XVM/RSX PART II
INTRODUCTION TO RSX

CHAPTER 1

INTRODUCTION TO RSX

1.1 THE XVM/RSX ENVIRONMENT

XVM/RSX is a large-scale general-purpose operating system designed to handle real-time and batch processing as well as interactive program development in a multiprogramming environment. The system provides facilities for efficient programming in the following application areas:

- High-speed real-time command and control
- Laboratory data acquisition and analysis
- Real-time process monitoring and control
- Multiscope interactive graphics for computer-aided design
- Multiuser program development and execution
- General-purpose batch-processing

XVM/RSX operates on Digital's XVM computer. A wide variety of devices can be added to the basic system, including terminals, tape and disk peripherals, special-purpose instruments for scientific and industrial applications, and small peripheral computers that use the XVM as the host machine.

The most significant capabilities of XVM/RSX are the following:

- Support of the MULTIACCESS facility, which allows multiple users to perform program development concurrently with batch and other real-time operations.
- Support of new memory management facilities that allow task size to expand to 114K and provide a low-overhead mechanism for data sharing and intertask communication.
- Greatly simplified and automated system generation procedures as compared to previous versions of RSX.
- A mechanism for obtaining the CPU time used by tasks.
- On-line Systems Reconfiguration (OSC); a facility for dynamic reconfiguration of partitions, COMMON blocks and other system resources.
- Support of a multiscope interactive graphics system to

facilitate such applications as architectural design, electronic circuit design and structural analysis.

- A batch environment that provides accounting control and a wide range of batch functions for task development, source file editing and operations control.

- Time-slicing; a facility that allows the RSX system to service non-time-critical tasks in round-robin "time-shared" fashion while providing preemptive scheduling more relevant to the real-time applications.

- UNICHANNEL XVM, a peripheral processor that joins XVM and PDP-11 capabilities to form a powerful multiprocessor system.

These capabilities amplify the basic RSX facilities for installing, requesting and scheduling user and system tasks in partitioned areas of core. RSX automatically queues I/O for these tasks and enters and maintains up-to-date information on all task partitions, COMMON blocks, devices and I/O requests in a series of system lists.

1.1.1 Differences Between XVM/RSX and RSX PLUS III

XVM/RSX builds on the features of its predecessor, the RSX PLUS III operating system. In addition to the features of RSX PLUS III, XVM/RSX adds the following:

- Support of XVM and the XM15 hardware option for the PDP-15. This hardware support allows:

    - Maximum task size to reach 114K

    - Low-overhead data sharing and intertask communication

    - Task processing times to be recorded

    - A special class of user-mode (protected) tasks to issue input/output instructions

- Support of MULTIACCESS, which expands the system program development facilities. In previous versions of RSX, it was possible to use only one terminal for program development. Alternately, batch operation could be invoked. MULTIACCESS allows all terminals to be used simultaneously for program development while batch is run as a background job.

- Automated procedures for building XVM/RSX into an existing XVM/DOS operating system.

- Greatly improved access capabilities to disk file directories.

- Support of an Octal Debugging Technique (ODT).

- Partitions and system common blocks may exceed 32K in size.

- System common blocks may reside above 32K for data sharing.

As a result of XVM/RSX support of these new features, the structure of various system tables and lists has also changed.

1.2 XVM/RSX HARDWARE

XVM/RSX can function in either a minimum or an expanded hardware environment. A basic XVM with at least 32K of core memory can use one of three disks as the system device. A high-speed paper tape reader/punch and either DECtape or magnetic tape should be available for file backup. Additionally, at least one (and preferably several) DECwriters or other terminals must be included in the system configuration. Line printers, card readers and punches, plotters and writing tablets, and industrial data acquisition and control devices of various kinds can be added to the basic system. The UNICHANNEL XVM interface can also make a wide range of PDP-11 peripheral devices available to XVM software. The specific hardware environment is described in detail in Chapter 2.

1.3 XVM/RSX SOFTWARE

XVM/RSX provides facilities for preparing, compiling, assembling, building, installing, scheduling and supervising the execution of system and user tasks in real-time, interactive and batch modes. The most significant software modules available to users of XVM/RSX are listed below:

- An Executive that supervises core and disk management, task scheduling, I/O queuing and dynamic system priority control.
- A Resident Monitor Console Routine (MCR) Dispatcher and a collection of MCR functions that allow the user to schedule tasks, request information about system lists, resources and allocations, and reconfigure the system dynamically.
- A MULTIACCESS Monitor for controlling task development facilities (TDV) and a set of routines with which the user can edit source text, compile FORTRAN programs, assemble MACRO programs, build tasks for execution, run and debug user-written tasks, and perform a variety of utility functions.
- An I/O device handler task for each device supported by XVM/RSX.
- A batch processor that allows batch invocation of all primary system functions; permits jobs to be submitted with an MCR function, a batch call or a system directive; and provides the batch operator with a repertoire of commands.

1.4 XVM/RSX CONCEPTS

This section outlines some of the most significant concepts that governed the development of XVM/RSX and its expansion into the current XVM/RSX system.

1.4.1 Task Scheduling

XVM/RSX is a task-oriented monitor system. It controls and supervises a virtually unlimited number of system-resident and user-supplied programs called "tasks". After these tasks have been prepared, built and installed in the system, they share core and disk storage, I/O device handlers and other system resources. When the user builds and installs a task, he specifies the core partition in which the task resides and the software priority at which he normally wants the task to run. Later, by issuing simple commands from the console terminal or from another task, the user can schedule the execution of the task at one particular time, after the occurrence of a particular event or at periodic intervals.

XVM/RSX controls the execution of both system- and user-supplied tasks by scheduling these tasks according to run priority. Actual time of execution depends on task priority and partition availability.

Control of task scheduling is determined by the relative position of tasks in a system list or "deque". Creation and maintenance of these deques is described in detail in the part of this manual entitled SYSTEM LISTS AND ORGANIZATION. When a task is installed in the system, a node identifying this task is placed automatically in the System Task List (STL). When a task is made active by means of the REQUEST directive, a node is inserted in the Active Task List (ATL).

Whenever a "significant event" (Task initiation (REQUEST), Task completion (EXIT), I/O request queuing, I/O completion, MARK-time expiration, etc.) occurs, the executive attempts to give control to the highest priority task capable of executing at this time. If a task able to be run is found, it is given control at API level 7 (called "task level"). If no task can be run, the executive idles at mainstream level (no API levels active). The API level 7 light is an indicator of CPU usage by tasks.

The priority of a task must be between 1 and 512. The larger the number, the lower the priority. Tasks are entered in the system either at the priority assigned at task-building time (i.e., the time at which the task's-partition was specified and when its object code is linked with any system library routines) or at the priority specified when the task is installed (i.e., when an image of the task in executable form is written onto the disk). For instance, the user can install a task named SCAN with default priority 100 by typing after the prompting symbol (>):

```
MCR>INSTALL SCAN
```

If a new priority of 78 is desired, the user can INSTALL SCAN by typing the following:

```
MCR>INSTALL SCAN 78
```

Task priorities can also be altered at run time either by the operator or by a currently executing task. An installed task is activated by typing any of the following:

MCR>REQUEST SCAN	Request execution of SCAN at its default priority.
MCR>SCHEDULE SCAN 13:30:00 30M	Schedule SCAN to execute at its default priority at 1:30 p.m. and at every 30 minutes thereafter.
MCR>RUN SCAN 25M	Schedule SCAN to execute at its default priority 25 minutes from now.
MCR>SYNC SCAN H 30M 2H 78	Schedule SCAN to execute at priority 78 30 minutes past the next hour and every 2 hours thereafter.

Task execution is honored if the appropriate partition is available and if a task with a higher priority is not currently executing. Once a task begins execution, it runs to completion unless interrupted by a task with a higher priority or by tasks running in a time-slicing environment. An interrupted lower priority task will be resumed only when higher priority tasks have completed execution or have relinquished control. Whenever one task is interrupted by another, its active registers are automatically saved by the Executive and restored when execution is resumed.

Control can be passed to a task with a lower priority if a currently executing task voluntarity gives up control. For example, a high priority task might yield control to a lower priority task by waiting for an input/output operation to complete.

It is possible to schedule tasks to be run at some future time and to reschedule them for execution at periodic intervals. Any command which refers to future execution causes a pointer to the scheduled task to be inserted in the clock queue.

The following example illustrates task scheduling by RSX. Assume that two tasks are installed in the system. The tasks are named TASK1 and TASK2 and have priorities of 50 and 100 respectively (50 is the higher priority). The operator requests that TASK1 be executed one minute from now and that TASK2 be executed immediately. The following commands are typed:

```
MCR>RUN TASK1 1M

MCR>REQUEST TASK2
```

Figure 1-1 describes the sequences of events during the REQUEST and activation of both Tasks.

1.4.2 Partitioned Core

The virtually concurrent processing of different tasks by RSX is facilitated by the organization of both system-resident and user-supplied programs in partitioned areas of core. Although DEC provides both system modules and recommended partitions in which to install these modules, an individual installation can rebuild or relocate areas of core when necessary for a particular application. The installation is also responsible for assigning areas of core in which users can build and execute their own tasks.

1.4.3 Dynamic Reconfiguration

The partitions and other system spaces allocated during the initial RSX system configuration can be altered at any time with an on-line configurator supplied with XVM/RSX. COMMON blocks can also be added.

1.4.4 Real-Time Multiprogramming

The design of RSX makes it particularly suitable to the scheduling and execution of real-time operations. The system facilitates the interaction of many effectively simultaneous functions by allowing individual tasks to share core and to execute on a priority basis.

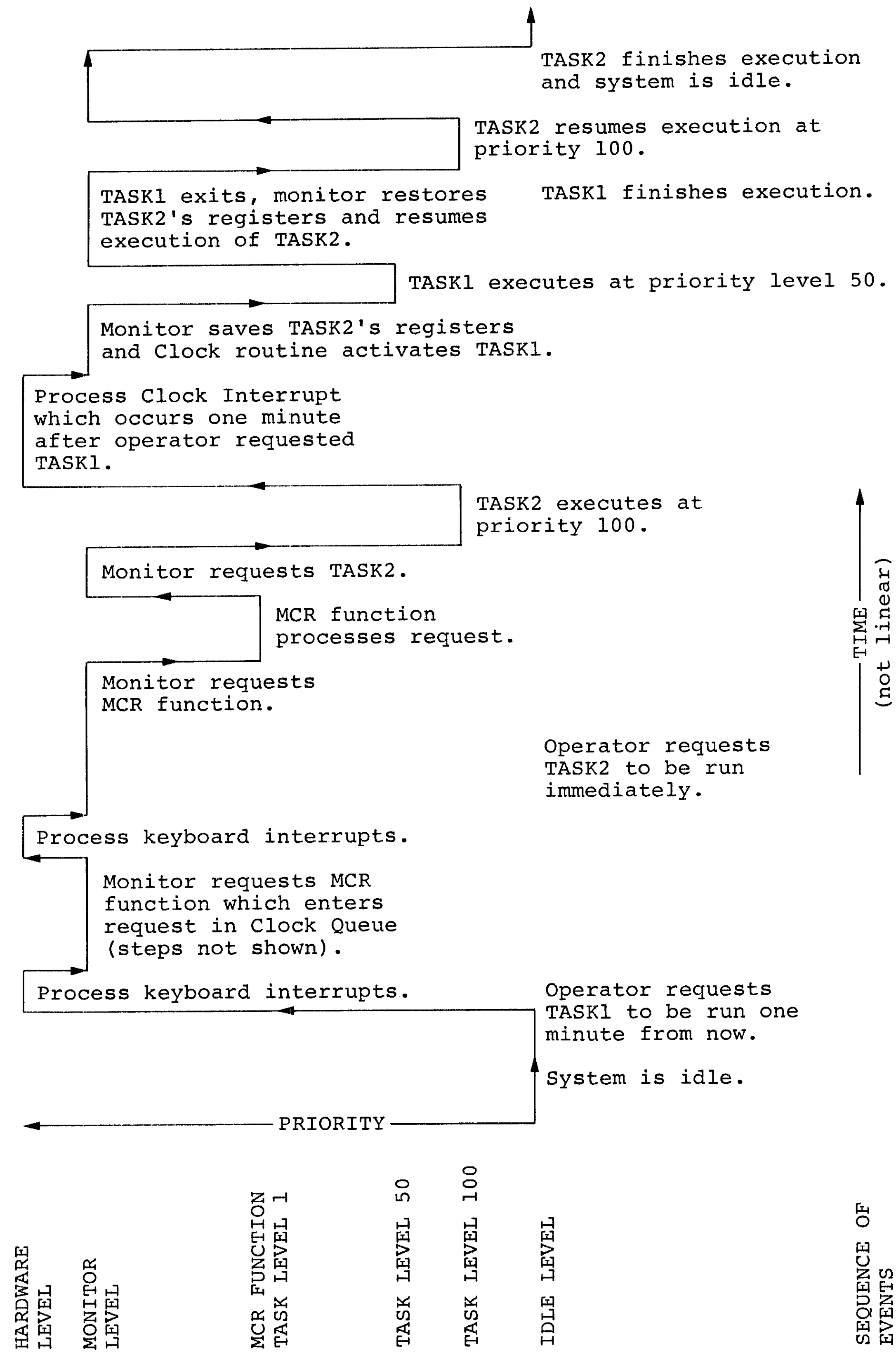

Figure 1-1
Task Scheduling

The user can specify polling or sampling sequences simply by setting appropriate priorities and execution cycles for different tasks. This capacity simplifies the creation and maintenance of store-and-forward message-processing systems, multiple instrument monitoring and control, laboratory data acquisition and analysis, and process- and command-control applications. The user may also specify that time-slicing is to be used to set a band of priorities in which tasks will execute at the same effective priority. With this facility, the system can not only provide immediate service for important processes but can also service many interactive users.

1.4.5 Memory Protection and Relocation

XVM/RSX supports memory protection and relocation. This means that a task can be built in such a way that it is prohibited access or execution outside its partition, that it cannot execute certain privileged instructions, and that the task can be relocated in another area of core without requiring rebuilding.

CHAPTER 2

XVM/RSX HARDWARE CAPABILITIES

XVM/RSX has been designed as a modular system, capable of functioning in a minimal hardware and software environment but also able to expand to accommodate a large number of I/O devices and a whole range of system programs for both real-time and batch operations.

2.1 MINIMUM HARDWARE CONFIGURATION

The following list summarizes the minimum set of hardware modules required for XVM/RSX operation on a DIGITAL XVM.

- 32K of core memory(1) (for some uses, 24K will suffice)
- Automatic Priority Interrupt (API)(1)
- Extended Arithmetic Element (EAE)(1)
- Real-time clock (period is 16.7 milliseconds for 60-Hz systems and 20 milliseconds for 50-Hz systems)(1)
- One RF15 DECdisk controller and two RS09 DECdisk platters (262,000-word fixed head)

  or:

  One RP15 controller and one RP02 disk pack

  or

  One UC15 peripheral processor (see below) and one RK05 disk cartridge
- One DECtape controller and one drive or
- One Magtape controller and one drive
- One terminal (DECwriter or other variety)(1)
- One PC15 high-speed paper tape reader/punch(1)
- Memory protection/relocation hardware(1)

---

(1) Included in all XVM's and PDP-15's with the XM-15 memory processor.

XVM/RSX supports three kinds of disks. Any of the three can be used as the system disk. All three kinds can be available at the same installation. Either DECtape or magtape must be available for XVM/RSX use. While only one terminal is specified as being essential in the minimum configuration, two are recommended for users of more than the basic software components.

2.2 EXPANDED HARDWARE FACILITIES

XVM/RSX is capable of supporting the following hardware in addition to the minimum configuration specified above:

- MX15 Memory Multiplexer.
- Up to 128K of core memory.
- FP15 Floating-Point Processor.
- Addition of one to eight RS09 disk platters for a maximum of two million words of storage.
- Addition of one or more TU56 dual DECtape units. The controller is designed to accommodate four TU56 units (eight tape drives).
- Addition of one to eight TU10 IBM-compatible magtape transports.
- As many as 17 terminals. These can be KSR33 or KSR35 teleprinters, LA30 or LA36 DECwriters, VT50 or VT05 display terminals, or any similar terminals *.
- Addition of one LP15 line printer (80, or 132 columns) **.
- AFC15 Analog Input Scanner (number of channels is optional).
- AD15 Analog-to-Digital Converter.
- UDC15 Digital Input/Output Controller (number of modules is optional).
- CR15 or CR03B card reader controller and reader **.

---

* The VT50 display terminal is supported at 2400, 1200 or 600 baud or at some rate less than 600 baud. The VT50 copier, cursor control commands and hold screen mode are not supported. The LA36 DECwriter is supported at a baud rate of 110 or 300. User MACRO programs can be written to take advantage of the LA36 132 column paper width. Because MCR and TDV functions have limited input buffers, a maximum of 80 characters per line can be input to such programs.

** Although XVM/RSX supports a variety of line printers, card readers and plotters, only one variety of each of these peripherals can be supported on a single XVM system.

- CP15 card punch controller and punch (100-card-per-minute, available through Computer Special Systems)
- As many as four VT15 graphics systems providing the GT15 graphics processor, including a display, a light pen, and the VW01 writing tablet
- Support of the UC15 UNICHANNEL peripheral processor, a PDP-11 interface which facilitates use of the LP11, LS11, or LV11 (print only), the CR11 card reader, and up to eight RK05 disk cartridges, and XY11 or XY311 plotter.(2)

2.3 EXTENDED MEMORY MANAGEMENT FACILITIES

XVM/RSX supports Digital's XVM and the XM15 option on PDP-15's. Support of this hardware expands the operating system's memory management facilities. First, it permits protected tasks to exceed 32K in size. Executable code is limited to the first 32K of a task's partition but partition space above the 32K boundary can be used for common blocks. As a result, a protected task can access, via indirect addressing, up to 114K, given sufficient partition space. Second, this hardware support provides the user with a low-overhead mechanism for data sharing and intertask communication for protected tasks. When this feature is invoked, via a call to the Share system directive, up to 7936 words of the calling task's addressing space can be mapped by hardware into some defined partition or system common block. Subsequently, the task can read or write data or execute instructions in the shared block of memory as though it were a common block within the task. Third, this hardware contains a high resolution timer which can be used to obtain the processing time consumed by all user-called tasks. XVM/RSX furnishes the user with a mechanism for obtaining this data in a format which is easy to store and manipulate. Refer to the documentation on the Exit system directive for more information on task timing.

2.4 DISK SUPPORT

The following types of disks are available on an XVM system:

- RS09 fixed-head DECdisk
- RP02 disk packs
- RK05 disk cartridges (controlled under UNICHANNEL XVM)

Any of these disks can be used as the system disk; the others can be used to store user files. Since each disk controls a full file structure, RP02 and RK05 disks can be interchanged while the system is running. The following table summarizes sizes and numbers of supported disks:

Controller	Disk	Number Supported	Words in Each
RF15	RS09	8 disk platters	250,000
RP15	RP02	8 disk packs	10,000,000
UC15	RK05	8 disk cartridges	1,200,000

The figure below illustrates a possible configuration of disks in XVM systems.

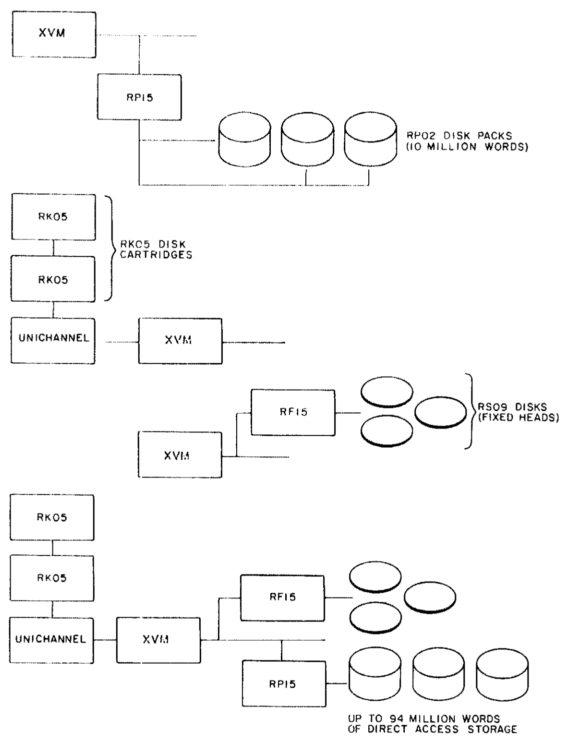

Figure 2-1
Disk Configuration

2.5 SUPPORT OF UNICHANNEL

The XVM/RSX support of UNICHANNEL is important to the expanded real-time system. UC15 is a peripheral processor which allows the XVM and PDP-11 to combine device and processing capabilities. Use of UC15 allows tasks running under RSX to access certain PDP-11 I/O devices. This expands the available XVM configuration by allowing the RK05 disk to be used as the system disk or for additional disk storage. Furthermore, support of the CR11 card reader, LV11, LP11, or LS11 printer, and the XY11 or XY311 plotter allows users to realize additional benefits of the UNICHANNEL beyond support of the RK05 disk. XVM/RSX uses the UNICHANNEL spooling capability to buffer card decks, listings, and plots between PDP-11 and XVM. Following is a graphic representation of UNICHANNEL use.

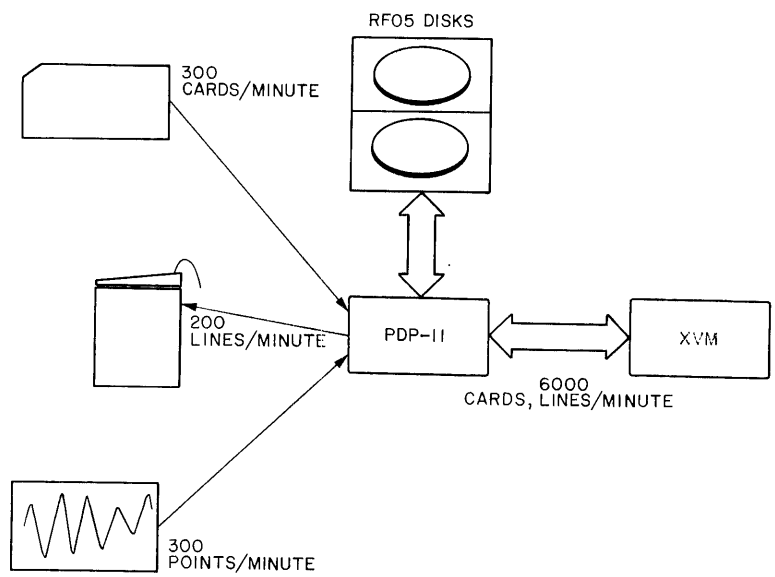

Figure 2-2
UNICHANNEL Configuration

CHAPTER 3

XVM/RSX SYSTEM SOFTWARE

3.1 INTRODUCTION TO SOFTWARE CAPABILITIES

XVM/RSX provides extensive capabilities for accessing software through real-time, interactive and batch approaches. In addition, XVM/RSX offers full facilities for preparing, compiling, assembling, task-building, installing and supervising the execution of batch and real-time tasks.

XVM/RSX makes available a full complement of file copy and manipulation routines, I/O functions, and special-purpose batch programs and interfaces. Figure 3-1 illustrates the preparation, installation and execution of a real-time task under RSX control. The rest of this chapter summarize the most important characteristics of the standard RSX system programs.

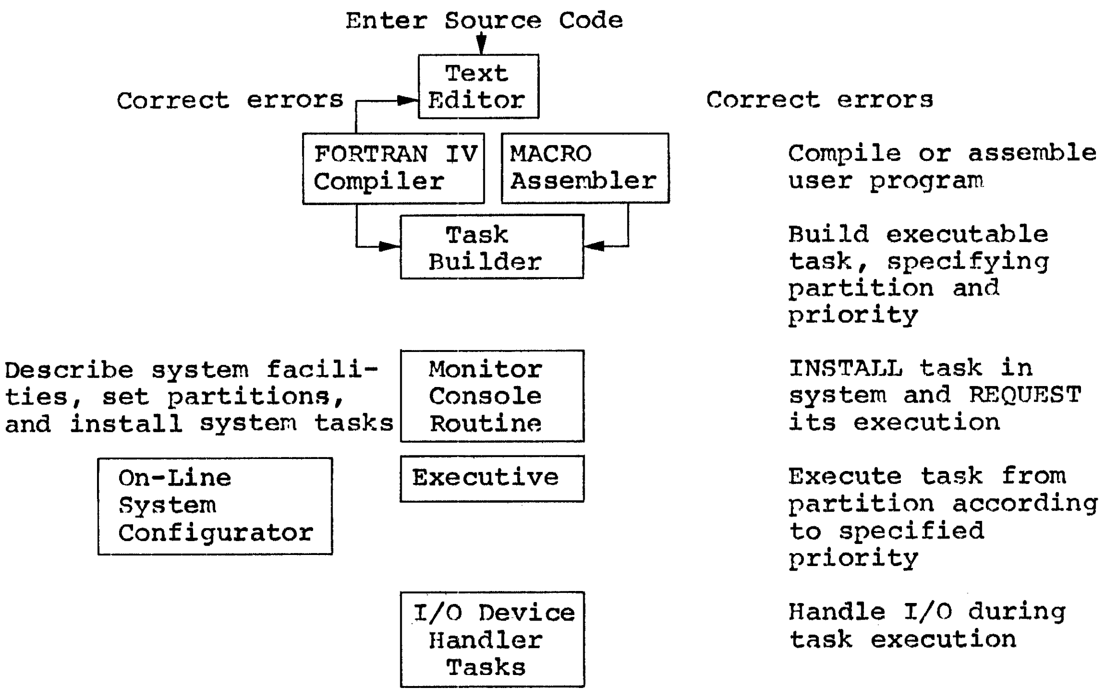

Figure 3-1
RSX On-Line Task Development

3.2 EXECUTIVE

The RSX executive controls system activities. Its primary functions include resource allocation, task scheduling on a priority basis, and queueing of I/O requests.

3.3 MONITOR CONSOLE ROUTINE

The Monitor Console Routine (MCR) consists of a Resident MCR Dispatcher and a series of functions which perform such operations as the following:

- Returning system status information
- Installing tasks
- Requesting, scheduling, or cancelling task execution
- Performing on-line configuration of partitions, COMMON blocks, and other system spaces

- Requesting batch job information

- Queuing batch jobs for execution

The user can call any of the MCR functions at the MCR terminal by supplying a unique MCR function name and any necessary parameters.

3.4 TASK DEVELOPMENT FACILITIES

On-line task development can be performed with the aid of a MULTIACCESS Monitor and a series of task development (TDV) functions such as the following:

- FORTRAN IV Compiler, which translates user source programs into binary object code in the RSX environment

- MACRO Assembler, which processes assembly language code and generates relocatable binary object files

- Text Editor, which is used for the on-line creation and modification of symbolic text

- SLIP, a line-oriented editor used primarily for file merging or for RSX batch operations

- Task Builder, an interactive program that links the user's relocatable binary files with library functions to build executable tasks

- BTK, the Basic Task Builder that assumes default task-building information and often operates in a batch environment

- Utility functions for copying and listing files

- Utility functions for initiating user task execution and queuing a task for batch operation

The user can call any of these TDV functions from any terminal by logging into the MULTIACCESS subsystem and typing the TDV function name along with necessary parameters.

3.5 BATCH PROCESSOR

The RSX batch environment offers extensive capabilities for job queuing, job accounting, and submitting user and operator commands. Some of the features of RSX batch include:

- Queuing jobs for batch operation from MCR, MULTIACESS or a user task

- Requesting a printout of the job account file and resetting control information in this file at the beginning of a new accounting period

- Operator ability to force execution of jobs, hold or cancel jobs in the queue, request the status of jobs in the system, and indicate operator availability

3.6 I/O DEVICE HANDLER TASKS

I/O device handlers are supplied to process queued task requests for standard devices supported by RSX. The user can write his own handlers for special-purpose devices not supported by RSX.

CHAPTER 4

GUIDE TO RSX DOCUMENTATION

Documentation for the XVM/RSX system is provided in the form of one manual containing 11 major parts. This chapter briefly summarizes the contents of each of the major parts of RSX documentation.

4.1 AN OVERVIEW OF THE XVM/RSX SYSTEM

The overview part serves as an introduction for users unfamiliar with RSX operations. It outlines some of the primary design concepts that governed the creation and support of RSX. It also summarizes the hardware and software capabilities of the system, emphasizing those features that are new to the latest system release.

4.2 SYSTEM MANAGEMENT

This part of the manual recommends a variety of viable core layouts and supplies some guidelines for allocating space for different system modules. The use of the checkout package supplied with the XVM/RSX system is explained, and a full listing of assembly parameters and task-building procedures is supplied.

4.3 MONITOR CONSOLE ROUTINE (MCR)

The MCR part provides full descriptions of all MCR functions implemented for XVM/RSX. This document also contains a detailed explanation of the syntax and analysis of MCR commands, the invocation and use of the Resident MCR Dispatcher, and the meaning of error messages that might be encountered during MCR interaction.

4.4 SYSTEM DIRECTIVES

This part of the manual describes all system directives implemented for XVM/RSX use. It provides full information on calling directives from MACRO and FORTRAN programs, and summarizes event variables that might be returned to indicate the success or failure of the requested operation.

4.5 INPUT/OUTPUT OPERATIONS

The I/O part summarizes the operations of all I/O device handler tasks available in the XVM/RSX system. A chapter is devoted to each handler, including general operations, implemented I/O function calls and returned event variables.

4.6 ON-LINE TASK DEVELOPMENT

This part of the manual describes the way in which tasks can be prepared, compiled or assembled, and task-built in an on-line environment under the MULTIACCESS Monitor. It explains the invocation of each major TDV function, describes the meaning of the error messages that might be encountered during TDV operation, and summarizes the invocation and use of the MULTIACCESS Monitor.

4.7 BATCH-PROCESSING

The batch-processing part describes in detail the batch capabilities available in XVM/RSX. It explains the way in which the batch environment is entered, the means of submitting batch jobs and the new set of operator batch-control commands.

4.8 SYSTEM LISTS AND ORGANIZATION

This part of the manual describes the internal organization of XVM/RSX information in a series of linked double-ended lists called deques (pronounced "decks"). It explains the basic concepts of the "pool" and the use of large and small nodes. It also details the structure and use of each of the system lists constructed and maintained by RSX system operations.

4.9 GRAPHICS SYSTEM

The documentation on XVM/RSX graphics describes the capabilities and features of the graphics software. It contains information on how to initialize an environment for running graphics and what changes have been made to this software since the previous release of RSX.

4.10 ADVANCED TASK DEVELOPMENT

This part of the manual summarizes approaches to writing advanced tasks, such as MCR or TDV functions, I/O device handlers or interrupt routines. It includes guidelines for constructing tasks and examples of code in each of the categories described.

4.11 XVM/RSX V1B INSTALLATION GUIDE

This part of the manual provides detailed instructions on performing initial system build procedures.

INDEX