
HP 3000 MPE/iX Computer Systems
Using HP 3000 MPE/iX:
Fundamental Skills
Tutorial



HP Part No. 32650-90871
Printed in U.S.A. 1998

Edition 1
E1098

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Copyright © 1998 by Hewlett-Packard Company

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. Rights for non-DoD U.S. Government Departments and agencies are as set forth in FAR 52.227-19 (c) (1,2).

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Restricted Rights Legend

Preface

This first edition of *Using HP 3000 MPE/iX: Fundamental Skills Tutorial* is a replacement for the training course, *Using the 900 Series HP 3000: Fundamental Skills*, which was printed in separate modules.

This self-paced course has hands-on exercises now contained in one booklet. There are six study modules composed of a number of lessons. Other chapters contain setup information and module solutions. A glossary of terms is included.

MPE/iX, Multiprogramming Executive with Integrated POSIX, is the latest in a series of forward-compatible operating systems for the HP 3000 line of computers. You will encounter references to MPE XL, the direct predecessor of MPE/iX. MPE/iX is a superset of MPE XL. All programs written for MPE XL will run without change under MPE/iX. You can continue to use MPE XL system documentation, although it may not refer to features added to the operating system to support POSIX (for example, hierarchical directories).

You may also encounter references to MPE V, which is the operating system for HP 3000s, not based on PA-RISC architecture. MPE V software can be run on the PA-RISC (Series 900) HP 3000s in what is known as *compatibility mode*.

Contents

1. Begin Here	
What Is This Course?	1-1
Intended audience	1-1
How Should This Course Be Used?	1-1
Organization	1-1
Procedure	1-2
Materials list	1-2
Course essentials	1-2
References	1-2
What Is This Training About?	1-2
Module topics	1-3
Module 1: Communicating with Your Computer	1-3
Module 2: Computer Overview	1-3
Module 3: Creating Text Files	1-3
Module 4: The Account Structure	1-3
Module 5: Working with Files	1-3
Module 6: Command Files	1-3
Other Information	1-4
Conventions	1-4
2. Module 1: Communicating with Your Computer	
Challenge Test	2-1
Lesson 1 Logging On and Off	2-3
Logging on	2-3
HELLO	2-3
Your logon identity	2-4
Logging on to the HP 3000	2-4
The prompt and the cursor	2-5
The welcome message	2-5
How to log off	2-5
Any problems?	2-5
Lesson 1 Exercises	2-6
Lesson 2 Session and Jobs—What Are They?	2-7
Sessions	2-7
Jobs	2-7
Advantages of using a job	2-7
Disadvantages of using a job	2-8
Lesson 2 Exercises	2-8
Lesson 3 Entering Commands	2-9
Understanding commands	2-9
Executing a command	2-9
The SHOWTIME command	2-9
The SHOWME command	2-9
Basic keyboard use	2-11
Terminal screen scrolling	2-11
One solution	2-11

Another solution	2-11
The SHOWJOB command	2-12
User-defined commands and command files	2-12
Lesson 3 Exercises	2-13
Lesson 4 Mistakes and Corrections	2-14
Mistakes	2-14
Error messages	2-14
Correcting an error	2-14
Reenter the command	2-14
Backspace over the error	2-14
The REDO command	2-15
REDO subcommands	2-15
Correcting commands with REDO	2-16
Lesson 4 Exercises	2-19
Lesson 5 The History Stack	2-20
What is the command line history stack?	2-20
Command history and the LISTREDO command	2-21
The DO command	2-21
History stack numbering	2-21
Command history and REDO	2-22
Lesson 5 Exercises	2-23
Lesson 6 Communicating with Other Users	2-24
Sending and receiving messages	2-24
Sending a message to another user	2-24
What the receiver sees	2-25
Blocking messages	2-25
Unblocking messages	2-26
Message to the system operator: TELLOP	2-26
On the lookout	2-27
Warning messages	2-27
Lesson 6 Exercises	2-28
Lesson 7 More Information	2-29
Logon problems	2-29
General user references	2-29
The reference manual	2-29
Lesson 7 Exercises	2-30
Lesson 8 Online Help	2-31
The online help facility	2-31
Getting into help	2-31
Getting out of help	2-31
Getting around in help	2-32
Syntax	2-32
The help keywords	2-32
The help facility screen	2-33
The help facility's online tutorial	2-33
Lesson 8 Exercises	2-34

3. Module 2: Computer Overview	
Challenge Test	3-1
Lesson 1 Computer Hardware	3-2
Hardware	3-2
Peripherals	3-2
Lesson 1 Exercises	3-4
Lesson 2 Software	3-5
Operating system	3-5
Sample programs	3-5
The command interpreter program	3-6
Lesson 2 Exercises	3-6
Lesson 3 The Keyboard	3-7
The return key	3-8
Backspace key	3-8
Numeric keys	3-9
Numeric keypad	3-9
Cursor control keys	3-9
Function keys	3-10
Other special keys	3-10
Lesson 3 Exercises	3-11
Lesson 4 Any Problems?	3-12
Sources of information	3-12
Terminal not responding	3-12
Terminal on but not responding	3-13
Changing modes with the softkeys	3-13
Initial display	3-14
The other softkey keys	3-15
Is the terminal locked up?	3-16
Remedies	3-16
Soft reset	3-16
Hard reset	3-17
Turn the terminal off	3-17
Talk to your account manager	3-17
Remedies - the pros and cons	3-17
Lesson 4 Exercises	3-18
4. Module 3: Creating Text Files	
Challenge Test	4-1
Lesson 1 Using EDIT/3000	4-2
What is EDIT/3000?	4-2
Why use EDIT/3000?	4-2
Starting EDIT/3000: EDITOR	4-2
Leaving the editor: END	4-3
Writing in the editor: ADD	4-3
Typing lines	4-4
Stop Writing (//)	4-5
Looking at your file: LIST	4-5
Saving a file: KEEP	4-6
Keep an existing file	4-6
Getting a listing of files: LISTFILE	4-8
Lesson 1 Exercises	4-8
Lesson 2 File Names	4-10
File naming conventions	4-10
A troublesome character	4-11
Invalid file names	4-11

Lesson 2 Exercise	4-11
Lesson 3 Editing a Text File	4-13
Open an EDIT/3000 file for editing	4-13
Sending a file to the printer	4-13
Identifying your document	4-14
Problems?	4-14
Adding lines of text: ADD	4-14
Adding lines in the middle of a text file: ADD	4-15
Erase a line: DELETE	4-16
Renumbering lines: GATHER ALL	4-16
Moving lines: GATHER	4-17
Renumbering lines	4-17
Editing a text line: MODIFY	4-18
MODIFY Works Like REDO	4-18
Starting a new editor file: DELETE ALL	4-19
Lesson 3 Exercises	4-20
Lesson 4 More About the Editor	4-22
Joining files together: JOIN	4-22
Listing files from within the editor	4-22
Joining at the end	4-23
Joining in the middle	4-24
Suspending EDIT/3000: the Break key	4-24
Using the Break key	4-24
Continuing a Program: RESUME	4-25
Use ABORT after BREAK to stop	4-26
Safety files: K files	4-27
What the numbers mean	4-28
What is saved in a K file	4-28
Recovering a K file	4-28
Lesson 4 Exercises	4-29

5. Module 4: The Account Structure

Challenge Test	5-1
Lesson 1 The MPE/iX Account Structure	5-3
What is an account structure?	5-3
Organization	5-3
Security	5-3
Billing	5-3
Elements of the account structure	5-3
Accounts	5-3
Users	5-4
Groups	5-4
Files	5-4
Elements of your logon	5-6
User and account names	5-6
Group names	5-7
Session names	5-7
Logon requirements	5-8
The PUB Group	5-8
The SYS account	5-8
Your Home Group	5-8
Discovering your home group	5-9
PUB as your home group	5-9
Lesson 1 Exercises	5-10
Lesson 2 Security and the Account Structure	5-11

MPE/iX command security and capabilities	5-11
Logon security	5-12
Lesson 2 Exercises	5-13
Lesson 3 File and Group Security	5-14
File security in your Home group	5-14
File Security in PUB	5-14
File security in other groups	5-16
Lesson 3 Exercises	5-16
Lesson 4 Changing Groups	5-18
Logging on	5-18
Using the CHGROUP command	5-18
CHGROUP and your home group	5-19
Lesson 4 Exercises	5-19
Lesson 5 Changing User Passwords	5-20
Your user password	5-20
Using the PASSWORD command	5-20
Lesson 5 Exercises	5-21

6. Module 5: Working with Files

Challenge Test	6-1
Lesson 1 Displaying Qualified File Names	6-3
Accessing files in other groups and accounts	6-3
Partly qualified name	6-3
Fully qualified name	6-3
Forming a fully qualified file name	6-3
When to use the fully qualified file name	6-4
File name and logon—the differences	6-4
Fully qualified name	6-4
The patterns	6-5
Displaying all of the file names in your group	6-5
Displaying selected file names in your group	6-5
The @ symbol	6-6
Display file names in other groups and accounts	6-6
Finding files	6-7
Lesson 1 Exercises	6-9
Lesson 2 Renaming Files	6-10
Safeguard against duplication	6-10
Lesson 2 Exercises	6-10
Lesson 3 Storing Seldom-Used Files	6-11
Lesson 3 Exercises	6-11
Lesson 4 Deleting Files	6-12
The PURGE command	6-12
Recovering a file	6-12
Lesson 4 Exercises	6-14
Lesson 5 Displaying File Contents	6-15
Displaying an entire file	6-15
Displaying selected lines	6-15
The PRINT parameters	6-15
Lesson 5 Exercises	6-15
Lesson 6 Copying Files	6-16
Using the COPY command	6-16
A simpler COPY	6-16
Copy to an existing file	6-17
Copy over an existing file	6-17
More practice with COPY	6-18

Releasing file security with the RELEASE command	6-18
Copying from another group	6-19
Default names in copying	6-19
Default group name	6-19
Default file name	6-20
Copy and change file name	6-20
Group to group security: what works	6-22
Reinstating system security to a file	6-23
The SECURE command	6-23
Conserving disk space	6-24
Lesson 6 Exercises	6-25

7. Module 6: Command Files

Challenge Test	7-1
Lesson 1 Using Command Files	7-2
What is a command file?	7-2
Creating and executing command files	7-2
Command files that execute other command files	7-3
Lesson 1 Exercises	7-4
Lesson 2 Naming Conflicts	7-5
Discovering a name duplication	7-5
Avoiding duplication	7-5
HELP and verifying name duplications	7-5
The XEQ command	7-6
Lesson 2 Exercises	7-7

8. Solutions

Solutions to Module 1 Communicating With Your Computer	8-1
Challenge Answers	8-1
Lesson 1: Logging On and Off	8-1
Lesson 2: Sessions and Jobs—What Are They?	8-2
Lesson 3: Entering Commands	8-2
Lesson 4: Mistakes and Corrections	8-2
Lesson 5: The History Stack	8-2
Lesson 6: Communicating with Other Users	8-3
Lesson 7: More Information	8-3
Lesson 8: Online HELP	8-3
Solutions to Module 2 Computer Overview	8-4
Challenge Answers	8-4
Lesson 1: Computer Hardware	8-4
Lesson 2: Software	8-4
Lesson 3: The Keyboard	8-5
Lesson 4: Any Problems?	8-5
Solutions to Module 3 Creating Text Files	8-6
Challenge Answers	8-6
Lesson 1: Using EDIT/3000	8-6
Lesson 2: File Names	8-6
Lesson 3: Editing a Text File	8-7
Lesson 4: More About the Editor	8-7
Solutions to Module 4 The Account Structure	8-8
Challenge Answers	8-8
Lesson 1: The MPE/iX Account Structure	8-8
Lesson 2: Security and the Account Structure	8-9
Lesson 3: File and Group Security	8-9

Lesson 4: Changing Groups	8-9
Lesson 5: Changing User Passwords	8-9
Solutions to Module 5 Working With Files	8-10
Challenge Answers	8-10
Lesson 1: Displaying Qualified File Name	8-10
Lesson 2: Renaming Files	8-11
Lesson 3: Storing Seldom-Used Files	8-11
Lesson 4: Deleting Files	8-11
Lesson 5: Displaying File Contents	8-11
Lesson 6: Copying Files	8-11
Solutions to Module 6 Command Files	8-12
Challenge Answers	8-12
Lesson 1: Using Command Files	8-12
Lesson 2: Naming Conflicts	8-13

Glossary

Index

Figures

3-1. A Typical Computer System	3-4
3-2. HP 2392 or HP 700/92 Keyboard	3-7
3-3. Video Screen at Logon	3-13
3-4. Softkey Function Display	3-14
5-1. Elements of Account Structure	5-5
5-2. Files in Groups in an Account	5-6
6-1. PUB and Your Home Group: Relaxed Security	6-21
6-2. Other Groups: Strict Security	6-22

Tables

3-1. Other Special Keys	3-11
3-2. Soft Reset Versus Hard Reset	3-18
4-1. File Names	4-10
4-2. File Naming Errors	4-11
4-3. The K Numbers	4-28
5-1. The Basic (Default) Set of Capabilities	5-12
6-1. Qualified File Names	6-4
6-2. You Are in PUB	6-23
6-3. You Are in MYGROUP (Your Home Group)	6-23
6-4. You Are in OTHERGRP (Not PUB, Not Home)	6-23

Begin Here

What Is This Course?

Using HP 3000 MPE/iX: Fundamental Skills Tutorial is a self-paced training tool for all users who are new to the HP 3000 family of computers. Lessons emphasize hands-on use that acts as the foundation for all other training in this series.

Intended audience

This training course was written for all users who log on and use a 900 Series HP 3000 computer.

The users include end users (non-data-processing professionals), system operators, system managers, and programmers. Not all users need to complete all of the modules in order to utilize the 900 Series HP 3000 successfully.

Using HP 3000 MPE/iX: Fundamental Skills Tutorial is prerequisite training for *Using HP 3000 MPE/iX: Advanced Skills*, also a self-paced tutorial.

How Should This Course Be Used?

Organization

The information in this course has been organized into separately bound modules.

The modules are organized into lessons. Each lesson gives practical examples and activities that can be tried on the computer. The exercises at the end of each lesson direct you to try things on the system in order to reinforce important concepts presented in the lesson. The answers to all of the exercises can be found in the "Solutions Guide".

Modules are numbered 1 through 6. All modules begin with a challenge test. The challenge test helps you determine if you need to study the entire module, study parts of it, or skip the module entirely. If you choose to skip certain lessons or modules, note that later modules refer to files created in earlier modules.

Procedure	<ul style="list-style-type: none"> ■ Use each module challenge test to decide which lessons you should cover in that module. ■ Begin each selected lesson by reading the lesson introduction. Try out the examples, and do all the exercises in that lesson. ■ The exercises at the end of each lesson reinforce the important concepts presented in the lesson.
Materials list	The <i>Using HP 3000 MPE/iX: Fundamental Skills Tutorial</i> course consists the book and a tape. Additional course materials may be ordered through your Hewlett-Packard sales representative.
Course essentials	<p>You need a logon to a 900 Series HP 3000 in order to complete the exercises in each lesson. You are the only person who should use this logon during this training.</p> <p>You need basic account capabilities (IA, BA, ND, SF).</p> <p>Module 4 requires you to have three groups: MYGROUP, OTHERGRP, and PUB in your account. MYGROUP should be your home group.</p> <p>Your system or account manager can arrange for you to have these capabilities and groups.</p>
Note	<hr/> <p>If you are responsible for setting up the student accounts for this course, you may use the lab tape along with the account setup instructions provided in the “Begin Here” section of the <i>Using HP 3000 MPE/iX: Advanced Skills Tutorial</i> course.</p> <hr/>
References	<p>Your system has an online Help Facility, which is discussed in Module 1 of this course.</p> <p>Additional information regarding subjects covered in this course can be found in the <i>MPE/iX Commands Reference Manual (32650-60002)</i>. This manual can be purchased separately from your Hewlett-Packard sales representative.</p>

What Is This Training About?

Long known for their reliability and ease of use, the HP 3000s are used in highly diverse settings: offices, factories, government, and educational organizations.

The 900 Series HP 3000 system implements Hewlett-Packard's Precision Architecture, which is in the forefront of computer technology. The 900 Series operating system, MPE/iX, creates a friendly and productive user environment.

Some features of MPE/iX that you will be learning about in this tutorial are listed here:

- Easy-to-use commands.
- Useful online Help Facility.
- Powerful text editor.

- Organization of the account structure.
- Ample means for maintaining system security.
- Command files (user-created commands).

Module topics

Module 1: Communicating with Your Computer

This module teaches you how to communicate with your HP 3000. This includes the process of logging on and off, the execution of several useful MPE/iX commands, and the online Help Facility. These lessons are essential for all new users of the HP 3000 system.

Module 2: Computer Overview

In this module you become familiar with general hardware and software computer concepts (lessons 1 and 2, respectively). If you are new to computers, you will be especially interested in these lessons. Basic keyboard skills (lesson 3) and terminal troubleshooting techniques (lesson 4) will benefit all new users of Hewlett-Packard terminals.

Module 3: Creating Text Files

This module introduces you to EDIT/3000, a text and document processor on the HP 3000. Users, including programmers, who need to gain a quick working knowledge of a text processor should go through the entire module. Later modules assume your knowledge of the text-processor and require that you use it to create files.

If you will be maintaining other users' files or running other programs, you may find the information in lesson 4 of interest.

Module 4: The Account Structure

This module discusses the computer's account structure, and explains how your system recognizes you as a user. Also discussed are the security measures that ensure only authorized use of the computer. Understanding the account structure will make you a more productive and effective user of the system.

System managers and system operators, in particular, will be interested in this module.

Module 5: Working with Files

In this module, you learn how to manipulate files on the system. This includes accessing specific files, copying, renaming, deleting, and printing files. This module also introduces you to common file housekeeping practices. Users who have a need to manage their own files or the files of others will benefit from this module.

Module 6: Command Files

This module presents a kind of user-created command called command files. Programmers will be particularly interested in this convenient feature of the system.

Other Information

Solutions to all exercises of this course can be found in the chapter, “Solutions Guide.”

The Glossary provides you with a glossary of general usage terms.

Conventions

Read this important information before you begin to do the modules.

These modules use instructions and different typefaces to clarify what to read and what to do. The following reflects the conventions used throughout this course:

Sample	Information for you to read and know.
Sample	A term you ought to recognize or understand.
<i>Sample</i>	Emphasis to bring something to your attention.
<u>SAMPLE</u> (Return)	Something that is underlined and in this typeface, followed by (Return), is what you should enter by typing the letters and pressing (Return).
SAMPLE	Something in this typeface, <i>without</i> (Return) shows you how something should look on your terminal screen. <i>Do not</i> enter anything you see in this typeface unless it is followed by (Return).
:SAMPLE	Something in this typeface preceded by a colon, is another way of showing you how something should look on your terminal screen. You will learn more about the colon a little later.

Module 1: Communicating with Your Computer

Module 1 introduces some basic commands and how to use them:

Lesson 1	Logging On and Off
Lesson 2	Sessions and Jobs—What Are They?
Lesson 3	Entering Commands
Lesson 4	Mistakes and Corrections
Lesson 5	The History Stack
Lesson 6	Communicating with Other Users
Lesson 7	More Information
Lesson 8	Online Help

Challenge Test

Challenge your current computer skills by taking the following *Challenge Test*. Use your test results to select those lessons that you should study and those that you might skip.

1. Name the command that initiates a session on the computer.
2. Name the command that ends a session with the computer.
3. Identify the two ways of “working” with the HP 3000 computer.
4. Which command displays the current date and time to your terminal screen?
5. Which command displays information about your session on your terminal screen?
6. Which HP 3000 command allows you to list to your terminal screen all jobs and sessions currently running on the computer?
7. Which command allows you to recall an earlier command for error correction?
8. What is the command line history stack?

9. Which command allows you to transmit a terminal screen message to another user on the system?
10. Name two resources from which you can get more information about HP 3000 commands.
11. What would you type at the system prompt to access the MPE/iX online Help facility?

Lesson 1 Logging On and Off

Lesson 1 presents logging on and logging off.

- the command to log on: HELLO
- the command to log off: BYE
- what to do if you have problems logging on or off

Logging on

- Logging on identifies you to the computer. It also starts the communication between you and the computer.
- If you have problems in the logon process, refer to “Any Problems?” at the end of this lesson, or ask for help from someone who is familiar with the computer.

HELLO

Before you log on, the MPE/iX system displays a **logon prompt** on the screen. The standard logon prompt looks like this on the screen:

```
MPE XL:
```

This display means the computer is waiting for you to log on.

Note

On some systems, terminals are connected to the computer over a telephone line. If you are on such a telephone connection and you attempt to log on and fail, your telephone connection to the computer will be terminated. This is no cause for alarm, but you may need assistance from someone who is familiar with the system to ensure that you are communicating with the computer.

The command to log on is HELLO.

Do not attempt to use the command yet. You need a little more information.

A sample logon might look like this on the screen:

```
MPE XL:HELLO JOHN.SMITHERS
```

Notice some things about this sample:

- The computer provides the MPE XL: prompt for you when it is waiting for you to log on.
- HELLO is the command for logging on.
- JOHN.SMITHERS is a **logon identity**. You need a logon identity in order to log on.
- You can enter your logon in uppercase or lowercase letters, or in a combination of uppercase and lowercase letters.

Notice the space between HELLO and JOHN and the period between JOHN and SMITHERS—punctuation is important!

Your logon identity

- Talk to your **account manager** or your **system manager** to get your logon identity and your passwords, if any.
- Write your logon identity on a piece of paper.
- Your passwords are confidential. Do not write them down. Memorize them.

Note

Look for the **Return** key on your keyboard. On some keyboards it might be labeled **Enter**.

Logging on to the HP 3000

Suppose that your logon identity is FRED.MERTZ. (Substitute your real logon identity.)

Here is how to log on to your MPE/iX system. Follow these steps on your terminal:

- STEP 1. Turn on your terminal if it is not already on. Consult the documentation for your terminal if you need to, or ask someone who uses the same kind of terminal.
- STEP 2. Press **Return** one or more times to get the logon prompt (MPE XL:).
- Your system may be set up in a slightly different manner. If you have difficulty getting the MPE XL: prompt, ask for instructions from your operations personnel.*
- STEP 3. Type HELLO followed by your logon identity.
- Example:
- ```
MPE XL:HELLO FRED.MERTZ
```
- If you have the logon prompt, type *your* logon sequence, followed by **Return**.
- If the computer displays a welcome message, go to STEP 5.
- If the computer prompts you for a password(s), go to STEP 4; otherwise, go to STEP 5.
- STEP 4. If you are prompted for a password, type it and press **Return**. For security reasons, what you enter does not appear on the screen.
- You might be prompted for more than one password. Have you memorized your password(s)? If you are prompted for a second password, type it and press **Return**.
- STEP 5. You should be logged on now. You have just started a **session** with the computer.
- You know that you have logged on, because you see the **system prompt**. The system prompt is usually a colon (:) along the left border of your terminal screen.

## Note

---

Your system management can change the colon prompt to something else. If you have questions about the prompt(s) on your terminal screen, talk to someone in your system management—or talk to someone who has used the system for a while.

---

### The prompt and the cursor

The prompt tells you that the computer is waiting for your next instruction. The cursor (a `_` or a small square of light) shows you where the next letter or number that you enter is going to appear on the screen.

### The welcome message

Did a message appear on the screen? If there is one, this **welcome message** might contain useful information from the system operator, who controls the computer from the computer room.

## How to log off

To log off, type BYE at the system prompt and press `(Return)`.

Log off now by entering:

```
BYE(Return)
```

You will see something like this at the end of your session:

```
CPU=3. CONNECT=12. WED, FEB 14, 1990, 11:55 PM.
```

## Any problems?

If you have problems logging on, these hints may help:

In STEP 2: No system prompt appears:

Press `(Return)` several times. If you still do not get a prompt, talk to your account manager or your immediate supervisor.

In STEP 3: You make a typing error in the logon identity, but you *have not* pressed `(Return)`:

Use the backspace key `(←)` to back up to the first error that you made. Then type the *rest* of your logon identity as it should be. Press `(Return)`.

You make a typing error in the logon identity and you *have* pressed `(Return)`:

Retype HELLO with your logon identity and press `(Return)`.

In STEP 4: If you make an error entering a password and you *have* pressed `(Return)`:

Type the password again. Press `(Return)`.

You have three chances to enter a correct password. If you fail in all three attempts, you have to start the logon process over again.



---

## Lesson 2

### Session and Jobs— What Are They?

Lesson 2 presents **sessions** and **jobs**.

Once you are logged on, you can work with your HP 3000 computer in two ways:

- in a **session**
- by **submitting a job**

#### Sessions

Logging on is the way you that begin any interaction with your MPE/iX system. A session begins the moment you that log on (HELLO) and ends when you log off (BYE).

MPE/iX is an **interactive** system. This means that what you enter is acted upon immediately by the computer.

In a session, this is what happens:

- You enter instructions to the computer at the keyboard, one **command** at a time.
- As you type it, your command appears on the screen. When you press **Return**, the computer attempts to carry out the command.
- The computer gives some response on the screen to each command that you try. It is a two-way conversation between you and the computer.

Working in a session is sometimes called **interactive processing**.

#### Jobs

When you work in a session, you see the results of your instructions *as they execute*.

You use a job when you want the computer to carry out several instructions on its own, perhaps after you leave for home.

You do not have to be logged on when the job starts, or when it ends. The job prints out a report for you, telling you what it did and whether it succeeded in doing what you wanted it to do.

#### Advantages of using a job

- You can process all of the commands in a job file at any time with a single command. Your terminal is available to do other work while the job is processed.

You can **schedule** a job to begin at 3:00 A.M. You could schedule a job to begin next year, on a particular day, at a particular time. Starting a job on its way to execution is called **submitting a job**. You may hear the term **streaming**, because the command used to *launch* a job is STREAM.

- A job that is properly created and started will stop itself at the end. At the end, the computer prepares a summary of the job in a **job listing** and prints the summary on the computer's printer.

### Disadvantages of using a job

- If an error occurs while a job is running, the job may stop running. Then you have to study your job listing in order to discover what went wrong.

Running a job is often called **batch processing**. You will hear it called a **batch job**, too.

### Lesson 2 Exercises

1. What are the two ways of “working” with the HP 3000 computer?
2. In the space provided, indicate with an “S” for Session or a “J” for Job which of the following statements best describes each.
  - a. This is also known as *interactive* processing.
  - b. This type of computer command processing allows you to execute several commands *without* your interaction.
  - c. In this kind of processing, commands are sent to the computer one at a time from a terminal keyboard.
  - d. This is also known as aP batch session.

---

## Lesson 3 Entering Commands

Lesson 3 presents entering **commands** and getting information from the computer.

- how to execute commands
- commands to get information:
  - get the current time and date: SHOWTIME
  - get information about your session: SHOWME
  - get information about sessions and jobs: SHOWJOB

### Understanding commands

Commands are the instructions that you give the computer. They may be simple and consist of a single word (SHOWME), or they may be complicated and require that you enter many words or numbers.

### Executing a command

Executing a *simple* command takes only two steps:

1. Type the name of the command at the system prompt.
2. Press the **(Return)** key.

With a few exceptions, commands either execute successfully or give you a warning or an error message. In either case, the prompt returns, and the computer waits for your next instruction.

### The SHOWTIME command

Start a session with your MPE/iX system by logging on now.

HELLO your logon identity**(Return)**

Type SHOWTIME at the prompt, and press **(Return)**.

SHOWTIME**(Return)**

On the screen, you should see something like this:

```
:SHOWTIME
FRI, JUN 30, 1989, 3:58 PM
:_
```

You may find this **online:clock** clock and calendar very useful.

### The SHOWME command

The SHOWME command displays information about your session and any special logon message from your **system operator** or system manager.

Enter:

SHOWME**(Return)**

On the screen you should see something like this:

```

:SHOWME
USER: #S118,JOHN.SMITHERS,MYGROUP (NOT IN BREAK)
RELEASE: A.21.13 MPE/iX HP31900 A.43.11 USER VERSION A.21.31
CURRENT : FRI, JUN 29, 1990, 3:58 PM
LOGON : FRI, JUN 29, 1990, 3:57 PM
CPU SECONDS: 0 CONNECT MINUTES: 1
$STDIN LDEV: 107 $STDLIST LDEV: 107

** WELCOME TO SYSONE **

** LOG OFF BY 7 PM TONIGHT, PLEASE **
** ** **
** NIGHT SHIFT OPERATOR WILL TAKE SYSONE OFFLINE TONIGHT TO **
** STORE ALL FILES ON THE SYSTEM TO A BACKUP TAPE. **
** ** **
** ALL SESSIONS AND JOBS WILL BE ABORTED AT 7:01 PM -- NO **
** EXCEPTIONS. **

** Call ext. 5555 if you have any problems or questions!! **

:_

```

Did the display scroll by too quickly to see? If it did, press the **Prev** key once or twice to go “backward.” Press the **Next** key the same number of times to return to where you started. The terminal has some memory of its own, and you can use these two keys to review whatever the screen “remembers.”

The SHOWME display provides valuable information. Locate each of the following:

|                 |                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| USER            | This identifies you on the computer in two ways: <ul style="list-style-type: none"> <li>■ <i>By session number</i>: in this example, #S118 (session number 118). This number is assigned to you by the system when you log on.</li> <li>■ <i>By logon identity</i>: in this example, JOHN.SMITHERS,MYGROUP (on the screen it would show the way you logged on).</li> </ul> |
| RELEASE         | Tells which version of your MPE/iX operating system you that are using.                                                                                                                                                                                                                                                                                                    |
| USER VERSION    | Allows for the identification of any software customization done for your system.                                                                                                                                                                                                                                                                                          |
| CURRENT         | Shows the day and time the SHOWME command was entered, as tracked by the system.                                                                                                                                                                                                                                                                                           |
| LOGON           | Shows the day and time you that logged on.                                                                                                                                                                                                                                                                                                                                 |
| CONNECT MINUTES | Shows how many minutes have passed since you logged on.                                                                                                                                                                                                                                                                                                                    |
| CPU SECONDS     | Shows the amount of processing time used by a job or session.                                                                                                                                                                                                                                                                                                              |
| \$STDIN LDEV    | This logical device number (here it is 107, a number assigned by the computer) tells your MPE/iX system which terminal you are using in your session.                                                                                                                                                                                                                      |

`$STDLIST`      This logical device number tells your MPE/iX system where file listings are to be sent during your current session. In this instance, it is shown as 107, the same number as your terminal.

Your LDEV number, session number, and logon identity are important. Knowing what version is on your system helps your system operator or system manager to help you with any difficulties.

## Basic keyboard use

Some commands may require that you know a few basic techniques for controlling your terminal.

### Terminal screen scrolling

If you have ever sat through the credits at the end of a movie and the words scrolled up too fast to read, you know why you want to be able to control the scrolling on your terminal.

The `SHOWJOB` command lists all of the users of your MPE/iX system who are currently logged on to a session or running jobs. Since different HP 3000s can accommodate from a few to hundreds of users at one time, some of the entries may scroll up and away too fast.

### One solution

`(STOP)`      If your terminal has a `(STOP)` key, this is the key to use.

If text is scrolling too fast to read, press `(STOP)`. The text will stop moving.

`(STOP)`      If you have stopped the text, press `(STOP)` one more time to start the scrolling again.

### Another solution

`(CTRL)(S)`      If your terminal does not have the `(STOP)` key, press `(CTRL)` and hold it down. While holding it down, press `(S)`. This will stop the scrolling.

`(CTRL)(Q)`      If your terminal does not have the `(STOP)` key, press `(CTRL)` and hold it down. While holding it down, press `(Q)`. This resumes the scrolling.

## Note

---

If you stop the scrolling with `(CTRL)(S)` or with `(STOP)`, you can examine the text on the screen, but you cannot issue any other commands until you start the screen again with `(CTRL)(Q)` or another `(STOP)`.

### *Advanced Users*

The `(STOP)` key may or may not work on some PC emulator terminals.

---

## The SHOWJOB command

The SHOWJOB command displays the identities of sessions and jobs currently logged on to the computer.

Enter:

```
SHOWJOB(Return)
```

Your screen will display something like this:

| JOBNUM | STATE | IPRI  | JIN | JLIST | INTRODUCED | JOB NAME       |
|--------|-------|-------|-----|-------|------------|----------------|
| #S116  | EXEC  |       | 101 | 101   | FRI 9:57A  | DONNA.HOLLAND  |
| #S118  | EXEC  |       | 107 | 107   | FRI 3:57P  | JOHN.SMITHERS  |
| #S119  | EXEC  | QUIET | 109 | 109   | FRI 4:02P  | LEDGER.ACCNTNG |
| #J13   | EXEC  |       | 10S | LP    | FRI 3:00P  | DONNA.HOLLAND  |

```
4 JOBS:
 0 INTRO
 0 WAIT; INCL 0 DEFERRED
 4 EXEC; INCL 3 SESSIONS
 0 SUSP
JOBFENCE= 7; JLIMIT= 60; SLIMIT= 60
```

SHOWJOB may produce much more information than this; but, for the moment, a small example is enough.

Notice that in the example of a SHOWJOB screen, the computer counts jobs and sessions together as JOBS when it totals this information. Notice that Donna Holland is using an interactive session *and* a job simultaneously. She is working on some project using Session 116 (#S116). She is also letting the computer execute a job file identified as Job 13 (#J13).

Notice, too, that someone has logged on using the identity LEDGER.ACCNTNG. A logon identity does not have to be the name of a person.

## User-defined commands and command files

You can create other files besides jobs that contain commands and data for immediate execution. One kind of file is called a **user-defined command** (UDC for short). The other kind is called a **command file**. Together they are referred to as **user commands**. They can be used during sessions.

Jobs and user-defined commands are topics for the *Advanced Skills Tutorial* course. Module 6 introduces command files.

### Lesson 3 Exercises

1. Which command displays the current date and time to your terminal screen?
2. Which command allows you to get current information regarding your present job or session?
3. Study the following:

```
:showme
USER: #S23,ETHEL.MERTZ,PUB (NOT IN BREAK)
RELEASE: X.21.13 MPE/iX HP31900 A.43.11 USER VERSION X.21.31
CURRENT: FRI, JUN 8, 1990, 10:48 AM
LOGON: FRI, JUN 8, 1990, 8:09 AM
CPU SECONDS: 0 CONNECT MINUTES: 159
$STDIN LDEV: 103 $STDLIST LDEV: 103

* WELCOME TO YOUR HP 3000 SYSTEM *
* THERE WILL BE A POWER SHUTDOWN FOR THE ENTIRE SITE *
* ON SATURDAY, JUNE 9, 1990 *

```

Answer the following questions:

- a. Who has initiated this work session?
  - b. What time was the session initiated?
  - c. What is the logical device number?
  - d. What is the present time on this session?
  - e. How many minutes has this session been active?
  - f. When will there be a site-wide power shutdown?
4. Which command displays the identities and status of jobs and sessions currently on the system?

---

## Lesson 4 Mistakes and Corrections

Lesson 4 presents error messages and correcting command errors.

- mistakes and **error messages**
- correcting errors
  - reenter the command
  - use backspace (←) to edit the command
  - use the REDO command to edit the command

### Mistakes

Are you still logged on? If you are not, log on again (HELLO your logon identity (Return)).

What if you make a mistake while entering a command?

Make a mistake. Enter:

```
SHOWHOB(Return)
```

Your screen looks like this:

```
:SHOWHOB
UNKNOWN COMMAND NAME. (CIERR 975)
:_
```

### Error messages

This mistake is just a typing error. The computer does not recognize the command, so it displays an error message that shows the following:

- a brief explanation of the problem (UNKNOWN COMMAND NAME)
- a code that identifies the error message (command interpreter error 975)

### Correcting an error

There are various ways of correcting errors. Choose the method that is most convenient.

#### Reenter the command

You may choose to reenter the command correctly.

Enter:

```
SHOWJOB(Return)
```

to send the correct command to the computer. If you enter the command correctly, it executes, and you see the results.

#### Backspace over the error

Another way to correct an error is to use the backspace key (←) to go back to the character where you made the *first* mistake and reenter the remainder of the command.

---

### Note

This method only works if you have *not yet* pressed the (Return) key.

---

Type SHOWHOB again. *DO NOT PRESS RETURN YET.*

- Press the backspace key three times  (or as many times as it takes to place the cursor directly on the second letter “H.”)

HOB may still appear on the screen, but it has been *erased* from your terminal’s memory. The letters “H,” “O,” and “B” would *not* be sent to the computer if you were to press . In effect, you have reduced the entry to SHOW.

- When the cursor is on H, type JOB.
- Press .

These two methods are best used for short commands. Notice that you have to reenter the command from the first mistake all the way to the end. As you become a more experienced MPE/iX user, reentering lengthy commands using these methods could become tedious.

### The REDO command

The REDO command allows you to edit commands and then reexecute them. When the command line becomes long and complex—and some do—REDO is the easier solution.

Enter:

```
COWEM
```

You should see this on the screen:

```
:COWEM
UNKNOWN COMMAND NAME. (CIERR 975)
:_
```

Now enter:

```
REDO
```

You should see this on the screen:

```
:REDO
COWEM
-
```

Read ahead before doing anything more at your terminal.

### REDO subcommands

REDO retrieves the *most recent* command entered on the computer and gives it back to you for editing. The REDO command allows you to make the editing changes on the *line below* the retrieved command. Notice that there is *no* system prompt (:), just the cursor.

Four simple and useful **subcommands** of REDO are DELETE, INSERT, REPLACE, and UNDO:

- |   |                                                            |
|---|------------------------------------------------------------|
| D | Deletes the character above it.                            |
| I | Inserts character(s), starting at the character above it.  |
| R | Replaces character(s), starting at the character above it. |

U Closes the effect of the previous D, I, or R subcommand.

These subcommands can be entered in uppercase or lowercase letters.

Use the spacebar and backspace (←) to place the cursor exactly under the place where you want to start your corrections.

### Correcting commands with REDO

Correcting with REDO follows a pattern of steps:

- STEP 1. Make *one* correction on the line.
- STEP 2. Press **(Return)**.  
Pressing **(Return)** once displays the line again for another correction.
- STEP 3. If the line is now correct, press **(Return)** again.  
Pressing **(Return)** twice in a row signals the end of your corrections and sends the command line to the computer for execution.  
If the line is still not correct, go to STEP 4.  
If you wish to cancel your previous correction(s), go to STEP 5.
- STEP 4. Make another correction on the line. Go to STEP 3.
- STEP 5. Type U and press **(Return)**. This cancels the effect of the previous D, I, or R subcommand.  
Entering a U**(Return)**, then another U**(Return)** cancels *all* previous REDO subcommands and restores the command being corrected to its original form.

Now correct your last command entered, COWEM

First, use the subcommand R to replace the C with an S.

You must place the subcommand letter R *directly under* the letter to be changed. For this first correction, be sure that the cursor is already under C.

Enter:

```
RS(Return)
```

Your screen should look like this:

```
:REDO
COWEM
RS
SOWEM
-
```

Each time you make a correction, the computer shows you the command with the new edits that you made and waits for more edits.

## Note

---

The `␣` mark is used here to represent pressing the spacebar one time.

---

Insert H between S and O this way:

```
␣IH(Return)
```

Now your screen should show the following:

```
:REDO

COWEM
RS
SOWEM
IH
SHOWEM
-
```

Now replace EM with ME this way:

```
␣␣␣␣RME(Return)
```

At this point your screen should show:

```
:REDO

COWEM
RS
SOWEM
IH
SHOWEM
RME
SHOWME
-
```

The word SHOWME should be on the screen, properly spelled.

Now that the command is correct, press the return key again:

```
(Return)
```

This concludes the editing and sends the command to the computer for execution. You should now see the SHOWME display.

As mentioned earlier, the U subcommand lets you cancel or *undo*, the effects of a D, R, or I subcommand.

To see how this works, execute the SHOWTIME command and make the following corrections using the REDO command.

Replace TIME with JOB this way:

```
␣␣␣␣ddddJOB(Return)
```

Next, replace JOB with ME this way:

```
␣␣␣␣dddIME(Return)
```

Delete SHOW as follows:

```
ddd(Return)
```

Now, enter:

```
U(Return)
```

Notice that the previous edit, deleting SHOW, has been *undone* and SHOWME reappears on the screen.

At this point, you have one of three choices:

1. Continue to edit with D, R, or I, or
2. Press `(Return)` once to execute the SHOWME, or
3. Press U`(Return)` again.

Perform the third choice now by entering:

U`(Return)`

SHOWTIME returns to the screen canceling all of the previous REDO subcommands.

Press `(Return)` to execute SHOWTIME.

Your screen should show something like this:

```
:REDO
SHOWTIME
 ddddJOB
SHOWJOB
 dddiME
SHOWME
dddd
ME
U
SHOWME
U
SHOWTIME

:SHOWTIME
FRI, JUN 29, 1990 4:35 PM
:_
```

## Lesson 4 Exercises

1. The following error message appears on your screen:

UNKNOWN COMMAND NAME. (CIERR 975)

Choose from the list below the meaning of this message (more than one answer may apply):

- a. The HP 3000 does not identify what has been entered.
  - b. Your logon is not identified by the HP 3000.
  - c. You may have misspelled a command name.
  - d. Your keyboard is locked.
2. Given the following pairs of command lines and using the I, D, and R subcommands of the REDO command (or any combination thereof), correct the first command line so that it will execute the command in the second line.

- a. SHOWMW

SHOWME

- b. SOWTIME

SHOWTIME

- c. SHOWTIME

SHOWME

- d. SHJOB

SHOWJOB

3. Study the following screen example:

```
:REDO
SHOWTIME
 dd
SHOWME
 iTY
SHOWTYME
```

What would you type next to:

- a. cancel only the effects of the “iTY” subcommand correction?
- b. cancel *all* previous corrections and restore the SHOWTIME command to its original form?

---

## Lesson 5 The History Stack

Lesson 5 presents reexecuting and editing commands in the **history stack**.

The history stack is a portion of the computer's memory where the computer records your most recent commands. It keeps them there for you to reuse or correct.

Having to reenter a complex command line could be tedious.

- The LISTREDO command displays the commands in the history stack.
- The DO command reexecutes commands in the history stack.
- The REDO command edits and reexecutes commands in the history stack.

The DO command allows you to retrieve a complex command and reexecute it without having to reenter it.

The REDO command allows you to retrieve a command, edit it, and reexecute it without having to reenter the entire command.

### Note

---

In order to do the following exercises, you must start a new session. If you are logged on, log off and then log back on again. Then follow the next steps exactly.

---

```
BYE(Return)
```

```
HELLO your logon identity(Return)
```

There is a reason for having you log off and log back on. Logging off clears the command line history stack. Logging back on starts you with an empty history stack; otherwise, the history stack would be cluttered with previous commands.

### What is the command line history stack?

Each time you enter a command, your MPE/iX system saves that command in a list called the **command line history stack**. You may hear it called the **redo stack**.

In the command line history stack, each command is added at the bottom of the list as it is entered:

- Only the *20 most recent* commands are kept in the history stack. (This number can be changed.)
- The LISTREDO command shows you the commands kept in the history stack.

Enter these commands just the way they are shown here:

```
CHOWMM(Return)
```

```
SHOWTIME(Return)
```

```
SHOWJOB(Return)
```

CHOWMM produces an error message. SHOWTIME and SHOWJOB produce the displays of the kind you have seen before.

## Command history and the LISTREDO command

LISTREDO displays past commands.

Enter:

```
LISTREDO(Return)
```

Here is how your screen looks:

```
:LISTREDO
 1) CHOWMM
 2) SHOWTIME
 3) SHOWJOB
 4) LISTREDO
```

All of the commands you issued, including the LISTREDO command itself, are stored in the history stack. They appear in the order you entered them.

## The DO command

If you want to reexecute a command in the history stack *without* editing, use the DO command:

- Typing DO(Return) reexecutes the most recent command.
- Typing DO N(Return) reexecutes command number *N*.
- Typing DO never appears in the command line history stack.

For example, reexecute the third command you that entered (SHOWJOB).

Enter:

```
DO 3(Return)
```

Did the SHOWJOB command execute?

Was it added to the history stack?

Enter:

```
LISTREDO(Return)
```

How would you use DO to reexecute the SHOWTIME command?

Enter:

```
DO 2(Return)
```

Was it added to the history stack?

Enter:

```
LISTREDO(Return)
```

## History stack numbering

The history stack by default holds only the last 20 commands of your *current* session. But the history stack never runs out of room. Instead, it continues recording commands by number, dropping the oldest command to make room for the newest command.

Suppose that the letters of the English alphabet (A through Z) were commands. If you were to use all 26 commands and then execute LISTREDO, you would see something like this:

```

:LISTREDO
 8) H !!Commands A through G!!
 9) I !!(1 through 7) have "fallen"!!
 10) J !!out of the history stack!!
 11) K !!They are no longer available!!
 12) L !!to REDO or DO!!
 13) M
 14) N !!The history stack continues!!
 15) O !!recording commands in the!!
 16) P !!absolute order in which!!
 17) Q !!they were executed!!
 18) R
 19) S !!Here the 8th command is!!
 20) T !!the "oldest" and the 27th!!
 21) U !!is the most recent!!
 22) V
 23) W !!The 28th command will!!
 24) X !!"push" the 8th command out!!
 25) Y !!of the history stack!!
 26) Z
 27) LISTREDO

```

## Command history and REDO

The REDO command works like the DO command, but with one difference: REDO allows you to edit a command in the command line history stack before you reexecute it.

- Typing REDO  allows you to edit and reexecute the most recent command.
- Typing REDO *N*  allows you to edit and reexecute command number *N*.
- The REDO command never appears in the command line history stack.

Enter :

```
LISTREDO
```

You should see the sequence of commands you have issued.

To edit the command that was entered incorrectly (CHOWMM) enter:

```
REDO 1
```

Command 1 in your history stack is retrieved, ready for you to make corrections to it, using the REDO line editing methods you are now familiar with:

```

CHOWMM
RS
SHOWMM
UUUUUU
SHOWM
UUUUUUIE
SHOWME


```

Try entering:

```
LISTREDO
```

Notice that the commands you that have issued have stacked up, one after the other.

HELLO never appears in the command line history stack.

### Note

---

The D0 and RED0 commands work only with those commands still kept in the history stack. If the history stack contains commands 8 through 27, attempting RED0 3 produces an error message:

```
INVALID NUMERIC INDEX INTO THE RED0 STACK (CIERR 9073)
```

---

### Lesson 5 Exercises

1. The command history stack is:
  - a. a list of the system commands available to the user for correction or reexecution
  - b. a list of the last 20 commands entered by a user in a session
  - c. a list of all of the commands entered by a user in a session
2. Which MPE/iX command lists the commands in your command history stack?
3. Enter several commands that you know. After you have entered a couple of commands, display your command history stack to your terminal screen. Then enter a misspelled command. How is it recorded in the history stack?
4. Which MPE/iX commands *never* appear in your command line history stack?
  - a. D0
  - b. HELLO
  - c. RED0
  - d. LISTREDO
5. Given a maximum history stack size of 20, what range of numbers is displayed by LISTREDO if the LISTREDO is command number 45?
6. Display your command history stack on the screen. Use D0 or RED0 with or without specifying command numbers to reexecute several of the commands in your command history stack.

---

## Lesson 6 Communicating with Other Users

Lesson 6 presents commands to **communicate** with other users.

- commands that send messages:
  - TELL
  - TELLOP
- blocking and unblocking messages:
  - SET;MSG= blocks or unblocks messages
  - the **Quiet mode**.

### Sending and receiving messages

You can send messages to other users on your system and receive messages from them:

- To send a message to another user on your system, use the TELL command.
- To send a message to the system operator, use the TELLOP command.

---

#### Note

In order to take full advantage of the exercises presented in this module, arrange to work with a partner who also logs on to your system.

---

### Sending a message to another user

The TELL command allows you to send a message to another user (actually to another session) who is logged on to the same MPE/iX system.

First find the **session number** (*#Snnn*) or the **logon identity** of the person to whom you want to send the message.

Enter:

```
SHOWJOB(Return)
```

You should see something like this on the screen:

```
:SHOWJOB

JOBNUM STATE IPRI JIN JLIST INTRODUCED JOB NAME
#S116 EXEC 101 101 FRI 9:57A NORMA.DESMOND
#S118 EXEC 107 107 FRI 3:57P MILDRED.PIERCE
#S119 EXEC QUIET 109 109 FRI 4:02P LEDGER.ACCNTNG
#J13 EXEC 10S LP FRI 3:00P JIM.HARINGTN
#J10 EXEC 10S LP FRI 6:00A OPERATOR.SYS

5 JOBS:
 0 INTRO
 0 WAIT; INCL 0 DEFERRED
 5 EXEC; INCL 3 SESSIONS
 0 SUSP
JOBFENCE= 7; JLIMIT= 60; SLIMIT= 60
```

Norma Desmond is using session 116. You could send a message to her in two ways, both using the TELL command:

- to her session number:

```
TELL S116 A NOTE TO YOU.(Return)
```

or

- to her logon identity (often—but not always—a person’s name):

```
TELL NORMA.DESMOND CAN YOU GET ME THE JAMES FILE?(Return)
```

Use either method to send a message to your partner.

### Note

---

You cannot send a message to a job.

---

## What the receiver sees

The person who receives the message sees something like this on the screen:

```
FROM/S118 MILDRED.PIERCE/CAN YOU GET ME THE JAMES FILE?
```

Notice that the sender’s session number and logon appear on the screen indicating the sender of the message. In this example, the sender is Mildred Pierce, working in session 118.

### Note

---

TELL messages are transmitted when the (Return) key is pressed. Lengthy messages that are longer than a terminal screen line wrap around (automatically continue) to the next line (you can type as many as 276 characters, including spaces, *after* the TELL command).

---

Have a partner send a message to you.

## Blocking messages

There might be times when having a message appear on the screen is annoying. In some circumstances, the message might actually interfere with your work.

You can tell the computer to prevent messages from appearing on your terminal.

To block messages, use the SET command. Be sure to enter the command exactly this way:

```
SET;MSG=OFF(Return)
```

Try it.

Enter:

```
SHOWJOB(Return)
```

```
:SHOWJOB
```

```
JOBNUM STATE IPRI JIN JLIST INTRODUCED JOB NAME
#S116 EXEC QUIET 101 101 FRI 9:57A NORMA.DESMOND
#S118 EXEC 107 107 FRI 3:57P MILDRED.PIERCE
#S119 EXEC QUIET 109 109 FRI 4:02P LEDGER.ACCNTNG
#J13 EXEC 10S LP FRI 3:00P JIM.HARINGTN
#J10 EXEC 10S LP FRI 6:00A OPERATOR.SYS
```

```
5 JOBS:
```

```
0 INTRO
0 WAIT; INCL 0 DEFERRED
5 EXEC; INCL 3 SESSIONS
```

```
0 SUSP
JOBFENCE= 7; JLIMIT= 60; SLIMIT= 60
```

In this example, both NORMA.DESMOND and LEDGER.ACCNTNG have used the SET command to block messages from appearing on their terminals. Both users' terminals are now in Quiet mode.

Your terminal should be in Quiet mode. Does SHOWJOB confirm this?

If you try to send a message to someone whose terminal is in Quiet mode, you get a message from the system:

```
S116 NORMA.DESMOND NOT ACCEPTING MESSAGES.
```

## Unblocking messages

To permit messages to get through to your terminal again, enter this:

```
SET;MSG=ON(Return)
```

The ;MSG=ON part of this command line takes your terminal out of Quiet mode.

Practice the TELL command with a partner on your system. Get in and out of terminal Quiet Mode. Send messages to one another while in each mode (on and off). Use the SHOWJOB command to keep track of which mode you and your partner are in.

## Message to the system operator: TELLOP

There may come a time when you need to send a message to the system operator who supports your HP 3000. The TELLOP command sends a message to the system operator's terminal (| system console) and to nowhere else. For that reason, you do not have to specify a session number or a logon identity when you use TELLOP.

### Note

Your system operator is obliged to respond to TELLOP messages. That takes up the operator's time. Do not send messages to the operator unless you really need to do so.

Instead, just study the examples below.

If you enter at the prompt:

```
TELLOP WHAT TIME IS SYSTEM BACKUP TONIGHT?(Return)
```

what the system operator would see on the console screen is something like this:

```
16:47/#S116/19/FROM/NORMA.DESMOND/WHAT TIME IS SYSTEM BACKUP
TONIGHT?
```

TELLOP supplies more information to the receiver than the TELL command:

|       |                                                                                                              |
|-------|--------------------------------------------------------------------------------------------------------------|
| 16:47 | The time the message was sent (in 24-hour time).                                                             |
| #S116 | The session from which the message was sent. (A message can be sent from a job, and then it might be #Jnnn.) |
| 19    | The sender's identification number, also called process identification number (PIN).                         |

Your system operator can also use the TELL command to send a message to you. If you get one, it will look something like this:

FROM/S581 CONSOLE.SYS/SYSTEM BACK UP AT 8:00 PM.

## **On the lookout**

At any time, the operator can deliver a terminal screen message to all users logged on to the system. These messages usually contain important information regarding your use of your MPE/iX system.

### **Warning messages**

The system operator can send you an urgent message at any time. A warning message appears on the screen no matter what. Using SET;MSG=OFF does not block a warning message from the operator.

If you get a warning message from the operator, it will look something like this:

```
OPERATOR WARNING: SYSTEM COMING DOWN IN 15 MINUTES. PLEASE LOG OFF.
```

Always pay close attention to messages from the system operator. If you ignore messages from the operator, you risk losing or corrupting some or all of your work in progress.

## Lesson 6 Exercises

### Note

---

For the following exercises, arrange with a fellow user on your system to send messages back and forth with you. This user must know how to execute the TELL and SET commands.

---

1. List on your screen all of the users who are currently working on your HP 3000 system.
2. Identify your partner for this exercise from this list. Send that user the following message (use your own name);

This is a test. Please send *name* a reply message.

Wait for a response.

3. Have your partner get into Quiet mode. Transmit a message to him or her. What happened on your screen?
4. Get into and out of Quiet mode. After each step, check to see if you were successful.
5. For the following statement, answer true or false.

Once in Quiet mode, messages from all other users of the system to you are suppressed from appearing on your terminal screen.

### Note

---

Have you taken a break to relax? This might be a good time.

---

---

## Lesson 7 More Information

Lesson 7 presents useful sources of information.  
Sometimes things simply do not go the way that you that expect.

### Logon problems

One day when you try to log on, you may get this message:

CAN'T INITIATE NEW SESSIONS NOW

The most likely explanation is that the system operator or system manager has reduced the **job/session limit** on your system. The operator or manager can restrict the number of jobs and sessions that the computer can accept.

Your system might be due for periodic maintenance and backup. The operator or manager might be using a program that requires large amounts of the computer's capacity. At such times, having too many jobs and sessions competing for the computer's attention could slow everything down.

If you get that message when you try to log on, talk to your account manager or your system operator. Ask when the job/session limit will be raised.

### General user references

Here are two good sources of information about all of the commands you are likely to use.

- the *MPE/iX Commands Reference Manual* (32650-60002)
- the online **help facility** (see lesson 8)

#### The reference manual

*MPE/iX Commands Reference Manual* (32650-60002)

Your system manager may have a copy of this. If not, the manual can be ordered through your Hewlett-Packard sales representative.

- It provides considerable detail about commands and how to use them.
- It has a detailed reference to all of the standard commands available in MPE/iX; they are listed alphabetically.
- It presents command files and user-defined commands (UDCs).

Module 6 in this tutorial introduces command files. UDCs are presented in the *900 Series HP 3000: Advanced Skills* course.

#### The online help facility

The help facility comes with your MPE/iX system. It also presents information on all of the system commands.

## **Lesson 7 Exercises**

1. Give a reason why you would see the message `CAN'T INITIATE NEW SESSION NOW` on your terminal screen.
2. List two resources where you can get information regarding HP 3000 commands that you are likely to use.



## Getting around in help

Suppose that you want to learn about the REDO command.  
Start the help facility again.

```
HELP(Return)
```

The help prompt at the lower left corner of your screen tells you that the help facility is waiting for your instruction.

Enter:

```
REDO(Return)
```

What happens?

You will see this on the screen:

```
:REDO

 Allows the user to edit and reexecute any command still
 retained in the command line history stack. (MODIFIED)

SYNTAX

 :REDO [[CMD=]cmdid][[;EDIT=]editstring]

NOTE:

 This command follows the more flexible MPE/iX command line syntax.
 Refer to the MPE/iX Commands Reference Manual (32650-60002).

KEYWORDS: PARMS,OPERATION,EXAMPLE
>
```

### Syntax

Syntax is a formal way of diagramming a command and all of its parameters. The rest of the help entry on each command is devoted to explaining the command, its parameters, and how to use them. You can study syntax diagrams in the *MPE/iX Commands Reference Manual* (32650-60002) and in the HELPSTUDY portion of the help facility.

### The help keywords

A **keyword** tells you about other information that is available on the command you that want to look up:

- The PARMS keyword (short for parameters) shows you definitions of the parameters for a command.

Enter:

```
REDO PARMS(Return)
```

- The OPERATION keyword explains the operation of the command.

Enter:

```
REDO OPERATION(Return)
```

- The EXAMPLE keyword shows you one or more examples of the command as it is used.

Enter:

```
REDO EXAMPLE(Return)
```

If you enter REDO(Return), you can press (Return) to jump forward to PARS.

Press (Return) again and jump forward to OPERATION, and then to EXAMPLE.

A final press of (Return) takes you to the next command entry.

Try looking up a command (SHOWJOB, for instance). Then press (Return) one or more times to see how it works.

### The help facility screen

Whenever you ask for information on a command (*without* specifying a keyword) you will see:

- the command name
- a definition of the command
- the command **syntax**
- any special notes about the command
- keywords for the command

A more detailed discussion of commands and their syntax can be found in the *900 Series HP3000: Advanced Skills* course, and in the help facility itself, under the entry HELPSTUDY.

### The help facility's online tutorial

The help facility has a self-paced, online tutorial called HELPSTUDY.

To get to HELPSTUDY, you must first be in the help facility. Once there, enter:

```
helpstudy(Return)
```

Here is what you see:

```
0000 0000 000 00
 00 00 00 0000
 00 00 0000000 00 0000000 00
00000000 00 00 00 00 00
 00 00 00000000 00 00 00 00
 00 00 00 00 00 00 00 0000
0000 0000 0000000 0000 0000000 00
 00
 00
This is a study aid for 00
beginners in MPE XL 000
```

CHOOSE ONE ... AND DO THIS ...

- |                                   |                                                                    |
|-----------------------------------|--------------------------------------------------------------------|
| 1) FIND A COMMAND                 | Enter its name and press <RETURN >                                 |
| 2) QUIT HELP                      | Enter E (or e) and press <RETURN >                                 |
| 3) SHOW ME MORE<br>ABOUT HELP AND | Simply press <RETURN ><br>...useful if you are new to HP computers |



## Module 2: Computer Overview

---

Module 2 introduces hardware, software, and the keyboard:

|          |                   |
|----------|-------------------|
| Lesson 1 | Computer Hardware |
| Lesson 2 | Software          |
| Lesson 3 | The Keyboard      |
| Lesson 4 | Any Problems?     |

---

### Challenge Test

Challenge your current computer skills by taking the following Challenge Test. Use your test results to select those lessons that you should study and those that you skip.

1. What is hardware?
2. What are peripherals?
3. What is software?
4. What is MPE/iX?
5. Give the meaning of the abbreviation MPE.
6. Name three types of programs that can run on the HP 3000.
7. TRUE/FALSE The terminal keyboard is exactly the same as a standard typewriter keyboard.
8. What is meant when your terminal is in local mode? How does this differ from remote mode?
9. Give three ways of troubleshooting a terminal that is not responding.

---

## Lesson 1 Computer Hardware

Lesson 1 presents the physical parts of the computer.

- the computer **hardware**
- the computer **peripherals**
- the **system console**

Computers consist of **hardware** and **software** components:

**Hardware**      The physical equipment: integrated circuit boards, disk drives, terminals (the screen and keyboard together), tape drives, printers, and plotters.

**Software**      The **programs** that instruct the hardware to do something (Lesson 2).

### Hardware

Hardware is all of the electronic equipment that a computer includes. If you can touch it, pick it up, or move it around, it is hardware.

A computer consists of a central unit, called the **central processing unit** (CPU), and external devices, called **peripherals**.

**CPU**              The central processing unit is the “brain” of the computer. Every instruction given to the computer passes through the electronic circuits of the **CPU**. When you program the computer to add two numbers together, the arithmetic is performed in the CPU. When you want something sorted, the CPU controls the task from start to finish.

**SPU**              The system processing unit contains all of the components of the computer system including the central processing unit. The SPU does not refer to the system console or any other single, peripheral device.

**Peripherals**      Peripherals are all of the other hardware connected to the CPU. Peripherals are often referred to as peripheral **devices**.

The computer assigns to each device an identifying number called an **LDEV number**. This number is the “address” (of your terminal, for instance) that the computer uses when it sends and receives information.

### Peripherals

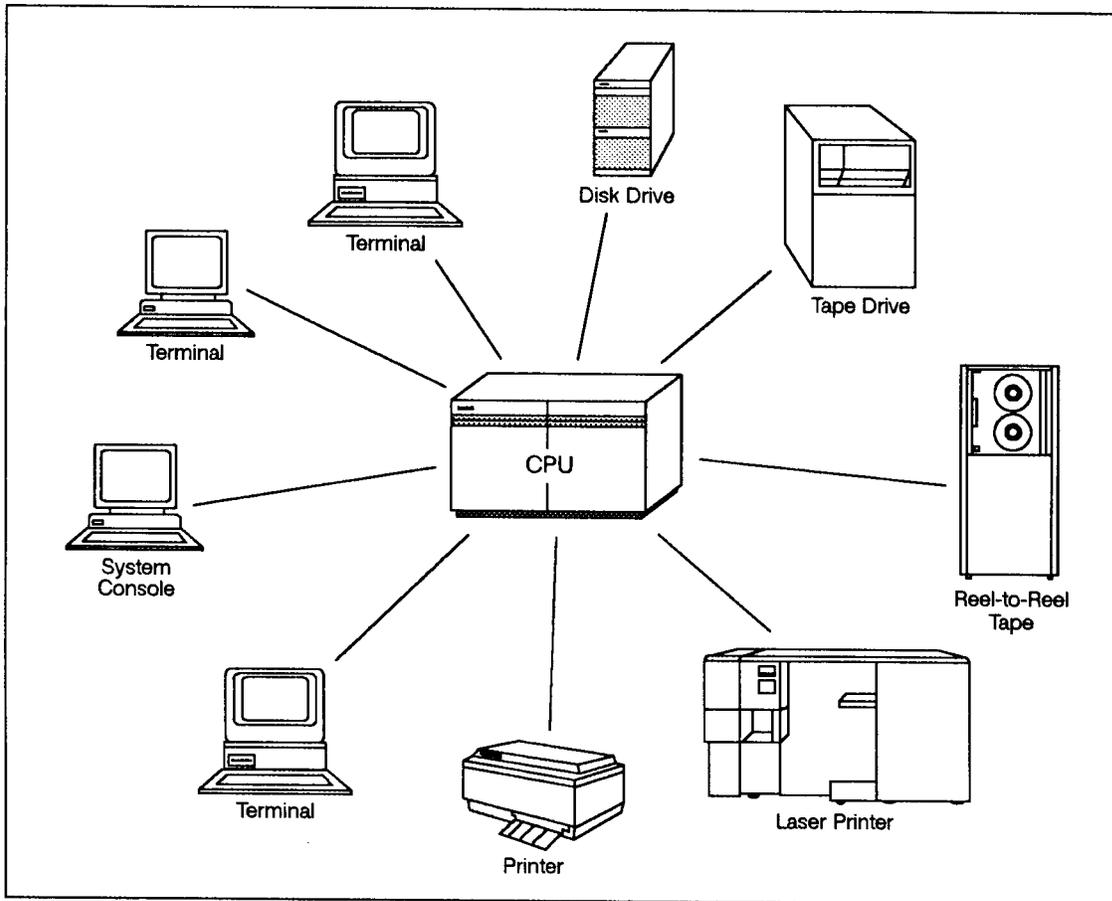
There are five classes of peripherals that may be connected to a standard HP 3000 system:

**Terminals**      Your **terminal** (the screen and keyboard) gives you a way of communicating with the computer. You can enter commands or run programs and see the results.

**Disk Drives**      Information and programs that are not being used by the SPU are stored on **disk drives**. The information could be the text of a letter, information in a database, or a program that you might use every day. There might be one or many disk drives attached to your computer.

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tape Drives    | <p>Information on a disk drive can be retrieved quickly. Information that is not accessed frequently but must be kept for a long time is stored on <b>magnetic tapes</b>. Tape drives are used to <b>store</b> information from the disk onto the tape.</p> <p>The computer takes longer to retrieve information from a tape. The advantage of tape lies in safely keeping <b>backup</b> copies of vital information or programs. If information is ever lost from the computer, it can be <b>restored</b> to the system from the tape(s); however, the restored version of a file might not be the most up-to-date version of the file. It might be an <i>earlier</i> version.</p> <p>It is common practice for system operators to store all of the disk files on the computer to back up tapes at regular intervals. Thus, if you lose or accidentally erase an important file, there is a reasonable chance of having that file restored to the computer.</p> |
| Printers       | <p>Printers come in all shapes and sizes. Some are designed for extremely high speeds. Some are designed for very high-quality, finished documents.</p> <p>There may be one or many printers attached to your computer. If there is only one, it may be referred to as the <b>line printer</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| System Console | <p>The <b>system console</b> is a special terminal that your system operator uses almost exclusively to control the day-to-day functioning of the computer. The system console is usually located close to the CPU.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

The following figure shows a typical computer system.



TG20510-02-1

**Figure 3-1. A Typical Computer System**

**Lesson 1 Exercises**

1. What is the name of the central unit of the computer?
  
2. Name four peripherals that make up a basic HP 3000 system.
  
3. What is the name of the terminal that is used by the system operator to control the entire HP 3000 system and that is usually located close to the CPU?

---

## Lesson 2 Software

Lesson 2 presents the computer software (programs).

- the **operating system**
- other programs on the system
- the **command interpreter**

Software are the programs that control everything the computer does for you. The software makes the computer run and determines whether the computer acts as a text processor, graphic artist, office manager, or performs any of the hundreds of tasks that a computer can do for you.

### Operating system

All computers come with special software called an **operating system** that oversees and directs the processing of all programs. The operating system for the HP 3000 family of computers is called **multiprogramming executive (MPE)**.

- MPE/iX is the operating system for the 900 Series HP 3000 computer systems, the system that you are working with, and the newest in the HP 3000 family of computers. The iX stands for integrated POSIX. That means that your computer system is able to work with many programs created for the UNIX\* operating system, through an enhancement that Hewlett-Packard calls POSIX.

\*UNIX is a registered trademark of AT&T in the U.S.A. and in other countries.

- The parent version of the operating system is called MPE V. It runs on earlier HP 3000 computer systems (Series 40 through 70).

The V refers to the fifth major revision of the software.

### Sample programs

Some of the types of programs that can run on the HP 3000 are briefly described here:

Electronic Mail Permits users of one or many systems to send and receive messages online.

Word Processing Permits users to create, edit, arrange, and print letters, memos, and other kinds of text that people deal with in their day-to-day work.

Data Processing Permits users to store, retrieve, integrate, and manage vast amounts of information. Keeping and managing a payroll is one familiar sort of data processing. Data processing might also involve mathematical and statistical computations.

Form Creation Putting information into the database is made easier if information is entered and retrieved by filling in standard forms on the terminal screen. There are programs just for designing the forms that are used in data entry.

Games There are thousands of games that run on computers, and more are being created every day.



## Lesson 3 The Keyboard

Lesson 3 presents some of the special keys on the terminal keyboard and how they are used.

The terminal keyboard appears similar to the keyboard of an electric typewriter. Yet, only the arrangement of the alphabetic keys is identical.

This module does not attempt to define every key on the keyboard. Instead, it concentrates on those keys that are of most immediate value to the new user and those that may differ on various terminals.

You should read this lesson while you are sitting at a keyboard for the HP 2392 series or HP 700/92 and HP 700/94 terminals.

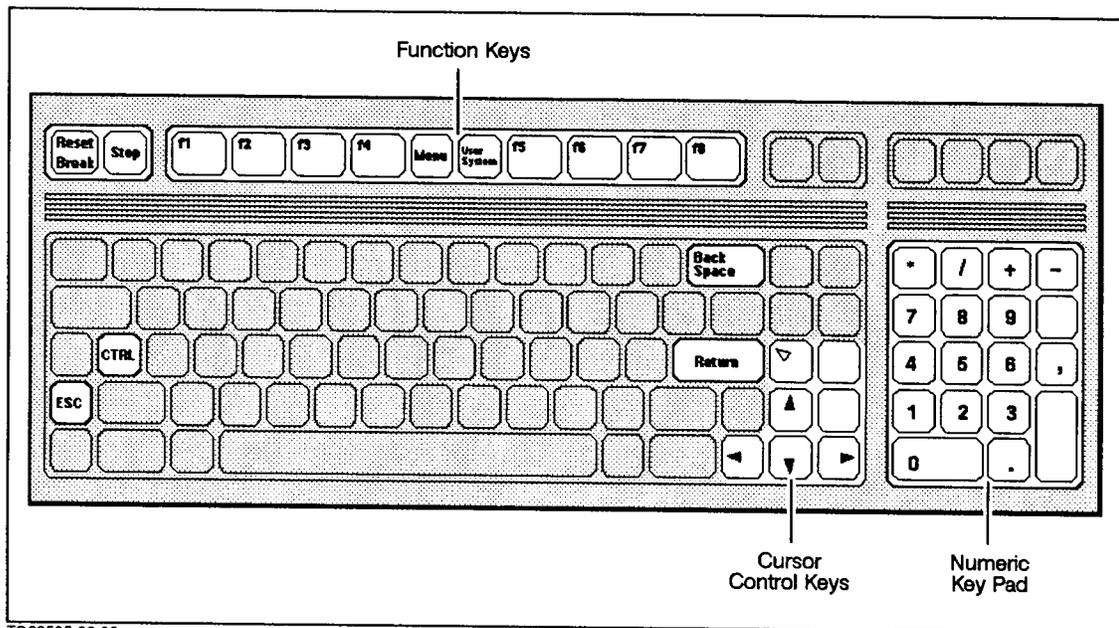
### Note

If you are using an HP Vectra™, an HP 150™, or some other terminal connected to an HP 3000 computer, your keyboard looks different, and some of the keys behave slightly differently.

*Read* the documentation that came with your terminal to get more information about the operation of your keyboard.

The summary chart at the end of this lesson lists some of the special keys that might be on your keyboard. It also gives a brief description of their function.

Look at your keyboard now.



TG20535-02-02

Figure 3-2. HP 2392 or HP 700/92 Keyboard

Figure 2-2 highlights the keys in the following discussions.

## The return key

If you are accustomed to using a typewriter, you will discover that the **Return** key on a computer behaves quite differently. On some terminals, this key is labeled **Enter**.

On a typewriter, **Return** advances the paper one line and returns the carriage (or the typing element) to start a new line.

The **Return** key on a computer does much more than it does on a typewriter.

Sometimes labeled **Enter**, the **Return** key is the keyboard's main communicator with the computer. What you type appears on the screen. (There are a few exceptions. Passwords that you enter in response to a challenge from the system do not appear on the screen when you type them.)

But until you press **Return**, the computer is unaware of what you have typed.

Pressing **Return** sends to the computer those characters that are displayed on the screen and held in the terminal's memory area.

The **Return** key:

- after a command name: tells the computer to execute the command
- after a program name: tells the computer to start the program
- in a document: ends one line of text and starts a new line
- alone: scrolls the screen (and whatever appears there) upward, one line at a time

## Backspace key

The backspace key **←** or **Backspace** (figure 2-2) and the arrow left key **⬅** (on the keypad) are two different keys, and they perform two different tasks. You will learn about the arrow keys later.

When you use **←**, MPE/iX moves the cursor backward and *removes* characters from the terminal's memory one character at a time. Remember that the computer does not receive what you type until you press **Return**.

Using **←** prevents the computer from ever receiving the characters that you backspace over, but the characters still appear on the screen.

As soon as you stop pressing **←**, the computer waits for you to type in new material over the old letters that you deleted from the terminal's memory.

### Note

---

This deleting function of **←** is invisible to you. The letters you backspace over still *appear* on the screen.

But don't be fooled.

When you use **←**, the letter that you see *above* the cursor has been deleted. And so have all of the letters that appear to the right of the cursor.

---

- You can type over those letters if you want to change or reenter them.

- You can press **(Return)** to enter the letters that remain (those to the *left* of the cursor).

## Numeric keys

The number keys (**(1)**, **(2)**, **(3)**, **(4)**, and so on, above the letter keys and across the top of the keyboard) work the way they do on a typewriter.

There are three keys that are likely to cause confusion for some users who are accustomed to typewriters or who have never used a computer:

- lower case “L”:

The “1” key cannot be used to represent the number 1 (one) on the computer. They are not the same. If you want the number 1 (one), press **(1)**. If you want the letter “l” press **(L)**.

- The letter “O”:

The letter “O” or “o” cannot be used to represent the number 0 (zero). They are not the same. If you want zero, press **(0)**.

- The **(Caps lock)** key.

This key turns all of the *letters* into capitals, but it does not force the shift of nonletter keys. For instance, when **(Caps lock)** is set, “a” is always rendered as “A”; however, “9” is not rendered as “(” (*left parenthesis*) but as “9”. To get “(” you must hold down **(SHIFT)(9)**.

## Numeric keypad

The number keys on the **keypad** on the right side of the keyboard also enter numbers. These keys are arranged like an adding machine or calculator. They are used for high-speed entry of numeric data (into a database, for instance).

On a Vectra™ keyboard, the keypad numbers are activated and deactivated by pressing the **(Num lock)** key.

## Cursor control keys

These keys allow you to move the cursor around on the terminal screen:

- (▲)** Moves the cursor up toward the top of the screen.
- (▼)** Moves the cursor down toward the bottom of the screen.
- (▶)** Moves the cursor toward the right of the screen.
- (◀)** Moves the cursor toward the left of the screen.
- (⏴)** Moves the cursor as far upward and to the left as it can go. How far it can go depends on the limit of your terminal’s screen memory.
- (Shift)(⏵)** Moves the cursor as far down as it can go. How far it can go depends on the limit of your terminal’s internal memory.
- (Prev)** (If there is one) moves the cursor up by one screenful of material—the cursor goes up, the material seems to go down. **(Prev)** is labeled **(Pg Up)** or **(Page Down)** on some keyboards.

**Next**

(If there is one) moves the cursor down by one screenful of material—the cursor goes to row one, column one of the next screenful of text. The material seems to go up. Labeled **Pg Dn** or **Page Down** on some keyboards.

---

**Note**

You will not use the arrow keys for any of the modules in this tutorial. The arrow keys are recognized by some programs and not by others.

Attempting to use the arrow keys in a program that does not recognize them can lead to confusion.

---

**Function keys**

The keys numbered *f1*, *f2*, *f3* through *f8* at the top of your keyboard are called **Function keys**. These keys correspond to the boxes at the bottom of your terminal screen. These function keys are also called **softkeys** because their function is not fixed (hard-coded), but can be changed, by the system itself or by an advanced user, to suit a particular purpose.

You will be learning more about the function keys in lesson 4, “Any Problems?”

**Other special keys**

The following is a summary of other special keys found on most Hewlett-Packard terminal keyboards:

**Table 3-1. Other Special Keys**

| Key                  | Function                                                                                                     |
|----------------------|--------------------------------------------------------------------------------------------------------------|
| <b>Clear Display</b> | Clears the screen (and its memory) of anything displayed below and to the right of the cursor position.      |
| <b>Clear Line</b>    | Erases characters from the cursor position to the end of the line.                                           |
| <b>CTRL</b>          | Used with some special commands.                                                                             |
| <b>Del/Esc</b>       | Deletes a character/Sends a special command.                                                                 |
| <b>Delete Char</b>   | Erases a character at the cursor position.                                                                   |
| <b>Delete Line</b>   | Erases the line in which the cursor is found.                                                                |
| <b>Extended Char</b> | Switches special character recognition on and off.                                                           |
| <b>Insert Char</b>   | Inserts a character at the cursor position.                                                                  |
| <b>Insert Line</b>   | Inserts a line at the cursor position.                                                                       |
| <b>Menu</b>          | Switches the softkey boxes on and off.                                                                       |
| <b>Next</b>          | Moves the display forward by one screenful.                                                                  |
| <b>Prev</b>          | Moves the display backward by one screenful.                                                                 |
| <b>Print/Enter</b>   | Prints what is on the screen. You must press <b>SHIFT</b> <b>Print/Enter</b> to print what is on the screen. |
| <b>Reset/Break</b>   | Interrupts the computer.                                                                                     |
| <b>Select</b>        | In <i>some</i> applications tells the computer to “do” or “choose” this (whatever is highlighted).           |
| <b>STOP</b>          | Stops and starts the scrolling on the screen.                                                                |
| <b>User/System</b>   | Switches between user-defined and system-defined softkeys.                                                   |

**Note**

**BE AWARE:**

This discussion deals *only* with the way MPE/iX uses the keys on the HP 2392, HP 700/92, and HP 700/94 terminals. Other programs and other terminals may handle the key functions differently. If you run into difficulties, consult the documentation for the program or for the terminal that you are using.

**Lesson 3 Exercises**

1. Explain how the **(Return)**/**(Enter)** key on a terminal is different from the RETURN key on a typewriter.
  
2. Explain how the backspace, (**←**), is different from the backspace key on a typewriter.

---

## Lesson 4 Any Problems?

Lesson 4 presents sources of information and some hints for solving common terminal problems.

Sometimes things don't go as expected.

### Sources of information

Two places provide detailed information about the way your terminal works:

- The documentation or instructions that came with your particular terminal.
- Other people. Often they are the fastest route to an answer.

When you need help with the computer, and you can't seem to find the answer on your own, certain persons can help you:

**Account Manager** If it's a "how to" question, your **account manager**, sometimes called the AM, probably has an answer. Your AM creates groups in your account and gives you your passwords. (The importance of accounts and groups is presented in module 4.)

**System Operator** The **system operator** (or operator or manager of operation) has charge of the console at any particular time of the day. If your terminal locks up on you and nothing that you do seems to help, talk to your operator. If you can't log on, or if your session or job is behaving oddly, the operator can probably find the answer.

**System Manager** The **system manager** (or manager) is responsible for ensuring that the computer is set up to meet the needs of the users. Unless you have problems that need some modification of the system itself, you probably won't need the help of your system manager.

### Terminal not responding

Nothing appears on the screen.

- Some terminals take a moment or two to warm up and run through a self-test.
- Press the **Return** key a couple of times.

If nothing appears on the screen after 15 or 20 seconds, do some investigating:

- Is the terminal plugged in and turned on?
- Are all of the right cables attached?
- Is your connector line working?

If you are not sure how to turn on and connect your terminal, talk to the operator, or consult the documentation that came with your terminal.

## Terminal on but not responding

You can type on the keyboard. Letters and numbers appear on the screen, but it looks as though the computer is not responding.

The terminal is probably in the wrong **mode**. The terminal has two working modes:

- |        |                                                                                                                                                                                                                                                                                   |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Local  | In <b>local mode</b> , your terminal is completely on its own. It does not interact with the computer. What you type appears on the screen, but none of it goes to the computer for processing. This is sometimes called <b>standalone mode</b> . You do not want local mode now. |
| Remote | In <b>remote mode</b> , your terminal talks directly to the computer and shows you the computer's responses. You want remote mode.                                                                                                                                                |

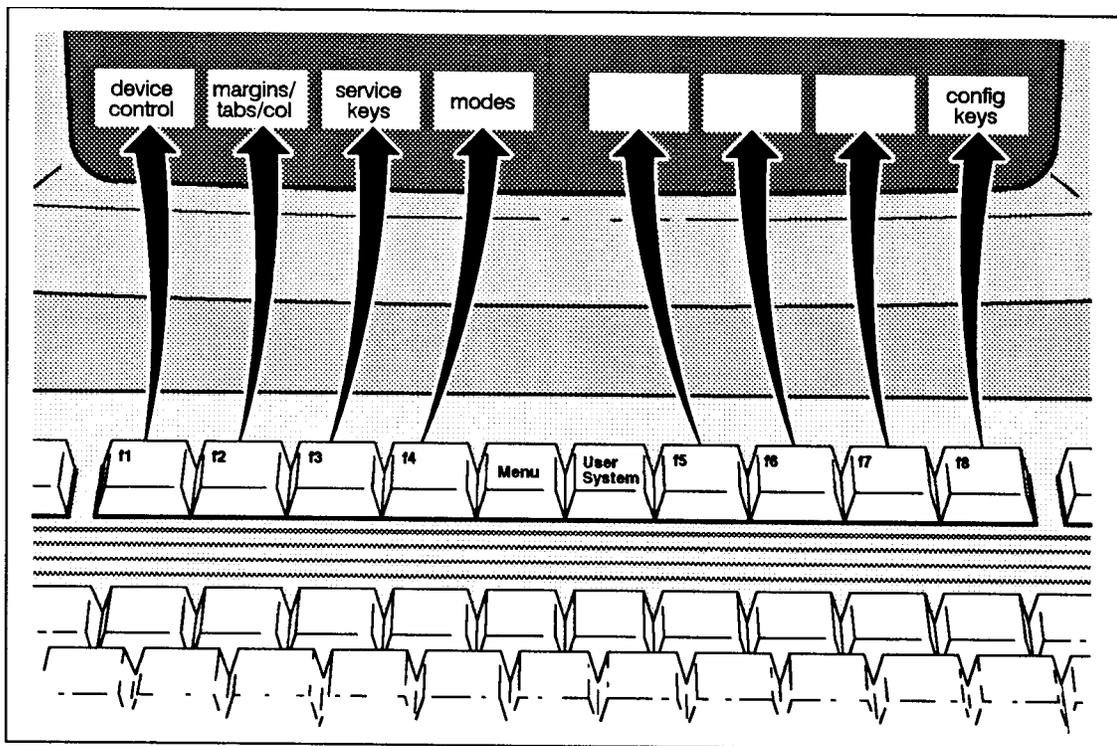
## Changing modes with the softkeys

### Note

If you have an HP personal computer connected to an HP 3000, your softkeys may behave differently from the discussion that follows. If you are not using an HP 2392 or an HP 700/92 Series terminal, consult the documentation that came with your terminal.

Log on again. Enter:

```
HELLO your logon identity(Return)
```



TG20535-02-03

Figure 3-3. Video Screen at Logon

Note that these boxes correspond to the **f1** through **f8** keys of your keyboard in figure 2-3.

**Note**

Some keyboards may have an additional set of function keys running vertically on the left side; some have twelve function keys instead of eight. For this exercise, focus on the function keys **f1** through **f8** running horizontally across the top of your keyboard.

On the HP 2392 and the HP 700/92 Series terminals, two more keys sit between **f4** and **f5**:

```
User
Menu System
```

**Initial display**

Compare the screen to the illustration in figure 2-3.

If the boxes along the bottom of the screen look different from those in figure 2-3, press the key labeled like this:

```
User
System
```

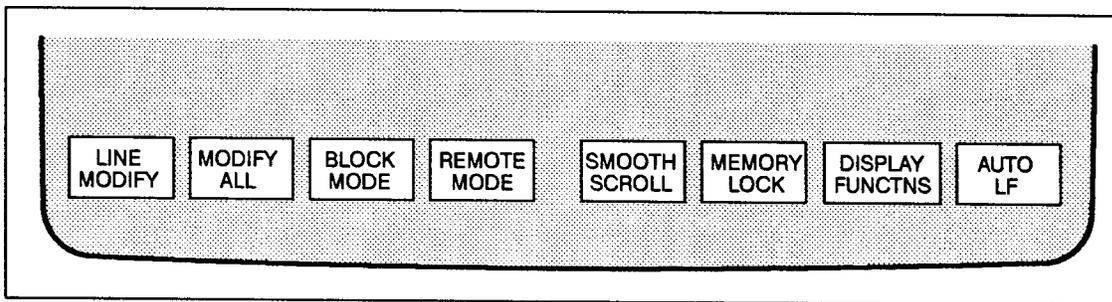
That should make the row of boxes on the screen look like the ones shown in figure 2-3. (If you are having trouble, talk with your supervisor or account manager.)

Right now, concentrate on the fourth box on the screen:

```
modes
```

It is controlled by the key labeled **f4**. Press the key labeled **f4**.

The eight boxes across the bottom of the screen should change to look like those in figure 2-4.



TG20535-02-04

**Figure 3-4. Softkey Function Display**

Notice that some keys change the label (and purpose) of other keys.

Once again look at the fourth box:

```
REMOTE
MODE
```

- Does an asterisk \* appear inside the box, like this?

```
REMOTE
MODE*
```

If it does, your terminal is already in remote mode and is ready to communicate with the computer. Do not do anything more.

- Or is the asterisk missing?

```
REMOTE
MODE
```

If the asterisk is missing, you are in local mode—the terminal is cut off from the computer. You do not want that. Press the key labeled **(f4)**.

That should change box 4 to this:

```
REMOTE
MODE*
```

- An asterisk (\*) appearing in a softkey box tells you that the function displayed there is switched On.
- If no asterisk appears in a softkey box, you know that the function displayed there is switched Off.

When you have set your terminal for remote mode, press the key labeled:

```
User
System
```

You should see the original eight boxes that you first saw when you logged on.

## The other softkey keys

You may have guessed already that you can use the softkeys to make other changes to other boxes.

These keys and the boxes they affect allow you to change the **configuration** of your terminal. The configuration settings tell your terminal how to behave:

- what to display
- how to display it
- whether to communicate with the computer
- how to communicate with the computer
- how to communicate with other devices

If you want to know more about configuring your terminal, or if you feel you need to change the configuration, consult the documentation that came with your terminal.

---

**Note**

Unless there is very good reason for doing so, *do not* change the configuration of your terminal without first getting advice from your account manager or your supervisor.

Without knowing the effect of the changes you are making, you could **configure** your terminal in a way that makes it completely unworkable for you. You can recover from configuration mistakes, but it's a nuisance.

---

**Is the terminal locked up?**

Sometimes the terminal screen seems to “freeze up.”

- You type something and press **(Return)**, and nothing happens.
- You type and nothing at all happens.

You may have a **hung** terminal. There are remedies.

It is difficult to say *exactly* what happened. Most likely you accidentally pressed **(CTRL)** at the wrong moment, or you pressed a combination of keys that sent an unexpected signal to the computer.

**Remedies**

First, wait a few seconds. The computer may be busy executing a high-priority program.

Next, ask others on your system whether they are having the same problem. (There might be a power shutdown or a system shutdown.)

If you still have no solution to the problem, there are four remedies you can try, *in this order*:

1. **Soft reset.**
2. **Hard reset.**
3. Turn the terminal off, then back on.
4. Talk to your account manager or supervisor.

---

**Note**

If you have an HP 150<sup>TM</sup>, an HP Vectra<sup>TM</sup>, or some other personal computer connected to an HP 3000, some of your keys may behave differently from the discussion that follows. If you are not using an HP 2392 or an HP 700/92 Series terminal, consult the documentation that came with your terminal.

---

**Soft reset**

Always try this first. It causes the least interruption to your work.

Hold down **(Shift)**. While holding it down, press **(Reset/Break)**.

This reset does not work if you press one key, let go, and then press the other key.

Did anything change? Using a soft reset is the “gentle” way to regain control of a hung terminal. You are still logged on, and all of your work is intact.

## Hard reset

If a soft reset does not work, a hard reset may.

Press **(Shift)** and **(CTRL)** simultaneously. While you hold them down, press **(Reset/Break)**.

Try that now.

Did the screen clear? Did you find the cursor in the upper left corner of the screen?

Press **(Return)** to get the system prompt back.

## Turn the terminal off

If a soft reset and a hard reset have no effect, turn off the terminal and turn it back on. If you have trouble finding the power switch, consult the manual for your terminal or ask someone who uses the same kind of terminal.

## Talk to your account manager

If your terminal is still hung, it is time to talk to your account manager or your system operator.

## Remedies - the pros and cons

|             |                                                                                                                                                                                                                                                     |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Soft Reset  | This never disturbs any of the work that you are doing, but it may not solve the problem.                                                                                                                                                           |
| Hard Reset  | This <i>might</i> erase your current work (what you have been doing on the screen), but it usually frees a hung terminal. It cannot be used to log off.                                                                                             |
| Turn Off/On | This <i>might</i> erase some of your work. If your terminal is not wired <i>directly</i> into your computer, this method could log off your session. It usually frees a hung terminal. Your operator knows whether your terminal is wired directly. |
| Operator    | The operator may have to <b>abort</b> your session (force a logoff). If your operator cannot remedy the “hang,” your terminal (or the line connecting it to the computer) may need some kind of maintenance.                                        |

## Note

---

For the work that you will do in this tutorial, soft reset and hard reset work equally well and cause you no serious interruptions, although a hard reset does require you to press **(Return)** to get back the system prompt.

---

**Table 3-2. Soft Reset Versus Hard Reset**

| <b>EFFECT</b>                   | <b>HARD<br/>RE-<br/>SET</b> | <b>SOFT<br/>RE-<br/>SET</b> |
|---------------------------------|-----------------------------|-----------------------------|
| Unlocks a hung terminal         | X                           | X                           |
| Clears the screen               | X                           |                             |
| Can erase terminal memory       | X                           |                             |
| Does not erase any of your work |                             | X                           |
| You stay logged on              | X                           | X                           |
| Not the same as logging off     | X                           | X                           |

**Lesson 4 Exercises**

1. Match the following terminal key(s) with the function(s) listed below.

Numeric Keypad

Cursor Control Keys

Function Keys

Break /  Reset Keys

- a. These keys, labeled  f1 through  f8, perform different functions depending on what is displayed in the lighted boxes at the bottom of your terminal screen. These keys are also user programmable.
  - b. These keys may be used when your keyboard does not respond.
  - c. These keys let you move the cursor around the terminal screen.
  - d. These keys are arranged similarly to a standard office calculator and are used for high-speed entry of numeric data.
2. For each item below, put an X in the appropriate column(s) to show whether the statement is a characteristic of a hard reset, a soft reset, or both.

|                                                                                                                                                                                                                                                                                                                                                | <b>Hard<br/>Re-<br/>set</b> | <b>Soft<br/>Re-<br/>set</b> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|-----------------------------|
| 1. Can be used to Unlock your keyboard.<br>2. Destroys information on your screen.<br>3. Does <i>not</i> destroy information on your screen.<br>4. The “gentle” way of regaining control of a hung terminal.<br>5. After this has been performed, the user still remains logged on.<br>6. Should <i>not</i> be used as a means of logging off. |                             |                             |

3. In which mode is your terminal when it is communicating directly with the HP 3000?

4. In which mode is your terminal when it is acting as a standalone terminal?



## Module 3: Creating Text Files

---

Module 3 introduces the tool for creating and editing files.

|          |                      |
|----------|----------------------|
| Lesson 1 | Using EDIT/3000      |
| Lesson 2 | File Names           |
| Lesson 3 | Editing a Text File  |
| Lesson 4 | More About EDIT/3000 |

---

### Challenge Test

Challenge your current computer skills by taking the following Challenge Test. Use your test results to select those lessons that you should study and that that you may skip.

1. What do you type to start the text editor program that runs on the HP 3000?
2. What constitutes a valid file name on the HP 3000?
3. Name six editing subcommands of the EDIT/3000 subsystem.
4. Which terminal key allows you to suspend a program?
5. Which HP 3000 command allows you to “reenter” a suspended program?
6. Which HP 3000 command terminates a suspended program?
7. What is an HP 3000 “K” file?

---

## Lesson 1 Using EDIT/3000

Lesson 1 presents

- the **EDIT/3000** editing program
- the command to start the EDIT/3000 program: EDITOR' \*the EDIT/3000 command to end the EDIT/3000 program: END
- the EDIT/3000 command to start writing: ADD
- the EDIT/3000 command to stop writing: //
- what the EDIT/3000 prompt tells you
- the EDIT/3000 command to save a text file: KEEP

### What is EDIT/3000?

The text editor EDIT/3000 is a Hewlett-Packard subsystem that comes bundled with your MPE/iX operating system. It is devoted to the general task of creating and editing text files. The editor has its own special set of commands and subcommands that give you considerable flexibility in your work.

At work you may hear the terms **application**, **subsystem**, **utility**, and **program**. They are all basically programs, but with subtle differences between them. Those differences are not especially significant to the general user.

### Why use EDIT/3000?

EDIT/3000 is available to everyone who uses your MPE/iX system. The fundamentals of EDIT/3000 are easy to learn. EDIT/3000 produces text that is fully compatible with all MPE/iX operations.

EDIT/3000 allows you to create any kind of file that you might need with MPE/iX.

- You will use EDIT/3000 to create text files in this and other Hewlett-Packard training courses.
- You will use EDIT/3000 to create command files in module 6 of this training course.
- You will use EDIT/3000 to create user-defined commands (UDCs) and job files in more advanced Hewlett-Packard training courses.

---

#### Note

*More advanced users:*

The editor creates a line-oriented ASCII text file with embedded carriage controls.

---

### Starting EDIT/3000: EDITOR

Starting the editor is simple. Just type the command name and press **Return**.

Enter:

```
EDITORReturn
```

---

#### Note

If this way of starting the editor does not work, it means that someone has modified the **path**. The path is an advanced topic covered in the *900 Series HP 3000: Advanced Skills* course.

In such a case, enter:

When the editor starts, your screen should look something like this:

```
:EDITOR
HP32201A.07.17 EDIT/3000 THU, SEP 15, 1988, 3:34 PM
(C) HEWLETT-PACKARD CO. 1985
/_
```

Notice that the MPE/iX system prompt has disappeared. Instead, the editor provides its own prompt, the slash (/).

When the editor prompt ( / ) and the cursor are alone on a blank line, the editor is in **command mode**. In command mode, you can issue EDIT/3000 commands, but you cannot create lines of text.

**Leaving the editor:**  
**END**

To get out of the editor program, use the END, EXIT, or E command.

Enter:

```
END(Return)
```

Get out of the editor program now.

Your screen should look like this:

```
/END
END OF SUBSYSTEM
: _
```

The computer once again displays the MPE/iX system prompt.

**Note**

Recall that the system prompt can be anything your system operations personnel decide to use.

---

**Writing in the editor:**  
**ADD**

Get into the editor again.

Enter:

```
EDITOR(Return)
```

Do you have the editor slash prompt on your screen? If you do, you are in command mode.

The command to start writing in the editor is ADD or A. Commands must be entered when the editor is in the command mode. You can enter commands in all uppercase letters, all lowercase letters, or a combination of uppercase and lowercase letters.

Enter:

```
ADD(Return)
```

It should look like this on your screen:

```
/ADD
1 _
```

ADD puts the editor in **edit mode**. Now you can enter text.

---

**Note** You cannot enter EDIT/3000 commands again until you return to the command mode. That comes a little later, in the section called “Stop Writing (/)”.  

---

The editor provides the number of the next line that you type. Since you have started a new file, the first line number is 1.

### Typing lines

You are now ready to begin typing. When typing in text, remember the following:

- Type the way that you would on a typewriter.
- EDIT/3000 will accept up to 72 characters on a line.
- To start a new line, press `(Return)`
- Every time that you press `(Return)`, the editor gives you a new line, with a new number.

---

**Note** The EDIT/3000 program limits the number of characters accepted per line to 72 characters. More experienced users can alter this limit.  

---

Type this:

```
EDIT/3000 is a hard-working program. Your operator would(Return)
```

Does your screen look like this?

```
 /ADD
 1 EDIT/3000 is a hard-working program. Your operator would
 2 -
```

Notice what happens:

- The editor labels the first line 1.
- Every time that you press `(Return)`, the editor gives you a new line, with the next number in sequence.

---

**Note** *Advanced Users*

EDIT/3000 is a line editor. Unlike many word processors found on a personal computer, the editor requires that you press `(Return)` to end one line and start another. It also requires that you indicate explicitly the line number(s) when you edit or manipulate a line or lines, rather than using the direction keys.  

---

Type in the rest of these lines, and notice where the `(Return)` key should be pressed:

```
call it an entire subsystem of your computer operating system.(Return)
It has dozens of its own commands, some of them quite(Return)
elaborate. It even uses three REDO subcommands that you(Return)
used in module 1.(Return)
```

Your file should look like this:

```

/ADD
1 EDIT/3000 is a hard-working program. Your operator would
2 call it an entire subsystem of your computer operating system.
3 It has dozens of its own commands, some of them quite
4 elaborate. It even uses three REDO subcommands that you
5 used in module 1.
6 -

```

Notice that the cursor is on an empty line. You are now ready to change back to command mode. If you were to change back to command mode while the cursor was at the end of line five, you would lose line five.

### Stop Writing (//)

To stop adding lines, enter:

```
//(Return)
```

It should look like this on your screen:

```

/ADD
1 EDIT/3000 is a hard-working program. Your operator would
2 call it an entire subsystem of your computer operating system.
3 It has dozens of its own commands, some of them quite
4 elaborate. It even uses three REDO subcommands that you
5 used in module 1.
6 //
...
/

```

The editor is now in command mode. You can now execute EDIT/3000 commands again.

### Note

---

Any line that ends in //(Return) is lost. EDIT/3000 discards that line and waits for your next command.

---

### Looking at your file: LIST

At this point, you may wish to see all of the file you have just written.

To see all of a file on screen, use the LIST or L command.

Enter:

```
LIST ALL(Return)
```

You should see all of the file scroll upward on your screen. If it were a long file, you might need to use (STOP) or (CTRL)(S) to stop the scrolling (and then (STOP) again or (CTRL)(Q) to let the scrolling continue).

You can display a single line from a file, too.

Try this:

```
LIST 5(Return)
```

You should see line 5 scroll up on your screen.

You can display a portion of a file:

Enter:

LIST 2/4(Return)

This tells the computer to show you lines 2 through 4. You should see lines 2, 3, and 4 scroll into view.

## Note

---

The editor recognizes the words “first” and “last”. LIST FIRST(Return) displays the first line of your file. LIST LAST(Return) displays the last line of your file.

---

## Saving a file: KEEP

So far your text exists only in the editor’s workspace. In order to keep text permanently, you must save the file on disk.

The command to save or keep a file permanently is KEEP. This saves the file on a computer disk.

The editor requires that you give the file a name *when* you keep it on the disk. Call it DOCUMENT.

Enter:

KEEP DOCUMENT(Return)

You know that your file has been kept when the / prompt reappears.

### Keep an existing file

Enter:

ADD(Return)

What happens? You have saved a copy of the file to disk. A copy of the file is still in the editor’s workspace, too.

The cursor is now at the beginning of a new line, line number 6.

Add this text at line number 6:

---

The ADD command adds lines at the end of an existing file.(Return)

Get out of edit mode (//(Return)).

By now the screen is a little cluttered. List your file again.

Enter:

LIST ALL

```
1 EDIT/3000 is a hard-working program. Your operator would
2 call it an entire subsystem of your computer operating system.
3 It has dozens of its own commands, some of them quite
4 elaborate. It even uses three REDO subcommands that you
5 used in module 1.
6 The ADD command adds lines at the end of an existing file.
```

Keep DOCUMENT once more.

Enter:

KEEP DOCUMENT(Return)

Do you get this message on the screen?

```
DOCUMENT ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW
PURGE OLD?_
```

The editor knows that a version of DOCUMENT already exists on the computer disk. It assumes that the file you are keeping is a new version of the same file, so it asks: "Do you want to purge (erase) the old version on the disk and replace it with the new version you are keeping?"

In this case, the computer is waiting for you to answer YES or NO to its question.

- If you answer YES to the PURGE OLD? inquiry, the editor destroys the older version of the file on the disk and replaces it with your new version.
- If you answer NO to the PURGE OLD? nothing changes. The old file remains, and the file that you just typed is not be saved.

This time, enter:

NO(Return)

This message should appear on the screen:

PURGE OF OLD FILE NOT CONFIRMED - TEXT NOT KEPT

Is your file still in the workspace?

Enter:

LIST ALL(Return)

You should see your file displayed on the screen.

Leave the editor by entering this:

END(Return)

You should see this on the screen:

IF IT IS OK TO CLEAR RESPOND "YES"  
CLEAR?

The editor knows that you added a line to the original text (line 6). So, it asks whether you want to save this new version of the file before ending the program.

- If you answer YES, the editor destroys the work file. The older version of the file, still on the disk, remains unchanged.
- If you answer NO, nothing changes. The old file remains, and the file that you just typed is still in your workspace. You stay in the editor.

This time, enter:

NO(Return)

The following message should appear on your screen:

CLEAR NOT CONFIRMED - TEXT IS UNCHANGED

To save the file, use:

KEEP DOCUMENT(Return)

Again, you are asked if you want to purge (erase) the old version of the file.

Enter:

YES(Return)

The editor now saves the most recent version of the file (and in doing so destroys the older version). If you want to keep two versions of the file, you have to save one version with one name and the other version with another name.

Finally, exit the editor. Enter:

`END(Return)`

---

**Note**

You will be using this file, `DOCUMENT`, again in later lessons.

---

**Getting a listing of files:  
LISTFILE**

Now that you have created and kept a file, how can you keep track of any or all of the files that you save?

The MPE/iX command `LISTFILE` alphabetically displays file names to your terminal screen.

At the MPE/iX prompt, enter:

`LISTFILE(Return)`

Your screen should look something like this:

`:LISTFILE`

`FILENAME`

`DOCUMENT`

`:`

There may be other files listed on the screen, but for now, check to see if `DOCUMENT` is there.

**Lesson 1 Exercises**

1. Name the text processing subsystem on the HP 3000.
2. What command do you type at the MPE/iX prompt to start the editor?
3. What do you type to end the editor?
4. Which prompt indicates that you are in the editor?
  - a. /
  - b. :
  - c. \
  - d. <
5. What `EDIT/3000` subcommand lets you start typing text?

6. What EDIT/3000 subcommand terminates the ADD subcommand?

7. Consider the following scenario:

You have just created a new file, which you try to save under the name MYFILE1. The following message appears on your terminal screen:

```
DOCUMENT ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW
PURGE OLD?
```

Would you answer yes or no to do the following:

- a. Instruct the editor program to *write over* your old file MYFILE1 and save the one that you have just created.
  
  - b. Instruct the editor to retain the old file and *not write over* the original MYFILE1.
8. Which MPE/iX command alphabetically lists files to the terminal screen?

## Lesson 2 File Names

Lesson 2 presents the concept of file names.

- understanding valid (acceptable) file names
- understanding invalid (unacceptable) file names

MPE/iX is very strict about the file names it accepts.

### File naming conventions

A valid (acceptable) file name:

**Table 4-1. File Names**

| Valid File Name                                                        | Notes:                                                                                                                  |
|------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Starts with a letter                                                   | A - Z <i>or</i> a - z (uppercase and lowercase are the same to MPE/iX)                                                  |
| Consists of only letters and numbers ( <b>alphanumeric</b> characters) | A - Z <i>or</i> a - z <i>or</i> 0 - 9                                                                                   |
| Contains no <b>special characters</b>                                  | + <i>or</i> - <i>or</i> * <i>or</i> ! <i>or</i> & <i>or</i> ? are not acceptable (/ is allowed, but it causes trouble.) |
| Has <i>no more than</i> eight (alphanumeric) characters in all         |                                                                                                                         |

All of these are valid (acceptable) file names:

NOTEREV1

NOTEREV2

MYFILE

LETTER10

What is wrong with saving an editor file as FIRSTFILE?

If you said there are *too many characters*, you are right. FIRSTFILE contains nine characters in all.

If you try to keep a file with an unacceptable name, you see something like this on the screen:

```

/KEEP FIRSTFILE

+-F-I-L-E---I-N-F-O-R-M-A-T-I-O-N---D-I-S-P-L-A-Y+
! ERROR NUMBER: 54 RESIDUE: 0 !
! BLOCK NUMBER: 0 NUMREC: 0 !
+-----+
*41*FAILURE TO OPEN KEEP FILE (54)
INVALID FILE REFERENCE (FSERR 54)
/

```

The solution is to choose another name -a valid name- for your file.

### A troublesome character

Although it is possible to put the slash character (/) in a file name, the slash has a special function in file names.

#### Caution

Do not put a slash (/) in a file name.

Putting a slash (/) in a file name tells the computer to put a lockword on that file. If you do that accidentally, you might have difficulty getting to the file later.

The concept of “locking” files is a separate topic not covered in this course but discussed in the *900 Series HP3000: Advanced Skills* course.

### Invalid file names

Table 4-2. File Naming Errors

| Unacceptable Name | What is Wrong                                      | A Better Name |
|-------------------|----------------------------------------------------|---------------|
| 1STFILE           | Starts with a number                               | FILE1ST       |
| FIRSTFILE         | Too many characters                                | FIRSTFIL      |
| FILE_1            | The <b>underscore</b> ( _ ) is a special character | FILE1         |
| FILE/             | <b>Slash</b> (/) puts a lockword on a file         | FILE1         |

Although you can choose any file naming convention that is valid and suits your purpose (or your group’s purpose), some general rules may help:

- A file name should suggest its content or purpose in some way that has value for you or your group.
- Files that have related contents or purpose should have names that are clearly related in some fashion.

### Lesson 2 Exercise

1. Which of the following names are *unacceptable* file names on the MPE/iX system and why?
  - a. MYFILE
  - b. yourfile2
  - c. 1BIGMEMO
  - d. a

e. FRED\*\*8

f. The\_Quick\_Brown\_Fox

g. filenum4

---

## Lesson 3

### Editing a Text File

Lesson 3 presents the commands that permit you to do the following:

- open an editor file for editing: TEXT
- print an editor file: LIST
- edit a line: MODIFY
- add new lines to the end of a text file: ADD
- add new lines in the middle of a text file: ADD
- erase a line of text file: DELETE
- renumber lines: GATHER ALL
- move lines: GATHER
- clear the file from the editor's workspace: DELETE ALL

#### Open an EDIT/3000 file for editing

Using the editor, you have learned to create and save files.

#### Note

---

Before trying this, find out whether your computer is connected to a printer and where that printer is located.

---

To print a file from the editor, the file first must be brought into the work area.

Get into the editor and at the / prompt enter:

```
TEXT DOCUMENT(Return)
```

The TEXT command instructs the computer to find the specified file, and, if it exists, to put it into the editor's workspace for printing or for file editing.

To display the entire DOCUMENT file to your screen, enter:

```
LIST ALL(Return)
```

This file should look very familiar to you by now.

#### Sending a file to the printer

With a slight change, the LIST or L command prints the file on your **line printer**.

To send the file that you are working on to the printer, enter:

```
LIST ALL, OFFLINE(Return)
```

- The LIST command tells the editor to display the file on the screen.
- The ALL parameter tells the computer to display all of the file.
- The addition of , OFFLINE tells the computer to send the file (or lines of the file) not to the screen but to the printer.

You should eventually see this message on the screen:

```
/LIST ALL, OFFLINE
*** OFF LINE LISTING BEGUN. ***
/
```

This tells you the computer has started working on your command.

If your printer is busy, your file might take a while to print.

It may take a few more moments before the editor prompt (/) reappears on the screen. Its reappearance tells that you the file is *on its way to the line printer*.

## Note

---

*On its way to the line printer:*

The printer can handle only one file at a time. If other people have sent files to the printer—especially if they are long files—your file might have to wait in line to be printed. Your file is given a number and is printed in proper order, along with all of the other waiting files. The order of printouts can be altered by the system manager.

---

You can also print a single line:

```
LIST 1, OFFLINE
```

This would print line 1.

Or you can print a range of lines:

```
LIST 2/5, OFFLINE
```

This would print lines 2, 3, 4, and 5.

## Identifying your document

The paper copy of your file is called a printout, hard copy, or **listing**. Your listing can be easily distinguished from other users' listings by the **header** (front) and **trailer** (back) pages of your listing. Your logon identifier appears several times across these pages.

Go pick up your listing.

## Problems?

Did your file not print out? Your system manager or someone else may have created another standardized way of indicating a line printer or some other printing device. If so, using OFFLINE might not succeed. Ask your account manager for the correct way to indicate the line printer for this use of the editor LIST command.

## Adding lines of text: ADD

The ADD command automatically begins adding lines at the *end* of the existing file, no matter where the end of the file is found.

From within the editor, display the entire DOCUMENT file:

```
LIST ALL(Return)
```

Next enter:

```
ADD(Return)
```

This brings a new, blank line labeled 7 onto the screen. You are now ready for typing. Add the following lines 7 through 10 to your file:

```
Remember that corrections have to be made one step at a time.(Return)
After each step you can see what you accomplished by pressing(Return)
Return. When you are done, leave the MODIFY step by pressing(Return)
Return twice in a row.(Return)
```

//(Return)

Your screen is probably cluttered, but your display should show this:

```
/LIST ALL
1 EDIT/3000 is a hard-working program. Your operator would
2 call it an entire subsystem of your computer operating system.
3 It has dozens of its own commands, some of them quite
4 elaborate. It even uses three REDO subcommands that you
5 used in module 1.
6 The ADD command adds lines at the end of an existing file.
/ADD
7 Remember that corrections have to be made one step at a time.
8 After each step you can see what you accomplished by pressing
19 Return. When you are done, leave the MODIFY step by pressing
10 Return twice in a row.
11 //
/
```

### Adding lines in the middle of a text file: ADD

What if you want to add text in the middle of a file? Using ADD with a line number starts a new line *immediately after* the number you have specified.

Enter:

ADD 6(Return)

and at 6.1 type the following:

```
I decided to add more lines in the middle of what I already(Return)
had.(Return)
(Return)
//Return
```

Enter:

LIST ALL(Return)

Your screen should look like this:

```
1 EDIT/3000 is a hard-working program. Your operator would
2 call it an entire subsystem of your computer operating system.
3 It has dozens of its own commands, some of them quite
4 elaborate. It even uses three REDO subcommands that you
5 used in module 1.
6 The ADD command adds lines at the end of an existing file.
6.1 I decided to add more lines in the middle of what I already
6.2 had.
6.3
7 Remember that corrections have to be made one step at a time.
8 After each step you can see what you accomplished by pressing
9 Return. When you are done, leave the MODIFY step by pressing
10 Return twice in a row.
```

Notice that to insert new lines between line 6 and line 7, the editor added the new lines with increments of 0.1 (one-tenth). It can also add lines in increments of 0.01 or 0.001.

The editor would allow you to do this (*but do not do this now*):

/ADD 6.001

The editor would let you start typing on line 6.001 and then on 6.002 and then on 6.003, and so on. When you got to line 6.009, the editor would increment the line count to 6.01.

Using increments of 0.001, the editor could add as many as 999 new lines between line 6 and line 7. If you need still more lines, you can use the GATHER ALL command to renumber the lines. Then add lines where you want them.

## Erase a line: DELETE

Make sure you are in command mode.

The DELETE or D command erases a line or a range of lines.

Try entering

```
DELETE 6.2(Return)
```

It should appear this way on the screen:

```
/DELETE 6.2
6.2 had.
```

DELETE displays the line that you erased.

Use LIST ALL(Return) to verify that the line has been erased.

To delete a series of consecutive lines, use DELETE and specify the line range you wish to erase. In the following example, lines 7 through 10 of your document are erased.

Try this and see what happens:

```
DELETE 7/10(Return)
```

## Renumbering lines: GATHER ALL

The GATHER ALL command instructs the editor to renumber lines. That's useful when you have added lines *between* lines and the numbering scheme begins to get out of hand.

Try this:

```
GATHER ALL(Return)
LIST ALL(Return)
```

You should see the following:

```
1 EDIT/3000 is a hard-working program. Your operator would
2 call it an entire subsystem of your computer operating system.
3 It has dozens of its own commands, some of them quite
4 elaborate. It even uses three REDO subcommands that you
5 used in module 1.
6 The ADD command adds lines at the end of an existing file.
7 I decided to add more lines in the middle of what I already
8
```

Now the line numbers are neat and tidy. Notice that you can now add lines between line 6 and line 7 and have the increment start with 0.1 all over again.

## Moving lines: GATHER

The GATHER command lets you move one or more lines from one place to another.

Enter this:

```
GATHER 7 TO 1.1(Return)
```

You will see this:

```
/GATHER 7 TO 1.1
7 => 1.1
```

The editor displays the number of the line being moved and shows the line number where it will appear (line 1.1).

Use LIST ALL(Return) to verify that line 7 has been moved to line 1.1 and that line 7 has disappeared.

You should see this:

```
1 EDIT/3000 is a hard-working program. Your operator would
1.1 I decided to add more lines in the middle of what I already
2 call it an entire subsystem of your computer operating system.
3 It has dozens of its own commands, some of them quite
4 elaborate. It even uses three REDO subcommands that you
5 used in module 1.
6 The ADD command adds lines at the end of an existing file.
8
```

Do you notice?

To move line 7 between lines 1 and 2, you used the GATHER command to move line 7 to line 1.1.

How would you move lines 3 through 5 directly under line 1.1?

```
GATHER 3/5 TO 1.2(Return)
```

The 3/5 tells the editor to move lines 3 through 5.

Try it. What happens?

Enter:

```
LIST ALL(Return)
```

What do you see?

Moving the lines has jumbled the meaning of the text. And you now have line numbers with decimals. It would be useful to renumber the lines.

## Renumbering lines

Do this:

```
GATHER ALL(Return)
```

Then use

```
LIST ALL(Return)
```

What happens?

The GATHER ALL command instructs the editor to renumber lines lines of the work file. That becomes useful when you have added or moved lines *between* lines and the numbering scheme threatens to get out of hand.

Save the file.

Enter:

KEEP(Return)

The editor already knows the name of the file, so you do not have to specify a file name this time. (If you specified a different file name—KEEP DOCFILE—the editor would know that you wanted to keep your work under a new file name.)

So you should see this on the screen:

```
DOCUMENT
DOCUMENT ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW
PURGE OLD?
```

Enter:

YES(Return)

### **Editing a text line: MODIFY**

Use the MODIFY command to edit lines of text.

For example: MODIFY 22 tells the computer to let you edit line 22, *if* there is a line 22.

#### **MODIFY Works Like REDO**

Do you recall the commands used with the REDO command in module 1 to edit command lines?

The MODIFY command in the editor uses three of the same commands: D, I, and R.

- |   |                                                          |
|---|----------------------------------------------------------|
| D | Deletes the character above it.                          |
|   | DDDD erases four characters, the ones above each D.      |
| I | Inserts character(s) above the I.                        |
| R | Replaces character(s) above the R with new character(s). |

Like REDO, MODIFY retrieves a line for correction.

- |         |                                                                                               |
|---------|-----------------------------------------------------------------------------------------------|
| STEP 1. | Make <i>one</i> correction on the line.                                                       |
| STEP 2. | Press (Return).                                                                               |
|         | Pressing (Return) once displays the line again for another correction.                        |
| STEP 3. | If the line is now correct, press (Return) again.                                             |
|         | Pressing (Return) twice in a row signals the end of your corrections on that particular line. |
|         | If the line is still not correct, go to STEP 4.                                               |
| STEP 4. | Make another correction on the line. Go to STEP 3.                                            |

Enter:

ADD(Return)

Then type the following line, *exactly* as shown, to the end of the DOCUMENT file:

```
SAVE is used to save all or prt of the work file.(Return)
// (Return)
```

Now, enter the following:

```
MODIFY 9 (Return)
```

Using the D, I, and R commands of MODIFY, correct the line above to read as follows:

KEEP is used to save all or part of the work file.

Follow the editing step by step.

```
/MODIFY 9
9 SAVE is used to save all or prt of the work file.
 RKEEP (Return)
9 KEEP is used to save all or prt of the work file.
 d (Return)
9 KEEP is used to save all or prt of the work file.
 ia (Return)
9 KEEP is used to save all or part of the work file.
 (Return)
```

Notice:

- The editor prompts you for the changes that you want to make.
- When the line is correct, press (Return) again to tell the editor to accept the line the way it is.

Unlike the REDO command, the MODIFY command lets you modify blocks of text.

- You could enter MODIFY 1/4 to edit lines 1 *through* 4.
- The computer would first display line 1 for editing.
- When you finish editing line 1 and press (Return) twice, the editor displays line 2 for editing, and so on.

Try that, if you like, with the file that you have been working on.

## Starting a new editor file: DELETE ALL

### Note

---

Read this lesson completely *before* performing any of the following. Then, if you like, go back to do the practice exercises.

---

The editor's workspace is an area of computer memory that is separate from and independent of disk memory. So, it is possible to have one version of a file on the disk and edit that same file in the editor's workspace. Naturally, the more you edit the version in the workspace, the more it comes to differ from the version that is saved on the disk.

If you wish, you can use KEEP DOCUMENT to save to disk the workspace version of the DOCUMENT file that you have been editing throughout this module. As soon as you do that, the two versions are identical—until you begin more editing on the workspace version; however, if you want to clear the workspace *without* saving your editing changes, enter:

```
DELETE ALL(Return)
```

This *erases everything* in the editor's workspace memory area, but *the file that is kept on disk is unchanged*.

Because this can have disastrous consequences if typed in error, the EDIT/3000 program gives the following warning:

```
IF IT IS OK TO CLEAR RESPOND "YES"
CLEAR?
```

A NO(Return) reply to this warning halts any deletion.

Use LIST ALL(Return) to verify that there is no file in the editor's workspace.

With the editor's workspace completely clear, you could start a new file if you wanted to. To start a new file, you would enter the ADD command:

```
ADD(Return)
```

If there is no file in the workspace (you entered DELETE ALL(Return)), then the editor starts a fresh new file with line 1.

### Lesson 3 Exercises

1. Create a 20-line file and save that file as FILEA.
2. List all of FILEA to your *terminal screen*.
3. List lines 5 through 15 of FILEA to your *terminal screen*.
4. Print a copy of FILEA to a *line printer*.
5. Print lines 15 through 20 to a *line printer*.
6. Delete the first three lines of FILEA.
7. How would you “undo” step 6 *without* reentering the three deleted lines?
8. Move line 3 of FILEA to line 6.1.
9. Move the last five lines of FILEA to the beginning of the file (before line 1).

10. Renumber the lines of your file.

11. Study the following text line:

Mary will be coming ovr for dinner.

If you were to *modify* this line, which subcommands would you use to make the following corrections?

a. Change "Mary" to "Geno".

Mary will be coming ovr for dinner.

b. Correctly insert the "e" in "ovr" so that it reads "over."

Geno will be coming ovr for dinner.

c. Delete the word "coming."

Geno will be coming over for dinner.

12. Suppose that you are in the editor and have just retrieved a file called MYFILE from the disk. You make some editing changes to MYFILE. After completing the edit, you are ready to start working on a new file.

If you performed a DELETE ALL right now (without performing a KEEP), what *version* of MYFILE will be retrieved from disk the next time that you needed to do some more editing?

What, if anything, would be lost?

---

## Lesson 4 More About the Editor

Lesson 4 presents these useful text-related tasks:

- the command to join files together: JOIN
- executing MPE/iX commands from within the editor
- using the `(BREAK)` key to suspend a program or command
- EDIT/3000 safety files (**K files**) and how to recover them

### Joining files together: JOIN

When you are editing one file, you can add the contents of another file to the file you are working on by using the JOIN command. This process is often called **appending**.

The JOIN command expects you to specify the following:

- which file you want added to the one that you are working on
- where you want the file added (at the beginning, at the end, or somewhere in the middle)

Use the editor to create a three-line file by typing the following:

```
This is line 1 of the file DRAFT.(Return)
This is the second line of DRAFT.(Return)
This is the last line of DRAFT.(Return)
```

Exit out of the add line mode by typing the following:

```
//(Return)
```

Now, save this file as DRAFT by entering:

```
KEEP DRAFT(Return)
```

Use

```
DELETE ALL(Return)
```

Now start a file that looks like the following:

```
This is the file DRAFT2.(Return)
This is a very short file.(Return)
This is line 3.(Return)
This is line 4.(Return)
This is the end of the file DRAFT2.(Return)
```

Again, stop adding lines with `//(Return)`. Save this file as DRAFT2 by entering:

```
KEEP DRAFT2(Return)
```

### Listing files from within the editor

Make sure that you are in the editor's command mode.

To do a quick check to see if both files are saved to disk, you can execute the system command LISTFILE from *within* the editor.

At the editor prompt, enter:

```
:LISTFILE(Return)
```

Did you enter a colon (:) ahead of the LISTFILE command? You must do it that way.

It will look something like this on the screen:

```
/:LISTFILE

FILENAME

DOCUMENT DRAFT DRAFT2 K1901102

/
```

---

**Note**

LISTFILE is not an editor command, but an MPE/iX command. Putting a colon (:) ahead of the command tells the editor to execute an MPE/iX command.

Not all MPE/iX commands can be executed from within a program, and not all programs allow you to execute MPE/iX commands. For information about commands, refer to the online help facility. For information about a program or application, refer to the documentation for the program or application in question.

---

Clear the workspace by entering:

```
DELETE ALL(Return)
```

---

**Note**

Remember that clearing the editor workspace erases the area of memory that the computer is using for your current editing. It does not clear the screen. Refer to table 2-1 in module 2.

---

### Joining at the end

Now that you know the DRAFT and DRAFT2 files exist, join DRAFT to the end of the file DRAFT2.

Text in DRAFT2 by entering:

```
TEXT DRAFT2(Return)
```

Then enter:

```
JOIN DRAFT'(Return)
```

It will look like this on the screen:

```
/TEXT DRAFT2
/JOIN DRAFT
```

The contents of DRAFT that are *appended* to the end of DRAFT2, on lines 6, 7, and 8, appear on the screen. To see all of the file, enter:

```
LIST ALL(Return).
```

You should see this:

```
/LIST ALL
1 This is the file DRAFT2.
2 This is a very short file.
3 This is line 3.
4 This is line 4.
5 This is the end of the file DRAFT2.
6 This is line 1 of the file DRAFT.
```

```
7 This is the second line of DRAFT.
8 This is the last line of DRAFT.
```

In this exercise, you did not specify where to put DRAFT, so the editor made the assumption that you wanted to put it at the end (appending).

Keep the joined file of DRAFT and DRAFT2 as a file named NEWDRAFT.

```
KEEP NEWDRAFT(Return)
```

Clear the editor's workspace by entering

```
DELETE ALL(Return)
```

### Joining in the middle

By specifying a line number with JOIN, you can add one file from the disk anywhere in the body of the file you are working on.

Try it. Text in DRAFT.

Enter:

```
TEXT DRAFT(Return)
JOIN DRAFT2 TO 1.1(Return)
```

To see the combined file, enter:

```
LIST ALL(Return)
```

The contents of DRAFT2 were inserted, creating new lines between line 1 and line 2. Notice that the added lines have increments of 0.1.

Keep this file as NXTDRAFT.

```
KEEP NXTDRAFT(Return)
```

## Suspending EDIT/3000: the Break key

Some programs won't allow you to execute a system command while the program is running. There may be times when you need to get to the system prompt (: ) to access MPE/iX commands.

Fortunately, most Hewlett-Packard programs do allow you to suspend program operations in order to execute a command, and most allow you to resume operation of the program without losing your place.

### Using the **(Break)** key

Try this exercise from within the editor. Once again, clear the editor's workspace.

Enter:

```
DELETE ALL(Return)
```

Text in the DRAFT.

```
TEXT DRAFT(Return)
```

Enter:

```
ADD(Return)
```

Add the following lines:

```
Once upon a midnight dreary(Return)
```

While I pondered weak and weary(Return)

Note the line number that you are on (6) and remember that you are still in the Add mode.

Locate the (Break) key on your keyboard and press it.

Your screen should look like this:

```
/ADD
4 Once upon a midnight dreary
5 While I pondered weak and weary
6 (Break)
:
```

You won't see (Break) on the screen.

Notice that the system prompt has returned.

Pressing (Break) forces the *suspension* of the editor. You can now communicate directly with the MPE/iX system. You are in **Break mode**.

---

### Note

Not all programs or commands can be suspended with (Break). Some commands ignore (Break) and proceed to their conclusion. The online help facility indicates which commands are *breakable*.

---

Use SHOWTIME to get the date and time from the system.

It should look like this on the screen:

```
:SHOWTIME
FRI, JUL 27, 1990, 4:58 PM
:
```

### Continuing a Program: RESUME

The RESUME command continues the operation of a program or command that has been suspended. The program or command resumes its action exactly where it left off when you pressed (Break).

---

### Note

Not all programs or commands can be continued with RESUME. Some commands terminate when you press (Break).

---

Now enter:

```
RESUME(Return)
```

Does this appear on the screen?

```
:RESUME
READ pending
```

This is what is happening:

- The RESUME command tells the computer to pick up where it left off with the editor (or whatever it was doing when you pressed (Break)).
- the READ pending message means that the computer is waiting for you to do something, but it may not be clear *what* it is waiting for.

When you pressed (Break) you were in Add mode. And when you tell MPE/iX to resume, it returns to the editor's Add mode.

The computer is waiting for you to do one of two things:

- continue typing a line
- signal the start of a new line by pressing `(Return)`

To continue typing a line, enter this:

```
Another line came back, ready for more editing.(Return)
```

This line becomes line 6, because you were on line 6 when you broke off line editing with `(Break)`. You cannot see the line number yet, but do you see that the editor is now ready for you to type line 7?

Now, type the following on line 7, but *do not* press `(Return)` at the end of the line.

```
I am going to type another sentence now.
```

At the end of line 7, press `(Break)`.

Here is the sequence as it occurs on the screen (`(Break)` and the `(Return)` keys are invisible, of course):

```
/ADD
 4 Once upon a midnight dreary
 5 While I pondered weak and weary
 6 (BREAK)
:SHOWTIME(Return)
FRI, JUL 27, 1990, 4:58 PM
:RESUME(Return)
READ pending
Another line came back, ready for more editing(Return)
 7 I am going to type another sentence now(Break)
:
```

Pressing `(Break)` obliterates line 7. In the editor, pressing `(Break)` when the cursor is on a line of text prevents the computer from ever receiving that line.

Get out of Break by typing RESUME. Press `(Return)` and get out of the Add mode of the editor.

Type the following to look at your file:

```
LIST ALL
```

### Use **ABORT** after **BREAK** to stop

Every program has its own way of allowing you to stop the program. For the editor, the command is END, EXIT, or E.

The ABORT command allows you to **terminate** the program without using the program-specific stop command. ABORT provides an emergency exit.

### Note

---

ABORT can be used only after you press `(Break)`.

---

Press `(Break)`

Enter:

```
ABORT(Return)
```

Here is what happens on the screen:

```
:ABORT
```

```
PROGRAM ABORTED PER USER REQUEST. (CIERR 989)
:
```

If you now try

```
RESUME(Return)
```

RESUME will not work.

You will see this on the screen:

```
:RESUME
COMMAND ONLY ALLOWED IN BREAK. (CIERR 1686)
:_
```

By executing ABORT, you have stopped the editor in its tracks. The only way to get back to the editor now is to start it all over again.

---

## Caution

Do not use ABORT to end programs. An ill timed ABORT can destroy information that you might want to keep.

---

## Safety files: K files

The editor does its best to ensure against emergencies that would compromise your current work. It even regards a Break or an ABORT command as a small emergency. To protect the work that you are doing, it creates K files while you are working on a text file. Here is how it works.

Enter:

```
LISTFILE(Return)
```

Look closely at the file names. Did you get something like this on the screen?

```
:LISTFILE

FILENAME

DOCUMENT DRAFT DRAFT2 K1901052 K1901102
NEWDRAFT NXTDRAFT
```

You should have one or more files that begin with the letter K, followed by seven digits. The K file appeared because you used the Break key while you were working on DRAFT.

The K files that appear on your screen have different numbers from the ones shown above.

- The K file is the editor's safeguard against emergencies.
- If an emergency comes up (a loss of power, an ABORT, or an interruption such as Break) the file that you are working on is saved with the K name.
- The K name consists of the letter K and a series of numbers that record the date and time.

The K file saved on your disk is the most current version of the file DRAFT. Notice, though, that another, older version of DRAFT is still on the disk.

### What the numbers mean

All seven digits following the K have a meaning. They tell you when the file was created. If an emergency arises (if the power goes out, for instance), look at the clock. Note the time.

**Table 4-3. The K Numbers**

| What they mean |                                                                                                                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| K1901052       | The <i>last four</i> digits (1052) tell you that the time (24-hour style) when you created the file. This one was created at 10:52 A.M.                                                           |
| K1901052       | The <i>first three</i> digits (190) tell that you the file was created on the 190th day of the year of the Julian calendar (for example, January 1 would be 001, January 2 would 002, and so on). |

Using these numbers, you can identify the file.

On what day and at what time was the file K0592305 created?

If you answered February 28, at 11:05 P.M., you were right!

### What is saved in a K file

The editor creates a new K file every time that you start a new file, text in an existing file for editing, or join one file to another file.

If you are creating a new file, there is nothing in the K file at first. If you are texting in an existing file, then the K file consists of whatever was in the existing file (on the disk) when you issued the TEXT command to bring the file into the editor workspace.

The editor adds new lines to the K file as you add lines to the work file that you see displayed on the screen. That means that if the power goes out, or you use **(Break)ABORT(Return)**, your file will be saved in a K file on the disk.

### Recovering a K file

Try this procedure to recover K files:

- Start the editor if you are not already in it.
- Enter:

```
:LISTFILE(Return)
```

Look for a file beginning with K and ending with 7 numbers.

- Use TEXT to put the K file into the editor's workspace (for example, TEXT K1901052(Return)). The following message appears on the screen:

```
WARNING TEXT FILE IS NOW WORK FILE.
```

- Use LIST ALL(Return) to see the file's contents.
- Use KEEP with a file name of your choice to save the file, KEEP KFILE(Return), or whatever valid file name you like.
- Leave the editor.

## Note

---

If your management treats K files as temporary files, your operator probably erases them from the disk regularly. If so, examine any K files that appear in your group. Decide whether to recover them (with a new name) or whether to let your operator purge them.

---

## Lesson 4 Exercises

1. Suppose that you have two disk files, one named FILE1, the other FILE2. Assume that FILE1 is in the editor's workspace.
  - a. What would you type at the editor prompt to join FILE2 to the end of FILE1?
  
  
  
  
  
  
  
  
  
  
  - b. What would you type at the editor prompt to join FILE2 to line 1.2 of FILE1?
  
2. How would you get a listing of the files kept to disk from within the editor? What would you type at the editor prompt?
  
  
  
  
  
  
  
  
  
  
3. If you are in the middle of a program and need to get to the MPE XL prompt, without exiting the program, what key would you press?
  
  
  
  
  
  
  
  
  
  
4. What MPE/iX command gets you back into the editor after you suspend the editor, as you did in answering the preceding question?
  
  
  
  
  
  
  
  
  
  
5. After suspending a program using the **Break** key, which MPE/iX command terminates the program that has been "broken" out of?
  
  
  
  
  
  
  
  
  
  
6. On what day and at what time were the following K-files created?
  - a. K3651230
  
  
  
  
  
  
  
  
  
  
  - b. K0020815
  
  
  
  
  
  
  
  
  
  
  - c. K0321152

d. K0152300

**Note**

---

Have you taken a break to relax? This might be a good time.

---

## Module 4: The Account Structure

---

Module 4 introduces the MPE/iX **account structure**.

|          |                                    |
|----------|------------------------------------|
| Lesson 1 | The MPE/iX Account Structure       |
| Lesson 2 | Security and the Account Structure |
| Lesson 3 | File and Group Security            |
| Lesson 4 | Changing Groups                    |
| Lesson 5 | Changing User Passwords            |

### Note

To make the fullest use of this module, you need at least two groups *other than* PUB. You learn about groups and accounts in this module. This module uses two groups called MYGROUP and OTHERGRP.

The ideal solution is to have your account manager create MYGROUP and OTHERGRP in your assigned account and give you access to them. Ask to have these capabilities: ND, SF, BA, and IA. Have your account manager designate MYGROUP as your **home group**. You learn about home groups in the section entitled “Your Home Group” in this module.

If you cannot have MYGROUP and OTHERGRP created for you, ask your account manager to give you access to two other, non-PUB groups in your account. Ask to have one group (*not* the PUB group) designated as your home group. Make a note of which group is your home group. Whichever group is your home group corresponds to MYGROUP. The other group then corresponds to OTHERGRP.

---

---

### Challenge Test

1. Name the four elements of the HP 3000's account structure.
2. Name four elements of a logon identifier.
3. What would you type at the system prompt to log on as the user NICK, in the CHARLES account, and the NORA group, also identifying your session as ASTA?
4. Generally, what is the name of the *public domain* group in an account?

5. TRUE or FALSE. All users on the HP 3000 have unlimited access to all of the system commands.
  
6. From which group does the system's security allow you to read, change, or *save* any file, regardless of where you are logged on?
  
7. Which MPE/iX command allows you to easily move from group to group in an account?
  
8. Which MPE/iX command allows you to change your user password?

---

## Lesson 1

### The MPE/iX Account Structure

Lesson 1 presents the MPE/iX account structure.

- understanding the account structure
- the four elements of the account structure
- the four elements of a logon
- the PUB group
- the SYS account

#### What is an account structure?

The MPE/iX account structure performs several functions. Anyone can log on and successfully use the HP 3000 without knowing about these functions, but you will be more effective in your work if you understand the account structure.

The account structure on the 900 Series HP 3000 is set up by your system management team and serves three main system functions: organization, security, and billing.

#### Organization

In order for the HP 3000 system to process all of the information from its user base, some structure is necessary to organize the information. This keeps your files from being mixed together with those of other users, for instance.

#### Security

The account structure can protect each user's files from unwarranted use by unqualified or unauthorized users. The system challenges the person logging on to enter one or more passwords, to be sure that the person really has authority to work on the system.

#### Billing

Your organization probably checks every so often to see how much time people have spent using the computer. Very likely the time used is the basis for determining who needs more computing facility, who needs less, and how to bill each department or unit for the internal cost of owning and running the computer. This billing process begins with your logon.

#### Elements of the account structure

The account structure deals with four elements: accounts, users (who belong to accounts), groups, and files.

You might think of the computer system as a filing room in which there are many filing cabinets.

#### Accounts

The basic unit in the MPE/iX account structure is the **account**. There are usually many accounts on the MPE/iX. As a user, you are assigned to a specific account. When you log on, you enter HELLO and your user name and the name of the account to which your user is assigned.

An account may have more than one user assigned to it.

If you think of the computer as a filing room, then each separate cabinet in the filing room is comparable to a separate account within the computer system.

### **Users**

**Users** are people like yourself who use the computer.

In a sense, users do not have a place in the account structure (cabinet). Instead, they are the people (more accurately, the *user names*) who have the key to an account (cabinet). A user name *is* the key to a particular account (cabinet).

### **Groups**

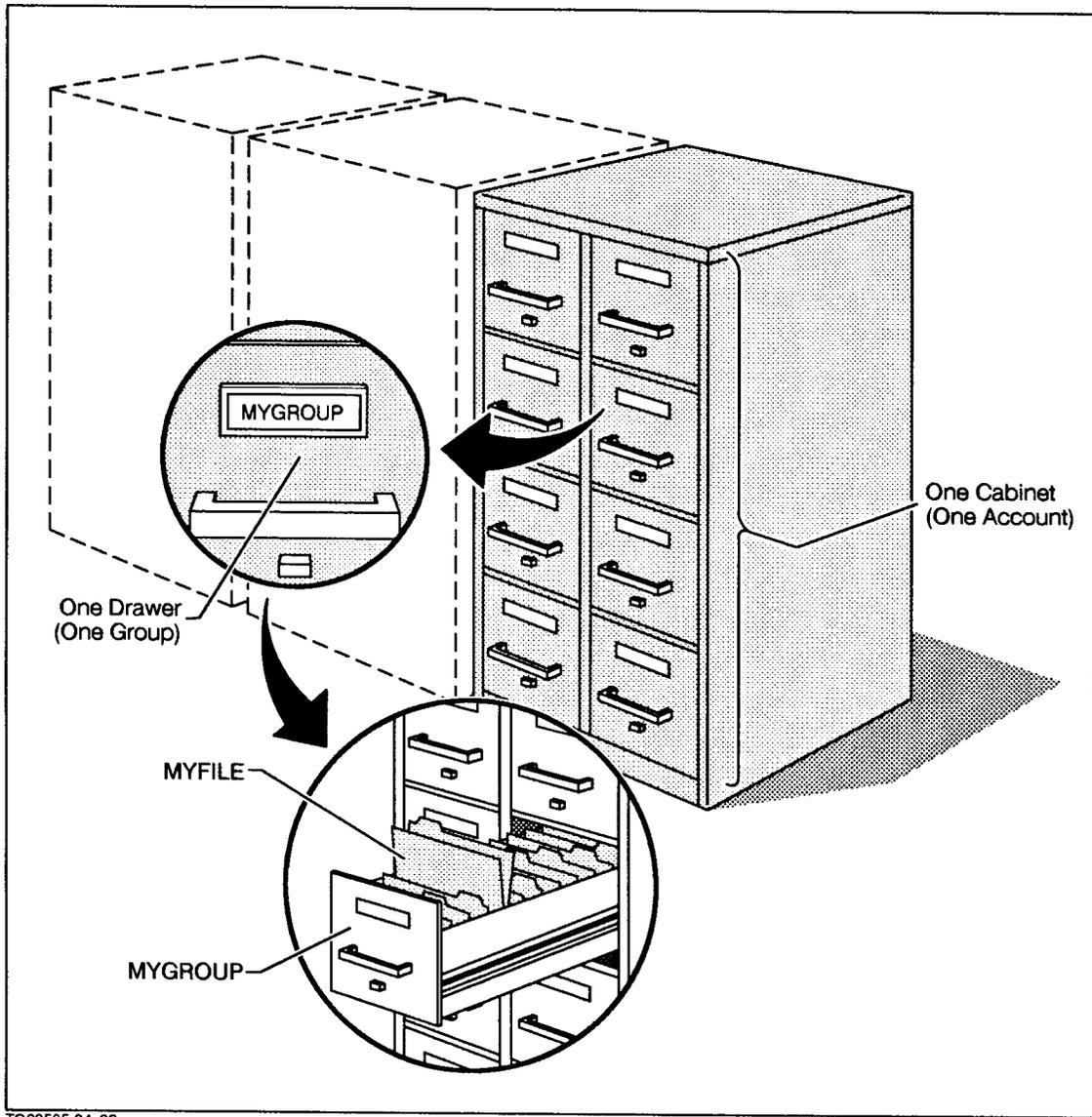
Accounts are divided into smaller units called **groups**. Groups are comparable to the drawers in each filing cabinet. Groups are separated from each other, just the way accounts are separated.

Groups contain files.

### **Files**

Files—all sorts of files—are the materials that users work with. They are kept in the groups. You have created, saved, and edited files, so you should feel somewhat familiar with them.

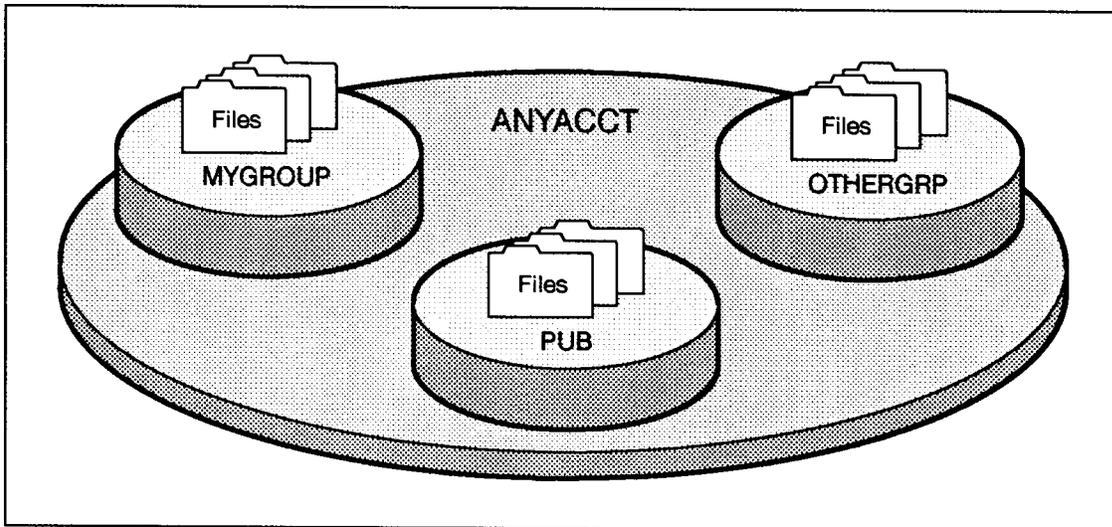
The following figure represents one way of looking at accounts, files, and groups.



TG20535-04\_02

**Figure 5-1. Elements of Account Structure**

The following figure represents another way of looking at accounts, files, and groups.



TG20535-04-01

Figure 5-2. Files in Groups in an Account

### Elements of your logon

There are four logon elements:

- user name (required at logon)
- account name (required at logon).
- group name (optional)
- session name (optional)

Like file names, these names must start with a letter and contain no more than eight characters in all.

### Note

In the following examples, the comma (,) and the period (.)—called **delimiters**—are important.

You do not have to end one logon session (with BYE), before starting another. If you are working in a session and you enter the HELLO command (plus your logon identity), your previous session automatically ends and a new one begins.

### User and account names

The simplest logon identity consists of a *user name* and an *account name*. You must specify at least a user name and an account name to log on.

```
HELLO username.acctname
```

```
HELLO JOHN.SMITHERS
```

If there are user and/or account passwords, you are prompted for them by the system security.

Look at your logon identifier. Which is the account name? Which is the user name?

## Group names

Before you can work on a file, you must log on to the account *and* *group* in which the file is found.

To log on to a particular group, add a comma and the group name at the end of your logon identifier:

```
HELLO username.acctname,group
```

Log on to MYGROUP, using your user and account name.

Example:

```
HELLO username.acctname,MYGROUP(Return)
```

---

### Note

If there is a group password assigned, you are prompted for it.

---

Execute SHOWME to check your user identity:

```
SHOWME(Return)
```

Now, log on to the OTHERGRP group. Remember that you do not have to log off in order to start a new session. Use the HELLO command with your appropriate logon identity.

Example:

```
HELLO username.acctname,OTHERGRP(Return)
```

Check your user identity again with the SHOWME command.

## Session names

Session names are used as an additional identifier, not for the computer, but for the benefit of those who are working with the computer. Session names can represent anything that helps identify your logon session, (persons, places, or things). To add a session name to the logon, put the session name and a comma ahead of everything else in the logon identity:

```
HELLO session,username.acctname,group
```

For example, you might choose to log on using a location, such as a building number, as a session name and the group name MYGROUP. (Sessions are limited to eight letters.)

Example:

```
HELLO BUILD20,JOHN.SMITHERS,MYGROUP
```

Execute a SHOWJOB at the prompt. In the above example, BUILD20 is the session identifier.

---

### Note

Actually, several people can log on simultaneously using the same user, account and group names. For example, several users working on a team project may share a logon identity. Session names can be used to identify specific users who may all be logged on at the same time.

---

## Logon requirements

These are the minimum requirements for logging on:

- A user name and an account name.
- The user name and account name must be separated by a period.
- The session name and group name are optional.
- The user, account, and group names may each be protected by a password.
- If you specify a session name, you must put a comma (,) *after* the session name, to separate it from the user name.
- If you specify a group name, you must put a comma (,) *before* the group name, to separate it from the account name.

## The PUB Group

Every account on the MPE/iX system has a special group called PUB that contains files that are accessible to any user assigned to that account.

### Note

---

Any file in the PUB group is accessible to anyone who is able to log on to the account, unless the file has been locked with a lockword or has special security assigned to it. A discussion on locking files can be found in the *900 Series HP 3000: Advanced Skills* course.

---

## The SYS account

Every MPE/iX system has a main system account called SYS that exists primarily for the use of the system manager. Like other accounts, the SYS account has a special group called PUB. Files stored in the PUB group of the SYS account are available without restriction to any system user, unless those files have lockwords or special file security.

One program in the PUB group of the SYS account that you have already used is the editor (EDIT/3000).

## Your Home Group

Recall that you do not have to specify a group name when you log on. A user name and an account name are all that you need.

It is common on most MPE/iX systems that if you do not specify a group name when you log on, MPE/iX *puts* you in some group. It puts you in the group designated as your home group. Your home group is your default logon group. If you do not have a home group (default group), you must specify a group name at logon.

Your account manager assigns your home group when your user name is created in an account. Recall at the beginning of this module that you could ask your account manager to create MYGROUP and OTHERGRP for you and designate MYGROUP as your home group.

Why do that?

For the purposes of these lessons, it is simply more convenient if a group called MYGROUP is your home group. That way you can be assured of working in MYGROUP whenever you log on without specifying a group. If you do not have MYGROUP and OTHERGRP,

but are using two different groups instead, you have to keep the difference in mind and adjust accordingly.

### Discovering your home group

To discover your home group, log on using just your user name and account name. Then enter `SHOWME` to find out what group you are in.

Do that now, using your own user name and account name:

```
HELLO username.accountname(Return)
```

You are now in your home group.

Now enter:

```
SHOWME(Return)
```

Can you find the group name in the display?

Another way to discover your home group is to enter the command `LISTUSER` at the system prompt.

Try this:

```
LISTUSER(Return)
```

Do you see something like this on your screen?

```
:LISTUSER ***** USER: ETHEL.MERTZ

HOME GROUP: MYGROUP PASSWORD: ** MAX PRI : 150 LOC ATTR:
$00000000 LOGON CNT : 1 CAP: ND,SF,BA,IA
```

The `LISTUSER` command provides you with some valuable information about you as a user of the MPE/iX system. But for now, just note what your home group is. Is it `MYGROUP` or something else?

### PUB as your home group

It is tempting to have `PUB` designated as your home group. A lot of users could share `PUB` as the home group in an account. After all, every file in `PUB` is accessible to everyone unless the file is locked; however, using `PUB` as your home group is not a good idea. Why not?

Suppose that you worked in personnel. Would you want to keep confidential files about persons in your firm in the `PUB` group, where other users might be able to read them? Most likely not.

Keep this in mind when using the `PUB` group. If you have access to a group *other* than `PUB` (your account manager has the ability to create one for you), make a practice of keeping your confidential files in the other group.

#### Note

---

The simplest logon, `HELLO username.acctname(Return)`, works only if the user has a designated home group.

---

## Lesson 1 Exercises

1. Give three reasons why there is an account structure on the MPE/iX system.
2. What four elements make up the MPE/iX account structure?
3. Given the following logon identities, correctly identify each session name, user name, group name, and account name.
  - a. HELLO norma.desmond,sunset
  - b. hello bob,carol.ted,alice
  - c. HELLO MGR.MKTG
  - d. hello emily.kado,pres
  - e. HELLO BLDG214,john.doe,audit
4. Unless you specify another group upon logon, you are automatically logged on to which group?
5. Pretend that you are JAN, assigned to the SALES account. Your home group is SUPPORT. What must you enter when you log on?
6. Pretend that you are CLERK, assigned to the DESK account and that you want to identify your session as FOLDER. What must you enter when you log on to your home group?
7. Pretend that you are USER1, assigned to the PROG account. You want to look at a file in the TEST group, which is *not* your home group. What must you enter when you log on?
8. Pretend that you are BILL, assigned to the PAYROLL account and the EXEMPT group, which is *not* your home group. What must you enter when you log on?
9. Explain why it is not a good idea to keep your files, especially sensitive ones, in the PUB group.

---

## Lesson 2 Security and the Account Structure

Lesson 2 presents the security provisions of the MPE/iX operating system:

- command security and capabilities
- logon security
- file security
- group security

MPE/iX provides several levels of security to ensure that your work remains private.

### MPE/iX command security and capabilities

There are more than 200 MPE/iX commands. You can use most of them, but not all of them. Some commands are restricted to certain users.

The **capabilities** assigned to you as a user determine which commands you are able to execute. These capabilities are assigned by your system operations personnel when they create your user, group, and account.

The capabilities assigned to you are usually determined by the type of work that you will do on the computer.

- Most commands are available to every user.
- Some commands are available only to those who manage your account, for example, account manager (AM).
- A number of commands are available only to those who manage your system, for example, system manager (SM).
- A few commands are restricted to users who have specific jobs, for example, the system operator (OP).
- Still other commands are available to all users, but the full power of the command is entrusted only to users with highly advanced capabilities.

At the system prompt, type the LISTUSER command. Your screen will look like this:

```
:LISTUSER ***** USER: ETHEL.MERTZ

HOME GROUP: MYGROUP PASSWORD: ** MAX PRI : 150 LOC ATTR:
$00000000 LOGON CNT : 1 CAP: ND,SF,BA,IA
```

The display for LISTUSER shows the capabilities that you have been assigned. Find the heading CAP. What do you see?

The capabilities listed in table 4-1 are the ones assigned to most new users.

**Table 5-1. The Basic (Default) Set of Capabilities**

| Capability | Meaning                        | What You Can Do                                                                            |
|------------|--------------------------------|--------------------------------------------------------------------------------------------|
| ND         | Access to Nonshareable Devices | Share printers and other devices that cannot be used simultaneously by two different users |
| SF         | Save Files                     | Save or keep your work file on the computer's disk drive(s)                                |
| BA         | Batch Access                   | Create and run jobs (a topic in the <i>900 Series HP 3000: Advanced Skills</i> course)     |
| IA         | Interactive Access             | Work in a session (work interactively with the computer)                                   |

If you try to use an MPE/iX command, but do not have the *right* capability to access it, you get a system error message.

Try the NEWACCT command, which creates new accounts on the system. For this example, try to create an account called LEARNING and assign a manager name (MANAGER) to the new account.

Enter:

```
NEWACCT LEARNING,MANAGER(Return)
```

Unless you have SM capability, your screen should look like this:

```
:NEWACCT LEARNING,MANAGER
THIS COMMAND REQUIRES SYSTEM MANAGER (SM) CAPABILITY. (CIERR 956)
```

Some commands affect fundamental or important operations of the computer. Misuse of those commands could harm the work of other users or interrupt the system.

The help facility states whether the command is available to everyone on the system or only to users who have special, advanced capabilities.

## **Logon security**

When you log on, the computer might prompt you to enter passwords:

- an account password
- a user password
- a group password

You might be prompted to provide only one of these, or two of them, or all three of them.

MPE/iX systems give you three chances to enter the right password. After a third failure, you see an error message. When that happens, you must start the logon procedure all over again. An incorrect entry of a password would look like this on the screen:

```
MPE XL:HELLO JOHN.SMITHERS
```

ENTER ACCOUNT PASSWORD:

ENTER ACCOUNT PASSWORD:

ENTER ACCOUNT PASSWORD:

INCORRECT PASSWORD. (CIERR 1441)

MPE XL:

*Passwords do not appear on the screen when you enter them in response to a password prompt.*

You get an error message if you attempt to log on to an account or group that does not exist:

MPE XL:HELLO JOHN.SMITHERS, TRAINING  
ACCT/USER EXIST, GROUP NAME DOESN'T. (CIERR 1436)

You also get an error message if you attempt to log on as a user who doesn't exist:

MPE XL:HELLO BETSY.SMITHERS  
ACCT EXISTS, USER NAME DOESN'T. (CIERR 1438)

---

## Lesson 2 Exercises

1. Suppose that you are trying to execute a command at the system prompt. The following message appears:

THIS COMMAND REQUIRES ACCOUNT MANAGER (AM) CAPABILITY.  
(CIERR 957)

What does this message mean?

2. How many times does the system prompt you for any one password?

---

## Lesson 3 File and Group Security

Lesson 3 presents the MPE/iX security provisions for files and groups.

- file security in your home group
- file security in the PUB group
- file security in other groups

MPE/iX protects files by restricting access to them.

With only the default set of capabilities, you cannot log on to one group and use a file that is located in *another* group.

Some exceptions exist:

- The file you want to work with has been released with the RELEASE command.
- The group in which the file is located is either the PUB group or your home group.

With only the default set of capabilities, if you can log on to a group, you can use any file in that group.

These are exceptions:

- The file is protected by being locked with a lockword.
- The file is protected by some other system security measure.

---

### Note

Be aware that your system manager *can* modify the security provisions on your system to meet the needs of your organization.

---

### File security in your Home group

With only the default set of capabilities, you can always access (read, change, save) any file in your home group, no matter what group you log on to.

### File Security in PUB

With only the default set of capabilities, you can at least read (and probably change) any file in the PUB group, no matter what group you log on to. But in most cases, you cannot then save that file *back* to the PUB group unless you have advanced capabilities, or unless the file has been released with the RELEASE command.

You can, however, save this “borrowed” file to your current group, the one in which you are actively working.

Using your logon identity, log on to your PUB group:

```
HELLO username.accountname,PUB' '(Return)
```

---

### Note

The following exercise only works if you *do not* have AM capability.

---

Create an EDIT/3000 file that contains the following three lines:

```
This is the file called REPORT.
It is very short.
The end.
```

Keep this file as REPORT and exit the editor.

Log on to your OTHERGRP group:

```
HELLO username.acctname,OTHERGRP(Return)
```

Get into the editor again:

```
EDITOR(Return)
```

Try to text in REPORT:

```
TEXT REPORT(Return)
```

The problem is that REPORT is in the PUB group, and you are now in OTHERGRP.

MPE/iX assumes that the file that you want to work on is in your *current* logon group. To tell MPE/iX that REPORT is in *another* group, you need to specify the REPORT file with a *partly qualified* file name.

A partly qualified file name is the file name plus a period plus the name of the group in which the file resides. The partly qualified form of REPORT is REPORT.PUB

Do this:

```
TEXT REPORT.PUB(Return)
```

Because the REPORT file is located in the PUB or public group, you are able to access and text it into your work session in the OTHERGRP group.

Enter:

```
LIST ALL(Return)
```

Add the following lines to the REPORT file. (Use the ADD subcommand.)

```
This is the REPORT file in the PUB group.
```

The goal now is to save (KEEP) REPORT back to the PUB group.

The problem is that you are still in group OTHERGRP. Do this:

```
KEEP(Return)
```

The editor remembers that the file came from the PUB group, so it attempts to save the edited version of the file back to the PUB group.

Enter YES when you are asked whether to purge the old REPORT file.

This should appear on the screen:

```
REPORT.PUB REPORT.PUB ALREADY EXISTS - RESPOND YES TO PURGE OLD
AND KEEP NEW PURGE OLD?YES *60*FCLOSE FAILURE (93)
SECURITY VIOLATION (FSERR 93)
```

This security violation occurred for the following reasons:

- You can look at a file in the PUB group. You can text it into the editor's workspace and edit it there. But if PUB is not your *home group*, you are not allowed to save the file back into the PUB group. (Saving a file really involves the opening and closing of a file, hence the wording of the error message.)
- Without advanced capabilities (AM, for instance), you cannot then *save* the altered file *back to* the PUB group.
- The file REPORT in the PUB group remains unchanged, despite the changes in the file held in the workspace for editing.

This prevents unauthorized changes to the original file in its original group (the PUB group in this case), however, it is possible to save the file in the group to which you are logged on.

Now try saving the file as REPORT, without the PUB qualifier:

```
K REPORT.OTHERGRP(Return)
```

REPORT has now been kept in your OTHERGRP group as a *new* file.

## File security in other groups

Stricter security applies to any group that is not PUB or your home group.

Log back on to your PUB group. Get into the editor and try to text in the REPORT.OTHERGRP file. What happened? You should see the following on your screen:

```
+--F-I-L-E---I-N-F-O-R-M-A-T-I-O-N---D-I-S-P-L-A-Y+ ! ERROR
NUMBER: 93 RESIDUE: 0 ! ! BLOCK NUMBER: 0 NUMREC: 0 !
+-----+ *23*FAILURE
TO OPEN TEXT FILE (93) SECURITY VIOLATION (FSERR 93) /
```

If you are not the creator of the file (the *username.accountname.groupname* that the system recognizes as the creator of the file), the system informs you that you have attempted to violate a security provision.

You cannot read, copy, or edit and keep a file from a non-PUB group that is not your home group unless:

- You are the manager of the account.
- You are the creator of the file releases the security provisions that protect that file.
- You are logged onto that group.

You learn how to release those security provisions in module 5, “Working with Files.”

### Note

---

The computer recognizes the *creator* of a file as the *user:account* identity of the person who first created the file.

---

The concept of account security becomes a little clearer in the context of copying and deleting files. That is why the subject comes up again in module 5, “Working with Files.”

Remember: if you have files that must be kept entirely secure, keep them in a group *other* than PUB.

## Lesson 3 Exercises

1. Choose true or false for items a through d.
  - a. A user with only the default capability set *cannot* read a file in the PUB group of the account.
  
  
  
  
  
  
  
  
  
  
  - b. You can read, change, and save any file in your home group.

- c. All users of an account can alter any file in the PUB group and save those files back to the PUB group.
  
- d. If you are logged on to a group other than your home group, you can read a file in your home group.

---

## Lesson 4 Changing Groups

Lesson 4 presents the CHGROUP command:

- changing groups by logging on
- changing groups with the CHGROUP command

There are two ways to change the group that you are accessing.

- At the colon prompt, use HELLO to log on again and specify the group that you want to be in.
- At the colon prompt, use the CHGROUP command and specify the group that you want to be in.

### Logging on

Logging on to another group certainly works, but it is time-consuming and uses a lot of system resources.

For example, this is how you would log on to your OTHERGRP group:

```
HELLO username.acctname,OTHERGRP
```

Now, using your logon identity, log on to your home group:

```
HELLO username.acctname(Return)
```

*Recall that your home group is your logon default.*

### Using the CHGROUP command

The CHGROUP command allows you to switch from one group to another without logging on again.

To see which group you are in, enter:

```
SHOWME(Return)
```

Switch into your PUB group by entering:

```
CHGROUP PUB(Return)
```

Use SHOWME to discover which group you are now in. You should be logged on to the PUB group now.

---

### Note

If a group in your account is protected with a password, you will be prompted for that password. You must correctly provide the password in order to access that group. You will not be prompted for a password when changing back to your home group.

---

Use the CHGROUP command now to switch into OTHERGRP.

Enter:

```
CHGROUP OTHERGRP(Return)
```

Now enter:

```
SHOWME(Return)
```

Which group are you in now? You should be logged on to OTHERGRP now.

## **CHGROUP and your home group**

If you execute CHGROUP without specifying another group, you are switched automatically to your home group.

This time enter:

```
CHGROUP(Return)
```

Enter:

```
SHOWME(Return)
```

You should be logged on to your home group (MYGROUP).

---

### **Note**

Have you noticed the parallel? Both CHGROUP and HELLO *default* to your home group if you do not specify a group.

---

### **Lesson 4 Exercises**

1. Name the two methods of changing groups from within your account?
2. Choose either method to change into another group within your account. Check to see if you were successful. Get back into your home group.
3. Repeat step 2 using the other method.

---

## Lesson 5 Changing User Passwords

Lesson 5 presents the PASSWORD command:

- changing your user password

### Your user password

The very first time that you tried to log on to your MPE/iX computer, you probably had to enter your user password in order to log on successfully.

That user password already existed, because your system manager or your account manager assigned that password to your user name. It might be a password that you have trouble remembering, or you might simply like to have a different user password.

You can change your user password if you want to do that—but do not do it so frequently that you have trouble remembering your user password.

If you do not have a user password, you can create one for yourself.

### Using the PASSWORD command

In order to change an existing user password, you must know what that password is.

If you are logged on in your account and your group, you would use the PASSWORD command in this fashion:

```
PASSWORD(Return)
```

If a password already exists, the computer will ask you to enter the existing password, to prove that you know what it is:

```
ENTER OLD USER PASSWORD:
```

Enter the correct password and press (Return). The word that you enter will not appear on your screen. That is to protect you from having anyone see the password that you enter.

If you have entered the existing password correctly, PASSWORD next asks you to enter the new password—the password that you want to have attached to your user name:

```
ENTER NEW USER PASSWORD:
```

Enter the password that you want. The password must begin with a letter of the alphabet. After that first letter, you may add as many as seven more characters—letters or numbers or a combination of letters and numbers.

Again, you will not see the password you enter. Enter it carefully.

Now the computer asks you enter the new password one more time:

```
ENTER NEW USER PASSWORD AGAIN:
```

This step is to ensure that you and the computer agree on what the new password is to be. You will not see the password that you enter.

If the password you enter this time matches the password you entered in the previous step, the computer changes your user password and confirms the change:

```
PASSWORD WAS CHANGED SUCCESSFULLY
```

The change does not become effected until you log off and log back on again.

If the password you enter this second time is different from the password you entered in the step before this, the computer assumes that you made a mistake, stops the password changing process, and gives you a message:

```
NEW PASSWORD IS NOT CONSISTENT. (CIERR 2503) PASSWORD WAS NOT CHANGED.
:_
```

In this case, the old, existing user password—whatever it was—remains the same. If you still want to change your user password, you must use the `PASSWORD` again.

If your user name does not have a password, and you would like to create one, use the `PASSWORD` command.

If there is no existing password for your user name, the computer begins by asking you to enter a (new) password. Then you will be asked to enter the (new) password one more time, to confirm your choice.

## Lesson 5 Exercises

1. Assume that you are logged on as a user whose password is `USER7`. How would you change this user password to `EDWARD`?



## Module 5: Working with Files

---

Module 5 introduces common tasks needed for using files.

|          |                                 |
|----------|---------------------------------|
| Lesson 1 | Displaying Qualified File Names |
| Lesson 2 | Renaming Files                  |
| Lesson 3 | Storing Seldom-Used Files       |
| Lesson 4 | Deleting Files                  |
| Lesson 5 | Displaying File Contents        |
| Lesson 6 | Copying Files                   |

---

### Challenge Test

1. To refer to a file in another group, a file needs to be qualified with what?
2. To refer to a file in another group and in another account, a file needs to be qualified with what?
3. How would you list to your terminal screen all of the files that begin with "A" in the group you are presently in?
4. What MPE/iX command allows you to change a file name?
5. Where should seldom-used files be kept for storage?
6. Which MPE/iX command allows you to delete files from your account?
7. Which MPE/iX command allows you to display the contents of a file to your terminal screen?
8. Which MPE/iX command allows you to copy one file to another, either by creating a new file or by overwriting an existing file?

9. Which MPE/iX command removes all of the security provisions of a file to allow a file to be copied by another user? Which MPE/iX command restores security provisions to a file?
  
10. From whom should you request more disk space on the HP 3000?

---

## Lesson 1 Displaying Qualified File Names

Lesson 1 presents displaying file names.

- accessing files from other groups
- displaying file names on the screen: LISTFILE
- using **wildcard** characters to display qualified file names to the screen

---

### Note

Many of the lessons in this module refer to files that were created in module 3 of this tutorial.

---

### Accessing files in other groups and accounts

A time may come when you need to use a file that resides in another group or account.

- In order to use a file from another group, you must use the file name plus the name of the group in which the file is found.
- To use a file in another account, you must use the file name plus the group name plus the account name.

### Partly qualified name

This technique of adding a group name creates a **partly qualified** file name (DOCUMENT.MYGROUP).

### Fully qualified name

A **fully qualified** file name is the file name, plus a group name, plus an account name. This information instructs the system to look for the file in the designated group and account. (DOCUMENT.MYGROUP.ANYACCT).

### Forming a fully qualified file name

You need to know:

- the file name (ACCTSREC)
- the name of the group where the file is found (MYGROUP)
- the name of the account where the file is found (ANYACCT)

The fully qualified name of the file above is

ACCTSREC.MYGROUP.ANYACCT

The punctuation is important.

---

### Note

Did you note the period or dot ( . ) between each part of the name? Each part of a qualified file name must be separated from another part with a period.

---

You can enter the file name in uppercase or lowercase letters or in a combination of uppercase and lowercase letters.

**When to use the fully qualified file name**

How much of the fully qualified file name you need to use depends upon where the file is found and how you logged on.

**Table 6-1. Qualified File Names**

| <b>You need</b>                 | <b>If</b>                                                                                                                                                                                |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Simple file name alone          | You are logged on to the group in which the file is found.                                                                                                                               |
| filename.groupname              | You are logged on to the same account but in a different group from the one in which the file is found.                                                                                  |
| filename.groupname. accountname | You are logged on to a different account from the one in which the file is found.<br><br>When you specify a file in a different account, you must specify the fully qualified file name. |

**File name and logon—the differences**

Logon identifiers and fully qualified file names are very similar in appearance, and the similarity *can* be confusing.

In fact, something like ROTOR.CAD could be either of these:

- a file name and a group name
- a user name and an account name

The only way of telling which it is, is to examine the way it is used. What command precedes it? If the command is HELLO, you know that you are looking at a logon identifier; that is, a user name and an account name.

If a text or file-handling command precedes it (COPY, PURGE, RELEASE, SECURE, which you learn about later in this module), you know that you are looking at a file name and a group name.

**Fully qualified name**

If you see ROTOR.CAD.MANUF, you would know immediately that you are looking at a fully qualified file name.

How would you know?

- The only valid delimiter (punctuation) in a file name is the *period*, or dot ( . ).
- A logon identifier having more than *two* elements must contain at least one *comma* delimiter.

## The patterns

Examine the two patterns together (the extra spaces are for clarity only).

LOGON:

*session , user . account , group*

FILE NAME:

*file . group . account*

Recall that a session name and a group name are optional in a logon identifier. Those optional elements are always set off from the user name and account name by a *comma* ( , ).

So . . . .

JJ,ROTOR.CAD—is a session name tied to a logon identifier.

ROTOR.CAD,DESIGN—is a logon identifier tied to a group name.

JJ,ROTOR.CAD,DESIGN—is a session name tied to a logon identifier tied to a group name.

JJ,ROTOR.CAD.DESIGN—cannot exist.

JJ.ROTOR.CAD,DESIGN—cannot exist either.

JJ.ROTOR.DESIGN—can only be a file name.

JJ.DESIGN—cannot be determined out of context.

## Displaying all of the file names in your group

Do you recall what command shows you all of the files in your current group?

Log on to your PUB group.

Enter:

```
LISTFILE(Return)
```

Do the file names in the PUB group appear on the screen? They should.

If you work in more than one group, it may become difficult to keep track of all of your files.

Fortunately, LISTFILE can be selective about the file names that it displays. It can also display file names outside your current group.

## Displaying selected file names in your group

Log on to your home group. Your logon group should be MYGROUP (or its equivalent if you do not have MYGROUP).

Using your *username* and *accountname*, enter:

```
HELLO username.accountname(Return)
```

MPE/iX allows you to display file names that are alike in some way. This is useful if you have established a naming convention for files that contain related information.

To display all of the file names that begin with D, do this:

```
LISTFILE D@(Return)
```

You will see the names of all of the files in your home group that begin with the letter “D.”

### The @ symbol

The @ Symbol (Shift2 on your keyboard) is the symbol for *all combinations of letters*. The symbol @ is called a **wildcard** character. In some card games, a “wild card” can take on any legitimate value that the player wants.

In MPE/iX, @ can represent anywhere from zero to eight characters and take on all of the possible letter and number values (A - Z and 0 - 9).

D@ means, literally, all of the files that begin with the letter “D” followed by any other combination of zero to seven characters.

What do you suppose this will show?

```
LISTFILE @2@
```

Try it.

It shows all of the file names that contain the number 2, if there are any.

- The first @ tells the computer to search for any file that begins with any combination of letters.
- The last @ tells it to search for any file that ends with any combination of letters.
- But the 2 in the middle restricts the search to file names that have the number 2 in them.

### Display file names in other groups and accounts

You can use the @ symbol to have LISTFILE display the file names found in other groups and accounts.

To see all of the file names in PUB group, enter:

```
LISTFILE @.PUBReturn
```

The names of the files in PUB should appear on the screen.

To see the names of files in OTHERGRP that begin with R, do this:

```
LISTFILE R@.OTHERGRPReturn
```

Now enter this:

```
LISTFILE @.@Return
```

You should see all of the file names in all of the groups in your account. If there are many files, the names will scroll by quickly. Use STOP or CTRLS to stop the scroll. Use another STOP or CTRLQ to resume the scroll.

Or, use PREV to jump backwards, one screen at a time. Use NEXT to return to your starting position.

## Note

---

For the next example, be ready to use `(BREAK)`.

---

What happens if you enter this?

```
LISTFILE @.@.(Return)
```

You should see *all* the file names in *all* the groups in *all* the accounts on your MPE/iX system—and that could be many files.

To stop the display, press the `(BREAK)` key.

To restrict the search to some account, such as SYS, you could specify:

```
LISTFILE @S.PUB.SYS
```

You would see all of the file names ending in the letter “S” that are found in the PUB group of the SYS account. This is a fully qualified file name, but the wildcard character @ is used to specify *a set* of the files in the PUB group of SYS.

What would you see if you entered this?

```
LISTFILE K@.@.SYS
```

You would see all of the files that begin with the letter K in all of the groups of the SYS account.

Be ready to use `(BREAK)`. Try this:

```
LISTFILE D@.MYGROUP.@(Break)
```

You should see a listing of all of the files beginning with the letter “D” in any group called MYGROUP anywhere on your system.

## Finding files

Are you having difficulty recalling exactly which group contains the file you are looking for?

There is a solution, and it involves another way of using the LISTFILE command.

Recall that @ means *all file names*. Recall, too, that @@ means *all of the files in all of the groups* of the account to which you are logged on.

Now do this:

```
LISTFILE @.@,6(Return)
```

What happened? Did you get a list of file names, perhaps quite a few file names, that are in the account to which you are logged on? All of the names are in their *fully qualified* form.

Now that you have the fully qualified file name, you can see at a glance not only the name of the file, but the group and account in which the file is found.

To find the location of one particular file, try this:

```
LISTFILE filename.@,6(Return)
```

The ,6 tells LISTFILE to list the *fully qualified* file name that identifies (and *locates*) the file called FILENAME.

The LISTFILE command has other features, not covered here. If you are interested in finding out what those features are, refer to the help facility.

## Lesson 1 Exercises

1. In each of the following instances, what file name would you supply to refer to the file, FILE1?
  - a. You are logged on to the same group and account where FILE1 resides.
  - b. You are logged on to the TEST group in the PROG account and FILE1 resides in the PAYROLL group of the ACCT account.
  - c. You are logged on to your home group (not PUB) in the MYACCT account and FILE1 resides in the PUB group of the same account.
  - d. You are logged on to the PUB group of the MYACCT account and FILE1 resides in the PUB group of the SYS account.
2. If you are logged on to your home group, how would you list to your terminal screen all of the files in this group?
3. How would you list to your terminal screen all of the files in your present group that begin with the letter "A"?
4. How would you list to your terminal screen all of the files in your present group that end with the letter "S"?
5. How would you list to your terminal screen all of the *fully qualified* files names in your account?
6. How would you list all of the files residing in the PUB group of the SYS account that start with the letter "S", contain the letter "R" and end with the letter "T"?

---

## Lesson 2 Renaming Files

Lesson 2 presents renaming files.

- the command to give a new name to a file: `RENAME`

To rename the file called `DOCUMENT` in your home group, first make sure you are logged on to your home group. You must also be the creator of the file in order to perform a file name change.

Enter:

```
CHGROUP(Return)
```

Then do this:

```
RENAME DOCUMENT,ADOCONE(Return)
```

### Note

---

Notice the punctuation. The comma ( , ) has to be there.

---

Use `LISTFILE` to check that `DOCUMENT` has disappeared and that a file called `ADOCONE` has appeared.

```
LISTFILE(Return)
```

### Safeguard against duplication

Now use `RENAME` to change the name of the file `DRAFT` to `ADOCONE` and see what happens when you enter:

```
RENAME DRAFT,ADOCONE(Return)
```

What happened? Did you get this error message?

```
DUPLICATE PERMANENT FILE NAME (FSERR 100)
RENAME FAILED DUE TO FILE SYSTEM ERROR, NOT RENAMED. (CIERR 373)
```

The file names in any *group* are always unique; there can be no duplication. When the computer discovered there was a file already named `ADOCONE`, it refused to change the name of `DRAFT` to `ADOCONE`.

Use `LISTFILE` again to verify that no change has taken place in the list of files.

```
LISTFILE(Return)
```

Now change the name of `ADOCONE` back to `DOCUMENT`.

Enter:

```
RENAME ADOCONE,DOCUMENT(Return)
```

### Lesson 2 Exercises

1. Create two files, each one consisting of at least 10 lines. Name one file `OAK` and the other `MAPLE`.

\*Try to rename the file `OAK` to `MAPLE`. What happened?

2. Rename the file `OAK` to `ACORN`. Check to see if the rename was a success.

---

## Lesson 3

### Storing Seldom-Used Files

More than likely you will be aware of the files that you use every day or every week.

You are likely to forget about files that you used last month, or the month before, that that you have not used at all this month.

Those files take up space on the computer's disk. Although the computer has a lot of disk space, the space is not endless.

- Look over the files in your group every now and then.
- Identify those files that you do not use any more or only seldom use.

Ask yourself:

- Is this older file going to be needed *some day*—and if so, *when*?
- Is this older file just taking up disk space?

For seldom-used files that you want to save, do this:

- Ask your system operator to store the files onto magnetic tape.
  - Ask your operator to create a printout of the names of the files stored to tape.
  - Ask for the name or number of the tape and note it down somewhere. A good practice is to put the name or number of the tape on the printout of stored files.
  - Keep the printout of the stored file names in a safe place, so that you can have someone restore the files to the computer if it is necessary.
- Erase from the disk those files that are safely stored to tape. You do that in Lesson 4, “Deleting Files.” You can ask your operator to erase the files that are stored to tape.
- Finally, erase any files that neither you nor anyone else will ever need again. You do that in Lesson 4, “Deleting Files.”

Keeping your files current—and not taking up useful disk space—is good file management. Your system operator will appreciate that.

### Lesson 3 Exercises

1. Explain why it is a good practice to store large, rarely used files to a magnetic tape.

---

## Lesson 4 Deleting Files

Lesson 4 presents deleting files.

- the command to erase a file: PURGE
- recovering from an accidental file erasure

### The PURGE command

The PURGE command erases a file. It erases only one file at a time. If you want to erase five files, you need to use the PURGE command five times.

Make sure that you are in the home group of your account. Recall from module 4 that the LISTUSER command shows which group is designated as your home group.

Enter:

```
LISTUSER(Return)
```

---

### Note

LISTUSER does not tell you which group you are currently in. It tells you only which group is designated as your home group.

---

To find out which group you are currently in, use the SHOWME command.

Enter:

```
SHOWME(Return)
```

Use CHGROUP if necessary get into your home group.

To purge the file called NEWDRAFT, enter:

```
PURGE NEWDRAFT(Return)
```

Use LISTFILE to make sure that NEWDRAFT has been removed.

Now erase NXTDRAFT in the same way.

```
PURGE NXTDRAFT(Return)
```

If you try to erase a file that does not exist, you get an error message.

Try to erase the file QZQXQZ. What happens? (Of course, if the file does exist, it is erased. MPE/iX assumes that you know what you want to do when you issue a command.)

### Recovering a file

The day will come when you accidentally erase a file that you need. It happens to almost everyone at some time.

MPE/iX does *not* provide an *unerase* command. MPE/iX assumes that you knew what you were doing when you erased a file.

There is a remedy. Your system operations personnel should perform regular system *backups* (storing all of the files on the system to tape).

- If you created the file *after* the last backup, you cannot retrieve the file.
- If you created the file *before* the last backup, you can restore the file to the computer.

- If you created the file *before* that last backup but you modified it *after* the last backup, you can restore the earlier version of the file; however, the modifications are lost.

For this exercise, suppose the following:

- Your system operator regularly backs up files on the system every day at 6:00 P.M.
- You made changes to MYFILE yesterday (Monday).
- Today is Tuesday, 11:00 A.M.
- You have been working on MYFILE all morning.
- You accidentally erased MYFILE five minutes ago.

What would you do?

- Ask your system operator to have MYFILE restored to the *group* and *account* where it belongs.

What is restored?

- All of the changes made to MYFILE on Monday are restored. (The backup was performed at 6:00 P.M. yesterday.)
- You have lost the work that you did today (Tuesday), and you need to reenter those changes.

How much work is lost depends on two things:

- how much time has passed since you made changes to the file and kept it
- how much time has passed since the files on your system were stored to tape in a backup

---

**Note**

System operation departments have different backup schedules, but all *should* perform regular file backups. Find out from your manager or system operations department what the backup schedule is. One day you may need to know.

---

## Lesson 4 Exercises

1. Delete the file MAPLE from your account. Check to see if you were successful.

2. Consider the following scenario:

In your office, the backup of files is done daily after 5:00 P.M. It is now 3:00 P.M. on Friday. You have just completed updating a file called STATUS. For the past week, you have been updating this file daily.

While purging other files from your account, you accidentally purged the STATUS file. Which one of the following statements is true?

- a. Two days' worth of updated material has been lost and cannot be restored to the system.
- b. Today's updated material has been lost and cannot be restored.
- c. The entire file is lost and cannot be restored, but Thursday's work can be restored.
- d. Nothing has been lost.

---

## Lesson 5

### Displaying File Contents

Lesson 5 presents the command to display the contents of text files.

- the command to display the contents of a text file: PRINT

Up to now you have used the editor to see a file, but you can display the contents of a text file on the screen without starting the editor to text in the file and list it.

You should be in MYGROUP.

#### Displaying an entire file

To see the contents of DOCUMENT on the screen, use PRINT command this way:

```
PRINT DOCUMENT(Return)
```

#### Displaying selected lines

To see only lines 3 through 6 of the file, enter this:

```
PRINT DOCUMENT;START=3;END=6(Return)
```

---

#### Note

Notice the punctuation? The semicolons ( ; ) are needed here.

---

#### The PRINT parameters

Here is what happens:

- START= tells the computer which line of the file to show first.
- END= tells the computer which line to show last.
- The semicolons ensure that the computer knows where one parameter ends and another begins.

What do you suppose the following prints to the screen:

```
PRINT DOCUMENT;START=-3
```

Try it.

The last three lines of the file DOCUMENT are printed on the screen. This happens because you specified a minus sign (-) ahead of the number 3, so the instruction becomes: *start printing at the third line from the end.*

---

#### Note

New users of MPE/iX should *always* specify a file name when using the PRINT command. Entering PRINT(Return) puts MPE/iX into a *create mode*, and that causes confusing results on the terminal screen. If you do this by accident, enter :eod (colon included) by itself on a new line and press(Return).

:EOD (colon included) by itself on a new line sends a message to the computer—End Of Data.

---

#### Lesson 5 Exercises

1. Display the file ACORN to your terminal screen. \*Display lines 3 through 7 of the file ACORN to your screen.
2. What method would you use to print the file ACORN to a line printer?

---

## Lesson 6

### Copying Files

Lesson 6 presents the commands needed to copy files.

- the command to copy a file: COPY
- the command to remove system security for a file: RELEASE
- the command to restore system security for a file: SECURE
- conserving disk space

#### Using the COPY command

To copy a file, use the COPY command, specifying the name of the file that you want to copy and the name that you want the new, identical copy to have.

First you must be in the group in which the file is found.

Enter:

```
LISTFILE DOCUMENT.@,6(Return)
```

Find the group in which the file called DOCUMENT is located.

If necessary, use CHGROUP to move into that group.

To make a copy of DOCUMENT, calling the new copy ANEWDOC, do this:

```
COPY FROM=DOCUMENT;TO=ANEWDOC(Return)
```

Here is what the parameters do:

- The FROM= parameter tells the computer which file to copy from. The original, from which the copy is made, is called the **source** file.
- The TO= parameter tells the computer what to call the copy. The *to* file is called the **target** or **destination** file.

Use LISTFILE to verify that a new file called ANEWDOC appears. Use PRINT to look at the file:

```
PRINT ANEWDOC(Return)
```

#### A simpler COPY

The COPY command permits you to shorten the command line a little. You can leave out ;FROM= and ;TO=.

Do this:

```
COPY DOCUMENT,OLDDOC(Return)
```

---

#### Note

In place of the semicolon ( ; ) is a comma ( , ) to separate the source file from the destination file.

---

Use LISTFILE to be certain that a new file called OLDDOC now appears.

```
COPY DOCUMENT,OLDDOC
```

is the same as

```
COPY FROM=DOCUMENT;TO=OLDDOC.
```

## Note

---

A *few* commands such as COPY permit you to use a somewhat simpler command line. Those MPE/iX commands with an alternative, simpler **syntax** are noted in the help facility.

---

### Copy to an existing file

You used COPY the first time to create a new file, a duplicate of the source file.

Suppose that you now try to copy the file ANEWDOC back to the original called DOCUMENT. The *source* becomes the *destination*, and the *destination* becomes the *source* of the COPY operation.

Enter this:

```
COPY FROM=ANEWDOC;TO=DOCUMENT(Return)
```

or enter this:

```
COPY ANEWDOC,DOCUMENT(Return)
```

Do you see something like this on the screen:

```
PURGE OLD DOCUMENT.MYGROUP.ANYACCT?
```

The file DOCUMENT already exists. The computer needs instruction from you.

Copying from one file to an already existing file implies the *destruction* of the *destination* file.

### Copy over an existing file

In order to copy ANEWDOC into the existing file DOCUMENT, the computer must do this:

- First destroy the contents of DOCUMENT.
- Then copy the contents of ANEWDOC into the newly emptied *destination* file.

This process is called **overwriting**. Before doing that, the computer needs instructions from you.

What to do:

- If you want to continue the COPY process (permit the overwriting—and destruction—of the destination file), enter YES or Y at the prompt question.
- If you want to stop the COPY process (prohibit the overwriting), enter NO or N at the prompt question.

This time, answer NO. This prevents the overwriting. You should see this on the screen:

```
:PURGE OLD DOCUMENT.MYGROUP.ANYACCT? NO
NO COPY WAS DONE (CIERR 9113)
:
```

Use COPY again, just as you did above, but this time, answer YES to the prompt. When you do that, you should see this on the screen:

```
:COPY FROM=ANEWDOC;TO=DOCUMENT
PURGE OLD? DOCUMENT.MYGROUP.ANYACCT? (Y/N) YES
:
```

## More practice with COPY

Make sure that you are in your home group.

Enter:

```
CHGROUP(Return)
```

---

### Note

Recall that files can be locked. A file that has been locked with a lockword cannot be copied by anyone who does not know how to unlock the file. The topic of locked files is covered in the tutorial *900 Series HP3000: Advanced Skills*.

---

Do this:

Use LISTFILE to find the file names in the group that you are in, in the PUB group, and in some OTHERGRP (your third group).

To get a feeling for what you can and cannot do with COPY (at your current level of experience and authority), try some experiments:

- Copy one file in the group that you are in and give the copy a new name.
- Copy one file in the group that you are in and try to give it the name of another file that already exists.
- Copy one file from the PUB group to the group that you are now in.
- Copy one file from the OTHERGRP into the group that you are in.

You should succeed at the first three.

The fourth copying task should produce an error message. That is because the file you are trying to copy from the third group into your current group has not been released for copying.

## Releasing file security with the RELEASE command

The MPE/iX command that removes the security provisions for a file is RELEASE.

Use CHGROUP to get into your OTHERGRP group.

Use the editor to create a small text file called TEST. Write in it anything that you like. Then save the file and exit the editor.

To remove the security provisions of the file TEST, enter:

```
RELEASE TEST(Return)
```

---

### Note

In order for a user to release a file, the user must be the creator of the file, or the following error message appears:

```
ACTION DISALLOWED SINCE NOT CREATOR OF FILE (CIERR 351)
```

---

## Copying from another group

To copy a file from another group to the group that you are in:

- You must know the qualified name of the file that you want to copy.
- You must be the account manager of the account in which both groups reside.

*or*

- The *owner* of the file that you want to copy must release that file for you with the RELEASE command.

*or*

- Group security must be organized to allow you access to the group.

Do this:

```
COPY DOCUMENT.MYGROUP,DOCUMENT(Return)
```

Group security allows you to copy a file from your home group into your current group, without first having to release the file. You also do not need to specify the name of the group you are currently in if the file is to be duplicated in that group.

Now use CHGROUP to return to MYGROUP:

Enter:

```
CHGROUP(Return)
```

Then enter this:

```
COPY TEST.OTHERGRP,TEST(Return)
```

Use LISTFILE to check to see that a copy of TEST now appears in MYGROUP.

The copying worked this time because you had previously used the RELEASE command to remove normal security from the file TEST.

## Default names in copying

MPE/iX makes assumptions when you copy files.

### Default group name

MPE/iX assumes that the file that you want to copy is in your current group. If you want to copy a file from another group, you must specify the partly qualified file name (*filename.groupname*).

MPE/iX assumes that you want the *copy* of the file to reside in your current group.

In the example above, MPE/iX *knows* that you are logged on to MYGROUP. So, unless you specify otherwise, it assumes that you want the *copy* of TEST to reside in MYGROUP.

The command above could just as easily be accomplished with this form:

```
COPY TEST.OTHERGRP, TEST
```

The effect would be just the same. A copy of TEST would appear in MYGROUP. Try it with another file in your group OTHERGRP.

### Default file name

MPE/iX also assumes when you copy a file from another group that you want the file in your current group to have the same name.

`COPY TEST.OTHERGRP` puts a copy of `TEST` into `MYGROUP`. The copy created in `MYGROUP` is also called `TEST`.

### Copy and change file name

You can use `COPY` to copy a file from one group to another group and give it a new name in the destination group.

Do this:

```
COPY TEST.OTHERGRP,TEST2(Return)
```

The computer puts a copy of `TEST` from `OTHERGRP` into `MYGROUP` and calls the copy `TEST2`. In this case, MPE/iX assumes once again that you want the copy of the file in your current group; but, because you specified a new file name, `COPY` gives that name to the copy it put into `MYGROUP`.

Try copying in the other direction now.

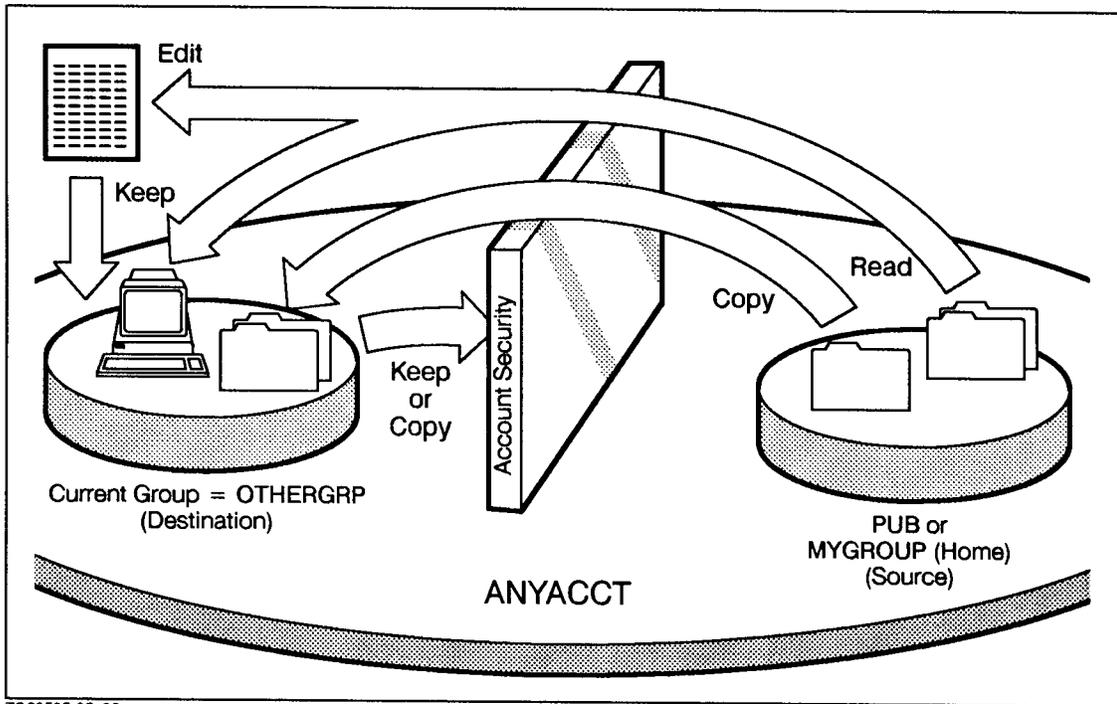
Do this:

```
COPY TEST2,TEST3.OTHERGRP(Return)
```

Do you get an error message? You should.

- You can *pull* a copy of a file from another group if the security has been released.
- You cannot *push* a copy of a file from one group to another group unless you have account manager capability, or unless you have security access to that group.

`PUB` is a special case of account security (Figure 5-1).



TG20535-05\_02

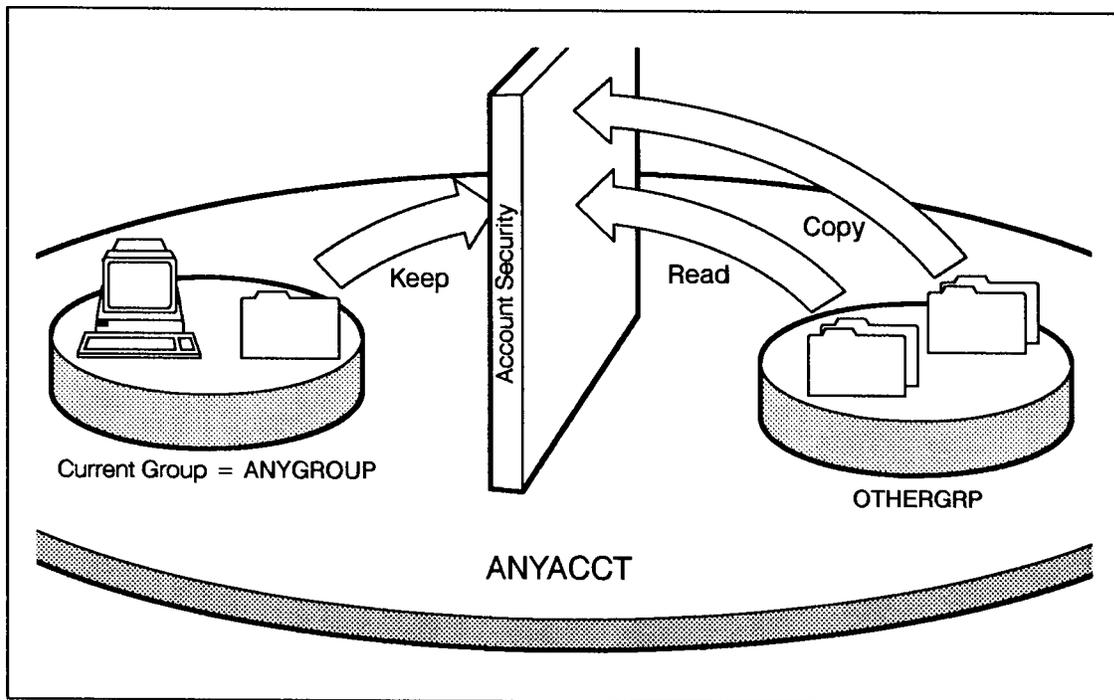
**Figure 6-1. PUB and Your Home Group: Relaxed Security**

Files in the PUB group are available to users in other groups, unless those files have been protected in some special way (by a lockword, for instance).

All of the files in your home group are available to you (the user).

Notice, though, that you cannot push a file to the PUB group (keep a file into the PUB group while working in MYGROUP) unless you have account manager capability, or unless you have security access to that group.

Standard security is maintained between other groups.



TG20535-05\_03

**Figure 6-2. Other Groups: Strict Security**

Files cannot be read (looked at) or copied from a non-PUB, non-home group, unless those files are released, or unless you have account manager capability, or unless you have access to that group.

Similarly, you cannot push a file into another group unless you have account manager capability, or unless you have security access to that group.

**Group to group  
security:  
what works**

The concept of account security becomes a little clearer in the context of copying and deleting files.

What works depends on a number of factors. Assume that:

- MYGROUP is your home group
- you do not have AM capability
- the file that you want to work with has *not* been released with the RELEASE command

Tables 5-2, 5-3, and 5-4 illustrate what you can do, depending upon which group that you are currently working in.

**Table 6-2. You Are in PUB**

| <b>FILE LOCATED IN</b> | <b>READ FILE</b> | <b>SAVE OR ERASE IN ORIGINAL LOCATION</b> |
|------------------------|------------------|-------------------------------------------|
| PUB                    | Yes              | Yes                                       |
| MYGROUP                | Yes              | Yes                                       |
| OTHERGRP               | No               | No                                        |

**Table 6-3. You Are in MYGROUP (Your Home Group)**

| <b>FILE LOCATED IN</b> | <b>READ FILE</b> | <b>SAVE OR ERASE IN ORIGINAL LOCATION</b> |
|------------------------|------------------|-------------------------------------------|
| PUB                    | Yes              | No                                        |
| MYGROUP                | Yes              | Yes                                       |
| OTHERGRP               | No               | No                                        |

**Table 6-4. You Are in OTHERGRP (Not PUB, Not Home)**

| <b>FILE LOCATED IN</b> | <b>READ FILE</b> | <b>SAVE OR ERASE IN ORIGINAL LOCATION</b> |
|------------------------|------------------|-------------------------------------------|
| PUB                    | Yes              | No                                        |
| MYGROUP                | Yes              | Yes                                       |
| OTHERGRP               | Yes              | Yes                                       |
| ANOTHER GROUP          | No               | No                                        |

**Reinstating system security to a file**

Any file that you release loses almost all protection from other users, unless it has a lockword. Other users in other groups or accounts can do the following:

- read the file
- copy the file
- edit the file
- erase the file

**The SECURE command**

Once the file is copied, it should *always* have the system security restored to it. To do this, use the MPE/iX SECURE command.

Get into OTHERGRP.

```
CHGROUP OTHERGRP(Return)
```

Enter this to reinstate the security on TEST:

```
SECURE TEST(Return)
```

This action ensures that no unauthorized access of your file can occur. You can use LISTFILE to discover the *kind* of security that is in effect for a file.

Do this:

```
LISTFILE TEST,3(Return)
```

Do you get a screenful of information? Much of the information in this display is intended for advanced users.

Look in the right column, where it says SECURITY IS ON. This tells you that TEST has *not* been released.

Now release TEST:

```
RELEASE TEST(Return)
```

Check to see the security status of TEST.

```
LISTFILE TEST,3(Return)
```

Now, reinstate the security for the file TEST.

Enter:

```
SECURE TEST(Return)
```

## Conserving disk space

You may have given little thought to the possibility of running out of disk storage on the computer.

It *is* unlikely. But it *can* happen, especially if you keep copies of various versions of large files.

The system manager can set limits on the amount of disk space that those files can occupy.

If you run out of space, you get an error message on the screen.

```
OUT OF DISC SPACE (FSERR 46)
```

There are two things you can do:

- Ask your system manager to increase the amount of disk space that you are allowed.
- Free disk space by having some of your files stored to tape and then erase those files from the disk by using PURGE.

## Lesson 6 Exercises

1.
  - a. Using the file ACORN, which was created in an earlier exercise, make a copy of the file and call it PINE. Make editing changes to the file PINE so that it now looks different. Save the file again as PINE.
  - b. Try to copy the file PINE to the file ACORN. Answer NO to the prompt and look at the file ACORN. What happened?
  - c. Try copying the file again. Answer YES to the prompt and look at the file ACORN. What happened?
2. Which MPE/iX command removes all of the security provisions of a file to allow the file to be copied by another user in another group and account?
3. Which MPE/iX command restores all of the security provisions to a file?
4. If a file has *not* been resecured with the SECURE command, any user can:
  - a. purge the file
  - b. copy the file
  - c. read the file
  - d. all of the above
  - e. none of the above
5. If you do not specify a target file group when using the COPY command, to what group does the file get copied?
6. If you run out of disk space in your account, to whom should you go to request more disk space?





---

## Lesson 1 Using Command Files

Lesson 1 presents creating and using command files.

- what is a command file is
- how to create and use a command file
- command files that execute other command files

### What is a command file?

A command file:

- is a text file, just like any text file that you create
- may contain MPE/iX commands, program file names, and the names of other command files
- executes the commands, programs, or other command files that are named inside it

### Creating and executing command files

At its simplest, a command file is an MPE/iX command that you shorten. For example, instead of typing in the whole command SHOWTIME, that you can use the editor to create a command file called ST that contains the command SHOWTIME. You save it and leave the editor. Every time that you enter ST<sup>(Return)</sup> at the system prompt, SHOWTIME executes.

Get into the editor. Create a one-line file. The only line that you need is SHOWTIME.

It will look like this on the screen:

```
/ADD
 1 SHOWTIME
 2 //
/KEEP ST
/EXIT
```

You have a command file called ST.

Enter:

```
ST(Return)
```

What happens on the screen?

Another example of a one-line command file is PR. It prints the last three lines of the file DOCUMENT to the terminal screen. (Of course, you would have difficulty if the file has fewer than three lines.)

Get into the editor now. Create a new file by doing the following:

```
ADD(Return)
PRINT FILE=DOCUMENT;START=-3(Return)
//(Return)
KEEP PR(Return)
```

Exit the editor.

Execute the command file PR by doing the following:

```
PR(Return)
```

You should see on the screen the last three lines of the file DOCUMENT that you created in module 3.

Suppose that you created a command file that has this one line:

```
LISTFILE @.PUB
```

What would this command file do?

This command file would list all of the files in the PUB group of your account. If this file were to be kept as PUBLIST, look at how many keystrokes that you would save each time this command file was executed.

Create and execute this command file, PUBLIST.

### **Command files that execute other command files**

One command file can call upon and execute another command file. The example below shows what that you might put in a command file named PS. It executes the command file PR and then executes the command file ST.

Try it. Create the file in the editor, just as it appears below.

```
1 PR
2 st
```

Save the file as PS and exit out of the editor.

To execute it, enter the following:

```
PS(Return)
```

Do you see the last three lines of the DOCUMENT file and the date and time on the screen?

### **Note**

---

If an error message occurs while you are trying to execute a command file, check the contents of the file first for command line syntax errors.

---

## Lesson 1 Exercises

1. Create one command file called P5 that lists the last five commands in your command line history stack. Be sure that you have at least five commands in your command line history stack. Execute the command file P5.
2. Use the editor to create a 10-line file. Save it as FOREST.  
Create a command file called PRF that prints the first four lines of the FOREST file to your screen. Execute it.
3. Create a command file called CG that puts you into the OTHERGRP group of your account when executed. Execute it.
  - a. Check to see if you were successful.
  - b. Get back into your home group.
4. Create a command file that executes any MPE/iX command that you know. Execute it.
5. Create a command file that executes another MPE/iX command that you know. Execute it.
6. Create a command file that executes the command file created in step 4 and the command file created in step 5. Execute it.

---

## Lesson 2

### Naming Conflicts

One day you will probably create a command file and keep it with a name you like, only to discover that the computer executes something *entirely different* when you enter the name of the command file.

File names *within* a group must be unique.

But identical file names may exist in *different* groups. If your command file has the same name as an MPE/iX command, the computer might execute that other command and not the command file that you created.

---

#### Note

The system does not allow duplication of file names in the same group.

---

#### Discovering a name duplication

Create a command file that lists all of the file names in your account. It should look like this when you create it in the editor:

```
1 listfile @.0
```

Keep this file as SHOWVAR. Execute it. What happened? Did you get the following on the screen?

```
:SHOWVAR
NO MATCH FOUND FOR THIS VARIABLE SET. (CIWARN 8116)
```

This is not what you would expect. What happened?

SHOWVAR is an MPE/iX system command, and *that* is what the computer executed, instead of your command file SHOWVAR.

#### Avoiding duplication

There are two ways to avoid file name duplication.

- Use the HELP command to check for duplicate file names before naming a file.
- Use the MPE/iX system command XEQ.

#### HELP and verifying name duplications

The MPE/iX system tries to interpret everything that you type at the prompt as an instruction of some kind. This includes MPE/iX system commands, programs, utilities, subsystems, and user-created executable files, such as command files.

If you are a new user of the MPE/iX system, you certainly cannot know all of the instructions that MPE/iX will accept. But you should always try to keep your file names from duplicating the names of other commands on the system.

Before you name a command file, check with the system first to see whether the name that you want for your command file is already in use.

Use the help facility to do this.

At the MPE XL prompt, enter:

```
HELP showvar(Return)
```

Since there *is* a SHOWVAR MPE/iX system command, you get the help facility display.

```
:SHOWVAR
```

Displays specific variable names and their current values, (NEW)

```
SYNTAX
```

```
:SHOWVAR [varid][,varid] ... [,varid]
```

Enter:

```
HELP PS(Return)
```

Something like the following appears on the screen if your proposed file name matches the name of another executable file on the system.

```
:HELP PS
USER DEFINED COMMAND FILE: PS.ETHEL.MERTZ
```

```
PR
st
```

Try it.

Something like this appears on the screen if your proposed file name matches a **program file** (or subsystem/utility) on the system.

```
HELP sort(Return)
PROGRAM FILE: SORT.PUB.SYS
```

The SORT command *is* an instruction already in the system. Saving your command file as SORT would be pointless.

Try the name GLOBAL:

```
HELP global(Return)
```

```
Can't find anything under this command or in table of contents.
```

You see this message if the name that you select is unique to the system.

You could, therefore, keep a command file called GLOBAL and successfully execute it without getting surprising results, but ... there is always an exception.

---

## Note

Some kinds of executable files can be protected with various forms of security. And one of those forms of security can *hide* the file name from the help facility.

In such a case, you might be fooled into thinking that the file name you want to use is unique and available for use.

---

## The XEQ command

The MPE/iX command XEQ bypasses system commands, programs, utilities, and subsystems and executes command files—even in the case of name duplication.

Try executing your SHOWVAR command file using the XEQ command:

```
XEQ showvar(Return)
```

This time your command file called SHOWVAR should execute and display the output from its one command, LISTFILE @. @.





## Solutions

---

### Solutions to Module 1 Communicating With Your Computer

---

#### Challenge Answers

|     | Is Your Answer ... ?                                            | If Not, Start With ... |
|-----|-----------------------------------------------------------------|------------------------|
| 1.  | HELLO                                                           | Lesson 1.              |
| 2.  | BYE                                                             | Lesson 1.              |
| 3.  | Jobs and sessions (batch and interactive)                       | Lesson 2.              |
| 4.  | SHOWTIME                                                        | Lesson 3.              |
| 5.  | SHOWME                                                          | Lesson 3.              |
| 6.  | SHOWJOB                                                         | Lesson 3.              |
| 7.  | REDO                                                            | Lesson 4.              |
| 8.  | A list of the last 20 commands entered in a user session.       | Lesson 5.              |
| 9.  | TELL or TELLOP                                                  | Lesson 6.              |
| 10. | <i>MPE/iX Commands Reference Manual</i> ; online help facility. | Lesson 7.              |
| 11. | HELP                                                            | Lesson 8.              |

#### Lesson 1: Logging On and Off

1. Log on
2. BYE

**Lesson 2: Sessions  
and Jobs—What Are  
They?**

1. Jobs and Sessions or Batch and Interactive
2. a. S  
b. J  
c. S  
d. J

**Lesson 3: Entering  
Commands**

1. SHOWTIME
2. SHOWME
3. a. ETHEL.MERTZ  
b. 8:09 A.M.  
c. 103  
d. 10:48 A.M.  
e. 159  
f. Saturday, June 11, 1988
4. SHOWJOB

**Lesson 4: Mistakes  
and Corrections**

1. a, c
2. a. SHOWMW  
rE  
b. SOWTIME  
iH  
c. SHOWTIME  
dd  
d. SHJOB  
iOW

**Lesson 5: The History  
Stack**

1. A b.
2. LISTREDO
3. Answers will vary. Here is an example:  
showme  
  
showtime  
  
listredo  
  
sowtime  
  
5) showme  
6) showtime  
7) listredo  
8) sowtime

If you misspell a command, the misspelled command is added to the history stack as is—in its misspelled state, for example—command line 8.

4. a, b, c
5. Commands 26-45 are displayed.
6. LISTREDO  
REDO  
REDO 2 (or any other line number)  
DO  
DO 3 (or any other line number)

### **Lesson 6: Communicating with Other Users**

1. SHOWJOB
2. TELL #S12 (or whatever Session Number your partner is) This is a test. Please send JoAnn a reply message.
3. A similar message appeared on your screen:  
S12 FRED.DERF IS NOT ACCEPTING MESSAGES.
4. SET;MSG=OFF  
SHOWJOB  
SET;MSG=ON  
SHOWJOB
5. False. The system manager or operator can override your Quiet mode, if they choose to do so.

### **Lesson 7: More Information**

1. The maximum number of users allowed on the system has been reached.
2. The *MPE/iX Commands Reference Manual* (32650-60003).

### **Lesson 8: Online HELP**

1. HELP
2. a. The SHOWME command has no parameters. (User types SHOWME PARM at the help facility prompt.)  
b. Yes, SHOWJOB can be executed in both jobs and sessions. (User types SHOWJOB OPERATION at the help facility prompt.)  
c. MON, JULY 24, 1987, 8:47AM. (User types SHOWTIME EXAMPLE at the help facility prompt.)

---

## Solutions to Module 2 Computer Overview

---

### Challenge Answers

|    | Is your Answer ... ?                                                                                                        | If Not, Start With ... |
|----|-----------------------------------------------------------------------------------------------------------------------------|------------------------|
| 1. | Hardware is the physical equipment of a computer system.                                                                    | Lesson 1.              |
| 2. | Peripherals are hardware connected to and controlled by the computer.                                                       | Lesson 1.              |
| 3. | Programs that instruct the hardware to do something are known as software.                                                  | Lesson 2.              |
| 4. | MPE/iX is the operating system on the 900 Series HP 3000.                                                                   | Lesson 2.              |
| 5. | MPE means multiprogramming executive.                                                                                       | Lesson 2.              |
| 6. | Electronic mail, word processing, games, and data processing can all run on the HP 3000.                                    | Lesson 2.              |
| 7. | False.                                                                                                                      | Lesson 3.              |
| 8. | Local mode is when your terminal is completely on its own. Remote mode is when the terminal talks directly to the computer. | Lesson 4.              |
| 9. | Here are three ways of troubleshooting: hard reset; soft reset; turn your terminal off and back on again.                   | Lesson 4.              |

#### Lesson 1: Computer Hardware

1. System processing unit (SPU).
2. Disk drive; tape drive; printer; terminals.
3. System console.

#### Lesson 2: Software

1. Any of the following are typical tasks: electronic mail; document processing; data processing; games; form or menu creation.
2. MPE stands for multi programming executive. MPE/iX is the operating system for the 900 Series HP 3000 computers. MPE/iX manages all system resources and coordinates the execution of all programs running on the system.
3. CI stands for the command interpreter.

**Lesson 3: The Keyboard**

1. *Information entered* is a set of characters—nothing is sent to the computer until **Return** or **Enter** is pressed.
2. Although the characters may still remain visible on the terminal screen, the **Back Space** key (←) “erases” characters from terminal memory as it spaces backwards over them.

**Lesson 4: Any Problems?**

1. Numeric Key Pad - d  
 Cursor Control Keys - c  
 Function Keys - a  
**Break/Reset** - b
2. Your table should have these check marks:

|                                                                     | <b>Hard Reset</b> | <b>Soft Reset</b> |
|---------------------------------------------------------------------|-------------------|-------------------|
| 1. Can be used to Unlock your keyboard.                             | X                 | X                 |
| 2. Destroys information on your screen.                             | X                 |                   |
| 3. Does <i>not</i> destroy information on your screen.              |                   | X                 |
| 4. The “gentle” way of regaining control of a hung terminal.        |                   | X                 |
| 5. After this has been performed, the user still remains logged on. | X                 | X                 |
| 6. Should <i>not</i> be used as a means of logging off.             | X                 | X                 |

3. Remote mode
4. Local mode

---

## Solutions to Module 3 Creating Text Files

---

### Challenge Answers

|    | Did You Answer ... ?                                                                         | If Not, Start With ... |
|----|----------------------------------------------------------------------------------------------|------------------------|
| 1. | EDITOR.                                                                                      | Lesson 1.              |
| 2. | Starts with an alpha character; limited to 8 alphanumeric characters; no special characters. | Lesson 2.              |
| 3. | ADD, TEXT, LIST, MODIFY, DELETE, GATHER.                                                     | Lesson 3.              |
| 4. | <input type="checkbox"/> Break.                                                              | Lesson 4.              |
| 5. | RESUME                                                                                       | Lesson 4.              |
| 6. | ABORT                                                                                        | Lesson 4.              |
| 7. | Created while working on a text file, "K" files are safety files.                            | Lesson 4.              |

#### Lesson 1: Using EDIT/3000

1. EDIT/3000
2. EDITOR
3. END, or E
4. a. /
5. ADD or A
6. //
7. a. Answer YES to the PURGE OLD? question.  
b. Answer NO to the PURGE OLD? question.
8. LISTFILE

#### Lesson 2: File Names

1. a. Acceptable  
b. Unacceptable, too many characters used—only eight are allowed.  
c. Unacceptable, first character must be a letter.  
d. Acceptable  
e. Unacceptable, no special symbols or characters are allowed.  
f. Unacceptable, too many characters. Also, underbar character is not allowed.  
g. Acceptable

### Lesson 3: Editing a Text File

1. Get into the editor. Add 20 lines of text. Exit ADD with //. Keep FILEA.
2. LIST ALL
3. LIST 5/15
4. LIST ALL, OFFLINE
5. LIST 15/20, OFFLINE
6. DELETE 1/3 or D 1/3
7. You can perform a DELETE ALL to delete your current file, which contains your most recent revisions. You will lose *all* of your changes. You can then retrieve the *original* FILEA using T FILEA, which allows you to start editing the FILEA again.
8. GATHER 3 to 6.1
9. GATHER 16/20 to .1
10. GATHER ALL
11. a. Mary will be coming ovr for dinner.  
rGeno
- b. Geno will be coming ovr for dinner.  
ie
- c. Geno will be coming over for dinner.  
DDDDDD

#### Note

---

Notice that when you deleted the wording “coming,” you also had to delete the space after it.

---

12. The “old” version of MYFILE is retrieved from disk. You will lose all work done on this file during this work session.

### Lesson 4: More About the Editor

1. Assume that FILE1 has been put into the editor workspace.
  - a. JOIN FILE2
  - b. JOIN FILE2 to 1.2
2. :LISTFILE
3. **Break** key
4. RESUME
5. ABORT
6. a. December 31, 12:30 P.M. (unless it is a leap year, in which case it would be December 30, 12:30 P.M.)
- b. January 2, 8:15 A.M.
- c. February 1 11:52 A.M.
- d. January 15, 11:00 P.M.

---

## Solutions to Module 4 The Account Structure

---

### Challenge Answers

|    | Is Your Answer ... ?             | If Not, Start With ... |
|----|----------------------------------|------------------------|
| 1. | User; account; group; files      | Lesson 1               |
| 2. | Session, user, account and group | Lesson 1               |
| 3. | ASTA,NICK.CHARLES,NORA           | Lesson 1               |
| 4. | PUB                              | Lesson 1               |
| 5. | False                            | Lesson 2               |
| 6. | Your home group                  | Lesson 3               |
| 7. | CHGROUP                          | Lesson 4               |
| 8. | PASSWORD                         | Lesson 5               |

### Lesson 1: The MPE/iX Account Structure

1. Security; billing; organization
2. Users; accounts; groups; files
3.
  - a.
 

```
HELLO norma.desmond,sunset
 ^ ^ ^
 user.account,group
```
  - b.
 

```
hello bob,carol.ted,alice
 ^ ^ ^ ^
 session,user.account,group
```
  - c.
 

```
HELLO MGR.MKTG
 ^ ^
 user.account
```
  - d.
 

```
hello emily.kado,pres
 ^ ^ ^
 user.account,group
```
  - e.
 

```
HELLO BLDG214,john.doe,audit
 ^ ^ ^
 session,user.account,group
```
4. Your home group.
5. HELLO JAN.SALES
6. HELLO FOLDER,CLERK.DESK

7. HELLO USER1.PROG,TEST
8. hello bill.payroll,exempt
9. The PUB group in all accounts is a public domain group. Any user assigned to the same account would have access to the files in this group unless the files were lockword protected.

### **Lesson 2: Security and the Account Structure**

1. You do not have the user capability (in this case, AM, account manager) to use this MPE/iX command.
2. Three times for each password.

### **Lesson 3: File and Group Security**

1.
  - a. False. All users of the account with the basic set of user capabilities, can read (look at) a file that is in the PUB group.
  - b. True. Users with the basic set of user capabilities can access (read, change, save) any file in their home group.
  - c. False. Only those users with either special capabilities or whose home group is PUB, can alter a file from the PUB group and save it back to the PUB group.
  - d. True. You can always (read, change, save) any file in your home group, no matter what group you log on to.

### **Lesson 4: Changing Groups**

1. You can log on to another group with HELLO, or use the CHGROUP command, to change groups in an account.
2. Sample answer:
 

```
CHGROUP newgroup
SHOWME
CHGROUP (This gets you back to your home group.)
```
3. Sample answer:
 

```
hello user.acct,newgroup
SHOWME
hello user.acct (This gets you back to your home group.)
```

### **Lesson 5: Changing User Passwords**

1. Sample answer:
 

```
PASSWORD
ENTER OLD USER PASSWORD: user7(RETURN)
ENTER NEW USER PASSWORD: edward(RETURN)
ENTER NEW USER PASSWORD AGAIN: edward(Return)
PASSWORD WAS CHANGED SUCCESSFULLY.
:_
```

---

## Solutions to Module 5 Working With Files

---

### Challenge Answers

|     | Is Your Answer ... ?                                                                                                 | If Not,<br>Start<br>With ... |
|-----|----------------------------------------------------------------------------------------------------------------------|------------------------------|
| 1.  | File name plus group name.                                                                                           | Lesson 1                     |
| 2.  | File name plus group and account names.                                                                              | Lesson 1                     |
| 3.  | LISTFILE A@                                                                                                          | Lesson 1                     |
| 4.  | RENAME                                                                                                               | Lesson 2                     |
| 5.  | Files take up valuable disk space on the computer; therefore, seldom-used files should be stored on a magnetic tape. | Lesson 3                     |
| 6.  | PURGE                                                                                                                | Lesson 4                     |
| 7.  | PRINT                                                                                                                | Lesson 5                     |
| 8.  | COPY                                                                                                                 | Lesson 6                     |
| 9.  | RELEASE; SECURE                                                                                                      | Lesson 6                     |
| 10. | Your system manager.                                                                                                 | Lesson 6                     |

#### Lesson 1: Displaying Qualified File Name

1. a. FILE1  
b. file1.payroll.acct  
c. FILE1.PUB  
d. file1.pub.sys
2. LISTFILE
3. LISTFILE A@
4. LISTFILE @S
5. LISTFILE @.@,6
6. LISTFILE s@r@t.pub.sys

## Lesson 2: Renaming Files

1. Get into the editor. Keep one file as OAK and the other as MAPLE.
2. You get the following error message;

```
RENAME FAILED DUE TO FILE SYSTEM ERROR, NOT RENAMED (CIERR 373)
```

3. RENAME oak,acorn

```
LISTFILE acorn or LISTFILE
```

## Lesson 3: Storing Seldom-Used Files

1. Files take up disk space on the system. If your file needs to be archived and it is a file that is relatively inactive, you should have the file stored to a magnetic tape as a good housekeeping practice.

## Lesson 4: Deleting Files

1. PURGE maple  
LISTFILE maple
2. b

## Lesson 5: Displaying File Contents

1. PRINT acorn
2. PRINT ACORN;START=3;END=7
3. You could get into the editor and text in the ACORN file then perform a LIST ALL, OFFLINE.

---

### Note

Advanced Users: the PRINT command has parameters that allow you to direct your output to a printer. This is covered in the *Advanced Skills* course.

---

## Lesson 6: Copying Files

1. a. COPY FROM=ACORN;TO=PINE

Get into the editor and change the file PINE so that it now looks different. Keep PINE.

- b. COPY PINE,ACORN  
PURGE OLD? no  
NO COPY WAS DONE (CIERR 9113)
- c. COPY PINE,ACORN  
PURGE OLD? yes  
PRINT ACORN

The PINE file writes over the “original” ACORN file. Printing the contents of the “new” ACORN file reveals the contents of the “old” PINE file.

2. RELEASE
3. SECURE
4. d
5. The file is copied to your current group.
6. Your system manager.

---

## Solutions to Module 6 Command Files

---

### Challenge Answers

|    | Is Your Answer ... ?                                                                                                     | If Not,<br>Start<br>With ... |
|----|--------------------------------------------------------------------------------------------------------------------------|------------------------------|
| 1. | A text file that can contain MPE/iX commands, program file names, or names of other command files, that can be executed. | Lesson 1                     |
| 2. | XEQ                                                                                                                      | Lesson 2                     |

#### Lesson 1: Using Command Files

- The command file P5:  
LISTREDO -5  
P5
- The file called PRF should consist of one line.  
PRINT FOREST;start=1;end=4  
or  
PRINT FOREST;END=4  
Type the following to execute it:  
PRF
- The command file called CG should consist of this line.  
CHGROUP OTHERGRP  
Type the following to execute:  
CG
  - To check to see if the change group took place, enter:  
SHOWME
  - To get back into your home group, enter:  
CHGROUP

---

#### Note

The answers to exercises 4-6 may vary.

---

- Example:  
A command file called SH:  
SHOWTIME  
Type the following to execute it:

SH

5. Example:

A command file called SM:

SHOWME

Type the following to execute it:

SM

6. Example:

A file called SHOW:

SH

SM

To execute it enter:

SHOW

## **Lesson 2: Naming Conflicts**

1. RUNIT

2. Use the MPE/iX command, XEQ plus the command file name.

XEQ runit



# Glossary

---

This glossary defines terms found in the *Fundamental Skills* and *Advanced Skills* courses. Terms that appear only in the *Advanced Skills* course are identified by the notation (AS) following the definition.

|                                     |                                                                                                                                                                                                                                                                                     |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>abort</b>                        | A procedure that terminates a program or session if an irrecoverable error, mistake, or malfunction occurs, or if the system manager requests termination.                                                                                                                          |
| <b>access codes</b>                 | Access codes are assigned by the system manager to accounts. Access codes are assigned by the account manager to groups and users. Access codes regulate who has the ability to read, write, append, lock, save, or execute a given file.                                           |
| <b>account</b>                      | An account is a collection of users and groups. Each account on the system has a unique name. A user must specify a valid account to access the system.                                                                                                                             |
| <b>account librarian capability</b> | AL capability: Assigned by the account manager to one or more users within an account. An account librarian is allowed special file access modes to maintain specified files within the account.                                                                                    |
| <b>account manager capability</b>   | AM capability: Assigned by the system manager to one user within each account who is then responsible for establishing users and groups.                                                                                                                                            |
| <b>account structure</b>            | The account structure provides organization, security, and billing for the system. It is used to allocate use of system resources such as central processor time, online connect time, and file space. The account is the principal billing entity for the use of system resources. |
| <b>append</b>                       | To join all or part of one existing file to the end of another existing file.                                                                                                                                                                                                       |
| <b>application program</b>          | An application program is a set of computer instructions that executes a specific set of tasks. Applications include spreadsheets, word processing,                                                                                                                                 |

graphics, database management, and data communication programs.

**ASCII**

American Standard Code for Information Interchange: ASCII is the standard method of representing character data (seven data bits plus one that can be used for parity). This method was established by the American National Standards Institute (ANSI) to achieve compatibility between data devices during an exchange of information. (AS)

**attribute**

Attributes enable the computer to determine what functions it will or will not allow a user, group, or account to perform. They include file access codes and special capabilities. They enable the computer to determine what functions it will or will not allow a user, group, or account to perform.

**backreference**

A technique of referencing a previously defined file. Backreferencing uses an asterisk (\*) before a formal file designator to indicate it that has been previously defined with the FILE command. (AS)

**backup**

A process that duplicates computer data to offline media such as magnetic tape. Backups protect data if a system problem occurs.

**batch processing**

A method of submitting a job for processing. A job, which is submitted as a single entity, can consist of multiple commands such as program compilation and execution, file manipulation, or utility functions. Once submitted, no further interaction between the user and the job is necessary.

**binary**

A method of representing numbers, alphabetic characters, and symbols in digital computers. Binary is a base two number system that uses only two digits, 0's and 1's, to express numeric quantities. (AS)

**Boolean**

A data type with a value that is either TRUE or FALSE (binary 1 or 0). (AS)

**byte**

A combination of eight consecutive bits treated as a unit. A byte represents one letter or number. The size of memory and disk storage is measured in bytes. (AS)

**capability**

A method for determining what commands account members are allowed to execute.

|                                           |                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                           | Capabilities are assigned to accounts, groups, and users to provide system security and access to the operating system. Account capabilities are assigned by the system manager when the account is created. The account manager then assigns capabilities to groups and users within the account. |
| <b>carriage-control characters (CCTL)</b> | Carriage control characters determine such things as double spacing, vertical line spacing, and page ejects.                                                                                                                                                                                       |
| <b>catalog</b>                            | A system file containing information such as a listing of user-defined commands. A UDC file is cataloged with the SETCATALOG command, and the information is added to the system file COMMAND.PUB.SYS. (AS)                                                                                        |
| <b>central processing unit</b>            | CPU: A part of a computer system that interprets and executes instructions and contains all or part of internal storage.                                                                                                                                                                           |
| <b>CI</b>                                 | See command interpreter.                                                                                                                                                                                                                                                                           |
| <b>command</b>                            | A system-reserved word that directs the operating system, subsystem, or utility program to perform a specific operation.                                                                                                                                                                           |
| <b>command file</b>                       | A file a user creates to execute multiple MPE commands (except D0 and RED0). To execute commands or UDCs referenced with the command file, enter the command file name at the system prompt.                                                                                                       |
| <b>command interpreter</b>                | CI: An MPE/iX program that reads command lines entered at the standard input device; interprets them; determines if they are valid, and if so, executes them.                                                                                                                                      |
| <b>command line history stack</b>         | See <b>history stack</b> .                                                                                                                                                                                                                                                                         |
| <b>communication</b>                      | The ability of one computer system to access or "talk" to other computer systems by way of telecommunication devices.                                                                                                                                                                              |
| <b>compile</b>                            | The process of changing a program written in a source language (for example, BASIC, C, FORTRAN) into a machine language routine that the computer can understand. The compiled routine is then ready to be loaded into storage and run.                                                            |
| <b>computer</b>                           | A device that accepts information, processes it, and produces an output. A computer usually contains memory, a control unit, arithmetic and logical manipulators, and a means for input and output.                                                                                                |

|                               |                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>configuration</b>          | The way in which a computer and peripheral devices are programmed to interact with each other.                                                                                                                                                                                                                                                                              |
| <b>console</b>                | A terminal, usually designated logical device 20, given unique status by the operating system. The operator uses the console to monitor and manage jobs, sessions and resources, respond to requests, and communicate with other user terminals. It is used to boot the system and receive system loader error messages, system error messages, and system status messages. |
| <b>continuation character</b> | An ampersand (&) character entered as the last character of a command line. A continuation character tells the command interpreter that the command is longer than one line and is continuing onto a second or subsequent lines. (AS)                                                                                                                                       |
| <b>cursor</b>                 | A character, such as a flashing rectangle or blinking underline character, on a terminal screen. The cursor marks the position where text or data can be entered, modified, or deleted.                                                                                                                                                                                     |
| <b>database</b>               | A collection of logically related data files, and structural information about the data.                                                                                                                                                                                                                                                                                    |
| <b>default</b>                | A predefined value or condition that is assumed, and used, by the operating system if no other value or condition is specified.                                                                                                                                                                                                                                             |
| <b>delimiter</b>              | A special character used to mark the end of a string of characters. Common delimiters are the semicolon (;), equal sign (=), <u>Return</u> , or the comma (,).                                                                                                                                                                                                              |
| <b>dereferencing</b>          | Dereferencing substitutes the value of a variable in place of the variable name. See also <b>explicit dereferencing</b> and <b>implicit dereferencing</b> . (AS)                                                                                                                                                                                                            |
| <b>device</b>                 | See <b>peripheral</b> .                                                                                                                                                                                                                                                                                                                                                     |
| <b>device file</b>            | A file associated with a nonshareable device (a spoolfile). Input and output spoolfiles are identified by a number in the DFID (device file identification) column of the SHOWIN and SHOWOUT command displays. A device file may also refer to any nondisk device, such as \$STDIN and \$STDLIST, the default input and output device files for a terminal. (AS)            |
| <b>directory</b>              | A system table defining where groups, users, accounts, and files are located. A                                                                                                                                                                                                                                                                                             |

|                               |                                                                                                                                                                                                                                                          |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                               | directory may also contain information such as file size, creation date, modification dates, creator, or security information. (AS)                                                                                                                      |
| <b>disk drive</b>             | A peripheral device that reads information from and writes information to the disk.                                                                                                                                                                      |
| <b>echo</b>                   | To display on the terminal screen data being typed on the keyboard. If echo is turned off, the computer receives the data but nothing appears on the screen. (AS)                                                                                        |
| <b>editor</b>                 | A word processing application used to prepare, modify, or delete text and program files. EDIT/3000 is the text editor used with MPE/iX.                                                                                                                  |
| <b>environment file</b>       | A compiled disk file containing all the specifications for a printed page of data. These specifications, which are not a part of the data, may include the page size, character fonts, and forms to be used in conjunction with the laser printer. (AS)  |
| <b>error listing</b>          | A report generated by the system describing the step-by-step processing of the job. (AS)                                                                                                                                                                 |
| <b>error message</b>          | A system message describing errors that occurred during either an interactive session or batch job. The messages appear on the standard list device, which is usually a terminal for a session, or a line printer for a job.                             |
| <b>execute</b>                | When a command is entered, the computer carries out the instructions or performs the routine indicated.                                                                                                                                                  |
| <b>explicit dereferencing</b> | When the command interpreter encounters an exclamation point immediately before a variable name, it substitutes the value for the variable name. Explicit dereferencing may be used in any MPE/iX command. See also <b>implicit dereferencing</b> . (AS) |
| <b>expression</b>             | A statement consisting of variables, constants, and operators. (AS)                                                                                                                                                                                      |
| <b>FCOPY</b>                  | An HP 3000 subsystem that allows the user to copy, append, translate data from one type to another (for example, ASCII to EBCDIC), verify, and compare files. The subsystem is activated with the MPE/iX FCOPY command. (AS)                             |
| <b>file</b>                   | A group of related records that represent either ASCII text (text files) or binary data (such as executable code).                                                                                                                                       |

|                                   |                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>file equation</b>              | A method of equating a file name to a device or another file. The MPE/iX FILE command is used to establish the relationship of the file to the device. Generally used to direct the input or output of a program, job, or session to a particular device by referencing the device class, such as TAPE or LP. (AS)                                                                                       |
| <b>file name</b>                  | An MPE/iX file name is a string of up to eight alphanumeric characters, the first of which must be an alphabetic character. The file name is assigned when the file is created or first saved.                                                                                                                                                                                                           |
| <b>formal file designator</b>     | An "alias" file name that is used either programmatically or in a file equation to reference a file. The formal file designator is not the file name found in the system file directory. (AS)                                                                                                                                                                                                            |
| <b>fully qualified file name</b>  | A complete file description that includes the file name, the group to which the file belongs, and the account to which the group belongs. The fully qualified file name of the LETTER file in the PUB group of the SYS account is expressed as LETTER.PUB.SYS.                                                                                                                                           |
| <b>function keys</b>              | Special keys on the terminal keyboard that are labeled sequentially, F1, F2, F3 . . . and correspond to the windows that appear at the bottom of the terminal screen. Function keys are used to perform various activities.                                                                                                                                                                              |
| <b>group</b>                      | A group is part of an account and is used to organize the account's files. All files must be assigned to a group; and within an account, each group has a unique name. Groups are the smallest entity for which use of system resources is reported. A PUB group is designated for each account when it is created. Additional groups are created within the account, as needed, by the account manager. |
| <b>group librarian capability</b> | GL capability: Assigned by the account manager, to a user within an account. A group librarian is allowed special file access modes for the maintenance of certain files within the user's home group.                                                                                                                                                                                                   |
| <b>hard reset</b>                 | A method to reset the computer or a terminal. A hard reset erases all information in memory. See also <b>soft reset</b> .                                                                                                                                                                                                                                                                                |

|                               |                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>hardware</b>               | All the physical components of the computer including the CPU cabinet, tape drives, disk drives, terminals, and other peripherals.                                                                                                                                                                                                                                      |
| <b>header</b>                 | The first page printed when output is directed to a line printer. It contains the session name (if designated), session number, logon identification, day of the week, date, and time. It corresponds to the trailer printed as the last page of the output.                                                                                                            |
| <b>Help facility</b>          | An online utility providing information on all MPE/iX commands. Information can be accessed by topic areas or by task.                                                                                                                                                                                                                                                  |
| <b>hexadecimal</b>            | A method of representing a single character with a combination of four bits. Hexadecimal also describes the base 16 number system, in which the first ten digits are 0 through 9, and the last six are A through F. A number written in base 16 is preceded by a dollar sign (\$). For example, \$F3 is the hexadecimal representation for the decimal number 243. (AS) |
| <b>history stack</b>          | The history stack is a CI table that contains, by default, the 20 most recent commands entered at the system prompt during a session. The history stack is used with the RED0 and D0 commands. To display the commands in the history stack use the LISTRED0 command.                                                                                                   |
| <b>home group</b>             | A home group may be assigned to each user. If no other group is specified with the HELLO or JOB command, users are logged on to their home group by default. If no home group is assigned, the user must always specify a group when logging on. The account manager assigns the home group when a user's name is first defined.                                        |
| <b>HP Desk</b>                | HP Desk is Hewlett-Packard's electronic mail product.                                                                                                                                                                                                                                                                                                                   |
| <b>IF-THEN-ELSE statement</b> | A statement used to determine what action occurs. When the IF condition is true, the THEN action is performed. When the IF condition is false the ELSE action is performed. (AS)                                                                                                                                                                                        |
| <b>implicit dereferencing</b> | A way of substituting the value of a variable in place of the variable name. Implicit dereferencing may be used with the CALC, IF, SETVAR, and                                                                                                                                                                                                                          |

|                       |                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | WHILE commands. See also <b>explicit dereferencing</b> . (AS)                                                                                                                                                                                                                                                                                                            |
| <b>input priority</b> | A number in the range of 1 (lowest priority) to 14 (highest priority) assigned to input jobs. The input priority can be assigned by the system (default is 8) or by the user. Jobs with an input priority less than or equal to the system jobfence (default 7) are deferred. (AS)                                                                                       |
| <b>integer</b>        | A data type that is either a positive or negative whole number, or zero. (AS)                                                                                                                                                                                                                                                                                            |
| <b>integer value</b>  | A sequence of digits preceded by a +, -, \$, or % sign. When neither a + nor - sign is provided, a positive number is assumed. A \$ indicates a hexadecimal integer, and a % indicates an octal integer. (AS)                                                                                                                                                            |
| <b>interactive</b>    | An interactive session allows users to enter commands and data at the terminal and receive an immediate response. Sessions are useful for data entry and retrieval, text editing, or program development when direct dialog with the computer is preferred.                                                                                                              |
| <b>job</b>            | A job is a method of submitting multiple operating system and utility commands for processing with a single command. Once submitted, the job executes independently of the user's session. Jobs are used to compile source programs, modify files, or perform other functions that do not require user interaction. See also <b>batch processing</b> and <b>stream</b> . |
| <b>jobfence</b>       | A limit established to manage jobs. If a job has an input priority higher than the jobfence it executes. If it has an input priority less than or equal to the jobfence, it does not execute. (AS)                                                                                                                                                                       |
| <b>job file</b>       | A job file is used to define a job to the system. It must start with a JOB command and end with an E0J command. (AS)                                                                                                                                                                                                                                                     |
| <b>job limit</b>      | A limit set to manage jobs. The system manager or operator can restrict system usage by limiting the number of jobs allowed to run on the system. If the LIMIT command is used to set the job limit to 0 (zero), no additional jobs can log on to the system.                                                                                                            |
| <b>job listing</b>    | See <b>listing</b> .                                                                                                                                                                                                                                                                                                                                                     |

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>job number</b>    | A system assigned identification number given to each job when it is submitted for processing.                                                                                                                                                                                                                                                                                                                                                           |
| <b>keyboard</b>      | A keyboard is attached to a terminal and used to input data to communicate with the system.                                                                                                                                                                                                                                                                                                                                                              |
| <b>keyword</b>       | A word assigned a specific meaning by the operating system, a subsystem, computer language, or utility.                                                                                                                                                                                                                                                                                                                                                  |
| <b>K file</b>        | A recovery file created by EDIT/3000, with a name in the form K#####, where the first three characters (ddd) show the Julian day, and the next four (hhmm) show the time in hours and minutes when work began on the files. A new K file is created every time that a new editor file is created or an existing file is loaded for editing. If a system problem occurs, the data in the new or loaded file is saved to the K file for recovery purposes. |
| <b>laser printer</b> | A hardware device used for system output. A laser printer prints output one page at a time.                                                                                                                                                                                                                                                                                                                                                              |
| <b>LDEV number</b>   | See <b>logical device number</b> .                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>line editor</b>   | A line editor requires you to press <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">Return</span> to end one line of text and to begin another. EDIT/3000 is an example of a line editor.                                                                                                                                                                                                                                       |
| <b>line printer</b>  | A hardware device used for system output. A line printer prints output one line at a time.                                                                                                                                                                                                                                                                                                                                                               |
| <b>link</b>          | To merge a compiled file and its libraries to create an executable file that allows a program to run.                                                                                                                                                                                                                                                                                                                                                    |
| <b>listing</b>       | A listing is the output of a job usually in the form of a printed document.                                                                                                                                                                                                                                                                                                                                                                              |
| <b>local mode</b>    | A standalone method of terminal operation. A terminal is operating in local mode when it is not connected to the computer. See also <b>remote mode</b> .                                                                                                                                                                                                                                                                                                 |
| <b>lockword</b>      | A word used as a security device on files. A lockword can be assigned to a file when it is created or renamed, and must be supplied to regain access to the file. The word may be from one to eight alphanumeric characters long and must begin with an alphabetic character. (AS)                                                                                                                                                                       |

|                                       |                                                                                                                                                                                                                                                         |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>logical device number</b>          | An LDEV number is assigned to all peripherals of a computer system and is used for identification purposes.                                                                                                                                             |
| <b>log off</b>                        | A method of terminating a session. To log off MPE/iX, enter the BYE or EXIT command.                                                                                                                                                                    |
| <b>log on</b>                         | A method of initiating a session. To log on to MPE/iX, enter the HELLO command and a valid user and account name, plus a group name if necessary, and any required passwords.                                                                           |
| <b>log on identity</b>                | A security device used to verify users to the system. A log on identity includes a valid user name and account name in the form <i>user:account</i> .                                                                                                   |
| <b>log on prompt</b>                  | A system prompt (MPE XL) that indicates that the computer is ready to initiate a session. See also <b>prompt</b> .                                                                                                                                      |
| <b>magnetic tape</b>                  | A data storage device used to duplicate online data to offline media. The duplicated data may also be copied from the tape back to disk. MPE/iX supports the use of magnetic tape in reel form.                                                         |
| <b>MPE/iX</b>                         | Multiprogramming Executive with Integrated POSIX: The operating system for the 900 Series HP 3000 computers. MPE/iX manages all system resources and coordinates the execution of all programs running on the system.                                   |
| <b>MPE XL</b>                         | Multiprogramming Executive with Extended Large Addressing: the operating system that preceded MPE/iX. All programs written for MPE XL will run without change under MPE/iX.                                                                             |
| <b>\$NEWPASS</b>                      | A system-defined file. \$NEWPASS is used as temporary storage for processes such as compilation. Only one \$NEWPASS file may exist during a single job or session. \$NEWPASS is the name temporarily assigned to any file referenced by \$OLDPASS. (AS) |
| <b>nonshareable device capability</b> | ND capability: Assigned to accounts and users allowing account members to own nonshareable devices such as unspooled tape drives, line printers, serial disks, and volumes.                                                                             |
| <b>\$NULL</b>                         | A system-defined file. \$NULL is a nonexistent file that is treated as an empty file. When referenced as an input file by a program, that program receives only an end-of-file indication upon first access.                                            |

|                           |                                                                                                                                                                                                                                                          |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           | When referenced as an output file, the associated write request is accepted by MPE/iX but no physical output is actually performed. \$NULL can be used to discard unneeded output from an executing program. (AS)                                        |
| <b>octal</b>              | Octal is a base eight number system in which digits 0 through 7 are used. One octal digit can be represented by three binary digits. Octal numbers are preceded by a percent sign (%); for example, %775 is the same decimal 509. (AS)                   |
| <b>\$OLDPASS</b>          | A system-defined file. \$OLDPASS is the name of the temporary file last closed as \$NEWPASS. (AS)                                                                                                                                                        |
| <b>online</b>             | Data stored in memory that is updated as soon as it changes and is, therefore, constantly current and accessible.                                                                                                                                        |
| <b>operating system</b>   | A software program that enables the computer to run. It contains programs such as basic file and I/O manipulators. All subsystems run upon the operating system.                                                                                         |
| <b>optional parameter</b> | A parameter that is not required when entering a command. In MPE/iX reference manuals, optional parameters appear within brackets ([ ]). (AS)                                                                                                            |
| <b>outclass priority</b>  | A value in the range of 1 to 13 used to determine if a job's error listing will print. If the outclass priority is higher than the system outfence value, the error listing will print.                                                                  |
| <b>outfence</b>           | A number in the range of 1 (lowest priority) to 14 (highest priority) used to control access to the system printer. If a job does not have an output priority higher than the system outfence (default 7), it will not print. (AS)                       |
| <b>output priority</b>    | A number in the range of 1 (lowest priority) to 13 (highest priority) assigned to an output spoolfile either by the system (a default value) or by a user. The output priority is used by MPE/iX to determine the order in which files are printed. (AS) |
| <b>overwrite</b>          | A method of erasing and replacing an existing file. If a file is saved under a name that already exists on disk, the new file overwrites the existing file.                                                                                              |
| <b>paging</b>             | A method to limit the amount of data appearing on the terminal screen to one                                                                                                                                                                             |

|                                   |                                                                                                                                                                                                                       |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                   | full screen (“page”) of information rather than having the data scroll.                                                                                                                                               |
| <b>parameter</b>                  | A value passed to a procedure, which then uses it in calculations or to determine operations within the procedure.                                                                                                    |
| <b>partly qualified file name</b> | A designation identifying the group in which a file resides. A partly qualified file name may be used to access a file in another group of the account to which you are logged on.                                    |
| <b>password</b>                   | A password is a string of predefined ASCII characters. The system uses the string as a security device to verify the identity of a user, group, or account.                                                           |
| <b>path</b>                       | See <b>search path</b> .                                                                                                                                                                                              |
| <b>peripheral</b>                 | Hardware devices that are attached to and controlled by the computer. Peripherals include terminals, disk drives, or printers.                                                                                        |
| <b>permanent file</b>             | A permanent file is stored on disk and has an entry identifying it in the system directory. To delete a file from the system use the PURGE command. (AS)                                                              |
| <b>printer</b>                    | A hardware device used for system output. There are various types of printers available for use with MPE/iX, including line printers and laser printers.                                                              |
| <b>program</b>                    | A program is a sequence of instructions that tells the computer how to perform a specific task.                                                                                                                       |
| <b>programmer</b>                 | A person who writes sets of instructions (programs) telling the computer how to perform a specific task. (AS)                                                                                                         |
| <b>prompt</b>                     | A character(s) displayed on the terminal screen indicating that the system is ready for a command. The default MPE/iX prompt is a colon (:). Other subsystems have different prompts. See also <b>log on prompt</b> . |
| <b>PUB group</b>                  | The public group of an account. Programs and files available to all account users reside here.                                                                                                                        |
| <b>PUB.SYS</b>                    | The public group of the system account. Programs and applications available to all users of the system reside here.                                                                                                   |
| <b>queue</b>                      | A job management technique. Jobs waiting in a line (queue) are usually processed on a first in, first out basis or by priority, if specified. For example, the output                                                 |

produced by a program is generally stored on disk in a queue until a printer becomes available. As each output is printed, the next job in priority is selected and processed.

|                             |                                                                                                                                                                                                                                                                                          |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>quiet mode</b>           | When a session is running in quiet mode, messages sent from other jobs or sessions are not displayed on the terminal screen. Warning messages from the system console override quiet mode and are displayed on the terminal screen.                                                      |
| <b>record</b>               | A collection of data treated as a unit, residing in a file. (AS)                                                                                                                                                                                                                         |
| <b>recursion</b>            | The ability of a procedure or function to call itself.                                                                                                                                                                                                                                   |
| <b>redo stack</b>           | See <b>history stack</b> .                                                                                                                                                                                                                                                               |
| <b>reexecute</b>            | A second or subsequent execution of a command.                                                                                                                                                                                                                                           |
| <b>remote mode</b>          | A method of terminal operation. A terminal is operating in remote mode when it is transmitting to and receiving data from a remote (or host) computer. See also <b>local mode</b> .                                                                                                      |
| <b>required parameter</b>   | A parameter that is required when entering a command. In MPE/iX reference manuals, required parameters appear within braces ( { } ). (AS)                                                                                                                                                |
| <b>RESTORE subsystem</b>    | A system utility program allowing the retrieval of user files from SYSGEN or Store tapes and writing them to disk. The RESTORE subsystem is activated with the RESTORE command. (AS)                                                                                                     |
| <b>save file capability</b> | SF capability: Assigned to users and accounts allowing users to save the files they create.                                                                                                                                                                                              |
| <b>scheduling</b>           | A method of determining when a job will be processed by the computer. Jobs are scheduled using the STREAM command.                                                                                                                                                                       |
| <b>scroll</b>               | A way to roll data up or down on the terminal screen.                                                                                                                                                                                                                                    |
| <b>search path</b>          | The hierarchy by which MPE/iX searches the system for UDCs, commands, programs, and command files. When you enter a command at the system prompt the standard MPE/iX search path is UDCs first, MPE/iX commands second, and file names (including command files and programs) last. (AS) |

|                       |                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>security</b>       | MPE/iX provisions to protect the system from unauthorized use. The most basic level of security includes organizing files into groups and users into accounts, both of which may be assigned a password. Security also refers to the ability to read, write, append, lock, save, and execute files. Assigned to accounts by the system manager and to groups and users by the account manager. |
| <b>session</b>        | An interactive way of communicating with a computer. In a session, commands are entered by way of the keyboard, and the computer responds with an action or a message to the terminal's screen. A session is initiated with the HELLO command and is terminated with the BYE command.                                                                                                          |
| <b>session limit</b>  | The maximum number of sessions allowed to log on at any given time. This is set with the LIMIT command. (AS)                                                                                                                                                                                                                                                                                   |
| <b>session name</b>   | An optional identification method for a session. A session name is specified when logging on in the form <i>session,user:account</i> .                                                                                                                                                                                                                                                         |
| <b>session number</b> | A system assigned identification number given to each new session as it is logged on to the system.                                                                                                                                                                                                                                                                                            |
| <b>softkeys</b>       | See <b>function keys</b> .                                                                                                                                                                                                                                                                                                                                                                     |
| <b>soft reset</b>     | A method of resetting a computer or terminal. A soft reset initializes various terminal functions but does not reset the memory. See also <b>hard reset</b> .                                                                                                                                                                                                                                  |
| <b>software</b>       | A set of computer instructions. Software programs are concerned with the operation of a computer and provide it with instructions on how to perform specific operations.                                                                                                                                                                                                                       |
| <b>SORT-MERGE/XL</b>  | An MPE/iX utility program. The SORT program allows you to sort data alphabetically or numerically. The MERGE program allows you to combine previously sorted data into a new file.                                                                                                                                                                                                             |
| <b>source code</b>    | A file(s) containing the instructions of a program. It must be compiled into machine-readable data (object code) and linked before it can be executed by the computer. (AS)                                                                                                                                                                                                                    |
| <b>spoolfile</b>      | A file on a storage device (usually a disk drive) that is either spooled from an input                                                                                                                                                                                                                                                                                                         |

|                        |                                                                                                                                                                                                                                                                                                                                         |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | device or spooled to an output device. Spoolfiles may be in any of the following states: either OPEN, ACTIVE, READY, CREATE, PRINT, DEFER, SPSAVE, PROBLM, DELPND, or XFER. These states describe different stages of the spooling process depending upon whether the file is an input or an output spoolfile. (AS)                     |
| <b>spooling</b>        | A method of managing jobs. Multiple users can send output to a nonshareable device, such as a tape or line printer, and their output is directed to spoolfiles on disk. The output is printed on a priority basis as the printer becomes available. Users can proceed with other processing activities without waiting for the printer. |
| <b>standalone</b>      | See <b>local mode</b> .                                                                                                                                                                                                                                                                                                                 |
| <b>\$STDIN</b>         | A system-defined file name. \$STDIN refers to the standard input device used to initiate a session or job; usually a terminal or tape drive. (AS)                                                                                                                                                                                       |
| <b>\$STDLIST</b>       | A system-defined file name. \$STDLIST indicates the standard job or session listing file corresponding to the particular input device being used. The listing device is usually a printer for batch jobs and a terminal for sessions. (AS)                                                                                              |
| <b>STORE subsystem</b> | An HP 3000 subsystem used to save files to magnetic tape. The STORE subsystem is executed by using the STORE command. (AS)                                                                                                                                                                                                              |
| <b>stream</b>          | A method of running a batch job. A batch job is initiated from a session or a job by using the STREAM command. Once a job is streamed, it executes as a separate process without any further user input or supervision.                                                                                                                 |
| <b>string value</b>    | A sequence of characters surrounded by quotation marks. (AS)                                                                                                                                                                                                                                                                            |
| <b>subcommand</b>      | A command performed under another command. For example, the EDIT/3000 MODIFY command enables you to use the D (delete), I (insert), and R (replace) subcommands.                                                                                                                                                                        |
| <b>subsystem</b>       | A system supported utility or program. MPE/iX subsystems include FCOPY, SORT-MERGE/XL, and EDIT/3000.                                                                                                                                                                                                                                   |
| <b>syntax</b>          | A set of rules defining the structure of a language, instruction, or command.                                                                                                                                                                                                                                                           |

|                                  |                                                                                                                                                                                                                            |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SYS account</b>               | A special account on the 900 Series HP 3000 that is included with the system when it is first installed. It contains all of the files for system-supported subsystems, utility programs, and compilers.                    |
| <b>system console</b>            | See <b>console</b> .                                                                                                                                                                                                       |
| <b>system-defined files</b>      | Files defined by MPE/iX and made available to all users. They indicate standard input or output devices, special temporary files, and files opened for output that do not perform an actual write operation. (AS)          |
| <b>system-defined variables</b>  | Command interpreter variables are used to store system-assigned information. MPE/iX system-defined variables usually begin with the letters "HP". They have a string, integer, or Boolean value. (AS)                      |
| <b>system manager capability</b> | SM capability: A user-assigned system manager capability is responsible for installing the computer, creating accounts, and assigning capabilities and resource limits to each account created.                            |
| <b>system operator</b>           | A user associated with the system console. The system operator monitors the system console and uses it to manage jobs and sessions, store and restore data, and perform system backups.                                    |
| <b>system processing unit</b>    | SPU: The SPU contains all the components of the computer system including the central processing unit. SPU does not refer to the system console or any other peripheral devices. See also <b>central processing unit</b> . |
| <b>system prompt</b>             | See <b>prompt</b> .                                                                                                                                                                                                        |
| <b>tape density</b>              | Tape density is measured in bits-per-inch. Hewlett-Packard supported tape densities are 800, 1600, and 6250 bits-per-inch. (AS)                                                                                            |
| <b>tape drive</b>                | A hardware device used to store and restore data from disk to magnetic tape and from magnetic tape back to disk.                                                                                                           |
| <b>temporary file</b>            | A file that exists only for the duration of a session or job. There is no entry in the system directory for a temporary file. (AS)                                                                                         |
| <b>terminal</b>                  | A hardware device consisting of a keyboard and a display screen. It is used for entering data to and receiving data from the computer.                                                                                     |
| <b>trailer</b>                   | The last page printed when output is directed to a line printer. It contains                                                                                                                                               |

|                               |                                                                                                                                                                                                                                                                                 |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                               | the session name (if specified), session number, log on identification, day of the week, date, and time. It corresponds to the header printed as the first page of a document.                                                                                                  |
| <b>truncate</b>               | To cut off or shorten data. If too many characters appear on a line they may not all be recognized by the system or printed as output.                                                                                                                                          |
| <b>user</b>                   | A person logged on to the computer. Each user is identified by a user and account name. A user logs on to a session, and can access files in the logon group.                                                                                                                   |
| <b>user command</b>           | A user command is a UDC or command file created to execute one or more MPE/iX commands. (AS)                                                                                                                                                                                    |
| <b>user-defined command</b>   | UDC: A file, or portion of a file, containing user-defined commands. A UDC can be used to execute one or more specified MPE/iX commands to perform a specific task. The header (name) of the UDC is defined before the commands are listed and is then used to execute the UDC. |
| <b>user-defined variable</b>  | Used to store user-assigned information. Variables can be used to assign a string value, an integer value, or a Boolean value. (AS)                                                                                                                                             |
| <b>utility</b>                | A utility is a system program that performs specific functions such as copying files, sorting or merging data, memory dump analysis, or monitoring available disk space.                                                                                                        |
| <b>variable</b>               | Variables can be system-defined or user-defined. They are specified with the SETVAR command and the values that they contain may be constant or dynamic. Variable names are a string of one or more alphanumeric characters beginning with an alphabetic character. (AS)        |
| <b>variable dereferencing</b> | Variable dereferencing refers to the act of substituting the value of a variable in place of the variable name. See also <b>explicit dereferencing</b> and <b>implicit dereferencing</b> . (AS)                                                                                 |
| <b>VERSION</b>                | A subsystem utility used to display the attributes of an executable file.                                                                                                                                                                                                       |
| <b>VOLUTIL</b>                | Volume Utility: A subsystem that allows for the management of volume sets (disk drives).                                                                                                                                                                                        |

|                           |                                                                                                                                                                                                                                                 |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Warn message</b>       | A message sent from the system console to all users. A <b>WARN</b> message interrupts all sessions on the system, including those running in quiet mode.                                                                                        |
| <b>welcome message</b>    | A message created by the system operator that is displayed each time that a user logs on to the system. It usually contains a greeting and important system information.                                                                        |
| <b>WHILE loop</b>         | A statement used to determine what action occurs. When the conditions are true, the action is performed repeatedly in a continuous loop. A <b>WHILE</b> statement is terminated with an <b>ENDWHILE</b> statement. (AS)                         |
| <b>wildcard character</b> | Wildcard characters are used to replace a character or set of characters. MPE/iX uses the pound sign (#), the at sign (@), and the question mark (?) as wildcard characters. Other subsystems may use different symbols as wildcard characters. |
| <b>word</b>               | A word consists of 32 bits (4 bytes) of information on the 900 Series HP 3000 (MPE/iX). A word consists of 16 bits (2 bytes) of information on other HP 3000 systems (MPE V). (AS)                                                              |
| <b>word processor</b>     | A word processor is a utility program that supports the creation, modification, or deletion of letters, memos, reports, and other written documents.                                                                                            |

# Index

---

- A**
  - ABORT, 4-26
  - accessing files, 6-3
  - account manager, 2-4, 3-12
  - accounts, 3-12, 5-3
  - account structure, 5-1
    - accounts, 5-3
    - billing, 5-3
    - elements, 5-6
    - files, 5-4
    - groups, 5-4
    - organization, 5-3
    - security, 5-3
    - users, 5-4
  - account structure, MPE/iX, 5-3
  - add a new line, 4-13
  - adding lines, 4-14
  - alphanumeric characters in file names, 4-10
  - answers, pretest, 8-1, 8-4, 8-6, 8-8, 8-10, 8-12
  - arrow keys, 3-9
  
- B**
  - BACKSPACE, 3-8
  - backspacing, 2-14
  - backups, 3-3, 6-12
  - batch job, 2-8
  - batch processing, 2-8
  - blocking messages, 2-24, **2-25**
  - BREAK, 4-24, 4-26
  - BYE, 2-3, 2-5
  
- C**
  - capabilities, 5-11
  - change
    - user password, 5-20
  - changing groups, 5-18
  - changing modes, 3-13
  - characters
    - alphanumeric, 4-10
    - special, 4-10
  - characters, special
    - slash (/), 4-10
    - underbar (\_), 4-10
  - CHGROUP, **5-18**, 5-19
  - command
    - ABORT, 4-26
    - BYE, 2-3, 2-5
    - CHGROUP, **5-18**, 5-19

- CONSOLE, 2-26
- COPY, 6-16, 6-17, 6-19, 6-20
- correcting, 2-15
- correcting errors, 2-14
- CTRL Q, 2-11
- CTRL S, 2-11
- editing, 2-14
- entering, 2-9
- EOD, 6-15
- executing, 2-9
- HELLO, 2-3, 2-5
- HELP, 2-31
- LISTFILE, 6-3, 6-5
- LISTREDO, 2-20, 2-21
- LISTUSER, 5-9, 5-11
- NEWACCT, 5-12
- PRINT, 6-15
- PURGE, 6-12
- REDO, 2-14, 2-15, 2-16, **2-22**
- reentering, 2-14
- RELEASE, 6-16, 6-18
- RENAME, 6-10
- RESUME, 4-25
- SECURE, 6-16, 6-23
- SET, 2-25
- shortening, 7-2
- SHOWJOB, 2-9, 2-11, 2-12
- SHOWME, 2-9
- SHOWTIME, 2-9
- STORE, 6-11
- TELL, 2-24, 2-26
- TELLOP, 2-24, **2-26**
- understanding, 2-9
- XEQ, 7-6
- command file
  - what is, 7-2
- command files, 2-12, 7-1-7
  - creating, 7-2, 7-3
  - executing, 7-3
  - using, 7-2
- command history, 2-21
- command interpreter, 3-5
- command interpreter (CI), 3-6
- command mode, 4-2, 4-5
- commands, 2-7, 2-9, 4-1
- command syntax, 2-32
- communicating with other users, 2-24
- communicating with the computer, 2-1
- components, 3-2
- computer
  - hardware, 3-2
  - overview, 3-1
  - peripherals, 3-2
- configuration, 3-15
- CONSOLE, 2-26

- continue a program, 4-25
- conventions, file names, 4-10
- conventions, file naming, 4-10
- conventions, logon, 1-4
- COPY, 6-16, 6-17, 6-19, 6-20
- copying files, 6-16
- correcting commands, 2-14, 2-15, 2-16
- CPU, 3-2
- creating command files, 7-2, 7-3
- CTRL Q, 2-11
- CTRL S, 2-11
- cursor, 3-8, 4-2, 4-5
- cursor control keys, 3-9

**D**

- data processing, 3-5
- date, 4-25
- default file names, 6-19
- default logon group, 5-8
- DELETE, 4-16
- delete a character, 4-18
- DELETE ALL, 4-19
- deleting files, 6-12
- delimiter, comma, 6-4
- delimiters, 5-6
- destination file, 6-16, 6-19
- destination group, 6-20
- destroy a file, 4-7
- devices, 3-2
- disk, computer, 4-6
- disk drives, 3-2
- display
  - file names, 4-8
  - lines, 4-5
  - portions of a file, 4-5
- displaying file contents, 6-15
- D0, 2-21
- duplication, name, 7-5

**E**

- EDIT/3000, 4-12-29
  - fundamentals, 4-2
  - starting, 4-2
  - what is EDIT/3000, 4-2
  - why use EDIT/3000, 4-2
  - writing, 4-3
- EDIT/3000 commands, 4-2
  - //, 4-2, 4-5
  - ADD, 4-2, 4-3, 4-6, 4-13, 4-14
  - DELETE, 4-13
  - END, 4-2, 4-7
  - Erase, 4-16
  - GATHER, 4-13, 4-17
  - GATHER ALL, 4-13, 4-16
  - JOIN, 4-22
  - KEEP, 4-2, 4-6

- LIST, 4-5, 4-13
- MODIFY, 4-13, 4-18
- TEXT, 4-13
- edit a line, 4-13, 4-18
- edit mode, 4-3
- editors, line
  - EDIT/3000, 4-4
- electronic mail, 3-5
- elements of account structure, 5-6
- ENTER, 3-8
- entering commands, 2-9
- :EOD, 6-15
- erase a line, 4-13, 4-16
- erasing files, 6-12
- error messages, 2-14
- examples, help, 2-32
- executing command files, 7-3
- executing commands, 2-9
- external devices, 3-2

**F** file

- copy, 6-19
- destination, 6-16
- source, 6-16
- target, 6-16
- file names, 4-1
  - conventions, 4-10
  - default, 6-19
  - display, 6-6
  - displaying, 6-3, 6-5
  - fully qualified, 6-3, 6-4, 6-7
  - invalid, 4-10
  - partly qualified, 6-3
  - valid, 4-10
- files, 5-4
  - accessing, 6-3
  - copying, 6-16
  - creating, 4-1
  - displaying, 6-15
  - editing, 4-1
  - erasing, 6-12
  - locking and unlocking, 4-10
  - purging, 6-12
  - qualifying, 6-3
  - recovering, 6-12
  - removing, 6-12
  - renaming, 6-10
  - storing, 6-11
  - text, 4-1
- files, command, 7-1
- file security, **5-14**, 5-16
- files, working with, 6-1
- finding files, 6-7
- form/menu creation, 3-5

function keys, 3-10

**G** groups, 5-4

**H** hard reset, **3-17**, 3-18  
hardware, computer, **3-2**  
header, 4-14  
HELLO, 2-3, 2-5  
HELP, 2-31  
help facility, 2-29, **2-31**  
HELPSTUDY, 2-33  
history stack, 2-21  
history stack, command line, 2-20  
home group, 5-1, 5-8, 5-9  
hung terminal, 3-16

**I** identifier, logon, 4-14  
identity, logon, 2-3, 2-4, 5-6  
identity, session, 2-12  
identity, user, 5-7  
information on terminal screen, 2-31  
insert a character, 4-18  
interactive processing, 2-7  
interactive system, 2-7

**J** job files, 2-7  
job limit, 2-29  
job listing, 2-8  
jobs, 2-7  
jobs, identity, 2-12  
joining files, 4-22  
    at the end, 4-23  
    in the middle, 4-24  
joint logon, 5-7

**K** keeping a file, 4-6  
keyboard, 3-7  
    HP 2392, 3-7  
    HP 700/92, 3-7  
keypad, 3-9  
keys  
    ARROW, 3-9  
    BACKSPACE, 3-8  
    CAPS LOCK, 3-9  
    CLEAR DISPLAY, 3-11  
    CLEAR LINE, 3-11  
    CTRL, 3-11  
    cursor control, 3-9  
    DEL/ESC, 3-11  
    DELETE CHAR, 3-11  
    DELETE LINE, 3-11  
    ENTER, 3-8  
    EXTENDED CHAR, 3-11

- function, 3-10
- INSERT CHAR, 3-11
- INSERT LINE, 3-11
- MENU, 3-11
- NEXT, 3-11
- numeric, 3-9
- PREV, 3-11
- PRINT/ENTER, 3-11
- RESET/BREAK, 3-11
- Return, 3-8
- SELECT, 3-11
- STOP, 3-11
- USER/SYSTEM, 3-11
- keywords, help, 2-32
- K files, 4-27, 4-28
- K numbers, 4-28

**L**

- launch a job, 2-7
- leave EDIT/3000, 4-7
- line numbers, 4-4
- line printer, 4-13
- line printer (LP), 3-3
- LIST, 4-13
- LISTFILE, 4-8, 6-3, 6-5, 6-6
- listing, 4-14
- listing file in EDIT/3000, 4-22
- listing offline, **4-13**
- LISTREDO, 2-20, 2-21
- LISTUSER, 5-9, 5-11
- local mode, 3-13
- locating files, 6-7
- locked terminal, 3-16
- locking files, 4-10
- logging off, 2-3, 2-5
- logging on, 2-3, 2-4, 2-5
- logon
  - identity, 5-7
  - requirements, 5-8
- logon conventions, 1-4
- logon elements
  - account name, 5-6
  - group name, 5-6
  - session name, 5-6
  - user name, 5-6
- logon group
  - default, 5-8
- logon identifier, 4-14, 6-4
- logon identity, 2-3, 2-4, 2-10, 2-24
  - account, 5-6
  - group, 5-6
  - user, 5-6
- logon prompt, 2-3, 2-5
- looking at your file, 4-5

- M**
  - main
    - logon security, 5-12
    - security:logon, 5-12
  - main system account, 5-8
  - manager
    - account, 2-4
    - system, 2-4
  - messages, send and receive, 3-5
  - mistakes, 2-14
  - mode
    - changing, 3-13
    - local, 3-13
    - remote, 3-13
    - terminal, 3-13
  - moving lines, 4-13, 4-17
  - MPE/iX operating system, 3-5
  - Multi-Programming Executive (MPE), 3-5
  
- N**
  - name duplication, 7-5
  - naming conflicts, 7-5
  - NEWACCT, 5-12
  - number
    - LDEV, 2-11
    - session, 2-11
  - numbers, line, 4-4
  - numeric keypad, 3-9
  
- O**
  - online, 2-9
    - calendar, 2-9
  - online, help, 2-31
  - online tutorial, HELPSTUDY, 2-33
  - open a file, 4-13
  - operating system, 3-5
    - MPE/iX, 3-5
  - operation, help, 2-32
  - overview, computer, 3-1
  - overwriting, 6-17
  
- P**
  - parameters, help, 2-32
  - password protection, 5-8
  - passwords, 2-4, 2-5, 5-3
    - account, 5-12
    - group, 5-12
    - user, 5-12, 5-20
  - peripherals, 3-2, **3-2**
    - disk drives, 3-2
    - printers, 3-3
    - system console, 3-3
    - tape drives, 3-2
    - terminals, 3-2
  - peripherals, computer, 3-2
  - physical equipment, 3-2
  - pretest

- answers, 8-4, 8-6, 8-8, 8-10, 8-12
- questions, 2-1, 3-1, 8-6, 8-10
- pretest, answers, 8-1
- print, 4-13
- PRINT, 6-15
- printer, line, 4-13
- printers, 3-3
- printing an editor file, 4-13
- program file, 7-6
- programs, 3-2, 3-5
- prompt, 4-6
  - EDIT/3000, 4-2
  - MPE XL, 4-2
- PUB group, 5-3, 5-8, 5-9
- pull file, 6-20
- PURGE, 4-7, 6-12
- purge a file, 4-7
- purging a file, 6-12
- push file, 6-20

**Q** qualifying files, 6-3  
questions, pretest, 2-1, 3-1, 8-6, 8-10  
queue, 4-13  
quiet mode, 2-24, **2-26**

**R** receive messages, 3-5  
receiving messages, 2-24  
recovering a file, 4-28, 6-12  
REDO, 2-14, 2-15, 2-16, 2-20, **2-22**  
reentering a command, 2-14  
reexecute a command, 2-15  
reinstating system security, 6-23  
RELEASE, 6-16, 6-18  
remote mode, 3-13, 3-15  
remove security provisions, 6-18  
removing a file, 6-12  
RENAME, 6-10  
renaming files, 6-10  
renumbering lines, 4-16  
renumber lines, 4-13  
replace a character, 4-18  
restore from tape, 3-3  
RESUME, 4-25  
retrieve a command, 2-20  
retrieve a line, 4-18  
Return, 3-8

**S** safety files, 4-27  
 saving a file, 4-6  
 schedule a job, 2-7  
 scrolling, 2-11, 3-8, 4-5  
 SECURE, 6-16, 6-23  
 security  
   command, 5-11  
   file, 5-11, **5-14**  
   group, 5-11  
   logon, 5-11  
   violation, **5-15**  
 security, file, 5-16  
 sending a file to the line printer, 4-13  
 sending messages, 2-24  
 send messages, 3-5  
 session, 5-7  
   identifier, 5-7  
   identity, 2-12  
   number, 2-24  
   starting, 2-5  
 sessions, 2-7  
 SET, 2-25  
 shortening commands, 7-2  
 SHOWJOB, 2-9, 2-11, 2-12  
 SHOWME, 2-9  
 SHOWTIME, 2-9, 4-25  
 softkeys, 3-10  
 soft reset, **3-16**, 3-18  
 software, 3-2, 3-5  
 source file, 6-16  
 special characters in file names, 4-10  
 spool, 4-13  
 SPOT key, 2-11  
 SPU, 3-2  
 stack, history, 2-20  
 starting a new editor file, 4-19  
 starting a session, 2-3  
 starting EDIT/3000, 4-2  
 stop a program  
   ABORT, 4-26  
 stopping scrolling , 4-5  
 stop writing (/), 4-5  
 STORE, 6-11  
 store on tape, 3-3  
 storing files, 6-11, 6-12  
 streaming a job, 2-7  
 submitting a job, 2-7  
 subsystem, EDIT/3000, 4-2  
 suspending EDIT/3000, 4-24  
 syntax, command, 2-32  
 SYS account, 5-3, 5-8  
 system console, 3-2, 3-3  
 system manager, 2-4, 2-9, 3-12  
 system operator, 2-9, 3-12  
 system processing unit, 3-2

system prompt, 2-5

- T**
  - tape drives, 3-3
  - target file, 6-16
  - TELL, 2-24, 2-26
  - TELL0P, 2-24, **2-26**
  - terminal
    - configuration, 3-15
    - locked, 3-16
  - terminals, 3-2
    - Hewlett-Packard 2392 series, 3-7
    - Hewlett-Packard 700/92 series, 3-7
    - HP 150, 3-7
    - HP Vectra, 3-7
  - TEXT, **4-13**
  - time, 4-25
  - trailer, 4-14
  - typing lines, 4-4
  
- U**
  - unblocking messages, 2-24, **2-26**
  - understanding commands, 2-9
  - unlocking files, 4-10
  - user
    - password, 5-20
  - user-defined commands, 2-12
  - user password
    - change, 5-20
  - users, 5-4
  - using command files, 7-2
  
- V**
  - verifying name duplications, 7-5
  - version, MPE, 2-11
  - violation, security, **5-15**
  
- W**
  - waiting files, 4-13
  - warning messages, 2-27
  - welcome message, 2-5
  - wildcard characters, 6-3, 6-6, 6-7
  - word processing, 3-5
  - word processors, 4-4
  - writing in EDIT/3000, 4-3
  
- X**
  - XEQ, 7-6