EY-0072E-SG-0101

RSX-11M/M-PLUS Utilities and Commands

A Self-Paced Course

Volume I

digital

EY-0072E-SG-0101

RSX-11M/M-PLUS Utilities and Commands

A Self-Paced Course

Student Workbook
Volume I

Prepared by Educational Services
of
Digital Equipment Corporation

Printed in U.S.A.

10/82-15

CONTENTS

SG STUDENT GUIDE

1 RSX-11M/M-PLUS SYSTEM OVERVIEW

2 GETTING STARTED ON THE SYSTEM

3 CREATING AND MODIFYING FILES

4 FILE AND DIRECTORY MAINTENANCE

5 PROGRAM DEVELOPMENT

6 USING THE EDITOR EFFECTIVELY

7 USING INDIRECT COMMAND FILES

8 CONTROLLING TASK EXECUTION

9 LIBRARIES

10 ADVANCED MAINTENANCE OPERATIONS

AP APPENDIX

FIGURES

TABLES

EXAMPLES

SG

STUDENT GUIDE

INTRODUCTION

RSX-11M/M-PLUS Utilities and Commands instructs you in the facilities of the RSX-11M operating system, one of many operating systems run on a PDP-11. Each module discusses a primary feature of RSX-11M that should be mastered, as well as the objectives, resources, and laboratory exercises needed to learn the material.

COURSE DESCRIPTION

This self-paced version of RSX-11M/M-PLUS Utilities and Commands is the first operating system specific course in the RSX-11M/M-PLUS course string. It is the base for all higher level RSX courses. It instructs system managers, operators, programmers and other users how to interact with the operating system to perform basic daily work functions.

Major course topics covered:

- How to log on and off the system and issue commands that provide system information.
- Use interactive editors and file maintenance utilities to create and maintain file collections.
- Use of compilers/assemblers and other utilities to develop MACRO-11 and FORTRAN programs.
- Use of indirect command files to control interactive job execution.
- Use of file and volume maintenance utilities to prepare new volumes, and to back up user files.

PREREQUISITES

1. Knowledge of the concepts of minicomputer organizations as in the DIGITAL course, Introduction to Minicomputers, audio/visual or lecture/lab format.

2. Knowledge of PDP-11 organization and programming as in the DIGITAL course, PDP-11 Concepts for High-Level Language Programmers.

COURSE GOALS

1. Knowledge of RSX-11M/M-PLUS terminology and operating philosophy.

2. Ability to find information in the system documentation.

3. Use a terminal to communicate with the system.

4. Create and maintain files.

5. Assemble, compile and task-build supplied programs.

6. Maintain a file directory.

7. Control the execution of supplied programs.

8. Create, maintain and use libraries.

9. Perform tasks needed to use private disk and tape volumes.

NONGOALS

1. This course does not teach the MCR command language. Some MCR commands are referenced in a particular table. If you wish to learn MCR, do so in the lab.

2. You will not learn the syntax rules of FORTRAN and MACRO, or how to write a FORTRAN or MACRO program. It is assumed that you are familiar with the syntax structure of either FORTRAN or MACRO-11.

COURSE RESOURCES

The following are available resources:

1. This Student Workbook.
2. Operating system information files provided by the HELP command.
3. RSX-11M/M-PLUS Documentation Set manuals.
4. On-line example files and indirect command files supplied by your instructor.
5. Your fellow students -- sharing ideas and questions is rewarding.
6. Your instructor.

The following documentation is essential to the course:

1. Introduction to RSX-11M and RSX-11M-PLUS
2. RSX-11M/M-PLUS Command Language Manual
3. MCR Operations Manual
4. Utilities Manual
5. Task Builder Manual
6. EDT Editor Reference Card

COURSE ORGANIZATION

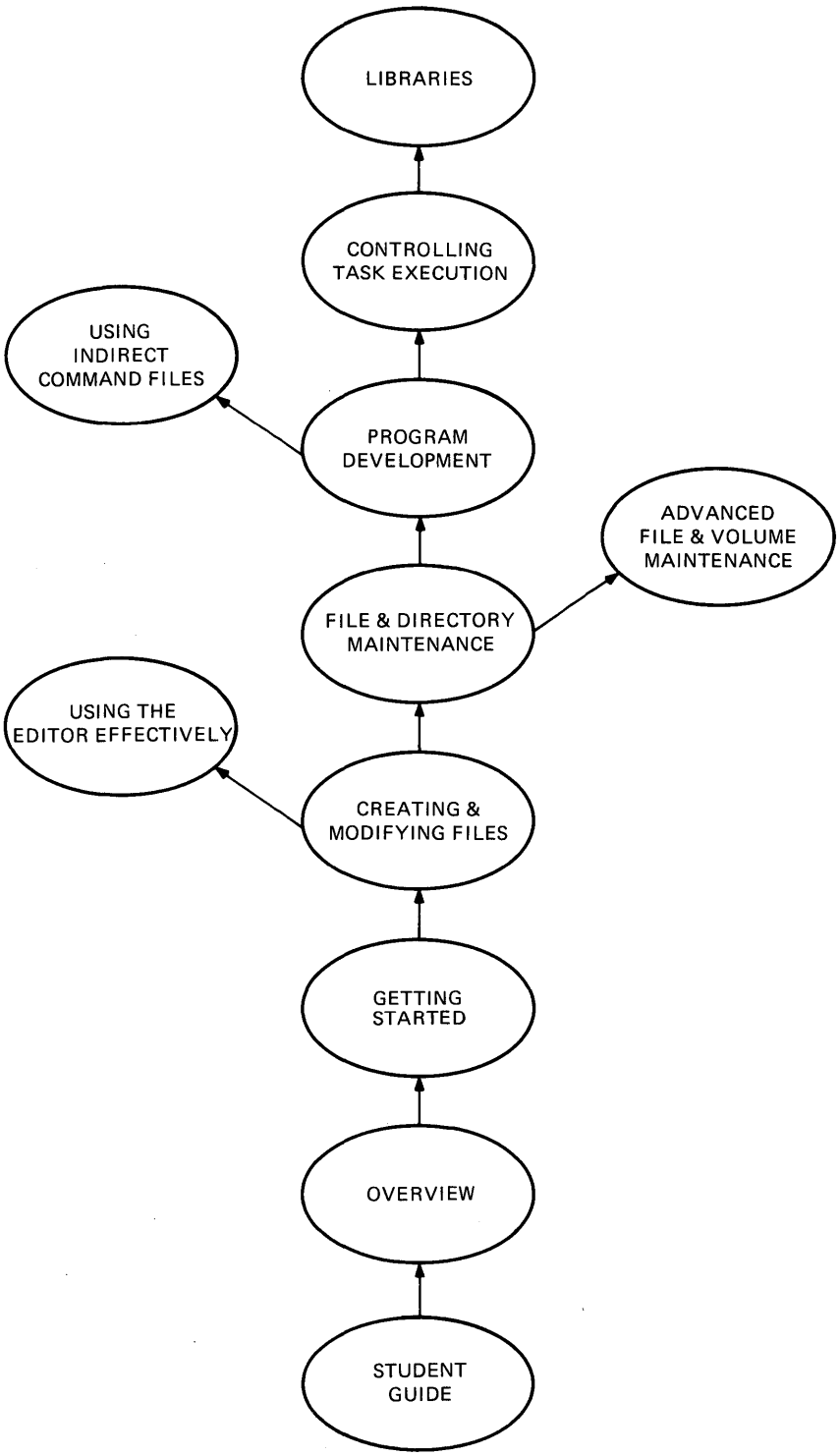
This is a self-paced course with laboratory sessions and written exercises. In place of a teacher you have a course administrator and a subject matter expert. In some cases, the same person may perform both functions. The course administrator manages the mechanics of this course. He makes sure you have easy access to the system and the on-line course materials. As you finish modules, he records your progress. The subject matter expert will answer any technical questions. Before you consult the expert, however, read the course materials and references in an effort to answer the question yourself.

The Student Workbook is divided into modules which consist of:

- An **introduction**, which describes the purpose(s) of the module and motivation for meeting its objectives.
- One or more **objectives**, which describe operations you should be able to perform when you have completed the module.
- A list of **resources** you may require to complete the module.
- The module **text**, which includes the following elements:
  - Descriptive text.
  - Illustrations, which clarify the relationships among various elements of a RSX-11M/M-PLUS system, or summarize steps of a particular process.
  - Tables, which summarize the operations covered by the module, and list the commands you enter to perform those operations.
  - Examples, containing sample listings from actual terminal sessions.
- Laboratory exercises, available in a separate book, which provide the practice needed to master the module objectives. Solutions to the exercises are also provided.

COURSE MAP

The course map shows how each module of the course is related to the other modules and to the course as a whole. Prerequisite modules are those whose arrows in the map point into a module and should be accomplished before proceeding to the next module.



HOW TO TAKE THIS COURSE

Because this is a self-paced course, you determine how much time to devote to each subject. You can quickly cover familiar topics. You can spend more time on topics which are of more interest or use to you in your job, and less time on topics of little use.

Whenever you are ready for a new module, read the introduction and the objectives. If you feel you already understand the material in the module, you can go immediately to the Tests and Exercises for that module. If you do not understand much of the material, read the module. The text explains new concepts and refers you to related readings in the supplementary manuals. The listings provide typical examples of the operations that a user performs.

Some of the readings in the manuals are essential and others are optional. If you feel that you need no further clarification on a topic, skip the reading. However, if it is a topic you wish to pursue, an appropriate reading reference is provided.

As you study the module, keep the module objectives in mind. If a skill is listed as an objective, be sure to master it. Later modules may depend on this skill.

While you are reading through the module, try the commands being discussed on your own terminal. This interactive way of learning will help your understanding of the command. Practice the commands repeatedly until you fully understand their function.

The files needed for this course are located in UFD [2Ø1,1] on your system disk. If they are not there, check with your course administrator to find out where they are located. Do not modify the files in UFD [2Ø1,1]. Instead, copy the files to your own UFD and use them there. In this way, the original files remain intact for the next student.

Once you have read the modules and done the learning activities, you can do the lab exercises. The solutions are included in the Test and Exercises book.

Once you have mastered the objectives of a module, the course administrator will record your progress on the progress plotter below. Then, you are ready to begin a new module. Otherwise, return to the module text for further study.

The time that you spend on this self-paced course is dependent upon your experience and interest in the topic as well as time spent experimenting with the system. Use Table 1 as a guide when you set your schedule.

PERSONAL PROGRESS PLOTTER

	Date Started	Date Completed	Time Spent	Sign-Off Initial
1. RSX-11M/M-PLUS System Overview				
2. Getting Started on the System				
3. Creating and Modifying Files				
4. File and Directory Maintenance				
5. Program Development				
6. Using the Editor Effectively				
7. Using Indirect Command Files				
8. Controlling Task Execution				
9. Libraries				
10. Advanced Maintenance Operations				

Table 1 Typical Course Schedule

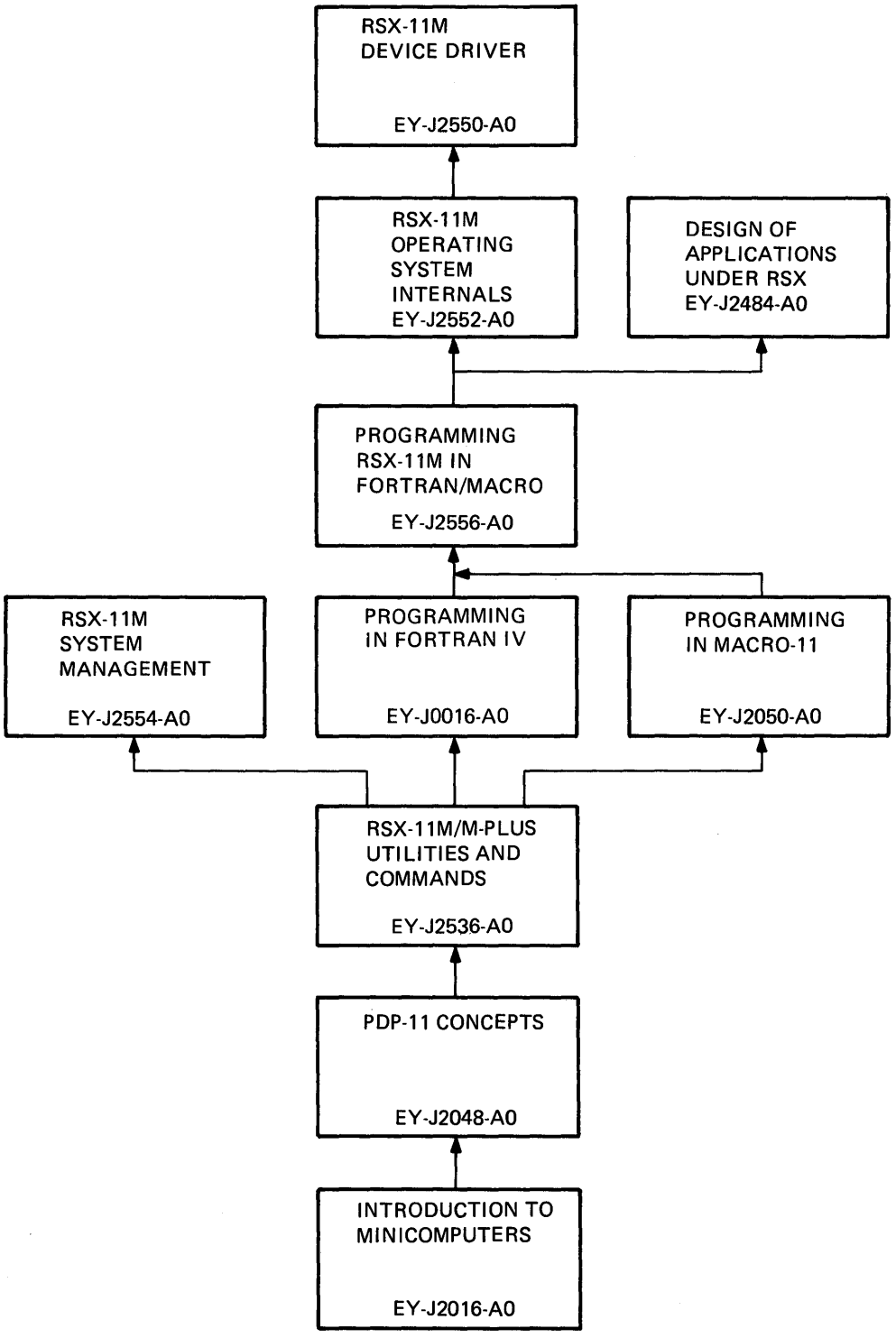
Module		More Experienced Student	Less Experienced Student
1	RSX-11M/M-PLUS System Overview	4.5	6.Ø
2	Getting Started on the System	5	6.Ø
3	Creating and Modifying Files	4	5.Ø
4	File and Directory Maintenance	4	5.5
5	Program Development	4	6.Ø
6	Using the Editor Effectively	3.5	4.5
7	Using Indirect Command Files	4	5.Ø
8	Controlling Task Execution	4	5.Ø
9	Libraries	3	4.5
1Ø	Advanced Maintenance Operations	4	5.5
Total Hours Study and Lab		4Ø.Ø	53.Ø

RSX-11M-PLUS CURRICULUM

The RSX-11M/M-PLUS curriculum is illustrated in the Course String below.

After completing this course, depending upon your final objective, you should take the System Management course, and either Programming in FORTRAN IV or Programming in MACRO-11. One or the other is a prerequisite for Programming RSX-11M in FORTRAN/MACRO, which is available in both lecture/lab and self-paced formats. To derive the most benefit from a higher level language course, the prerequisite courses must be taken.

COURSE STRING



TK-7819

1

RSX-11M/M-PLUS SYSTEM OVERVIEW

INTRODUCTION

Every computing system is designed for a purpose. This purpose determines how the system component parts are assembled to make a unit workable. The RSX-11M/M-PLUS Operating Systems were designed to solve real-time problems, those often requiring complete dedication to the computer's resources. This module explains:

1. How the operating system keeps track of your program
2. How it allocates the CPU and memory to your program when requested to run
3. How it decides who gets CPU time and memory space when programs are competing for these resources.

An understanding of the processing philosophy of this operating system will help you compare the differences between this and other operating systems used with PDP-11 hardware.

OBJECTIVES

1. List the characteristics of the RSX-11M/M-PLUS operating system.
2. List the facilities available with RSX-11M/M-PLUS and the documentation provided.
3. Describe the resource management techniques used by RSX-11M/M-PLUS for CPU and memory.
4. List the various states of an RSX-11M/M-PLUS task and how each affect task execution.
5. State the difference between a mapped and unmapped operating system.
6. Define the terms virtual and physical address and state how they apply to a task and physical memory.

RESOURCES

In order to complete this unit, you must have access to the following document:

Introduction to RSX-11M and RSX-11M-PLUS

OVERVIEW

A computer system consists of two essential components, the hardware and the software.

The Hardware

The hardware can consist of any combination of a CPU, memory and peripheral devices depending upon your site requirements. One important feature of the PDP-11 hardware architecture is the ability to easily add peripheral devices to a configuration, but only the devices necessary to a given application. Like a fingerprint, each PDP-11 hardware configuration can be unique. In addition to a Central Processing Unit (CPU), memory, and a communications link between all the hardware devices, the system may have a mix of the following devices:

- Disk drives
- Floppy disk drives
- Magnetic tape drives
- Hardcopy terminals
- CRTs
- Line printers
- Paper tape readers and punches
- A/D converters
- Communication Interfaces

The Software

The hardware alone is not enough to make a computer a useful tool. Without software to allow the printer to print, a terminal to send and receive data, and a magnetic tape drive to capture information, the hardware would remain just hardware. It is the software coupled with the hardware that makes the computer a useful tool. In general, the software is the instructions (programs) executed by the computer system to perform a perticular function.

THE OPERATING SYSTEM

Having the hardware and software alone does not guarantee rapid, easy development and execution of programs. An additional element is needed to tie these two together. Coordinating the many activities and features of the hardware and software, and managing these resources at computer speeds is the task of the operating system.

An operating system is a collection of programs that organizes a set of hardware devices into a working unit which people can use. It generally consists of two parts:

- An Executive
- A set of System Utilities

The Executive acts as the primary interface between the hardware and a program running on the system, and between the program and the people who run the program. The basic functions of the Executive are:

- Scheduling of the CPU
- Allocation of physical memory
- User/hardware interface
- Device/data management
- Program processing facilities

There are many operating systems that will execute on the PDP-11 hardware. Table 1-1 lists those that are currently supplied by Digital Equipment Corporation. Each was developed for a particular processing environment.

LEARNING ACTIVITY

For further information regarding each operating system, READ the respective chapters in the PDP-11 Software Handbook.

Table 1-1 PDP-11 Operating Systems

Name	Description
RT-11 and CTS-300	Real-Time Operating System for PDP-11 Processors. A small, single-user foreground/background system that can support a real-time application job's execution in the foreground, and an interactive or batch program development job in the background.
DSM-11	DIGITAL Standard MUMPS Operating System for PDP-11 Processors. A small- to large-size timesharing system that offers a unique fast access data storage and retrieval system for large data base processing; originally designed for medical record management and now available for similar data base applications.
RSTS/E and CTS-500	Resource-Sharing Timesharing System/Extended Operating System for PDP-11 Processors. A moderate- to large-size time-sharing system that can support up to 63 concurrent jobs, including interactive terminal user jobs, detached jobs, and batch processing.
RSX-11M	Real-Time System Executive Operating System for PDP-11 Processors. A small-to moderate-sized real-time multiprogramming system that can be generated for a wide range of application environments -- from small, dedicated systems to large, multipurpose real-time application and program development systems.
RSX-11M-PLUS	Real-Time System Executive Operating System-PLUS for high-end PDP-11 Processors. A large real-time system that takes advantage of the enhanced hardware features and larger memory available on the PDP-11/23-PLUS/24/44/70 processors. (RSX-11M-PLUS is a superset of RSX-11M.)

Table 1-1 PDP-11 Operating Systems (Cont)

Name	Description
RSX-11S	Real-Time Multiprogramming Executive Operating System for PDP-11 Processors. A small, execute-only member of the RSX-11 family for dedicated real-time multiprogramming applications (requires a host RSX-11M, RSX-11M-PLUS, IAS or VAX/VMS system).
IAS	Interactive Application System for PDP-11 Processors. A large multiuser time-sharing system, allowing real-time application execution concurrent with time-shared interactive and batch processing.

THE RSX-11M-PLUS OPERATING SYSTEM

Every operating system's architecture is built around some processing philosophy. The RSX-11M/M-PLUS operating systems are two of many operating systems that will execute on the PDP-11 hardware. Your organization purchased the RSX-11M/M-PLUS operating system because its processing philosophy fits the application for which it was purchased. The following list shows the characteristics of the RSX-11M/M-PLUS operating system.

RSX-11M/M-PLUS Characteristics

- Real-Time - The operating system is designed primarily to handle the execution of real-time tasks as opposed to tasks that lend themselves to a time-sharing system. Real-time tasks are critical tasks that need exclusive use of the system resources on short notice and until they are no longer needed. In this type of a system, a task gains control of the CPU by priority and keeps CPU control until it completes, aborts, or some significant event occurs. Also, when the system is not servicing real-time tasks, the CPU's idle time can be divided into time slices and shared among tasks that lend themselves to a time-sharing environment, such as editing programs.

- Multiprogramming - RSX-11M/M-PLUS allows more than one task to be resident in memory and competing for CPU time. The CPU is the most important resource in a computer system. Having many tasks resident in memory at the same time minimizes CPU idle time.

- Multiuser - RSX-11M/M-PLUS allows more than one person to be able to use the system at one time. Each individual can be sitting at a terminal performing work, and it will appear that the system is dedicated to him alone.

- Interactive - Being able to talk directly to the operating system allows the user to interact with each step in a process. It can be verified that the step was executed correctly before proceeding on to the next step. This flexibility is not available in a batch system, where the complete job stream is executed whether or not there was an error in the first step.

- Disk-Based - The tasks that make up the RSX-11M/M-PLUS operating systems are resident on a disk pack, and read into memory from this disk when needed. The disk is also used to temporarily store tasks when freeing up memory space for higher-priority tasks. It also provides a file structure for maintaining user programs and other files.

- Intertask Communication - A task executing under the RSX-11M/M-PLUS operating system has the ability to pass data on to another task, can cause another task to begin execution, and can notify another task that a certain event has occurred.

RSX-11M/M-PLUS Components

Although we generally think of an operating system as a single entity, it is really comprised of smaller pieces that are closely associated with each other. The two major groupings of these pieces are the Executive and the supporting utilities.

The Executive is the root of the operating system and is always resident in memory. It is made up of the following parts:

- Executive
  - CPU Scheduler
  - Memory Allocator
  - System Services (Directives)
  - System Software:
  - Command Line Interpreters

        DCL
        MCR
        User-supplied

  - FILES-11 Data and File Manager
  - Device Drivers

The second group, supporting utilities, only becomes memory-resident when a request for its services is made.

- Supporting Utilities comprise:
  - Program Development Tools

        TEXT Editors (EDI, EDT)
        MACRO-11 Assembler
        Task Builder
        Librarian

            System Macro Source, Object, and Universal Libraries

- High-Level Languages (Optional)

  FORTRAN IV
  FORTRAN IV-PLUS
  FORTRAN-77
  COBOL
  BASIC-11
  BASIC-PLUS-2

- File Maintenance Tools

  Backup and Restore Utility (BRU)
  Peripheral Interchange Program (PIP)
  File Transfer Program (FLX)
  Print and Queue Utility
  Queue Manager (QMG)

- Volume Maintenance Tools

  Disk Volume Formatter (FMT)
  Bad Block Locator Utility (BAD)
  File Structure Verification Utility (VFY)
  File Dump Utility (DMP)
  Disk Save and Compress (DSC)

- Program Maintenance Tools

  File Compare Utility (CMP)
  Source Language Input Program (SLP)
  Object Module Patch Utility (PAT)
  Task/File Patch Program (ZAP)

Documentation

A complete set of documents is supplied with each operating system that DIGITAL ships. The documents contain information on all the facilities available to the user. There are seven volumes in the set, each containing one or more manuals. Their organization is shown in Figure 1-1. Sheet three of this figure shows the documentation available for languages that are available under separate license. Your computer center will only have the documentation for those languages which you have purchased. For your convenience, Figure 1-2 indicates the recommended sequence for reading the manuals. You should read the RSX-11M/M-PLUS Information Directory first. The release notes will provide information on changes in the operating system that are included in this version.

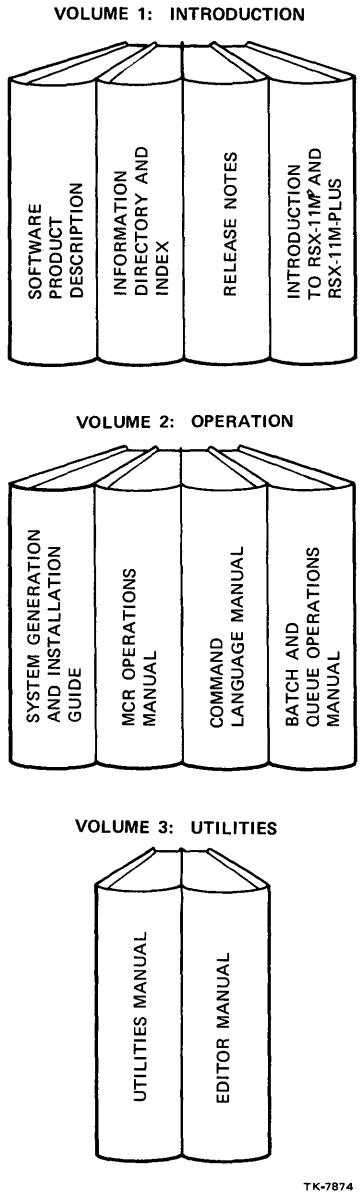


Figure 1-1 The Documentation (Sheet 1 of 3)

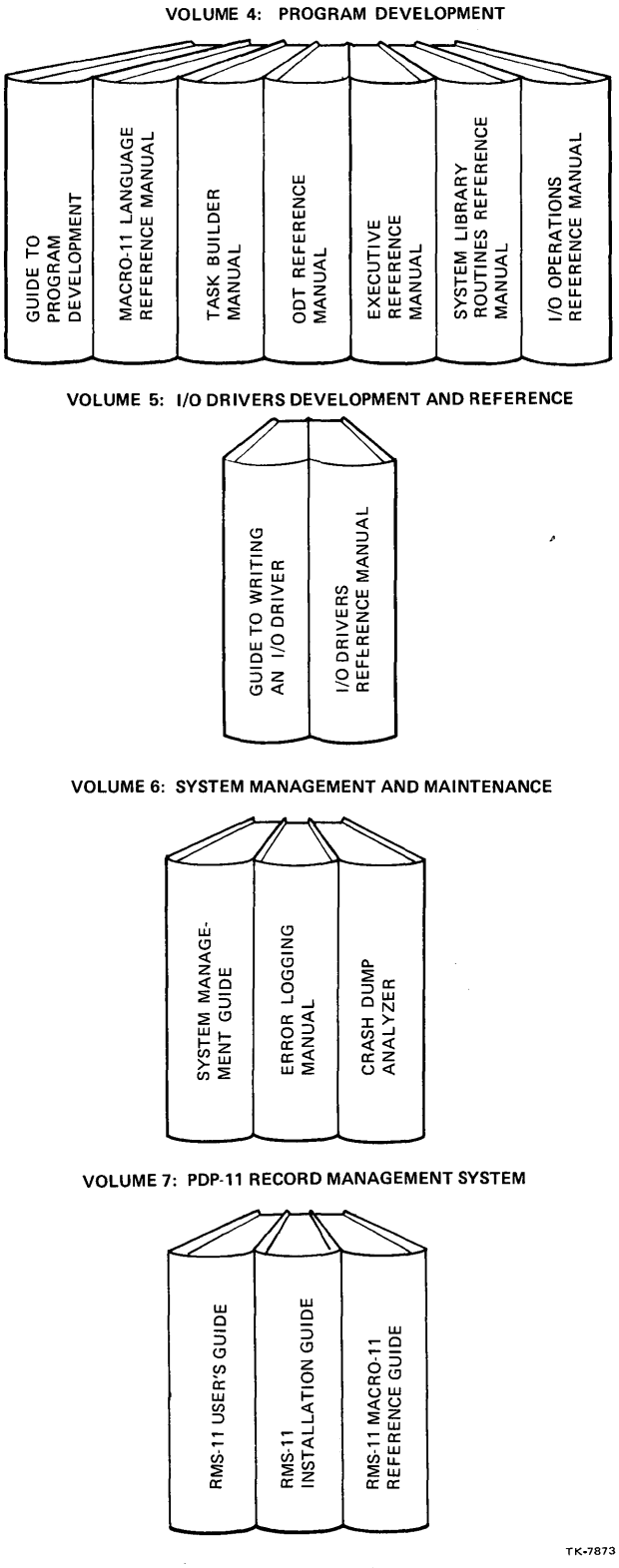


Figure 1-1 The Documentation (Sheet 2 of 3)

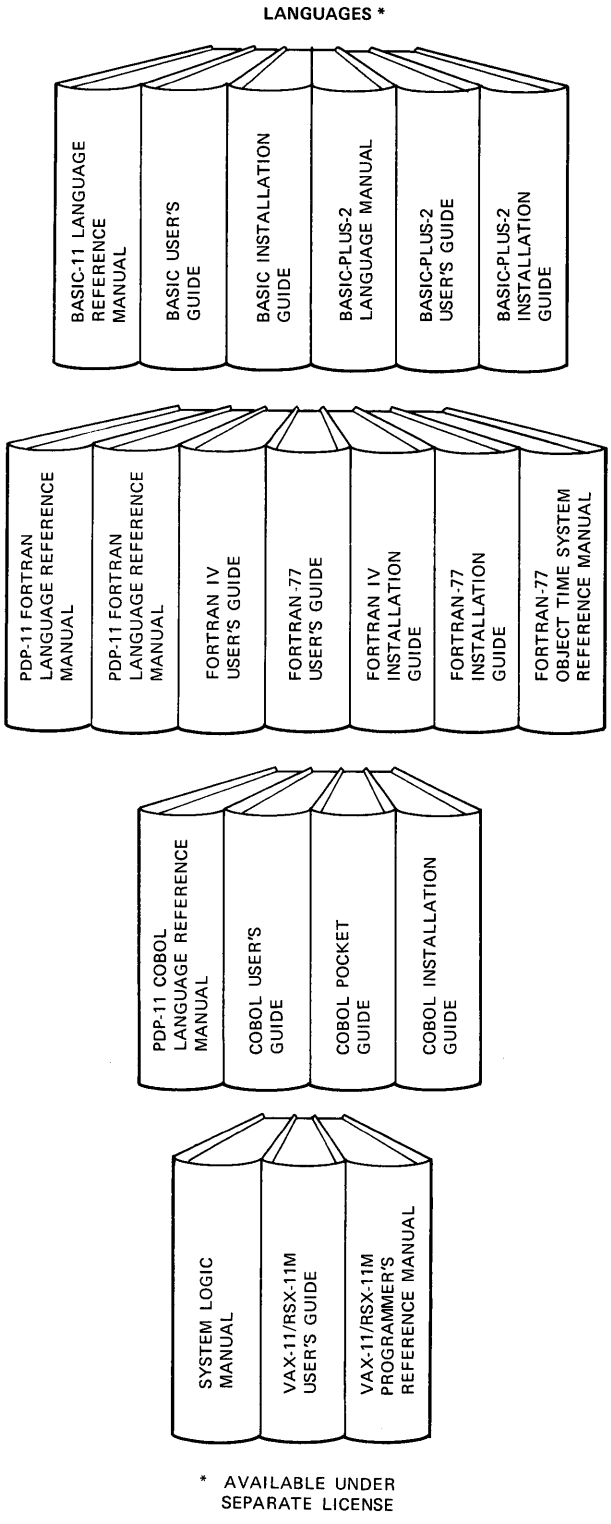


Figure 1-1 The Documentation (Sheet 3 of 3)

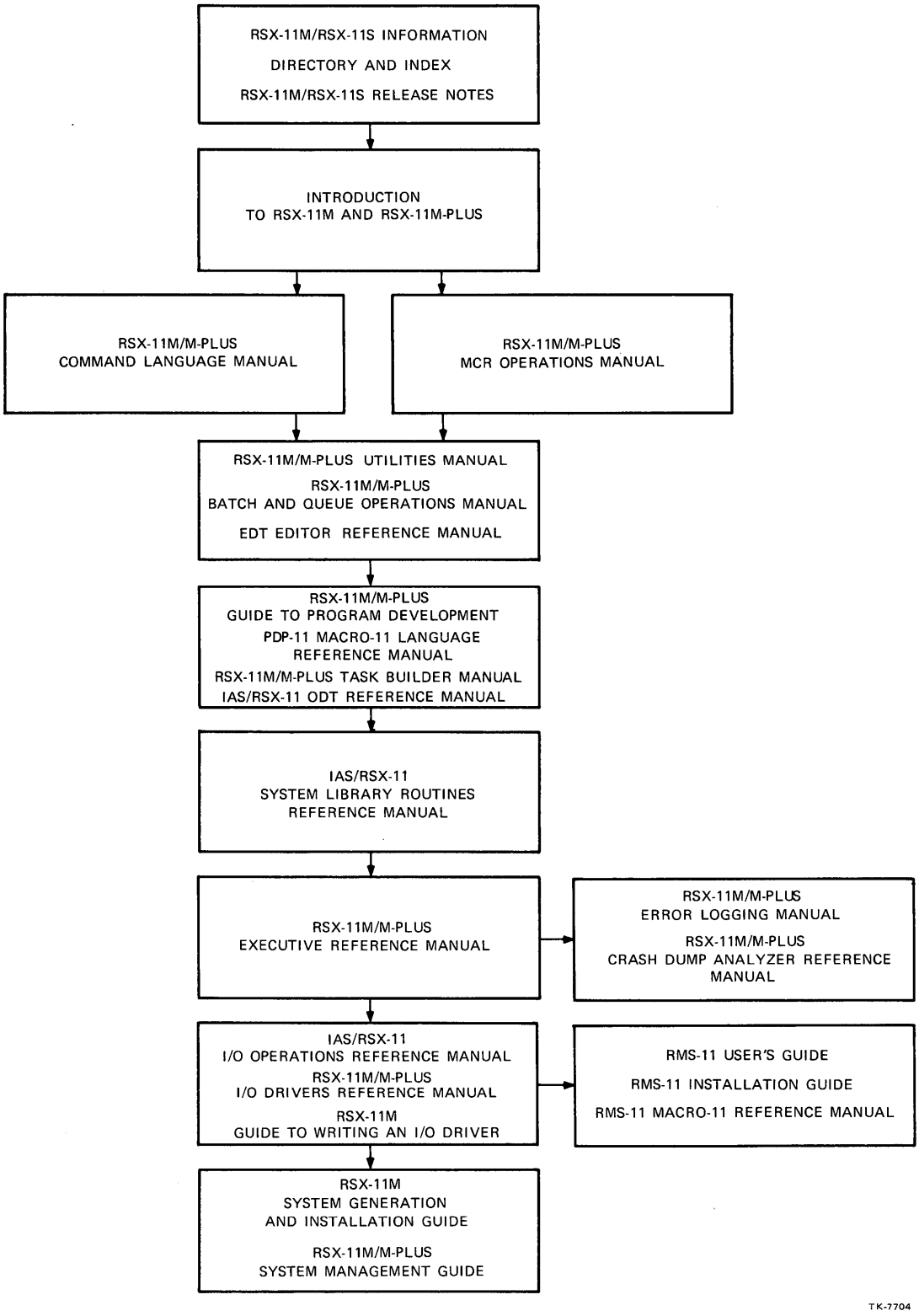


Figure 1-2 Sequence for Reading Manuals

Other Resources

In addition to the system documentation set, the following are other resources to increase your system knowledge.

1. RSX-11 Technical Summary - A technical introduction to many aspects of the three RSX real-time operating systems including information and Digital Support Services available with RSX systems.

2. PDP-11 Processor Handbook - A discussion of the various features of each PDP-11 CPU, the instruction set, UNIBUS, programming techniques, memory management, commercial instruction set and instruction timing.

3. PDP-11 Software Handbook - A discussion of the major operating systems, programming languages, file and record management, data base management, utilities and interprocessor communication available on the PDP-11.

4. DECUS - The Digital Equipment Computer User's Society - A worldwide association of customers and employees providing a forum for the exchange of useful information, new program packages and other innovations. (Membership is free to owners of DIGITAL computers.)

THE OPERATING SYSTEM

Before the operating system can be loaded into the computer's memory, your system manager (the person responsible for the operation and maintenance of the system) goes through a procedure called system generation (SYSGEN). This procedure builds the operating system, and fine tunes it with the hardware and software requirements of your computing site. At the end of the SYSGEN, the executable image of the operating system is written to a disk called the system disk. To start the computer operation, this executable image must be loaded into memory and an instruction issued to start the operating system executing in the central processor. This procedure is called "booting the system".

Memory Use

One of the resources that the operating system must manage is memory. Figure 1-3 shows how memory is used. Depending upon the operating system and the hardware, the amount of memory that can be connected can range from 16KW to 2MW. The operating system always resides in the low addresses of memory, and can range in size from 16KW to 20KW, depending on the software and hardware options generated in the SYSGEN process. Part of that 20KW (maximum) is an area called Pool or Dynamic Storage Region (DSR) set aside for the dynamic storage of executive data structures (These are little lists that the operating system uses to keep track of information about tasks and devices.) The top 4KW of physical memory addresses are reserved for the I/O page and are not available for general use. The remainder of memory is available for user programs.

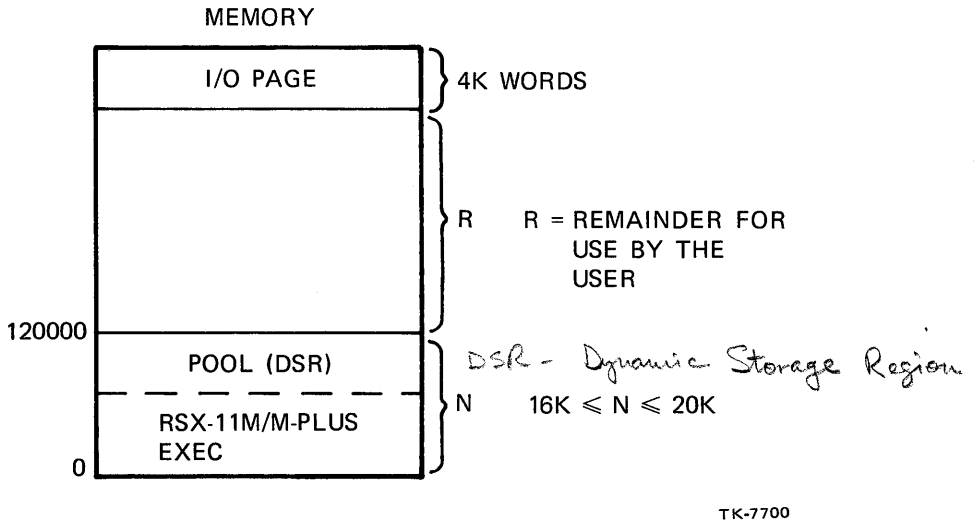


Figure 1-3 Memory Use

Tasks

The smallest executable piece of code on this operating system is a task. Tasks are problem solutions that have been processed through the program development cycle producing an executable task image (Figure 1-4). For this problem solution to execute in the CPU a user must make a request to the operating system to run that program. It is the operating system's responsibility to locate the task image on disk, check the availability of memory space, load the task into memory, and schedule the task to run.

Examples of tasks:

- Payroll Program
- Text Editor
- FORTRAN Compiler
- Monitor Console Routine
- File Maintenance Program
- Data Analysis Program

A task is built with two (or many) attributes that the operating system uses to schedule that task to run. These are the priority at which the task will run and the area in memory where it will be loaded (partition). Before a task can be run on the system, the operating system must know this information (and more) about it.

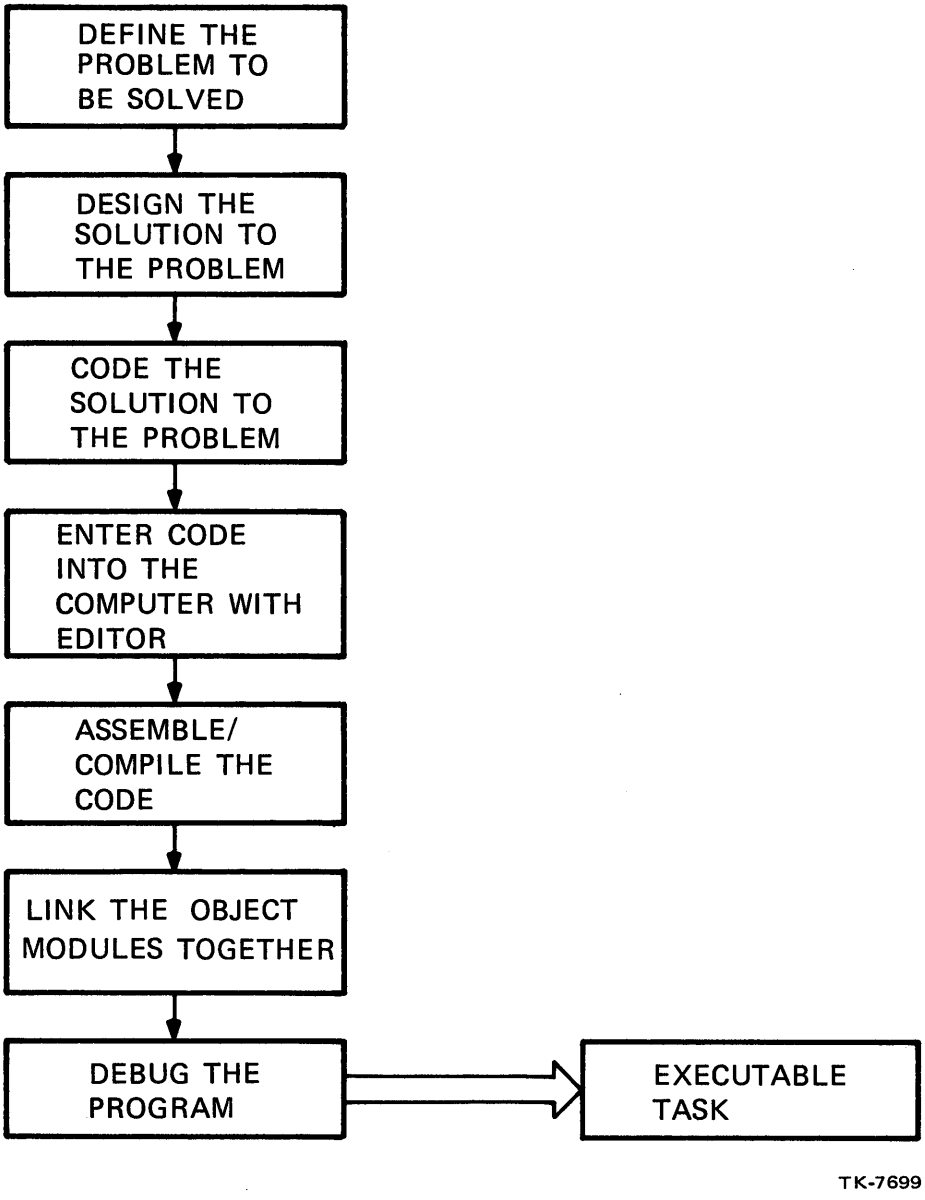


Figure 1-4 A Task is the Result of Program Development

Steps Required to Run a Task

1. A request-to-run must be issued for the task.
2. System checks to see if it "knows" this task.
3. System adds the task to a list of active tasks competing for resources.
4. Task image must be loaded into memory from disk.
5. Task is scheduled for CPU time.

The System Task Directory (STD)

The operating system maintains a list of known tasks within the area called Pool. This data structure is called the System Task Directory (Figure 1-5) which is a linked list of another data structure called the Task Control Block (TCB). The TCB contains such task information as the task name, the priority at which it will run, the location in memory where it will be loaded, and the disk address where the task executable image can be found. All tasks to be run on the system must have this information recorded in the STD. This process is called installing a task and is accomplished by the INSTALL operating system command. (Installing a task does not cause it to run.) Some tasks are permanently installed, and others temporarily installed in the STD for the time they are executing.

Active Task List (ATL)

When a user requests a task to be run, the operating system creates an entry into another data structure maintained in Pool called the Active Task List (Figure 1-6). This is a list of all tasks competing for the system resources (i.e., someone requested the task to run) at any one time.

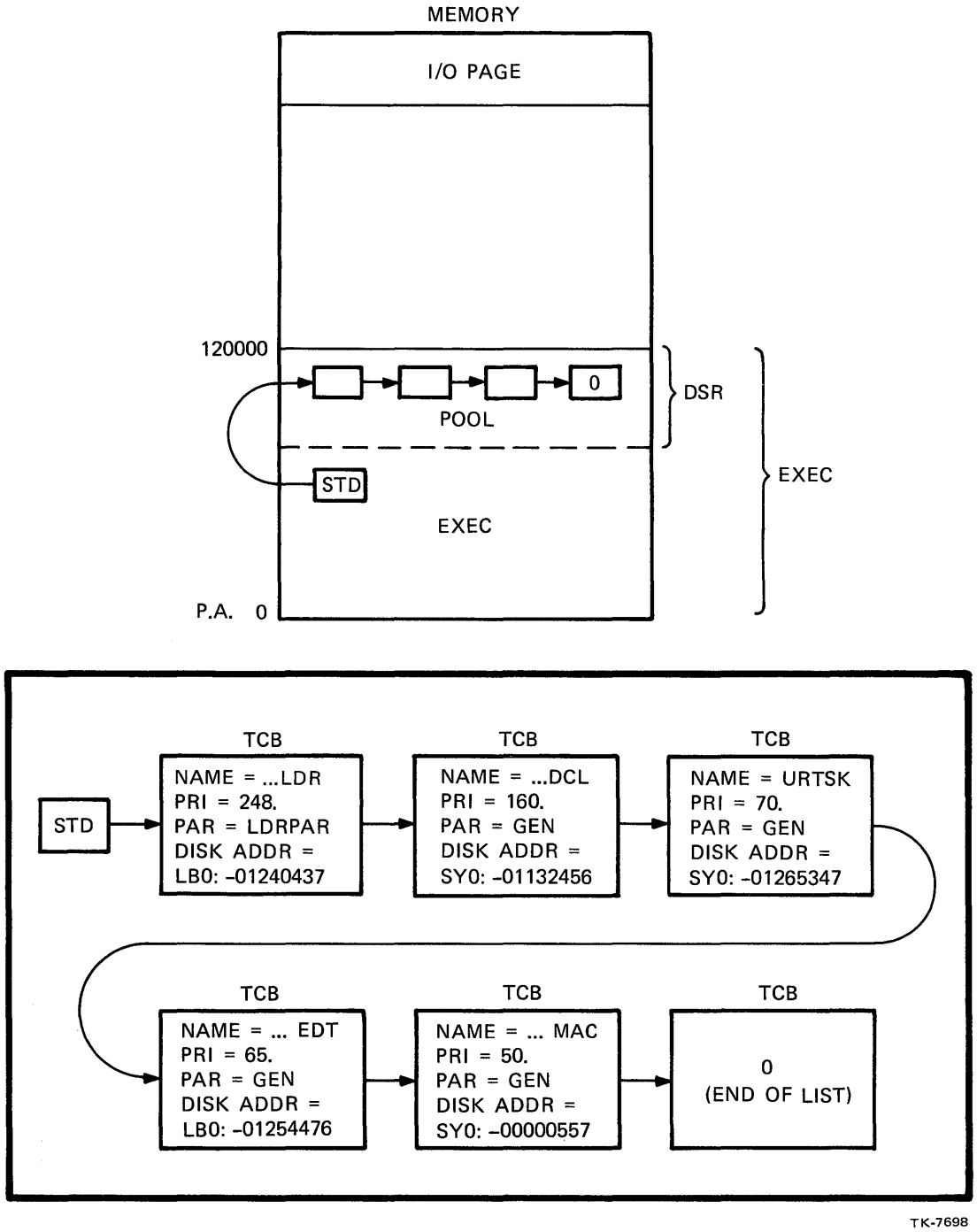


Figure 1-5 System Task Directory

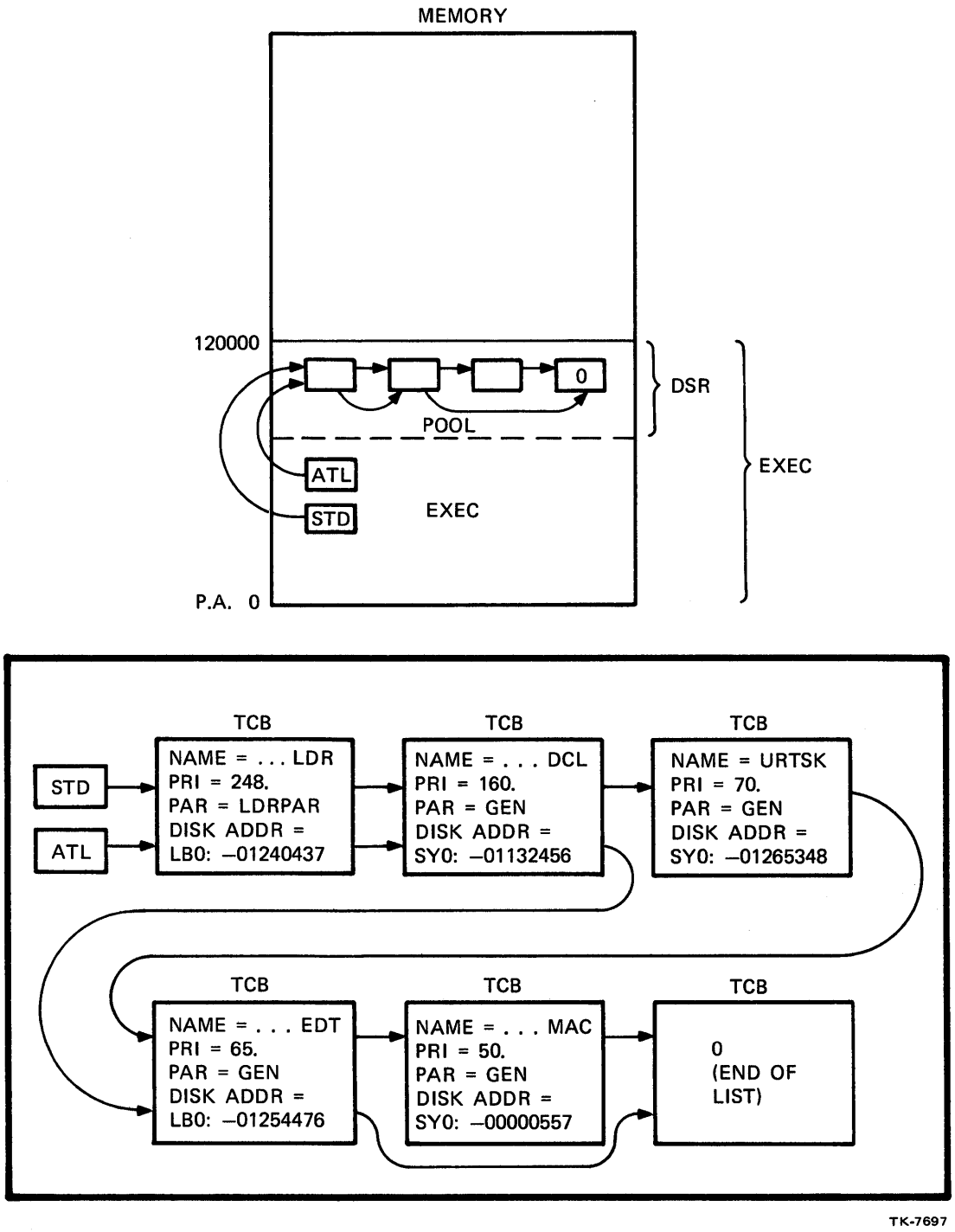


Figure 1-6 Active Task List

Allocating Memory to a Task

Before a task can gain control of the CPU and begin executing, it must be resident in memory. In other words, the operating system must determine the addresses where the task will be loaded into memory and then check to see if those addresses are available for use. Tasks compete for memory space by the task's priority and the partition of memory in which it has been built to run. A task can be built with a priority from 1 to 250. Real-time tasks are generally given a priority within the range of 150 - 250. 250 is the highest priority. Most tasks run at a priority of 50, which is the default priority for a task.

Figure 1-7 shows the priority range; notice that the Digital Command Language (DCL) runs at a priority of 248, while the Editor runs at a priority of 65. This means that when the operating system is looking for a task to run in the CPU, it will choose the Task Loader over the Editor if both are competing for the CPU.

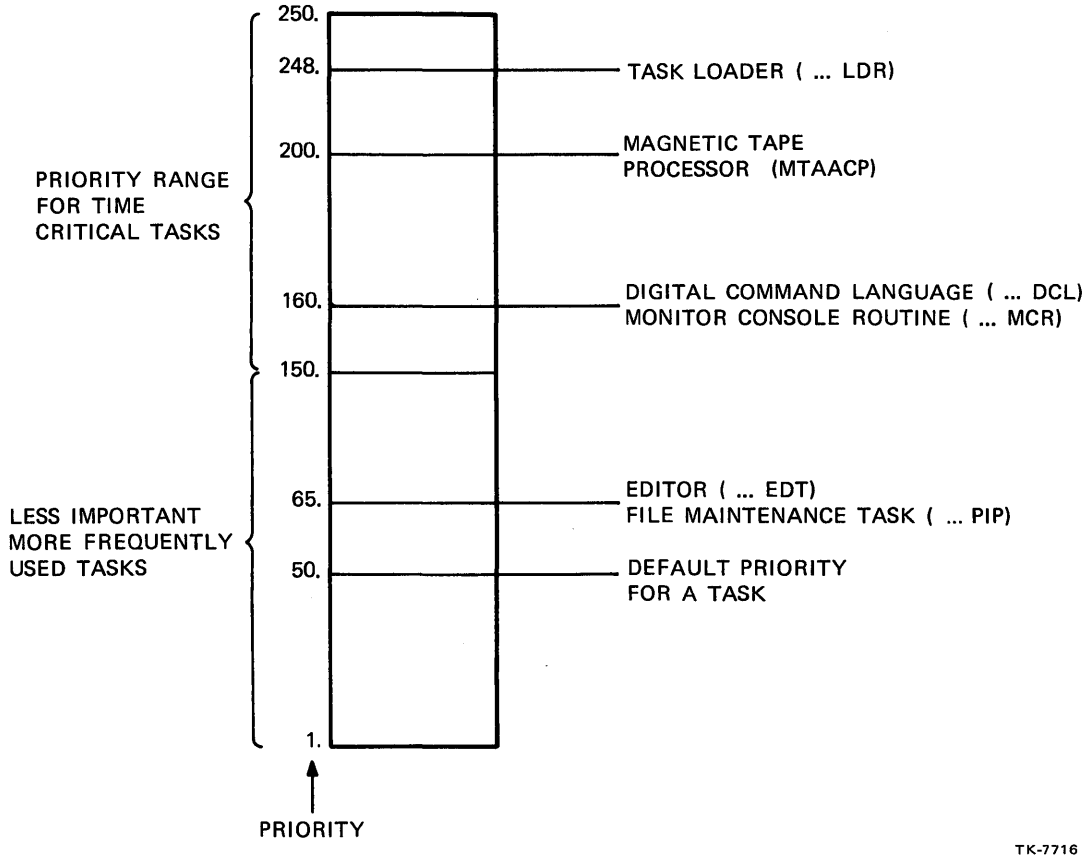


Figure 1-7 Tasks are Built with a Priority

Partitions

Tasks also compete for memory space depending upon the partition in which they were built to run. A partition (Figure 1-8) is a contiguous section of memory. The operating system takes up 16 to 2ØKW of low memory addresses, and the I/O page takes up the top 4KW. During SYSGEN, the system manager, taking into account your site requirements, divides up the remaining portion of memory into sections giving them a name, starting address, size and type. These sections of memory, called partitions, can be changed by the system manager when necessary.

In Figure 1-8 there are six partitions listed. Tasks can be loaded into each of these partitions. For example, DRVPAR is a special partition in which the code loads that controls the operation of devices (such as your terminal, disk drives, etc.). Generally, no other tasks are built to be loaded into this partition.

The default partition (GEN) is the one you get if you do not specify a partition when you build your task. Generally, most tasks are built to load into this partition and, therefore, compete with each other for this memory space when they are requested to run.

Figure 1-9 shows some tasks and their corresponding partitions. Notice that the Task Loader (...LDR) has its own partition. This task is the routine that loads all other tasks into memory. It is used often, so speed is important. To ensure that it always has a spot in memory in which to load, and does not have to compete with other tasks for that spot, a special partition was created just for it. This guarantees that the loader is available to load tasks when needed.

The size of a partition determines if more than one task can be loaded into it. The GEN partition is usually large enough to handle many tasks at one time. The number of tasks is dependent upon: how much memory is allocated to the partition (partition size), the mix of the number of tasks requested to run in the partition and their sizes, and the type of partition. On RSX-11M systems there are two types of partitions, user-controlled and system-controlled. RSX-11M-PLUS systems only have system-controlled partitions.

Table 1-2 lists the differences between system- and user-controlled partitions. The major difference is in the way partitions are subdivided. In a system-controlled partition (Figure 1-10), subdivision is done dynamically to accommodate the mix of tasks that have been requested to run in the partition at the time. At another time, the number of subpartitions and their sizes will be quite different, depending upon the mix of tasks again. In a user-controlled partition (Figure 1-11), the subpartitions are generated statically at SYSGEN time. There can be as many as seven subpartitions. The subpartitions can be changed but do not change dynamically at run time to accommodate different sized tasks. A task can be loaded into the main partition, in which case the subpartition will remain unavailable; or a task can be loaded into each of the subpartitions, in which case the main partition is unavailable.

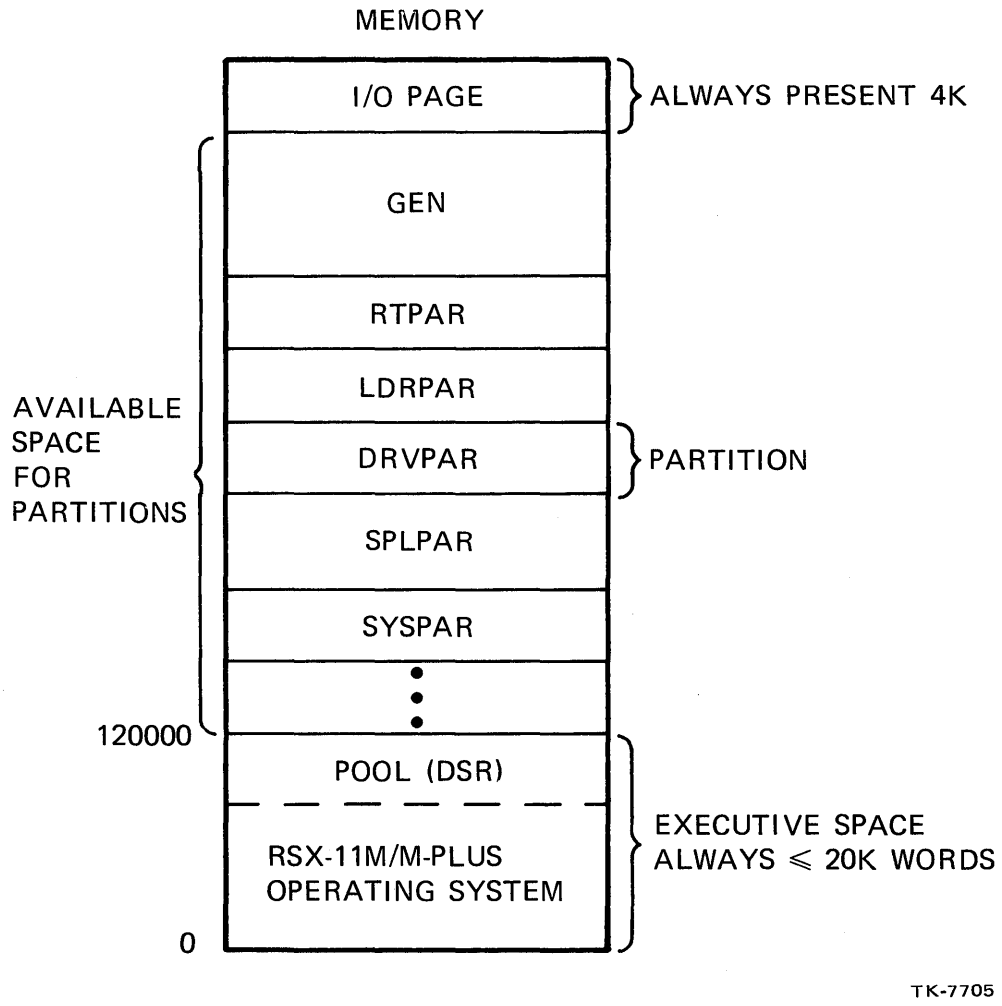


Figure 1-8 Memory Divided Into Partitions

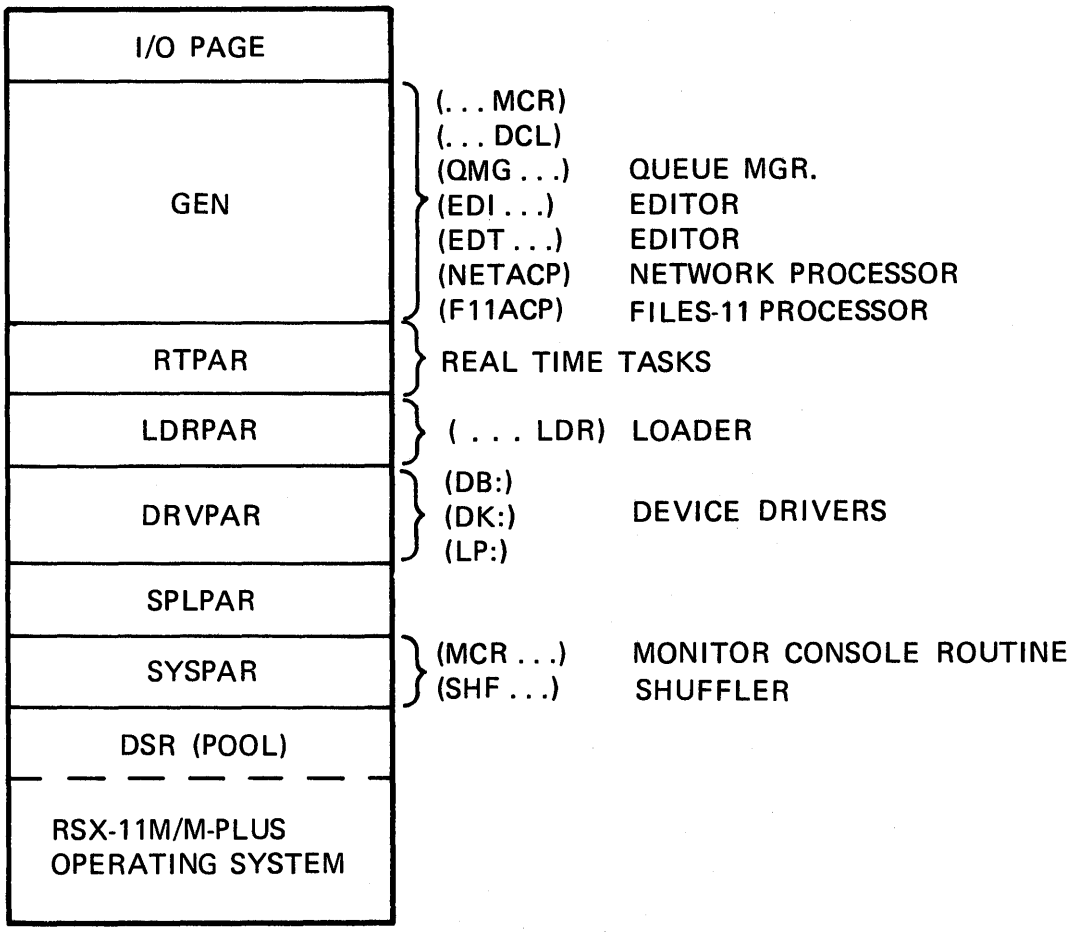


Figure 1-9 Tasks Execute in a Specific Partition

Table 1-2 Characteristics of System- and User-Controlled Partitions

Type	How Controlled	Characteristics
I	System-Controlled	• Partition subdivided dynamically. • Subpartitions vary in size and number depending on tasks residing in memory.
II	User-Controlled	• Partition subdivided statically. • Subpartitions size and number do not change. • Can have at most seven subpartitions. • Tasks can reside in each subpartition or one task in main partition, but not both.

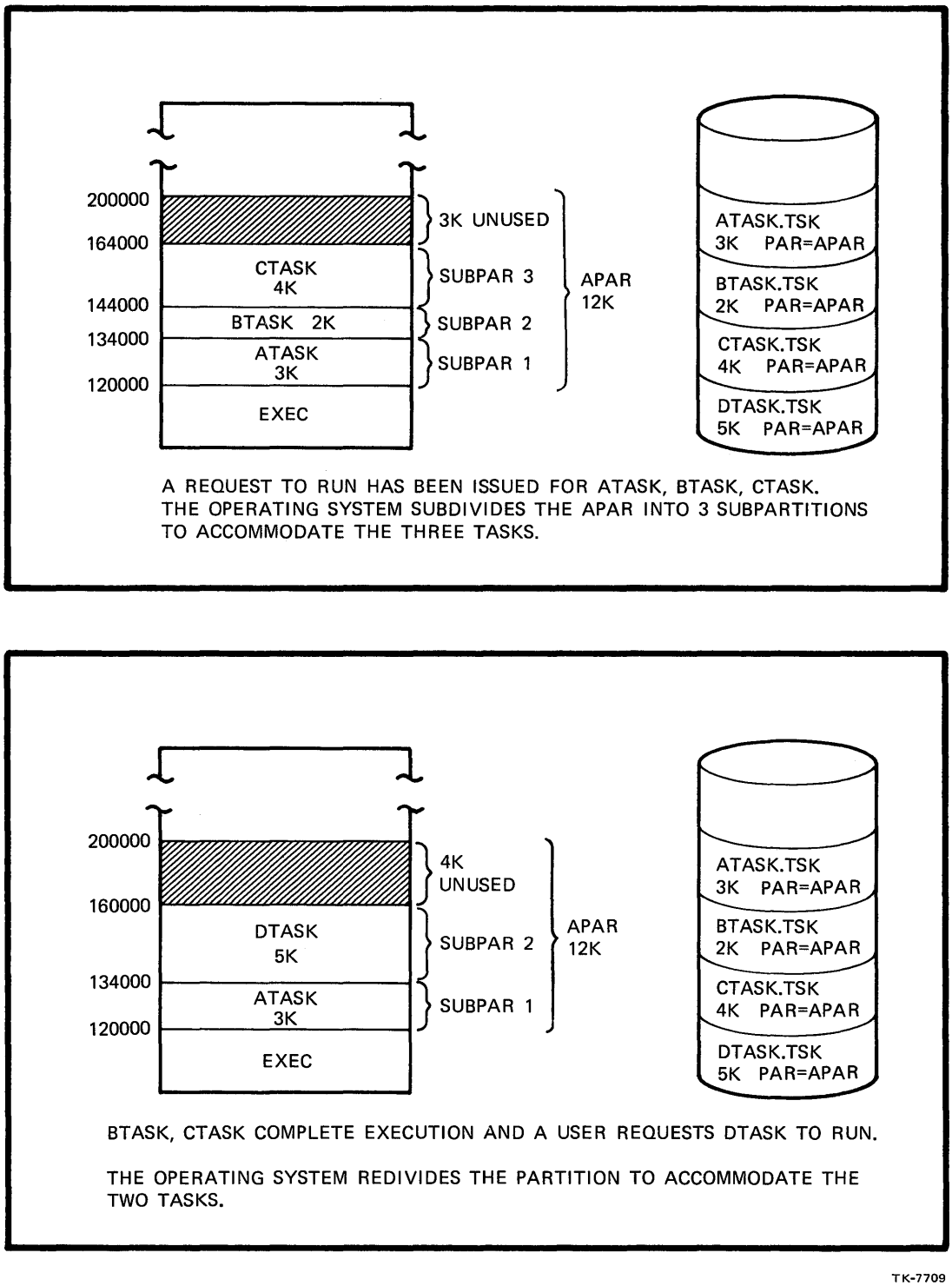


Figure 1-1Ø Running Tasks in a System-Controlled Partition

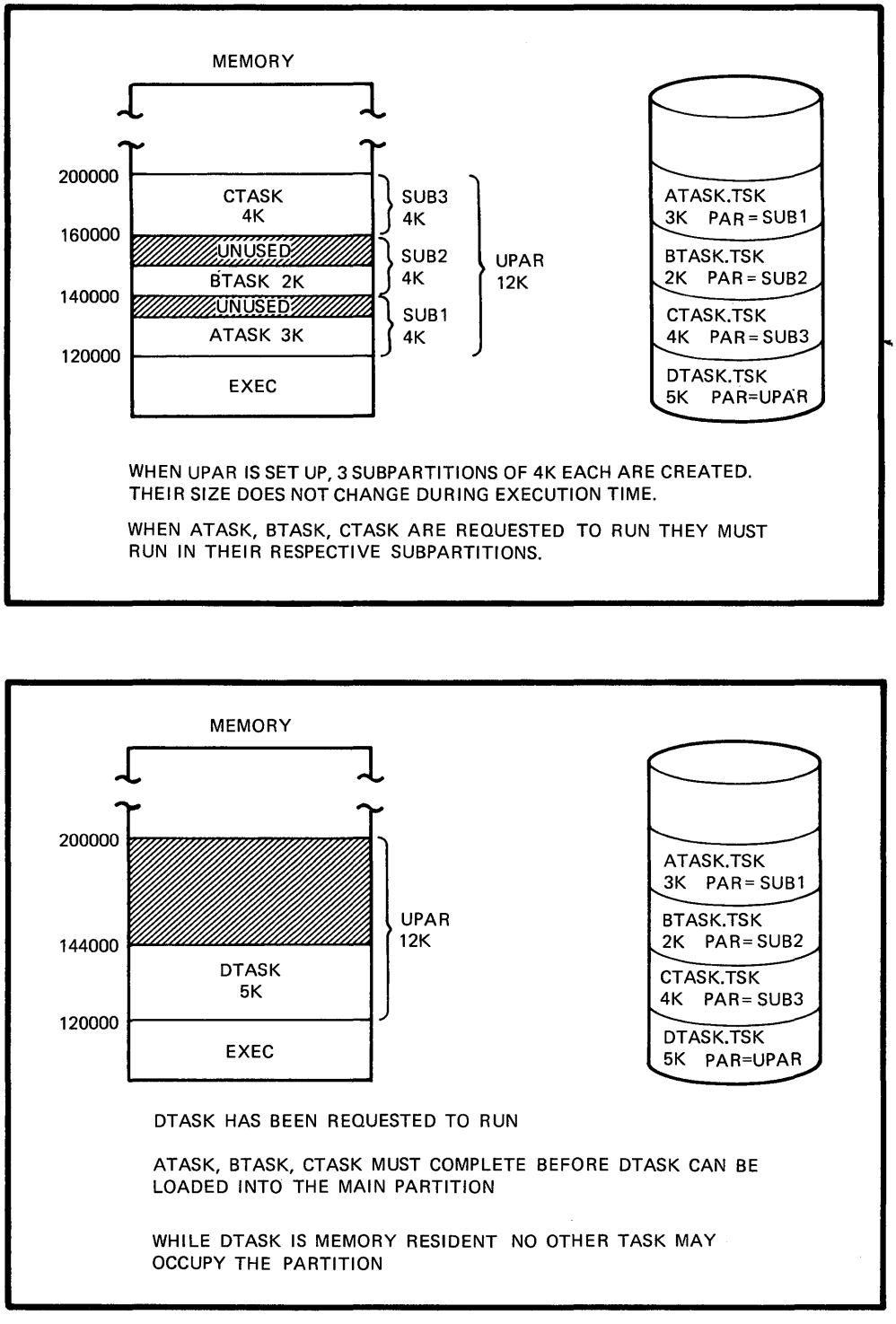


Figure 1-11 Running Tasks in a User-Controlled Partition

Checkpointing

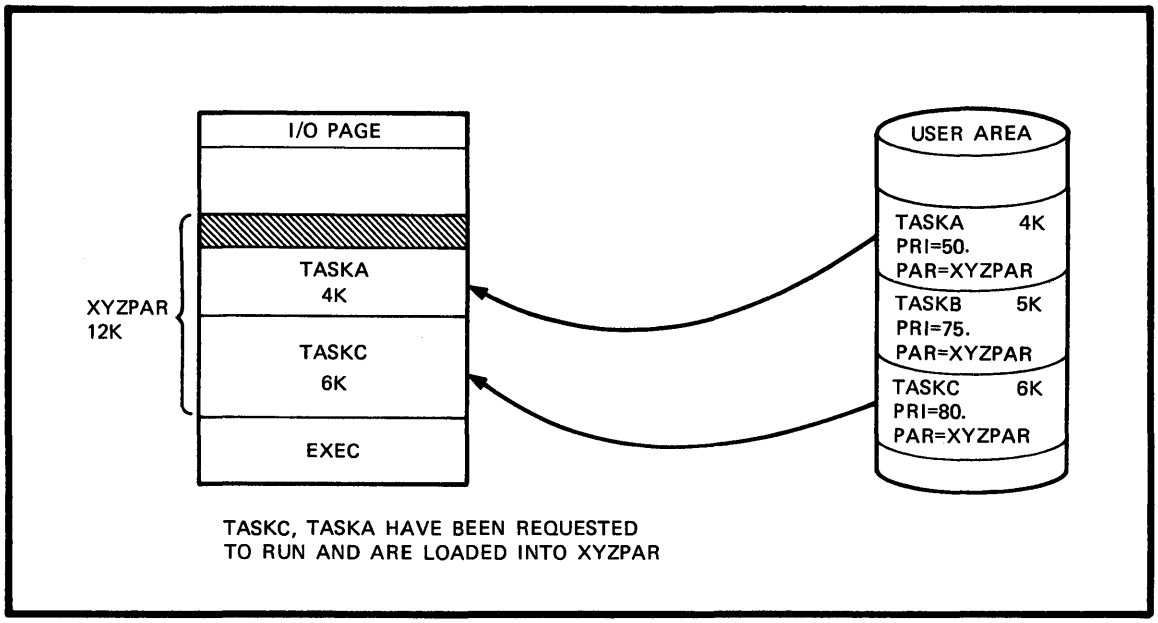
A task's ability to remain in memory (discounting the condition of completion or aborting) depends upon another concept called checkpointing. Checkpointing is the removal of a task from its partition (before it has completed execution) and temporarily storing its memory image on disk to allow a higher-priority task to replace it in the partition. Figure 1-12 shows this procedure.

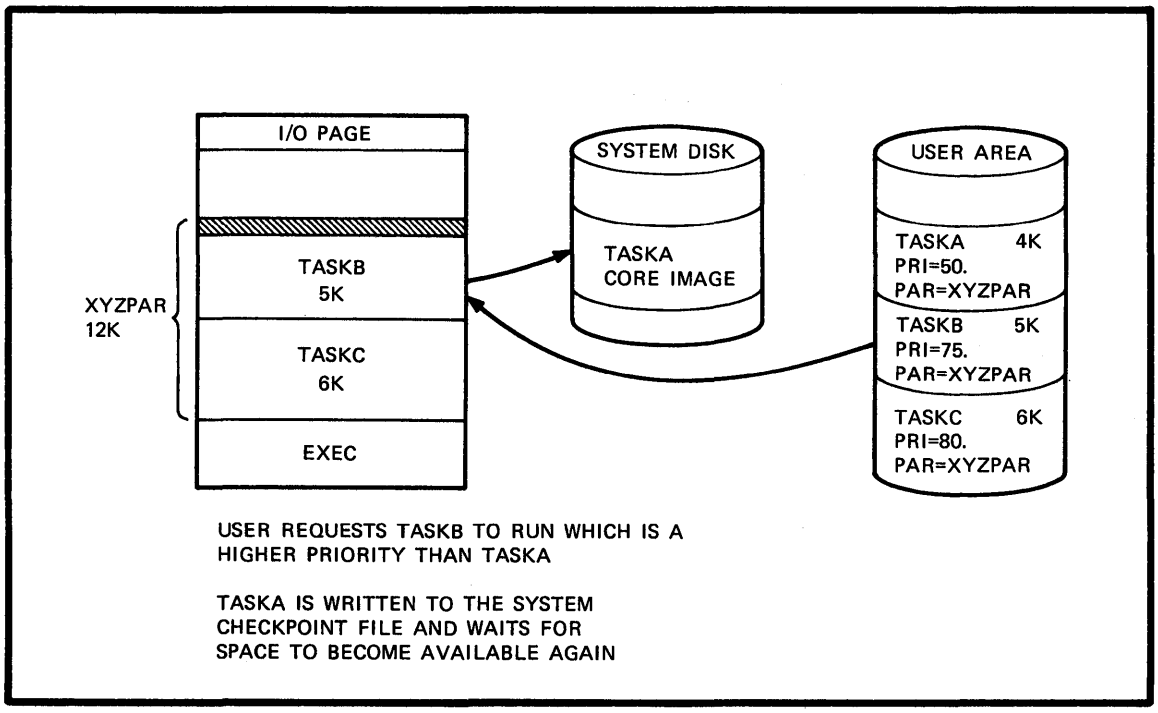
1. TASKA has been built for XYZPAR with a priority of 50. A request to run TASKA is made. Free space is available in the partition, so TASKA is loaded into memory.

2. TASKC has also been built to run in XYZPAR but has a priority of 80. A request to run is made for TASKC. There is room in the partition for both tasks, so TASKC is loaded.

3. TASKB, which also has been built to run in the XYZPAR partition, has been requested to run. The operating system, in finding that there is no room in the partition to load the task, checks to see if there is a task of lower-priority that it can remove temporarily to accommodate TASKB. It determines that TASKA meets those requirements, copies TASKA's memory image to a file (called the system checkpoint file) on disk, and then loads TASKB into the partition. TASKB then competes with other active tasks for CPU time.

4. When space becomes available again, TASKA's saved memory image will be loaded back into the partition from the checkpoint file, and TASKA will again compete for CPU time.





TK-7717

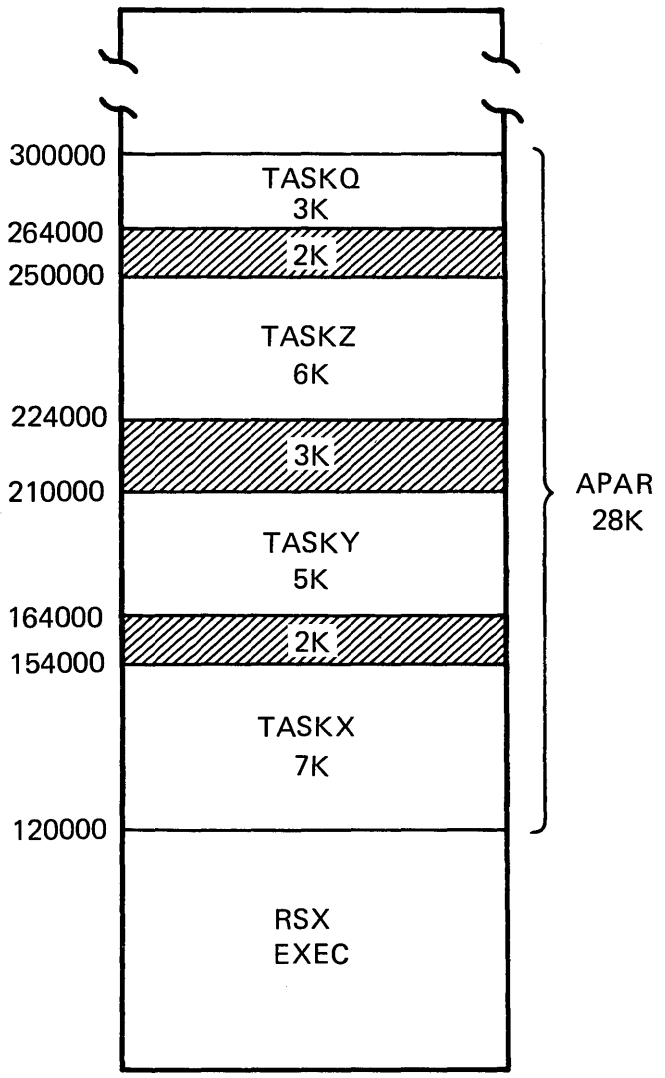
Figure 1-12 Checkpointing a Task

Shuffling

Tasks are constantly being loaded, checkpointed and completed. When activity in a partition continues like this for a period of time, fragmentation of unused space can occur, as shown in Figure 1-13. In the partition, APAR, there is 7KW of unused space that could be used to load another task. However, because it is not contiguous space, it prevents another task from being loaded. There is a system routine called the SHUFFLER that corrects this fragmentation problem within a system-controlled partition. Figure 1-14 shows the function that the SHUFFLER performs.

1. Due to heavy activity on the partition, there are two sections of memory available, a 1K section and a 3K section. Together the space would be large enough to load in TASKZ. However, because the space is not in one section, TASKZ is blocked from being loaded into memory.

2. When the shuffler is invoked by the operating system, it searches through the partition for free space. When space is found, tasks are moved down so that the free space (in this case, 4K of free space) is in one area. Then TASKZ is no longer blocked and can be loaded into the space.



DUE TO HIGH ACTIVITY WITHIN A PARTITION FRAGMENTS OF UNUSED SPACE CAN OCCUR.

IF THIS FREE SPACE WERE CONTIGUOUS ANOTHER TASK COULD POSSIBLY RESIDE IN THE PARTITION.

TK-7707

Figure 1-13 Partition Fragmentation

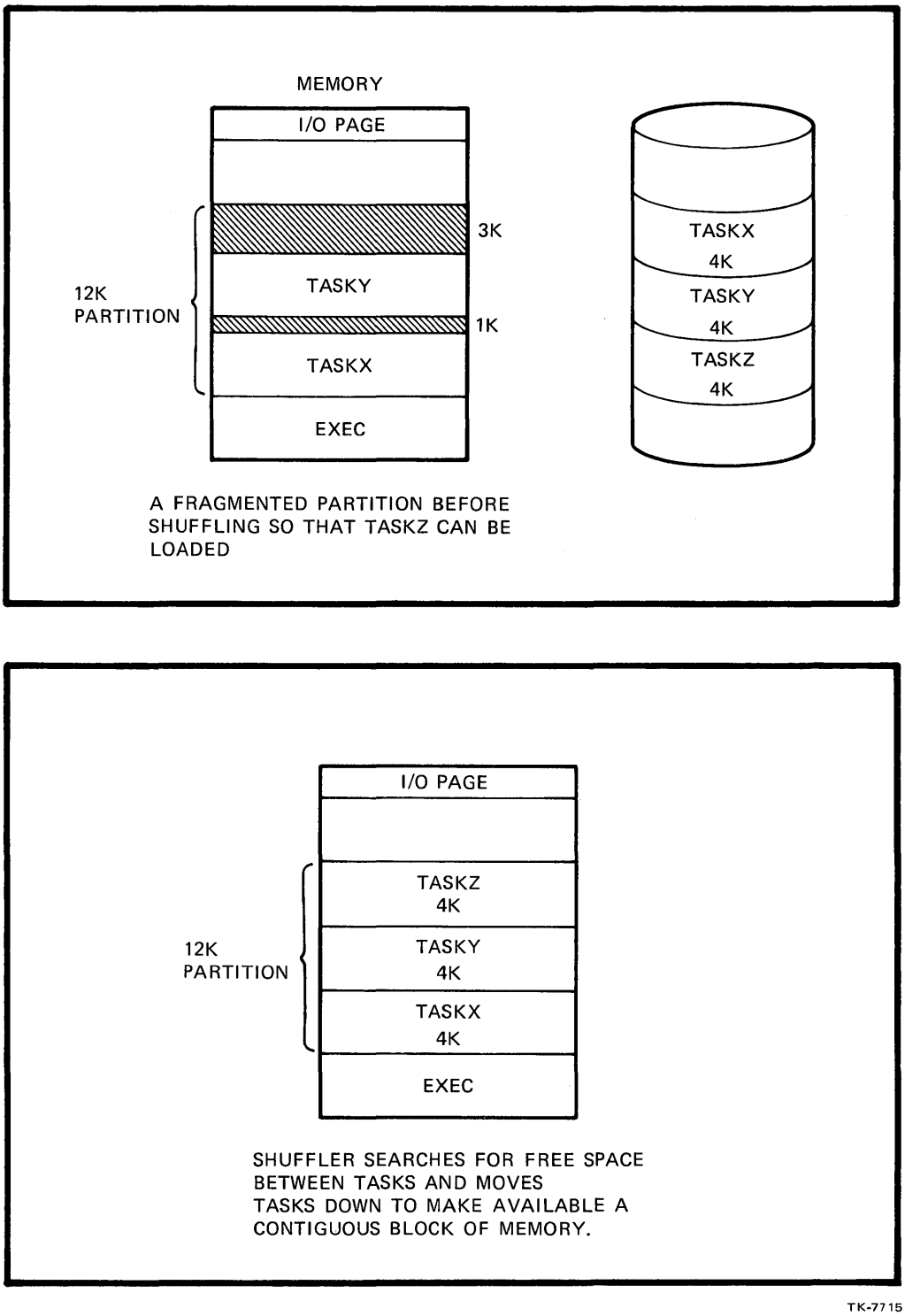


Figure 1-14 Partition Shuffling to Obtain Contiguous Memory

Allocating CPU Time to a Task

Once a task has won a spot in memory, it is then able to compete for CPU time. A task is given control of the CPU based upon its priority and its task state. The CPU is always given to the highest-priority, ready-to-run task. The operating system schedules the CPU at certain intervals called significant events (Table 1-3). Whenever a significant event occurs, the operating system reconsiders the state of all active tasks and determines which task will be given the CPU.

Figure 1-15 shows the different states of a task:

1. KNOWN - A task that has had an entry placed in the System Task Directory (STD). The INSTALL system command is used to create a TCB with a task name, priority, partition and disk address, and then links the TCB into the STD.

2. UNKNOWN - A task that has not been installed in the STD. There are many more unknown tasks in the system than there are known tasks.

3. ACTIVE - An installed task that has been requested to run and is competing for system resources.

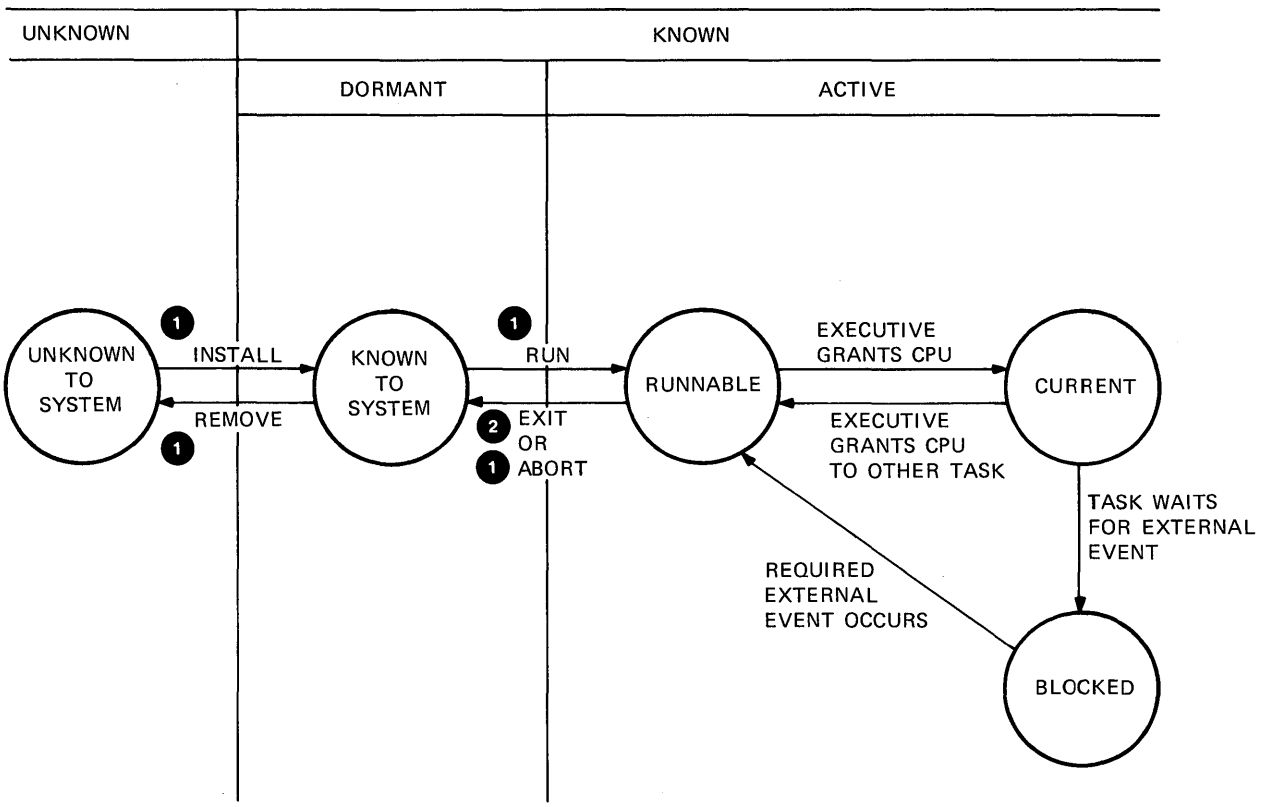
4. DORMANT - An installed task that has not been requested to run.

5. RUNNABLE - An active task that is ready-to-run.

6. CURRENT - An active task that currently has control of the CPU.

7. BLOCKED - An active task that is kept from running due to lack of resources.

A task changes state when some event applies to it. For example, an UNKNOWN task becomes KNOWN when the INSTALL command is used to make it known. A KNOWN task becomes ACTIVE when a user requests it to be run with the RUN command. An ACTIVE task becomes RUNNABLE when all the resources it needs are available.



- A TASK REMAINS ACTIVE FROM THE TIME IT IS REQUESTED TO RUN, UNTIL THE TIME IT EXITS OR IS ABORTED.
- THERE MAY BE ANY NUMBER OF ACTIVE TASKS AT A GIVEN TIME.
- THERE IS ONLY ONE CURRENT TASK AT ANY GIVEN TIME.

1 OPERATING SYSTEM COMMAND USED TO CHANGE THE STATE OF A TASK.

2 TASK COMPLETES EXECUTING AND ITS STATE CHANGES.

TK-7718

Figure 1-15 Task States

Table 1-3 Significant Events that Cause Rescheduling of the CPU

Events that Cause an ATL Scan from Task Position Down	Events that Cause an ATL Scan from Top Down
Task enters wait state • Waits for an event flag • Significant event • "or" of event flags • Suspends itself • Initiates synchronous I/O Task exits/aborts/is aborted	I/O completion Program declares a significant event Mark time expiration Program issues a Send Data directive RUN command issued

Round Robin Scheduling

When the CPU is idle and not servicing real-time tasks, its time can be divided among many tasks giving each a slice of time in which to execute. This time-sharing method of scheduling CPU time is implemented through the round robin scheduler. It is a system generation option that your system manager most generally will incorporate into your operating system. It only applies to tasks running within a certain range of priorities set up by the system manager. Generally, that range is from 1 to 150. Real-time tasks are never considered in this scheduling algorithm. At a set interval, the round robin scheduler executes in the manner shown in Figure 1-16.

1. Round robin scheduler traverses the ATL looking at tasks within a priority.

2. Within the priority, the scheduler will rotate the positions of the tasks, so that the first task in the list becomes the last task within the priority, the second task then becomes the first task within the priority, etc.

3. The scheduler then moves on to the next priority and rotates the task's position in the same manner.

The procedure allows for fair distribution of CPU time among tasks within the same priority. Without the round robin scheduler, the first task in the priority range would always get the CPU until it completed or aborted. Then the second task in the priority range would get control.

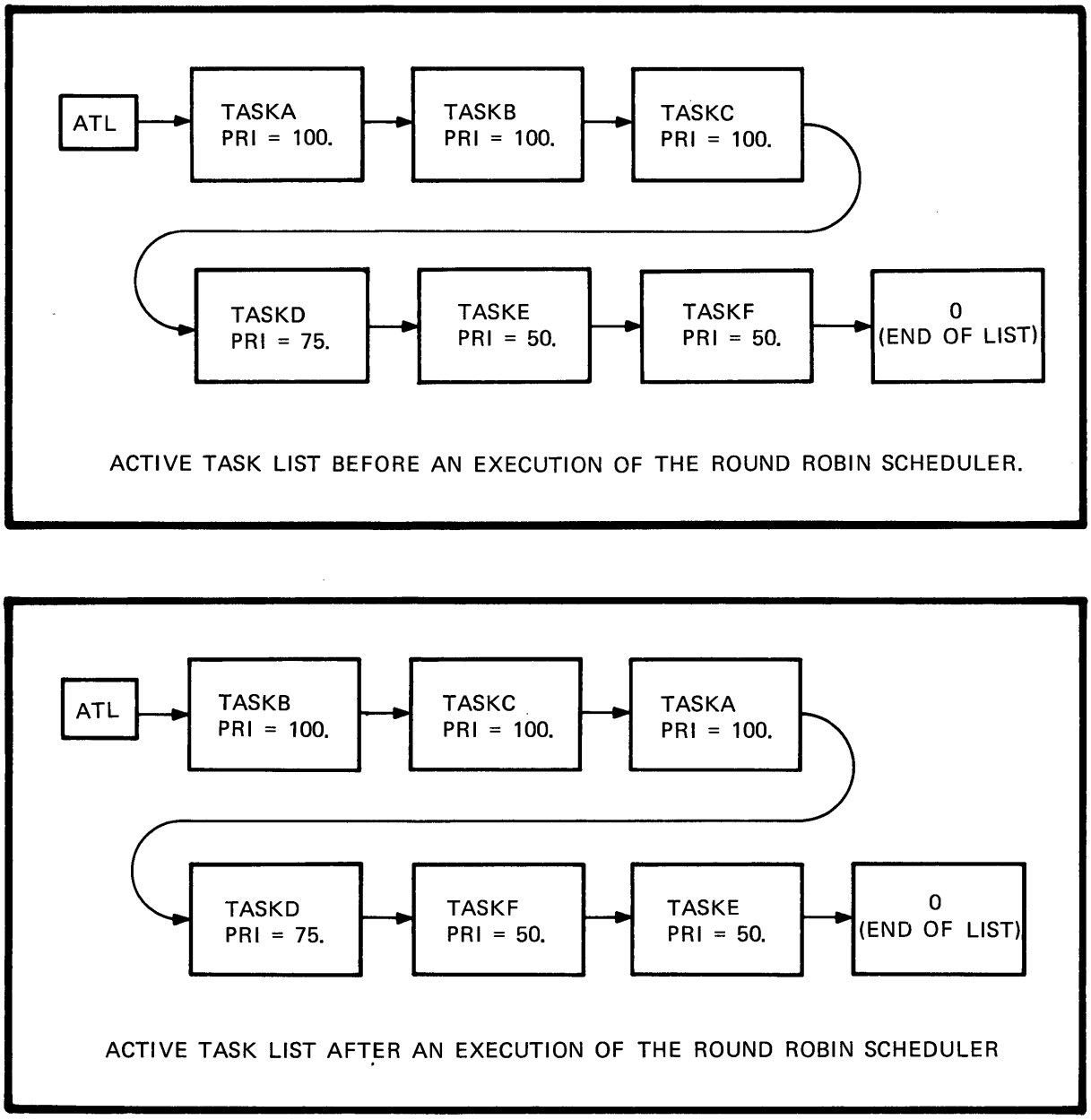


Figure 1-16 Round Robin Scheduling

Task Addresses

When a task is developed, source code instructions get converted into machine instructions, each with an address. These addresses are referred to as virtual addresses; that is, they are not the final addresses, or they do not refer to the actual physical memory address of where the program will reside. They are the addresses used in listings, debugging, etc. A program's virtual addresses can range only from Ø - 177777 (Ø-32KW). Figure 1-17 depicts the concept of a task's virtual address space.

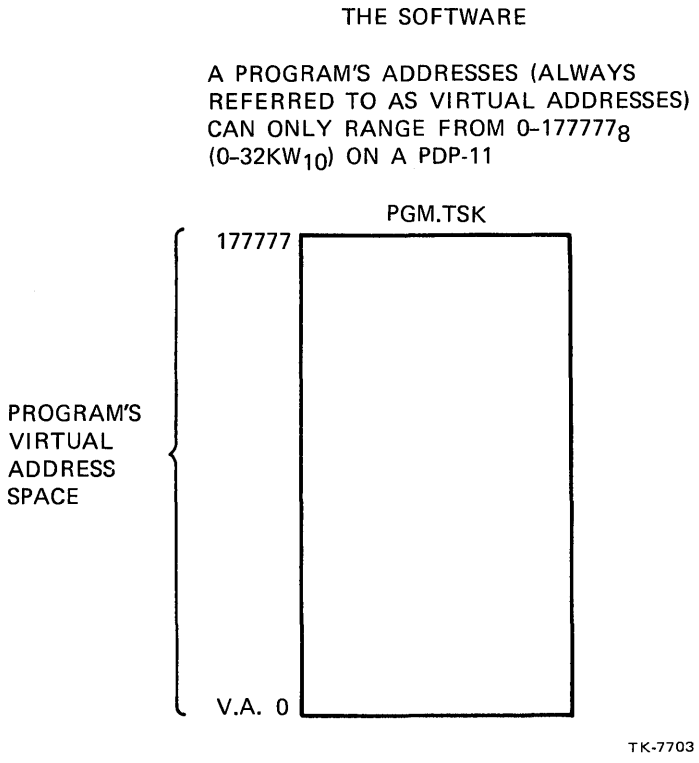


Figure 1-17 Virtual Address Space

Physical Addresses

The physical address refers to the actual physical memory location (Figure 1-18) where a task resides. Physical addresses can range from Ø to 2MW. This range is determined by the actual amount of memory on your system.

Because the PDP-11 hardware is a 16-bit architecture, and the largest address that can be contained in 16 bits is 65,536 KBytes or 32K words, if you wish to access a memory location larger than 32K words, you need some assistance. This assistance comes through adding the KT-11 Memory Management hardware to the system. A computer system **without** this hardware is called an unmapped system and will have at most 32K words of memory. A system **with** this hardware is called a mapped system and can have as much as 2 megawords of memory.

When a program is loaded into memory, the operating system's responsibility is to map the program's virtual address into the physical memory location where it will reside.

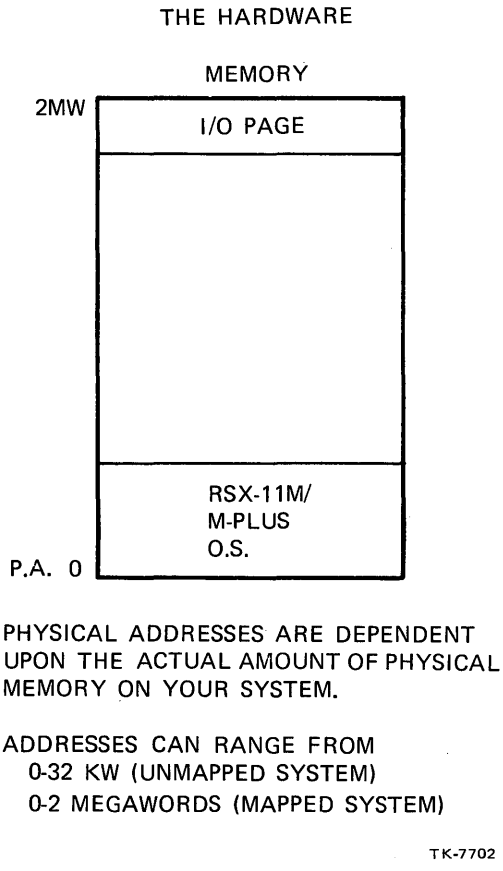


Figure 1-18 Physical Address Space

Mapped Systems

A PDP-11 system that has the KT-11 Memory Management hardware attached is called a mapped system. This hardware provides for the conversion of a 16-bit virtual address into an 18- or 22-bit physical address. This feature allows a task to address a memory location greater than 32K words and also to have up to 2 megawords of memory on the system. The KT-11 hardware also provides for the relocation of tasks in memory because an absolute physical address need not be built into the task's image. The operating system in conjunction with the KT-11 hardware generates the physical address, indicates where to load the task, and then later, if necessary, is able to move the task by remapping the task addresses (Figure 1-19). The KT-11 hardware also provides security by checking for memory protection violations by a task.

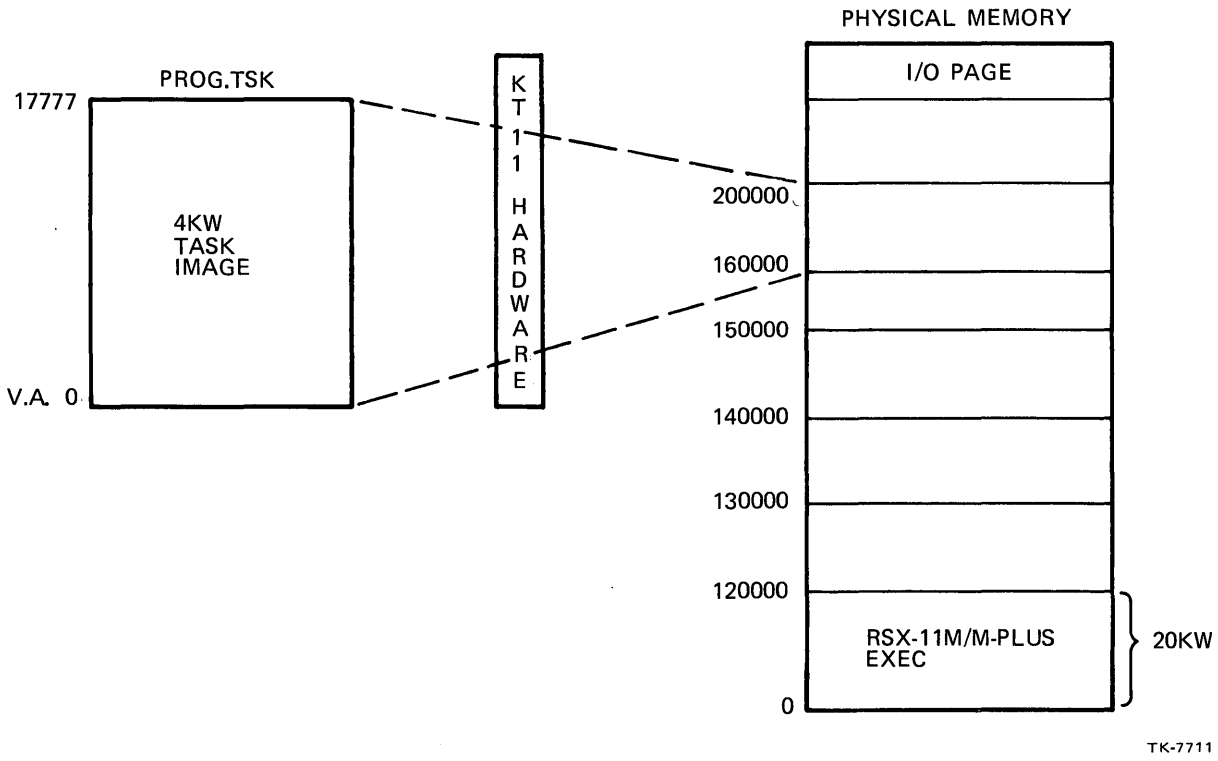


Figure 1-19 Mapping a Virtual Address Into a Physical Address on a Mapped System

Unmapped Systems

An unmapped system is one without the KT-11 hardware and, therefore, can have at most 32K words of memory attached to it. Tasks on this system are built to load at an absolute physical address (Figure 1-2Ø). Task addresses map directly to their physical addresses. To run the task at another physical address, the task must be rebuilt using the new desired address.

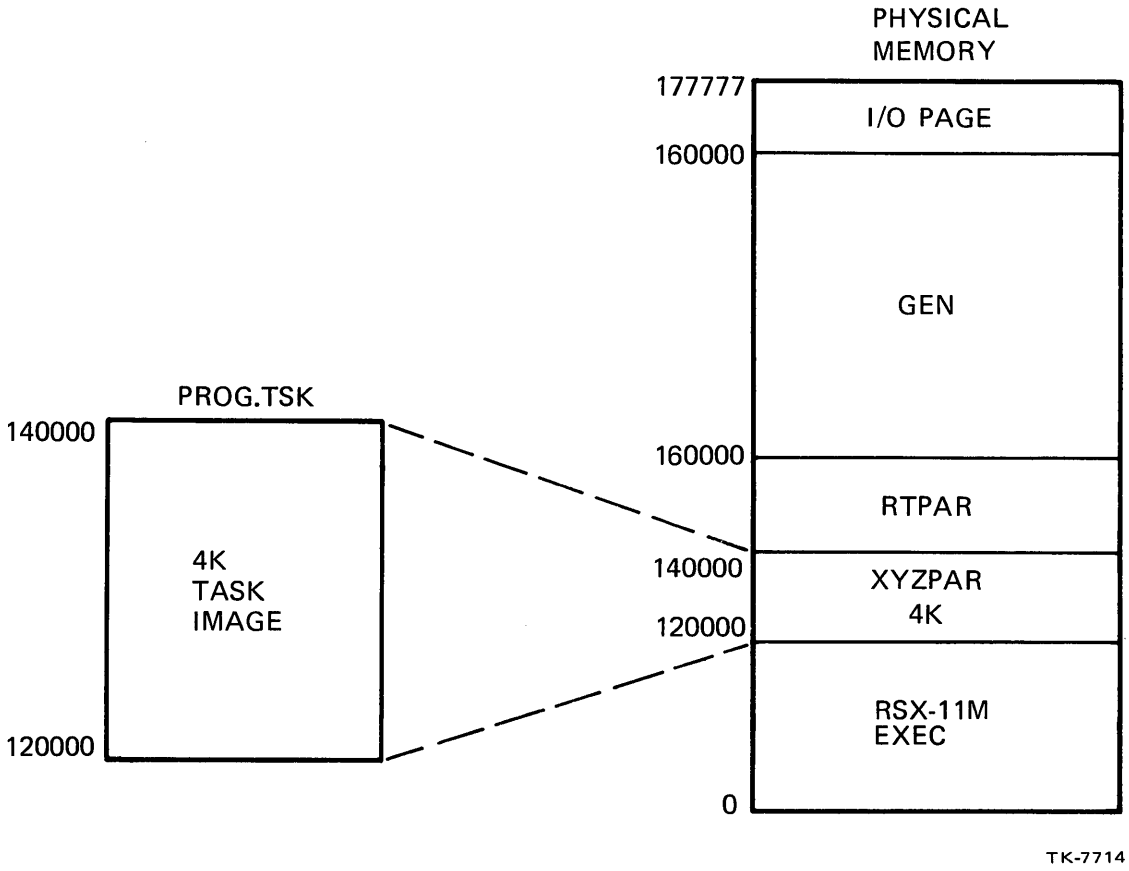


Figure 1-2Ø Task Built to Execute on an Unmapped System

2

GETTING STARTED ON THE SYSTEM

INTRODUCTION

You interact with your RSX-11M/M-PLUS system through a terminal at which you type commands. Most typed commands are processed by the Command Line Interpreter (CLI).

In this module you study the basics of terminal interaction including: 1) the conventions available for terminal use and communication with system-supplied CLIs, 2) how to issue system commands, and 3) how to get on-line system assistance.

OBJECTIVES

1. Log on and off an RSX-11M/M-PLUS system.
2. Control terminal behavior using control characters.
3. Use system HELP facilities.
4. Invoke system operations using MCR or DCL commands.
5. Switch Command Line Interpreters.

RESOURCES

1. RSX-11M/M-PLUS MCR Operations Manual
2. RSX-11M/M-PLUS Command Language Manual

LOGGING ON THE SYSTEM

Before you can attempt to do any work on the system, you must go through a procedure called "logging on" in which you supply information when prompted by the operating system. This logging on procedure identifies you as a valid user of the system. One of the features of an operating system in a multiuser configuration is system security. By asking for your name and password and comparing this to information stored in the computer by your system manager, the operating system can verify whether you are a valid user. If you are a valid user, the operating system performs some initialization steps, and then grants you access to the system resources.

Before you can attempt to log on to the system, you must ask your system manager to set up an account for you in the system accounts file. The accounts file contains the following information about each authorized system user:

- User Identification Code
- User Name
- User Password
- User Default Command Line Interpreter (CLI)

User Identification Code (UIC)

The User Identification Code is a number assigned to a user that uniquely identifies him from other users of the system.

It has the form [g,m] where:

g = an octal number from 1-377 that defines a group of users.

m = an octal number from 1-377 that uniquely identifies a group member.

A group number is usually assigned to a particular group of users such as the payroll department, or the production group. Each user in each of these departments would have his own unique member number. Suppose the production group was given a group number of 305 and you, a member of that group, were given a member number of 7. Your UIC would then be [305,7]. Suppose that the payroll department was given the group number of 210 and your friend, a member of that group, was assigned the member number of 105. His UIC would then be [210,105]. This number is used to keep track of all the privileges and information that you have stored in the system.

Passwords

When your system manager sets up an account for you, he will ask you to supply him with a string of six characters (or less) that will constitute your password. This string of characters will have to be supplied every time you log into the system. It is not echoed (i.e., displayed on the terminal) by the operating system when you respond to the password prompt. The operating system will check the accounts file to see if the password supplied with the UIC or name matches exactly with what is stored there. If it does not match exactly, access rights to the system resources will not be given to you. Your password can be changed by the system manager at any time if you request it.

Command Line Interpreter (CLI)

Once you have successfully logged into the system, a routine called a Command Line Interpreter (CLI) becomes active at your terminal. This routine will accept commands made up of keystrokes and a line terminator from your terminal, do some interpretation on the command line and pass it along to the operating system for action. There are two different CLIs available with the operating system; computing sites may write their own interpreter. You must tell your system manager (when he creates your account) which one of these interpreters you wish to have active at your terminal when you log in. This will become your default CLI, discussed later in this module.

Notes on Example 2-1

Example 2-1 shows the log in procedure. The following comments are keyed to the example.

1. User logs in (LOGIN or HELLO), supplying system with a UIC.
2. User supplies his password, but the operating system does not echo the characters.
3. Message indicates what system is being logged into, the date, time and terminal number.
4. System messages displayed at terminal.
5. User logs off the system. (LOGOUT or BYE)
6. User logs on using the slash (/) as a delimiter in the UIC to suppress system messages. You will notice that the messages shown in note 3 above are not displayed this time.
7. User logs on using user name instead of UIC.
8. User logs on, system prompts for the account or name and the password.

```
1 LOGIN 305,303
2 Password:
3 RSX-11M BL32   [2,54] System     QUASAR
  06-JAN-82 20:02 Logged on Terminal TT56:

  Good Evening

4 6-JAN-81        Kogan

        QUASAR will be down FRIDAY (JAN/8/82) from 11:30 to 13:00.
        Please let me know if you have any objections.

  30-DEC-81       Siler

        The DCL that's now in [2,54] (and installed by STARTUP) is built so
        that it passes unrecognized commands to MCR.  If your terminal is set
        to DCL, you can use whichever form of command is easier, MCR or DCL.

5 >LO
  Have a Good Evening
  06-JAN-82 20:02 TT56: logged off QUASAR
6 >LOGIN 305/303
  Password:

  RSX-11M BL32   [2,54] System     QUASAR
  06-JAN-82 20:02 Logged on Terminal TT56:

  Good Evening

  >LO
  Have a Good Evening
  06-JAN-82 20:02 TT56: logged off QUASAR
7 >HELLO EDSERV
  Password:

  RSX-11M BL32   [2,54] System     QUASAR
  06-JAN-82 20:02 Logged on Terminal TT56:

  Good Evening

  6-JAN-81        Kogan

        QUASAR will be down FRIDAY (JAN/8/82) from 11:30 to 13:00.
        Please let me know if you have any objections.

  30-DEC-81       Siler

        The DCL that's now in [2,54] (and installed by STARTUP) is built so
        that it passes unrecognized commands to MCR.  If your terminal is set
        to DCL, you can use whichever form of command is easier, MCR or DCL.

  MCR>BYE
  Have a Good Evening
  06-JAN-82 20:03 TT56: logged off QUASAR
```

Example 2-1 Logging on the System (Sheet 1 of 2)

```
8  >LOGIN
   Account or name: EDSERV
   Password: ■■■■■■■■

   RSX-11M BL32   [2,54] System     QUASAR
   06-JAN-82 20:04 Logged on Terminal TT56:

   Good Evening

   6-JAN-81        Kogan

           QUASAR will be down FRIDAY (JAN/8/82) from 11:30 to 13:00.
           Please let me know if you have any objections.

   30-DEC-81       Siler

           The DCL that's now in [2,54] (and installed by STARTUP) is built so
           that it passes unrecognized commands to MCR.  If your terminal is set
           to DCL, you can use whichever form of command is easier, MCR or DCL.

   >BYE
   Have a Good Evening
   06-JAN-82 20:04 TT56: logged off QUASAR
   >
```

Example 2-1 Logging on the System (Sheet 2 of 2)

Notes on Example 2-2

Example 2-2 shows a sample terminal session. The following comments are keyed to the example.

1. User logs on the system with his last name and password. He notes the system name and his terminal number. As this is the first time he logged in for the day he reads the system messages for important information.

2. User issues the HELP command to see what information is available to help him learn the system.

3. User decides he needs help using the Editor in DCL so he types the HELP EDIT command.

4. User issues the SHOW USERS command to see who is logged on.

5. User checks the time of day by issuing the SHOW DAYTIME command.

6. User checks to see if there are any active tasks at his terminal.

7. User then logs off the system.

```
❶ >LOGIN EDSERV
Password:

RSX-11M BL32   [2,54] System      QUASAR
06-JAN-82 20:04 Logged on Terminal TT56:

Good Evening

6-JAN-81        Kogan

        QUASAR will be down FRIDAY (JAN/8/82) from 11:30 to 13:00.
        Please let me know if you have any objections.

30-DEC-81       Siler

        The DCL that's now in [2,54] (and installed by STARTUP) is built so
        that it passes unrecognized commands to MCR.  If your terminal is set
        to DCL, you can use whichever form of command is easier, MCR or DCL.

❷ >HELP

        Help is available for many RSX-11M commands and utilities.

        For help in logging in to the system, type HELP HELLO or HELP
        LOGIN.  You'll need an user-ID and password to log in.

        RSX-11M systems have two major command languages or CLIs.
        These are MCR and DCL.  Once you log in, your terminal is set
        to either MCR or DCL.  All terminals are set to MCR prior to
        logging in.

        The general form of the HELP command is

                >HELP[/cli] topic [subtopic[s]]

                >HELP commandname [switch]

        Once you are logged in, you need not include the name of the CLI
        to which your terminal is set.  For information on what further help
        is available, type HELP[/MCR] LIST or HELP/DCL.  For a listing
        of help available on other topics, type HELP[/MCR] MORE or
        HELP/DCL MORE.  You need not log in to get help.

❸ >HELP EDIT
EDIT[/qualifier] [command line]
     /EDI
     /EDT[/qualifier[s]
     /KED
     /K52
     /MAKE
     /MUNG
     /SLP[/qualifier[s]
     /SOS
     /TECO
     /USING:usereditor

The EDIT command invokes an editor.  Only EDI, EDT, and SLP are
supported on RSX-11M/M-PLUS systems.  KED and K52 are included with
FMS-11 but are not bundled.  Abbreviation: E

For EDI, and the unsupported editors, the optional command line is a command
line to the editor which is passed to it without any checking whatsoever.
```

Example 2-2 Sample Terminal Session (Sheet 1 of 2)

```
See HELP EDIT EDT and HELP EDIT SLP for information on qualifiers
available when invoking EDT, the DEC Standard Editor, and SLP, the
Source Language Input Program.

(4) >SHOW USERS
HT0: [7,30]
TT6: [7,374]
TT11: [7,32]
TT12: [7,302]
TT13: [7,42]
TT15: [7,113]
TT16: [7,103]
TT20: [7,26]
TT21: [7,372]
TT23: [301,333]
TT24: [16,34]
TT27: [7,361]
TT47: [7,57]
TT52: [304,6]
TT55: [7,32]
TT56: [305,303]
(5) >SHOW DAYTIME
11:12:29 21-OCT-81
(6) >SHOW TASKS
MCR...
SHOT56
(7) >LO
Have a Good Morning
21-OCT-81 11:12 TT56: logged off QUASAR
```

Example 2-2 Sample Terminal Session (Sheet 2 of 2)

LOGGING OFF THE SYSTEM

At the close of your working day or whenever you leave your terminal for any length of time, it is suggested that you log off the system. Logging off the system ends the terminal session and closes off access to your files, system files and privileges. You end the session by typing the BYE command, or if you are using DCL as your CLI you type LOGOUT. The operating system will type a greeting and then give you the date and time that you logged off the system. If you are running RSX-11M-PLUS, the system will also include accounting information for your terminal session. Example 2-3 shows the two methods of logging off the system.

```
>
>LOGIN EDSERV
Password:

RSX-11M BL32   [2,54] System     QUASAR
06-JAN-82 21:07 Logged on Terminal TT56:

Good Evening

6-JAN-81        Kogan

        QUASAR will be down FRIDAY (JAN/8/82) from 11:30 to 13:00.
        Please let me know if you have any objections.

30-DEC-81       Siler

        The DCL that's now in [2,54] (and installed by STARTUP) is built so
        that it passes unrecognized commands to MCR.  If your terminal is set
        to DCL, you can use whichever form of command is easier, MCR or DCL.

>LO
Have a Good Evening
06-JAN-82 21:07 TT56: logged off QUASAR
>
>
>HELLO EDSERV
Password:

RSX-11M BL32   [2,54] System     QUASAR
06-JAN-82 21:08 Logged on Terminal TT56:

Good Evening

6-JAN-81        Kogan

        QUASAR will be down FRIDAY (JAN/8/82) from 11:30 to 13:00.
        Please let me know if you have any objections.

30-DEC-81       Siler

        The DCL that's now in [2,54] (and installed by STARTUP) is built so
        that it passes unrecognized commands to MCR.  If your terminal is set
        to DCL, you can use whichever form of command is easier, MCR or DCL.

>BYE
Have a Good Evening
06-JAN-82 21:08 TT56: logged off QUASAR
>
```

Example 2-3 Logging Off the System

USING THE TERMINAL EFFECTIVELY

Terminals

Your terminal is your vehicle of communication with the operating system. Its capabilities extend beyond the function of responding to your keystrokes by sending the applicable signal to the operating system. Learning more about this device and its capabilities will make your sessions at the terminal more productive.

Terminals are classified as either video (more commonly called Cathode Ray Tubes (CRTs) or hard-copy. Both have advantages and disadvantages. While CRTs generally transmit and receive at a faster rate, they do not provide a permanent record of a terminal session. Hard-copy terminals do provide a permanent record, but generally transmit and receive at slower rates. They also require changing the paper occasionally, so a paper supply must be handy. Hard-copy terminals are also useful for those just learning the operating system. You may want to start out using a hard-copy terminal so that if you work yourself into a corner you will have documentation on how you got there. Your course administrator will be able to show you where you went wrong.

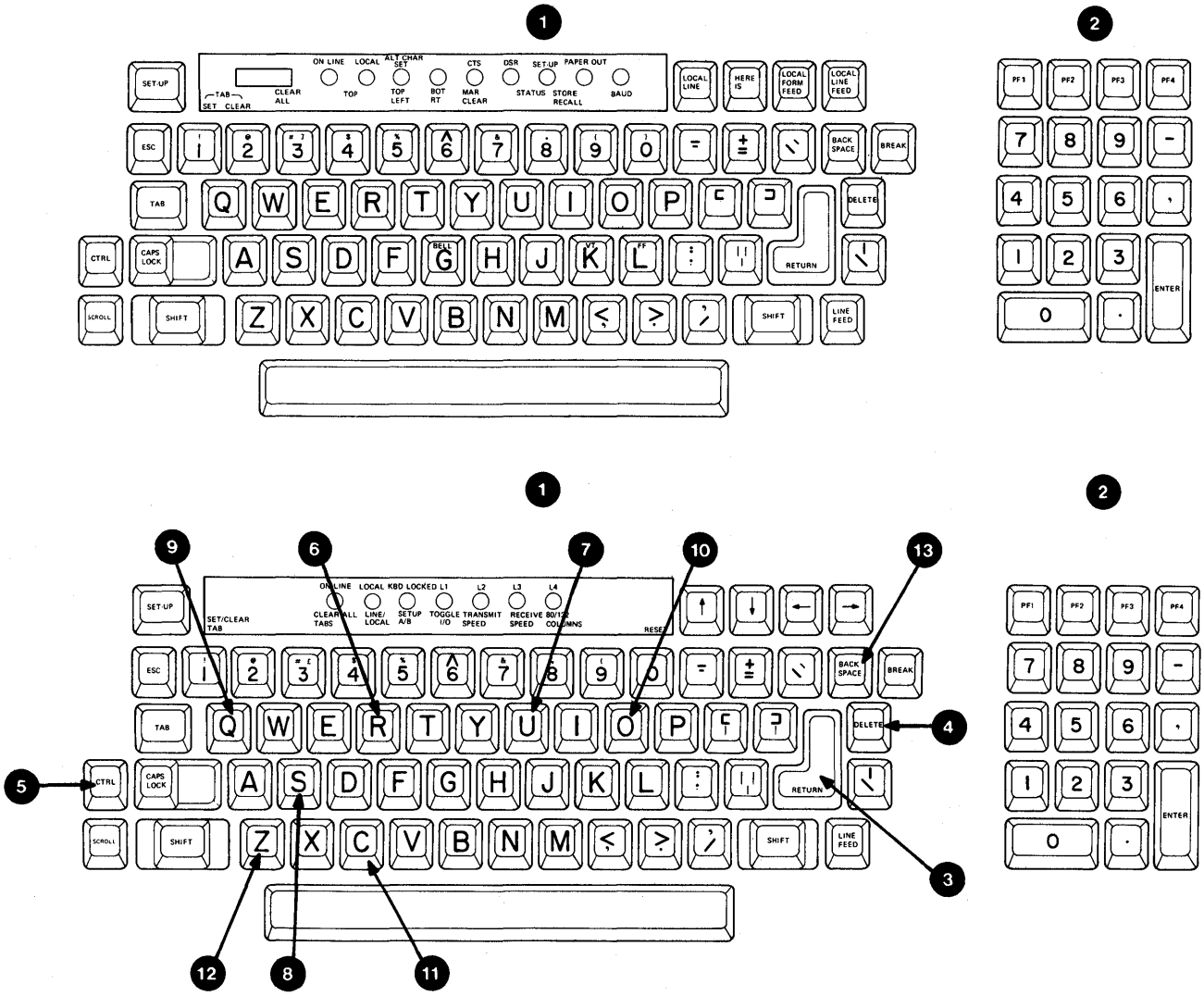
Your terminal is made up of two independent devices:

- an input device (the keyboard) and
- an output device (a display - either a screen or a piece of paper)

When you press a key on the keyboard, the terminal sends the applicable signal down the terminal line to the operating system. The operating system in return lets you know that it received that character by sending it back where it is drawn on the screen (or typed on the paper). This process is called "echoing." You may have noticed during the log in procedure, that when you typed in your password, the characters did not echo at your terminal. This is to ensure the privacy of your password.

The arrangement of the keys on the keyboard may vary from terminal to terminal. Figure 2-1 shows an example of two keyboards. In addition to the keys found on a standard typewriter keyboard, a computer terminal has other keys that perform special functions such as starting and stopping output to the terminal, erasing typographical errors, retyping a command line, etc. It may also have an auxiliary keypad containing the numerical keys Ø through 9, a decimal point, an enter key and arrow keys. Some system programs (like the editor) use this keypad for performing special functions.

Table 2-1 shows the special keys and their function. Some of them require pressing two keys simultaneously to achieve the action; these are indicated in the table. When you are correcting errors within a command line be sure to use the <delete> key instead of the <backspace> key. Although the <backspace> key appears to perform correctly, it is not interpreted that way by the operating system. Table 2-2 shows how the delete key echoes on both a video and a hard-copy terminal. In the case of a video terminal, every time the delete key is pressed, the operating system (in particular, the terminal driver) transmits a command to back up the cursor one position, and then transmits a space. In the case of a hard-copy terminal, the operating system transmits a backslash and then the last previous character is typed.



TK-7669

Figure 2-1 LA120 and VT100 Keyboard Layouts

Table 2-1 Special Terminal Keys
(Two Key Combinations)

User Wants To	Key to Use*	Name of Key
Terminate command or line of data you have typed	<RET> (3)	Carriage Return
Delete last character you typed	<DEL> (4)	Delete
Use with another key to obtain a control key combination	<CTRL> (5)	Control
Retype line you have been typing	<CTRL/R> (5) (6)	Control-R
Delete entire line you have been typing	<CTRL/U> (5) (7)	Control-U
Stop output coming to a terminal	<CTRL/S> (5) (8)	(X-ON) Control-S
Continue output to the terminal	<CTRL/Q> (5) (9)	(X-OFF) Control-Q
Ask system not to print remainder of output to the terminal	<CTRL/O> (5) (10)	Control-O
Interrupt program that is executing; then type a command line	<CTRL/C> (5) (11)	Control-C
Exit from a utility	<CTRL/Z> (5) (12)	Control-Z

*Reversed-out numbers refer to Figure 2-1

NOTE

The BACKSPACE key ((13) on Figure 2-1) is not to be used to delete characters in RSX-11M/M-PLUS operating systems.

Table 2-2 Echoing of Delete Key on CRTS/Hardcopy

If you type:	What appears on video screen:	What appears on hard-copy printout:
SHPW	SHPW	SHPW
<DEL><DEL>	SH	SHPW\WP
OW	SHOW	SHPW\WP\OW

LEARNING ACTIVITIES

1. READ the following sections in the RSX-11M/M-PLUS Command Language Manual:

   - 3.1, Logging In and Logging Out
   - 3.2, The Keyboard
   - 3.3, The Keypad
   - 3.4, The CTRL key and Control Characters

2. DO the following at your terminal:

   a. Example 2-1 shows four different methods of logging in. Try each method using your UIC and last name, and decide which method you prefer. Ensure that DCL is active at your terminal.

   b. Table 2-1 lists special keys that help control the action of your terminal. Practice using each of the keys by issuing the following command when you have the system prompt:

      ```
      >TYPE LB:[1,2]DCL.HLP
      ```

      This file is a text file used by the HELP command to display the commands for which it has further information. Concentrate on how the keys control your terminal input and output.

CONVERSING WITH THE OPERATING SYSTEM

Command Line Interpreter (CLI)

CLIs are the languages used for communication between you and the operating system. On RSX-11M/M-PLUS systems, the user has the capability of talking to the operating system in more than one language. Two such languages are supplied by DIGITAL. They are the Digital Command Language (DCL) and Monitor Console Routine (MCR). You may talk to the operating system in only one language at a time, but can switch easily from one language to another. MCR is always present on your system. DCL most likely will be present, and your site may include a CLI not provided by DIGITAL. You may want to talk to your system manager regarding the availablity of CLIs on your system. We will discuss both DCL and MCR in this module and will use DCL throughout the course in listings and examples.

CLI Prompts

When the operating system is ready to accept commands from your terminal, it will display its default system prompt, ">", at the left-most position of your display. To determine which CLI is active at your terminal and to get the explicit CLI prompt, you should type <CTRL/C>. Do this by holding down the control key and then typing the letter "C". The operating system will then respond with the appropriate explicit prompt, either the MCR prompt:

```
MCR>
```

or the DCL prompt:

```
DCL>
```

You can perform this operation whenever you are unsure which CLI is active.

Default CLI

The accounts file contains a default CLI for each user which is set up by the system manager when your account is created. When you log on the system, the accounts file is checked to see which CLI is your default CLI. Once the log in procedure is complete, the operating system starts your default CLI running at your terminal, and displays the implicit operating system prompt at your terminal. You are then ready to type system commands in the format required by the active CLI.

Changing from One CLI to Another

When you wish to change from one CLI to another you must issue the appropriate command. To change from DCL to MCR, type the following command:

```
>SET TERMINAL/MCR
```

To change from MCR to DCL, type the following command:

```
>SET /DCL=TI:
```

USING THE DIGITAL COMMAND LANGUAGE (DCL)

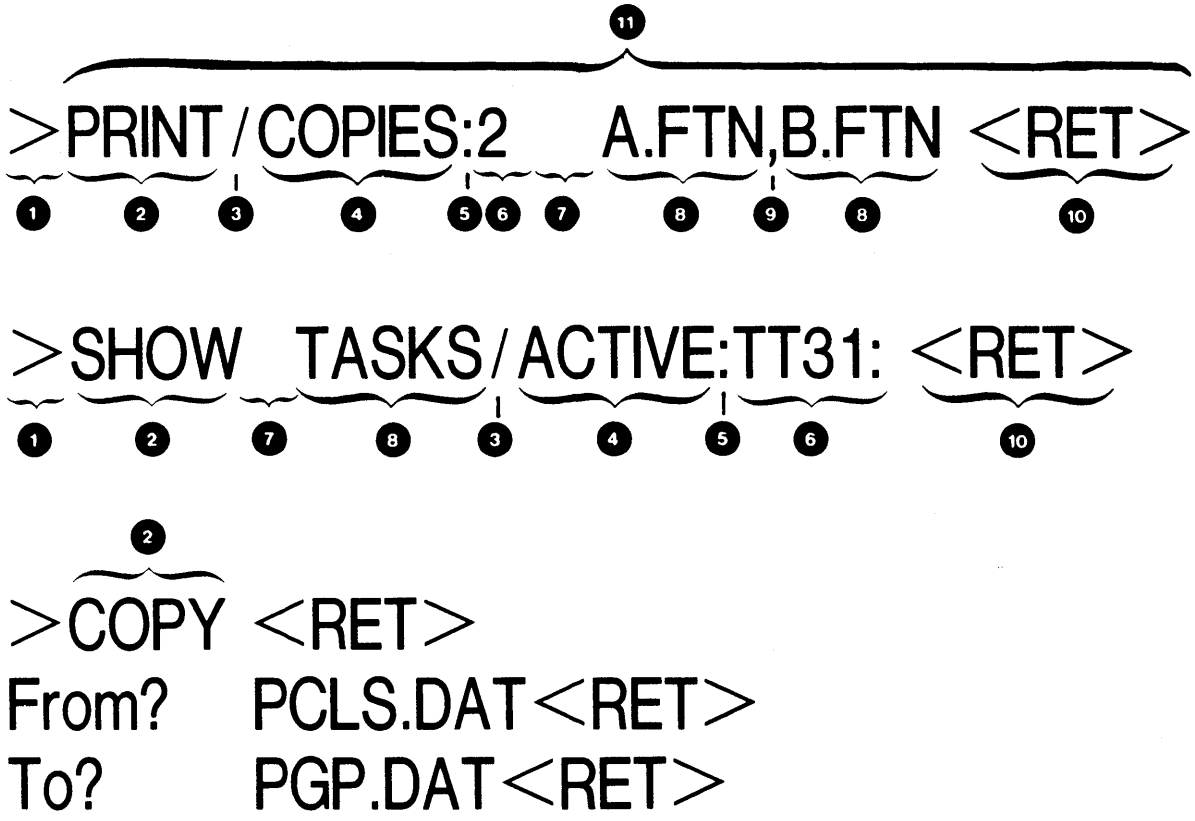
The Digital Command Language (DCL) is the standard command language found on DIGITAL-supplied operating systems. Becoming familiar with this language will make your migration to other operating systems easier. The commands are more English-like, so most users prefer to use it.

Features of DCL

- Standard command language used on most DIGITAL operating systems.
- English-like language.
- Prompts for unsupplied information.
- Command line continuations: If the command requires more characters than will fit on a line, end the first line with a hyphen and continue the command on the next line.
- Commands and keywords can be abbreviated (Four characters always sufficient since DCL ensures that first four characters are always unique).
- Commands can be placed in a file and executed from that file.
- HELP facility available for all DCL commands.
- Can specify comments within the command line.

The command line structure is shown below.

DCL Command Format



1. Operating system prompt - Whenever it is present, commands may be entered
2. DCL command - Can be abbreviated to at least 4 characters
3. Command qualifier delimiter - Always a "/"
4. Command qualifier - A keyword that further defines how to interpret the command
5. Argument delimiter - Always a ":"
6. Argument value
7. Delimiter (blank or TAB) - Must be included
8. Parameter
9. List element delimiter
10. Command line terminator
11. Command line

Table 2-3 lists some DCL commands. Example 2-4 shows a sample session using these commands. Refer to the RSX-11M/M-PLUS Command Language Manual for additional information on each command.

Table 2-3 Partial List of DCL Commands

User Wants To	DCL Command To Use
Check the time and date	SHOW DAYTIME
Determine who is logged on the system	SHOW USERS
Check the print or batch job status	SHOW QUEUE
Find his session I.D., terminal number, accounting information	SHOW ACCOUNTING/ INFORMATION (M-PLUS only)
Obtain information on terminal settings	SHOW TERMINAL
Determine what hardware is available on the system	SHOW DEVICES
Determine what partitions are available on the system	SHOW PARTITIONS
Protect a file from deletion	SET PROTECTION
Determine what tasks are installed	SHOW TASKS/INSTALLED
Determine what tasks are active on the system	SHOW TASKS/ACTIVE

```
>SHOW DAYTIME
10:50:56 22-OCT-81
>SHOW USERS
HT1: [7,22]
TT6: [7,374]
TT11: [7,32]
TT12: [7,302]
TT13: [7,42]
TT14: [7,110]
TT15: [7,113]
TT16: [301,303]
TT17: [7,366]
TT21: [7,372]
TT22: [7,116]
TT23: [301,333]
TT24: [52,30]
TT25: [301,44]
TT27: [7,361]
TT51: [100,24]
TT53: [7,337]
TT55: [2,201]
TT56: [305,303]
>SHOW QUEUE
** PRINT QUEUES **
PRINT  => LPO
  [7,113]   DCLRES    ENTRY:433              ACTIVE ON LPO
       >    1 DR0:[7,113]DCLRES.MAP;41
LPO    => LPO
>SHOW TERMINAL
TT56:   [305,303]       [305,303]
        CLI    = DCL    BUF    = 132.   HFILL = 0       SPEED=(9600:9600)
        LINES = 66.     TERM   = LA120  OWNER = none    BRO     NOABAUD
        LOWER   NOPRIV  NOHOLD  NOSLAVE NOESC   NOCRT   FORM    NOREMOTE
        ECHO    NOVFILL NOHHT   NOFDX   WRAP    NORPA   NOEBC   TYPEAHEAD

>SHOW DEV MM:
MM0: TT23: - Private Mounted Loaded
MM1: Loaded
```

Example 2-4 DCL Commands (Sheet 1 of 6)

```
>SHOW DEVICES
HT0: Offline Loaded
HT1: [7,22] - Logged in  Loaded
HT2: Offline Loaded
HT3: Offline Loaded
LP0: Loaded
DB0: Loaded Type=RP05
DB1: Public Mounted Loaded Type=RP06
DB2: Public Mounted Loaded Type=RP06
DD0: Loaded
DD1: Loaded
DK0: Loaded
DK1: Loaded
DK2: Loaded
DK3: Loaded
DL0: Loaded Type=RL01
DL1: Loaded Type=RL02
DL2: Loaded Type=RL01
DM0: TT51: - Private Mounted Loaded Type=RK06
DM1: Loaded Type=RK06
DM2: Loaded Type=RK07
DR0: Public Mounted Loaded Type=RP07
DR1: Loaded Type=RM03
DS0:    EM0: Type=RS04
DS1:    EM0: Type=RS04
DT0: Loaded
DT1: Loaded
DX0: Loaded
DX1: Loaded
EM0: Public Mounted Loaded Type=ML11 Size=4096. blks
MM0: TT23: - Private Mounted Loaded
MM1: Loaded
CQ0: Loaded
NS0: Public Mounted Loaded
TT0: Loaded
TT1: Loaded
TT2: Loaded
TT3: Loaded
TT4: Loaded
TT5: Loaded
NL0:
TI0:
CL0:    LP0:
LB0:    DR0:
SY0:    DR0:
```

Example 2-4 DCL Commands (Sheet 2 of 6)

```
>SHOW PARTITIONS
CEXPAR 114734 00115000 00003000 MAIN COM
TTPAR  114670 00120000 00040000 MAIN TASK
EXCOM1 114624 00160000 00014700 MAIN COM
EXCOM2 114560 00174700 00006200 MAIN COM
SYSPAR 114514 00203100 00012000 MAIN TASK
TKNPAR 114450 00215100 00010000 MAIN TASK
DRVPAR 114404 00225100 00030600 MAIN SYS
       114340 00225100 00002100 SUB  DRIVER -DB:
       114240 00227200 00001200 SUB  DRIVER -DK:
       114140 00230400 00003100 SUB  DRIVER -DM:
       114040 00233500 00003000 SUB  DRIVER -DR:
       113740 00236500 00001000 SUB  DRIVER -EM:
       113640 00237600 00001100 SUB  DRIVER -DT:
       113540 00240700 00001400 SUB  DRIVER -DX:
       113440 00242300 00002500 SUB  DRIVER -DL:
       113340 00245000 00002600 SUB  DRIVER -DD:
       113240 00247600 00001300 SUB  DRIVER -LP:
       113104 00251100 00004300 SUB  DRIVER -MM:
       112450 00255400 00000300 SUB  DRIVER -CO:
LDRPAR 112404 00255700 00002500 MAIN TASK
BASIC2 112340 00260400 00040000 MAIN COM
FCSRES 112274 00320400 00040000 MAIN COM
TSTPAR 112230 00360400 00100000 MAIN TASK
GEN    112164 00460400 03317400 MAIN SYS
       045174 00460400 00006300 SUB  (PMT...)
       061140 00466700 00014200 SUB  (EVP...)
       046100 00507200 00017000 SUB  (COT...)
       046160 00526500 00044000 SUB  (WKFFCP)
       046034 00572500 00044000 SUB  (SYOFCP)
       100544 00643600 00013600 SUB  (MTAACP)
       075740 00666000 00023300 SUB  (RMHACP)
       045430 00723600 00044000 SUB  (DB1FCP)
       046370 00767600 00024000 SUB  (DB2FCP)
       046270 01013600 00032200 SUB  (QMG...)
       044770 01046000 00014100 SUB  (LP0   )
       047560 01062100 00027000 SUB  (RMDEMO)
       060534 01111100 00044000 SUB  (F11ACP)
       062220 01232000 00025600 SUB  (NETACP)
       070214 01257600 00020300 SUB  (SHOT56)
       070750 01362200 00132000 SUB  (...EDT)
       045240 01516200 00132300 SUB  (...PIP)
       071730 01650500 00107400 SUB  (HT1   )
       075014 01760100 00055700 SUB  (EX1   )
       074554 02112100 00060000 SUB  (...AT.)
       072450 02172100 00132000 SUB  (EDTT16)
       071040 02324400 00055700 SUB  (EX2   )
       100610 02402300 00132300 SUB  (MAL.0 )
       100500 02707700 00175700 SUB  (...TKB)
NT.PCL 062634 03671600 00003500 SUB  DYNAMIC
NT.NSP 053244 03675300 00016000 SUB  DYNAMIC
NT.AUX 062264 03713300 00001100 SUB  DYNAMIC
POOL.. 061074 03714400 00061700 SUB  DYNAMIC
       045034 03776300 00001500 SUB  DRIVER -HT:
```

Example 2-4 DCL Commands (Sheet 3 of 6)

```
>SHOW TASKS INSTALLED
LDR... 12.19  LDRPAR 248. 00002500 LB0:-01240437 FIXED
TKTN   04.7   TKNPAR 248. 00010000 LB0:-01242244
RMDEMO V1.05  GEN    225. 00027000 LB0:-01262570
MTAACP 0013   GEN    200. 00013600 EM0:-00000325
F11MSG V0012  GEN    200. 00005500 LB0:-01251052
NETACP V02.00 GEN    200. 00025600 SY0:-01266761
EVP... V01.00 GEN    199. 00013100 SY0:-01266561
MCR... 2.02   SYSPAR 160. 00012000 LB0:-01244424
...MCR 1.1    GEN    160. 00020300 LB0:-01244760
SHOT56 1.1    GEN    160. 00020300 LB0:-01244760
...MOU 2503   GEN    160. 00037600 LB0:-00117642
...DCL 00     GEN    160. 00035600 SY0:-01132456
...HEL 01.0   GEN    150. 00037600 LB0:-01250645
...CA. 4.00   GEN    150. 00005100 LB0:-01251064
...CVT 4.02   GEN    150. 00052200 SY0:-01007774
...DLG 4.02   GEN    150. 00052200 SY0:-01007774
NVP... V01.00 GEN    150. 00012700 SY0:-01265031
...BYE 04.0   GEN    150. 00012500 SY0:-01264767
F11ACP M0401  GEN    149. 00044000 LB0:-01247227
WKFFCP M0401  GEN    149. 00044000 LB0:-01262523
SYOFCP M0401  GEN    149. 00044000 EM0:-00000260
DB1FCP M0401  GEN    149. 00044000 SY0:-01262313
DB2FCP M0401  GEN    149. 00024000 SY0:-01262404
PMT... 01.52  GEN    148. 00006300 LB0:-01246016
ERRLOG V1.04  GEN    148. 00040000 LB0:-01251006
COT... 01.01  GEN    145. 00017000 LB0:-01250736
...DMO 23.25  GEN    140. 00014600 LB0:-01245770
...INI 22.02  GEN    140. 00034600 LB0:-01246445
...UFD V0412  GEN    140. 00005700 LB0:-01246615
PMD... 07.2   GEN    130. 00017600 LB0:-01250210
SHF... 5.08   SYSPAR 105. 00012000 LB0:-01250203
...INS 06     GEN    100. 00034400 LB0:-01243744
...SEN 22JUL  GEN    100. 00104400 LB0:-00662645
RMHACP V02.00 GEN    100. 00023300 SY0:-01265073
MAL$$$ 11RX   GEN    100. 00132300 SY0:-00663342
PMR$$$ V01.05 GEN    100. 00034300 SY0:-00660507
QMG... 1.8    GEN     70. 00032200 LB0:-01252004
LP0    1.9    GEN     70. 00014100 LB0:-01252136
PRT... 1.2    GEN     70. 00001100 LB0:-01251443
...BRU 1.03   GEN     70. 00173500 LB0:-01256414
...KKK        GEN     70. 00163400 DK2:-00010412
...TEC V36    GEN     65. 00177100 EM0:-00001215
...EDI M11.04 GEN     65. 00123600 EM0:-00001371
...SOS 07.00  GEN     65. 00073300 EM0:-00001547
...EDT 02.00  GEN     65. 00132000 LB0:-01254476
...MAI 01SEP  GEN     65. 00123700 SY0:-00555723
RMT... V02.01 GEN     65. 00027300 SY0:-01266505
...RVT X02.19 GEN     65. 00032500 SY0:-00663556
EDTT16 02.00  GEN     65. 00132000 LB0:-01254476
...AT. 02     GEN     64. 00060000 LB0:-01246107
NTL... V02.00 GEN     60. 00036500 SY0:-01265603
NTINIT V02.00 GEN     55. 00005000 SY0:-01265347
...BRO V04    GEN     50. 00031000 LB0:-01250552
LPINIT 01.03  GEN     50. 00016200 LB0:-01250260
```

Example 2-4 DCL Commands (Sheet 4 of 6)

```
>SHOW TASKS /ACTIVE
MCR...
SHOT56
>SHOW TASKS /ACTIVE /ALL
LDR...
RMDEMO
NETACP
EVP...
MCR...
SHOT56
F11ACP
DB1FCP
PMT...
COT...
QMG...
LPO
...EDT
...AT.
```

Example 2-4 DCL Commands (Sheet 5 of 6)

```
>SHOW TASKS ACTIVE
MCR...
SHOT56
>SHOW TASKS ACTIVE ALL
LDR...
RMDEMO
MTAACP
NETACP
EVF...
MCR...
SHOT56
F11ACP
WKFFCP
SY0FCP
DB1FCP
DB2FCP
PMT...
COT...
RMHACP
QMG...
LP0
...TEC
...EDT
EDTT16
...AT.
...PIP
...TKB
EX1
EX2
EXST
HT1
>
```

Example 2-4 DCL Commands (Sheet 6 of 6)

THE HELP COMMAND

One of the most useful features of this operating system is the help that it provides on-line for the system commands. It is not necessary to search through manuals to find the correct format of a command or the qualifier that you need. By typing the HELP command in response to the system prompt, you will get a list of items for which the system has more information. HELP is the only command that you can type before logging onto the system.

Examples of the HELP command are shown below.

HELP Command Format

```
>HELP<RET>

>HELP/MCR HELLO<RET>

>HELP SET<RET>

>HELP SET TERMINAL<RET>
```

Notes on Example 2-5

The following comments are keyed to the example.

1. In response to the HELP command, the operating system displays the list of active CLI commands at your terminal for which it has information.

2. To obtain information on any of those listed, type HELP and the command you wish the information for. Notice that the text gives you the syntax of the command and an explanation of what the command will do.

3. To get help on a command of another CLI, use the /"CLI" qualifier shown here. Hello is an MCR command and is not listed in the HELP text shown in note 1 above.

4. In this example, you will notice that the text includes the command qualifiers that can be used with the command.

5. This and the following two examples show how to obtain further information on a command. Notice that the SET command has many attributes which it can change. To get further information, for example, on the TERMINAL attribute, you would concatenate this attribute to your original HELP command.

6. This example shows how the procedure described above is accomplished.

7. The HELP command is further qualified to obtain more information on setting the terminal, in particular the commonly used attributes.

```
1 >HELP

  Help is available for the following DCL commands:

ABORT          ALLOCATE       APPEND         ASSIGN         BACKUP
BASIC          BROADCAST      CANCEL         COBOL          CONTINUE
CONVERT        COPY           CREATE         DEALLOCATE     DEASSIGN
DEBUG          DELETE         DIFFERENCES    DIRECTORY      DISMOUNT
EDIT           FIX            FORTRAN        HELP           HOLD
INITIALIZE     INSTALL        LIBRARY        LINK           LOGIN
LOGOUT         MACRO          MCR            MOUNT          PRINT
PURGE          RELEASE        REMOVE         RENAME         REQUEST
RUN            SET            SHOW           SORT           START
STOP           TYPE           UNFIX          UNLOCK

  For information on a command, type HELP commandname.  Additional help on
  a particular qualifier is often available by typing

               HELP commandname qualifier.

  You can also get help on DCL commands by typing a ? in response to a prompt.

  For information on utilities, system tasks, and other system information,
 type HELP MORE.  For the short forms of some commands, type HELP BRIEF.
 For help on logging in type HELP LOGIN.
2 >HELP LOGIN
 LOGIN userid

 LOGIN allows you access to the system if you have a valid acount.

 'userid' is either your last name, or a UIC with the two numbers separated
 by either a comma (,) or a slash (/). The square brackets ([]) are optional
 for this command. After entering your userid, you will be prompted
 for your password.  The password will not echo on the your
 terminal. You can enter the password on the same line as the LOGIN
 command by preceding it with a slash, but your password will be
 echoed.

 If the userid and password are valid, you will be logged on the system.
 The login message file, LB:[1,2]LOGIN.TXT, will be printed on your
 terminal, unless you entered 'userid' as a UIC separated by a slash, in
 which case only the most recent messages will be printed. If you have
 a login command file (LOGIN.CMD) in your directory, it will then be
 executed.

 LOGIN and HELLO are identical, but LOGIN does not work on systems
 without DCL.
```

Example 2-5 HELP Command (Sheet 1 of 3)

```
(3) >HELP/MCR HELLO

To log onto this system, you must know an account or last name and
the appropriate password.  Log on by saying HELLO.  The HELLO command
has the following general forms:

        HELLO

You will be prompted for your name or account and password.

        HELLO name/password

Where name is your last name, and password is your account password.
If the slash and password are left off (they are optional), you will
be prompted for your password.

        HELLO [grp,mem]/password
        HELLO [grp/mem]/password

Where grp and mem are the group and member numbers of your account.
The square brackets and password are optional.  If a comma separates
grp and mem, system messages (the LB:[1,2]LOGIN.TXT file) will be dis-
played on your terminal immediately after you have logged in.  If a slash
is the separator, the LOGIN.TXT file will be displayed on your
terminal until a % is found in column 1 of the file.

(4) >HELP COPY

COPY[/qualifier[s]]  infile[s] outfile

        Qualifiers

        /BLOCKSIZE:n
        /[NO]CONTIGUOUS
        /OWN
        /REPLACE
        /DATE:dd-mmm-yy
        /SINCE:dd-mmm-yy
        /SPAN_BLOCKS
        /THROUGHL:dd-mmm-yy
        /SINCE:dd-mmm-yy/THROUGH:dd-mmm-yy
        /TODAY
        /EXCLUDE:filespec

The COPY command copies one or more input files to an output file.
COPY does not alter the organization of the file.  See HELP CONVERT
for information on changing the organization of the file.

Abbreviation: C
(5) >HELP SET
SET things

The SET command can be used to set something. The following things
can be set with this command:

[DAY]TIME       DEBUG           DEFAULT         DEVICE          GROUPFLAGS
LIBRARY         [NO]PARTITION   PRIORITY        PROTECTION      QUEUE
SYSTEM          TERMINAL
```

Example 2-5 HELP Command (Sheet 2 of 3)

```
6 >HELP SET TERMINAL

  SET TERMINAL[:ttnn:]/attribute[s]

  The SET TERMINAL command sets various attributes of your terminal. A
  privileged user can set attributes of any other terminal.

  The RSX-11M/M-PLUS Command Language Reference Manual divides the attri-
  butes into the 3 categories of common use, terminal setup, and task
  setup.
          See HELP SET TERMINAL COMMON_USE
              HELP SET TERMINAL TERMINAL_SETUP
              HELP SET TERMINAL TASK_SETUP

              HELP SET TERMINAL QUALIFIERS
              HELP SET TERMINAL qualifiername

7 >HELP SET TERMINAL COMMON_USE

  SET TERMINAL[:ttnn:]/attribute[s]

  The following qualifiers set terminal characteristics that are regularly
  needed by the average terminal user. For more information on a parti-
  cular characteristic, type HELP SET TERMINAL characteristic.

        /[NO]BROADCAST
        /DCL
        /[NO]HOLD_SCREEN
        /LOWERCASE
        /MCR
        /[NO]PRIVILEGED
        /[NO]SERIAL
        /SPEED:(t,r)
        /UPPERCASE
        /WIDTH:n
>
>
```

Example 2-5 HELP Command (Sheet 3 of 3)

LEARNING ACTIVITIES

1. READ the following sections in the RSX-11M/M-PLUS Command Language Manual:

   - 1.1, RSX-11M/M-PLUS and DCL: Basic Concepts
   - 1.2, The DCL Command Line
   - 1.3, Functional Grouping of DCL Commands
   - 3.5, HELP

2. DO the following at your terminal:

   a. Issue the HELP command to see the list of topics for which there is information

   b. Using the SET command, change your current CLI to MCR

   c. Issue the HELP command to see what MCR topics are available

   d. Using the MCR SET command, change your current CLI back to DCL

FILE SPECIFICATIONS

Files

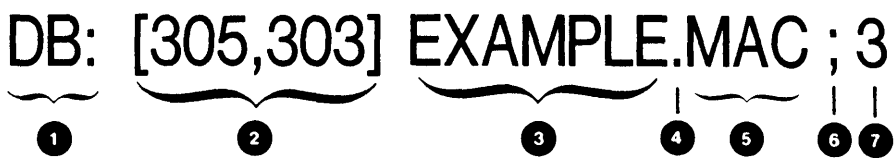
Information stored on the system is contained in an entity called a file. A file is the smallest unit of storage maintained by the operating system. A user may store files on many different devices. To retrieve the files, the operating system must have a means of keeping track of the name of the file, to whom it belongs and where that file is located on a device. There is a special set of software that maintains the file structure called FILES-11. It is the software's responsibility to keep track of files by maintaining directories for each system user. Many of the operating system commands perform functions on a file. To identify the correct file, a standard format for specifying a file is required.

File Specification Command Format

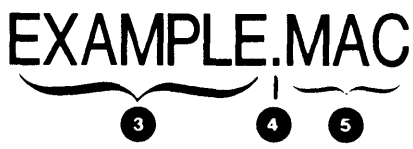
The following is an example of the complete file specification format. Whenever a file specification is required in a command, this is the format in which it is to be supplied. It is not always necessary to supply information in each field. If a field is not supplied, the system will make some assumptions about the entry for that field. The second example below shows a "minimum required" file specification. In some instances this specification can be shortened to just the file name.

The notes below describe each field of the file specification. We will discuss each one in detail. Study the notes and fix in your mind each of the fields of the specification.

A complete file specification



A minimum required file specification



1. Physical, logical or pseudo device name
2. User File Directory (UFD)
3. File name (Ø to 9 characters)
4. File name delimiter
5. File type (Ø to 3 characters)
6. File type delimiter
7. File version number

Devices

Physical Device Names

The first field in the file specification specifies the device where the file is stored.

A computer system consists of various hardware components including:

- Line printers
- Disk drives
- Floppy disk drives
- Magnetic tape drives
- Analog to digital converters
- Paper tape reader/punch

To distinguish a line printer from a floppy disk drive or even from another line printer, the operating system must have a unique name with which it can associate the piece of hardware. As the user, you want to be able to say "I want the file that is stored on the floppy disk drive rather than the one on the magnetic tape drive." And also, "I want this file printed on the second line printer on the system, not the first." To do this, you need a format for specifying devices. When you refer to a specific device on the system, you can use the device's physical device name.

The physical device name has the format:

DDnn:

Device Name Element	Meaning	Example
DD	Two alpha characters depicting a device.	TT
nn:	Two numeric characters depicting the device unit number. The first unit number is Ø. Colon always required.	31:

Figure 2-2 illustrates physical device names. Each device has a name made up of a two character mnemonic representing the type of device it is, and a two digit number representing which unit of that device type it is. Table 2-4 lists the devices and their physical device name. In Figure 2-2, there is a device with the name DB1:. In Table 2-4, DB is the physical device name for RPØ4, RPØ5, and RPØ6 disk drives.

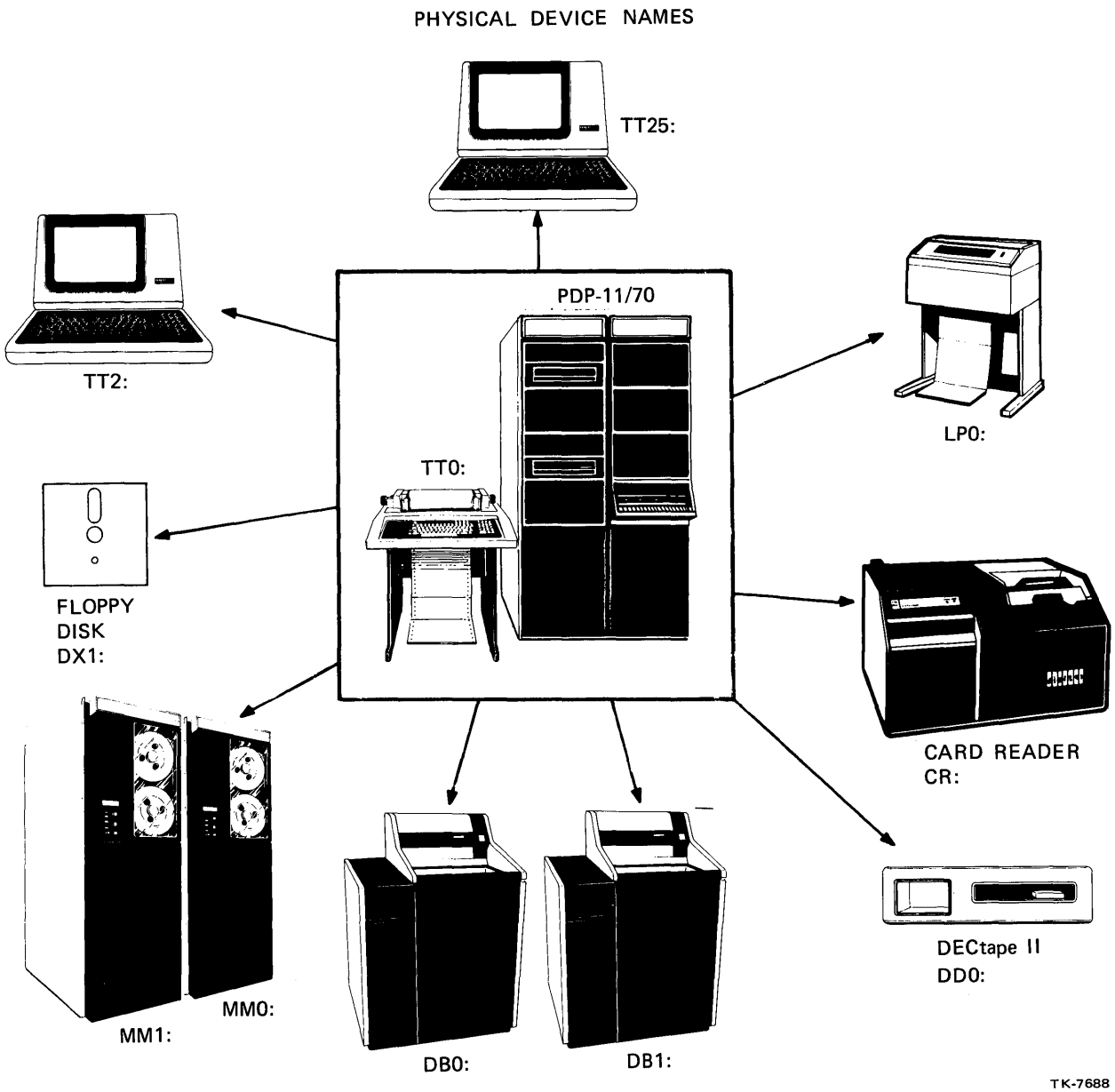


Figure 2-2 Physical Device Names

Table 2-4 RSX-11M/M-PLUS Peripheral Devices

Peripheral Devices	Device-Unit
Analog-to-Digital Converter (AD01-D)	ADnn:
(AFC11)	AFnn:
Laboratory Peripheral System (AR11)	ARnn:
Card Reader (CR11)	CRnn:
Cassette (TA11)	CTnn:
DECtape (TC11)	DTnn:
DECtape II (TU58)	DDnn:
Disks	
RP04/05/06 disk packs (RH11/RH70/RP04/RP05/RP06)	DBNN:
RF disks (RF11)	DFnn:
RK05 cartridge disks (RK11)	DKnn:
RK06/07 cartridge disks (RK611)	DMnn:
RL01 cartridge disks (RL11)	DLnn:
RM02/03 disks (RH11/RM02, RH70/RM03)	DRnn:
RP02/03 disks (RP11)	DPnn:
RS03/04 disks (RH11/RH70/RS03/RS04)	DSnn:
RX01 floppy disk (RX11)	DXnn:
RX02 floppy disk (RX211)	DYnn:
Graphics Display Processor and Scope (VT11/VS60)	GRnn:
Industrial Control System Local and Remote (ICS/ICR-11)	ICnn:
Industrial Control Subsystem (DS11/DRS11)	ISnn:

Table 2-4 RSX-11M/M-PLUS Peripheral Devices (Cont)

Peripheral Devices	Device-Unit
Laboratory Peripheral Accelerator (LPA11-K)	LAnn:
Laboratory Peripheral System (LPS11)	LSnn:
Line Printer (LA11/LS11/LP11/LV11)	LPnn:
Magtapes	
TU45/TU16/TE16/TU77 (RH11/RH70/TM02/TM03)	MMnn:
TS03/TU10/TE10 (TM11/TMA11/TMB11)	MTnn:
TS04 (TS11)	MSnn:
Paper Tape Punch (PC 11)	PPnn:
Paper Tape Reader (PC11/PR11)	PRnn:
Terminal (DL11/DH11/DJ11/DZ11)	TTnn:
Parallel Interprocessor Link (PCL-11B) [RSX-11M-PLUS systems]	
TXnn:/RXnn:	
Parallel Line Interface (DA11-B)	XBnn:
Asynchronous Line Interface (DL11-E)	XLnn:
Interprocessor Link (DMC)	XMnn:
Synchronous Line Interface (DP11)	XPnn:
(DQ11)	XQnn:
(DUP11)	XWnn:
(DU11)	XUnn:
Universal Digital Controller (UDC11)	UDnn:

Logical Device Names

You may also specify a logical device name in the first field of the file specification. A logical device name is a generic name assigned to a physical device by the user. Logical device names provide device independence for programs and system operations because the association between the logical device and a specific physical device can easily be changed to another physical device. A logical device can be used in the device specification, but must first be associated with the actual physical device chosen by the user. The format of the logical device is the same as the physical device. However, the first two digit mnemonic is of your choosing and has the meaning that you associate with it.

DDnn:

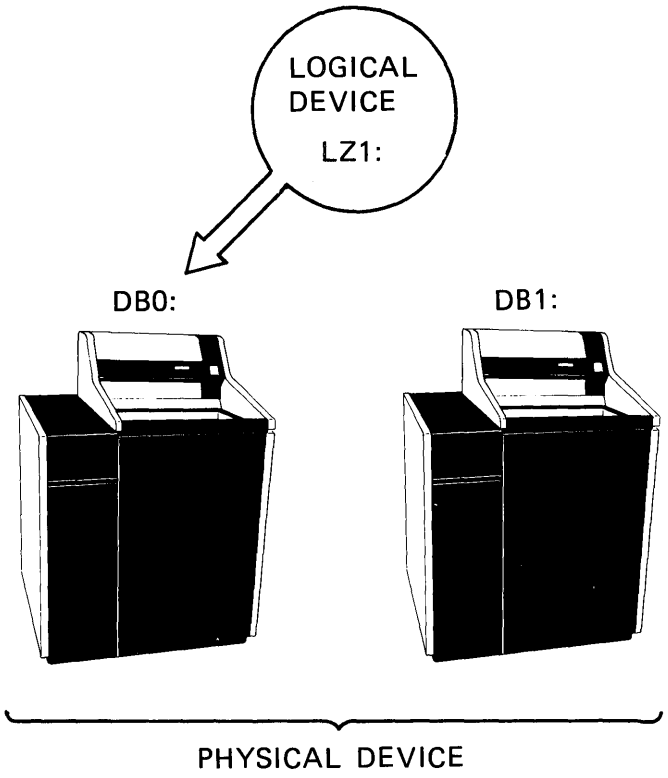
Logical Device Name Element	Meaning	Example
DD	Two alpha characters depicting a device.	LZ
nn:	Two numeric characters depicting a unit number. Colon always required.	Ø1:

Notes on Figure 2-3

Figure 2-3 shows how logical device names can be used. The following comments are keyed to the example.

1. The association between the physical device DBØ: and the logical device LZ1: is made using the ASSIGN command.
2. The Directory command lists the files in the user's work area. Specifying the logical device name of LZ1: gets a directory from the physical device DBØ:
3. The logical device LZ1: is reassigned to the physical device DB1:
4. Performing the same directory command as before, you now get a directory from the physical device DB1:

❶ > ASSIGN DB0: LZ1:



❷ > DIR LZ1:[305,303]
DIRECTORY DB0:[305,303]
16-JULY-81 10:30

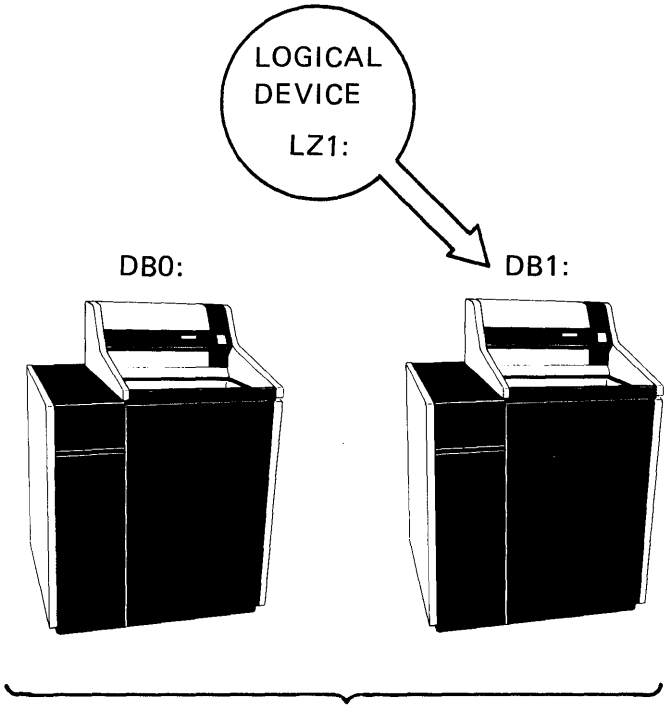
A.FTN;1	2.	10-JULY-81	14:41
B.FTN;1	2.	9-JULY-81	8:33
PROG.FTN;1	4.	9-JULY-81	10:05

TOTAL OF 8./10. BLOCKS IN 3. FILES
>

TK-7681

Figure 2-3 Logical Device Assignments (Sheet 1 of 2)

❸ > ASSIGN DB1: LZ1:



❹ > DIR LZ1:[305,303]
DIRECTORY DB1:[305,303]
16-JULY-81 10:32

VIP.FTN;2	4.	15-JULY-81	15:12
VIP.FTN;3	5.	15-JULY-81	16:42
VIP.TSK;3	6.	16-JULY-81	17:01

TOTAL OF 15./16. BLOCKS IN 3. FILES
>

TK-7689

Figure 2-3 Logical Device Assignments (Sheet 2 of 2)

Pseudo Device Names

You may also specify a pseudo device name in the first field of the file specification. Pseudo device names are a small set of logical device names that have special meaning to the operating system. They refer to devices that most users and tasks often use. Table 2-5 lists the set of pseudo device names. Of particular interest to the general user are the devices TI: and SY:. Wherever your terminal number is part of the required syntax you may substitute the pseudo device of TI:. The operating system then makes the association of TI: to the actual physical device name for you. Anyone may use TI: to represent the physical device name of their terminal; the operating system makes the association to the actual physical device. SY: is used in the same manner. It represents the device on which your UFD is located.

Not everyone using the system has their UFD on the same device. When the System Manager sets up your account, he establishes a UFD on a particular device. This becomes your default device. When you specify SY: in a command or file specification, the operating system makes the association to the actual physical device name.

Table 2-5 Pseudo Device Names

Pseudo Device Name	System-Wide or User-Specific?	Description
CL:	System	Console listing device, usually line printer
CO:	System	Console output device, usually TTØ:
LB:	System	System default device, "library device" containing system files
SP:	System	System spooling device, usually line printer
SY:	User	User default device, assigned individually by system manager
TI:	User	User terminal
WK:	System	Workfile device, usually fastest disk

Figure 2-4 shows the correlation between the pseudo devices (TI: and SY:) and the physical device for two users. For the user on terminal number 5, TI: equates to TT5: and SY: equates to DBØ:. For the second user, TI: equates to TT31: and SY: equates to DB1:.

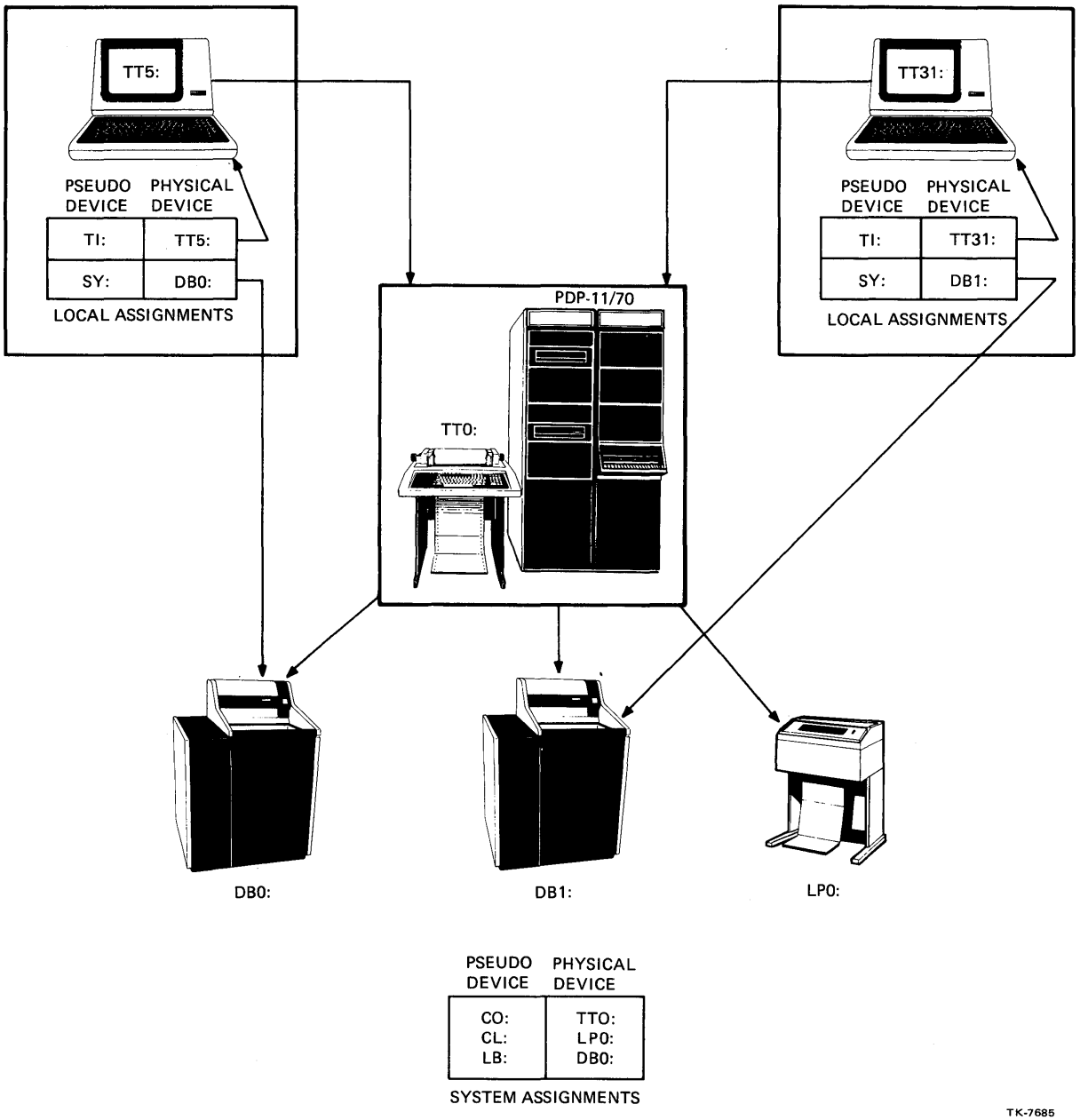


Figure 2-4 Pseudo Device Names

User File Directories (UFDs)

When your account is created for you, a file called a User File Directory (UFD) is also created. This file has the same number as your UIC and will contain the names and locations of all the files stored on the disk structure that belong to you. At least one UFD exists for every user with a system account, and a user may have a UFD on many different devices.

File Names

The name given to a file generally reflects what the file is for. It can be from Ø to 9 alphanumeric characters and is separated (delimited) from the file type by a period. File names are given to files either by you when you create them with the editor, or copy them, or by programs that generate output files.

File Types

The file type is a Ø to 3 alphanumeric character extension of the file name to further qualify the type of file. It is separated from the version number by a ";". There are standard file types and each utility recognizes its own particular file type (Figure 2-5). Whenever you see a file with a .MAC file type you may assume that the contents of that file contain MACRO-11 program source statements. Likewise, if you see a file with a .FTN file type, you can assume that the contents of the file are FORTRAN program source statements. Table 2-6 shows the standard file extensions and the type of file they represent.

Versions

Each file has a version number that distinguishes it from other versions of the same file. Version numbers are octal and range from 1 to 77777.

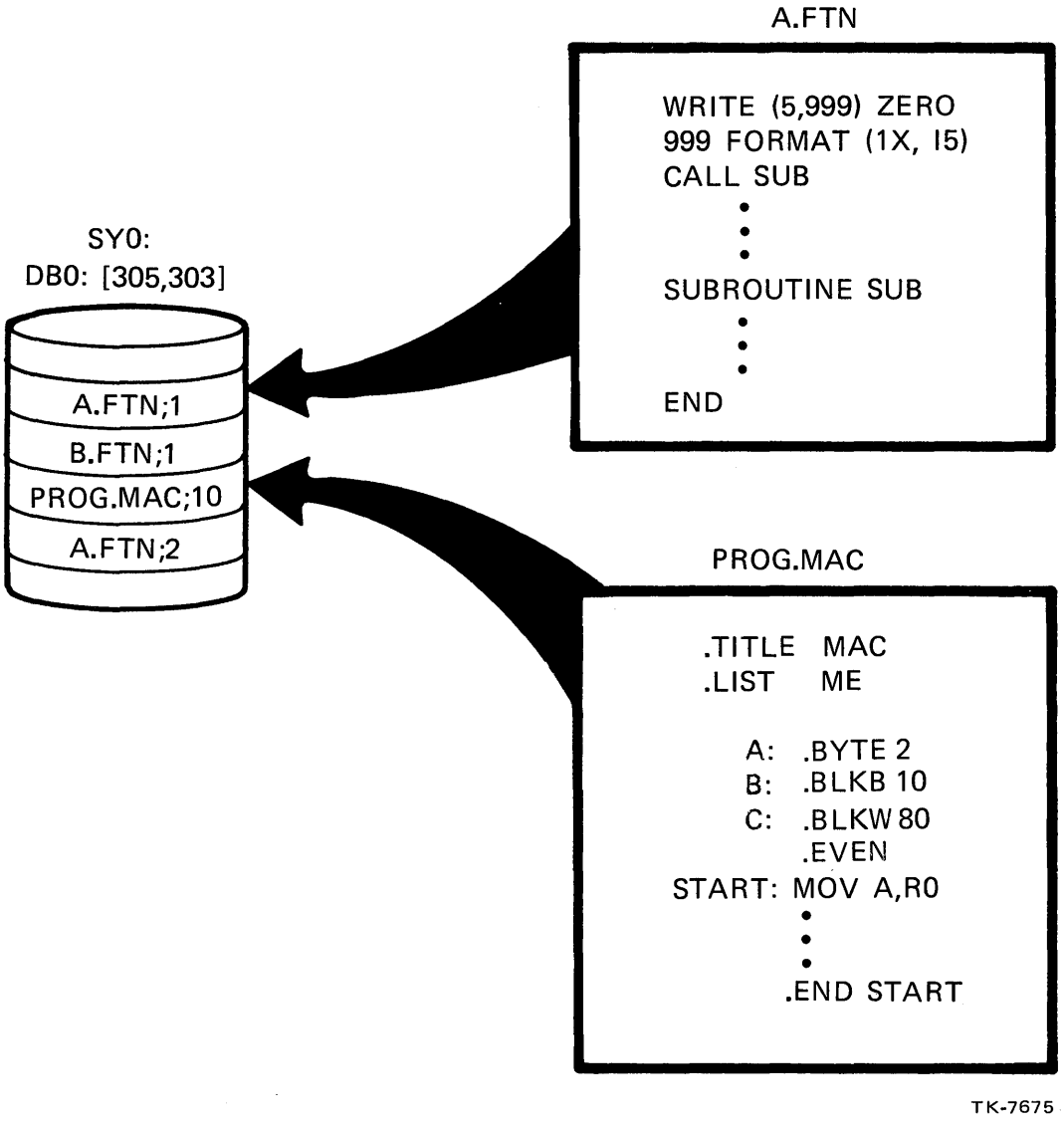


Figure 2-5 Standard File Types

Table 2-6 Standard File Types

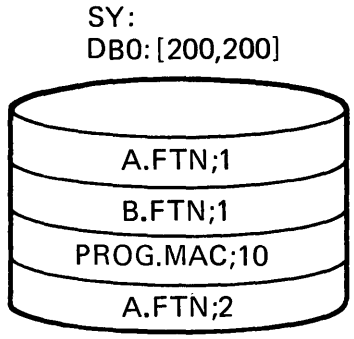
Type	File Contents
BAS	A BASIC language source program
B2S	A BASIC-PLUS-2 source program
CBL	A COBOL language source program
CMD	MCR or task commands (an indirect command file)
COR	A SLP correction file
DAT	Data (as opposed to a program)
DIR	A directory (for example, a User File Directory)
FTN	A FORTRAN language source program
LST	A listing
MAC	A MACRO-11 source program
MAP	A Task Builder memory allocation map
MLB	A macro library
OBJ	An object program (output from the MACRO-11 Assembler or a compiler)
ODL	A Task Builder overlay description
OLB	An object module library
SML	The system macro library
STB	A symbol table
SYS	A bootable system image
TMP	A temporary file
TSK	A task image
TXT	A text file
ULB	A universal library file

It is not necessary at all times to give complete file information when specifying a file. The system will make some assumptions about each field if you do not supply the information. We call this "taking the defaults." Table 2-7 lists the defaults the operating system takes for each field of the file specification. Taking the defaults and allowing the system to fill in the information saves you typing, time and possible typographical errors.

Table 2-7 File Specification Defaults

File Specification Element	Default
Device	SY:
UFD	User's default UFD
File Name	No default, must be supplied
File Type	Depends on command or utility
Version	On input file - latest version On output file - latest version plus 1

Figure 2-6 shows all the valid file specifications for the latest version of the file A.FTN. The amount of information necessary depends upon the situation in which the specification is being used. If [200,200] is your UFD, then you would not need to specify the UFD. If you want the latest version of the file, you would not have to specify the version. And if you were supplying the file to the FORTRAN compiler, you would not have to specify the file type. The FORTRAN compiler searches for a file type of FTN by default. You will become aware of how much information you must supply through your use of the system.



DB0:[200,200]A.FTN;2
SY:[200,200]A.FTN;2
[200,200]A.FTN;2
A.FTN;2
A.FTN
A.

TK-7672

Figure 2-6 Valid File Specifications for Latest A.FTN Version

Wildcards in the File Specification

Wildcards

Often, we wish to perform the same operations on many files with the same name or the same file type; for instance, copying all our macro source files from one UFD to another, or perhaps, deleting all our listing files. To type each of these files in the command line would be time-consuming and we would probably spend more time than desired correcting our typing errors. There is a feature of the operating system that allows us to substitute a special character to represent the concept of "all" in the file specification. Two special characters called wildcard characters can be used to do this.

The character "*" is the wildcard character representing "all" in a variable number of character positions. The character "%" is the wildcard character representing "all" in a single character position. For example, we can use these characters when we want to represent all:

- UFDs
- Groups
- Group members
- File names
- File types
- File versions

Notes on Example 2-6

The following comments keyed to Example 2-6 show the use of the wildcard characters and the results.

1. A listing of the directory file for the UFD [305,303].

2. A listing of the directory file for the UFD [305,304].

3. The (*) is used in the file type field to represent all file types. The system interprets this as the latest version of all files with the name PROG and any file type, and displays all the files that meet this criteria.

4. Using the (*) in the file name field allows us to obtain all our macro source files without having to type all the file names and extensions.

5. The (%) is used to indicate all possibilities in one character position. In this example we are asking the system to provide all listing files with a four character file name beginning with PR.

6. In this example, we are looking for six character task file names beginning with PR.

7. You may also use the (*) in the UFD. In this example we are looking for a listing of the directories of all the members of group 305. Member numbers 301,303, and 304 are listed for us.

8. In this example we are asking for all the task image files found in the directories of all the members of group 305.

9. Here we are asking for all files that have a file name with an "O" as the second character of the name. The file name will have at least two characters in it and may have as many as nine.

Table 2-8 shows the rules for using wildcard characters.

```
❶ Directory DK1:[305,303]
21-OCT-81 09:40

LOGIN.CMD;1              1.          04-AUG-81 08:43
VIP.TSK;1                4.       C  14-AUG-81 10:48
PAYROLL.CBL;1            2.          08-JAN-81 15:31
INFO.TXT;1              60.          07-JUL-80 12:47
LOGOUT.CMD;5             1.          28-MAY-80 08:49
PROG.FTN;1               6.          14-AUG-81 08:35
PROG.OBJ;1               0.          14-AUG-81 08:36
PROG.LST;1               0.          14-AUG-81 08:36
PROG.OBJ;2               5.          14-AUG-81 08:36
PROG.LST;2              11.          14-AUG-81 08:36
PROG.FTN;2               6.          14-AUG-81 08:36
PROG.FTN;3               6.          14-AUG-81 08:37
PROG.TSK;1               6.       C  14-AUG-81 08:37
MACIO.MAC;5              3.          04-AUG-81 09:14
MACIO.OBJ;5              1.          14-AUG-81 10:46
MACIO.LST;1              5.          14-AUG-81 10:46
DIRECTORY.LST;1          3.          18-AUG-81 16:28
XYZ.MAC;1                3.          19-AUG-81 09:43
XYZ.TSK;1                4.       C  19-AUG-81 09:43
VIP.OBJ;3                1.          19-AUG-81 09:46
VIP.MAC;5                3.          19-AUG-81 09:46
VIP.OBJ;4                1.          19-AUG-81 09:47
VIP.OBJ;5                1.          19-AUG-81 09:47
VIP.LST;5                5.          19-AUG-81 09:47
MACIO.MAC;7              3.          19-AUG-81 09:48
MACIO.MAC;6              3.          19-AUG-81 09:48
MACIO.MAC;10             3.          19-AUG-81 09:49
PRIMES.TSK;1            46.       C  19-AUG-81 13:57

Total of 193./199. blocks in 28. files

❷ Directory DK1:[305,304]
21-OCT-81 09:41

LOGIN.CMD;1              1.          04-AUG-81 08:43
LOGOUT.CMD;5             1.          28-MAY-80 08:49
STARTUP.CMD;57          14.          06-JUL-81 10:53
PRIMES.MAC;43            8.          15-JAN-81 16:15
ADVENTURE.TSK;1        151.       C  05-AUG-81 16:29
CLAMBAKE.TXT;1           3.          05-AUG-81 16:18
NOTICE.TXT;1            14.          16-MAR-81 10:34
PRIMES.OBJ;1             2.          14-AUG-81 11:09
PRIMES.LST;1            13.          14-AUG-81 11:09
PRIMES.TSK;1            46.       C  14-AUG-81 11:09
PRIMES.MAP;1             3.          14-AUG-81 11:09
A.MAC;2                  3.          18-AUG-81 16:30
VIP.TSK;1                4.       C  19-AUG-81 13:58
PROG.TSK;1               6.       C  18-AUG-81 16:31

Total of 269./273. blocks in 14. files
```

Example 2-6 Using Wildcards in a File Specification (Sheet 1 of 4)

```
❸ >DIR PROG.*

Directory DK1:[305,303]
21-OCT-81 09:41

PROG.FTN;3          6.          14-AUG-81 08:37
PROG.OBJ;2          5.          14-AUG-81 08:36
PROG.LST;2          11.         14-AUG-81 08:36
PROG.TSK;1          6.      C   14-AUG-81 08:37

Total of 28./28. blocks in 4. files


❹ >DIR *.MAC

Directory DK1:[305,303]
21-OCT-81 09:40

MACIO.MAC;10        3.          19-AUG-81 09:49
XYZ.MAC;1           3.          19-AUG-81 09:43
VIP.MAC;5           3.          19-AUG-81 09:46

Total of 9./13. blocks in 3. files


❺ >DIR PR%%.LST

Directory DK1:[305,303]
21-OCT-81 09:42

PROG.LST;2          11.         14-AUG-81 08:36

Total of 11./11. blocks in 1. file


❻ >DIR PR%%%.TSK

Directory DK1:[305,303]
21-OCT-81 09:42

PRIMES.TSK;1        46.     C   19-AUG-81 13:57

Total of 46./46. blocks in 1. file
```

Example 2-6 Using Wildcards in a File Specification (Sheet 2 of 4)

```
(7) >DIR [305,*]

Directory DK1:[305,303]
21-OCT-81 09:41

LOGIN.CMD;1          1.          04-AUG-81 08:43
VIP.TSK;1            4.       C  14-AUG-81 10:48
PAYROLL.CBL;1        2.          08-JAN-81 15:31
INFO.TXT;1           60.         07-JUL-80 12:47
LOGOUT.CMD;5         1.          28-MAY-80 08:49
PROG.FTN;1           6.          14-AUG-81 08:35
PROG.OBJ;1           0.          14-AUG-81 08:36
PROG.LST;1           0.          14-AUG-81 08:36
PROG.OBJ;2           5.          14-AUG-81 08:36
PROG.LST;2           11.         14-AUG-81 08:36
PROG.FTN;2           6.          14-AUG-81 08:36
PROG.FTN;3           6.          14-AUG-81 08:37
PROG.TSK;1           6.       C  14-AUG-81 08:37
MACIO.MAC;5          3.          04-AUG-81 09:14
MACIO.OBJ;5          1.          14-AUG-81 10:46
MACIO.LST;1          5.          14-AUG-81 10:46
DIRECTORY.LST;1      3.          18-AUG-81 16:28
XYZ.MAC;1            3.          19-AUG-81 09:43
XYZ.TSK;1            4.       C  19-AUG-81 09:43
VIP.OBJ;3            1.          19-AUG-81 09:46
VIP.MAC;5            3.          19-AUG-81 09:46
VIP.OBJ;4            1.          19-AUG-81 09:47
VIP.OBJ;5            1.          19-AUG-81 09:47
VIP.LST;5            5.          19-AUG-81 09:47
MACIO.MAC;7          3.          19-AUG-81 09:48
MACIO.MAC;6          3.          19-AUG-81 09:48
MACIO.MAC;10         3.          19-AUG-81 09:49
PRIMES.TSK;1         46.      C  19-AUG-81 13:57

Total of 193./199. blocks in 28. files

Directory DK1:[305,304]
21-OCT-81 09:41

LOGIN.CMD;1          1.          04-AUG-81 08:43
LOGOUT.CMD;5         1.          28-MAY-80 08:49
STARTUP.CMD;57       14.         06-JUL-81 10:53
PRIMES.MAC;43        8.          15-JAN-81 16:15
ADVENTURE.TSK;1      151.     C  05-AUG-81 16:29
CLAMBAKE.TXT;1       3.          05-AUG-81 16:18
NOTICE.TXT;1         14.         16-MAR-81 10:34
PRIMES.OBJ;1         2.          14-AUG-81 11:09
PRIMES.LST;1         13.         14-AUG-81 11:09
PRIMES.TSK;1         46.      C  14-AUG-81 11:09
PRIMES.MAP;1         3.          14-AUG-81 11:09
A.MAC;2              3.          18-AUG-81 16:30
VIP.TSK;1            4.       C  19-AUG-81 13:58
PROG.TSK;1           6.       C  18-AUG-81 16:31

Total of 269./273. blocks in 14. files


Directory DK1:[305,301]
21-OCT-81 09:41

INFO.TXT;1           60.         17-AUG-81 14:54

Total of 60./60. blocks in 1. file

Grand total of 522./532. blocks in 43. files in 3. directories
```

Example 2-6 Using Wildcards in a File Specification (Sheet 3 of 4)

```
(8) >DIR [305,*]*.TSK

Directory DK1:[305,303]
21-OCT-81 09:44

VIP.TSK;1            4.      C  14-AUG-81 10:48
PROG.TSK;1           6.      C  14-AUG-81 08:37
XYZ.TSK;1            4.      C  19-AUG-81 09:43
PRIMES.TSK;1        46.      C  19-AUG-81 13:57

Total of 60./60. blocks in 4. files


Directory DK1:[305,304]
21-OCT-81 09:44

ADVENTURE.TSK;1     151.     C  05-AUG-81 16:29
PRIMES.TSK;1         46.     C  14-AUG-81 11:09
VIP.TSK;1             4.     C  19-AUG-81 13:58
PROG.TSK;1            6.     C  18-AUG-81 16:31

Total of 207./207. blocks in 4. files

Grand total of 267./267. blocks in 8. files in 2. directories


(9) >DIR %0*.*

Directory DK1:[305,303]
21-OCT-81 09:45

LOGIN.CMD;1          1.          04-AUG-81 08:43
LOGOUT.CMD;5         1.          28-MAY-80 08:49

Total of 2./2. blocks in 2. files
```

Example 2-6 Using Wildcards in a File Specification (Sheet 4 of 4)

Table 2-8 Allowed Use of Wildcard Characters

File Specification Field	Is (*) allowed?	Is (%) allowed?
Device name	No	No
UFD	Yes, to replace an entire group and/or member field	No
File name	On an input file, allowed to replace all or part of the file name	On an input file, allowed to replace all or part of the file name
	On an output file, allowed to replace only the entire file name	Not allowed on an output file
File type	Same as file name	Same as file name
Version number	Yes, to replace an entire version number	No

LEARNING ACTIVITY

1. READ the following sections in the RSX-11M/M-PLUS Command Language Manual:

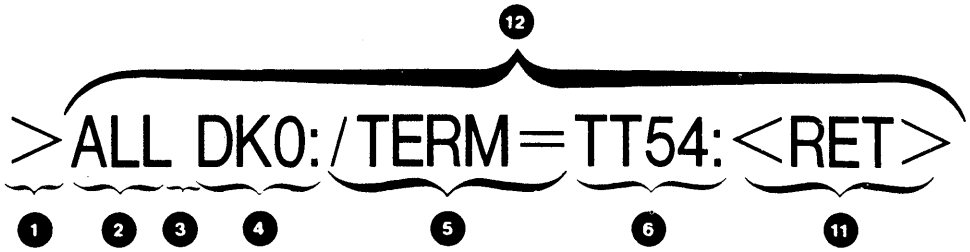
   - 4.1, Files on RSX-11M/M-PLUS Systems
   - 5.1, Devices on RSX-11M/M-PLUS Systems
   - 5.1.2, Physical Devices, Pseudo Devices, LUNS and Logical Names

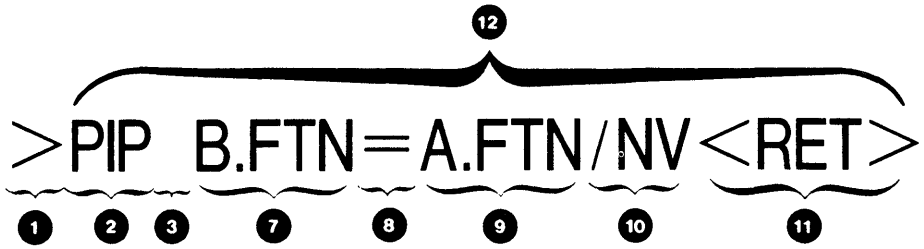
USING THE MONITOR CONSOLE ROUTINE (MCR)

MCR

MCR is the original command line interpreter of the RSX-11M/M-PLUS operating systems. It is the root for all DCL commands, and contains all the functionality of DCL and some privileged functions that cannot be accomplished in DCL. The format of the MCR command line is shown below. A partial list of MCR commands is shown in Table 2-9. Appendix B of the RSX-11M/M-PLUS Command Language Manual contains associated MCR commands for each DCL command.

MCR Command Format





1. Operating system prompt
2. MCR command
3. Command delimiter (blank or tab)
4. Parameter
5. Keyword
6. Keyword value
7. Output file specification
8. Delimiter
9. Input file specification
10. Switch
11. Command line terminator
12. Command line

Table 2-9 Partial List of MCR Commands

User Wants To	MCR Command To Use
Check the time and day	TIM[E]
Determine who is logged on the system	DEV[ICE]TT:
Check the print queue or batch job status	QUE[UE]/LI
Get information on terminal settings	SET
Determine what hardware is available on the system	DEV
Determine what partitions are available on the system	PAR
Protect a file from being deleted	PIP
Determine which tasks are installed on the system	TAS
Determine which tasks are active on the system	ATL
Determine which tasks are available at user's terminal	ACT
Run a task	RUN
Mount a volume in FILES-11 structure	MOU
Determine the logical units associated with a task	LUN
Send a message to another user	BRO
Allocate a device for private use	ALL
Abort a running task	ABO
Cancel a time-based initiation request from the clock queue	CAN

Notes on Example 2-7

The following comments are keyed to the example.

1. Active command issued that shows there are three tasks active at your terminal.

2. ABORT command issued.

3. ACTIVE command issued again to see what tasks were active.

4. ABORT command issued to abort the task named TT56.

5. SET command issued to make DCL active at the terminal.

6. SHOW USERS command issued improperly. An error message is returned to indicate that the command was not recognized by DCL.

7. SHOW USERS command properly issued.

8. Active Task List (ATL) command issued to display all the tasks that are active on this system at the time. See the RSX-11M/M-PLUS MCR Operations Manual for description of the printout.

9. Installed Task List Short Form (TAS) command to display all tasks that are installed in the System Task Directory.

10. Partition (PAR) command issued to display the partition configuration of memory. See the RSX-11M/M-PLUS MCR Operations Manual for a description of the printout.

11. Device (DEV) command issued to display information about the magnetic tape units on the system. This display shows that there are two units. The first unit is privately allocated to TT2Ø:; the second is available for use.

12. Queue (QUE) command issued to list information about the print queue. There are no jobs waiting to be printed.

13. Device (DEV) command issued again, only this time to display information about all the devices (both physical and pseudo) that are part of the system.

```
1  MCR>ACT
   MCR...
   ...MCR
   TT56
2  >ABO
3  >ACT
   MCR...
   ...MCR
   TT56
4  >ABO TT56
   ABO -- Task marked for abort
5  >SET /DCL=TI:
6  >SOW USERS
   DCL  --  Illegal command
   >
   >
   >SET /DCL=TI:
   >
7  DCL>SHOW USERS
   HT1: [7,22]
   TT6: [7,374]
   TT11: [7,32]
   TT12: [7,302]
   TT13: [7,42]
   TT14: [7,110]
   TT15: [7,113]
   TT16: [301,303]
   TT17: [7,366]
   TT21: [7,372]
   TT22: [7,116]
   TT23: [301,333]
   TT24: [55,30]
   TT25: [301,44]
   TT27: [7,361]
   TT51: [100,24]
   TT53: [7,337]
   TT55: [2,201]
   TT56: [305,303]
   TT57: [7,113]
   >LO
   Have a Good Morning
   22-OCT-81 11:03 TT56: logged off QUASAR
   ^
```

Example 2-7 MCR Command Samples (Sheet 1 of 6)

```
(8) >ATL
LDR... 112050  LDRPAR 112404 00255700-00260400  PRI - 248.  DPRI - 248.
   STATUS:  -CHK FXD STP -PMD PRV NSD
   TI - C00:  IOC - 0.  BIO - 0.  EFLG - 000001 000000  PS - 170000
   PC - 120370 REGS 0-6 120212 013541 177777 107646 074700 107614 120166
RMDEMO 053120  GEN    047560 01062100-01111100  PRI - 225.  DPRI - 225.
   STATUS:  -CHK WFR -PMD PRV MCR
   TI - TT0:  IOC - 1.  BIO - 0.  EFLG - 000020 040000  PS - 170010
   PC - 136254 REGS 0-6 000000 136333 132110 000000 136375 000243 121164
MTAACP 051274  GEN    100544 00643600-00657400  PRI - 200.  DPRI - 200.
   STATUS:  STP ACP -PMD PRV NSD
   TI - C00:  IOC - 0.  BIO - 0.  EFLG - 000006 040002  PS - 170001
   PC - 120650 REGS 0-6 051306 077304 051306 077244 000000 120656 120352
NETACP 047214  GEN    062220 01232000-01257600  PRI - 200.  DPRI - 200.
   STATUS:  STP ACP -PMD PRV NSD
   TI - C00:  IOC - 0.  BIO - 0.  EFLG - 000000 000000  PS - 170000
   PC - 120452 REGS 0-6 000004 000010 104170 000000 065504 063404 120242
EVP... 060754  GEN    061140 00466700-00503100  PRI - 199.  DPRI - 199.
   STATUS:  STP -PMD PRV
   TI - C00:  IOC - 0.  BIO - 0.  EFLG - 000001 000000  PS - 170000
   PC - 120512 REGS 0-6 127536 000000 003424 130556 000011 125674 120272
MCR... 110160  SYSPAR 114514 00203100-00215100  PRI - 160.  DPRI - 160.
   STATUS:  STP -PMD PRV MCR CLI NSD CAL
   TI - TT56:  IOC - 0.  BIO - 0.  EFLG - 000001 040000  PS - 170000
   PC - 122436 REGS 0-6 000000 120476 122032 120432 010766 044146 120366
...MCR 107614  GEN    074700 01155100-01175400  PRI - 160.  DPRI - 160.
   STATUS:  CKD -PMD PRV MCR NSD
   TI - TT56:  IOC - 0.  BIO - 0.  EFLG - 000001 040000  PS - 170000
   PC - 121454 REGS 0-6 120424 121276 121706 000000 121302 000000 000742
CA.T24 101444  GEN    077424 00657400-00664500  PRI - 150.  DPRI - 150.
   STATUS:  WFR -PMD REM PRV
   TI - TT24:  IOC - 0.  BIO - 0.  EFLG - 000000 000000  PS - 170000
   PC - 123266 REGS 0-6 124114 000037 122470 000001 135600 121257 121172
F11ACP 111550  GEN    060534 01111100-01155100  PRI - 149.  DPRI - 149.
   STATUS:  STP ACP -PMD PRV NSD CAL
   TI - C00:  IOC - 0.  BIO - 0.  EFLG - 000002 000001  PS - 170000
   PC - 135350 REGS 0-6 155272 000000 000020 102373 060644 076224 120310
WKFFCP 111320  GEN    046160 00526500-00572500  PRI - 149.  DPRI - 149.
   STATUS:  -CHK STP ACP -PMD PRV NSD
   TI - C00:  IOC - 0.  BIO - 0.  EFLG - 000002 000001  PS - 170000
   PC - 135350 REGS 0-6 000400 155254 000000 000034 047344 100500 120310
SY0FCP 047434  GEN    046034 00572500-00636500  PRI - 149.  DPRI - 149.
   STATUS:  -CHK STP ACP -PMD PRV NSD
   TI - C00:  IOC - 0.  BIO - 0.  EFLG - 000002 000001  PS - 170000
   PC - 135350 REGS 0-6 000000 074130 000032 151460 046640 074070 120310
DB1FCP 046470  GEN    045430 00723600-00767600  PRI - 149.  DPRI - 149.
   STATUS:  -CHK STP ACP -PMD PRV NSD
   TI - C00:  IOC - 0.  BIO - 0.  EFLG - 000002 000001  PS - 170000
   PC - 135350 REGS 0-6 034024 076444 000032 150570 051114 076404 120310
DB2FCP 050460  GEN    046370 00767600-01013600  PRI - 149.  DPRI - 149.
   STATUS:  -CHK ACP -PMD PRV NSD
```

Example 2-7 MCR Command Samples (Sheet 2 of 6)

```
❾ >TAS
LDR...  12.19   LDRPAR  248.  00002500 LB0:-01240437 FIXED
TKTN    04.7    TKNPAR  248.  00010000 LB0:-01242244
RMDEMO  V1.05   GEN     225.  00027000 LB0:-01262570
MTAACP  0013    GEN     200.  00013600 EM0:-00000325
F11MSG  V0012   GEN     200.  00005500 LB0:-01251052
NETACP  V02.00  GEN     200.  00025600 SY0:-01266761
EVP...  V01.00  GEN     199.  00013100 SY0:-01266561
MCR...  2.02    SYSPAR  160.  00012000 LB0:-01244424
...MCR  1.1     GEN     160.  00020300 LB0:-01244760
...MOU  2503    GEN     160.  00037600 LB0:-00117642
...DCL  00      GEN     160.  00035600 SY0:-01132456
...HEL  01.0    GEN     150.  00037600 LB0:-01250645
...CA.  4.00    GEN     150.  00005100 LB0:-01251064
...CVT  4.02    GEN     150.  00052200 SY0:-01007774
...DLG  4.02    GEN     150.  00052200 SY0:-01007774
NVP...  V01.00  GEN     150.  00012700 SY0:-01265031
...BYE  04.0    GEN     150.  00012500 SY0:-01264767
F11ACP  M0401   GEN     149.  00044000 LB0:-01247227
WKFFCP  M0401   GEN     149.  00044000 LB0:-01262523
SY0FCP  M0401   GEN     149.  00044000 EM0:-00000260
DB1FCP  M0401   GEN     149.  00044000 SY0:-01262313
DB2FCP  M0401   GEN     149.  00024000 SY0:-01262404
PMT...  01.52   GEN     148.  00006300 LB0:-01246016
ERRLOG  V1.04   GEN     148.  00040000 LB0:-01251006
COT...  01.01   GEN     145.  00017000 LB0:-01250736
...DMO  23.25   GEN     140.  00014600 LB0:-01245770
...INI  22.02   GEN     140.  00034600 LB0:-01246445
...UFD  V0412   GEN     140.  00005700 LB0:-01246615
PMD...  07.2    GEN     130.  00017600 LB0:-01250210
SHF...  5.08    SYSPAR  105.  00012000 LB0:-01250203
...INS  06      GEN     100.  00034400 LB0:-01243744
...SEN  22JUL   GEN     100.  00104400 LB0:-00662645
RMHACP  V02.00  GEN     100.  00023300 SY0:-01265073
MAL$$$  11RX    GEN     100.  00132300 SY0:-00663342
PMR$$$  V01.05  GEN     100.  00034300 SY0:-00660507
QMG...  1.8     GEN      70.  00032200 LB0:-01252004
LPO     1.9     GEN      70.  00014100 LB0:-01252136
PRT...  1.2     GEN      70.  00001100 LB0:-01251443
...BRU  1.03    GEN      70.  00173500 LB0:-01256414
...KKK          GEN      70.  00163400 DK2:-00010412
...TEC  V36     GEN      65.  00177100 EM0:-00001215
...EDI  M11.04  GEN      65.  00123600 EM0:-00001371
...SOS  07.00   GEN      65.  00073300 EM0:-00001547
...EDT  02.00   GEN      65.  00132000 LB0:-01254476
...MAI  01SEP   GEN      65.  00123700 SY0:-00555723
RMT...  V02.01  GEN      65.  00027300 SY0:-01266505
...RVT  X02.19  GEN      65.  00032500 SY0:-00663556
EDTT16  02.00   GEN      65.  00132000 LB0:-01254476
EDTT22  02.00   GEN      65.  00132000 LB0:-01254476
...AT.  02      GEN      64.  00060000 LB0:-01246107
NTL...  V02.00  GEN      60.  00036500 SY0:-01265603
NTINIT  V02.00  GEN      55.  00005000 SY0:-01265347
...BRO  V04     GEN      50.  00031000 LB0:-01250552
LPINIT  01.03   GEN      50.  00016200 LB0:-01250260
```

Example 2-7 MCR Command Samples (Sheet 3 of 6)

```
...QUE 1.10    GEN     50. 00025200 LB0:-01252041
...PRI 1.10    GEN     50. 00025200 LB0:-01252041
...PIP M1343   GEN     50. 00132300 EM0:-00000443
...MAC M1200   GEN     50. 00110100 EM0:-00000557
...TKB M40.00  GEN     50. 00102400 EM0:-00002310
...LBR 06.00   GEN     50. 00154400 EM0:-00001103
CRF... V01     GEN     50. 00116600 EM0:-00001165
...MUN V36     GEN     50. 00177100 EM0:-00001215
...CMP V0109   GEN     50. 00151300 EM0:-00001341
...RNO M0101   GEN     50. 00061600 EM0:-00001443
...SLP 11.03   GEN     50. 00027400 EM0:-00001515
...SRD 4.1F    GEN     50. 00043500 EM0:-00001750
...BCK 0002CM  GEN     50. 00065600 SY0:-00065660
...DOC 06DEC   GEN     50. 00061600 SY0:-00063566
...DTR V01.10  GEN     50. 00175000 SY0:-00063652
MAL... 30OCT   GEN     50. 00057700 SY0:-00553461
NICE.. V02.00  GEN     50. 00026600 SY0:-01266362
...NCP V01.00  GEN     50. 00037100 SY0:-01266061
FAL... V02.01  GEN     50. 00075200 SY0:-01267327
...NFT V02.00  GEN     50. 00035400 SY0:-01267224
.CMTS. V02.00  GEN     50. 00002000 SY0:-01235557
TCL... V02.00  GEN     50. 00002400 SY0:-01265126
...NTD V01.00  GEN     50. 00021200 SY0:-01265005
NTD... V01.00  GEN     50. 00011700 SY0:-01266544
...RMT V02.00  GEN     50. 00002100 SY0:-01235551
...FLX M15     GEN     50. 00037700 LB0:-01253617
...LOA 03.3    GEN     50. 00033100 LB0:-01244354
...UNL 3.0     GEN     50. 00025100 LB0:-01245146
...F4P V3.0    GEN     50. 00122600 SY0:-00107722
...QWE  KHTUE  GEN     50. 00040000 SY0:-00144024
...TMC 30SEP   GEN     50. 00043700 SY0:-01106661
EX1    21OCT   GEN     50. 00055700 SY0:-00132564
EX2    21OCT   GEN     50. 00055700 SY0:-00132564
EXST   21OCT   GEN     50. 00055700 SY0:-00132564
EX3    21OCT   GEN     50. 00055700 SY0:-00132564
HT1    22OCT   GEN     50. 00107400 SY0:-00425313
TT12   18JAN   GEN     50. 00112600 SY0:-00546303
TT23   M1332   GEN     50. 00156400 SY0:-00134222
```

Example 2-7 MCR Command Samples (Sheet 4 of 6)

```
(10) >PAR
CEXPAR 114734 00115000 00003000 MAIN COM
TTPAR  114670 00120000 00040000 MAIN TASK
EXCOM1 114624 00160000 00014700 MAIN COM
EXCOM2 114560 00174700 00006200 MAIN COM
SYSPAR 114514 00203100 00012000 MAIN TASK
TKNPAR 114450 00215100 00010000 MAIN TASK
DRVPAR 114404 00225100 00030600 MAIN SYS
       114340 00225100 00002100 SUB  DRIVER -DB:
       114240 00227200 00001200 SUB  DRIVER -DK:
       114140 00230400 00003100 SUB  DRIVER -DM:
       114040 00233500 00003000 SUB  DRIVER -DR:
       113740 00236500 00001000 SUB  DRIVER -EM:
       113640 00237600 00001100 SUB  DRIVER -DT:
       113540 00240700 00001400 SUB  DRIVER -DX:
       113440 00242300 00002500 SUB  DRIVER -DL:
       113340 00245000 00002600 SUB  DRIVER -DD:
       113240 00247600 00001300 SUB  DRIVER -LP:
       113104 00251100 00004300 SUB  DRIVER -MM:
       112450 00255400 00000300 SUB  DRIVER -CO:
LDRPAR 112404 00255700 00002500 MAIN TASK
BASIC2 112340 00260400 00040000 MAIN COM
FCSRES 112274 00320400 00040000 MAIN COM
TSTPAR 112230 00360400 00100000 MAIN TASK
GEN    112164 00460400 03317400 MAIN SYS
       045174 00460400 00006300 SUB  (PMT...)
       061140 00466700 00014200 SUB  (EVP...)
       046100 00507200 00017000 SUB  (COT...)
       046160 00526500 00044000 SUB  (WKFFCP)
       046034 00572500 00044000 SUB  (SYOFCP)
       100544 00643600 00013600 SUB  (MTAACP)
       075740 00666000 00023300 SUB  (RMHACP)
       045430 00723600 00044000 SUB  (DB1FCP)
       046370 00767600 00024000 SUB  (DB2FCP)
       046270 01013600 00032200 SUB  (QMG...)
       044770 01046000 00014100 SUB  (LP0   )
       047560 01062100 00027000 SUB  (RMDEMO)
       060534 01111100 00044000 SUB  (F11ACP)
       062434 01155100 00020300 SUB  (...MCR)
       062220 01232000 00025600 SUB  (NETACP)
       073370 01300100 00055700 SUB  (EXST  )
       070750 01362200 00132000 SUB  (...EDT)
       045240 01516200 00132300 SUB  (...PIP)
       071730 01650500 00107400 SUB  (HT1   )
       076224 01760100 00132000 SUB  (EDTT22)
       074554 02112100 00060000 SUB  (...AT.)
       076760 02172100 00132000 SUB  (EDTT16)
       071040 02324400 00055700 SUB  (EX2   )
       075650 02402300 00112600 SUB  (TT12  )
       100610 02777700 00153100 SUB  (...TKB)
NT.PCL 062634 03671600 00003500 SUB  DYNAMIC
NT.NSP 053244 03675300 00016000 SUB  DYNAMIC
NT.AUX 062264 03713300 00001100 SUB  DYNAMIC
POOL.. 061074 03714400 00061700 SUB  DYNAMIC
       045034 03776300 00001500 SUB  DRIVER -HT:
```

Example 2-7 MCR Command Samples (Sheet 5 of 6)

```
>SET TERMINAL /MCR
>TIME
20:11:56 06-JAN-82
(11) >DEV MM:
MM0: TT20: - Private Loaded
MM1: Loaded
(12) >QUE/LI
** PRINT QUEUES **
PRINT  => LP0
LP0    => LP0
(13) >DEV
HT0: Offline Loaded
HT1: Offline Loaded
HT2: Offline Loaded
HT3: Offline Loaded
LP0: Loaded
DB0: Loaded Type=RP05
DB1: Public Mounted Loaded Label=11MDEV Type=RP06
DB2: Public Mounted Loaded Label=RSXUTIL Type=RP06
DD0: Offline Loaded
DD1: Offline Loaded
DK0: Loaded
DK1: Loaded
DK2: Loaded
DK3: Loaded
DL0: TT20: - Private Loaded Type=RL01
DL1: Loaded Type=RL01
DL2: Loaded Type=RL02
DM0: Offline Loaded Type=RK06
DM1: Loaded Type=RK06
DM2: Loaded Type=RK07
DR0: Public Mounted Loaded Label=QUASARUSERS Type=RP07
DR1: Loaded Type=RM03
DS0: Offline Unloaded Type=RS04
DS1: Offline Unloaded Type=RS04
DT0: Loaded
DT1: Loaded
DX0: Loaded
DX1: Loaded
EM0: Offline Unloaded Type=ML11
MM0: TT20: - Private Loaded
MM1: Loaded
CO0: Loaded
NS0: Public Mounted Loaded
TT0: Loaded
TT1: Loaded
TT2: Loaded
TT3: Loaded
TT4: Loaded
TT5: Loaded
NL0:
TI0:
CL0:    LP0:
LB0:    DR0:
SY0:    DR0:
>
```

Example 2-7 MCR Command Samples (Sheet 6 of 6)

PRIVILEGED COMMANDS

There are commands that could possibly harm the work of other users and even affect system performance. These are privileged commands. To execute these commands you must either have an account with a group number of 1Ø or less (a system user, also called a privileged user) or have had your terminal set to privileged by a system user. Table 2-1Ø lists some of the commands that require having privileges to perform them. If at any time you need to perform a privileged command and do not have a privileged account, see your system manager or a system programmer for help.

Table 2-1Ø Partial List of Privileged Commands

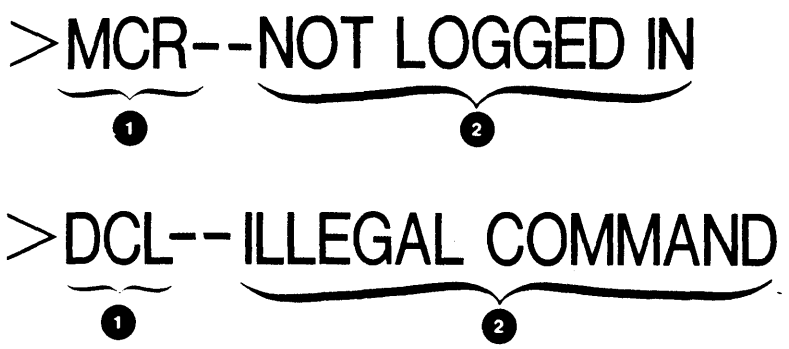
User Wants To	DCL Command to Use
Install a task in the STD	INSTALL
Run a task on a time-based interval	RUN
Broadcast a message to all terminals	BROADCAST
Dismount any mounted volume	DISMOUNT
Set the time and day	SET DAYTIME
Set a terminal to slave	SET TERMINAL
Create partitions and subpartitions	SET PARTITIONS
Enable or disable logins	SET SYSTEM
Set the privilege status of a terminal	SET TERMINAL
Allocate checkpoint space	SET DEVICE
Change the priority of a running task	SET PRIORITY
Fix a task in memory	FIX
Remove an entry from the STD	REMOVE

INTERPRETING ERROR MESSAGES

Error messages are returned when you have improperly supplied information in a command line, or the operating system wants to inform you that what you have requested is not available.

The error message format is shown below. Table 2-11 shows some common error messages.

Error Message Format



1. Task from which the error was generated
2. Message describing the error condition

Table 2-11 Some Common Error Messages

Message	Explanation
MCR -- Not logged in	You tried to enter a command before logging in. Type LOGIN or HELLO.
HEL -- Other user logged on	Someone left the terminal without logging out. Type LOGOUT or BYE, then log in.
HEL -- Logins are disabled	Usually means system is still in the process of starting up, or is being shut down. If it persists, see your system manager or instructor.
MCR -- Task not in system	The task you specified was not located in the installed task system list.
DCL -- Illegal command	The command you issued was not a DCL command.

LEARNING ACTIVITIES

1. READ Chapter 1, Introduction to MCR, in the RSX-11M/M-PLUS MCR Operations Manual.

2. READ Appendix A: Common Error Messages, in the RSX-11M/M-PLUS Command Language Manual.

3. DO the Lab Exercises for this module.

CREATING AND MODIFYING FILES

3

INTRODUCTION

Users spend considerable time creating and modifying text files. Text files are files consisting of ASCII characters - either straight text (e.g., a letter or a manual) or source text (e.g., a MACRO or FORTRAN program). There are a number of DEC editors available for this purpose.

In this module, you will learn to use one of the more powerful DEC editors, EDT. In another module, you will learn some additional techniques available with EDT. The two modules together cover a subset of the full range of EDT facilities needed for most file editing. After you become adept at using this editor, read over the resources below to find some short cuts and even more sophisticated techniques.

OBJECTIVES

1. Invoke the EDT editor, execute editor commands, and exit from the editor.

2. Use the editor to perform the following functions:

   a. Create text files.

   b. Change existing text files using line-oriented and character-oriented editing techniques.

   c. Produce output files containing the results of an editing session.

RESOURCES

EDT Editor Manual

OVERVIEW

There are many devices used to input data into the computer. Everything from magnetic tape to pressure and temperature measuring devices can send information to the CPU, memory or disk. However, these are mechanical devices that do not allow for human interaction. Not long ago punched cards were the most popular method used for inputting data. Today, we consider that method rather archaic and use instead a faster, more interactive device, the terminal.

Purpose of an Editor

A text editor is an interactive task used in conjunction with a terminal to create and edit ASCII text files that are stored on disk. It is used to create:

- documentation files
- memo files
- program source files
- program data files

Figure 3-1 shows the process of creating a text file using EDT, DIGITAL's standard editor. A user, sitting at a terminal, will invoke the editor and after supplying the proper commands for specifying a new file name and indicating the input mode, will start inputting the lines of desired text. The editor program collects the string of characters (exactly as typed) into a buffer until the user terminates the input mode. It is not until the user instructs the editor to save the buffer that a file is written to disk to preserve the editing session. The file written to disk is in ASCII format; no translation has been performed on its contents. The contents of the file may be viewed at the terminal with the aid of a utility task.

Figure 3-2 shows the process of editing a text file that already exists. When the name of an existing file is supplied to the editor, a copy of that file is read into the editor's main buffer. The editor then waits for commands to change the copy of the file that exists in this buffer. Once again, those changes are not reflected in a disk file until the user instructs the editor to save the contents of the buffer. When such a command is issued, a new copy (version) of the file is written to disk.

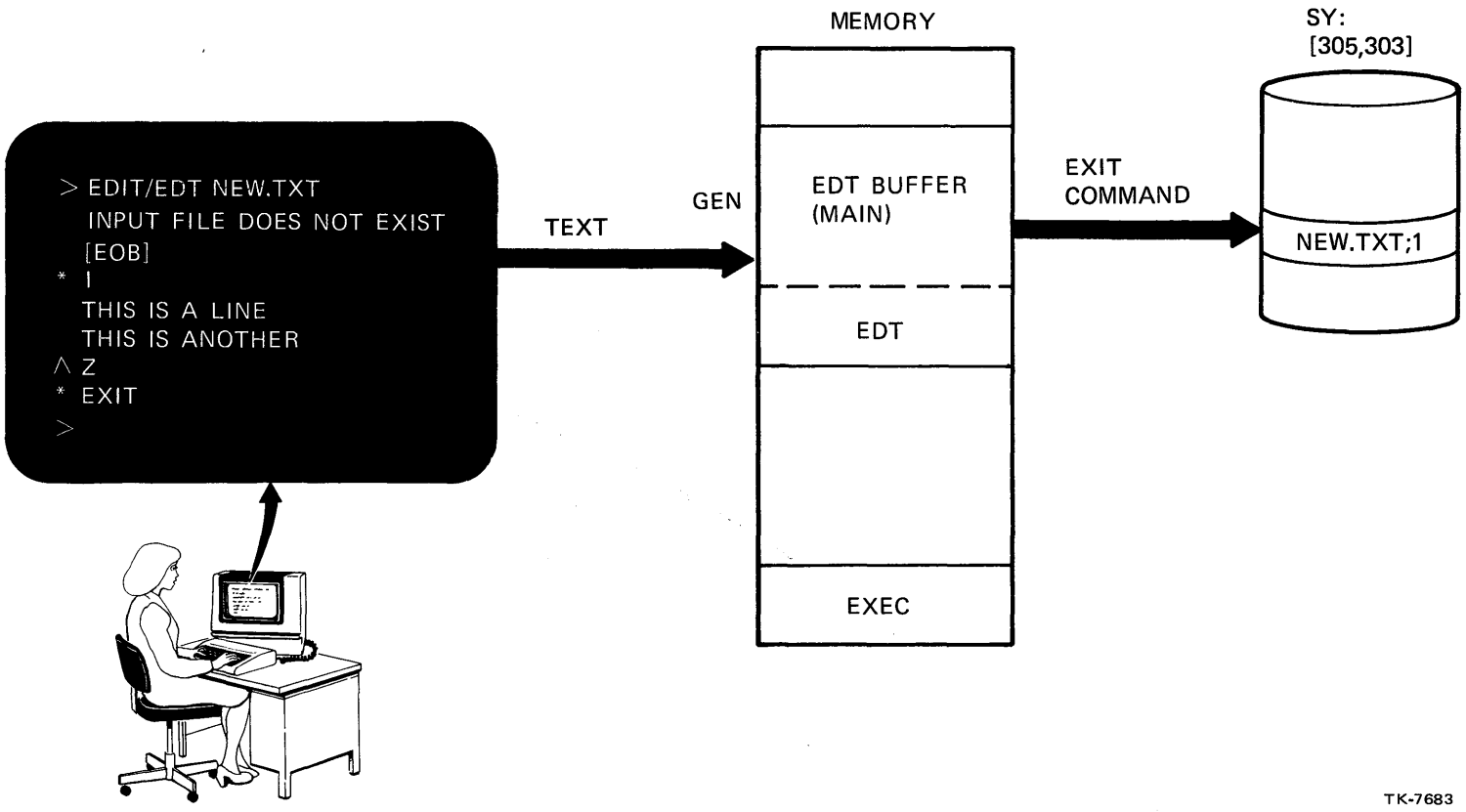


Figure 3-1 Creating a Text File

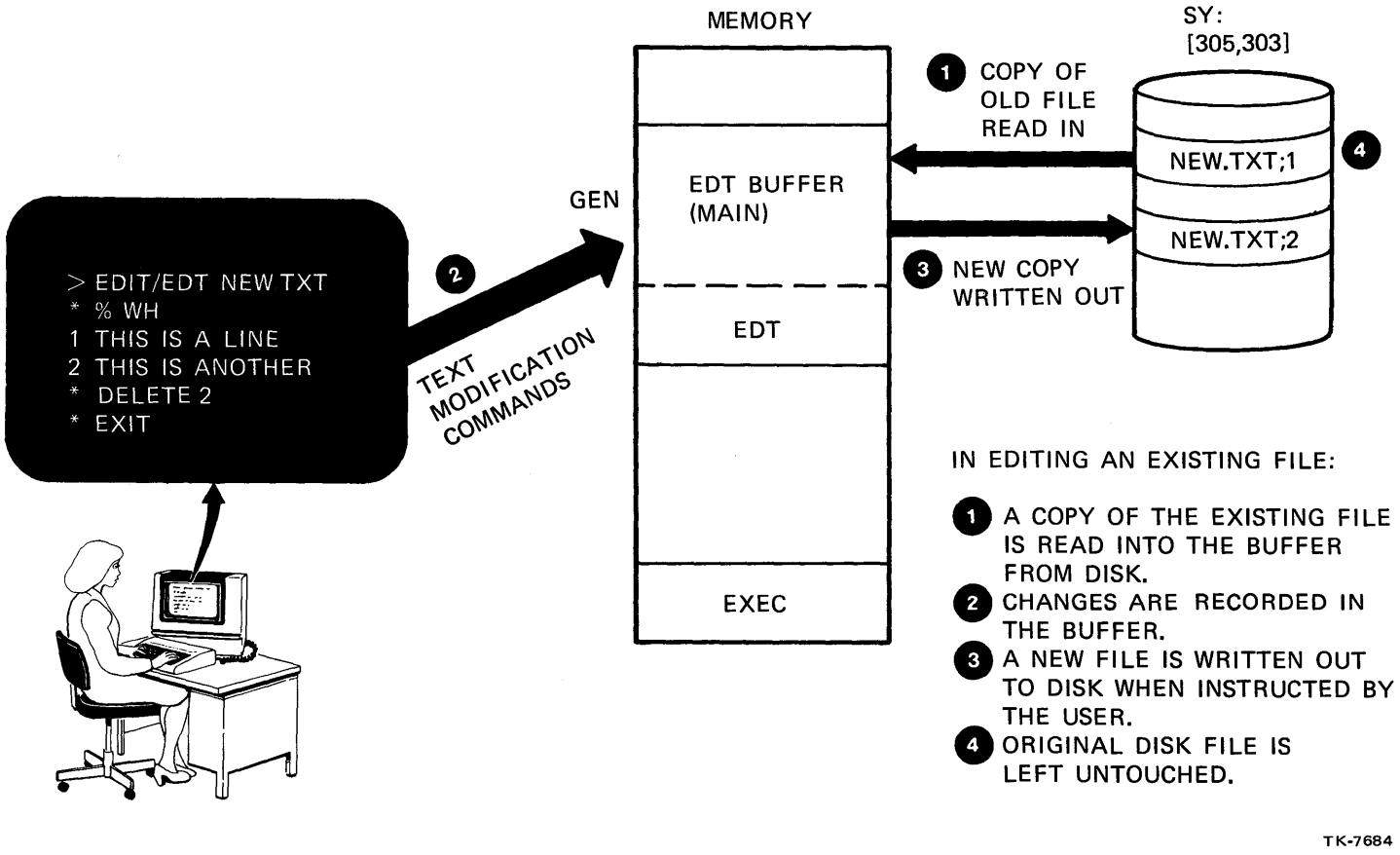


Figure 3-2 Modifying Existing Text Files

EDT Editor Features

EDT is a text editor that is available on many of DIGITAL's operating systems. It offers many features that make text editing easier and more efficient. These features are listed below.

- DIGITAL's standard editor
- Line mode/character mode - EDT performs editing operations on a line-by-line or character-by-character basis. In line mode the cursor position is always at the left-most side of the display. In character mode the cursor moves freely from left to right and up and down on the display.
- English-like commands
- Help facility
- File protection by journaling - While performing your editing operations, EDT is recording all the keystrokes in a file. If for any reason the editing session abnormally terminates, this file is available for processing to recover what has been edited.
- Startup command files - The editor reads a special file containing editing commands that create a processing environment tailored to the user's preferred methods of editing.
- Redefine key functions
- Multiple buffers and file access - A buffer to an editor user is like a sheet of paper to a writer. A writer might want a long sheet of paper to write his story and another sheet on which to make notes. More than one buffer can be created in EDT. These alternate buffers can be used to store things like notes, routines, or editing commands. Other files may be read into a buffer and searched without having to exit from the editor. EDT also provides for reading and writing other files while remaining in an editing session.
- Automatic backup - The editor always creates a new file for the user leaving the previous version intact.
- Macro capability - A powerful feature of the editor is the ability to create new commands that are made up of a series of editing commands. The new command is issued just like a standard editor command. This expands the functionality of the editor to whatever the user is able to create.

CREATING A TEXT FILE

Example 3-1 shows the procedure for creating a text file.

```
❶ $
  $EDIT/EDT NEW.TXT
  Input file does not exist
  [EOB]
❷ *I
                    TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
                    AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
                    ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.

                    UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
                    TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VAI\I\RIABLES THE
                    ORGANISM WILL DO AS IT DARN WELL PLEASES.
                    ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.

                    THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
                    TO THE LEVEL OF MANAGEMENT.
                    ^Z
  [EOB]
❸ *EXIT
❹ DR0:[305,303]NEW.TXT;1 11 lines
❺ $
```

Example 3-1 Creating a Text File

Notes on Example 3-1

The following comments are keyed to the example.

❶ DCL command EDIT used to invoke an editor. /EDT qualifier tells which editor to use. The file name NEW.TXT is supplied. EDT responds by indicating that the file does not exist and displays the "end of buffer" ([EOB]) symbol to indicate an empty buffer.

❷ (*) is EDT's prompt indicating that it is ready to accept commands. EDT, by default, begins in line mode. In line mode the cursor always remains at the left-most position of the display and only moves forward and backward line-by- line in the file. All operations are performed on a complete line. To enter lines of text issue the I[NSERT] command. The cursor moves over and waits for your input. Although not displayed here, EDT assigns line numbers to the lines of text that are entered. This input mode is terminated by typing <CTRL/Z>.

❸ EDT prompts (*) to indicate that it is again ready to accept commands. The EXIT command is issued to save the contents of the main buffer into a disk file.

4. EDT responds to indicate completion of the exit function by displaying the complete file specification, including the version number and number of lines saved in the file. Version numbers are octal numbers, beginning at 1 for a new file and increasing by 1 for every edit of that file.

5. EDT returns control to the system, and the operating system displays its prompt indicating readiness to accept another command.

EDITING AN EXISTING FILE

Example 3-2 shows the process of editing an existing file.

```
>
>
>
1 >EDIT/EDT NEW.TXT
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
2 *%WH
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
    3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
    4
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
    8           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    9
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   11           TO THE LEVEL OF MANAGEMENT.
[EOB]
*EXIT
3 DR0:[305,303]NEW.TXT;2 11 lines
>
>
>
```

Example 3-2 Editing an Existing File

Notes on Example 3-2

The numbered comments are keyed to the example.

1. The editor is invoked using the DCL command and qualifier specifying EDT. A file name is supplied referencing an existing file. The contents of that file are read into the EDT buffer.

❷ EDT prompts and a command is issued to display the complete buffer ending with [EOB] symbol. Line numbers are displayed which reference a particular line or range of lines in EDT commands. They are present only during the terminal session, and not saved in the output file. The exit command is issued to save another version of the file.

❸ EDT responds that the save operation is complete, and shows the new file specification and version number. A directory taken on NEW.TXT produces the following:

```
*
*DIR NEW.TXT;*

Directory DR0:[305,303]
6-JAN-82 22:39

NEW.TXT;1          2.          15-DEC-81 16:51
NEW.TXT;2          2.          06-JAN-82 22:39

Total of 4./7. blocks in 2. files
```

THE HELP FACILITY

Just as you have a HELP function at the operating system level, EDT also has a HELP command to provide information about the commands available to you in line mode. Issuing the HELP command gives you a list of all the commands for which information is provided. To obtain help for a command or a subtopic of a command , type HELP along with the command name and subtopic as shown in the second and third examples below. Example 3-3 shows how to access the HELP command.

Command Format

```
* HELP<RET>

* HELP COPY<RET>

* HELP COPY/QUERY<RET>
```

```
>EDIT/EDT NEW.TXT
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
*HELP
1 HELP
You can get help on a topic by typing

        HELP topic subtopic subsubtopic...

A topic can have one of the following forms:

        1. An alphanumeric string (e.g. a command name, option, etc.)
        2. The match-all or wild card symbol (*)

        Examples:       HELP SUBSTITUTE NEXT
                        HELP CHANGE SUBCOMMAND
                        HELP CH

If a topic is abbreviated, the text for all topics which match the
abbreviation is displayed.


Additional information available:
KEYPAD          CLEAR           CHANGE          COPY            DEFINE
DELETE          EXIT            FILL            FIND            INCLUDE
INSERT          JOURNAL         MOVE            PRINT           QUIT
REPLACE         RESEQUENCE      RANGE           SET             SHOW
SUBSTITUTE      TABS            TYPE            WRITE
*HELP COPY
The COPY (abbreviation: CO) command copies text from one location to
another within a buffer or between buffers.  When text is copied,
the source text remains intact.

Format:         COPY [range-1] TO [range-2] [/QUERY] [/DUPLICATE:n]

The lines specified by range-1 are copied in front of the first line
specified  by range-2.  Either range defaults to the current line.

If the destination is not the current buffer, put the name of the
receiving  buffer immediately after TO (=buffer). Give the
full name of  the buffer.  To copy text from an external file, see
INCLUDE.


Additional information available:
/DUPLICATE      /QUERY
*HELP COPY /QUERY
Format:         /QUERY

When you use the /QUERY (abbreviation: /Q) qualifier, EDT prompts you
with a '?' to verify each line to be copied.

Responses are:

        Y   Yes, copy this line
        N   No, do not copy this line
        Q   Quit, do not copy any of the rest of the lines
        A   All, copy all the rest of the lines


*EXIT
DR0:[305,303]NEW.TXT;3 11 lines
>
```

Example 3-3 HELP Command

EDITING COMMANDS USING LINE MODE

Once you have entered some text into the EDT buffer, you can issue commands that alter the contents of the buffer. Such commands allow you to copy lines, move them, delete them or change the text within the line. Table 3-1 lists some of the commands available in line mode.

Table 3-1 Line Mode Commands

User Needs	EDT Command To Use
Information on EDT commands and syntax	HELP
To display lines on terminal	T[YPE]
To create a file or insert lines in an existing file	I[NSERT]
To delete lines	D[ELETE]
To duplicate lines within a file	CO[PY]
To change the location of some lines	M[OVE]
To change a text string within a line	S[UBSTITUTE]
To renumber line numbers	RES[EQUENCE]
To change from line mode to character mode	C[HANGE]
To execute a sequence of EDT commands contained within the text buffer	DEFINE MACRO
Information on buffers, terminal settings	SH[OW]
To terminate EDT session without saving contents of the main buffer	QUIT
To terminate EDT session by saving contents of the main buffer	EX[IT]

COMMAND MODE COMMANDS

Displaying Lines

The TYPE command is used to display lines of text at the terminal. It takes as a parameter a range of line numbers to be displayed. This is the default command for the editor, so if you only supply a range of lines, EDT assumes that you mean the TYPE command. This is generally the way the type command is used. Table 3-2 shows acceptable range specifications. A convenient feature in specifying a range of lines is the ability to mix numerics and mnemonics. For example, the following is an acceptable range for specifying lines 5 through the end of the file: 5:E.

The TYPE command also changes the current line pointer. If you are at line 1 and type line 10, when it finishes typing, your current line is line 10. If you type 2:5, when it finishes typing, your current line is line 2. Example 3-4 shows how to use the TYPE command.

Command Format

1. EDT command name to type lines on the terminal; everything within brackets is optional.

2. Range of lines to be typed in the general form:

   FIRSTLINE:LASTLINE.

3. Command line terminator.

Table 3-2 Range Specifications

User Wants to Indicate	Range Specification
The current line	.
An explicit single line number	5
Lines 5 through 10	5:10
5 lines beginning with line 5	5#5
A single line which is the current line plus three lines	+3
A single line equal to the current line minus three lines	-3
A range of lines from the current line through the current line plus 10	.:+10
The whole buffer; first line through the last line	WH[OLE]
The first line of the buffer	BE[GIN]
The last line of the buffer	E[ND]
All lines before the current line	BEF[ORE]
All lines after the current line	R[EST]

```
>EDIT/EDT NEW.TXT
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
*TYPE 5:10
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
    8           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    9
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
*T 9
    9
*2#5
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
    3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
    4
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
*%WH
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
    3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
    4
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
    8           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    9
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   11           TO THE LEVEL OF MANAGEMENT.
[EOB]
*6
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
*-4
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
*+8
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
*.
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
*-5
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
*+2
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
*.
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
*.-2
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
*6
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
*EXIT
DR0:[305,303]NEW.TXT;4 11 lines
>
>
>
>
```

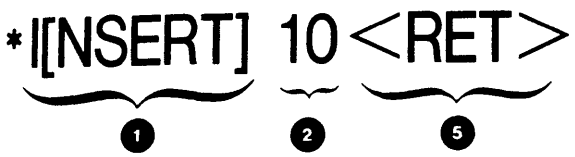
Example 3-4 TYPE Command

Inserting Lines

When you are creating a new file or adding to an existing file, the INSERT command is used to add lines of text. INSERT takes as a qualifier the range of where the text is to be added. The editor inserts the added text before the specified line number. When adding multiple lines after INSERT is entered, the cursor moves over two tab stops and waits for the lines of text to be typed. To terminate the input process, a <CTRL/Z> must be typed. When the (*) prompt is displayed, EDT is ready to accept another command.

When inputting a single line of text, the line of text is added to the command line (see the second example below) when the insert command is issued. You need not type <CTRL/Z>. After inserting the text line, EDT prompts for more commands. It does not wait for further input lines. Example 3-5 shows how to use the INSERT command.

Command Format



1. EDT command to insert text
2. Line number range
3. Delimiter
4. Line of text to be entered
5. Command line terminator

```
*I 10
                A MEETING IS AN EVENT AT WHICH THE MINUTES ARE KEPT
                AND THE HOURS ARE LOST.
                ^Z
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
*8:12
    8           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    9
    9.1         A MEETING IS AN EVENT AT WHICH THE MINUTES ARE KEPT
    9.2         AND THE HOURS ARE LOST.
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   11           TO THE LEVEL OF MANAGEMENT.
*I 10;ALL'S WELL THAT ENDS.
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
*8:12
    8           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    9
    9.1         A MEETING IS AN EVENT AT WHICH THE MINUTES ARE KEPT
    9.2         AND THE HOURS ARE LOST.
    9.3         ALL'S WELL THAT ENDS.
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   11           TO THE LEVEL OF MANAGEMENT.
*%WH
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
    3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
    4
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
    8           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    9
    9.1         A MEETING IS AN EVENT AT WHICH THE MINUTES ARE KEPT
    9.2         AND THE HOURS ARE LOST.
    9.3         ALL'S WELL THAT ENDS.
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   11           TO THE LEVEL OF MANAGEMENT.
[EOB]
```

Example 3-5 INSERT Command

LEARNING ACTIVITIES

1. READ the following in the EDT Reference Manual:

   - Chapter 1, Introduction to EDT
   - Chapter 2, Sample Editing Sessions - Getting Started and Line Editing Sections
   - Chapter 6, Line Numbers, Text Buffers and Ranges
   - Chapter 7, Line Editing - Insert command

2. DO the following at your terminal:

   a. Create a file with the name TEST.TXT. Insert the line "The quick brown fox jumped over the lazy dog's back." three times.

   b. Exit from the editor, saving the lines in a file.

   c. Reedit the file and issue the HELP command.

   d. Obtain HELP on TYPE.

   e. Display the lines in your file by issuing the following command:

      TYPE W

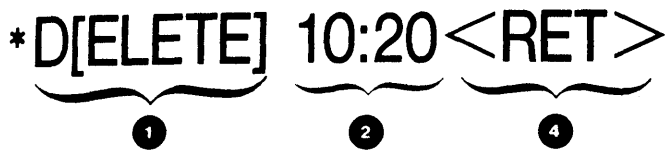
      The line numbers will appear with the lines of text.

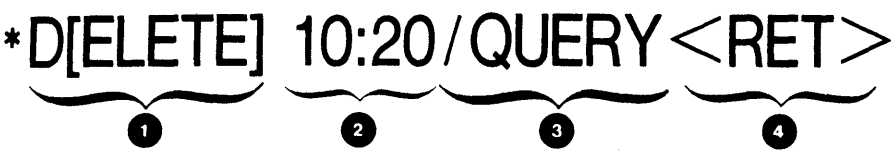
   f. Exit from the editor.

Deleting Lines

To remove lines from the text buffer use the DELETE command. The QUERY qualifier enables the user to selectively delete lines within a range of lines. When this form of the command is used, the editor displays the first line in the range of lines and then waits for the user to respond with "Y" for yes (meaning to delete the line), "N" for no, "A" for all (delete all remaining lines in the range) or "Q" for quit (do not delete any of the remaining lines in the range). Example 3-6 shows how to use the DELETE command.

Command Format





1. EDT command to delete lines
2. Range specification
3. Command qualifier
4. Command line terminator

```
*
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
*D 8
1 line deleted
    9
*D 9.1;9.4
3 lines deleted
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
*%WH
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
    3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
    4
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
    9
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   11           TO THE LEVEL OF MANAGEMENT.
[EOB]


* D9;10 /Q
    9
?N
    9.1         THE ATTENTION SPAN OF A COMPUTER IS ONLY AS LONG AS ITS ELECTRIC
AL
?N
    9.2         CORD.
?N
    9.3         THE ATTENTION SPAN OF A COMPUTER IS ONLY AS LONG AS ITS ELECTRIC
AL
?Y
    9.4         CORD.
?Y
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
?N
2 lines deleted
   11           TO THE LEVEL OF MANAGEMENT.
*9;11
    9
    9.1         THE ATTENTION SPAN OF A COMPUTER IS ONLY AS LONG AS ITS ELECTRIC
AL
    9.2         CORD.
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   11           TO THE LEVEL OF MANAGEMENT.
```

Example 3-6 DELETE Command

Copying Lines

To copy a range of lines to another section of the file, use the COPY command. The original lines remain intact and a duplicate set of lines is inserted at the new location. The copy command also allows you to selectively copy lines within a range of lines (as shown in the second command example below) and also to make more than one copy of the lines (as shown in the third command example). Example 3-7 shows how to use the COPY command.

Command Format

*CO[PY] 1:4 TO E <RET>

*CO[PY] 1:4 TO 13/Q<RET>

*CO[PY] 1:4 TO 13/DUPLICATE:2<RET>

```
*CO 1:4 TO E
4 lines copied
*%WH
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
    3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
    4
    4.1         ALL'S WELL THAT ENDS.
    4.2
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
    8           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    9
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   11           TO THE LEVEL OF MANAGEMENT.
   12           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
   13           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
   14           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
   15
[EOB]
*EX
DR0:[305,303]NEW.TXT;5 17 lines
```

Example 3-7 COPY Command

Moving Lines

To change the physical location of a line or group of lines, use the MOVE command. This command moves the lines defined by the first range specified, to the location preceding the line specified by the second range. In sheet one of Example 3-8, the MOVE command is issued to move lines 1 through 4 to line 2Ø. Notice that when the complete buffer is displayed, lines 1 through 4 no longer precede line 5 and are now positioned before line 2Ø. They now have line numbers 19.1 through 19.4.

Command Format

*M[OVE] 1:4 TO 20<RET>

*M[OVE] 1:6 TO 21/Q<RET>

```
*%wh
     1          TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
     2          AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
     3          ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
     4
     5          UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
     6          TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
     7          ORGANISM WILL DO AS IT DARN WELL PLEASES.
     8
     9          ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    10
    11          THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
    12          LONG AS ITS ELECTRICAL CORD.
    13
    14          THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
    15          TO THE LEVEL OF MANAGEMENT.
    16
    17          A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
    18          FINAL INSPECTION.
    19
    20
    21
[EOB]
*m 1:4 to 20
4 lines moved
*%wh
     5          UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
     6          TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
     7          ORGANISM WILL DO AS IT DARN WELL PLEASES.
     8
     9          ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    10
    11          THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
    12          LONG AS ITS ELECTRICAL CORD.
    13
    14          THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
    15          TO THE LEVEL OF MANAGEMENT.
    16
    17          A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
    18          FINAL INSPECTION.
    19
  19.1          TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
  19.2          AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
  19.3          ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
  19.4
    20
    21
[EOB]
```

Example 3-8 MOVE Command (Sheet 1 of 2)

```
*%wh
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
    3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
    4
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
    8
    9           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
   10
   11           THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
   12           LONG AS ITS ELECTRICAL CORD.
   13
   14           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   15           TO THE LEVEL OF MANAGEMENT.
   16
   17           A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
   18           FINAL INSPECTION.
   19
   20
   21
[EOB]
*m 1:6 to 21 /a
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
?y
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
?y
    3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
?y
    4
?n
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
?n
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
?n
3 lines moved
*%wh
    4
    5           UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
    6           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
    7           ORGANISM WILL DO AS IT DARN WELL PLEASES.
    8
    9           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
   10
   11           THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
   12           LONG AS ITS ELECTRICAL CORD.
   13
   14           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   15           TO THE LEVEL OF MANAGEMENT.
   16
   17           A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
   18           FINAL INSPECTION.
   19
   20
 20.1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
 20.2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
 20.3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
   21
[EOB]
```

Example 3-8 MOVE Command (Sheet 2 of 2)

Text Substitution

When you wish to correct a typing error or a string of text within a line, you must use the SUBSTITUTE command. It is the only line mode command that will perform this function. The format of the command is shown below, and Example 3-9 shows the result of issuing the command. When you select a text string for correction, you must supply a unique character string for the editor to match, or unexpected results may occur. The Editor substitutes for every occurrence of the specified string, regardless of surrounding characters. In the second example below, sthea$16:19, the editor found a match in line 17 in the word "other", and made the substitution to give the word "oar". The result is logically correct, however, not exactly what was intended. The correct command would be s$ the $ a $16:19 (with spaces).

The format of the SUBSTITUTE command requires a delimiter to separate the fields of the command. In the first command example below, the slash is used to delimit the command name from the string to be substituted, and that from the replacement string. Any non-alphanumeric character can be used as a string delimiter provided you use the same delimiter in all places. An example of this is shown in the second command example. You use this feature when you want to remove a string of characters that contains a slash.

Command Format

```
*S/PROGRAM/TASK/5 <RET>

*S$THE$A$16:19<RET>
```

```
>EDIT/EDT NEW.TXT
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
*%WH
    1           TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
    2           AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
    3           ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
    4
    5           ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
    6
    7           THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
    8           LONG AS ITS ELECTRICAL CORD.
    9
   10           THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   11           TO THE LEVEL OF MANAGEMENT.
   12
   13           A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
   14           FINAL INSPECTION.
   15
   16           UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
   17           TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
   18           ORGANISM WILL DO AS IT DARN WELL PLEASES.
   19
[EOB]
*S/PROGRAM/TASK/5
    5           ANY GIVEN TASK, WHEN RUNNING IS OBSOLETE.
1 substitution
*S$THE$A$16:19
   16           UNDER A MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
   17           TEMPERATURE, VOLUME, HUMIDITY, AND OAR VARIABLES A
3 substitutions
*QUIT
```

Example 3-9 SUBSTITUTE Command

Renumbering Lines

During an active editor terminal session, it may be helpful to renumber the lines within the file. Although it is not possible to get line numbers out of sequence, they can become fractionalized and cumbersome to use. Issuing the RES command resequences the complete file beginning with 1 and incrementing by 1 for each line in the file. A range can be specified to resequence a particular section of the file. If the sequence qualifier is used, as in the second command example below, a starting sequence number and an increment number can be specified.

This command renumbers lines 1Ø through the end of the file. Line 1Ø becomes line 1ØØ and each line after that has a line number incremented by 1Ø. Line 11 becomes 11Ø, line 12 becomes 12Ø. In the command, the first parameter supplies the range of lines to be numbered. In the command qualifier, /SEQ, the first parameter is the starting line number and the next parameter is the increment value. Example 3-1Ø depicts using the RESEQUENCE command.

Command Format

```
*RES<RET>

*RES 10:E/SEQ:100:10
```

```
*RES 10:%E /SEQ:100:10
10 lines resequenced
*%WH
     1          TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
     2          AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
     3          ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
     4
     5          ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
     6
     7          THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
     8          LONG AS ITS ELECTRICAL CORD.
     9
   100          THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
   110          TO THE LEVEL OF MANAGEMENT.
   120
   130          A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
   140          FINAL INSPECTION.
   150
   160          UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
   170          TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
   180          ORGANISM WILL DO AS IT DARN WELL PLEASES.
   190
[EOB]
*RES
19 lines resequenced
*%WH
     1          TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
     2          AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
     3          ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
     4
     5          ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
     6
     7          THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
     8          LONG AS ITS ELECTRICAL CORD.
     9
    10          THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
    11          TO THE LEVEL OF MANAGEMENT.
    12
    13          A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
    14          FINAL INSPECTION.
    15
    16          UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
    17          TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
    18          ORGANISM WILL DO AS IT DARN WELL PLEASES.
    19
[EOB]
*EXIT
DB2:[305,303]NEW.TXT;6 19 lines
```

Example 3-10 RESEQUENCE Command

Ending the Edit Session

To terminate the editing session and to preserve the contents of the EDT buffer, use the EXIT command. A file is generated with the name supplied when you invoked EDT and a version number that is one higher than the input file version number. If you wish to give your file a different name than the one supplied at the beginning of your session, supply the file name along with the EXIT command, as shown in the second example below.

The QUIT command terminates the editing session without saving the buffer contents in the file. Use QUIT when you are just examining the contents of a file, or when you have inadvertently corrupted the contents of a file.

Command Format

*EXIT<RET>

*EXIT NEWER.TXT<RET>

*QUIT<RET>

LEARNING ACTIVITIES

1. READ Chapter 7, Line Editing, in the EDT Reference Manual.

2. DO Lab Exercises 1 and 2 for this module.

Character Mode Features

In contrast to line mode where the cursor is always positioned at the beginning of a line and operations are performed on a line-by-line basis, character mode allows cursor positioning anywhere in the text buffer, and operations can be performed on a character-by-character basis. EDT shows immediate results of a keystroke which creates a great deal of activity on the screen. For example, if you delete a word, the letters are removed and the line is adjusted to close up the space where the word resided.

This mode makes adding and deleting text within a line extremely simple. Using the keypad function keys, a complete word can be deleted with one keystroke. The following is a list of features that makes character mode so attractive.

- Free form cursor positioning
- Dynamic display of editing operations
- Keypad function keys that perform specified operations in one keystroke
- Instantaneous addition of text
- Help facility
- Cut and paste operation for moving blocks of text
- Ability to use line mode editing commands
- Ability to change between upper- and lowercase alphabetic characters
- Ability to select a range of lines to limit line width
- Ability to set tabs
- Ability to define key functions

THE KEYPADS

Figures 3-3 and 3-4 show the keypads of the VT100 and VT52 terminals and include a diagram showing each key's function. Most keys have two functions. To invoke the lower function, the gold key must be pressed first. For example, if you wish to employ the TOP function on a VT100, you would first press the gold key (PF1) and then the TOP (5) key.

VT100 Keypad Functions

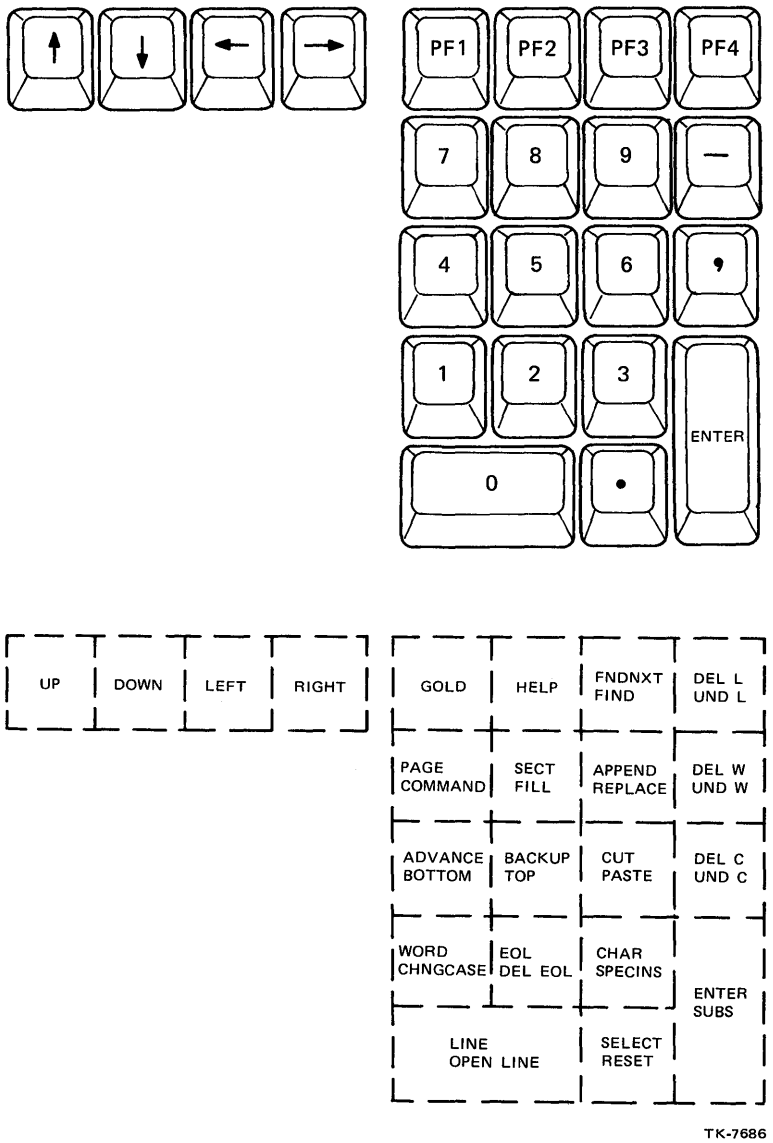


Figure 3-3 VT100 Keypad Functions

VT52 Keypad Functions

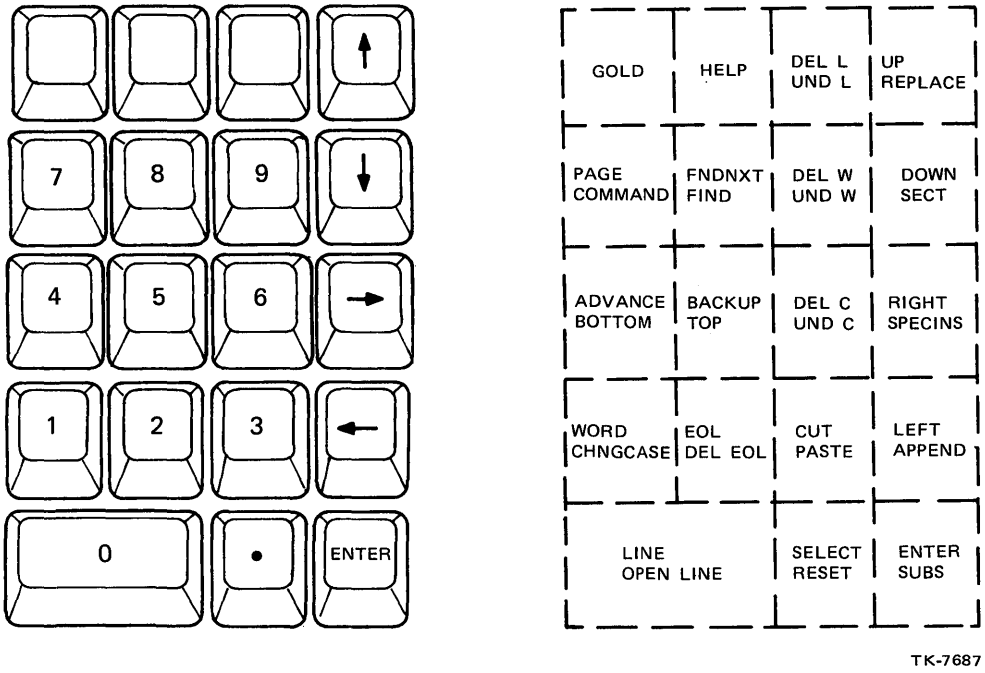


Figure 3-4 VT52 Keypad Functions

Changing to Character Mode

When you invoke EDT, the mode that is active is line mode. To be able to work in character mode, issue the change command (a line mode command) as shown in Figure 3-5. When this is done, EDT erases the screen and starts displaying the first 22 lines of the buffer. Line numbers are not displayed, as they have no function in character mode. All operations are now dependent upon cursor positioning. Also, EDT does not display a prompt in character mode to indicate that it is ready to accept commands. When the cursor sits on the first character and blinks, EDT is ready to accept keystrokes.

```
EDIT/EDT NEW.TXT
*%WH
 01   TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
 02   AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
 03   ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.
 04
 05   UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
 06   TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
 07   ORGANISM WILL DO AS IT DARN WELL PLEASES.
 08
 09   ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.
 10
 11   THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
 12   LONG AS ITS ELECTRICAL CORD.
 13
 14   THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
 15   TO THE LEVEL OF MANAGEMENT.
 16
 17   A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
 18   FINAL INSPECTION.
 19
*C
```

```
TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.

UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
ORGANISM WILL DO AS IT DARN WELL PLEASES.

ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.

THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
LONG AS ITS ELECTRICAL CORD.

THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
TO THE LEVEL OF MANAGEMENT.

A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
FINAL INSPECTION.
|EOB|
```

TK-7865

Figure 3-5 Changing to Character Mode

Positioning the Cursor

To position the cursor a character or a line at a time, the arrow keys on the keypad can be used, as shown in Figure 3-6.

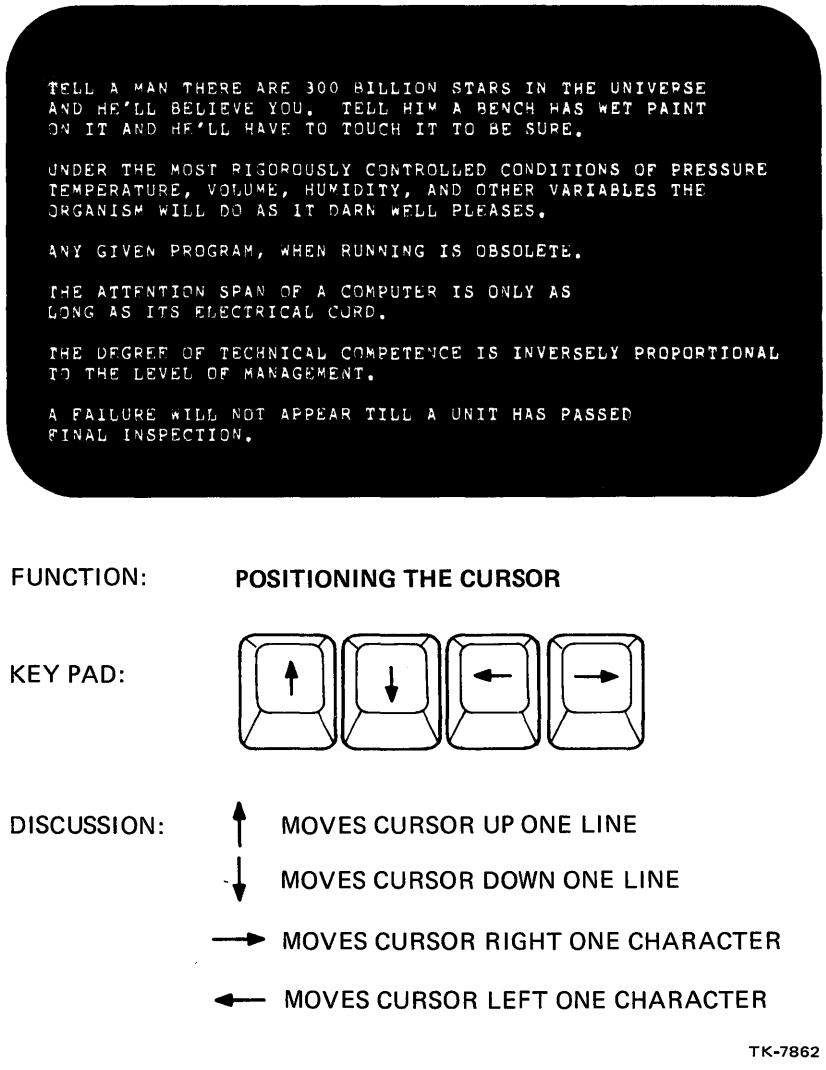
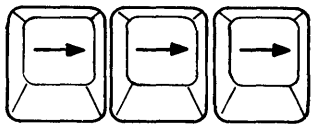
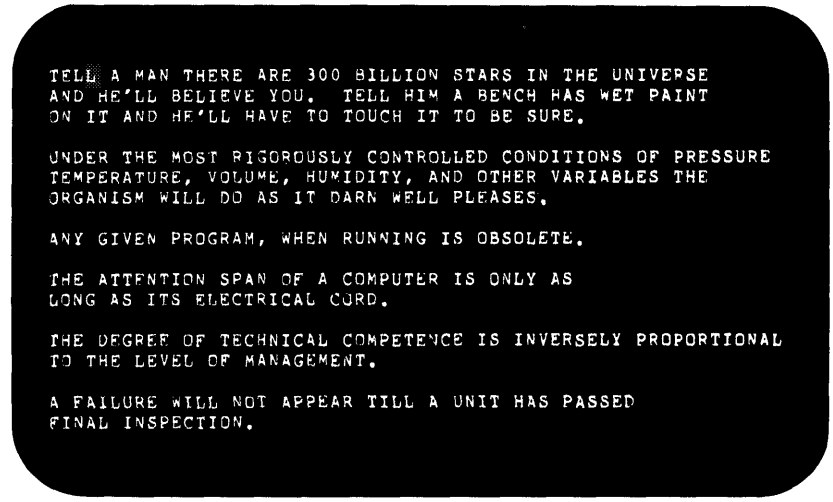


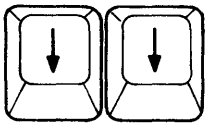
Figure 3-6 Positioning the Cursor (Sheet 1 of 3)

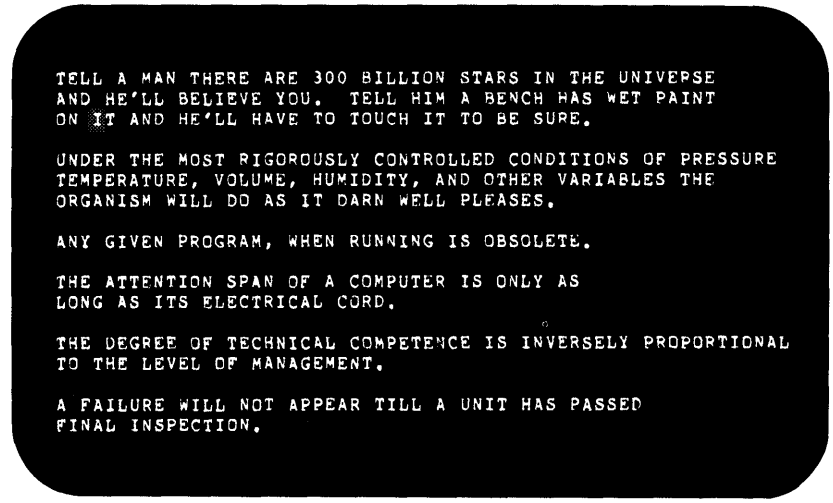
EXAMPLES:

USER TYPES **RESULTS**







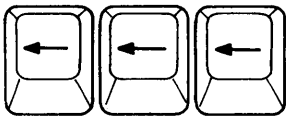


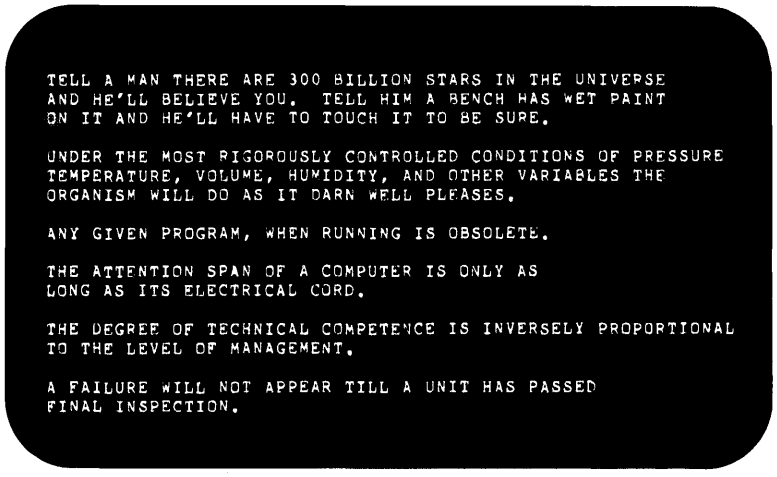
TK-7863

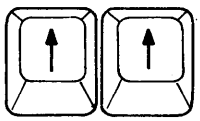
Figure 3-6 Positioning the Cursor (Sheet 2 of 3)

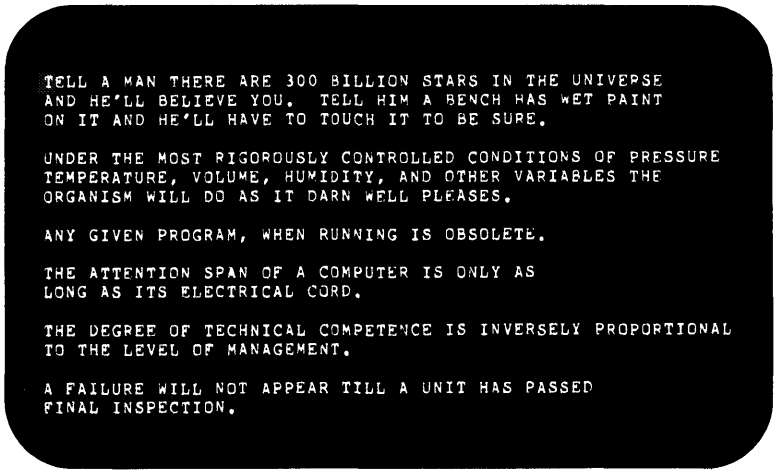
EXAMPLES: (CONT)

USER TYPES **RESULTS**







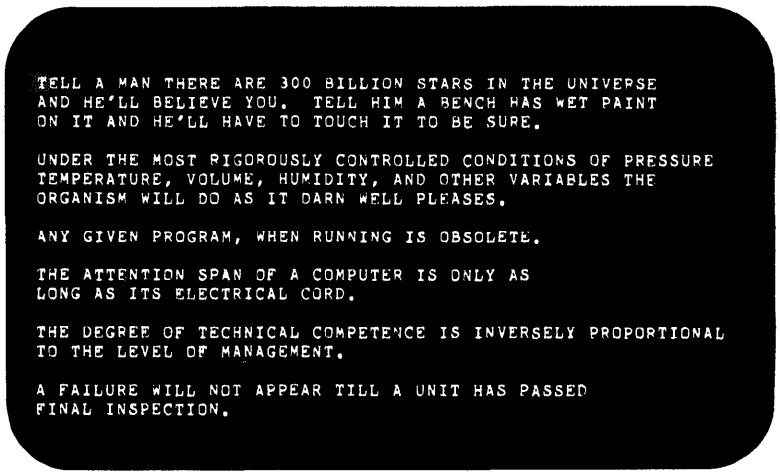


TK-7864

Figure 3-6 Positioning the Cursor (Sheet 3 of 3)

Positioning the Cursor (Alternate Methods)

Another method of moving the cursor (usually quicker) is to use the function keys in two key combinations, as shown in Figure 3-7.



FUNCTION: **POSITIONING THE CURSOR**

KEYPAD:

ADVANCE BOTTOM	BACKUP TOP	WORD CHNGCASE	EOL DEL EOL	LINE OPEN LINE

DISCUSSION:

ADVANCE – SETS CURSOR DIRECTION FORWARD (DEFAULT DIRECTION OF CURSOR). MOVEMENT REMAINS IN EFFECT UNTIL BACKUP IS DEPRESSED.

BACKUP – SETS CURSOR MOVEMENT TOWARD THE START OF THE FILE

TOP – POSITIONS CURSOR AT THE TOP OR START OF FILE.

BOTTOM – POSITIONS CURSOR AT THE BOTTOM OF THE FILE BUT BEFORE [EOB] SYMBOL.

WORD – MOVES THE CURSOR ONE WORD
WORD = ONE OR MORE CHARACTERS PRECEDED AND FOLLOWED BY KEY SPACES.

TK-7866

Figure 3-7 Positioning the Cursor (Alternate Methods) (Sheet 1 of 4)

EOL – POSITIONS CURSOR AT THE END OF THE LINE
LINE – POSITIONS CURSOR AT THE BEGINNING OF THE LINE

EXAMPLES:

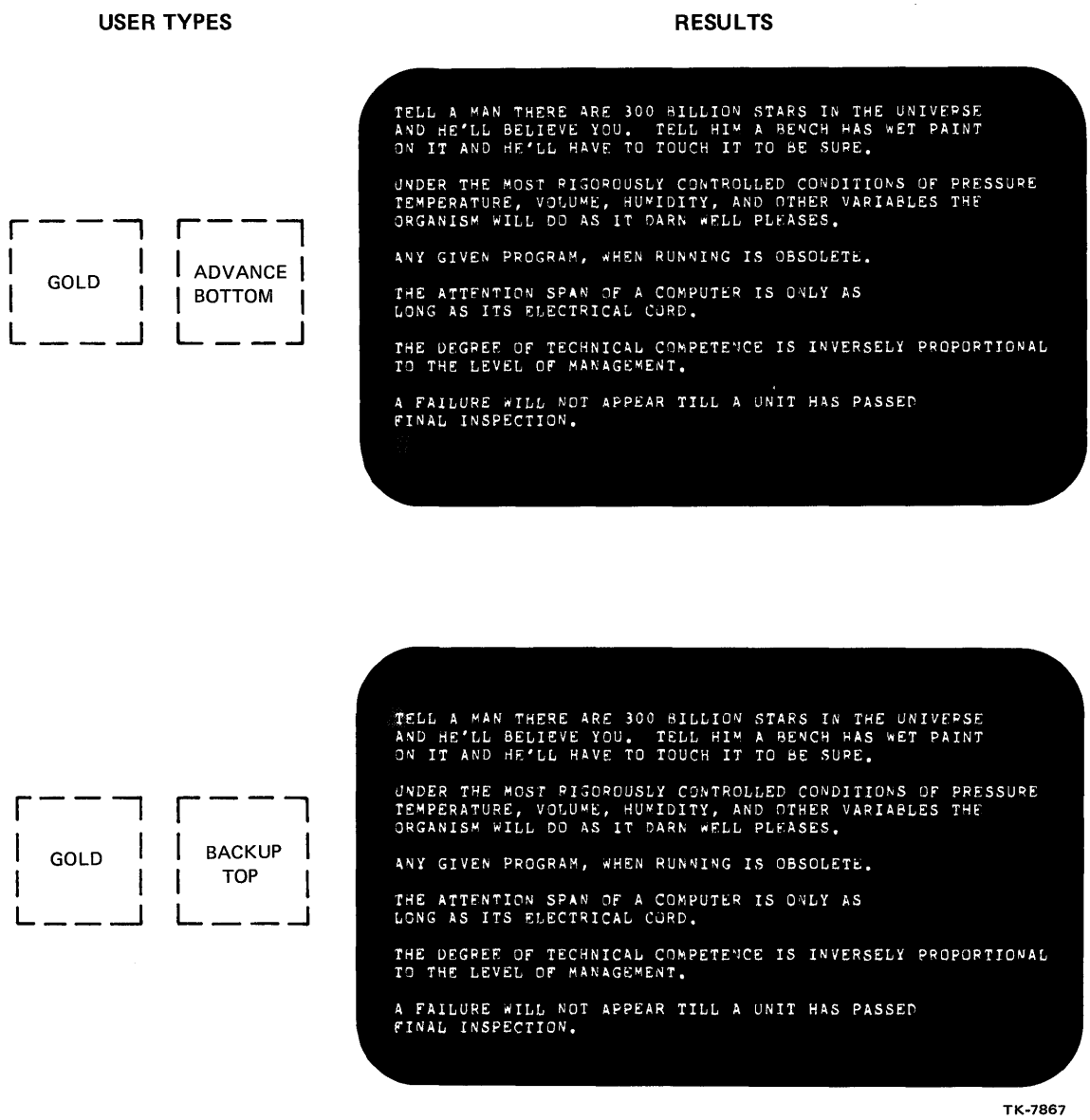


Figure 3-7 Positioning the Cursor (Alternate Methods) (Sheet 2 of 4)

EXAMPLES: (CONT)

USER TYPES

RESULTS

```
EOL
DEL EOL
```

```
TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.

UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
ORGANISM WILL DO AS IT DARN WELL PLEASES.

ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.

THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
LONG AS ITS ELECTRICAL CORD.

THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
TO THE LEVEL OF MANAGEMENT.

A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
FINAL INSPECTION.
```

```
LINE
OPEN LINE
```

```
TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.

UNDER THE MOST RIGOROUSLY CONTROLLED CONDITIONS OF PRESSURE
TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
ORGANISM WILL DO AS IT DARN WELL PLEASES.

ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.

THE ATTENTION SPAN OF A COMPUTER IS ONLY AS
LONG AS ITS ELECTRICAL CORD.

THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
TO THE LEVEL OF MANAGEMENT.

A FAILURE WILL NOT APPEAR TILL A UNIT HAS PASSED
FINAL INSPECTION.
```

TK-7868

Figure 3-7 Positioning the Cursor (Alternate Methods) (Sheet 3 of 4)

EXAMPLES: (CONT)

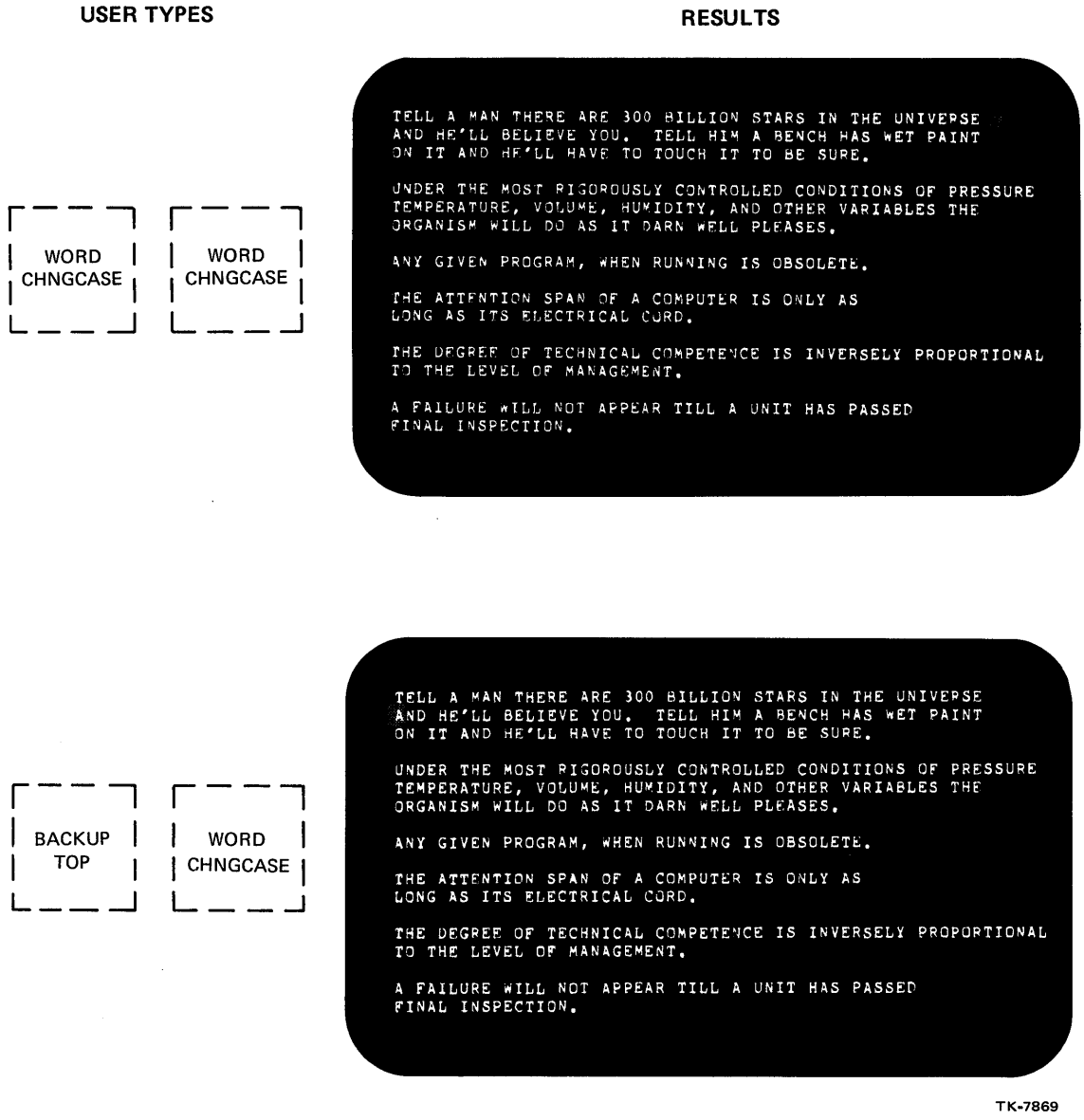
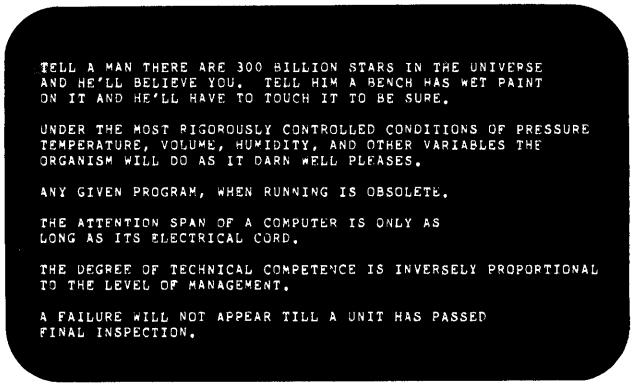


Figure 3-7 Positioning the Cursor (Alternate Methods) (Sheet 4 of 4)

Inserting Text

Inserting text in character mode is almost automatic. The only required operation is to position the cursor on the character in front of where you want to add the text. When the cursor is correctly positioned, you type the required text. This process is shown in Figure 3-8.



FUNCTION: **INSERTING TEXT**

KEYPAD: ANY KEYPAD KEY MAY BE USED TO POSITION THE CURSOR, ETC. HOWEVER, UNLIKE LINE MODE, THESE KEYS CAN NOT BE USED TO ENTER NUMERICS INTO A TEXT FILE.

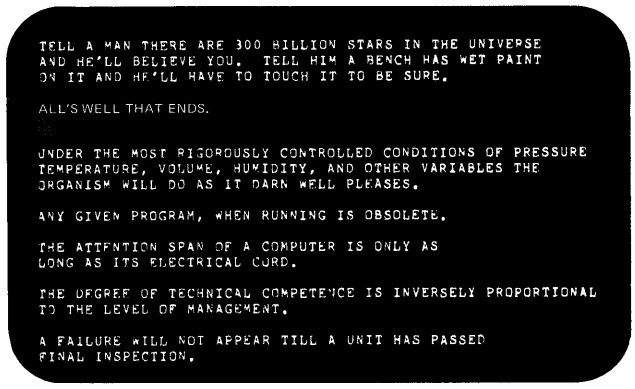
DISCUSSION: AFTER POSITIONING THE CURSOR TO THE PROPER POSITION, YOU CAN BEGIN TYPING THE DESIRED TEXT. IT WILL AUTOMATICALLY BE INSERTED INTO YOUR FILE IMMEDIATELY BEFORE THE CURSOR'S POSITION.

EXAMPLE:

USER TYPES

↓↓↓ <RET>
ALL'S WELL THAT ENDS <RET>

RESULTS



TK-7870

Figure 3-8 Inserting Text

Deleting Text

To delete text, first position the cursor to the line, character, or word that is to be deleted, and then press the appropriate delete key on the keypad. If a piece of text was deleted in error it can be restored by first pressing the GOLD key and then the appropriate undelete key (UND C, UND L, UND W). Figure 3-9 shows the process of deleting text.

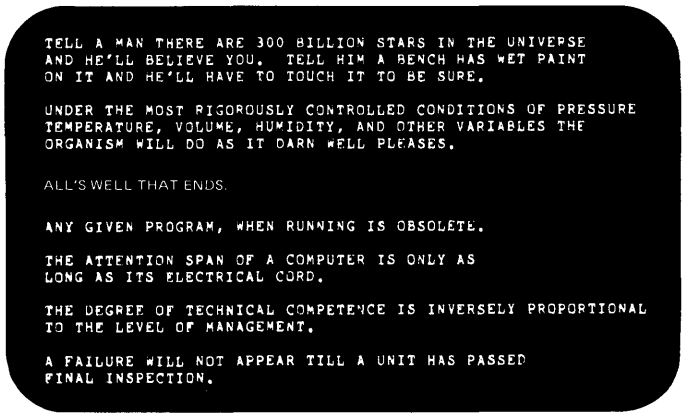


Figure 3-9 Deleting Text

Help Function

The Help key on the keypad provides the same function as typing help in line mode. It provides a keypad/function diagram and instructions on how to get help on each of the keys used in character mode (Figure 3-10).

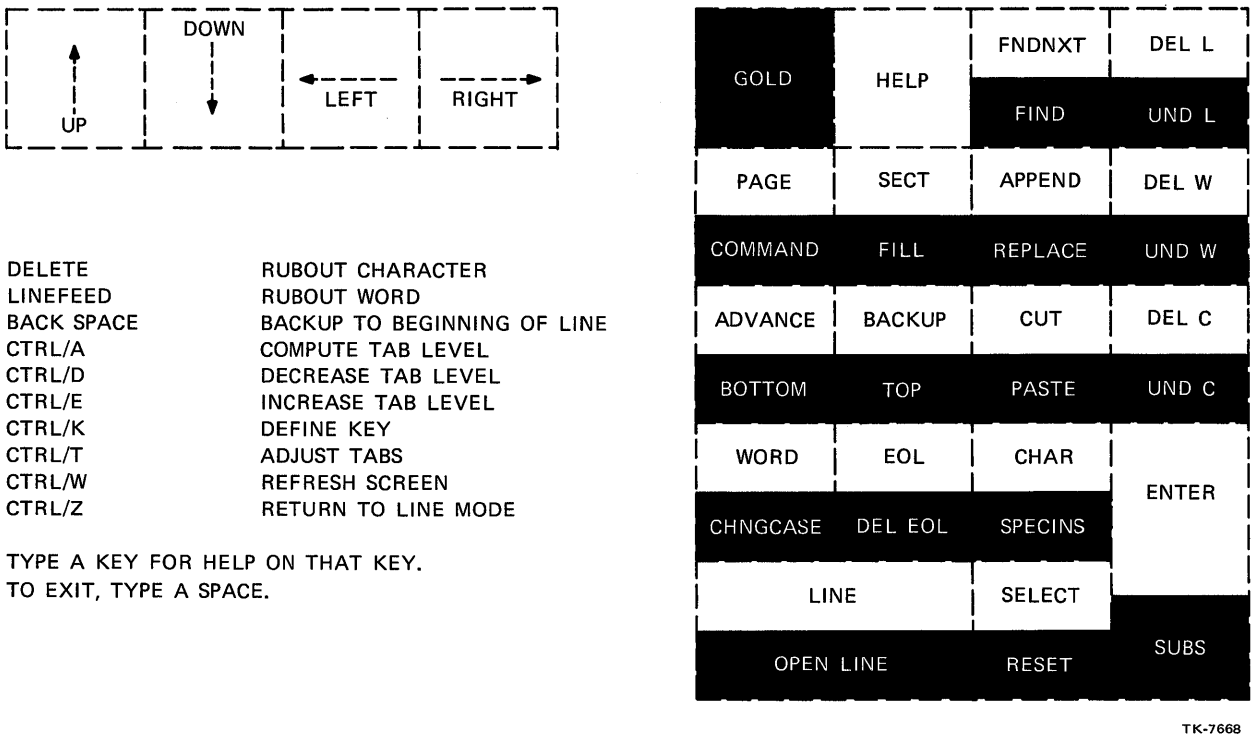


Figure 3-10 VT100 Keypad Editing Help Text

Exiting Character Mode

When your work in character mode is finished, you can save your edit session in either of two ways.

1. Press the GOLD and then the COMMAND function key. Type EXIT after the prompt and press the ENTER function key.

2. Press <CTRL/Z> which returns you to line editing. After the (*) prompt, type EXIT <RET>.

LEARNING ACTIVITIES

1. READ the following in the EDT Reference Manual:

   - Chapter 2, Sample Editing Sessions - Keypad Editing Section
   - Chapter 5, Keypad Editing

2. DO Lab Exercises 3 through 6 for this module.

FILE AND DIRECTORY MAINTENANCE

4

INTRODUCTION

Most work on the system, directly or indirectly, involves files. You will need to obtain files from other users and/or transfer your own files to them, and keep track of files that you have created. System commands are available to allow the user to maintain files and directories. Some available functions are:

- Listing directories of files
- Deleting unwanted files
- Renaming files
- Making copies of files
- Displaying files at the terminal
- Printing files on the system printer

This module focuses on these file operations.

OBJECTIVES

1. List directories of files.
2. Delete unwanted files.
3. Rename files.
4. Make copies of files.
5. Inspect the contents of files, using the terminal or a printer.
6. Set the access which other users have to your files.

RESOURCES

1. RSX-11M/M-PLUS Command Language Manual
2. Introduction to RSX-11M/M-PLUS

FILE STRUCTURE

FILES-11

Another facility of an operating system that benefits its use is the task of storing, maintaining and locating files on auxiliary storage devices. This is done by a software system called FILES-11. To be able to locate and maintain files on volumes mounted on disk drives, floppy disks and DECtapes, FILES-11 uses a two-level directory structure, as shown in Figure 4-1. For each volume, the structure consists of one Master File Directory (MFD) and one User File Directory (UFD) for each user on the volume.

Master File Directory (MFD)

The Master File Directory is a file located in the UFD [Ø,Ø] named ØØØØØØ.DIR. It contains a list of all the UFDs located on the volume.

User File Directory (UFD)

In turn, a User File Directory (UFD) (Figure 4-2) is a directory file that keeps track of all files on the volume belonging to a user. A UFD is created when an account is established for a user, and is identified with the user's UIC. For example, if your UIC is [3Ø5,3Ø3], you would have a directory file in UFD [Ø,Ø] named 3Ø53Ø3.DIR.

A UFD can also be created later by using the DCL command CREATE/DIRECTORY. This command is also used to create a UFD on another volume, or another UFD on the same volume. A user may have a UFD on many different devices, but one device is set up to be the default device for the user.

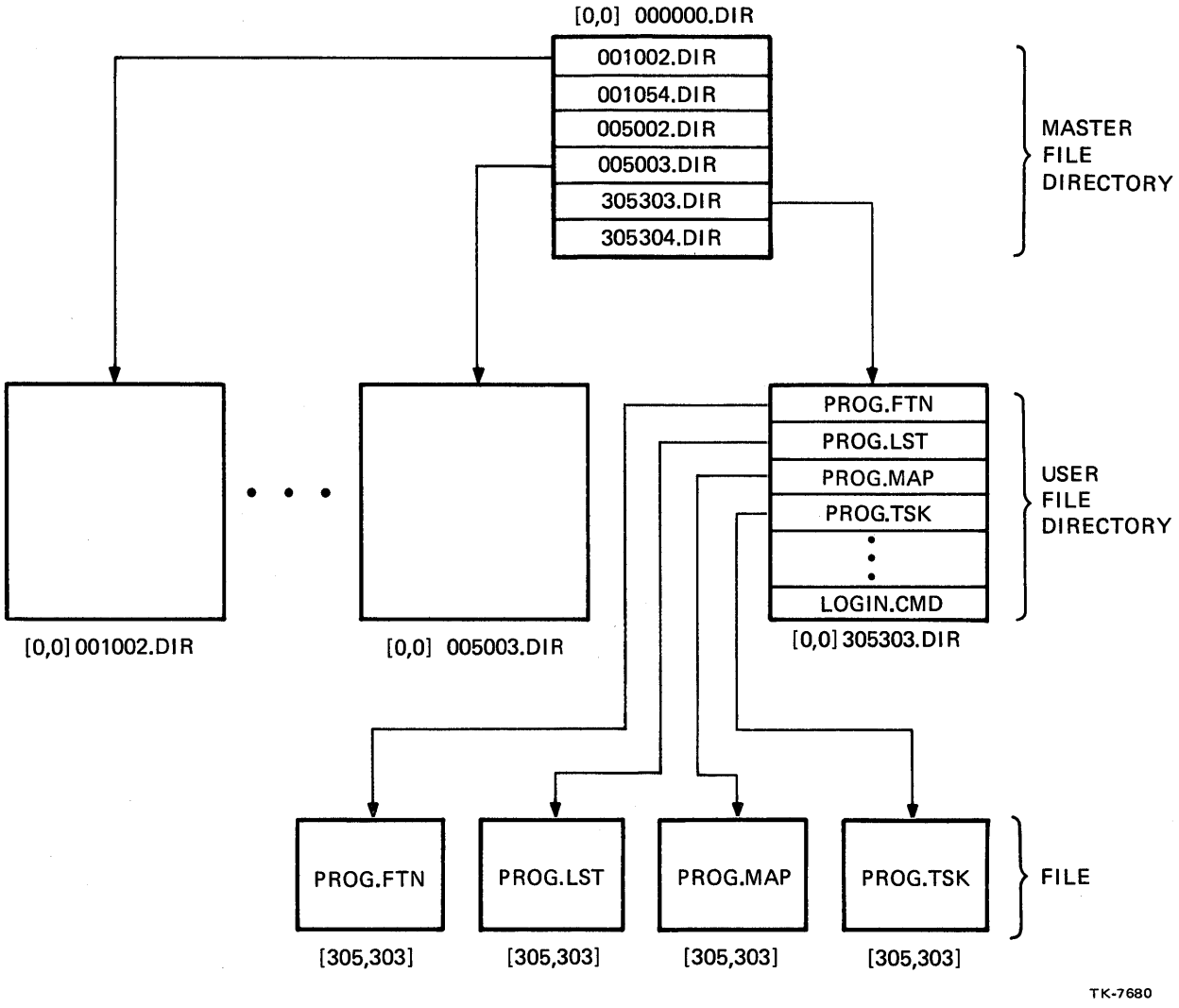


Figure 4-1 FILES-11 Directory Structure

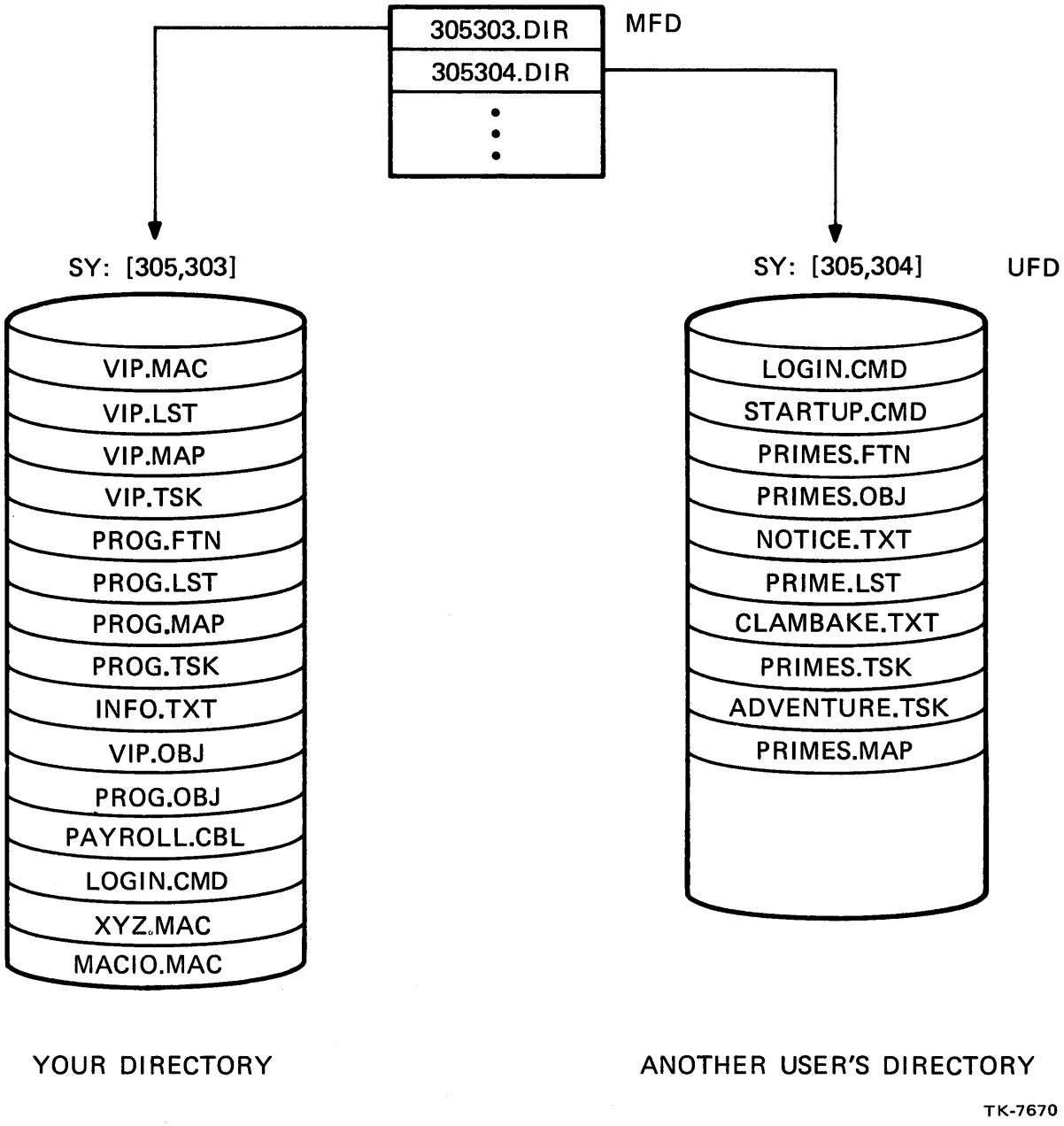


Figure 4-2 User File Directories

User Default UFD

When you successfully log in, the operating system grants you access to your User File Directory and all the files contained within it. This becomes your default UFD. Whenever a UFD is required syntax and is not supplied, the operating system will assume you mean your default UFD. It is possible and often to your advantage to change your default UFD to something other than the one you log in to. This is done using the DCL command SET DEFAULT, as shown in Figure 4-3. Example 4-1 shows the results of issuing the DIRECTORY command before and after using SET DEFAULT. The DIRECTORY command (which can be shortened to DIR) provides a listing of the files contained in a UFD. The first directory listing in the example is for UFD [305,303]. Because the UFD was not specified with the DIR command, the system assumes the user's default UFD. After issuing SET DEFAULT to change the default UFD to [305,304], DIR produces a listing for the second UFD.

User Default Device

It is also possible to change your default device. This is useful when you wish to work on a disk volume other than the default device set up for you when your account was established. By changing the default device, you will save having to specify the device every time you issue a command. SET DEFAULT is also used to change the default device. In both cases, the change is effective until the command is issued again or the user logs off.

System UFDs

Table 4-1 lists special UFDs established by the system manager when the operating system is generated. These UFDs contain files related to a particular subject. For example, the text files used to display HELP messages are stored in LB:[1,2]. The login message file is also stored in LB:[1,2].

Table B-3 in the RSX-11M/M-PLUS System Management Guide gives a complete list of the system UFDs.

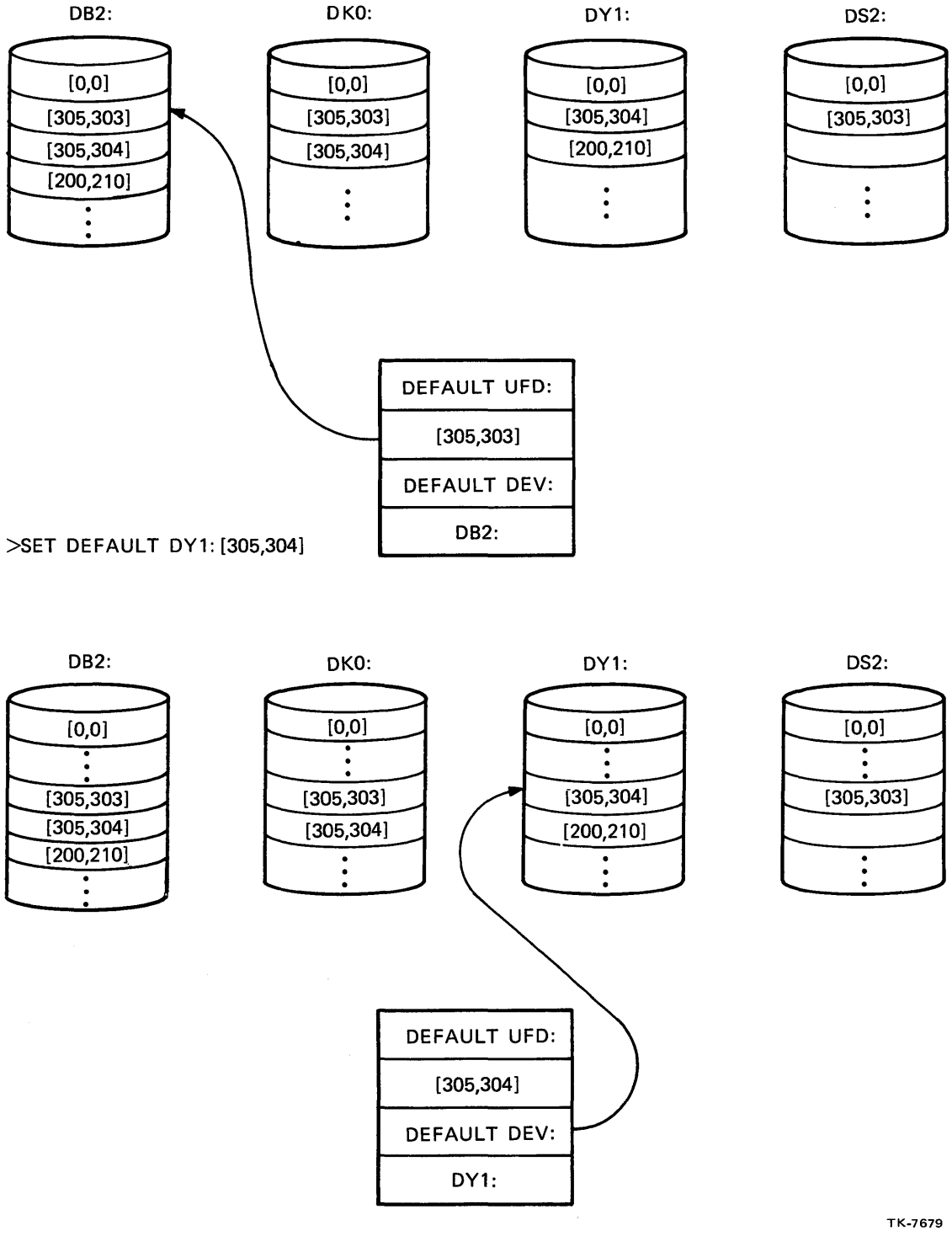


Figure 4-3 Changing the Default UFD and Device

```
>DIR


Directory DK2:[305,303]
15-DEC-81 11:39

LOGIN.CMD;1              1.          04-AUG-81 08:43
VIP.TSK;1                4.       C  14-AUG-81 10:48
PAYROLL.CBL;1            2.          08-JAN-81 15:31
INFO.TXT;1               60.         07-JUL-80 12:47
LOGOUT.CMD;5             1.          28-MAY-80 08:49
PROG.FTN;1               6.          14-AUG-81 08:35
PROG.OBJ;1               0.          14-AUG-81 08:36
PROG.LST;1               0.          14-AUG-81 08:36
PROG.OBJ;2               5.          14-AUG-81 08:36
PROG.LST;2               11.         14-AUG-81 08:36
PROG.FTN;2               6.          14-AUG-81 08:36
PROG.FTN;3               6.          14-AUG-81 08:37
PROG.TSK;1               6.       C  14-AUG-81 08:37
MACIO.MAC;5              3.          04-AUG-81 09:14
MACIO.OBJ;5              1.          14-AUG-81 10:46
MACIO.LST;1              5.          14-AUG-81 10:46
DIRECTORY.LST;1          3.          18-AUG-81 16:28
XYZ.MAC;1                3.          19-AUG-81 09:43
XYZ.TSK;1                4.       C  19-AUG-81 09:43
VIP.OBJ;3                1.          19-AUG-81 09:46
VIP.MAC;5                3.          19-AUG-81 09:46
VIP.OBJ;4                1.          19-AUG-81 09:47
VIP.OBJ;5                1.          19-AUG-81 09:47
VIP.LST;5                5.          19-AUG-81 09:47
MACIO.MAC;7              3.          19-AUG-81 09:48
MACIO.MAC;6              3.          19-AUG-81 09:48
MACIO.MAC;10             3.          19-AUG-81 09:49
PRIMES.TSK;1             46.      C  19-AUG-81 13:57

Total of 193./199. blocks in 28. files

>
>
>SET DEFAULT [305,304]
>DIR


Directory DK2:[305,304]
15-DEC-81 11:40

LOGIN.CMD;1              1.          04-AUG-81 08:43
LOGOUT.CMD;5             1.          28-MAY-80 08:49
STARTUP.CMD;57           14.         06-JUL-81 10:53
PRIMES.MAC;43            8.          15-JAN-81 16:15
ADVENTURE.TSK;1          151.     C  05-AUG-81 16:29
CLAMBAKE.TXT;1           3.          05-AUG-81 16:18
NOTICE.TXT;1             14.         16-MAR-81 10:34
PRIMES.OBJ;1             2.          14-AUG-81 11:09
PRIMES.LST;1             13.         14-AUG-81 11:09
PRIMES.TSK;1             46.      C  14-AUG-81 11:09
PRIMES.MAP;1             3.          14-AUG-81 11:09
A.MAC;2                  3.          18-AUG-81 16:30
VIP.TSK;1                4.       C  19-AUG-81 13:58
PROG.TSK;1               6.       C  18-AUG-81 16:31

Total of 269./273. blocks in 14. files
```

Example 4-1 How the Default UFD Works

Table 4-1 Special User File Directories

User Wants to Find	Look On
General System Files - Macro, System, FCS, Memory Management Libraries	LB:[1,1]
Help files and command files - Startup and Shutup Command Files, Login.TXT, MCR, DCL, Utility Help Files	LB:[1,2]
BATCH, PRINT Queues, Spooling Files	LB:[1,7]
Operating System Images (M and M-PLUS)	SY:[1,54]
Files associated with the Introduction to RSX-11M Manual	[200,1]
Nonprivileged System Utility Programs (M systems)	SY:[1,54]
Nonprivileged System Utility Programs (M-PLUS only)	SY:[6,54]

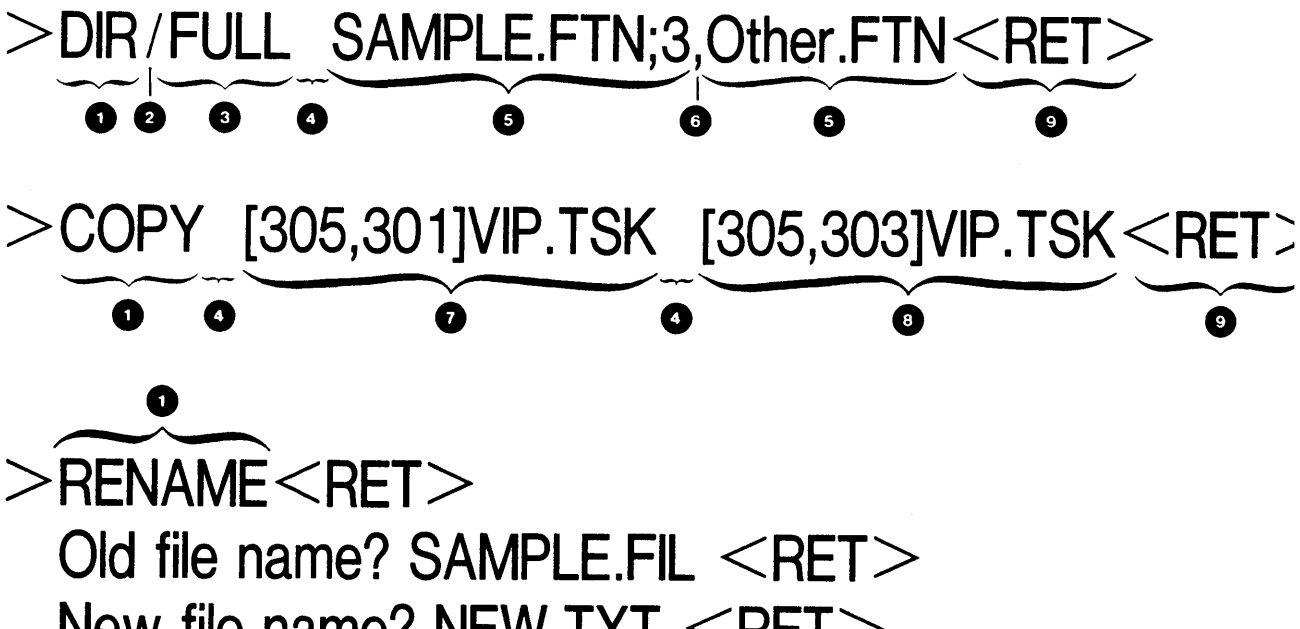
LEARNING ACTIVITY

1. READ Sections 4.1 through 4.1.4, in Chapter 4 of the RSX-11M/M-PLUS Command Language Manual.

MAINTAINING A USER FILE DIRECTORY

When your terminal session activity has been heavy, your UFD may contain unnecessary files. You may want copies of files from someone else's UFD, or to copy some of yours to another UFD. Although there is no limit to the number of files you may have in your UFD, there is a physical limit to the number of files that can exist on a volume. Therefore, some file maintenance must be performed to ensure that unneeded files are removed to free up disk space for everyone's use. Table 4-2 lists some of the DCL file maintenance commands that will maintain your UFD and other UFDs. The DCL command line is reviewed below.

Command Formats



1. DCL command name
2. qualifier delimiter
3. command qualifier
4. command delimiter
5. file specification
6. file specification list delimite
7. input file specification
8. output file specification
9. command line terminator

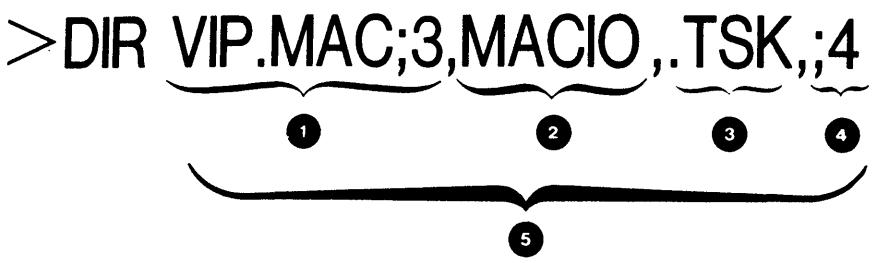
Table 4-2 File Maintenance Commands

User Wants To	Command To Use
List the files in a User File Directory	DIRECTORY
Delete files from a User File Directory	DELETE
Delete all versions except the latest version of a file	PURGE
Copy a file within a UFD or between UFDs	COPY
Append one file to another	COPY
Change the name of a file	RENAME
Display the contents of a file at the terminal	TYPE
Print a copy of a file on a line printer	PRINT
Merge	CONVERT

Defaults in File Specification

Defaults allow you to specify files without having to type in the full specification. Shown below is an example of using defaults to your advantage. The results of issuing the DIR command are shown in Example 4-2. Refer to Table 4-3 for the rules governing the use of defaults in a file specification.

Command Format



1. The first file specification is explicit on all fields.

2. The file type is implied, and the default taken. The default file type is .MAC from the previous file specification. The version number is implied; the default is always the latest version no matter what the previous file specification states.

3. The file name is implied, and the default taken. The default file name is MACIO from the previous file specification.

4. The file name and type are implied from the previous file specification. Therefore, this file specification equates to MACIO.TSK;4.

5. In all cases the device and UFD are implied, the default being the user's default device and UFD.

Table 4-3 Defaults in a File Specification

Field	First File Spec in Command	Next File Spec
Device	SY:	Last device specified
UFD	Default UFD	Last UFD specified
File name	None	Previous file spec
File type	None	Last type specified
Version	Highest version	Highest version

```
>
>DIR VIP.MAC;3,MACIO,,TSK,;4

Directory DK2:[305,303]
15-DEC-81 11:46

VIP.MAC;3          3.          19-AUG-81 09:46
MACIO.MAC;10       3.          19-AUG-81 09:49
MACIO.TSK;5        4.       C  15-DEC-81 11:43
MACIO.TSK;4        4.       C  15-DEC-81 11:43

Total of 14./18. blocks in 4. files
```

Example 4-2 Using Defaults in a File Specification

Listing Directories

When you wish to list the files that are in your directory, use the DIRECTORY command. It is also used to list files in other directories, to list a particular group of files (such as all MACRO source files), to search a group of directories for a particular file, to determine the amount of free space on a device, and to determine the amount of space used by files in a UFD. Example 4-3 shows the results of issuing the directory commands in the examples. Refer to the notes for an explanation of the results.

Command Format

```
>DIR

>DIR VIP.MAC

>DIR/FULL PAYROLL.CBL

>DIR [305,304]

>DIR/BRIEF

>DIR/SUMMARY

>DIR/PRINT

>DIR/OUTPUT:DIRECTORY.LST

>DIR/FREE DB0
```

```
(1) >DIR

(2) Directory DK2:[305,303]
    15-DEC-81 11:38
        (3)                  (4)              (5)      (6)
    LOGIN.CMD;1          1.              04-AUG-81 08:43
    VIP.TSK;1            4.          C   14-AUG-81 10:48
    PAYROLL.CBL;1        2.              08-JAN-81 15:31
    INFO.TXT;1           60.             07-JUL-80 12:47
    LOGOUT.CMD;5         1.              28-MAY-80 08:49
    PROG.FTN;1           6.              14-AUG-81 08:35
    PROG.OBJ;1           0.              14-AUG-81 08:36
    PROG.LST;1           0.              14-AUG-81 08:36
    PROG.OBJ;2           5.              14-AUG-81 08:36
    PROG.LST;2           11.             14-AUG-81 08:36
    PROG.FTN;2           6.              14-AUG-81 08:36
    PROG.FTN;3           6.          (7) 14-AUG-81 08:37
    PROG.TSK;1           6.          C   14-AUG-81 08:37
    MACIO.MAC;5          3.              04-AUG-81 09:14
    MACIO.OBJ;5          1.              14-AUG-81 10:46
    MACIO.LST;1          5.              14-AUG-81 10:46
    DIRECTORY.LST;1      3.              18-AUG-81 16:28
    XYZ.MAC;1            3.              19-AUG-81 09:43
    XYZ.TSK;1            4.          C   19-AUG-81 09:43
    VIP.OBJ;3            1.              19-AUG-81 09:46
    VIP.MAC;5            3.              19-AUG-81 09:46
    VIP.OBJ;4            1.              19-AUG-81 09:47
    VIP.OBJ;5            1.              19-AUG-81 09:47
    VIP.LST;5            5.              19-AUG-81 09:47
    MACIO.MAC;7          3.              19-AUG-81 09:48
    MACIO.MAC;6          3.              19-AUG-81 09:48
    MACIO.MAC;10         3.              19-AUG-81 09:49
    PRIMES.TSK;1         46.         C   19-AUG-81 13:57

(8) Total of 193./199. blocks in 28. files

(9) >DIR VIP.MAC;*

    Directory DK2:[305,303]
    15-DEC-81 11:45

    VIP.MAC;5            3.              19-AUG-81 09:46

    Total of 3./5. blocks in 1. file

(10) >
    >DIR/FULL PAYROLL.CBL

    Directory DK2:[305,303]
    15-DEC-81 11:47

                          (11)
    PAYROLL.CBL;1        (17,2)          2./2.              08-JAN-81 15:31
(12)  [305,303](RWED,RWED,RWED,R)    14-AUG-81 10:51(10.)
                    (13)
    Total of 2./2. blocks in 1. file

    >
```

Example 4-3 DIRECTORY Command Samples (Sheet 1 of 3)

```
(14) Directory DK2:[305,304]
15-DEC-81 11:48

LOGIN.CMD;1          1.          04-AUG-81 08:43
LOGOUT.CMD;5         1.          28-MAY-80 08:49
STARTUP.CMD;57       14.         06-JUL-81 10:53
PRIMES.MAC;43        8.          15-JAN-81 16:15
ADVENTURE.TSK;1      151.    C   05-AUG-81 16:29
CLAMBAKE.TXT;1       3.          05-AUG-81 16:18
NOTICE.TXT;1         14.         16-MAR-81 10:34
PRIMES.OBJ;1         2.          14-AUG-81 11:09
PRIMES.LST;1         13.         14-AUG-81 11:09
PRIMES.TSK;1         46.     C   14-AUG-81 11:09
PRIMES.MAP;1         3.          14-AUG-81 11:09
A.MAC;2              3.          18-AUG-81 16:30
VIP.TSK;1            4.      C   19-AUG-81 13:58
PROG.TSK;1           6.      C   18-AUG-81 16:31

Total of 269./273. blocks in 14. files

>
>
>
(15) >DIR/BRIEF

Directory DK2:[305,303]

LOGIN.CMD;1
MACIO.TSK;5
MACIO.TSK;4
VIP.MAC;3
VIP.TSK;1
PAYROLL.CBL;1
INFO.TXT;1
LOGOUT.CMD;5
PROG.FTN;1
PROG.OBJ;1
PROG.LST;1
PROG.OBJ;2
PROG.LST;2
PROG.FTN;2
PROG.FTN;3
PROG.TSK;1
MACIO.MAC;5
MACIO.OBJ;5
MACIO.LST;1
DIRECTORY.LST;1
XYZ.MAC;1
XYZ.TSK;1
VIP.OBJ;3
VIP.OBJ;4
VIP.OBJ;5
VIP.LST;5
MACIO.MAC;7
MACIO.MAC;6
MACIO.MAC;10
PRIMES.TSK;1
```

Example 4-3 DIRECTORY Command Samples (Sheet 2 of 3)

```
>
(16) >DIR/SUMMARY

Storage used/allocated for Directory DK2:[305,303]
15-DEC-81 11:48

Total of 201./207. blocks in 30. files

>
>
(17) >DIR/OUTPUT:DIRECTORY.LST
>DIR DIRECTORY.LST

Directory DK2:[305,303]
15-DEC-81 11:49

DIRECTORY.LST;2      4.          15-DEC-81 11:49

Total of 4./4. blocks in 1. file

>DIR/FREE

DK2: has 4091. blocks free, 709. blocks used out of 4800.
Largest contiguous space = 2242. blocks
239. file headers are free, 55. headers used out of 294.

>
>
>
(18) >DIR/FREE DB1:

DB1: has 55407. blocks free, 285263. blocks used out of 340670.
Largest contiguous space = 33634. blocks
13885. file headers are free, 11708. headers used out of 25593.
```

Example 4-3 DIRECTORY Command Samples (Sheet 3 of 3)

Notes on Example 4-3

1. Command to obtain a listing of files in the user's default device and UFD.
2. Device, UFD for which the directory, day, and time are given.
3. File name, type and version.
4. Size of the file in blocks (decimal).
5. Creation date.
6. Creation time.
7. C indicates file stored contiguously on the disk. L indicates file is locked (not properly closed). You will not be able to read this file.
8. Summary including the number of blocks used, the number of blocks allocated, and the number of files.
9. Directory on a specific file.
10. Example of full directory.
11. File ID.
12. Owner of the file.
13. Protection setting.
14. Obtaining a directory of another UFD.
15. Obtaining a quick directory of the user's UFD.
16. Obtaining only summary information.
17. Putting the directory information into a file.
18. Obtaining information on the volume.

Deleting Files

When you wish to remove a file from your UFD, you may do so by using the DELETE command. Care must be taken to ensure that the correct file specification has been supplied or you may delete a needed file. If a file is deleted by mistake, it may be possible to recover that file from the system backup tapes and disks. This depends upon when and if your system manager has performed a backup. Backing up a volume is copying its files to another disk, or to a magnetic tape for security purposes. The DELETE command allows you to selectively delete files from a UFD (as shown in the fourth command example below) and also to provide a log of the files that have been deleted (third command example).

Example 4-4 provides some samples of the DELETE command.

Command Format

```
>DEL MOV.DAT;*

>DEL A.*;*

>DEL/LOG  A*.OBJ;*,B.MAC;5

>DEL/QUERY  *.MAC;*
```

```
>DIR MACIO.TSK;*

Directory DK2:[305,303]
15-DEC-81 11:51

MACIO.TSK;5          4.       C  15-DEC-81 11:43
MACIO.TSK;4          4.       C  15-DEC-81 11:43
MACIO.TSK;1          4.       C  15-DEC-81 11:50
MACIO.TSK;3          4.       C  15-DEC-81 11:50
MACIO.TSK;2          4.       C  15-DEC-81 11:50

Total of 20./20. blocks in 5. files

>
>DEL/LOG MACIO.TSK;*

The following files have been deleted:
DK2:[305,303]MACIO.TSK;5
DK2:[305,303]MACIO.TSK;4
DK2:[305,303]MACIO.TSK;1
DK2:[305,303]MACIO.TSK;3
DK2:[305,303]MACIO.TSK;2
>
>
>DIR MACIO.*;*

Directory DK2:[305,303]
15-DEC-81 11:52

MACIO.MAC;5          3.          04-AUG-81 09:14
MACIO.OBJ;5          1.          14-AUG-81 10:46
MACIO.LST;1          5.          14-AUG-81 10:46
MACIO.MAC;7          3.          19-AUG-81 09:48
MACIO.MAC;6          3.          19-AUG-81 09:48
MACIO.MAC;10         3.          19-AUG-81 09:49

Total of 18./22. blocks in 6. files

>DEL *.LST;*
>DIR *.LST;*
DIR -- No such file(s)
```

Example 4-4 DELETE Command Samples

Purging Old Files

To ensure that at least one version of a file is always retained in your UFD after deleting files, use the PURGE command. This command offers the option to save a variable number of versions of a file. This ability is especially useful after a series of editing sessions, to clean up unwanted files. It is also useful to purge all files in your UFD at the end of the day. This frees up storage space on the volume and makes it available for all to use.

Example 4-5 shows how to purge files.

Command Format

```
>PURGE  VIP.MAC

>PURGE/LOG  A.MAC,B.*

>PURGE/KEEP:3  C.LST
```

```
>
>DIR VIP.MAC;*

Directory DK2:[305,303]
15-DEC-81 11:54

VIP.MAC;1            3.          15-DEC-81 11:53
VIP.MAC;2            3.          15-DEC-81 11:53
VIP.MAC;3            3.          19-AUG-81 09:46
VIP.MAC;4            3.          15-DEC-81 11:53
VIP.MAC;5            3.          15-DEC-81 11:53

Total of 15./17. blocks in 5. files

>
(1) >PURGE VIP.MAC
>DIR VIP.MAC;*

Directory DK2:[305,303]
15-DEC-81 11:58

VIP.MAC;5            3.          15-DEC-81 11:53

Total of 3./3. blocks in 1. file

>DIR VIP.LST;*

Directory DK2:[305,303]
15-DEC-81 12:01

VIP.LST;1            5.          15-DEC-81 11:59
VIP.LST;2            5.          15-DEC-81 11:59
VIP.LST;3            5.          15-DEC-81 12:00
VIP.LST;4            5.          15-DEC-81 12:00
VIP.LST;5            5.          15-DEC-81 12:00

Total of 25./25. blocks in 5. files

(2) >purge/log vip.lst

The following files have been deleted:
DK2:[305,303]VIP.LST;1
DK2:[305,303]VIP.LST;2
DK2:[305,303]VIP.LST;3
DK2:[305,303]VIP.LST;4
>
>
>
```

Example 4-5 Purging Files (Sheet 1 of 2)

```
>dir vip.obj;*

❸ Directory DK2:[305,303]
15-DEC-81 12:02

VIP.OBJ;6              1.          15-DEC-81 11:59
VIP.OBJ;7              1.          15-DEC-81 11:59
VIP.OBJ;10             1.          15-DEC-81 11:59
VIP.OBJ;11             1.          15-DEC-81 11:59
VIP.OBJ;12             1.          15-DEC-81 12:00
VIP.OBJ;13             1.          15-DEC-81 12:00
VIP.OBJ;14             1.          15-DEC-81 12:00
VIP.OBJ;3              1.          19-AUG-81 09:46
VIP.OBJ;4              1.          19-AUG-81 09:47
VIP.OBJ;5              1.          19-AUG-81 09:47

Total of 10./10. blocks in 10. files

>
>
>
>
>purge/keep:3 vip.obj
>dir vip.obj;*


Directory DK2:[305,303]
15-DEC-81 12:02

VIP.OBJ;12             1.          15-DEC-81 12:00
VIP.OBJ;13             1.          15-DEC-81 12:00
VIP.OBJ;14             1.          15-DEC-81 12:00

Total of 3./3. blocks in 3. files
```

Example 4-5 Purging Files (Sheet 2 of 2)

Notes on Example 4-5

❶ Directory of VIP.MAC before and after using the purge command. The latest version is retained.

❷ Example using /LOG qualifier on the purge command to report deleted files.

❸ Example using the /KEEP qualifier of the purge command, which in this case retained the latest three versions of file VIP.OBJ.

Copying Files

The COPY command is used to make a copy of a file. In addition to creating a copy, the command makes the necessary entries into the directory structure so that the file can be located at a future time. The command can be used to copy files within a UFD, between UFDs on the same device, and between devices. It is also used to merge multiple files into one file.

Command Format

```
>COPY VIP.MAC [305,304]VIP.MAC

>COPY VIP.TSK,PROG.TSK [305,304]*.*

>COPY
  From? VIP.MAC
  To? [305,304]*.*
```

```
>copy vip.mac [305,304]vip.mac
>

>dir [305,304]vip.mac

Directory DK2:[305,304]
15-DEC-81 12:03

VIP.MAC;1           3.          15-DEC-81 12:03

Total of 3./3. blocks in 1. file


>copy
From? vip.tsk,prog.tsk
To? [305,304]*.*
>
>
```

Example 4-6 COPY Command

Renaming Files

The RENAME command is used to change the name, type or version of a file. It does not make a new copy of the file; it simple alters the directory entry.

Command Format

```
>RENAME A.MAC B.MAC

>RENAME A.MAC A.MAC;2

>RENAME A.* B.*
```

```
>
>RENAME VIP.MAC A.MAC
>DIR

Directory DK2:[305,304]
15-DEC-81 12:08

LOGIN.CMD;1          1.          04-AUG-81 08:43
LOGOUT.CMD;5         1.          28-MAY-80 08:49
STARTUP.CMD;57       14.         06-JUL-81 10:53
PRIMES.MAC;43        8.          15-JAN-81 16:15
ADVENTURE.TSK;1      151.     C  05-AUG-81 16:29
CLAMBAKE.TXT;1       3.          05-AUG-81 16:18
NOTICE.TXT;1         14.         16-MAR-81 10:34
PRIMES.OBJ;1         2.          14-AUG-81 11:09
PRIMES.LST;1         13.         14-AUG-81 11:09
PRIMES.TSK;1         46.      C  14-AUG-81 11:09
PRIMES.MAP;1         3.          14-AUG-81 11:09
A.MAC;1              3.          15-DEC-81 12:03
VIP.TSK;1            4.       C  19-AUG-81 13:58
PROG.TSK;1           6.       C  18-AUG-81 16:31
VIP.TSK;2            4.       C  15-DEC-81 12:04
PROG.TSK;2           6.       C  15-DEC-81 12:04

Total of 279./283. blocks in 16. files
```

Example 4-7 Renaming Files (Sheet 1 of 2)

```
>RENAME A.MAC A.MAC;2
>DIR

Directory DK2:[305,304]
15-DEC-81 12:08

LOGIN.CMD;1          1.          04-AUG-81 08:43
LOGOUT.CMD;5         1.          28-MAY-80 08:49
STARTUP.CMD;57       14.         06-JUL-81 10:53
PRIMES.MAC;43        8.          15-JAN-81 16:15
ADVENTURE.TSK;1      151.     C  05-AUG-81 16:29
CLAMBAKE.TXT;1       3.          05-AUG-81 16:18
NOTICE.TXT;1         14.         16-MAR-81 10:34
PRIMES.OBJ;1         2.          14-AUG-81 11:09
PRIMES.LST;1         13.         14-AUG-81 11:09
PRIMES.TSK;1         46.      C  14-AUG-81 11:09
PRIMES.MAP;1         3.          14-AUG-81 11:09
A.MAC;2              3.          15-DEC-81 12:03
VIP.TSK;1            4.       C  19-AUG-81 13:58
PROG.TSK;1           6.       C  18-AUG-81 16:31
VIP.TSK;2            4.       C  15-DEC-81 12:04
PROG.TSK;2           6.       C  15-DEC-81 12:04

Total of 279./283. blocks in 16. files

>
>


>
>
>RENAME VIP.*;* A.*;*
>
>
>
>DIR A.*;*

Directory DK2:[305,303]
15-DEC-81 12:06

A.MAC;5              3.          15-DEC-81 11:53
A.TSK;1              4.       C  14-AUG-81 10:48
A.OBJ;12             1.          15-DEC-81 12:00
A.OBJ;13             1.          15-DEC-81 12:00
A.OBJ;14             1.          15-DEC-81 12:00
A.LST;5              5.          15-DEC-81 12:00

Total of 15./15. blocks in 6. files
```

Example 4-7 Renaming Files (Sheet 2 of 2)

INSPECTING THE CONTENTS OF FILES

The contents of a file can either be displayed on your terminal (hard-copy or CRT) or sent to a line printer for fast hard-copy output. Only ASCII files should be displayed or printed. Other files such as object files and task image files are in a format understood only by a machine. Displaying them at your terminal or on the printer produces undesirable results.

Displaying File Contents at Your Terminal

You will frequently want to inspect the contents of a file quickly without using an editor. This is done by issuing the TYPE command. The contents of the file are then displayed at the receiving speed of your terminal. When using a CRT, it is useful to use the Scroll key or <CTRL/S> and <CTRL/Q> to aid in reading the data. Typing a non-ASCII file, such as an object file or a task image file, to a CRT produces undesirable results. It is also possible that the combinations of characters created by the interpretation of non-ASCII data will cause certain terminal features, like holdscreen, to be set. In the case of hard-copy terminals, the interpretation of non-ASCII data may cause form ejects and line feeds to spew paper behind the terminal.

Command Format

```
>TYPE  NEW.TXT

>TYPE  DK1:[305,303]MACIO.MAC

>TYPE  VIP.LST;3

>TYPE  LB:[1,2]DCLLOGIN.HLP
```

```
>
>
>TYPE NEW.TXT
TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.

ALL'S WELL THAT ENDS.

UNDER THE MOST RIGOROUSLY CONTROLLED CONITIONS OF PRESSURE
TEMPERATURE, VOLUME, HUMIDITY, AND OTHER VARIABLES THE
ORGANISM WILL DO AS IT DARN WELL PLEASES.
ANY GIVEN PROGRAM, WHEN RUNNING IS OBSOLETE.

THE DEGREE OF TECHNICAL COMPETENCE IS INVERSELY PROPORTIONAL
TO THE LEVEL OF MANAGEMENT.
TELL A MAN THERE ARE 300 BILLION STARS IN THE UNIVERSE
AND HE'LL BELIEVE YOU.  TELL HIM A BENCH HAS WET PAINT
ON IT AND HE'LL HAVE TO TOUCH IT TO BE SURE.

>
```

Example 4-8 Using the TYPE Command

Displaying Files on a Line Printer

To obtain a hard copy of a text file on the line printer, issue the PRINT command in one of the formats shown below. An entry will be placed in a list managed by a task called the Queue Manager, and when the printer is available the file will print. This command has many qualifiers that enhance its use. For printing more than one copy at a time, the /COPIES:n qualifier is used. If special forms are needed when the job is printed, that requirement is indicated by using the /FORMs:n qualifier. n is a number and specifies the type of form . The correlation between the number and type of form is a site consideration that varies from location to location. The first page that is printed on a print job is called the job flag-page and contains information regarding the job, i.e., the file name, the date and time of printing. Before each file is a file flag-page, giving information about the file. When printing multiple files with one print command, there will be one job flag-page and no file flag-pages printed.

Command Format

```
>PRINT VIP.MAC

>PRINT/COPIES:2  INFO.TXT

>PRINT/FLAG_PAGE  INFO.TXT,VIP.MAC/CO:2

>PRINT/PRIORITY:100  VIP.MAC

>PRINT/DELETE  INFO.TXT

>PRINT/FORMS:1  PROG.LST
```

PRINT Command Defaults

```
/NO DELETE
/COPIES:1
/FORMS:Ø
/PRIORITY:5Ø.
/NOFLAG_PAGE
```

LEARNING ACTIVITIES

1. READ the sections on the following commands in Chapter 4 of the RSX-11M/M-PLU Command Language Manual.

   - Directory
   - Delete
   - Purge
   - Copy
   - Type
   - Print

PROTECTING YOUR FILES

In a multiuser system, it is advantageous to be able to protect files from being deleted or corrupted by unknowing users, and to provide data security when it is necessary. In the RSX-11M/M-PLUS FILES-11 system, this is provided through two file attributes; file ownership and file access rights. Every file has stored with it the owner of the file and the access privileges that have been granted to the owner and other users of the system. In addition, access rights can be established on a UFD and on a volume. To be able to access a file, a user must meet the conditions at all three levels.

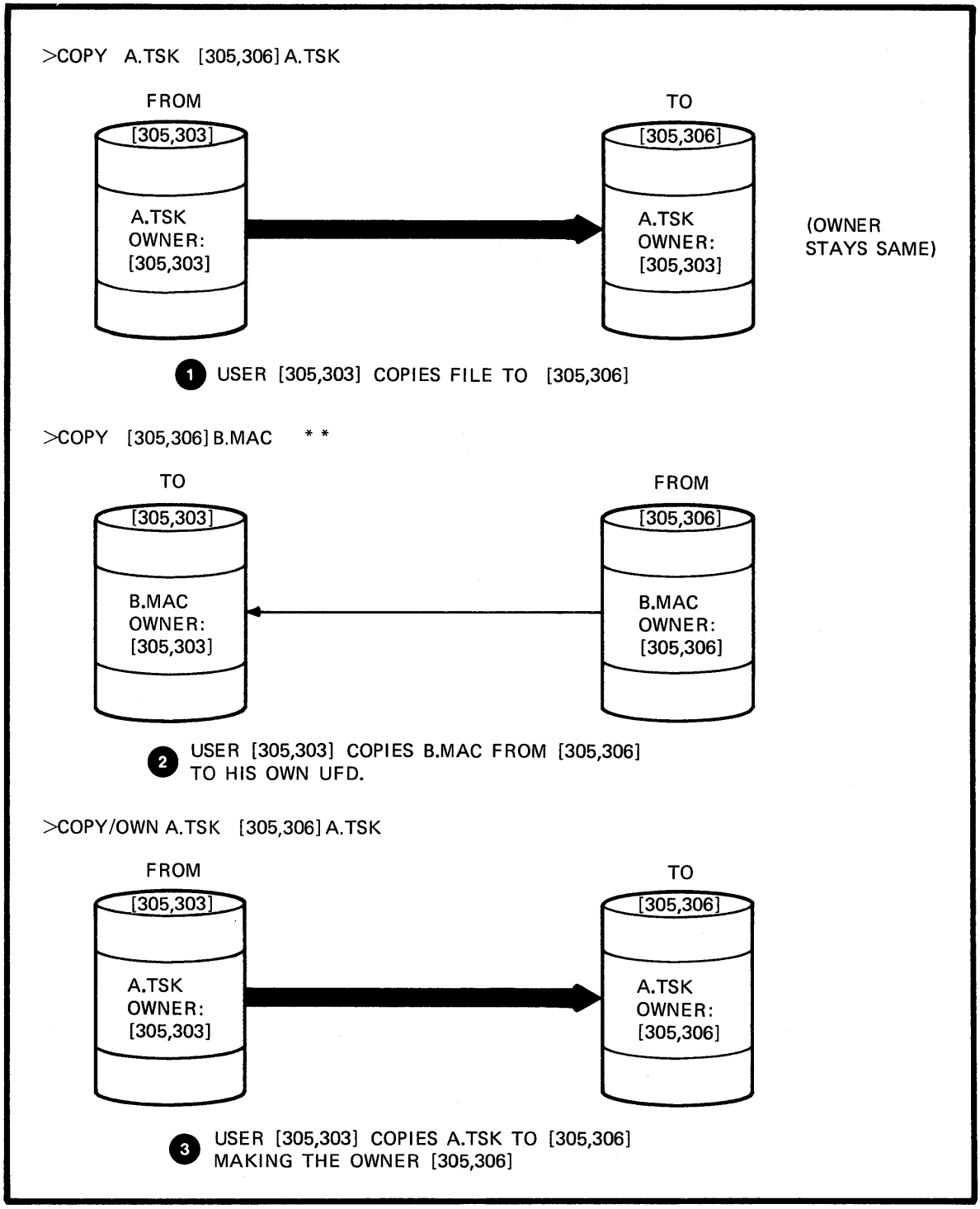
File Ownership

Every file has stored with it the UIC of the user who created the file. (Refer to item 12 in Example 4-3 to review how to obtain this information.) In item 1 in Example 4-9, the person who owns the UFD is also the owner of the file PAYROLL.CBL located in that UFD. When a user or a task attempts to access the file, the UIC of the requesting user or task is compared with that of the owner of the file and access is granted or denied depending upon that relationship. The owner of a file does not always coincide with the owner of the UFD in which the file resides, as in item 2 of Example 4-9.

Figure 4-4 illustrates the transfer of file ownership.

```
>
>DIR/FULL PAYROLL.CBL

Directory DK2:[305,303]
15-DEC-81 11:47

(1) PAYROLL.CBL;1        (17,2)          2./2.          08-JAN-81 15:31
    [305,303][RWED,RWED,RWED,R]    14-AUG-81 10:51(10.)

Total of 2./2. blocks in 1. file

>DIR [305,304]


>
>DIR/FULL EX2D4T.LST

Directory DR0:[305,303]
15-OCT-81 16:09

(2) EX2D4T.LST;4         (3734,150)      8./8.          15-OCT-81 16:08
    [7,305]  [RWED,RWED,RWED,R]

Total of 8./8. blocks in 1. file
```

Example 4-9 File Ownership



TK-7676

Figure 4-4 Transferring File Ownership

When a user copies a file from his UFD to another UFD, he maintains ownership of the file and his UIC will be stored with the file. In Figure 4-4, at item 1, user [305,303] copies the file A.TSK to UFD [305,306] and remains the owner of file. The owner of UFD [305,306] may or may not have access to that file, depending upon what protection rights the owner of the file has given to members of the same group. When a user copies a file from another UFD, the user becomes the owner of the file as shown in item 2 of the figure. If the user wishes to transfer file ownership to the UFD to which it is being copied, the /OWN qualifier should be used, as shown in item 3.

```
>
>COPY VIP.TSK [305,304]VIP.TSK

>
>DIR/FULL [305,304]VIP.TSK


Directory DK2:[305,304]
15-DEC-81 12:22

VIP.TSK;1           (14,5)          7./7.         C  15-DEC-81 12:21
  [305,303][RWED,RWED,RWED,R]

Total of 7./7. blocks in 1. file

>


>
>COPY [305,304]PRIMES.TSK [305,303]
>DIR/FULL PRIMES.TSK


Directory DK2:[305,303]
15-DEC-81 12:22

PRIMES.TSK;2        (16,5)          46./46.       C  15-DEC-81 12:22
  [305,303][RWED,RWED,RWED,R]

Total of 46./46. blocks in 1. file

>
<


>
>COPY/OWN VIP.TSK [305,304]VIP.TSK
>
>
>DIR/FULL [305,304]VIP.TSK


Directory DK2:[305,304]
15-DEC-81 12:23

VIP.TSK;1           (14,6)          7./7.         C  15-DEC-81 12:23
  [305,304][RWED,RWED,RWED,R]

Total of 7./7. blocks in 1. file
```

Example 4-10 Transferring File Ownership Samples

File Access Rights

There are four groups of users, as shown in Table 4-4. The owner is the user whose UIC is saved with the file. Each one of these groups can be awarded up to four different types of access, as listed in Table 4-5. If the user has not specified a protection, the system will assign the default file protection shown in Table 4-6.

Table 4-4 Groups of Users

Owner	
[305,303]	Owner's UIC
Group	
[305,306] [305,301] [305,303]	Same UIC group as owner
System	
[5,6]	Special UIC group
[7,2]	Group # <= 108
[10,300]	Privileged User
World	
[307,5] [322,323] [200,20]	All others

Table 4-5 Access Rights

Access Right	Operations That Can be Performed
READ	Copy, Print, Type Directory, Run Task Image
WRITE	Includes READ rights and may alter file (WRITE)
EXTEND	Includes READ, WRITE rights plus the ability to extend space allocated to file
DELETE	Includes all rights above plus the ability to delete

Default File Protection

Table 4-6 Default File Protection

SY:RWED	OW:RWED	GR:RWED	WO:R
System	Read, Write, Extend, Delete		
Owner	Read, Write, Extend, Delete		
Group	Read, Write, Extend, Delete		
World	Read		

Setting and Changing File Protection

The user may change the protection on any file that he owns, no matter where it resides, by using the SET PROTECTION command. However, another user who is a member of the user's group or world may not change the protection on the user's file even if he has read, write, extend and delete privileges. A privileged user may change the protection on any file. The format of the command is shown below. An example of issuing the command is shown in Example 4-11.

Command Format

```
>SET PROTECTION
  File? INFO.TXT
  Code? (SY:RWED,OW:RWED,GR:RWED,WO:R
```

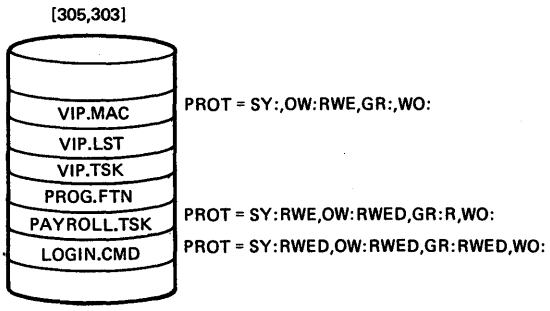
```
>
>
>DIR/FULL INFO.TXT

Directory DK2:[305,303]
15-DEC-81 12:23

INFO.TXT;1          (20,2)          60./60.          07-JUL-80 12:47
  [305,303](R,R,,)                14-AUG-81 10:54(11.)

Total of 60./60. blocks in 1. file

>
>
>
>SET PROTECTION
File? INFO.TXT
Code? (SY:R,OW:RWED,GR:R,WO:)
>DIR/FULL INFO.TXT

Directory DK2:[305,303]
15-DEC-81 12:24

INFO.TXT;1          (20,2)          60./60.          07-JUL-80 12:47
  [305,303](R,RWED,R,)            14-AUG-81 10:54(11.)

Total of 60./60. blocks in 1. file
```

Example 4-11 Setting File Protection

Figure 4-5 shows the accessibility of each group of users to the files located in the UFD [305,303]. It is understood that [305,303] is the owner of all the files located there. A member of the group [305,xxx] would not be able to type the file [305,303]VIP.MAC, because the owner of the file has not given members of his group read privileges, which you must have to type a file. The owner is the only one who can access this file, as no privileges have been given to any other group. Do not be misled into thinking that a privileged user is not able to access this file. A privileged user may change the protection on any file. This is important to remember: if you should issue the SET PROTECTION command, you can take away all rights to every group, including yourself.

Equivalent MCR Commands

Table 4-7 gives the equivalent MCR commands for the file maintenance operations discussed in this module. The utility task, PIP, performs the file maintenance tasks. Documentation for PIP is found in the RSX-11M/M-PLUS Utilities Manual.



COMMANDS	SYSTEM [5,6]	OWNER [305,303]	GROUP [305,304]	WORLD [200,20]
COPY PAYROLL.TSK [UIC] NEW.TSK	ACCESS = YES	ACCESS = YES	ACCESS = YES	ACCESS = NO
DEL [305,303] PAYROLL.TSK	NO	YES	NO	NO
RUN [305,303] PAYROLL.TSK	YES	YES	YES	NO
TYPE [305,303] VIP.MAC	NO	YES	NO	NO
EDIT/EDT [305,303] LOGIN.CMD	YES	YES	YES	NO

TK-7682

Figure 4-5 Accessing a File

LEARNING ACTIVITIES

1. READ the Section on File Protection and Volume Protection, in Chapter 4 of the RSX-11M/M-PLUS Command Language Manual.

2. DO the Written Exercises for this module.

3. DO the Lab Exercises for this module.

Table 4-7 Equivalent MCR File Maintenance Commands

DCL Command	Equivalent MCR Command
DIR	PIP /LI
DIR VIP.MAC	PIP VIP.MAC/LI
DIR/FULL PAYROLL.CBL	PIP PAYROLL.CBL/FU
DIR [305,304]	PIP [305,304]/LI
DIR/BRIEF	PIP /BR
DIR/SUMMARY	PIP /TB
DIR/PRINT	PIP LP0:=/LI
DIR/OUTPUT:DIRECTORY.LST	PIP DIRECTORY.LST=/LI
DIR/FREE DB0:	PIP DB0:/FR
DEL VIP.MAC;*	PIP VIP.MAC;*/DE
DEL/LOG A.OBJ;*	PIP A.OBJ;*/DE/LD
DEL/QUERY *.MAC;*	PIP *.MAC;*/SD
PURGE A.MAC	PIP A.MAC/PU
PURGE/LOG A.MAC,B.*	PIP A.MAC,B.*/PU/LD
PURGE/KEEP:3 C.LST	PIP C.LST/PU:3
COPY VIP.MAC [305,304]VIP.MAC	PIP [305,304]VIP.MAC/NV=VIP.MAC
COPY VIP.TSK,PROG.TSK [305,304]*.*	PIP [305,304]*.*/NV=VIP.TSK,-PROG.TSK
RENAME A.MAC B.MAC	PIP B.MAC=A.MAC/RE
TYPE NEW.TXT	PIP TI:=NEW.TXT
SET PROTECTION NEW.TXT CODE?(SY:RWED,OW:RWED,GR:RWED,WO:R)	PIP NEW.TXT/PR/SY:RWED/OW:RWED-/GR:RWED/WO:R
COPY/OWN NEW.TXT [305,304]NEW.TXT	PIP [305,304]NEW.TXT/NV/FO=NEW.TXT

Digital Equipment Corporation • Bedford, MA 01730