-d** switch in Chapter 2 for more details.) Once UX/RPM has initialized, it collects data every cycle on the disks in its list.

Screen Fields

Each screen field is described below.

Disk Name

This is the UNIX standard notation name of the disk drive. It indicates the controller type and number along with the drive number within that controller.

Number of Requests

This column shows how many read and write requests have been processed by a particular drive.

Percent of Total

This column shows what percentage of the total disk I/Os are being handled by a given drive.

Percent Busy

Percent Busy indicates what percentage of the time a drive was busy handling requests.

Average Queue

If a drive is busy when a request is issued, the request is placed in a queue to await service. The length of this queue is an indication of how "backed up" a given drive may be. The Average Queue column shows the average length of a drive's request queue that has been observed, either since the last cycle or since UX/RPM was started.

Blocks Read and Blocks Written

This column shows how many blocks have been read from and written to the device.

Average Service

The average service time of a drive is defined as the total amount of time the drive has been busy servicing requests divided by the total number of requests. This value is represented in milliseconds.

Average Response

Although the Average Service time tells how long it takes to process a request, it does not take into account the time a request may spend in the queue awaiting service. Average Response takes the queue time into account to give an indication of how long it takes, from the time a request is sent to the drive until it is completed, to get a disk request processed.

- End of Chapter -

Chapter 6

Virtual Disk Screens

UX/RPM presents two screens that contain information for activity on virtual disk units and displays data for up to fifteen disks on a standard 23-line screen. If more disks are present, a "scrollable indicator" appears above and/or below the lines of disk information. (For information on how to scroll the screen, see Chapter 2.)

While the layout of each screen and the type of information are identical, different actual values are displayed. One screen shows incremental information; that is, information on activity that has occurred since the previous cycle. The other screen shows accumulated data. Accumulated data indicates activity that has occurred since UX/RPM was started.

Accessing the Virtual Disk Screens

Access the Incremental Virtual Disk Screen by pressing the **x** key. To see the Accumulated Virtual Disk Screen, press **X**. If your environment supports function keys, function key three (F3) toggles the Incremental Virtual Disk Screen and the Accumulated Virtual Disk Screen.

```
Date: 07/08/94      SEPAC DG/UX Real-time Performance Monitor        Cycle:  10
Curr: 10:43:45          Virtual Disk Screen - Incremental (x)       Actual: 10
Start:10:43:24      CPU: Idle:  93%   Sys:   6%   User:   1%
disk                  # of     %     %  aver      blocks      blocks  aver  aver
name              requests   tot  busy  queue        read     written  serv  resp

swap                     0     0     0   0.0           0           0   0.0   0.0
root                     7   100     3   1.0           0          82  47.1  47.1
usr                      0     0     0   0.0           0           0   0.0   0.0
usr_opt_X11              0     0     0   0.0           0           0   0.0   0.0
usr_opt_networker        0     0     0   0.0           0           0   0.0   0.0
usr_opt_xdt              0     0     0   0.0           0           0   0.0   0.0
var_opt_networker        0     0     0   0.0           0           0   0.0   0.0
swap1                    0     0     0   0.0           0           0   0.0   0.0
var_tmp                  0     0     0   0.0           0           0   0.0   0.0
usr_opt_mxdb             0     0     0   0.0           0           0   0.0   0.0
usr_opt_mxdb.demo        0     0     0   0.0           0           0   0.0   0.0
usr_opt_rpm              0     0     0   0.0           0           0   0.0   0.0
tools                    0     0     0   0.0           0           0   0.0   0.0

[M]enu  [H]elp  Accum[X]                                        [C]ycle  [Q]uit
```

Figure 6.1 - Incremental Virtual Disk Screen

When UX/RPM is initialized, it scans the contents of **/dev/dsk** to find all of the virtual disk drives. Once UX/RPM has initialized, it collects data every cycle on the virtual disks in its list.

Screen Fields

Each screen field is described below.

Disk Name

This is the name of the virtual disk drive. There is a long and short name for each virtual disk which appears in /dev/dsk. UX/RPM displays the short version of the virtual disk name.

Number of Requests

This column shows how many read and write requests have been processed by a particular virtual disk.

Percent of Total

This column shows what percentage of the total disk I/Os are being handled by a given drive.

Percent Busy

Percent Busy indicates what percentage of the time a drive was busy handling requests.

Average Queue

If a drive is busy when a request is issued, the request is placed in a queue to await service. The length of this queue is an indication of how "backed up" a given drive may be. The Average Queue column shows the average length of a drive's request queue that has been observed, either since the last cycle or since UX/RPM was started.

Blocks Read and Blocks Written

This column shows how many blocks have been read from and written to the device.

Average Service

The average service time of a drive is defined as the total amount of time the drive has been busy servicing requests divided by the total number of requests. This value is represented in milliseconds.

Average Response

Although the Average Service time tells how long it takes to process a request, it does not take into account the time a request may spend in the queue awaiting service. Average Response takes the queue time into account to give an indication of how long it takes, from the time a request is sent to the drive until it is completed, to get a disk request processed.

- End of Chapter -

Chapter 7

Network Screens

UX/RPM collects statistics on both Network File Systems (NFS) and Remote Procedure Calls (RPC). These statistics are grouped logically into the Network Screen and the Network History Screen.

Accessing the Network Screen

Access the Network Screen by pressing the **n** key.

The Network Screen shows both current and average per second rates for a variety of NFS and RPC activities. Both server and client statistics are shown. The current rates reflect the most recent activity (within the previous cycle). Average rates show the activity observed since UX/RPM was started, averaged over the entire period.

```
Date: 06/28/94      SEPAC DG/UX Real-time Performance Monitor        Cycle:  60
Curr: 17:07:31              Network Screen (n)                      Actual: 60
Start:16:53:23          Current(/s)  Average(/s)   Since Boot
RPC Server Calls:               0.3          0.8      3362991        CPU
RPC Server Bad Calls:           0.0          0.0            0
RPC Server Null Received:       0.0          0.0            0        idl
RPC Server Bad Lengths:         0.0          0.0            0         36%
RPC Server Bad Headers:         0.0          0.0            0
RPC Client Calls:               1.2          0.6        94723        sys
RPC Client Bad Calls:           0.0          0.0            0         49%
RPC Client Retransmissions:     0.0          0.0         6517
RPC Client Bad XID:             0.0          0.0          332        usr
RPC Client Timeouts:            0.0          0.0         6517         15%
RPC Client Waits:               0.0          0.0            0
RPC Client New Cred:            0.0          0.0            0
NFS Server Calls:               0.3          0.8      3362979
NFS Server Bad Calls:           0.0          0.0            1
NFS Server Procedure Calls:     0.3          0.8      3362979
NFS Client Sleeps:              0.0          0.0            0
NFS Client Gets:                1.2          0.6       181816
NFS Client Calls:               1.2          0.6        94785
NFS Client Bad Calls:           0.0          0.0            8
NFS Client Procedure Calls:     1.2          0.0        94785
 [M]enu  [H]elp  Hist [N]                          [C]ycle  [Q]uit
```

Figure 7.1 - Network Screen

Accessing the Network History Screen

Access the Network History Screen by pressing the **N** key. If your environment supports function keys, function key three (F3) toggles the Network Screen and the Network History Screen.

```
Date: 06/28/94      SEPAC DG/UX Real-time Performance Monitor        Cycle:  60
Curr: 17:10:31            Network History Screen (N)                Actual: 60
Start:16:53:23                  Max (/s)    Time        Min (/s)    Time
RPC Server Calls:                    8.5  16:56:31           0.0  16:53:25
RPC Server Bad Calls:                0.0  16:53:25           0.0  16:53:25
RPC Server Null Received:            0.0  16:53:25           0.0  16:53:25
RPC Server Bad Lengths:              0.0  16:53:25           0.0  16:53:25
RPC Server Bad Headers:              0.0  16:53:25           0.0  16:53:25
RPC Client Calls:                    4.7  16:54:55           0.0  16:53:25
RPC Client Bad Calls:                0.0  16:53:25           0.0  16:53:25
RPC Client Retransmissions:          0.0  16:53:25           0.0  16:53:25
RPC Client Bad XID:                  0.0  16:53:25           0.0  16:53:25
RPC Client Timeouts:                 0.0  16:53:25           0.0  16:53:25
RPC Client Waits:                    0.0  16:53:25           0.0  16:53:25
RPC Client New Cred:                 0.0  16:53:25           0.0  16:53:25
NFS Server Calls:                    8.5  16:56:31           0.0  16:53:25
NFS Server Bad Calls:                0.0  16:53:25           0.0  16:53:25
NFS Server Procedure Calls:          8.5  16:56:31           0.0  16:53:25
NFS Client Sleeps:                   0.0  16:53:25           0.0  16:53:25
NFS Client Gets:                     5.2  16:54:55           0.0  16:53:25
NFS Client Calls:                    4.7  16:54:55           0.0  16:53:25
NFS Client Bad Calls:                0.0  16:53:25           0.0  16:53:25
NFS Client Procedure Calls:          4.7  16:54:55           0.0  16:53:25
 [M]enu  [H]elp  Curr [n]                                  [C]ycle  [Q]uit
```

Figure 7.2 - Network History Screen

The Network History Screen displays maximum and minimum per second rates that have been observed since UX/RPM was started. The times of those occurrences are also shown.

Screen Fields

Statistics are grouped into Remote Procedure Calls, or RPC, and Network File System, or NFS, data.

RPC Statistics

Remote Procedure Calls, or RPC, is the name given to a layer of services which distribute computing resources across a network. RPC consists of Clients and Servers. An RPC Client intercepts a service request on one system in a network, packages the request and transmits it over the network to an RPC Server. The RPC Server unpackages the request on another system and executes the request. The results are then passed back to the RPC Client which passes them to the original requestor.

RPC Server Calls

This shows the total number of RPC calls per second that have been received by the server.

RPC Server Bad Calls
Bad RPC Server Calls are calls that were made to the server which were rejected by the server for whatever reason.

RPC Server Null Received
This is the number of times that no RPC packet was available when the server tried to receive a request.

RPC Server Bad Lengths
This is the number of times that the RPC Server received a packet with an invalid length.

RPC Server Bad Headers
This is the number of times that an RPC packet with a malformed header was received by the server.

RPC Client Calls
This is the total number of calls made by the RPC client.

RPC Client Bad Calls
This indicates the number of calls made by the RPC client that were rejected by the server.

RPC Client Retransmissions
The number of times that an RPC call had to be retransmitted by the client.

RPC Client Bad XID
Number of times that a reply to a RPC client call did not match what the client sent.

RPC Client Timeouts
Number of times that a call was deferred because a client handle was busy.

RPC Client Waits
Number of times that a reply to a call was not received within the time allowed by the timeout value.

RPC Client New Cred
Number of times that the client had to refetch authentication information.

NFS Statistics

The Network File System, or NFS, is a distributed file system product developed and licensed by Sun Microsystems. NFS gives users the ability to access files over a communications link as if they were resident on the user's local machine. NFS is typically found running over Ethernet LANs. NFS uses RPCs and, like RPC, has both servers and clients.

NFS Server Calls
This is the total number of NFS calls received by the NFS server.

NFS Server Bad Calls
Number of NFS calls rejected by the NFS server.

NFS Server Procedure Calls
The number of RPC calls handled by NFS.

NFS Client Calls
Total number of NFS client calls.

NFS Client Sleeps
Number of times that a client NFS call had to be deferred because resources were not available.

NFS Client Gets
Number of times that a client handle had to be created.

NFS Client Bad Calls
Number of NFS client calls that were not received or were rejected.

NFS Client Procedure Calls
The number of Remote Procedure Calls generated by NFS.

- End of Chapter -

Chapter 8

Process Screens

Process information is a bit different from other performance statistics. The main difference is that processes are extremely dynamic. Processes are constantly being created and destroyed under normal system usage. The number of processes in existence is variable with time. Whereas other statistics are fixed metering points, process information is constantly changing. With this in mind, the Process screens have been designed to provide flexibility in how process data is presented without sacrificing the ability to make comparisons between processes.

Accessing the Process Screens

Two general Process screens are provided: one screen shows incremental data while the other shows accumulated data. Process screens can also present seven different displays. Like the Disk screens, the two Process screens differ only in the values displayed. Depending on your needs, you may view either all the processes in existence or only those that are currently active. No data is ever shown for processes that no longer exist.

Process screens also allow you to change the sort order for process data and to change the window display.

To Display Incremental or Accumulated Data

Press the **p** key to display incremental data (shown in Figure 8.1). "Process Screen - Incremental" shows the amount of activity that occurred during the previous cycle.

Press the **P** key to view accumulated data. "Process Screen - Accumulated" is identical to the Incremental screen except that it shows activity since UX/RPM was started.

If your environment supports function keys, function key three (F3) toggles the Incremental and Accumulated screens.

```
Date: 06/28/94      SEPAC DG/UX Real-time Performance Monitor        Cycle:  10
Curr: 17:25:35        Process Screen - Incremental (p) -std          Actual: 10
Start:16:53:23      CPU: Idle:  57%   Sys:  33%   User:  11%

  pid user       fsi  pri nice size tty       cpu     command

27520 scj        2W3 1530  20   565 20,02     0.41000 ./rpm
 4005 usenet     2W4 1522  22    50 na        0.20000 sh /usr/lib/news/newsbin/
26235 0          2W4 1544  20    93 na        0.15000 nntpd
  311 0          2W4 1544  20     1 na        0.06000 biod 8
  553 0          2W4 1544  20    90 na        0.04000 dgsvc_d /dev/tty00
 1035 0          2W4 1544  20    62 na        0.03000 tpd -c A4_PRINTER
  313 0          2W4 1544  20     1 na        0.03000 biod 8
  847 0          2W4 1544  20    60 na        0.03000 nfsd 12
  315 0          2W4 1544  20     1 na        0.02000 biod 8
  898 0          2W4 1676   8    36 na        0.02000 /usr/bin/xntpd -b
  314 0          2W4 1544  20     1 na        0.01000 biod 8
  930 0          2W4 1544  20   170 na        0.01000 /usr/bin/gated
 1092 daemon     2W4 1544  20   309 na        0.01000 ora_pmon_RAM7

[M]enu  [H]elp  Accum[P]  [A]ll          [R]eordr [W]indow [C]ycle  [Q]uit
```

Figure 8.1 - Standard Incremental Process Screen

To Display Active or All Processes

Press **a** to see only active processes, or **A** to see all processes. Function key four (F4) toggles the display.

To Change the Sort Order

You can sort processes by several criterion. When examining a system that is short of CPU time, you may want to order the processes by their consumption of CPU time. If memory usage is of more concern than CPU time, you may wish to order the processes by their resident size in memory. UX/RPM allows you to sort processes by:

- CPU usage
- Nice value
- Process ID (PID)
- Priority
- Resident process size

To change the process sort order, press the **r** key (or F5). (The **r** key is effective only when viewing one of the Process screens.) A pop-up window appears (Figure 8.2), and you may enter the letter code for the sort order you want.

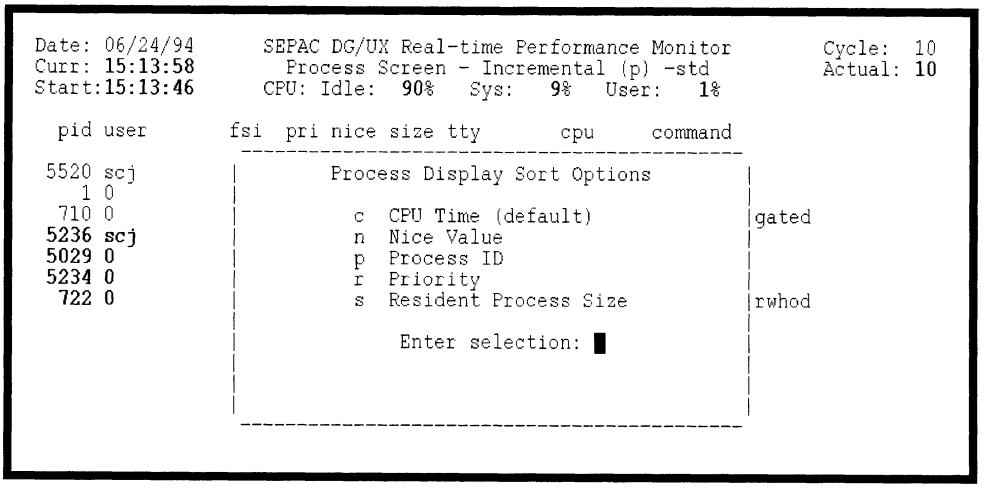

Figure 8.2 - Changing Process Sort Order

To Change Column Data Display

You can display additional process data by changing the window display. While viewing the Process screen, press the **w** key or function key six (F6). A pop-up menu displays an ENTER SELECTION prompt (Figure 8.3).

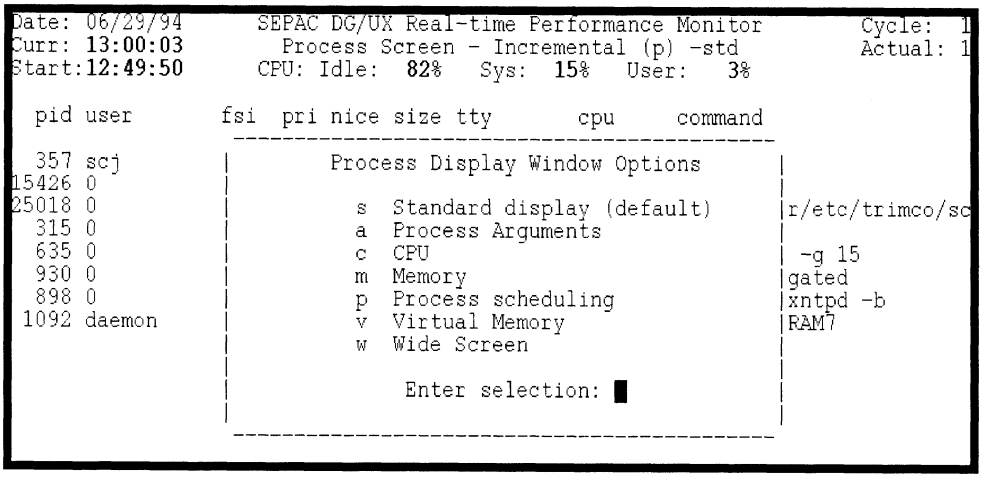

Figure 8.3 - Process Display Pop-up Menu

At the ENTER SELECTION prompt, type the letter for the display you want to see.

s Shows the default display. The default display contains a mix of general information for memory and CPU usage. See Figure 8.1.

a Displays the process command line arguments. UX/RPM devotes most of the screen width to displaying the command line arguments. See Figure 8.4.

c Displays user, system, and %CPU time data. UX/RPM displays additional columns of data on the right side of the screen. The column labelled "user" displays the amount of CPU time consumed by the process in user mode; the "system" column displays the amount of CPU time consumed by the process in kernel mode; and the "%CPU" column displays the amount of CPU time consumed by the process relative to the amount available. For example, on a four processor system, a process which completely uses one of the processors would show 25% for this value. See Figure 8.5.

m Displays memory and wait data. UX/RPM displays additional columns of data on the right side of the screen to show Swap Space, Memory, and Wait data. The column marked "swap" displays the anonymous memory reserved for use by the process, whether actually used or not. The column labelled "memory" displays the memory address of the process. The column labelled "wait" displays the event for which the process is waiting or sleeping. See Figure 8.6.

p Displays process scheduling information. UX/RPM displays additional columns of data to show the scheduling class, emulated instruction count, floating point exception count and other interesting data. The information for this display is available only with DG/UX 5.4R3.0 or later. If you are using an earlier release of DG/UX, these fields default to "n/a" because the information is not available. See Figure 8.7.

v Displays information about each process' use of virtual memory. UX/RPM displays information about virtual memory usage for each process. This information is available only if you are using DG/UX 5.4R3.0 or later. If you are using an earlier release of DG/UX these fields default to "n/a" because the information is not available. The Virtual Memory screen is not shown.

w Displays all available process data; you must first resize your x-term window to accommodate wide-screen displays. To resize your window, either use the geometry command line option when creating the window or use the Window Manager to resize it. This display does not include the information from the process scheduling and virtual memory windows.

a - The process arguments screen (Figure 8.4) displays the pid, user, tty, and command fields

.

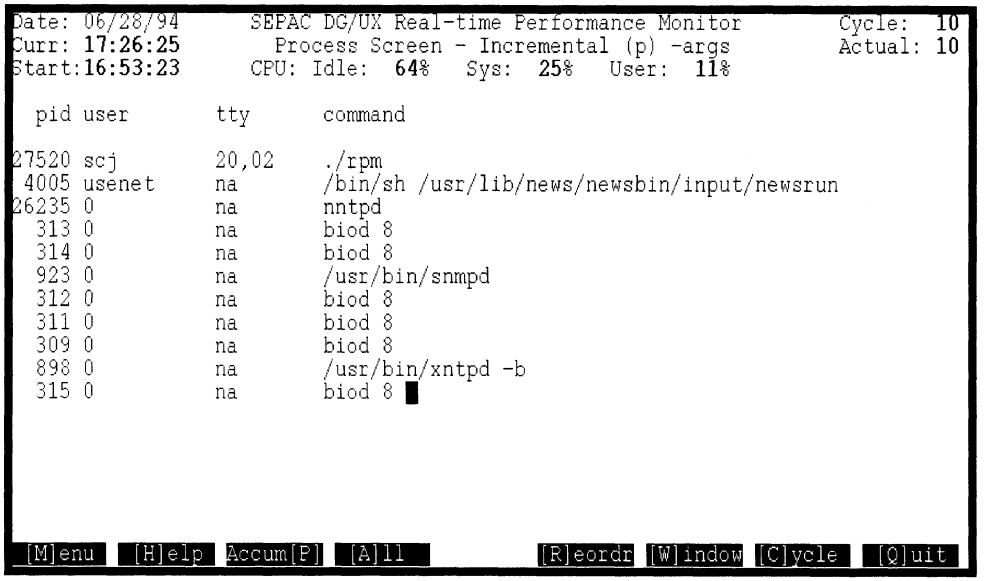

Figure 8.4 - Incremental Process Screen - Process Arguments

c - The CPU usage screen (Figure 8.5) displays the pid, user, f, s, i, pri, nice, size, tty, user cpu, system cpu, %cpu, and command fields. Note that %CPU is of the total available.

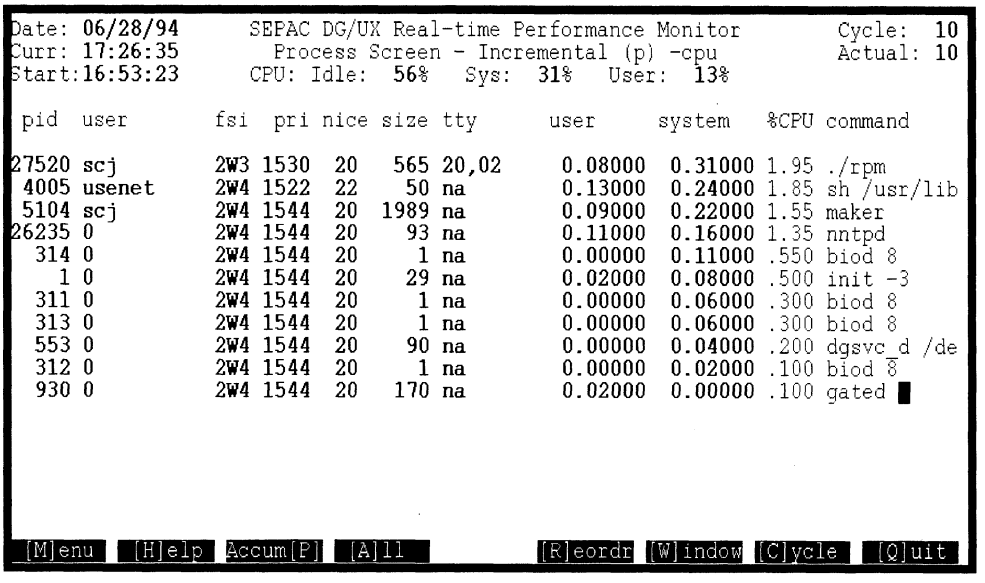

Figure 8.5 - Incremental Process Screen -CPU

m - The memory screen (Figure 8.6) displays the pid, user, f, s, i, pri, nice, size, swap, tty, memory, wait and command fields:

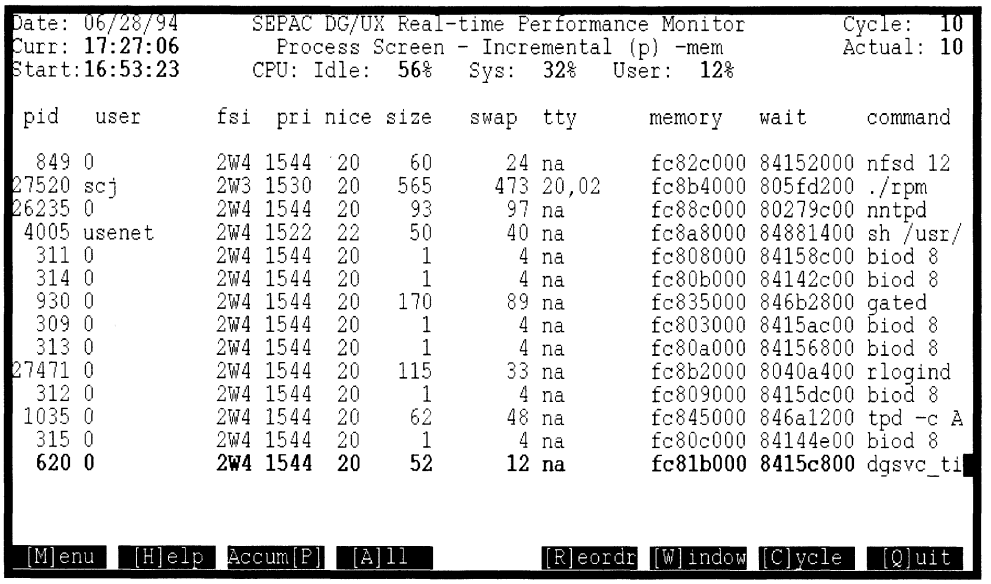

Figure 8.6 - Incremental Process Screen - Memory

p - The process scheduling screen (Figure 8.7) displays the pid, user, pri, size, cpu, sc, emul, fpex, lwps, binds, sock_w, sock_r, sigs, v_switch, iv_switch and command fields:

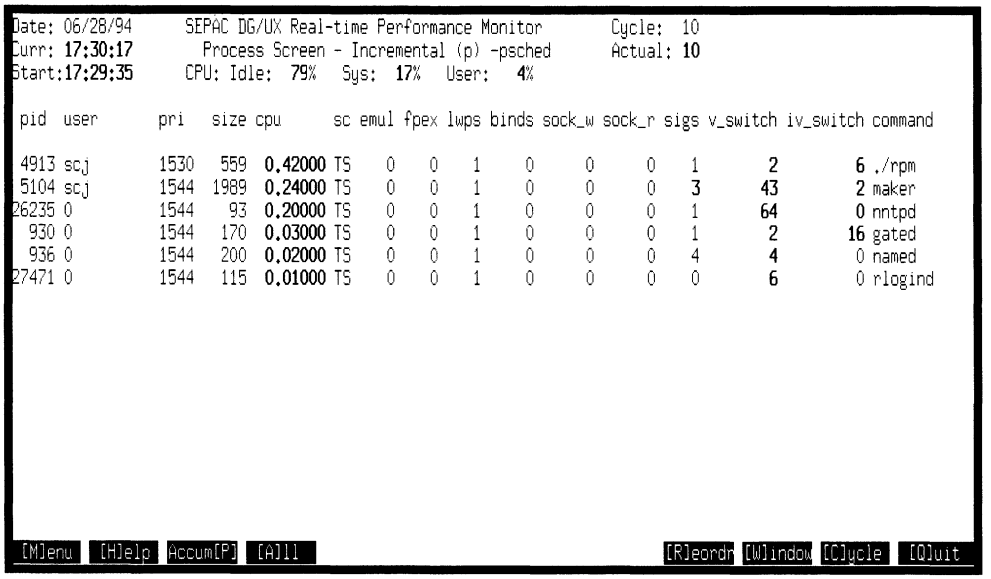

Figure 8.7 - Incremental Process Screen - Scheduling

Screen Fields

Process data is shown in columns. The formats of both the Incremental and Accumulated screens are the same; only the data values themselves differ. Each screen field is described below, including fields that display only when you change the window to a wide display or when you change the column data display.

PID

This column shows the Process ID, or PID of the process. Process IDs are assigned sequentially from 1 to 65535. When PID numbers reach 65535, the assigned numbers go back to 1, skipping any numbers that are still in use.

PPID

This column shows the parent Process ID, or PPID of the process. They have the same range as PIDs.

PGRP

This column shows the Progress Group ID, or PID of the process. This is the PID of the process group leader. Many daemon processes do not have a process group leader.

User

This column displays the username assigned to the process. In some cases, there is no user name associated with a process, such as a system daemon. In those cases, a user number is displayed.

Process Flags

The column labelled "f" shows the process flags for each process. Possible flags are:

1 Process is being traced
2 Process is bound to a virtual processor
4 Process is not bound to a virtual processor

Process State

The column labelled "s" shows process state. Possible states are:

S Sleeping
W Waiting
R Running
I Intermediate
Z Terminated
T Stopped

Interactivity

Interactivity is a measure of how a process is utilizing the processor time allocated to it. A highly interactive process tends to release the processor before its time slice has expired, whereas a compute-bound process will use the entire slice. The column labelled "i" shows each process' interactivity rating (sometimes called "processor utilization"). Interactivity is represented by an integer from 0 to 7. A process with a 7 is highly interactive. A process with a 0 is not considered interactive and uses mostly CPU resources. A process' interactivity influences its priority.

Priority

The assigned priority of each process is shown in the column labelled "pri". Higher priority numbers mean lower priority. Priorities are assigned by DG/UX and are totally dependent on the revision of the operating system. For more information on how priorities are assigned, see Chapter 2 of *Analyzing DG/UX System Performance.*

Nice Value

The nice value is a number, ranging from -20 to 20, which is added to the priority to alter its scheduling. Since UNIX priorities decrease as the priority number increases, a higher nice value lowers the actual scheduling priority of a process. The nice value is sometimes looked at as a rating of "niceness."

Size

The column labelled "Size" displays the resident program size of a process. The size is expressed in pages of memory. A page of memory is 4k (4096) bytes.

Swap Space

The column labelled "swap" displays the anonymous memory (swap space) reserved for use by the process, whether actually used or not.

TTY

This column shows the terminal line associated with each process. If no console is associated with the process, n/a appears in this column. By default, UX/RPM shows only the major and minor device number for TTYs (in the format *major,minor*). If you wish to see the associated device names instead of the device numbers, use the **-t** switch when starting the monitor. For more information on the **-t** switch, see Chapter 2.

Start

This column shows the time of day this process was started. If over a day old, it shows the date the process was started.

CPU

This column shows the amount of CPU time used by each process. The format used depends on the amount of time used:

< 1 minute	SS.sssss
< 1 hour	MM:SS.ss
< 1 day	HH:MM:SS
>= 1 day	DD/HH:MM

CPU is the sum of User CPU and System CPU.

User CPU

This column shows the amount of CPU time used by the process in user mode.

System CPU

This column shows the amount of CPU time used by the process in kernel mode.

% CPU

This column shows the amount of CPU used by the process relative to the amount available. For example, on a four processor system, a process which completely uses one of the processors would show 25% for this value.

Memory

This column shows the memory address of the process.

Wait

This column shows the event for which the process is waiting or sleeping.

Scheduling class

The column labelled "sc" displays the scheduling class for the process. The values shown are FF (FiFo), RR (Round Robin), TS (Time Sharing), LF (LiFo) and DG(Data General fifo).

Emulated Instruction Count

The column labelled "emul" displays the number of 88100 instructions which have been emulated for this process. On 88100 based systems this is always zero. On 88110 based systems, if this number is non-zero your application needs to be recompiled with a recent compiler for optimum performance.

Floating Point Exception Count
The column labelled "fpex"displays the number of floating point exceptions.

Light Weight Processes
The column labelled "lwp" displays the number of light weight processes associated with a process. Non-threaded applications will show 1 in this column.

Binds
This column shows the number of times the process has been bound to a virtual processor.

Socked Writes
The column labelled "sock_w" displays the number of times the process has written to a socket.

Socket Reads
The column labelled "sock_r" displays the number of times the process has read froma socket.

Signals Processed
The column labelled "sigs" displays the number of signals which have been handled by the process.

Voluntary Context Switches
The column labelled "v_switch" displays the number of times the process has been context switched through actions of its own, such a making a system call requiring I/O.

Involuntary Context Switches
The column labelled "iv_switch" displays the number of times the process has been context switch through reaching the end of its time slice.

Shared Anonymous RSS
The column labelled "shared anon rss" displays the size of the processes shared anonymous address space.

Private Data RSS
The column labelled "private data rss" displays the size of the processes private data address space.

Private Stack RSS
The column labelled "private stack rss" displays the size of the processes stack segment.

Private mapped
This column shows the size of the mapped segments private to the process.

Shared mapped
This column shows the size of the mapped segments shared by the process.

Total bytes mapped
This column shows the total size of the mapped segments used by the process.

Hard faults
This column shows the number of hard faults this process has caused.

Soft faults
This column shows the number of soft faults this process has caused.

Maximum RSS
The column labelled "max rss" shows the maximum size of the process resident set size.

File system input operations
The column labelled "fsin op" shows the number of file system input operations performed by the process.

File system output operations
The column labelled "fsout op" shows the number of file system output operations performed by the process.

Command

The command or program being run is shown in the column labelled "cmd". This field will change to display as much data as will fit on the screen. Using a wider screen will result in more arguments, etc., being displayed.

- End of Chapter -

Chapter 9

Scheduling Screens

The Scheduling screens contain information related to process scheduling and management. Two screens are provided with scheduling information: the Scheduling Screen and the Scheduling History Screen.

Accessing the Scheduling Screen

Press the **s** key to view the Scheduling Screen. This screen contains current and average counts and per second rates for a variety of scheduling parameters. The current statistics reflect the most recent cycle. The average figures are averaged since the time UX/RPM was started. This screen also shows the current CPU utilization.

```
Date: 12/10/93      SEPAC DG/UX Real-time Performance Monitor      Cycle:  10
Curr: 16:08:21              Scheduling Screen (s)                 Actual: 10
Start:16:04:59

                              Current      Average                    CPU
Bound Runnable Processes:          15           14
Unbound Runnable Processes:         0            0                   idl
Bound Processes:                   35           35                    48%
Unbound Processes:                 23           23
Eligible Processes:                 2            1                   sys
Processes:                         59           59                    39%
CPU Count:                          1            1
Process Table Overflows:            0            0                   usr
Process Binds/sec:                0.0          0.0                    13%
Process Unbinds/sec:              0.0          0.0
System Calls/sec:               640.4        380.1
fork System Calls/sec:            0.0          0.0
exec System Calls/sec:            0.0          0.0
read System Calls/sec:           66.9         57.9
write System Calls/sec:          26.6         13.1
Process Switches/sec:           288.9        147.1

[M]enu [H]elp Hist [S]                                  [C]ycle [Q]uit
```

Figure 9.1 - Scheduling Screen

Accessing the Scheduling History Screen

Press the S key to view the Scheduling History Screen. If your environment supports function keys, F3 toggles the Scheduling Screen and the Scheduling History Screen.

The Scheduling History Screen shows the maximum and minimum values observed since UX/RPM was started. The times these values were reached are also shown.

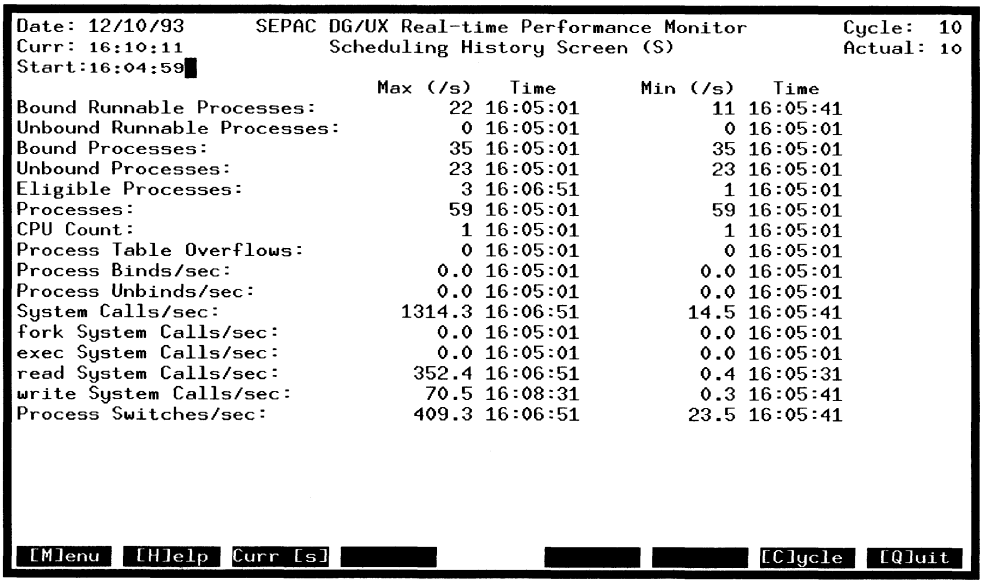

Figure 9.2 - Scheduling History Screen

Scheduling Statistics

An AViiON multiprocessor system is a *Symmetric Multiprocessing* (SMP) system. The key word here is *symmetric*, which means that the multiple CPUs (the job processors, or JPs) in an AViiON SMP system are seen by user programs and the DG/UX operating system as equivalent.

Processes are programs in execution. For process scheduling, the number of Bound Runnable processes can indicate, over time, the load of processes on the system. For a process to run and get CPU time, the DG/UX scheduler must perform a bind operation. Once bound, the process can then be scheduled to run on a CPU or Job Processor. Most systems will normally have more bound than unbound processes. Eligible processes are those bound runnable processes which are ready to run or are running on a CPU. As the number of eligible processes increases beyond the number of physical processors or CPUs, the system load will normally increase

An Unbound Runnable process is ready to run but has been unbound and placed on a queue waiting to be bound. A non-zero value for Unbound Runnable indicates an issue in terms of process scheduling. When the Unbound Runnable value is non-zero, either the DG/UX

Scheduler has determined the there is memory thrashing and reduced the number of processes that can be bound or the number of Virtual Processors (for DG/UX revision 2.10 and earlier) or Maximum Bound Transients (MAXBOUND in DG/UX revision 5.4R3.00 and later) configured on the system is insufficient.

With DG/UX revision 5.4R2.10 and earlier, processes are bound to Virtual Processors (VPs) and are then scheduled to run on CPUs. The number of VPs is determined by the DG/UX parameter NPROC (maximum number of processes on the system at one time) and the amount of memory or by the DG/UX parameter NVPS (number of virtual processors). If the Unbound Runnable processes value is non-zero and the Frames Purged rate is greater then 25/second, the Scheduler has more than likely reduced the available Virtual Processors. If this is the case, memory use is affecting performance of the system. If increasing the NPROC value does not cause the Unbound Runnable count to be 0, further analysis of memory usage of the environment is necessary. Examine the amount of memory consumed by each process running on the system to determine whether any process has a large resident or swap size in relation to other processes.

With DG/UX revision 5.4R3.00 and later, processes run as Light Weight Process (LWPs) groups which in turn can run one or more LWPs. When LWPs are running in the DG/UX kernel, transient data sections must be bound to the LWPs so that state information is stored. DG/UX dynamically adjusts the number of bound transient data sections that are available. If the scheduler determines that memory is thrashing through monitoring the program page reclaim rate, DG/UX will reduce the available number of transient data sections available to be bound. If the current number of required transients exceeds the current available, there will be unbound runnable processes on the scheduling screen. To resolve this issue, first try increasing the DG/UX parameter MINBOUND to equal the value for NPROC. If this does not resolve the issue, evaluate the environmental memory usage by examining the memory demands of on-line processes. Note - DG/UX revision 3.00 without patch 70 reports the number of unbound runnable processes as number of CPUs + 2. This problem is resolved by installing patch 70 or going to DG/UX revision 5.4R3.10.

Screen Fields

Each field is described below.

Process Statistics

Bound Runnable Processes

This is the current and average number of processes that are ready to run and are bound to a Virtual Processor (DG/UX revision 5.4R2.10 and earlier) or Transient Data Section (DG/UX 5.4R3.00 and later).

Unbound Runnable Processes

This value indicates the current and average number of processes that are ready to run, but are not bound. If this value is consistently above zero then your system does not have enough

Virtual Processors or Transient Data Sections or the DG/UX scheduler has reduced the number of processes that can be bound.

Bound Processes
This is the number of processes that are bound.

Unbound Processes

This the number of processes that are not bound.

Eligible Processes
The number of eligible bound runnable processes. These processes are either running on or are queued to run on a physical Job Processor (CPU).

Processes
This is the total number of processes in existence (currently and average).

CPU Count
This is the number of active physical processors. This is always exactly equal to the number of Job Processors present and does not change.

Process Table Overflows
The number of times that an attempt to create a process failed due to reaching the maximum number of processes allowed. If this value is consistently greater than zero, you should increase the kernel configuration parameter NPROC.

Process Binds
The per-second rate at which processes are being bound.

Process Unbinds
The per-second rate at which processes are being unbound.

System Call Rates
The per-second rates for system calls are shown in this section. The total of all system calls is given first, followed by the rates for fork(2), exec(2), read(2) and write(2) calls.

Process Switch Rates
The rate at which new processes are being loaded onto JPs.

- End of Chapter -

Chapter 10

Virtual Memory Screens

You can view two types of Virtual Memory screens to examine virtual memory statistics: The Virtual Memory Screen and the Virtual Memory History Screen. In addition, you can display four different sets of data for Memory screens.

Accessing the Virtual Memory Screens

The Virtual Memory screen shows current and average values; the Virtual Memory History screen shows minimum and maximum values.

To Display Current and Average Values

Press the **v** key to access the Virtual Memory Screen. This screen shows the current and average values for statistics related to memory management.

```
Date: 06/30/94      SEPAC DG/UX Real-time Performance Monitor      Cycle:  10
Curr: 14:23:32           Virtual Memory Screen (v) Std (s)        Actual: 10
Start:14:23:11

                          Current(/s)  Average(/s)                    CPU
Message Operations:               0.0          0.0
Semaphore Operations:             0.0          0.0                    usr
Frames Purged:                    0.0          0.0                     0%
Bound Frames Purged:              0.0          0.0
User Page Faults:                27.3         53.6                    sys
Hard Page Faults:                 0.0          0.1                     5%
Copy-On-Write Page Faults:        0.0          0.5
Fill-From-File Page Faults:       0.0          0.1                    idl
                                                                      95%

                                Current      Average
Physical Memory (k):              36864
Free Memory (frames):              4737         4740
Total Anonymous (frames):         20426        20426
Free Anonymous (frames):          14333        14333
Reserved Anonymous (frames):       6093         6093
Inode Table Size:                  1190         1190
File Table Size:                    319          319
Lock Table Size:                      0            0

 [M]enu  [H]elp  Hist [V]                    [W]indow [C]ycle  [Q]uit
```

Figure 10.1 - Virtual Memory Screen

To Display Minimum and Maximum Values

Press the **V** key to access the Virtual Memory History Screen. If your environment supports function keys, the F3 key toggles the Virtual Memory Screen and the Virtual Memory History Screen.

The Virtual Memory History Screen shows the minimum and maximum values observed since UX/RPM was started.

```
Date: 06/30/94      SEPAC DG/UX Real-time Performance Monitor        Cycle:  10
Curr: 14:25:50        Virtual Memory History Screen (V) Std (s)      Actual: 10
Start:14:23:11
                              Max (/s)   Time       Min (/s)   Time        CPU
Message Operations:                0.0 14:23:12          0.0 14:23:12
Semaphore Operations:              0.0 14:23:12          0.0 14:23:12      usr
Frames Purged:                     0.0 14:23:12          0.0 14:23:12       2%
Bound Frames Purged:               0.0 14:23:12          0.0 14:23:12
User Page Faults:                231.4 14:23:12         25.0 14:24:20      sys
Hard Page Faults:                  6.2 14:23:42          0.0 14:23:12       6%
Copy-On-Write Page Faults:         5.7 14:23:42          0.0 14:23:12
Fill-From-File Page Faults:        5.7 14:23:42          0.0 14:23:12      idl
                                                                           92%
                                    Max   Time             Min   Time
Physical Memory (k):              36864
Free Memory (frames):              4748 14:23:12          4559 14:24:20
Total Anonymous (frames):         20426 14:23:12         20426 14:23:12
Free Anonymous (frames):          14333 14:23:12         14286 14:23:42
Reserved Anonymous (frames):       6140 14:23:42          6093 14:23:12
Inode Table Size:                  1199 14:23:42          1190 14:23:12
File Table Size:                    323 14:23:42           319 14:23:12
Lock Table Size:                      0 14:23:12             0 14:23:12

 [M]enu  [H]elp  Curr [v]                          [W]indow [C]ycle  [Q]uit
```

Figure 10.2 - Virtual Memory History Screen

To Change Column Data Display

Additional flexibility is available with the Memory screens in that you may choose between four different displays. Press the **w** key while viewing a Memory screen. The system displays a pop-up window (Figure 10.3).

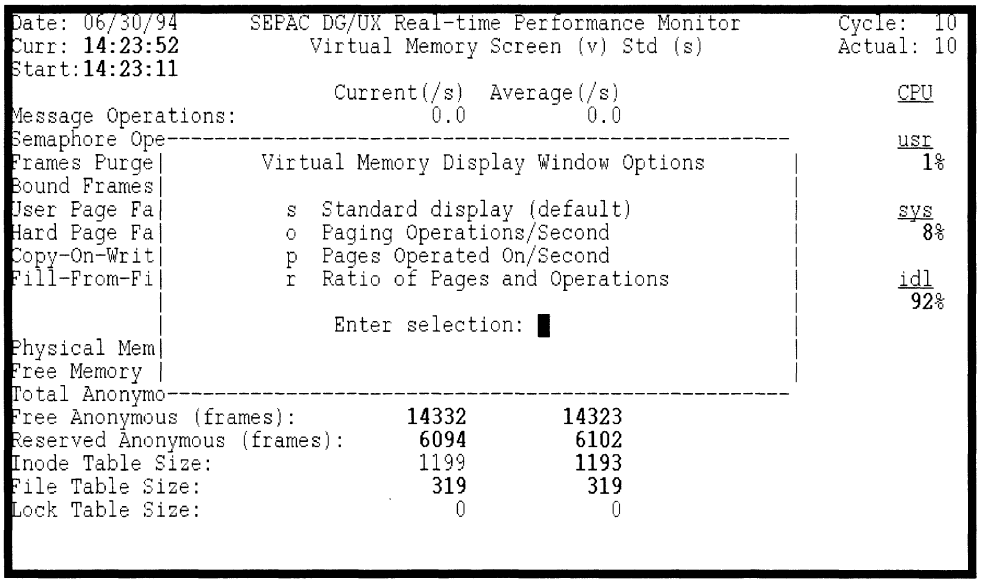

Figure 10.3 - Pop-up Window

At the ENTER SELECTION prompt, type the letter for the display you want to see:

- **s** Standard display. This is the display seen in UX/RPM 2.10 (shown in Figure 10.2).
- **o** Paging operations per second (Figure 10.4).
- **p** Pages involved in paging activities per second. This display uses the same format as shown in Figure 10.4.
- **r** The ratio of pages per paging operation. This display uses the same format as shown in Figure 10.4.

Paging Operations per Second Screen

The paging operations per second screen (Figure 10.4) displays a matrix of information. The (p) and (r) screens use the same format. Across the top are categories of memory usage. Along the side are types of activity. The very bottom row shows the number of pages in memory for each category mentioned across the top.

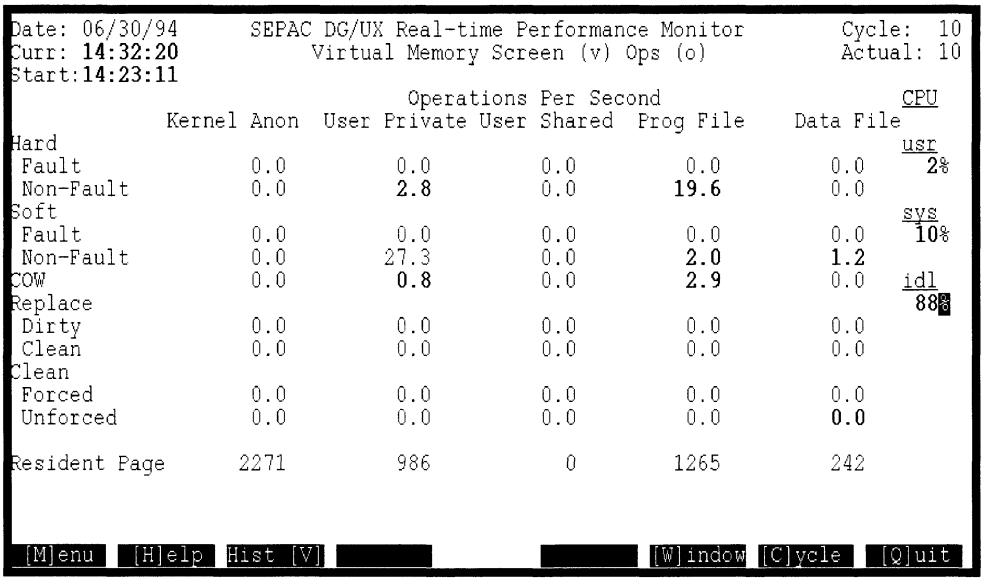

Figure 10.4 - Paging Operations per Second

Virtual Memory Data

In the DG/UX operating system, application programs and files, as well as kernel programs (text), are demand paged into main memory (from disk) by the Virtual Memory Manager (VMM).

In the Demand Paged File I/O Model, virtually all of a computer's memory is treated as a cache for programs and files. Unlike some UNIX implementations that still use Virtual Memory for programs and a buffer cache for files, the DG/UX operating system uses the kernel data cache only for file system metadata. Files system metadata is "data about data" - data that the File System uses to describe and locate files (such as inodes).

A *page* is the logical granularity at which the DG/UX Virtual Memory system allocates and transfers data. A *frame*, in the context of main memory, is a container that holds a page or part of a page. In current AViiON computers, the frame size is 4 Kbytes. Page size must be a multiple of frame size, and in DG/UX, the page size is 4 Kbytes (1 X frame size).

Each process has its own page table, which contains the mappings for the process' address space pages and physical memory frames. The kernel creates the page table when it starts the process.

When the Virtual Memory Manager (VMM) receives a request for memory to be mapped (such as a mmap(2) system call), the VMM sets up in the process' page table the mappings of virtual pages to file pages. The VMM does not read the pages from disk yet - it only sets up the mappings. As the process references particular pages, hardware page faults cause the VMM to perform "page in" operations to read the pages from disk.

Page faults can be satisfied in several ways:

- by simply changing pointers in a process' page table (a soft page fault),
- by reading a page from disk (a hard page fault), or
- by reading a page into a frame from the swap area.

Screen Fields

Each field displayed on Virtual Memory screens is described below.

Operations

Message Operations

This is the number of message operations performed per second via the msgsnd(2) call.

Semaphore Operations

The number of semaphore operations per second performed by the semop(2) call.

Frames

Frames Purged

The rate at which frames are being purged by the frame purger daemon. This figure includes the number of bound frames purged.

Bound Frames Purged

The rate at which frames which belonged to bound processes are being reclaimed by the frame purger daemon. If this value is consistently greater than zero, it is an indication that your system does not have enough main memory to handle the current processing mix.

Page Faults

User Page Faults

The rate at which page faults are being taken by processes running in user space. This measurement includes hard page faults, copy on write faults, and fill from page faults. It does not include page faults taken by the operating system kernel.

Hard Page Faults

The rate at which faults are taking place that require disk I/O. Hard faults are different from soft page faults. Soft page faults are satisfied when the VMM can simply change a frame pointer in a process' page table. A typical hard page fault requires on the order of 25 milliseconds to complete - a soft page fault requires on the order of 2 milliseconds to complete.

Copy On Write Page Faults

This is the rate at which Copy On Write (COW) page faults are occurring. Copy On Write is a Virtual Memory accelerator technique which can significantly reduce the amount of disk I/O when processes fork other processes. The Copy On Write operations are invisible to the user processes.

When a process forks a process, the new process shares the parent process' page table entries - rather than copies of all the parent process' pages. Multiple processes can read and execute the shared page. The VMM makes a copy of the parent process' page (and updates the child process' page table) only when the either the parent or the child process writes to the page.

Fill From File Page Faults

This item shows the rate at which page faults are being satisfied by page-in operations from a file system. Page faults that result in a page-in operation from a file system are in contrast to page faults that are satisfied by page-in operations from the swap area. The swap area stores "anonymous" data, which is not associated with a file. An example of anonymous data is data in a FORTRAN array.

Free Memory

This shows the number of free memory frames in the free memory pool.

Free Swap

This is the number of free 512 byte blocks in the swap area which is used to store pages that contain anonymous data.

Table Sizes

Inode Table Size

This the current and average number of inodes that are in use.

File Table Size

This is the number of object pointers in use. This number is roughly equivalent to the number of files in use, but also includes object pointers to such things as sockets and pipes.

Lock Table Size

The number of shared memory data segments in use.

Kernel Anon

The anonymous memory allocated to the kernel.

User Private

The anonymous memory allocated to individual processes.

User Shared

The anonymous memory allocated to more than one process.

Prog File

The memory holding program file pages.

Data File

The memory holding non-program file pages.

- End of Chapter -

Appendix A

DIF File Field Descriptions

During logfile conversion, UX/RPM outputs the following DIF files:

Default Name	Description
uxrpmhdr.dif	Header Record
uxrpmsin.dif	System Incremental Snapshots
uxrpmsum.dif	System Cumulative Snapshots
uxrpmdin.dif	Disk Incremental Snapshots
uxrpmdum.dif	Disk Cumulative Snapshots
uxrpmvin.dif	Virtual Disk Incremental Snapshots
uxrpmvum.dif	Virtual Disk Cumulative Snapshots

DIF stands for Data Interchange Format. It is a portable format for spreadsheet data. Nearly all spreadsheet packages can read DIF files. The DIF files output by UX/RPM have as the first row descriptive labels. The labels define the data appearing in each column. These labels are listed and described below for each file.

==========

Label: RPM header data File: uxrpmhdr.dif

Fields: 19

"RPM Revision"	This field holds the numeric revision number of RPM which was used to collect the data. The current value is 210.
"OS Revision"	This field holds the numeric revision number of the running DG/UX under which the data was collected.
"Log Level"	This field holds the log level selected when the data was collected.

"Number of Disks" This field holds the number of disks for which data was collected. Note that if a log level which collected no disk information was selected, that this will be 0.

"Number of Usernames" This field holds the number of username entries for which data was collected. Note that if a log level which collected no process information was selected, that this will be 0. Note that username entries are not output at this time.

"Number of TTY Entries" This field holds the number of ttyname entries for which data was collected. Note that if the -t switch was not selected, that this will be 0. Note that TTY entries are not output at this time.

"Boot Time" The wall clock time which has passed since the system was rebooted. This value is a string in the form "Dow Mon DD HH:MM:SS YYYY", for example "Tue Nov 16 11:13:05 1993"

"Total PMem" The total physical memory (in Kilobytes) present in the machine.

"Available UMem" The available user memory (in Kilobytes) present in the machine.

"NFS flag" A flag indicating the presence of NFS support in the kernel. Its value is 1 or 0.

"Model id" A number which indicates the type of machine the data was collected for.

"PROM revision" The revision number of the firmware.

"Number of processors"
The number of job processors (JPs) on the machine.

"Timezone Offset" The offset for the timezone this data was collected in. This value is in the form "HH:MM:SS" or "-HH:MM:SS". This value is subtracted from GMT to yield local time.

"Alternate Timezone Offset"
The offset for the daylight savings time timezone this data was collected in. This uses the same format as Timezone Offset.

"Daylight savings time flag"
A flag indicating whether or not the Alternate timezone offset is valid or not. Its value is 1 or 0.

"Hostname"	The hostname of the system the data was collected on.
"Cycle time"	The number of seconds between collections for this data set.
"CPU ID"	The CPU Id of the system the data was collected on. This value is a hexadecimal number.

==========

Label: RPM snapshot data File: uxrpmsum.dif

Fields: 113

"Cycle Number"	The number of the individual snapshot.
"Current Time"	The time at the time of the snapshot.
"Message Operations Count"	The number of messages sent via the msgsnd(2) call.
"Semaphore Operations Count"	The number of semaphore operations performed via the semop(2) call.
"Binds"	The number of processes bound.
"User Page Faults"	The number of page faults associated with user address spaces.
"Unbinds"	The number of processes unbound.
"Bound Frames purged"	The number of frames reclaimed by the frame purger demon that belonged to bound processes.
"Hard Page Faults"	The number of page faults that resulted in physical I/O.
"COW Page Faults"	The number of copy on write page faults, these generally happen after a fork().

"Fill from file Page Faults" The total number of page faults satisfied by a page in from the file system.

"Frames Purged" The number of frames freed for use by the frame purger demon.

"Free Memory" The number of free memory frames.

"Free Swap" The number of free frames in the paging area.

"Total anonymous pages" The system's total anonymous memory (swap space, in pages). Includes swap areas plus much of main memory. Not reported prior to DG/UX 5.4 Release 2.

"Reserved anonymous pages"
The amount of anonymous memory (swap space, in pages) reserved for use system-wide. Total minus reserved indicates how close the system is to rejecting memory allocation requests (e.g. mallocs). Not reported prior to DG/UX 5.4 Release 2.

"Snapshot Time" Same as Current Time.

"User Time" Amount of user processor time spent by processes.

"System Time" Amount of system processor time spent by processes.

"I/O Wait Time" Amount of time the system has spent waiting on I/O operations. This is always 0 under DG/UX.

"Idle Time" Amount of time the system has spent idle.

"Bound runnable" Number of processes which are ready to run and have been assigned to a virtual processor.

"Unbound runnable" Number of processes which are ready to run, but have not been assigned to a virtual processor.

"Bound processes" Number of processes which have been assigned to a virtual processor.

"Unbound processes" Number of processes which have not been assigned to a virtual processor.

"Eligible processes" The number of eligible bound runnable processes per JP at the time the dg_sys_info call is made. An eligible processes is a process that will make forward progress given a JP and is not suspended waiting on some event.

"CPU Count"" The number of active physical processors.

"Process count" The number of processes existing on the system at the time of the snapshot.

"Process Table Size" The configured maximum size of the process table. This is the maximum number of process possible on this system.

"Process Table Overflow Count"
The number of times an attempt to create a new process failed because the process table was full.

"System call count" The number of system calls made.

"Fork count" The number of fork system calls made.

"Exec count" The number of exec system calls made.

"Read count" The number of read system calls made.

"Write count" The number of write system calls made.

"Chars read" The number of bytes of data handled by the read() call.

"Chars written" The number of bytes of data handled by the write() call.

"Inode table size" The number of entries in the inode table.

"File table size" The number of entries in the file table.

"Lock table size" The number of shared memory data segments in use.

"Process switch count" The total number of times a new process has been loaded onto a JP.

"IGet count" The number of inodes initialized.

"Pathname search count"	The number of times a pathname has been resolved.
"Directory block reads"	The total number of reads associated with buffer portion of a directory file.
"Physical Blocks read"	The total number of physical blocks read into the system buffers.
"Physical Blocks written"	The total number of physical blocks written from the system buffers.
"Logical Blocks read"	The total number of logical blocks read from system buffers.
"Logical Blocks written"	The total number of logical blocks written from system buffers.
"Physical Read requests"	The total number of reads made to disk excluding block special I/O.
"Physical Write requests"	The total number of writes made to disk excluding block special I/O.
"Raw input char count"	The total number of characters read in raw mode from terminal devices.
"Canon input char count"	The total number of characters read from terminal devices and processed in canonical mode.
"Output char count"	The total number of characters output to terminal devices.
"RX interrupt count"	The total number of receive interrupts from terminal devices.
"TX interrupt count"	The total number of transmit interrupts to terminal devices.
"Modem interrupt count"	The total number of modem interrupts.
"RPC server calls"	Total number of RPC calls received by the server.
"RPC server bad calls"	Total number of RPC calls rejected by the server.
"RPC server null recv"	Number of times that no RPC packet was availablewhen trying to receive an RPC request on the server.
"RPC server bad length"	Number of times that an RPC packet with an invalid length was received by the server.

"RPC server xdr calls" Number of times that an RPC packet with a malformed header was received by the server.

"RPC client calls" Total number of RPC calls made by the client.

"RPC client bad calls" Total number of RPC calls made by the client that were rejected by the server.

"RPC client retransmission" Number of times that an RPC call had to be retransmitted by the client.

"RPC client bad xid" Number of times that a reply to a client RPC call did not match what the client sent.

"RPC client timeout" Number of times that a call was deferred because a client handle was busy.

"RPC client wait" Number of times that a reply to a call was not received within the period allowed by the timeout value.

"RPC client new cred" Number of times that the client had to refetch authentication information.

"NFS server calls" Number of NFS calls received by the NFS server(s).

"NFS server bad calls" Number of NFS calls rejected by the NFS server(s).

"NFS server procedure calls[]"
An array of counts, one entry for each of the NFS operations supported by the NFS server. Each entry indicates the total number of corresponding operations performed by the server.

"NFS client sleeps" Number of times that a client NFS call had to be deferred because resources were not available.

"NFS client gets" Number of times a new client handle had to be created.

"NFS client calls" Total number of NFS client calls.

"NFS client bad calls" Total number of NFS client calls that were not received or were rejected.

"NFS client procedure calls[ ]"
An array of counts, one entry for each of the NFS operations supported by the NFS client. Each entry indicates the total number of corresponding operations performed by the client.

"One minute" The one minute load average as reported by DG/UX.

"Five minutes" The five minute load average as reported by DG/UX.

"Fifteen minutes" The fifteen minute load average as reported by DG/UX.

"User Count" The number of users logged in.

==========

Label: RPM incremental snapshot data File: uxrpmsin.dif

Fields: 117

"Cycle Number" The number of the individual snapshot.

"Current Time" The time at which the snapshot was taken.

"Message Operations Count/Sec"
The number of messages sent per second via the msgsnd(2) call.

"Semaphore Operations Count/Sec"
The number of semaphore operations performed per second via the semop(2) call.

"Binds/Sec" The number of processes bound per second.

"User Page Faults/Sec" The number of page faults per second associated with user address spaces.

"Unbinds/Sec" The number of processes unbound per second.

"Bound Frames purged/Sec"
The number of frames reclaimed per second by the frame purger demon that belonged to bound processes.

"Hard Page Faults/Sec" The number of page faults per second that resulted in physical I/O.

"COW Page Faults/Sec" The number of copy on write page faults per second.

"Fill from file Page Faults/Sec"
The total number of page faults per second satisfied by a page in from the file system.

"Frames Purged/Sec" The number of frames per second freed for use by the frame purger d
emon.

"Free Memory" The number of free memory frames.

"Free Swap" The number of free frames in the paging area.

"Total anonymous pages" The system's total anonymous memory (swap space, in page s). Includes swap areas plus much of main memory. Not reported prior to DG/UX 5.4 Release 2.

"Reserved anonymous pages"
The amount of anonymous memory (swap space, in pages) reserved for use system-wide. Total minus reserved indicates how close the system is to rejecting memory allocation requests (e.g. mallocs). Not reported prior to DG/UX 5.4 Release 2.

"Snapshot Length" The duration in seconds of the current snapshot.

"User Time" Amount of user processor time spent by processes during this snapshot.

"System Time" Amount of system processor time spent by processes during this snapshot.

"I/O Wait Time" Amount of time the system has spent waiting on I/O opera tions during this snapshot. This is always 0 under DG/UX.

"Idle Time" Amount of time the system has spent idle during this snapshot.

"Total CPU Time" Sum of the user and system time for this snapshot.

"Percent User Time" The percent of the available CPU which was spent as user processor time by processes during this snapshot.

"Percent System Time" The percent of the available CPU which was spent as system processor time by processes during this snapshot.

"Percent Idle Time" The percent of the available CPU which was spent as idle processor time by the system during this snapshot.

"Bound runnable" Number of processes which are ready to run and have been assigned to a virtual processor.

"Unbound runnable" Number of processes which are ready to run, but have not been assigned to a virtual processor.

"Bound processes" Number of processes which have been assigned to a virtual processor.

"Unbound processes" Number of processes which have not been assigned to a virtual processor.

"Eligible processes" The number of eligible bound runnable processes per JP at the time the dg_sys_info call is made. An eligible processes is a process that will make forward progress given a JP and is not suspended waiting on some event.

"CPU Count" The number of active physical processors.

"Process count" The number of processes existing on the system at the time of the snapshot.

"Process Table Size" The configured maximum size of the process table. This is the maximum number of process possible on this system.

"Process Table Overflow Count"
The number of times per second an attempt to create a new process failed because the process table was full.

"System call count/Sec" The number of system calls made per second.

"Fork count/Sec" The number of fork system calls made per second.

"Exec count/Sec" The number of exec system calls made per second.

"Read count/Sec" The number of read system calls made per second.

"Write count/Sec" The number of write system calls made per second.

"Chars read/Sec" The number of bytes per second of data handled by the read() call.

"Chars written/Sec" The number of bytes per second of data handled by the write() call.

"Inode table size" The number of entries in the inode table.

"File table size" The number of entries in the file table.

"Lock table size" The number of shared memory data segments in use.

"Process switch count/Sec"
The number of times per second a new process has been loaded onto a JP.

"IGet count/Sec" The number of inodes initialized per second.

"Pathname search count/Sec"
The number of times per second a pathname has been resolved.

"Directory block reads/Sec"
The number of reads per second associated with buffer portion of a directory file.

"Physical Blocks read/Sec" The number of physical blocks per second read into the system buffers.

"Physical Blocks written/Sec"
The number of physical blocks per second written from the system buffers.

"Logical Blocks read/Sec" The number of logical blocks per second read from system buffers.

"Logical Blocks written/Sec"
The number of logical blocks per second written from system buffers.

"Physical Read requests/Sec"
The number of reads per second made to disk excluding block special I/O.

"Physical Write requests/Sec"
The number of writes per second made to disk excluding block special I/O.

"Raw input char count/Sec"
The number of characters per second read in raw mode from terminal devices.

"Canon input char count/Sec"
The number of characters per second read from terminal devices and processed in canonical mode.

"Output char count/Sec" The number of characters per second output to terminal devices.

"RX interrupt count/Sec" The number of receive interrupts per second from terminal devices.

"TX interrupt count/Sec" The number of transmit interrupts per second from terminal devices.

"Modem interrupt count/Sec"
The number of modem interrupts per second.

"RPC server calls/Sec" Number of RPC calls per second received by the server.

"RPC server bad calls/Sec" The number of RPC calls per second rejected by the server.

"RPC server null recv/Sec" The number of times per second that no RPC packet was availablewhen trying to receive an RPC request on the server.

"RPC server bad length/Sec"
The number of times per second that an RPC packet with an invalid length was received by the server.

"RPC server xdr calls/Sec" The number of times per second that an RPC packet with a malformed header was received by the server.

"RPC client calls/Sec" The number of RPC calls per second made by the client.

"RPC client bad calls/Sec" The number of RPC calls per second made by the client that were rejected by the server.

"RPC client retransmission/Sec"
The number of times per second that an RPC call had to be retransmitted by the client.

"RPC client bad xid/Sec" The number of times that a reply to a client RPC call did not match what the client sent.

"RPC client timeout/Sec" The number of times that a call was deferred because a client handle was busy.

"RPC client wait/Sec" The number of times that a reply to a call was not received within the period allowed by the timeout value.

"RPC client new cred/Sec" The number of times that the client had to refetch authentication information.

"NFS server calls/Sec" The number of NFS calls received by the NFS server(s).

"NFS server bad calls/Sec" The number of NFS calls rejected by the NFS server(s).

"NFS server procedure calls[ ]"
An array of counts, one entry for each of the NFS operations supported by the NFS server. Each entry indicates the number of corresponding operations per second performed by the server.

"NFS client sleeps/Sec" The number of times per second that a client NFS call had to be deferred because resources were not available.

"NFS client gets/Sec" The number of times per second that a new client handle had to be created.

"NFS client calls/Sec" The number of NFS client calls per second.

"NFS client bad calls/Sec" The number of NFS client calls per second that were not received or were rejected.

"NFS client procedure calls[ ]"
An array of counts, one entry for each of the NFS operations supported by the NFS client. Each entry indicates the number of corresponding operations per second performed by the client.

"One minute" The one minute load average as reported by DG/UX.

"Five minutes"	The five minute load average as reported by DG/UX.
"Fifteen minutes"	The fifteen minute load average as reported by DG/UX.

==========

Label: RPM disk data File: uxrpmdum.dif
also: RPM virtual disk data File: uxrpmvum.dif

Fields: 2 + (8 * # of disks)

"Cycle Number"	The number of the individual snapshot.
"Current Time"	The time at the time of the snapshot.

The following fields repeat for each disk reported.

"Disk Number"	The number of the disk reported.
"Disk Name"	The long name of the disk (from /dev/pdsk) reported.
"Read Block Count"	The number of 512-byte blocks that have been read from the disk unit.
"Write Block Count"	The number of 512-byte blocks that have been written to the disk unit.
"Read Request Count"	The total number of read requests that have been handled by the disk unit.
"Write Request Count"	The total number of write requests that have been handled by the disk unit.
"Busy Time"	The total amount of time that the disk unit has been busy working on requests.
"Response Time"	The total amount of time that requests to the unit have spent waiting to be completed. This includes time spent on various driver queues.

==========

Label: RPM incremental disk data File: uxrpmdin.dif
also: RPM incremental virtual disk data File: uxrpmvin.dif

Fields: 2 + (14 * # of disks)

"Cycle Number" The number of the individual snapshot.

"Current Time" The time at the time of the snapshot.

The following fields repeat for each disk reported.

"Disk Number" The number of the disk reported.

"Disk Name" The long name of the disk (from /dev/pdsk) reported.

"Read Block Count/Cycle" The number of 512-byte blocks that have been read from the disk unit within this snapshot.

"Write Block Count/Cycle" The number of 512-byte blocks that have been written to the disk unit within this snapshot.

"Read Request Count/Cycle"
The total number of read requests that have been handled by the disk unit within this snapshot.

"Write Request Count/Cycle"
The total number of write requests that have been handled by the disk unit within this snapshot.

"Busy Time" The amount of time that the disk unit has been busy working on requests within this snapshot.

"Response Time" The amount of time that requests to the unit have spent waiting to be completed within this snapshot. This includes time spent on various driver queues.

"Total Request Count" The sum of the the read and write request counts for this cycle.

"Request Percentage" The percentage of the requests for all drives received by this drive.

"Busy Percentage"	The percentage of time spent busy by this disk during this cycle.
"Average Queue"	The average length of the service queue: Response Time / Busy Time.
"Average Service"	The average time in milliseconds it took to service a request.
"Average Response"	The average time in milliseconds it took to respond to a request.

Appendix B

Quick-Reference Tables

This appendix contains tables that show default files used by UX.RPM, valid UX/RPM switches, and valid UX/RPM command keys.

Table 2 describes various files used as input to UX/RPM or are created as output by UX/RPM.

.

Table 2: UX/RPM Files

File Description	Filename/Comments
ASCII text file that lists physical disk numbers to exclude from data collection	No default; user-supplied. Specified with the -d switch.
ASCII text file to which UX/RPM writes screens upon termination	rpm.lst Cannot be changed.
Files output as a result of logfile conversion	Defaults to uxrpm*identifier*.dif *identifier* is supplied by UX/RPM and currently is hdr, sin, sum, din, dum, vin, and vum. All five files are created for each conversion. The "uxrpm" prefix can be changed by using "-C -F *prefix*" on the command line.
Logfiles to which UX/RPM writes collected data	Defaults to: rpm.*<mm>.<dd>[.<hh>.<min>.<sec>]*.log and rpm.*<mm>.<dd>[.<hh>.<min>.<sec>]*.idx. UX/RPM appends the date stamp. Hours, minutes, and seconds are supplied by UX/RPM as needed for unique filename identification. The filename (except the .log and .idx extensions) can be changed by using "-l *level* -f *filename*" on the command line. Do not include the .log or .idx extensions when specifying a logfile to view in playback mode.

Table 3 alphabetically lists valid UX/RPM default command keys. Some keys provide multiple functionality, depending on the task you are performing. In such cases, each funtionality is enumerated.

Table 3: UX/RPM Command Keys

Keystroke	Function
!	Forks a DG/UX shell from within UX/RPM
/	Searches for a text string
?	Displays on-line Help
1G	Goes to the top of the current list display
j	Moves the current display down one line
a	**1.** Displays only active processes; for Process screens only. **2.** If you are changing the window display while viewing Process screens, this key displays Process arguments.
A	Displays ALL processes; for Process screens only.
b	Resets accumulated/history values
CTRL-b	Moves up one screen; for scrollable screens only.
c	**1.** Changes UX/RPM's cycle time **2.** If you are changing the window display while viewing Process Screens, this key displays user, system, and CPU time data **3.** If you are changing the sort order for Process Screens, this key specifies the sort order to be by CPU Time
d	Displays Disk Screen - Incremental
D	Displays Disk Screen - Accumulated
f	Pauses and unpauses data collection
CTRL-f	Moves down one screen; for scrollable screens only.
G	Goes to the end of the current list display
h	Displays on-line Help for UX/RPM switches
i	Displays I/O Screen
I	Displays I/O History Screen
k	Moves the current display up one line
l	Writes the current display to a disk file
L	Writes all screens to a disk file
m	**1.** Displays the Screen Menu **2.** If you are changing the window display while viewing Process Screens, this key displays memory and wait data

Table 3: UX/RPM Command Keys

Keystroke	Function
n	**1.** Displays Network Screen **2.** If you are changing the sort order for Process Screens, this key specifies the sort order to be by Nice Value
N	Displays Network History Screen
o	**1.** Displays System Overview Screen **2.** If you are changing the window display while viewing Virtual Memory screens, this key displays paging operations per second
p	**1.** Displays Process Screen - Incremental **2.** If you are changing the sort order for Process screens, this key specifies the sort order to be by Process ID **3.** If you are changing the window display while viewing Process screens, this key displays Process scheduling (for DG/UX 5.4R3.0) **4.** If you are changing the window display while viewing Virtual Memory screens, this key displays paging activity data
P	Displays Process Screen - Accumulated
Q, q	Exits UX/RPM
r	**1.** Reorders the sort order for process displays; valid only from Process Screens **2.** If you are changing the sort order for Process Screens, this key specifies the sort order to be by Priority **3.** If you are changing the window display while viewing Virtual Memory screens, this key displays the ratio of pages per paging operation
s	**1.** Displays Scheduling Screen **2.** If you are changing the window display while viewing Process screens, this key returns to the default Process Screen display **3.** If you are changing the sort order for Process Screens, this key specifies the sort order to be by Resident Process Size
S	Displays Scheduling History Screen
t	Specifies a specifiy snapshot to examine; valid only from the System Overview Screen
v	**1.** Displays Virtual Memory Screen **2.** If you are changing the window display while viewing Process screens, this key displays Process virtual memory data (for DG/UX 5.4R3.0)
V	Displays Virtual Memory History Screen
w	**1.** Allows you to change the window displays (that is, column data) for Process screens and Virtual Memory screens **2.** If you are changing the window display while viewing Process screens, this key changes the display to wide-screen display

.Table 4 alphabetically lists valid UX/RPM switches and indicates valid arguments.

Table 4: UX/RPM Switches

Switch	Function
-b *snapshot*	Specifies the beginning snapshot in a range when converting the logfile. *snapshot* format is either simply a positive integer or "HH:MM:SS MM/DD/YY".
-c *n*	Specifies the cycle time to be *n* seconds. *n* must be between 0 and 999, inclusive.
-C *format*	Converts the default logfile to the *format* specified; currently, the only valid entry for *format* is **d**.
-d *filename*	Excludes the physical disk numbers listed in *filename* from data collection. *filename* must be the name of a text file.
-e *snapshot*	Specifies the ending snapshot in a range when converting the logfile. *snapshot* format is either simply a positive integer or "HH:MM:SS MM/DD/YY".
-E	Specifies that an express logfile conversion is to be performed. Recommended method for converting logfiles.
-f *logfile*	**1.** If used without -l or -E, specifies the logfile to use for playback mode. **2.** If used with the -l switch, specifies the name of the logfile to be created. **3.** If used with the -E switch, specifies the logfile to convert.
-F *prefix*	When used with the -E switch, specifies the first part of the output filenames generated during conversion. If *prefix* is 5 or fewer characters, the resulting filenames are compatible with DOS naming conventions.
-g *drawing*	Specifies that screen displays should use line drawing graphics. Currently, *drawing* must be either **l** (for limited) or **f** (for full) line drawings.
-h	Displays a usage message and exits UX/RPM.
-l *level*	Specifies logging is to be done. Currently, *level* must be **0** (no logging), **1** (exclude disk and process data), **2** (exclude process data), or **3** (all information).
-n *cycles*	Terminates UX/RPM after the specified number of cycles have been performed. *cycles* must be a positive integer.
-s *screen*	Specifies the screen UX/RPM that should display when first executed. *screen* must be a valid screen command key.
-t	Resolves TTY names for Process screens.
-L	Writes the contents of all UX/RPM information screens to a disk file (rpm.lst) when UX/RPM terminates.

Reader's Comment Form

Data General's SEPAC group welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revisions.

Name of manual: _____

About the manual:

Did you find any errors?

Is the information clearly presented?

Are the topics logically organized?

Do you need more information? If so, where?

Can you easily find what you want?

What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the topic, chapter, and page number below:

_____

_____

_____

_____

_____

_____

_____

Please send your comments to:

SEPAC
Data General Corporation
4170 Ashford-Dunwoody Rd., Suite 300
Atlanta, GA, 30319
(404) 705-2520

If you would like a reply, please give your name, address, and telephone number below.

_____

_____

_____

_____

Thank you for helping us improve our documentation.

Quick-Reference Card for UXRPM

COMMAND LINE SYNTAX

) **rpm** *[-switch switch_value] ..* (↵)

Switches

-b *snapshot*

Specifies the beginning snapshot in a range when converting the logfile. *snapshot* format is either simply a positive integer or "HH:MM:SS MM/DD/YY".

-c *n*

Specifies the cycle time to be *n* seconds. *n* must be between 0 and 999, inclusive.

-C *format*

Converts the default logfile to the *format* specified; currently, the only valid entry for *format* is **d**.

-d *filename*

Excludes the physical disk numbers listed in *filename* from data collection. *filename* must be the name of a text file.

-e *snapshot*

Specifies the ending snapshot in a range when converting the logfile. *snapshot* format is either simply a positive integer or "HH:MM:SS MM/DD/YY".

-E

Specifies that an express logfile conversion is to be performed. Recommended method for converting logfiles.

-f *logfile*

1. If used without -l or -E, specifies the logfile to use for playback mode.
2. If used with the -l switch, specifies the name of the log file to be created.
3. If used with a conversion switch, specifies the logfile to convert.

-F *prefix*

When used with the -E switch, specifies the first part of the output filenames generated during conversion. If *prefix* is 5 or fewer characters, the resulting filenames are compatible with DOS naming conventions.

-g *drawing*

Specifies that screen displays should use line drawing graphics. Currently, *drawing* must be either **l** (for limited) or **f** (for full) line drawings.

-h

Displays a list of UX/RPM switches and exits UX/RPM.

-l *level*

Specifies logging is to be done. Currently, *level* must be **0** (no logging), **1** (exclude disk and process data), **2** (exclude process data), or **3** (all information).

-n *cycles*

Terminates UX/RPM after the specified number of cycles have been performed. *cycles* must be a positive integer.

-s *screen*

Specifies the screen UX/RPM that should display when first executed. *screen* must be a valid screen command key.

-t

Resolves TTY names for Process screens.

-L

Writes the contents of all UX/RPM information screens to a disk file (rpm.lst) when UX/RPM terminates.

Sample Command Lines

Minimum to start UX/RPM: **) rpm**
Reset cycle time to (for instance) 30 seconds and display the Disk Screen - Accumulated on entry: **) rpm -c 30 -s D**
Log all available data: **) rpm -l 3**
Log all available data to a file named "for_sepac": **) rpm -l 3 -f for_sepac**
Log system data only, every 30 seconds, until 240 snapshots have accumulated: **) rpm -l 1 -c 30 -n 240**
Log all available data and execute in background mode until killed: **) rpm -l 3 &**
Express convert the logfile "rpm.12.15" to DIF format: **) rpm -E -f rpm.12.15**
Convert the logfile "rpm.12.15", using the range of snapshots indicated: **) rpm -b 12:30:10 -e "12:25:10 12/05/93" -f rpm.12.15**
Convert the logfile "newdata" and write output to "log1": **) rpm -E -f newdata -F log1**
Start UX/RPM in playback mode and read "log1" for data: **) rpm -f log1**
Write all screens to disk upon termination: **) rpm -L**

UX/RPM Command Keys

Keystroke	Function
!	Forks a DG/UX shell from within UX/RPM
/	Searches for a text string
?	Displays on-line Help
1G	Goes to the top of the current list display
j	Moves the current display down one line
a	1. Displays only active processes; for Process screens only.
A	Displays ALL processes; for Process screens only.
b	Resets accumulated/history values
CTRL-b	Moves up one screen; for scrollable screens only.
c	1. Changes UX/RPM's cycle time 2. If you are changing the window display while viewing Process screens, this key displays user, system, and CPU time data 3. If you are changing the sort order for Process Screens, this key specifies the sort order to be by CPU Time
d	Displays Disk Screen - Incremental
D	Displays Disk Screen - Accumulated
f	Pauses and unpauses data collection
CTRL-f	Moves down one screen; for scrollable screens only.
G	Goes to the end of the current list display
h	Displays on-line Help

UX/RPM Command Keys

Keystroke	Function
i	Displays I/O Screen
I	Displays I/O History Screen
k	Moves the current display up one line
l	Writes the current display to a disk file
L	Writes all screens to a disk file
m	1. Displays the Screen Menu 2. If you are changing the window display while viewing Process screens, this key displays memory and wait data
n	1. Displays Network Screen 2. If you are changing the sort order for Process screens, this key specifies the sort order to be by Nice Value
N	Displays Network History Screen
o	Displays System Overview Screen
p	1. Displays Process Screen - Incremental 2. If you are changing the sort order for Process screens, this key specifies the sort order to be by Process ID
P	Displays Process Screen - Accumulated
Q, q	Exits UX/RPM
r	1. Reorders the sort order for process displays; valid only from Process Screens 2. If you are changing the sort order for Process Screens, this key specifies the sort order to be by Priority

UX/RPM Command Keys

Keystroke	Function
s	1. Displays Scheduling Screen 2. If you are changing the window display while viewing Process screens, this key returns to the default Process Screen display 3. If you are changing the sort order for Process screens, this key specifies the sort order to be by Resident Process Size
S	Displays Scheduling History Screen
t	Specifies a specific snapshot to examine; valid only from the System Overview Screen
v	Displays Virtual Memory Screen
V	Displays Virtual Memory History Screen
w	Allows you to change the window displays (that is, column data) for Process and Virtual Memory screens
x	Displays Virtual Disk Screen - Incremental
X	Displays Virtual Disk Screen - Accumulated

093-000874-02

DG/UX™
Real–Time
Performance
Monitor
(UX/RPM)
Reference Manual

093–000874–02

Cut here and insert in binder spine pocket