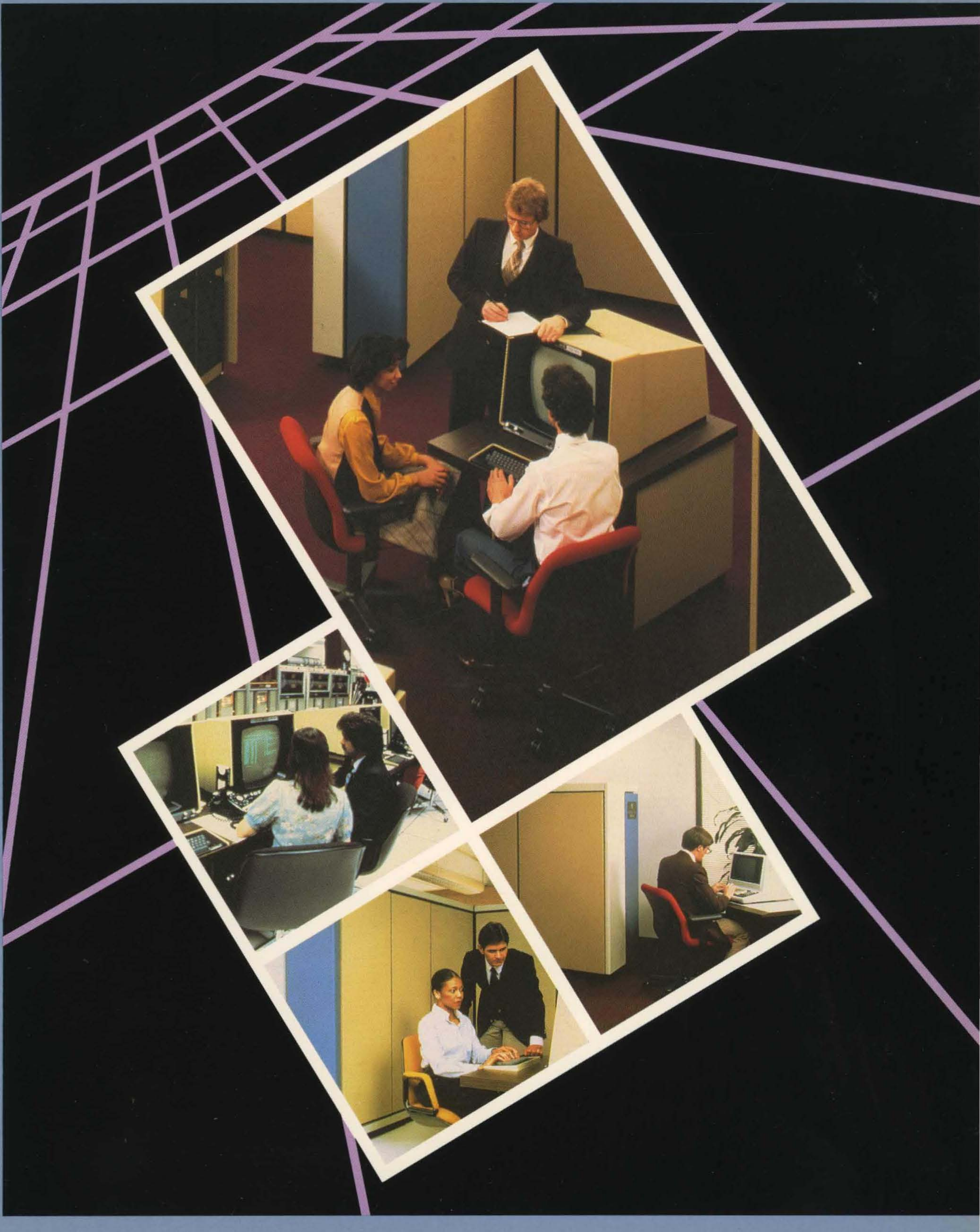
NOS 2 Reference Set Volume 1
Introduction to Interactive Usage

CONTROL DATA



Index to Information

Index to Commands and Other Entries

NOS 2 Reference Set

Volume 1

Introduction to Interactive Usage

This guide describes a subset of the features documented in Volumes 2, 3, and 4 of the NOS 2 Reference Set. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Publication Number 60459660

Revision Record

Revision C reflects NOS 2.3 released on 09-19-84 at PSR level 617. This revision includes miscellaneous technical and editorial changes. This edition obsoletes all previous editions.

Previous Revisions	System Level	Date
A	562	04-26-82
B	596	10-11-83

Preface

This manual describes the CONTROL DATA® Network Operating System (NOS) Version 2. NOS controls the operation of CDC® CYBER 170 Computer Systems; CDC CYBER 70 Computer Systems, Models 71, 72, 73, and 74; and CDC 6000 Computer Systems.

Organization and Audience

This manual is the first of four volumes in the NOS 2 Reference Set. Volume 1 shows you how to enter, run, and correct programs and how to create, retrieve, and maintain permanent files at an interactive terminal. Other topics covered include physical terminal connection and procedures for entering and leaving the system. It is written for the person who has had little or no experience using computer systems in general and NOS in particular. The user writing programs is assumed to be either familiar with a programming language or taking a programming class. Descriptions and examples use FORTRAN Version 5 and BASIC Version 3, but generally the information applies to other languages also. A glossary of terms is provided in appendix A. Use of this manual does not require familiarity with any other NOS manual, but for a general introduction, we recommend that you first read the *NOS Version 2 System Overview.*

Volume 2, Guide to System Usage, describes the general concepts of NOS and some of the utilities used with NOS. Topics included are job processing, file concepts, procedures, magnetic tape processing, modifying files, and loading files. This volume is a learning tool and does not contain comprehensive descriptions of all NOS commands. It is written for the applications or systems programmer who is unfamiliar with NOS.

Volume 3, System Commands, describes the system commands that form the user interface with NOS. It is written for all NOS users.

Volume 4, Program Interface, describes the COMPASS program interface with NOS. Detailed descriptions of function processors and macros available to COMPASS user programs are included. It is written for the experienced COMPASS applications programmer or systems programmer.

The reader of each volume should have a knowledge of the material contained in the preceding volumes.

Conventions and Assumptions

In this manual, entries you can make to the computer system are shown in blue. When you type an entry at a terminal, you type the characters without pressing the shift key or any other key unless specifically instructed to do so. On most terminals, the letters you type will appear as lowercase letters. Therefore, in examples that show how your entries and the system's responses might look at a terminal, your entries are shown in lowercase blue letters, and system responses are shown in uppercase black letters.

In general instructions in the text, however, entries are shown in both uppercase and lowercase blue letters to indicate required characters. You must type the uppercase letters and the punctuation exactly as shown; you replace the lowercase letters with characters you have chosen. For example, an entry shown in text as

```
SAVE,filename
```

could be entered at a terminal as

```
save,myfile
```

The notation yy/mm/dd is used throughout this manual to indicate the current date, for example, 84/01/06 means that the date is January 6, 1984. Similarly, the notation hh.mm.ss is used to indicate the current time, for example, 14.25.30 means that the time is 30 seconds after 2:25 p.m.

The procedures described in this manual (such as entering the system and using the carriage return, the termination sequence, the interruption sequence, and the cancel line sequence) assume you have a standard ASCII terminal and are using a standard NOS system. If you are using any other kind of terminal or if your computer center has made modifications to the system, you need additional or amended procedures. Appendix B contains some additional information on terminals.

Related Publications

Following is a list of NOS manuals that you may be interested in.

The NOS Version 2 Manual Abstracts is a pocket-sized manual containing brief descriptions of the contents and the intended audience of all NOS 2 manuals and NOS 2 optional products manuals. The abstracts can be useful in determining which manuals are appropriate for your use.

Control Data publishes a Software Publications Release History of all software manuals and revision packets it has issued. This history lists the revision level of a particular manual that corresponds to the level of software installed at the site.

Control Data manuals are available through Control Data sales offices or Control Data Literature Distribution Services, 308 North Dale Street, St. Paul, MN 55103.

Control Data Publication	Publication Number
BASIC Version 3 Reference Manual	19983900
FORTRAN Version 5 Reference Manual	60481300
NOS Full Screen Editor User's Guide	60460420
NOS Version 2 Manual Abstracts	60485500
NOS Version 2 Reference Set, Volume 2, Guide to System Usage	60459670
NOS Version 2 Reference Set, Volume 3, System Commands	60459680
NOS Version 2 Reference Set, Volume 4, Program Interface	60459690
NOS Version 2 System Overview	60459270
NOS Version 2 Screen Formatting Reference Manual	60460430
Software Publications Release History	60481000
Text Editor Version 1 Reference Manual	60436100
XEDIT Version 3 Reference Manual	60455730

Contents

Appendixes

Index

Introduction 1

This manual shows you how to use a terminal to communicate with your computer system. Knowing this will help you do your work faster and more efficiently. The computer system you are going to use is extremely powerful, but it is easy to use. It understands a large set of instructions, but allows you to perform most operations with just a few instructions.

How to Use This Manual

If you have specific instructions to perform as soon as the system responds to you, you need only read sections 2 and 3. They explain how you connect a terminal to the system, tell the system that you are ready, and leave the system when you are done. If you have any problems, refer to section 7. This section describes some common problems users have and suggests solutions.

If you are writing and running your own programs, you also need to read sections 4, 5, and 6. They show you how to enter a program, run it, and get the results back at your terminal. If the program doesn't work, these sections tell you how to correct and retry it. They also describe how you tell the system to save your program so you need not reenter it each time you use the terminal. We assume you either are taking a programming class or know a programming language already. This book alone does not teach programming concepts or specific languages.

In this manual, entries you can make to the computer system are shown in blue. When you type an entry at a terminal, you type the characters without pressing the shift key or any other key unless specifically instructed to do so. On most terminals, the letters you type will appear as lowercase letters. Therefore, in examples that show how your entries and the system's responses might look at a terminal, your entries are shown in lowercase blue letters, and system responses are shown in uppercase black letters.

In general instructions in the text, however, entries are shown in both uppercase and lowercase blue letters to indicate required characters. You must type the uppercase letters and the punctuation exactly as shown; you replace the lowercase letters with characters you have chosen. For example, an entry shown in text as

```
SAVE,filename
```

could be entered at a terminal as

```
save,myfile
```

The last page of this manual is a comment sheet. If you have problems using a section or have suggestions, we would like to hear from you. Simply write your comments on the page and mail it to us.

How to Use the System

Each time you use the system, you:

1. Connect your terminal to the system.

   Some terminals are wired so that they are always connected to the system. Other terminals you must connect to the system using telephone lines. The next section explains how you can tell the difference.

2. Tell the system that you are ready to start.

   This is called logging in to the system. The system asks for specific information to identify you as an authorized user. All you need do is respond to its prompting.

3. Do your work.

4. Tell the system that you're done.

   This is called logging out of the system. Just type

   ```
   BYE
   ```

Every person who uses the system, whether a beginner or an experienced programmer, follows these same steps. Section 2 describes step 1; section 3 describes steps 2 and 4.

The best way to become comfortable with the system is to experiment with it. It's almost impossible for you to accidentally damage the system or someone else's work. Before you can start, however, you need some information.

The following page, *Information Specific to Your System and Terminal,* lists the information you will need. You may want to fill in this sheet and keep all the information together. Before starting, however, you should either remove the page from the manual or make a copy for your own use. This is done to protect the information should you lose the manual or let others use it.

NOTE

Keeping this information in a safe, secure place prevents others from being able to access your programs and data.

The first section on the information sheet is for terminal settings. They are discussed in section 2 which explains how to set up your terminal. You can fill in this section later.

Next is a section for computer system phone numbers. For terminals that you must connect to the system by phone, you need to know the telephone numbers used to reach your system. Your employer, instructor, or someone at your computer center can give you these numbers. There may be one or several. Sometimes your computer center will associate certain phone numbers with certain characteristics of a terminal, such as terminal line speeds (the speed at which the terminal prints characters). You can list the numbers (and associated line speeds, if any) on the information sheet.

INFORMATION SPECIFIC TO YOUR SYSTEM AND TERMINAL

Terminal settings: ____

____

____

____

____

____

Computer system phone numbers: ______________ Associated terminal line speeds: ______________

______________ ______________

______________ ______________

______________ ______________

______________ ______________

Family name: ______________________________

User name: ______________________________

Password: ______________________________

Special procedures for login: ____

____

____

____

____

____

Interruption sequence: ____

Termination sequence: ____

Cancel line sequence: ____

Computer center phone number: ______________

You must be able to tell the system who you are. Again, your employer, instructor, or computer center personnel can give you:

- A family name (may not be required).

- A user name.

- A password.

The system asks you for some or all of this information when you log in. You can record these names on the information sheet.

Finally, ask your employer, instructor, or computer center personnel if there are any special procedures for using your particular terminals or system. For example, it's possible your system may actually consist of more than one physical computer. In this case, you'll be asked to choose one while you are logging in. This is the sort of additional information you could be given. You may want to insert these procedures in this manual or to record them on the information sheet.

The interruption sequence, termination sequence, and cancel line sequence are entries you may want to make while you're using the system. We'll talk more about them and show you what their usual values are later in this manual.

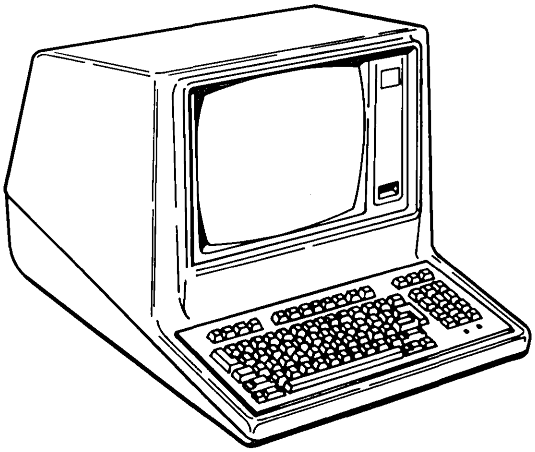
Now, go to a terminal. You don't need work ready yet. To start, you can connect the terminal, log in, and then log out.

If possible, have an experienced person show you how to connect the terminal, log in, and log out. This process is easy once you've seen it done, but the process on your system may differ slightly from that described in the next two sections of this manual. Because there are many kinds of terminals available and many ways your system could be modified, we've had to make certain assumptions about the terminals and system features that we describe. If you follow along using the next two sections and the system doesn't respond as expected, ask your computer center personnel for help.

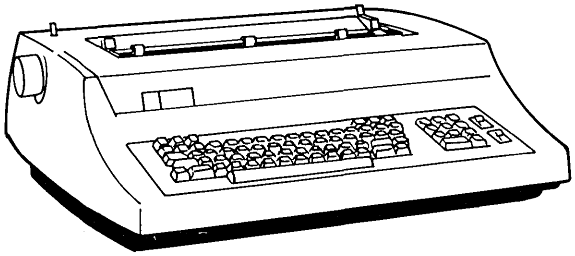
If you run into problems anytime you're using a terminal, refer to section 7, *What to Do When You Have Problems.*

Connecting Your Terminal to the System 2

Many different kinds of terminals are available. Some communicate with you using a screen (somewhat like a television screen). Others type on paper (like a typewriter). But all terminals have a keyboard that you use to type instructions to the system. Every instruction that you type and the system's response to that instruction appear either on the screen or the paper.



Display Screen Terminal



Printing Terminal

Setting Up Your Terminal

Before you can connect your terminal to the system, the terminal must be turned on and certain switches must be in the correct positions. The switches and their settings vary from terminal to terminal. If the terminal you're at has been used by other people working with the same system, the switches should be set correctly; don't change them. Or, directions for setting the switches may be posted near the terminal. If you're not sure what to do, ask someone to show you. You may want to record some of this information about your terminal on the information sheet in section 1.

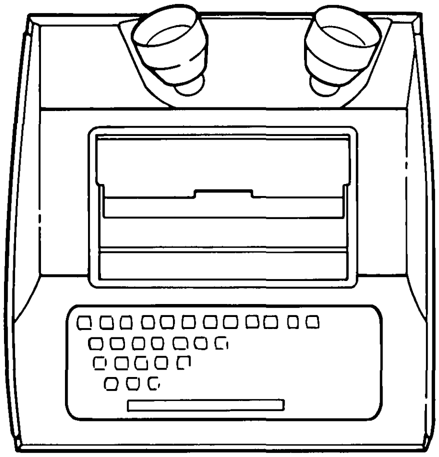
Remember that certain terminal line speeds may be associated with certain phone numbers. If applicable, set the line speed switch to a number that matches one of those associated with the phone numbers you were given. If you're not sure of the speed you should use, set the switch to any position (usually 10 to 30 characters per second). The system may be able to detect automatically what line speed you are using. Now you're ready to connect your terminal to the system.

Connecting Your Terminal

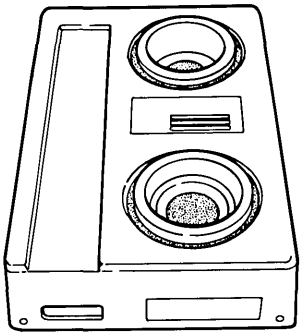
Some terminals are wired so that they are automatically connected to the system when they're turned on. You may hear them called hardwired terminals. Other terminals are connected to the system using a telephone and telephone lines. If there is no telephone near the terminal you are using, you can assume it is a hardwired terminal. In this case, you can skip the rest of this section and go on to section 3.

A terminal that requires a telephone uses either an acoustic coupler or a data set.

An acoustic coupler has two rubber cups into which the phone receiver will fit. The coupler is either built into the terminal or is a separate device beside the terminal. If it is a separate device, turn it on.

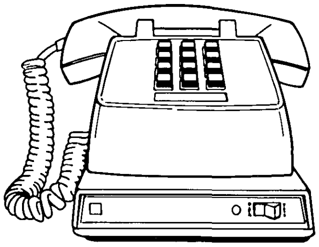


Built-in Coupler

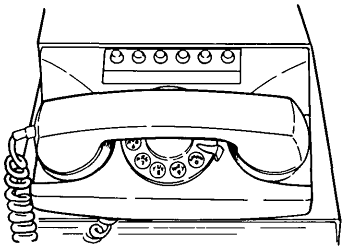


Separate Coupler

On the front of a data set is either a switch labeled TALK and DATA or individual buttons labeled TALK and DATA. Most commonly, the data set is separate from the terminal. Either set the switch to TALK or push the TALK button.



Data Set with Switch

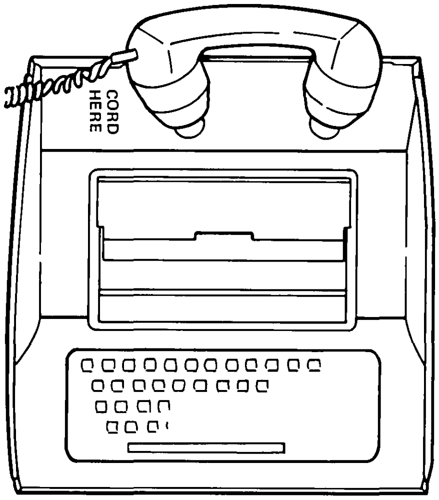


Data Set with Buttons

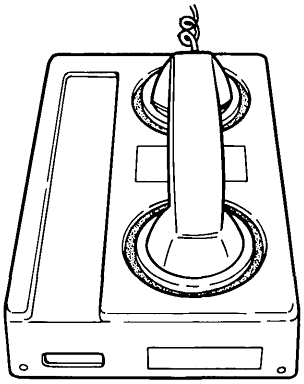
To connect your terminal to the system, call one of the phone numbers you were given. When the system is available, you hear a constant high-pitched tone. Usually this happens within a few rings. If it doesn't, assume the system is not available and try again later. If the line is busy, try another phone number, if you have more than one.

When you hear a constant high-pitched tone, your next action depends on whether you are using a terminal with an acoustic coupler or a data set.

If the terminal uses an acoustic coupler, insert the phone receiver in the two rubber cups. Usually the coupler is marked to indicate how the receiver should be inserted.



Built-in Coupler



Separate Coupler

If the terminal uses a data set, either set the switch on the front of the data set to DATA or push the DATA button. Then replace the phone receiver.

Usually an indicator on the terminal lights to let you know that it is connected to the system. Wait a full 2 seconds after the light comes on. You are now ready to enter the system and do your work.

Entering and Leaving the System 3

Entering the System

This is a procedure that identifies you to the system. In the first section, you were told to gather some information (your family name, user name, and password). The system now asks you for that information, one item at a time. All you have to do is respond with the names you were given. Figures 3-1 and 3-2 are two examples of logins as they might appear during a terminal session (a terminal session begins when you log in and ends when you log out). In the examples:

- The family name is SYSTEMA.
- The user name is JLC6021.
- The password is JLC6.

Figure 3-1 shows a login with no errors. Figure 3-2 shows a login where the user makes a mistake typing the user name; the system asks for all the entries again. Everything that the user types is shown in lowercase blue letters. After every entry, the user presses the carriage return.

Terminal Session	Comments
`carriage return`	The user presses the carriage return twice to begin the process. Pressing the carriage return helps the system identify line speed, character set, and so forth, of the terminal being used.

`carriage return`
`WELCOME TO THE NOS SOFTWARE SYSTEM.`
`COPYRIGHT CONTROL DATA 1978, yyyy.`
`yy/mm/dd. hh.mm.ss. T01A78`
`CDC NETWORK OPERATING SYSTEM   NOS 2`	

`FAMILY: systema`
`USER NAME: jlc6021`
`PASSWORD:jlc6`
	The user responds to system prompts. On most terminals, the screen clears immediately after the password is entered correctly.

`T01A78   - APPLICATION: iaf`
`JSN: AAYJ, NAMIAF`
	In response to the prompt for an application name, the user types
	`IAF`

`READY.`	The system indicates that it's ready to accept instructions.

Figure 3-1. Login with No Errors

Terminal Session	Comments
`carriage return`	The user presses the carriage return twice to begin the process. Pressing the carriage return helps the system identify line speed, character set, and so forth, of the terminal being used.

`carriage return`
`WELCOME TO THE NOS SOFTWARE SYSTEM`
`COPYRIGHT CONTROL DATA 1978, yyyy.`
`yy/mm/dd. hh.mm.ss. T01A6D`
`CDC NETWORK OPERATING SYSTEM   NOS 2`
`FAMILY: systema`
`USER NAME: uod6021`
	The user types the user name incorrectly.

`PASSWORD: jlc6`
`IMPROPER LOGIN, TRY AGAIN.`
`FAMILY: systema`	

`USER NAME: jlc6021`
`PASSWORD: jlc6`
	This time the user types the user name correctly. On most terminals, the screen clears immediately after the password is entered correctly.

`T01A6D   - APPLICATION: iaf`
`JSN: ABZH, NAMIAF`
` READY.`
	The user types
	`IAF`
	in response to the prompt for an application name.

Figure 3-2. Login with Error

Now try it yourself using the information you were given. Wait a full 2 seconds after the light that indicates terminal connection comes on. Then press the carriage return twice. (Depending on your terminal, this key may be labeled RETURN, CR, CARRIAGE RETURN, NEXT, SEND, NEW LINE, or ETX. If you're not sure which key is the carriage return, ask someone.)

The system responds to each carriage return with one or two blank lines. If you don't see a response, press the carriage return again. The system then displays four lines that will look something like this:

```
WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, yyyy.

yy/mm/dd. hh.mm.ss. T01A78
CDC NETWORK OPERATING SYSTEM   NOS 2
```

The first two lines are always the same. The third line gives the current date (yy/mm/dd) and time (hh.mm.ss) and a terminal name (you don't need to be concerned about the name). The fourth line usually identifies your computer center or the system you're using in more detail.

At this point, the system may ask for your family name with the following prompt.

```
FAMILY:
```

If it does, type the family name you were given and press the carriage return. If you are using a family name supplied by the system (a default name), pressing the carriage return is the only entry you need make. You must spell every entry correctly and follow each entry with a carriage return. If you do make a mistake, however, the system gives you an opportunity to retry your entries later.

Next, the system asks for your user name with the following prompt.

```
USER NAME:
```

Type the user name you were given.

Next the system asks for your password with the following prompt.

```
PASSWORD:
```

Type the password you were given. Although our examples show the password that is entered, some display terminals do not actually display the characters on the screen. Many printing terminals overprint several characters on a line and ask that you type your password on that line.

```
PASSWORD:
████████
```

This is done to obscure your password and to protect the security of your user name should you lose or throw away paper printed out from a terminal session.

If the system displays the message

```
IMPROPER LOGIN, TRY AGAIN.
```

check the names you just entered; you may have entered one incorrectly. (For example, the letter O, the numeral 0, the letter L, and the numeral 1 each has its own key on the terminal keyboard. If you type the letter O for the numeral 0 or the letter L for the numeral 1, the system regards it as an error.) When the system prompts you, try again. If you are entering everything correctly, but the system still gives you the message, tell your computer center personnel. It is a problem that they must handle.

If the system asks you to enter an application name, type

```
IAF
```

Don't worry if your system does not request all the information indicated in the example terminal sessions or mentioned in the text. Your system has been set up to automatically use certain entries (defaults) for you.

When the system is ready for you to continue, it displays another identification line (which you needn't be concerned with now) and the word

```
READY
```

Now you are at the point where you can do your work. If you have specific instructions, begin them. If you want to enter and run a program, go on to section 4. If you have no work ready yet, you can either log out (as described below) or continue on to section 4. It contains practice programs that you can enter and run to become familiar with the system. When you are done, or at any time you want to end the terminal session, follow the instructions given next for leaving the system.

Leaving the System

When you have completed your work, or at any time the system is waiting for an instruction and you want to end the terminal session, type

```
BYE
```

This tells the system that you want to leave. The system displays several lines that, for the examples we showed on login, look somewhat like this:

```
bye
UN=JLC6021   LOG OFF   10.49.38
JSN=ABDV      SRU-S         5.059
CHARACTERS=    55.146KCHS
IAF     CONNECT TIME 00.26.53.
LOGGED OUT.

HOST DISCONNECTED        CONTROL CHARACTER=(ESC)
ENTER INPUT TO CONNECT TO HOST
```

It's unlikely that you need to be concerned with these lines. They show the time you ended the terminal session, accounting information, the number of input and output characters used during the session, and other system information.

If your system is a multihost system, information about the availability of each host may also be displayed.

If no one is waiting to use the terminal and it was turned off when you found it, turn it off again. If an acoustic coupler is used, hang up the phone. If a data set is used, either set the switch to TALK or push the TALK button.

Running a Program at the Terminal 4

Putting the Program on the System

After you've logged in and the system responds

```
READY
```

you must tell it what language you want to use. This book assumes that you are using FORTRAN or BASIC, so type

```
FORTRAN
```

or

```
BASIC
```

Your system may have other languages available also; if so, enter the name of the language you want to use. Remember, after each line you type, you must press the carriage return.

For its own use, the system is going to refer to your program as a file. (We'll talk more about files later.) When it asks

```
OLD, NEW, OR LIB FILE:
```

type

```
NEW
```

We're assuming that this is the first time you've used the system and that you have no existing files (programs). The system now asks what you want to name your file with the prompt

```
FILE NAME:
```

You can enter any name you want as long as it is made up of letters and/or numbers, begins with a letter, and is no more than seven characters. This name can be the same as the name of your program, but it need not be.

When the system is ready, it displays

```
READY
```

Now you simply begin typing your program, line by line, with each line followed by a carriage return. You must assign a number to each line, so that you can refer to specific lines later. The numbers must be sequential, because the system automatically sorts and orders the lines of the program according to these line numbers. We recommend numbering by tens, so that you can insert lines easily if you want to change the program after you have entered it.

The following example shows a user logging in, creating a new file named FORTEST, and entering a FORTRAN program that reads two numbers, multiplies them, and prints the result.

```
WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, yyyy.

yy/mm/dd. hh.mm.ss. T01A6D
CDC NETWORK OPERATING SYSTEM   NOS 2
FAMILY: systema
USER NAME: jlc6021
PASSWORD: JLC6
T01A6D   - APPLICATION: iaf
JSN: ACZH, NAMIAF

 READY.
fortran
 OLD, NEW, OR LIB FILE:new
 FILE NAME:fortest
 READY.
10 program test
20 read *,i,j
30 i=i*j
40 print *,i
50 end
```

The following example shows a user logging in, creating a new file named BASTEST, and entering a BASIC program that reads two numbers, multiplies them, and prints the result.

```
WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, yyyy.

yy/mm/dd. hh.mm.ss. T01A6D
CDC NETWORK OPERATING SYSTEM   NOS 2
FAMILY: systema
USER NAME: jlc6021
PASSWORD: JLC6
T01A6D   - APPLICATION: iaf
JSN: ACZR, NAMIAF

 READY.
basic
 OLD, NEW, OR LIB FILE:new
 FILE NAME:bastest
 READY.
10 input i,j
20 i=i*j
30 print i
40 end
```

Now enter your own program or copy one of the examples. Don't worry about typing mistakes. We explain how to correct them next.

Correcting and Changing the Program

If you see an error in a line before you press the carriage return, you can backspace to the error and retype it. The backspace key is usually labeled BACK SPACE, BKSP, ← , or something similar. (If you can't find this key on your terminal, ask someone to show you the equivalent key or keys.) When you backspace, however, you are in effect erasing every character you backspace over, even though it may still appear at the terminal. After you have corrected the error, you must retype the rest of the line before pressing the carriage return.

If you have already pressed the carriage return when you see the mistake, you can still change, delete, or insert lines using the line numbers as references. Either you can do this as soon as you see the mistake, or you can wait until you've entered the entire program. Suppose you entered the FORTRAN program in the first example like this:

```
10 program test
20 trsf *,i,j
30 print *,i
40 stop
50 end
```

To change a line, simply retype the entire line, including the line number, as you would like it to appear.

```
20 read *,i,j
```

The system automatically replaces the line in the program.

To delete a single line, such as line 40, type

```
delete,40
```

The system deletes the entire line that started with 40.

To delete several consecutive lines, include the first and last line numbers separated by two periods. For example, typing

```
delete,20..40
```

would delete lines 20, 30, and 40 from the program.

If you leave out a line or find later that you need to add one between two existing lines, pick a number between the two existing line numbers. Type it and the new line as you want it to appear.

```
25 i=i*j
```

The system automatically inserts the new line.

If you insert several lines in the same area of a program, you may run out of line numbers. If this happens, type

```
RESEQ
```

and the system resequences the program with new line numbers. To check it, type

```
LIST
```

and the system prints the resequenced program.

```
reseq
 READY.
list
00100 PROGRAM TEST
00110 READ *,I,J
00120 I=I*J
00130 PRINT *,I
00140 END
```

With just these techniques, you can correct any program at any time. There are many other ways to change a program, but they are essentially short cuts to accomplish the same thing. If you're interested, the system's primary editing commands are described in *Volume 3, System Commands.*

In addition, there are several text editors which may be available on your system. They provide even more capabilities, including the ability to edit lines that don't begin with line numbers (for example, a collection of data). We'll talk more about text editors in section 5 but if you want to see some examples now, go to appendix C.

Using the Program

Now that you have a program on the system, you can list it at your terminal, run (or execute) it, and save it for future use.

To list the contents of your program, type

```
LIST
```

and the system displays the contents of your program.

```
list
10 PROGRAM TEST
20 READ *,I,J
25 I=I*J
30 PRINT *,I
50 END

 READY.
```

To display several consecutive lines, include the first and last line numbers separated by two periods. For example,

```
list,20..30
```

displays lines 20, 25, and 30.

Using the program you have created, try a few combinations to see how they work.

Running a program is the same thing as executing a program. You might hear both terms used. To run your program and get the results at your terminal, type

```
RUN
```

The following example runs the FORTRAN program that was just entered and listed.

Terminal Session	Comments

`run`
`? 4 15`
`  60`
	The system requests data for program execution by printing a question mark. You enter your data and press the carriage return. The system returns the result.

`SRU       0.289 UNTS.`
` RUN COMPLETE.`
	This number is a measure of system use; you needn't be concerned with it.

Remember, typing

```
RUN
```

runs the program only once. When the system responds

```
RUN COMPLETE
```

the program is done and the system is waiting for you to tell it what to do next. If you enter numbers now, the system assumes you are entering line numbers and it inserts them in your program.

The programs you create at your terminal are temporary. When you log out, they disappear completely unless you save them. To save your program, type

```
SAVE
```

The system makes a copy of the program and puts it in permanent storage under the file name you entered earlier. The next time you log in to use the system, ask for the file and the system will retrieve it for you.

The following example shows how to create, list, run, and save the BASIC program we showed in this section.

```
 READY.
basic
 OLD, NEW, OR LIB FILE:new
 FILE NAME:bastest
 READY.
10 input i,j
20 i=i*j
30 print i
40 end
list
10 INPUT I,J
20 I=I*J
30 PRINT I
40 END

 READY.
run
? 7 12
 84

 RUN COMPLETE.
save
 READY.
```

Suppose you end this terminal session, log in later, and want to run the program you saved. When the system is ready, type

```
OLD
```

The system asks which file you want. Enter the name you gave to the file when you created it. When the system says

```
READY
```

the program is available to you as if you had just entered it. You can then use any of the commands shown in this section to change it, list it, or run it, as we show in the following examples.

```
Terminal Session                                  Comments

WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, yyyy.

yy/mm/dd. hh.mm.ss. T01A6D
CDC NETWORK OPERATING SYSTEM   NOS 2
FAMILY: systema                                   The user logs in.
USER NAME: jlc6021
PASSWORD: JLC6
T01A6D   - APPLICATION: iaf
JSN: ACBM, NAMIAF

 READY.
old                                               The user retrieves the saved file
 FILE NAME:fortest                                FORTEST.
 READY.
list                                              FORTEST is listed.
10 PROGRAM TEST
20 READ *,I,J
25 I=I*J
30 PRINT *,I
50 END

 READY.
run                                               FORTEST is run.
? 5 30
 150

SRU      0.286 UNTS.

 RUN COMPLETE.
old                                               The user retrieves the saved file
 FILE NAME:bastest                                BASTEST.
 READY.
list                                              BASTEST is listed.
10 INPUT I,J
20 I=I*J
30 PRINT I
40 END

 READY.
run                                               BASTEST is run.
? 30 5
 150

 RUN COMPLETE.
```

Remember, programs that you saved are not automatically available at the terminal when you log in. You must ask for them explicitly.

Notice that you didn't have to tell the system what language you wanted to use this time. That's because the system remembers what language you were using when you saved the file.

Getting Help

The instructions you give the system are called commands. Many commands are actually the word that describes the operation you want the system to perform, like LIST or SAVE. If you can't remember the name of a command or want to know more about that command, type

```
HELP
```

Then just follow the instructions.

The system gives you three choices and asks you to pick one.

- You can get help with a particular command.
- You can read from a manual that will be displayed at your terminal.
- You can see a list of acceptable commands.

When you choose to get help with a particular command, the system tells you what the command does and then prompts you for any information needed to use the command, such as a file name. When it has the information it needs, the system performs the command.

Interrupting a System Operation

To interrupt a system operation, such as running or listing a program, hold down the control key (labeled CONTROL, CTRL, or something similar) and press

```
P
```

Then release both keys and press the carriage return. (This combination of keys is called the interruption sequence. If it doesn't work on your terminal, ask someone what the equivalent key or keys are. You may want to record this sequence on the information sheet in section 1.)

The system displays

```
*INTERRUPTED*
```

To continue the operation, press the carriage return.

Stopping a System Operation

To stop a system operation, such as running or listing a program, hold down the control key (labeled CONTROL, CTRL, or something similar) and press the

```
T
```

Then release both keys and press the carriage return. (This combination of keys is called the termination sequence. If it doesn't work on your terminal, ask someone what the equivalent key or keys are. You may want to record this sequence on the information sheet in section 1.)

The system displays

```
*TERMINATED*
```

You can then enter a new command.

Using Files 5

In the preceding section you used only one file at a time, the one that held your program. As you use the system more, you will probably want to have several files available during a terminal session. To use files effectively, you should know some things about them and the commands you use to manipulate them.

A file is a collection of information that you and the system refer to by a common name. A file contains anything you want, for example, a program, data to be used in a program, data resulting from the execution of a program, text data, or reports. A file is either permanent or temporary.

Permanent Files

Permanent files are those that you explicitly save during a terminal session. You can retrieve them from permanent storage at any time. When you retrieve a permanent file, the system makes a copy of the file for you to use. Any changes you make to the file are only on the copy; the actual file in permanent storage is not changed. To save the changed version of the file, you must either replace the permanent file with the changed version or save the changed version as a new file with a new file name.

Temporary Files

All files that you use at your terminal during a terminal session are temporary. That is, they disappear when you log out. This includes the copy of a permanent file, although the permanent file itself is kept for you by the system. You create or bring a file to the terminal:

- As a new, empty file (as we did when entering the program in the preceding section).
- As a copy of a permanent file (as we did when retrieving the program in the preceding section).
- As data resulting from the execution of a program.

You can designate one of your temporary files as the primary file (we describe how later in this section). In fact, the program you created in the preceding section was a primary file. The primary file has some special features.

- You can enter information into a primary file either directly from the keyboard (as we described in the preceding section) or by using the TEXT command (which we describe later in this section).
- You can edit the primary file using line numbers (also described in the preceding section).
- The primary file is the file the system uses by default if you don't specify a file name on a command. That's why you could list, run, and save your program by entering only the name of the command. There are a few special file names that you shouldn't use when naming your primary files. For example, avoid names that start with several Z's as well as names that start with the letters SCR. If you do choose to begin you file names with these letters, some of the system commands may not work for you.

- The primary file is positioned to its beginning before every operation performed on it. The system has a pointer that keeps track of where you are within a file. For example, as the system lists a file, the pointer moves down the file. When the pointer reaches the end of the file, the system stops listing. If the file is a primary file, the system automatically goes back to the beginning (rewinds the file) before the next operation. That's why you could use your program in the preceding section without worrying about where you were positioned within the file. The system does not automatically rewind nonprimary files. Before every use, you must enter a command (REWIND) to rewind them.

You will usually have one file that you use more than the others. Typically it is the file containing your program. Therefore, you will probably make that file your primary file and use one or more nonprimary files for data to be used in and/or resulting from your program's execution.

In the rest of this section, we use some figures to illustrate how file commands work. The symbol □ represents a primary file. If the box is shaded, ▨ , it represents a nonprimary file. The name of the file is in the upper left corner. The relative positions of the files have no significance. Figure 5-1 illustrates primary and nonprimary files.

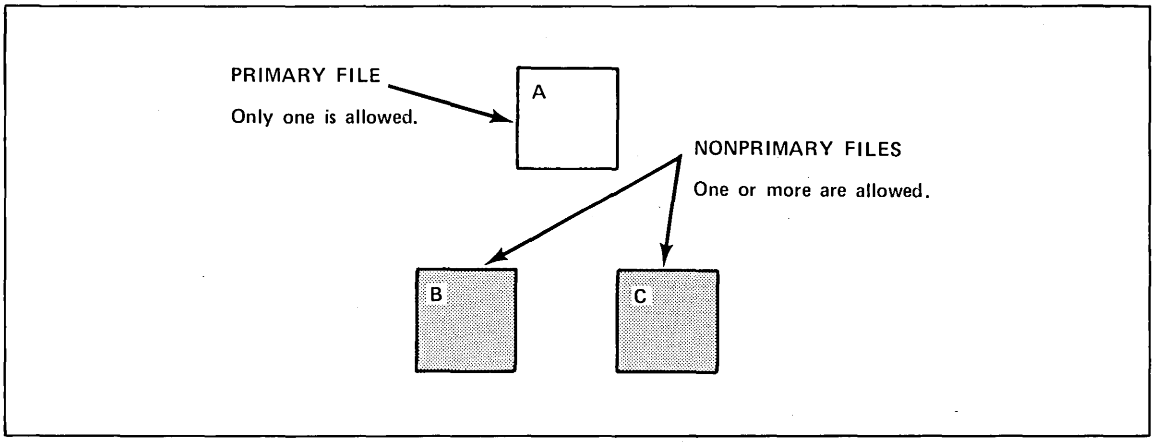


Figure 5-1. Primary and Nonprimary Files

In this section, you will also learn how to specify a particular file name on a command. Wherever you have to fill in the name of one of your files, the word filename is shown in lowercase letters. You must still enter anything shown in uppercase letters exactly as shown. For example,

```
SAVE,filename
```

means you can save a file by typing

```
SAVE,
```

followed by the name of the file.

Designating a Primary File

In the preceding section you created a primary file by specifying a programming language immediately after logging in and responding to prompts by the system. At any time during a terminal session you can also create a new, empty file and designate it the primary file simply by typing

```
NEW
```

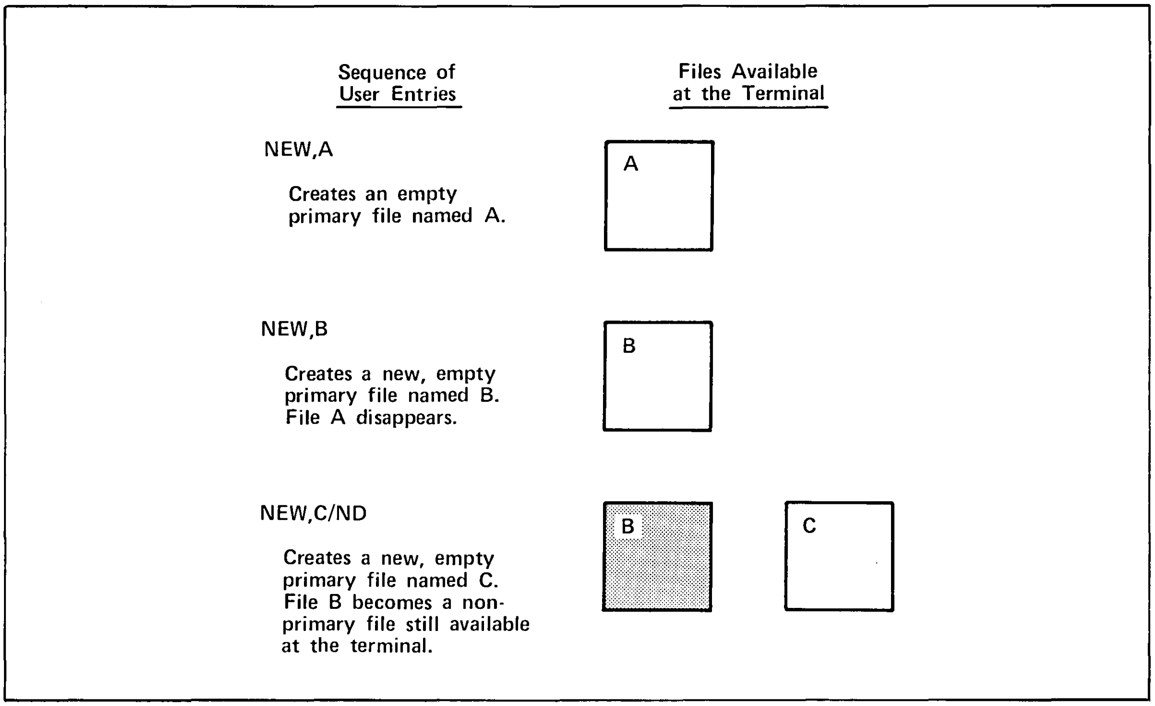
The system then asks you for the new file's name as before. If you want to combine these two steps, you can include the file name with the NEW command like this:

```
NEW,filename
```

All of the files that were available at your terminal, including the old primary file, disappear (files in permanent storage are not affected). If you want to keep them available, include /ND (for no drop) in the command like this:

```
NEW,filename/ND
```

The old primary file then becomes a nonprimary file available at the terminal. Figure 5-2 illustrates the NEW command. Figure 5-3 shows it in an example.



Sequence of User Entries

Files Available at the Terminal

NEW,A

Creates an empty primary file named A.

NEW,B

Creates a new, empty primary file named B. File A disappears.

NEW,C/ND

Creates a new, empty primary file named C. File B becomes a nonprimary file still available at the terminal.

Figure 5-2. Creating a New, Empty File and Designating It the Primary File (NEW Command)

Terminal Session	Comments
fortran READY. new,octfile READY.	The user creates a new, empty primary file named OCTFILE.
10 program octal 20 read 100,j 30 100 format(i4) 40 print 200,j,j 50 200 format(2x,i4,' decimal = ',o10,' octal') 60 end	The user enters a FORTRAN program that converts a decimal number to octal. In this example, many spaces are typed between the line number and the beginning of the FORTRAN statement, although only one space is required. The additional spaces allow FORTRAN statement numbers to be easily identified.
run ? 8 8 DECIMAL = 0000000010 OCTAL SRU 0.314 UNTS. RUN COMPLETE.	The program is run successfully.
save	The user saves file OCTFILE.

Figure 5-3. Example of Creating and Saving a Permanent File

You use NEW only when creating a new, empty file. To designate a file already existing at your terminal as the new primary file, type

```
PRIMARY,filename
```

filename is the name of the file you are designating as primary. The old primary file is changed to a nonprimary file and remains available at the terminal. Other files at the terminal are not affected by this command.

Figure 5-4 illustrates the PRIMARY command.

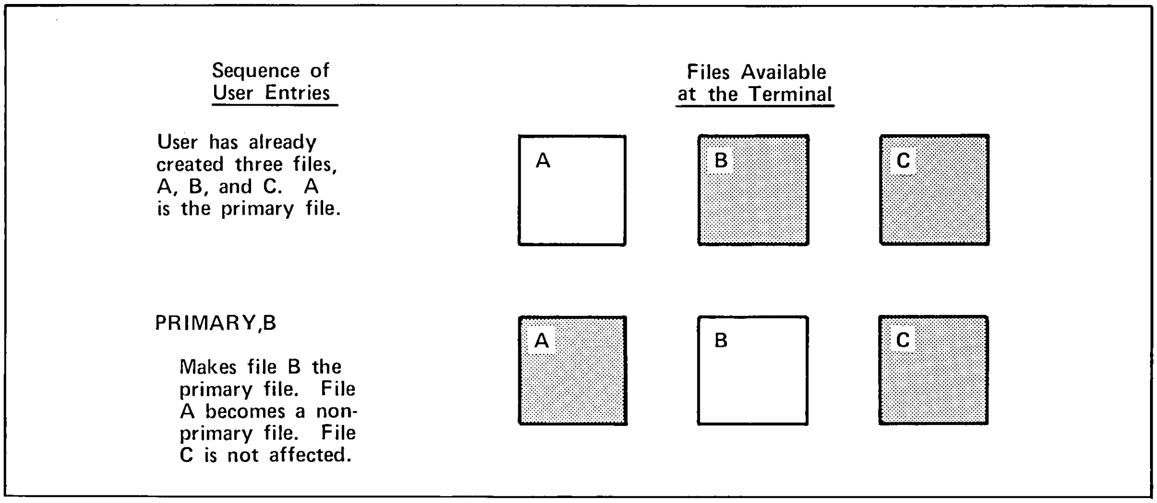

Sequence of User Entries
Files Available at the Terminal
User has already created three files, A, B, and C. A is the primary file.
A
B
C
PRIMARY,B
Makes file B the primary file. File A becomes a non-primary file. File C is not affected.
A
B
C

Figure 5-4. Designating an Existing File as the Primary File (PRIMARY Command)

Entering Information into the Primary File

You can enter information into the primary file either directly (as we described in the preceding section) or by using the TEXT command. Previously, you entered lines with line numbers. The TEXT command allows you to enter information without line numbers. Typically, it is used to create files containing data or text rather than a program.

First, create a new, empty primary file with the NEW command. Then type

```
TEXT
```

When the system responds

```
ENTER TEXT MODE.
```

enter the information without line numbers, ending each line with a carriage return.

If you see a mistake on a line before you've pressed the carriage return, either you can backspace over the mistake and reenter the line from that point, or you can delete the line by holding down the control key (labeled CONTROL, CTRL, or something similar) and pressing the X key, then releasing them and pressing the carriage return. (This is called the cancel line sequence. If this sequence doesn't work on your terminal, ask someone what the equivalent key or keys are. You may want to record this sequence on the information sheet in section 1.)

In text mode, once you have entered a line and pressed the carriage return, you can't change the line unless you use one of the system's editing facilities, such as the Full Screen Editor. If you're interested, appendix C describes some basic Full Screen Editor commands. If the Full Screen Editor is not available on your system, try one of the other editors, XEDIT or Text Editor. You can find out how to use XEDIT, Text Editor, and more about the Full Screen Editor from the manuals listed in the preface.

When you are finished entering information and are at the beginning of a new line, press either the ETX key, the interruption sequence (usually the control key and the P key), or the termination sequence (usually the control key and the T key). Then press the carriage return. The system responds:

```
EXIT TEXT MODE.
```

To add more lines, type

```
TEXT
```

and repeat the procedure. The new lines are added to the end of the file. Figure 5-5 shows a user creating a file that contains data to be used in a program.

Terminal Session	**Comments**

`new,data`
` READY.`
`text`
` ENTER TEXT MODE.`
	The user creates an empty primary file named DATA.

`3 4 5`
`2 1 5`
`1.5197 1.5197 1.5197`
`0 0 0`
	The user enters four lines of data.

` EXIT TEXT MODE.`
` READY.`
	The user exits text mode using either the ETX key, interruption sequence, or termination sequence.

`list`
`3 4 5`
`2 1 5`
`1.5197 1.5197 1.5197`
`0 0 0`
	If the system had not been in text mode when these numbers were entered, it would have assumed the numbers were line numbers and resequenced them.

Figure 5-5. Example of Creating a Data File

When you're entering text into a primary file in text mode, you might want to use uppercase and lowercase letters. On most terminals, your entry appears as lowercase letters but the system automatically converts it to uppercase letters. To tell the system not to convert lowercase letters to uppercase letters, type

```
ASCII
```

before entering text mode. When you are done using text mode and want to return to the normal mode of operation, type

```
NORMAL
```

The system then begins converting lowercase letters to uppercase again. You shouldn't mix ASCII and normal mode when entering data into a file. Lowercase letters entered under ASCII mode are translated incorrectly if they are listed under normal mode.

Figure 5-6 shows a user entering uppercase and lowercase letters in ASCII mode.

```
Terminal Session                          Comments

new,exam1                                 The user creates an empty primary file.
 READY.
text
 ENTER TEXT MODE.

this is an example of lowercase           The user enters text in lowercase letters.
letters entered in normal mode.

                                          The user exits text mode.
 EXIT TEXT MODE.

 READY.
list
THIS IS AN EXAMPLE OF LOWERCASE           The system converts lowercase letters to
LETTERS ENTERED IN NORMAL MODE.           uppercase letters.

 READY.
new,exam2                                 The user creates a new primary file.
 READY.
ascii
 READY.
text
 ENTER TEXT MODE.

This shows entry of lowercase             The user enters text in uppercase and
letters and UPPERCASE LETTERS             lowercase letters.
in ASCII mode.


 EXIT TEXT MODE.

 READY.
list
This shows entry of lowercase             The system does not convert lowercase
letters and UPPERCASE LETTERS             letters to uppercase.
in ASCII mode.

 READY.
normal
```

Figure 5-6. Example of Using ASCII Mode

Saving a File

As we said in the preceding section, to save a primary file, just type

```
SAVE
```

To save a nonprimary file, include the file name like this:

```
SAVE,filename
```

Figure 5-7 illustrates the SAVE command. Figure 5-3 shows it in an example.

If you already have a permanent file with that name, the system tells you

```
filename ALREADY PERMANENT
```

To replace the file in permanent storage with a copy of your primary file, type

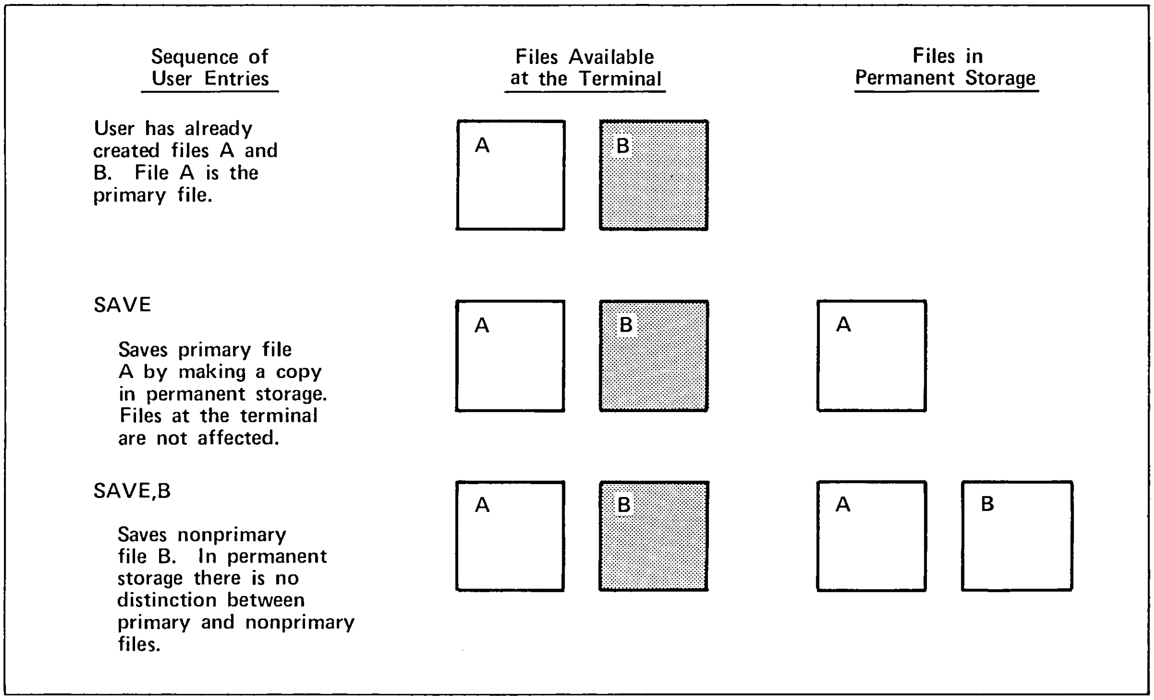
```
REPLACE
```

To replace a file in permanent storage with a copy of a nonprimary file, include the file name like this:

```
REPLACE,filename
```

Figure 5-8 illustrates the REPLACE command. Figure 5-9 shows it in an example.

The language you are using when you save or replace a file is automatically associated with the file when you retrieve it. You don't need to enter the name of the language.



User has already created files A and B. File A is the primary file.

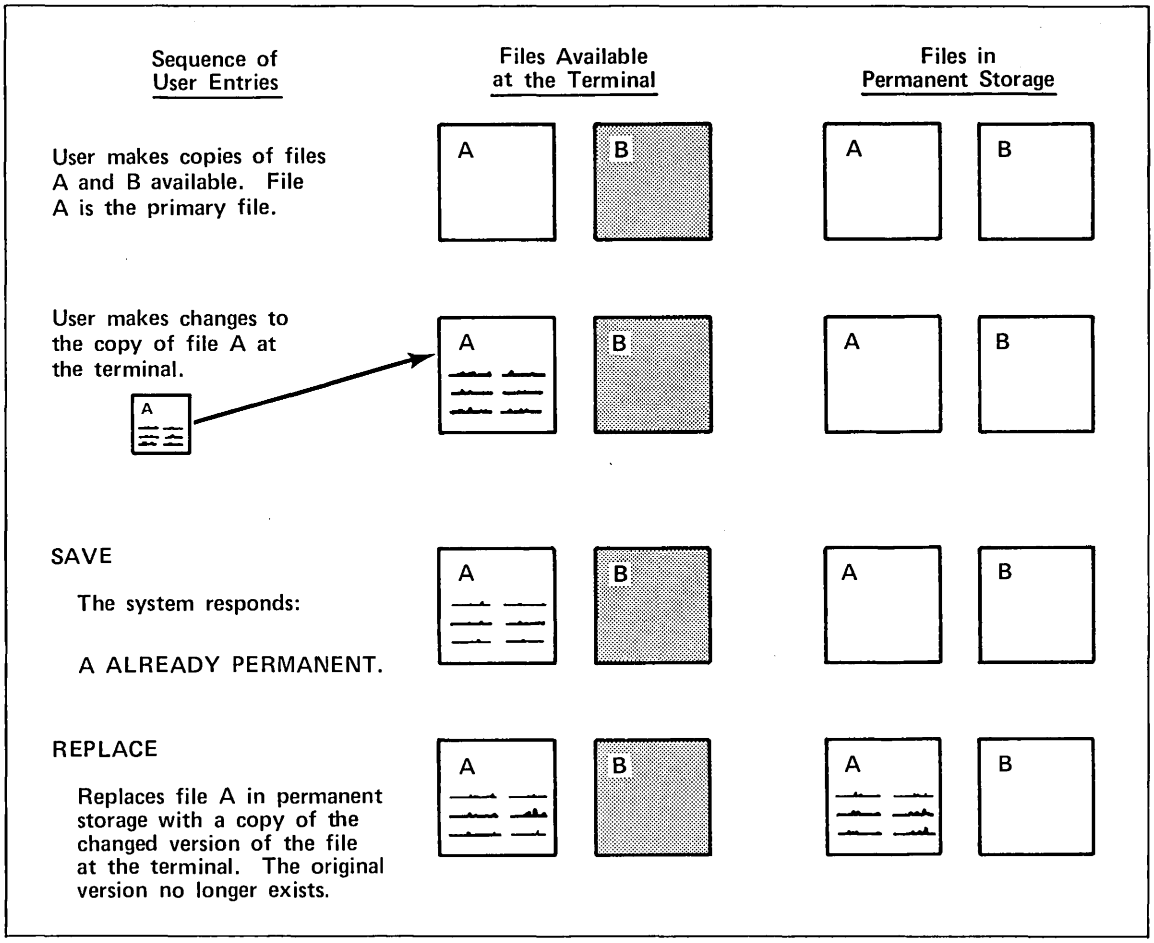
SAVE

Saves primary file A by making a copy in permanent storage. Files at the terminal are not affected.

SAVE,B

Saves nonprimary file B. In permanent storage there is no distinction between primary and nonprimary files.

Figure 5-7. Saving Files (SAVE Command)



User makes copies of files A and B available. File A is the primary file.

User makes changes to the copy of file A at the terminal.

SAVE

The system responds:

A ALREADY PERMANENT.

REPLACE

Replaces file A in permanent storage with a copy of the changed version of the file at the terminal. The original version no longer exists.

Figure 5-8. Replacing Files in Permanent Storage with Changed Copies (REPLACE Command)

Terminal Session | **Comments**

```
old,octfile
 READY.
list
10         PROGRAM OCTAL
20         READ 100,J
30  100    FORMAT(I4)
40         PRINT 200,J,J
50  200    FORMAT(2X,I4,' DECIMAL = ',O10,' OCTAL')
60         END
```

The user retrieves a copy of the saved file OCTFILE as the primary file and lists it.

```
 READY.
13         print 50
16  50     format('enter a decimal number')
```

The user changes the temporary copy of the file at the terminal.

```
run
ENTER A DECIMAL NUMBER
? 16
    16 DECIMAL = 0000000020 OCTAL

SRU        0.314 UNTS.

 RUN COMPLETE.
```

The program runs successfully with the changes just entered.

```
replace
 READY.
```

The user replaces OCTFILE in permanent storage with the copy of OCTFILE that includes the changes.

Figure 5-9. Example of Retrieving and Replacing a Copy of a Permanent File

Retrieving a Copy of a Permanent File for Use as a Primary File

To retrieve a copy of a permanent file and make it the primary file, type

```
OLD
```

The system asks for the name of the file. To combine these two steps, include the file name in the command like this:

```
OLD,filename
```

If you already have a primary file, it and all other files currently at your terminal are no longer available to you at the terminal. To keep them at the terminal, include /ND (for no drop) in the command like this:

```
OLD,filename/ND
```

The old primary file (if there was one) becomes a nonprimary file and remains available, along with the other files.

If the system displays

```
filename NOT FOUND.
```

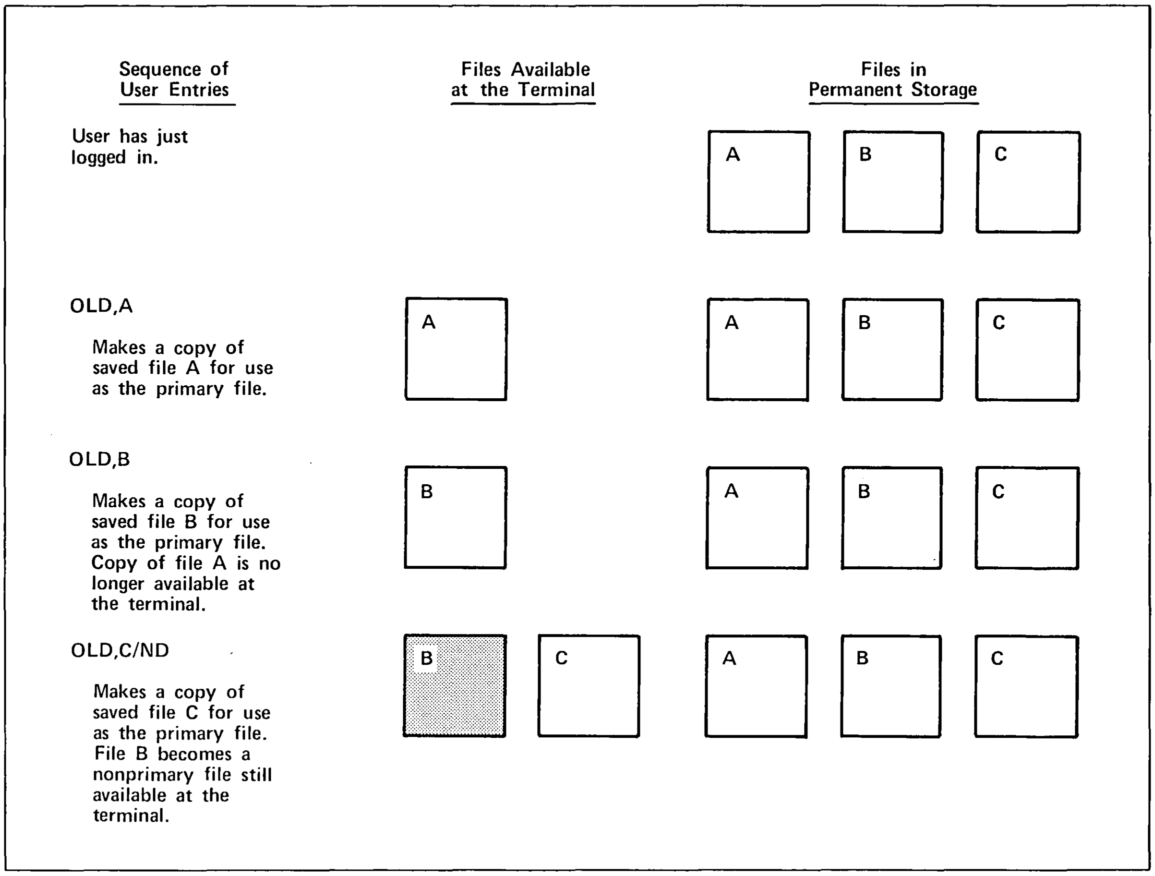
you have no permanent file with that name. Check your spelling of the file name. If you're not sure whether the name is correct, type

```
CATLIST
```

This command gives you a list of all your permanent files. It's possible you saved the file using a slightly different name than you remember.

The language you were using when you saved the file is associated with the file automatically when you retrieve it. You don't need to enter the name of the language.

Figure 5-10 illustrates the OLD command. Figure 5-9 shows it in an example.



Sequence of User Entries

Files Available at the Terminal

Files in Permanent Storage

User has just logged in.

OLD,A

Makes a copy of saved file A for use as the primary file.

OLD,B

Makes a copy of saved file B for use as the primary file. Copy of file A is no longer available at the terminal.

OLD,C/ND

Makes a copy of saved file C for use as the primary file. File B becomes a nonprimary file still available at the terminal.

Figure 5-10. Retrieving Copies of Permanent Files as Primary Files (OLD Command)

Retrieving a Copy of a Permanent File for Use as a Nonprimary File

To retrieve a copy of a permanent file for use as a nonprimary file, type

```
GET,
```

and the name of the file like this:

```
GET,filename
```

Remember, since it is not a primary file, you must specify the file name when you want a command to operate on a nonprimary file.

To list the contents of the file at the terminal, type

```
LIST,F=filename
```

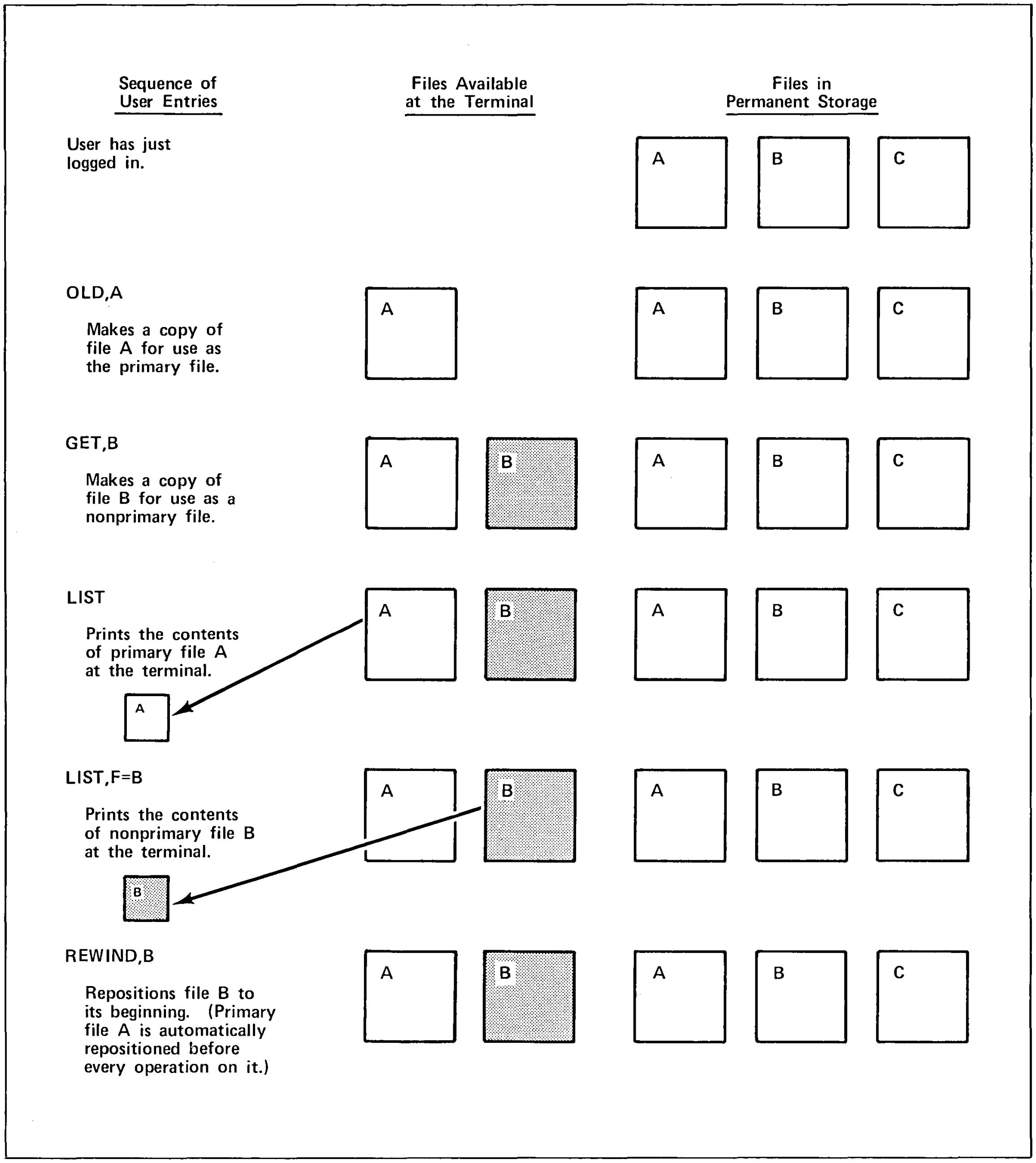
Nonprimary files are not repositioned before every operation as is the primary file. If you try to perform an operation using a nonprimary file and it doesn't work correctly, the file may be positioned at its end.

```
REWIND,filename
```

and try the command again.

You can't edit a nonprimary file simply by entering the changes with line numbers. The system would assume that you wanted the changes made to your primary file. To edit a nonprimary file, you must use one of the text editors available on your system. Appendix C shows you how to edit a nonprimary file using Full Screen Editor.

Figure 5-11 illustrates the GET, LIST, and REWIND commands.



User has just logged in.

OLD,A

Makes a copy of file A for use as the primary file.

GET,B

Makes a copy of file B for use as a nonprimary file.

LIST

Prints the contents of primary file A at the terminal.

LIST,F=B

Prints the contents of nonprimary file B at the terminal.

REWIND,B

Repositions file B to its beginning. (Primary file A is automatically repositioned before every operation on it.)

Figure 5-11. GET, LIST, and REWIND Commands

Changing a Permanent File

You can change a permanent file that has line numbers the same way you changed a program in the preceding section. First you must make it the primary file, using either the OLD or PRIMARY command. Then follow the same directions for line replacement, deletion, and insertion given in the preceding section.

You can change a nonprimary file either by making it the primary file and using the methods we described in the preceding section or by using one of the text editors available on your system. Appendix C describes some basic Full Screen Editor directives.

Examples of permanent file changes are shown in figures 5-9 and 5-12. Remember, the system changes only the copy at your terminal. To save the changes, you must either replace the file in permanent storage with your changed copy or, if you want to save the changes and also keep the original version of the file, save the changed copy with a new file name.

```
SAVE,oldname=newname
```

Then the original version remains as it was and the changed version of the file is saved. Figure 5-13 illustrates this variation of the SAVE command. Figure 5-12 shows it in an example.

Terminal Session **Comments**

```
old,octfile
 READY.
list
10       PROGRAM OCTAL
13       PRINT 50
16 50    FORMAT('ENTER A DECIMAL NUMBER')
20       READ 100,J
30 100   FORMAT(I4)
40       PRINT 200,J,J
50 200   FORMAT(2X,I4,' DECIMAL = ',O10,' OCTAL')
60       END


 READY.
```

The user retrieves a copy of the saved file OCTFILE as the primary file and lists it.

```
10       program hex
50 200   format(2x,i4,' decimal = ',z10,' hexadecimal')
```

The user changes the copy.

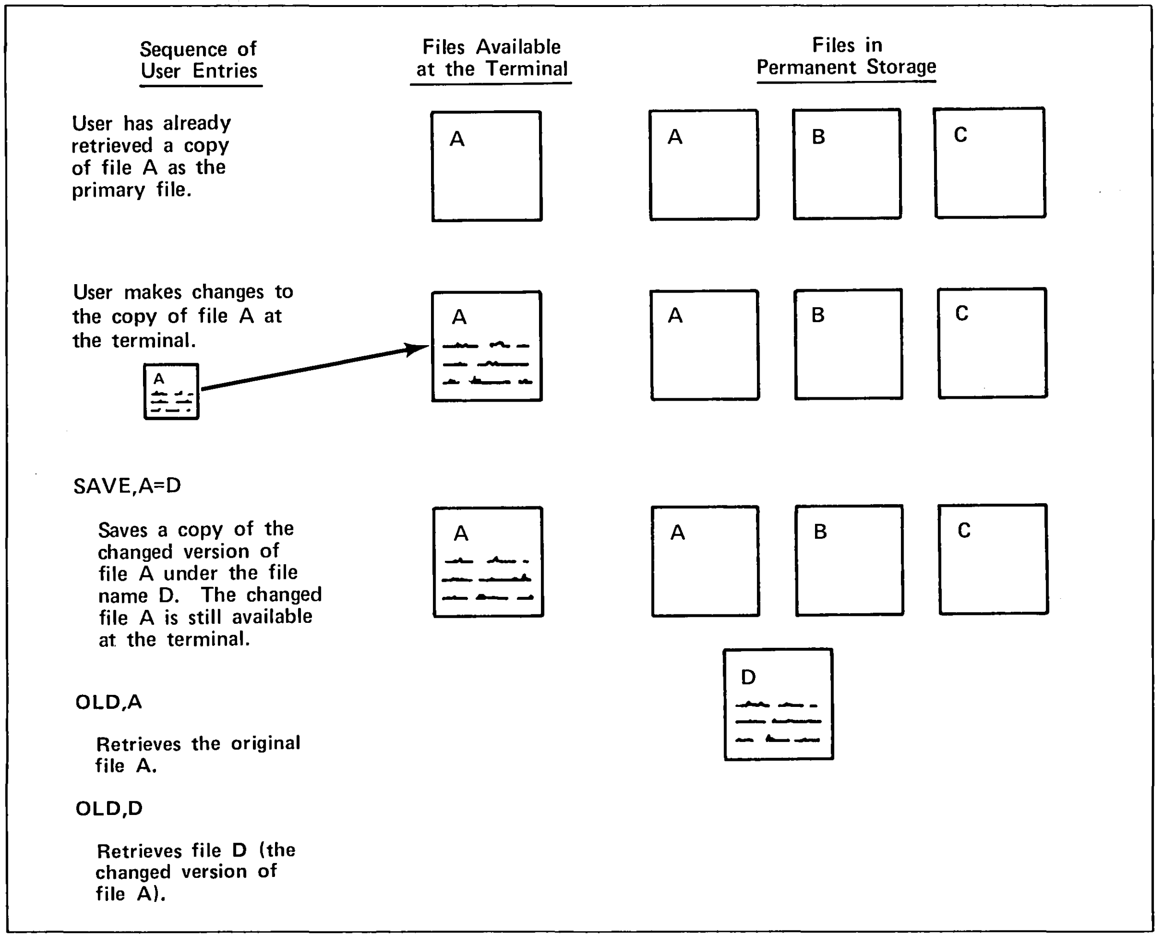
```
run
ENTER A DECIMAL NUMBER
? 16
    16 DECIMAL = 0000000010 HEXADECIMAL

SRU      0.313 UNTS.

 RUN COMPLETE.
```

The program runs successfully with the changes just made. (The number entered is converted to hexadecimal rather than octal.)

```
save,octfile=hexfile
 READY.
```

The user saves the changed version of OCTFILE as a new file named HEXFILE. Both the original file and the modified file, OCTFILE and HEXFILE, exist in permanent storage.

Figure 5-12. Saving a File with Changes Under a New Name



User has already retrieved a copy of file A as the primary file.

User makes changes to the copy of file A at the terminal.

SAVE,A=D

Saves a copy of the changed version of file A under the file name D. The changed file A is still available at the terminal.

OLD,A

Retrieves the original file A.

OLD,D

Retrieves file D (the changed version of file A).

Figure 5-13. Saving Both the Original and Changed Versions of a File (SAVE Command)

Listing and Rewinding Files

The commands to list and rewind files have already been mentioned briefly. In summary:

- To list the contents of a primary file, type

```
LIST
```

- To list the contents of a nonprimary file, type

```
LIST,F=filename
```

- To rewind a nonprimary file, type

```
REWIND,filename
```

Remember to rewind nonprimary files, particularly a file containing data to be used in program execution (refer to section 6). If you run the program once and then run it a second time without rewinding the data file (which is nonprimary), the program won't produce the correct results because it has no data (that is, because the system pointer is at the end of the nonprimary file).

Listing the Names of Your Permanent Files

To get a list of your permanent files, type

```
CATLIST
```

and the system displays the names of your files similar to this.

```
catlist
 CATALOG OF  JLC6021          FM/SYSTEMA 82/03/19. 16.57.31.

 INDIRECT ACCESS FILES

  BASTEST   DATA      FORTEST   HEXFILE   OCTFILE

         5 INDIRECT ACCESS FILES,     TOTAL PRUS =        5.
```

The files you've saved are called indirect access files, because you access a copy of the file, not the file itself. The TOTAL PRUS entry at the bottom refers to the size of your files. It's unlikely that you need be concerned with this entry.

Removing a Permanent File from the System

To remove the permanent file associated with your primary file, type

```
PURGE
```

To remove other permanent files, type

```
PURGE,filename
```

If you were using a copy of the file at your terminal, it is still available, but it disappears when you log out. Figure 5-14 illustrates the PURGE command.

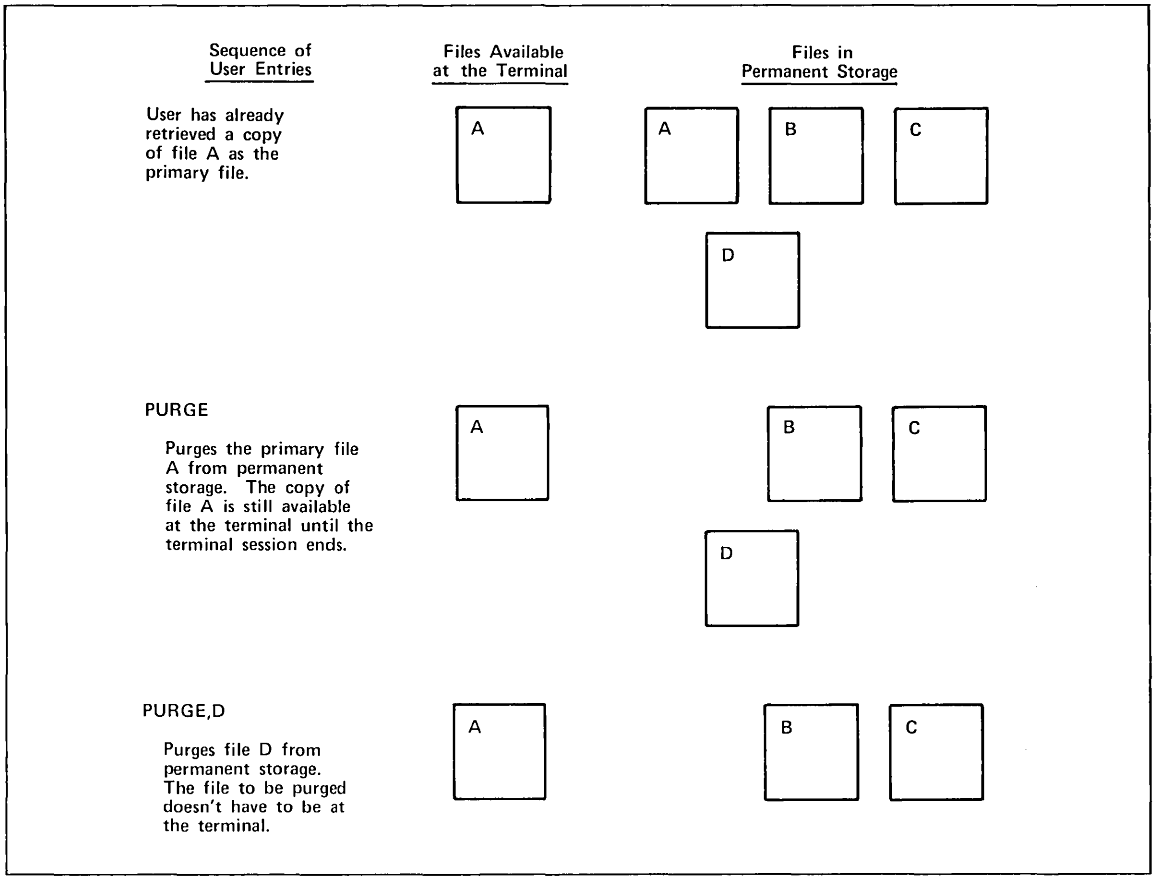

Sequence of User Entries
Files Available at the Terminal
Files in Permanent Storage
User has already retrieved a copy of file A as the primary file.
A
A
B
C
D
PURGE
Purges the primary file A from permanent storage. The copy of file A is still available at the terminal until the terminal session ends.
A
B
C
D
PURGE,D
Purges file D from permanent storage. The file to be purged doesn't have to be at the terminal.
A
B
C

Figure 5-14. Purging Files in Permanent Storage (PURGE Command)

Connecting Files to Your Programs 6

So far in this manual, the system has requested data for program execution (input data) by displaying a question mark at the terminal and waiting for you to enter the data from the keyboard. The system then displayed the results of program execution (output data) at the terminal.

However, if you have a lot of data to enter, you can put it on a file and instruct the system to get the data from the file (that is, to read the file). In the same way, if you expect a lot of data to be displayed as the result of program execution, you can create a file and instruct the system to enter the results in the file (that is, to write the file).

Reading and Writing Data Files in FORTRAN

In order for the system to read or write a file used within a FORTRAN program, you must first associate a number with the file using the FORTRAN OPEN statement.

```
OPEN (UNIT=number,FILE='filename')
```

The number is a number you choose between 1 and 99. In the FORTRAN program, this number is associated with the file that you name following FILE=.

The file name follows the rules you've used previously to create file names. It can be both letters and numbers, must begin with a letter, and must be no more than seven characters. You must include an apostophe (') before and after the file name.

For example, in order for the system to read and write files named DATA and RESULT, you could use the OPEN statements:

```
OPEN (UNIT=76,FILE='DATA')

OPEN (UNIT=3,FILE='RESULT')
```

After associating a number with the file, you must specify that number in a FORTRAN READ or WRITE statement like this:

```
READ(number,format)variables

WRITE(number,format)variables
```

In both these statements, number is the number associated with the file in the OPEN statement, format is the statement number of the appropriate FORMAT statement, and variables is a list of the FORTRAN variables you want to read or write.

To read variables A, B, and C from file DATA, you could use a READ statement like this:

```
READ(76,100)A,B,C
```

To write A, B, and C on file RESULT, you could use a WRITE statement like this:

```
WRITE(3,200)A,B,C
```

If file RESULT does not exist when the WRITE statement is executed, the system creates it.

Figure 6-1 shows how to connect a file to a FORTRAN program.

Terminal Session **Comments**

```
old,forarea
 READY.
list
10         PROGRAM TRY
20         OPEN (UNIT=3,FILE='DATA')
30 1       READ (3,100) A, B, C
40 100     FORMAT (3F7.0)
50         IF (A .EQ. 0.0) STOP
60         S = (A + B + C) / 2.0
70         RDCL=S * (S-A) * (S-B) * (S-C)
80         IF (RDCL .LT. 0.0) GO TO 2
90         AREA = SQRT(RDCL)
100        PRINT 200, A, B, C, AREA
110 200    FORMAT ('SIDES ARE ',3F9.4,/,'AREA IS ',F11.4/)
120        GO TO 1
130 2      CONTINUE
140        PRINT 300, A, B, C
150 300    FORMAT ('SIDES OF ',3F9.4,' FORM AN INVALID TRIANGLE',/)
160        GO TO 1
170        END

 READY.
```

The user retrieves a program created earlier and lists it. The program reads three numbers, which are sides of a triangle, and calculates the area of the triangle.

```
get,data
 READY.
list,f=data
3.0000 4.0000 5.0000
2.0000 1.0000 5.0000
1.5197 1.5197 1.5197
0 0 0

 READY.
```

The user now retrieves a file that was created earlier in text mode and lists it. It contains data for three triangles and, to terminate execution, a zero line.

```
run
OEND-OF-FILE ENCOUNTERED ON FILE DATA
 FTN - FATAL ERROR NUMBER 65
 TRACEBACK INITIATED BY SYSERR AT REL(ABS) ADDRESS 122(15737).
 CALLED BY FECESE. AT ADDRESS 304(16073) WITH NO AP-LIST.
 CALLED BY INPC= AT ADDRESS 227(5226) WITH NO AP-LIST.
 CALLED BY TRY AT LINE 20, ADDRESS 22(133) WITH APLIST 11631(11742).
       .014 CP SECONDS EXECUTION TIME.

SRU       0.352 UNTS.

 RUN COMPLETE.
```

This is included to show an error. The RUN command causes the program to be compiled and executed. The program runs but no results appear, because the user did not rewind the file after listing it.

```
rewind,data
 READY.
run
SIDES ARE        3.0000    4.0000    5.0000
AREA IS          6.0000
SIDES OF        2.0000   1.0000    5.0000 FORM AN INVALID TRIANGLE
SIDES ARE        1.5197    1.5197    1.5197
AREA IS          1.0000

SRU      0.349 UNTS.

 RUN COMPLETE.
```

The user rewinds the file, and the program is executed again. This time the results appear.

Figure 6-1. Connecting a File to a FORTRAN Program

Reading and Writing Data Files in BASIC

In order for the system to read or write a file used within a BASIC program, you must first associate a number with the file, using one or more FILE statements in the BASIC program.

```
FILE #number="filename"
```

You must include the number sign (#) before the number, the equal sign (=) before the file name, and quotation marks (") before and after the file name. The number can be between 1 and 131071. The file name can be both letters and numbers, must begin with a letter, and must be no more than seven characters. To specify more than one file, you can either put them on one FILE statement separated by commas or put one file each on several FILE statements.

For example, to allow the system to read and write files named DATA and RESULT, you could use:

```
FILE #76="DATA",#3="RESULT"
```

or

```
FILE #76="DATA"

FILE #3="RESULT"
```

After associating a number with the file in a FILE statement, you must specify that number in a BASIC INPUT (read) or PRINT (write) statement like this:

```
INPUT #number,variables

PRINT #number,variables
```

or

```
PRINT #number USING linenumber variables
```

In all forms of the statements, variables is a list of the BASIC variables you want to read or write. In the PRINT USING statement, linenumber is the line number of the image line to follow when printing.

To read variables A, B, and C from file DATA, you could use an INPUT statement like this:

```
INPUT #76,A,B,C
```

To write A, B, and C on file RESULT, you could use either:

```
PRINT #3,A,B,C
```

or

```
PRINT #3 USING 180,A,B,C
```

If file RESULT does not exist when the PRINT statement is executed, the system creates it.

Figure 6-2 shows how to connect a file to a BASIC program.

Terminal Session **Comments**

```
old,basarea
 READY.
list
10  FILE #3 = "DATA"
20  INPUT #3, A, B, C
30  IF A = 0 THEN 140
40  S = (A + B + C) / 2
50  R = S * (S-A) * (S-B) * (S-C)
60  IF R < 0 THEN 110
70  X = SQR(R)
80  PRINT "SIDES ARE ",A, B, C
90  PRINT "AREA IS ",X
100 GOTO 20
110 PRINT "SIDES ARE ",A, B, C
120 PRINT "INVALID TRIANGLE"
130 GOTO 20
140 END

 READY.
```

The user retrieves a program created earlier and lists it. The program reads three numbers, which are sides of a triangle, and calculates the area of the triangle.

```
get,data
 READY.
list,f=data
3 4 5
2 1 5
1.5197 1.5197 1.5197
0 0 0

 READY.
```

The user now retrieves a file that was created earlier in text mode and lists it. It contains data for three triangles and, to terminate execution, a zero line.

```
run
 END OF DATA ON FILE  AT  20
BASIC EXECUTION ERROR

  RUN COMPLETE.
```

This is included to show an error. The RUN command causes the program to be compiled and executed. The program runs but no results appear, because the user did not rewind the file after listing it.

```
rewind,data
 READY.
run
SIDES ARE       3         4         5
AREA IS         6
SIDES ARE       2         1         5
INVALID TRIANGLE
SIDES ARE       1.5197    1.5197    1.5197
AREA IS         1.00004

 RUN COMPLETE.
```

The user rewinds the file, and the program is executed again.

This time the results appear.

Figure 6-2. Connecting a File to a BASIC Program

What To Do When You Have Problems 7

This section describes some of the more common problems that users have and suggests some possible solutions. If you have difficulty after reading this, ask your employer, instructor, or computer center personnel for help.

Setting Up Your Terminal

If you're having problems with your terminal and aren't sure of the proper switch settings, try these. They are the most common.

- Duplex switch set to HALF. If you enter data but nothing is printed, check the duplex switch on your terminal; it is probably set incorrectly.
- Parity switch set to EVEN.
- Line-speed switch set to match the speed of the phone line you plan to use. If you're not sure of the speed you should use, set the switch to any position (usually 10 to 30 characters per second). The system may be able to detect automatically what line speed you are using.
- The line/local switch (sometimes called the mode switch) set to the line (online) position.

Make sure the terminal is plugged in and turned on. If a separate acoustic coupler is used, it too must be turned on. For any type of acoustic coupler, check that the phone receiver is inserted with the cord facing the correct direction.

During the Dialing Process

If you dial the phone number and it rings unanswered or is busy, either the system may not be operating, or the maximum number of users may be using that phone line already. Try other numbers if you have them. If you still get no response (response is a constant, high-pitched tone), try again later.

During the Login Process

If you hear the high-pitched tone, connect your terminal, and press the carriage return twice, but don't get a login message, go back to *Setting Up Your Terminal* in this section and check the switch settings mentioned. If the switches are all set correctly and the system still doesn't print anything at your terminal, tell your computer center personnel. It may be a problem with the equipment.

If the login sequence has started and the system displays the message

```
IMPROPER LOGIN, TRY AGAIN.
```

you've entered some of your login information incorrectly. The system prompts you for the information again and you can retry the procedure. If this happens four times, the system assumes you are not an authorized user; it displays

```
USER RETRY LIMIT.
LOGIN TERMINATED.
```

and disconnects the terminal. If this happens, tell your computer center personnel. They will help you with the problem.

If you get the message

```
ILLEGAL APPLICATION, TRY AGAIN.
```

and the system prompts you for an application, enter

```
IAF
```

If you get the same message, or the message

```
APPLICATION RETRY LIMIT.
LOGGED OUT.
```

tell your computer center personnel.

If you get the message

```
APPLICATION NOT PRESENT.
```

or

```
CONNECTION REJECTED.
```

log out and try again later. You haven't made any mistakes; the system is just not available at this time. If you continue to get the

```
CONNECTION REJECTED
```

message, tell your computer center personnel.

If you get the message

```
TIMEOUT.
```

you waited too long to respond to a request from the system. Usually the time limit during the login procedure is 2 minutes. If this happens, connect your terminal again and retry the login sequence.

During the Terminal Session

When the system can't understand what you want done, or when it detects an error of any sort, it tries to diagnose the problem and to give you a message that explains the error and tells you what to do. At any time during your terminal session, you may get messages such as

```
INCORRECT COMMAND
```

that indicate you've entered information that the system doesn't recognize. If this happens, check your last entry to make sure that everything is spelled correctly and that commas are included where required. Remember that the letter O, the numeral 0, the letter l, and the numeral 1 each has its own key on the terminal keyboard. If you type the letter O for the numeral 0, or the letter l for the numeral 1, the system regards it as an error.

If you get a system message that you don't understand, refer to Volume 3 of the NOS 2 Reference Set, appendix B, for more information.

At times, you may get messages that do not come from the system but instead, come from the programming language you are using. These messages usually appear when you try to compile or run a program that has an error in it. Most programming languages have reference manuals that explain these messages. The manual that you will need depends on the language you are using. For example, if you try to run a BASIC program and you get a message telling you that you made an error, refer to the BASIC Reference Manual for an explanation of that message.

If the terminal gives you some sort of audible signal and displays the message

```
TERMINAL TIMEOUT IN 30 SECONDS.
```

you are waiting too long to enter an instruction (command) or data. The default time limit is 10 minutes, but it may be different at your site. Check with your site administrator to be sure.

When you get this message, any entry, even a carriage return, is sufficient to keep the terminal active. If you don't enter anything within 30 seconds of the signal, the system logs you out automatically. You can log in again and try to resume your work as described in *After Accidental Disconnection from the System* later in this section.

During Program Execution

If you get the message

```
*TIME LIMIT*
ENTER T TO CONTINUE OR CR KEY TO STOP
```

your program needs more time to execute. Just follow the directions; type

```
T
```

to get more time or press the carriage return to stop the program.

If you get the message

```
*SRU LIMIT*
ENTER S TO CONTINUE OR CR KEY TO STOP
```

your program needs more system resources. Again, just follow the instructions; type

```
S
```

to continue your program execution or press the carriage return to stop the program.

If these messages continue during program execution, be sure to check your program for any errors that may be present, such as incorrect loop statements.

Any time you want the system to stop what it's doing, hold down the control key (labeled CONTROL, CTRL, or something similar) and press the T key. Then release both keys and press the carriage return. (This combination of keys is called the termination sequence. If it doesn't work on your terminal, ask someone what the equivalent key or keys are.)

After Accidental Disconnection from the System

A mechanical failure or system malfunction may disconnect you from the system unexpectedly. If this happens, log in again. The system tells you if you can recover your work by listing one or more recoverable jobs. The system refers to your terminal session as a job and gives it a job sequence name (JSN). You may have noticed a job sequence name being printed during the login process. The job sequence name printed during your previous login should match the name of the job that the system says you can now recover.

To continue your previous terminal session, type the job sequence name. To start a new session, type

```
GO
```

Figure 7-1 shows a user being disconnected, logging back in, and resuming work from the previous terminal session.

Terminal Session **Comments**

```
WECOME TO THE NOS SOFTWARE SYSTEM
COPYRIGHT CONTROL DATA 1978, yyyy.

yy/mm/dd. hh.mm.ss. T01A6D
CDC NETWORK OPERATING SYSTEM    NOS 2
FAMILY:  systema
USER NAME: jlc6021
PASSWORD: jlc6
T01A6D    - APPLICATION: iaf
JSN: ADDQ, NAMIAF
```

The user logs in.

The system assigns job sequence name ADDQ to the terminal session.

```
 READY.
old,fortest
 READY.
list
10 PROGRAM TEST
20 RE
```

The user retrieves saved file FORTEST and tries to list it.

At this point, the terminal is disconnected.

```
WECOME TO THE NOS SOFTWARE SYSTEM
COPYRIGHT CONTROL DATA 1978, yyyy.

yy/mm/dd. hh.mm.ss. T01A61
CDC NETWORK OPERATING SYSTEM    NOS 2
FAMILY:  systema
USER NAME: jlc6021
PASSWORD: jlc6
T01A61    - APPLICATION: iaf
JSN: ADDS, NAMIAF

 RECOVERABLE JOB(S)

 JSN    UJN      STATUS      TIMEOUT

 ADDQ   AIDQ     SUSPENDED   29 MIN.

ENTER  GO  TO CONTINUE CURRENT JOB,
         RELIST  TO LIST RECOVERABLE JOBS,
         OR DESIRED JSN:  addq
 JSN: ADDQ  SYSTEM: FORTRAN  SRU:     2.005  FILE NAME: FORTEST
 STATUS:
 CHARACTER SET: NORMAL   MODES: PROMPT ON
 IDLE. ENTER COMMAND.
```

The user logs back in.

The system says that job ADDQ has been suspended and can be recovered, so the user enters that name.

```
 READY.
list
10 PROGRAM TEST
20 READ *,I,J
25 I=I*J
30 PRINT *,I
50 END

 READY.
```

At this point, job ADDQ is recovered and is in an idle state. All of the files and conditions of the job are the same as they were before the terminal became disconnected.

The user lists the primary file.

Figure 7-1. Recovering a Terminal Session After Disconnection

Additional Information About the System 8

The information in this manual is sufficient for you to enter programs and to correct them, run them, and save them. In addition, you've been introduced to some important system concepts, such as that of the primary file.

It's quite possible that the commands in this book are the only ones you will ever need to do your work. But if you find your needs increasing, or if you're curious, we suggest you go on to *NOS 2 Reference Set, Volume 2, Guide to System Usage.* Volume 2 explains more of the basic concepts of the system, for both interactive and batch users, and describes more system commands.

NOS 2 Reference Set, Volume 3, System Commands, is the reference manual for all system commands discussed in this manual and in Volume 2.

You may want to check with your employer, instructor, or computer center personnel for advice on which manual they would recommend and how you can get a manual. Ordering information is also provided in the preface.

Appendixes

Glossary A

A

Acoustic Coupler

A device in which you place a telephone receiver to connect a terminal to a telephone line. The telephone line is in turn connected to the system. After dialing the system, you place the telephone receiver in the rubber cups of the acoustic coupler, ensuring that the telephone-cord end of the receiver is at the correct end (marked on the coupler). An acoustic coupler is either built into the terminal (common on portable terminals) or a separate part (common on nonportable terminals). Contrast with Data Set.

Alphanumeric Characters

The letters of the alphabet (A through Z) and the digits (0 through 9).

ASCII Mode

Mode which uses the entire ASCII character set. ASCII stands for American Standard Code for Information Interchange. It is a code that can represent 128 characters, including uppercase and lowercase letters, numbers, and special characters. In ASCII mode, you can enter uppercase and lowercase letters and the system will not translate them all to uppercase as it does in normal mode. Contrast with Normal Mode.

B

BASIC

Beginner's All-Purpose Symbolic Instruction Code. A high-level programming language used for fairly simple programs.

C

Cancel Line Sequence

The character or sequence of characters that cancels the entry of the current line. When you type a line, the system does not accept it until you press the carriage return. If you enter the cancel line sequence before you press the carriage return, the line is ignored by the system. The sequence may differ on the terminal you're using, but usually you hold down the control key and press the X key. Then release both keys and press the carriage return.

Catalog

A collection of information about the permanent files associated with a particular user name. Every user who saves permanent files has a permanent file catalog. Each time you create, modify, or purge a permanent file, the system updates your catalog accordingly. You can list the names of your permanent files and other information about them using the CATLIST command.

Command

An instruction you give to the system that causes it to perform an operation.

Control Key

A terminal key labeled CONTROL, CTRL, CNTL, CNTRL, or something similar. Many terminals require the control key to be pressed and held while another key is pressed for the cancel line sequence, interruption sequence, or termination sequence.

D

Data Set

A device, similar in appearance to a normal telephone, that is used to connect a terminal to a telephone line. The telephone line is in turn connected to the system. After dialing the system, you either set the switch on the front of the data set to DATA or press the DATA button. You then replace the telephone receiver. The connection is made without placing the receiver in a separate device. Contrast with Acoustic Coupler.

Default

A value or name supplied by the system, if you do not supply it. For example, if you enter LIST without specifying a file name, the primary file (the default) is listed.

Duplex

Mode of transmission on a communication line. You need be concerned about duplex mode only to the extent of checking that the duplex switch on your terminal is set correctly for the line being used. Although the switch is normally set to HALF, personnel at your computer center can tell you the correct setting for your terminal. If you enter data but nothing is printed, check the duplex switch on your terminal; it is probably set incorrectly.

E

Execution

The process of carrying out an instruction or a sequence of instructions (a program) by a computer.

F

Family Name

Name of the permanent file storage device or set of devices on which all of your permanent files are stored. When you request a permanent file, the system looks for it on this family (group) of devices. Usually a system has only one family of permanent file devices, but it is possible to have alternate families in the system. You may have to specify which family you are using when you log in. Your family name is given to you by your employer, instructor, or computer center personnel.

File

A collection of information referred to by a file name (from one through seven characters, either letters or numbers). You can create a file at the terminal or retrieve a file from permanent file storage for use during a terminal session.

FORTRAN

Formula Translation. A high-level programming language used for fairly complicated programs.

FSE (Full Screen Editor)

See Full Screen Editor.

Full Screen Editor

A text editor that gives you two methods of editing: line editing and screen editing. Line editing can be used on any kind of terminal and allows you to edit a line by typing a directive and specifying the appropriate line number. Screen editing can be used only on certain display terminals and allows you to move the cursor about freely on the screen and use special function keys on the keyboard to perform editing operations.

H

Hardwired Terminal

A terminal that is always connected to the system. To begin a terminal session on a hardwired terminal, all you need to do is identify the terminal by pressing the proper key(s), and the login sequence begins (refer to *Entering the System* in section 3).

I

IAF (Interactive Facility)

See Interactive Facility.

Interactive Facility

A program within NOS which provides a terminal operator with interactive (or conversational) processing capability. IAF is the application used in this manual.

Indirect Access File

The type of file described in this manual. When you access (retrieve) a file, you get a copy of the file and can make changes to the copy but not directly to the file.

Interruption Sequence

The character or sequence of characters that interrupts a system operation (for example, listing a program or running a program). It may differ on the terminal you're using, but usually you hold down the control key and press the P key. Then release both keys and press the carriage return. To resume the operation, you press the carriage return.

J

Job Sequence Name

A four-character name assigned by the system to your terminal session. It is displayed at the end of the login procedure. If the system disconnects you before you are done with your work, you can log in again, and you may be able to resume your work by specifying the job sequence name assigned to your previous terminal session.

JSN (Job Sequence Name)

See Job Sequence Name.

L

Line/Local Switch

Determines transmission of data. For normal operations, the switch should be set to the line position. When set to the local position, entries are printed at the terminal, but they are not transmitted to the system (even though the terminal is connected). This switch is also called the mode switch.

Line Speed

Rate of transmission of information between the terminal and the system measured in either characters per second or baud. (In general, the baud rate is 10 times the character-per-second rate.) Your computer center may restrict some terminals or phone lines to certain line speeds; ask personnel there what line speeds you can use.

Local File

A name used by the system to refer to a temporary file available at the terminal.

Login

Procedure during which you identify yourself to the system as an authorized user. To identify yourself, you enter certain responses to prompts from the system. These responses are given to you initially by your employer, instructor, or computer center personnel. They usually include a family name, user name, and password.

Logout

Procedure by which you end a terminal session. You type BYE.

M

Mode Switch

See Line/Local Switch.

N

Nonprimary File

A file that is available for use at the terminal but is not the primary file. Nonprimary files are often used to store the data needed to run a program or to store the data resulting from the execution of a program. Nonprimary files are not automatically rewound before a read or write operation. Contrast with Primary File.

Normal Mode

Mode in which the system translates any lowercase letters that you enter to uppercase letters. Contrast with ASCII Mode.

P

Parity

Mode indicating a method of error detection. You need be concerned about parity mode only to the extent of checking that the parity switch on your terminal is set correctly. Personnel at your computer center can tell you the correct setting for your terminal. Parity is either even, odd, or disabled (not used). When used, a special bit (the parity bit) is set or cleared in the string of binary digits, so that the sum of the digits set, including the parity bit, is always even or always odd. When data is read, the sum is checked and, if it doesn't agree with the parity setting (even or odd), the system knows the data is incorrect.

Password

A name or word you enter during login to provide extra security for your user name. A unique password ensures that no one else can log into the system with your user name and access your files. Your password is given to you by your employer, instructor, or computer center personnel.

Permanent File

A file you create at the terminal and then save, so that you can retrieve it at a later terminal session. When you retrieve the file, the system makes a copy of the file for use at the terminal. The original file remains unchanged, even if you make changes to the copy at your terminal. If you want to save the changed copy of the file in place of the original file, you enter the REPLACE command.

Primary File

The file on which operations are performed if no other file name is specified. You designate the primary file with the NEW, OLD, or PRIMARY command. The system automatically rewinds the primary file to its beginning before every operation performed on it.

Program

A sequence of instructions that are executed by a computer to perform a task. A program can be composed of several subprograms.

PRU

Physical record unit. It is a unit of size that is applied to files.

R

Run

See Execution.

S

SRU (System Resource Unit)

See System Resource Unit.

System

Software and hardware that controls the execution of computer programs and provides scheduling, error detection, input/output control, accounting, compilation, storage assignment, and related services.

System Resource Unit

A unit measuring system use that the system displays following certain operations and your logout.

T

Temporary File

A file at the terminal that disappears when you either log out or enter the NEW or OLD command without /ND. You create a temporary file either by entering information in a primary file, retrieving a permanent file (thus making a temporary copy at your terminal), or as a result of program execution.

Terminal Session

Period between the time you physically connect the terminal to the system in preparation for login to the time you log out.

Termination Sequence

The character or sequence of characters that terminates a system operation that is currently in progress (for example, listing a program or running a program). It may differ on the terminal you're using, but usually you hold down the control key and press the T key. Then release both keys and press the carriage return.

Text Mode

Mode of entering information into a primary file without specifying line numbers. It is usually used to create data files. If a file does not contain line numbers, you can change lines only by using a text editor, such as Full Screen Editor.

U

User Name

Name given to you by your employer, instructor, or computer center personnel. Your user name has certain resources and privileges assigned to it. When you log in, you specify your user name to identify yourself to the system, so that it knows that you are an authorized user and what resources you are entitled to use. Your user name also represents a specific catalog in the permanent file system. All files you make permanent are associated with your user name and this catalog.

Additional Information About Terminals B

This manual has assumed that you have a standard ASCII terminal. To start the login process on that kind of terminal, you press the carriage return twice (described in section 3). To start the login procedure on any other kind of terminal, follow these procedures.

- On mode 4A terminals, press the SEND key. (On these terminals, the SEND key takes the place of the carriage return.)
- On mode 4C terminals, press ETX. (On these terminals, the ETX key takes the place of the carriage return.)
- On all other terminals, press the carriage return. Wait until the system responds with two blank lines, then type a right parenthesis

  ```
  )
  ```

  followed by a carriage return.

Using the Full Screen Editor C

The Full Screen Editor has been designed to take advantage of the features found on certain display screen terminals. For example, the Full Screen Editor can fill a screen with lines of a file and allow you to edit anywhere on the screen by moving the cursor to a particular position. Then you can either type directly over existing characters or, by pressing a single key, perform an editing operation at that position. The Full Screen Editor also has directives that can be used on any terminal to edit line by line.

The Full Screen Editor has two ways of editing: line editing and screen editing. As you may have guessed, line editing allows you to edit on a line by line basis using line numbers. You type in a directive, specifying the number of the line you want to be affected by the directive. You can perform line editing on any terminal, whether it's a display terminal or a printing terminal.

Screen editing allows you to edit by moving the cursor to a position within a displayed file and either entering the characters directly or pressing a key that performs the operation. You can perform screen editing on any of these display terminals: Control Data Corporation Viking 721, CDC 722, DEC VT100,† Zenith Z19,† Heathkit H19,† Lear Siegler ADM3A,† Lear Siegler ADM5,† and Tektronix 4115.†

If you don't have one of these terminals, you may still be able to use the screen editing feature. Through the use of a special software program described in the NOS Screen Formatting Reference Manual, it may be possible to make your display terminal operate in screen mode. Check with your site adminsitrator for more information.

This appendix shows you how to edit a file using line editing and screen editing. Some of the line spacing shown in the terminal sessions is changed to allow easier reading of the comments.

First we'll show an example of line editing. You can always use the line editing directives regardless of the kind of terminal you're using.

Suppose the data file created in figure 5-5

```
3 4 5
2 1 5
1.5197 1.5197 1.5197
0 0 0
```

had been entered like this instead.

```
3 4 5
1.5297 1.597 1.51987
21 5
0 0 0
```

†DEC is a registered trademark of the Digital Equipment Corporation. Zenith Z19 is a product of the Zenith Corporation. Heathkit H19 is a product of the Heath Corporation. Lear Siegler ADM3A and Lear Siegler ADM5 are products of Lear Siegler Incorporated. Tektronix 4115 is a product of Tektronix Incorporated.

Follow this example to see what a few Full Screen Editor directives can do.

Terminal Session **Comments**

```
fse,data
EDIT: DATA
NOS FULL SCREEN EDITOR
```

This command starts the Full Screen Editor. If a file name is not included, the Full Screen Editor asks for the file name or, if the editor has already been used during this terminal session, it edits the same file.

```
?? print all
   1 3 4 5
   2 1.5297 1.597 1.51987
   3 21 5
   4 0 0 0
```

PRINT ALL displays the whole file. Notice that the Full Screen Editor has added line numbers that were not in the original file. They are used only by the Full Screen Editor and disappear when editing is through.

```
?? insert 1
```

The user wants to insert a new line after line 1.

```
   2 ? 2 1 5
   3 ?
```

When prompted by a single question mark, the user enters the new line. Pressing the carriage return after the question mark tells the Full Screen Editor to stop inserting lines.

```
?? print all
   1 3 4 5
   2 2 1 5
   3 1.5297 1.597 1.51987
   4 21 5
   5 0 0 0
```

Just to check, the user displays the file again. Notice that the subsequent line numbers have increased by one.

```
?? delete 4
   4 21 5
```

The user deletes the fourth line.

```
?? print all
   1 3 4 5
   2 2 1 5
   3 1.5297 1.597 1.51987
   4 0 0 0
```

The user checks the file again.

Terminal Session | **Comments**

```
 ?? alter 3
```

Next the user wants to alter line 3 which has several errors.

```
  3 1.5297 1.597 1.51987
A??    1
  3 1.5197 1.597 1.51987
```

The Full Screen Editor displays the line to be changed. To replace a character, the user spaces over to it and types the new character.

```
?? alter 3
```

Now the user wants to alter line 3 by inserting a character.

```
  3 1.5197 1.5197 1.51987
A??            ^1
  3 1.5197 1.51197 1.51987
```

To insert a character, the user spaces over to the position where the character should be, types ∧ (the circumflex), and the new character.

```
?? alter 3
```

Finally the user wants to delete a character from line 3.

```
  3 1.5197 1.51197 1.51987
A??                     #
  3 1.5197 1.51197 1.5197
```

To delete a character, the user spaces over to the character and types # (the number sign).

```
?? print all
   1 3 4 5
   2 2 1 5
   3 1.5197 1.51197 1.5197
   4 0 0 0
```

The user displays the entire file again and sees that it is correct.

```
?? quit
File: DATA (LOCAL)
File: FSEPROC  (NO CHANGES)
```

QUIT stops the Full Screen Editor.

```
 READY.
List,f=data
3 4 5
2 1 5
1.5197 1.51197 1.5197
0 0 0
```

When the file is listed after editing, notice that the line numbers no longer appear.

Remember, these changes are made only on the copy of the file at your terminal. To keep them, you must save or replace the file.

In summary, for line editing:

Entry	Description
FSE,filename	Starts editing on the file specified.
PRINT ALL	Displays the entire file.
INSERT number	Inserts following the line specified by number.
DELETE number	Deletes the line specified by number.
ALTER number	Alters the line specified by number. Characters can be replaced one-for-one, inserted (using ^), and deleted (using #).
QUIT	Ends editing on the file.

Now we'll show the same example using screen editing. Remember, you must have a display terminal that supports screen editing to perform screen editing. We'll assume that a Viking 721 is used.

First you must tell the system that you want to use screen mode rather than line mode which is the default. Type

```
screen,721
```

When the system responds

```
READY.
```

type

```
FSE,filename
```

In this case, the command that starts the Full Screen Editor is

```
fse,data
```

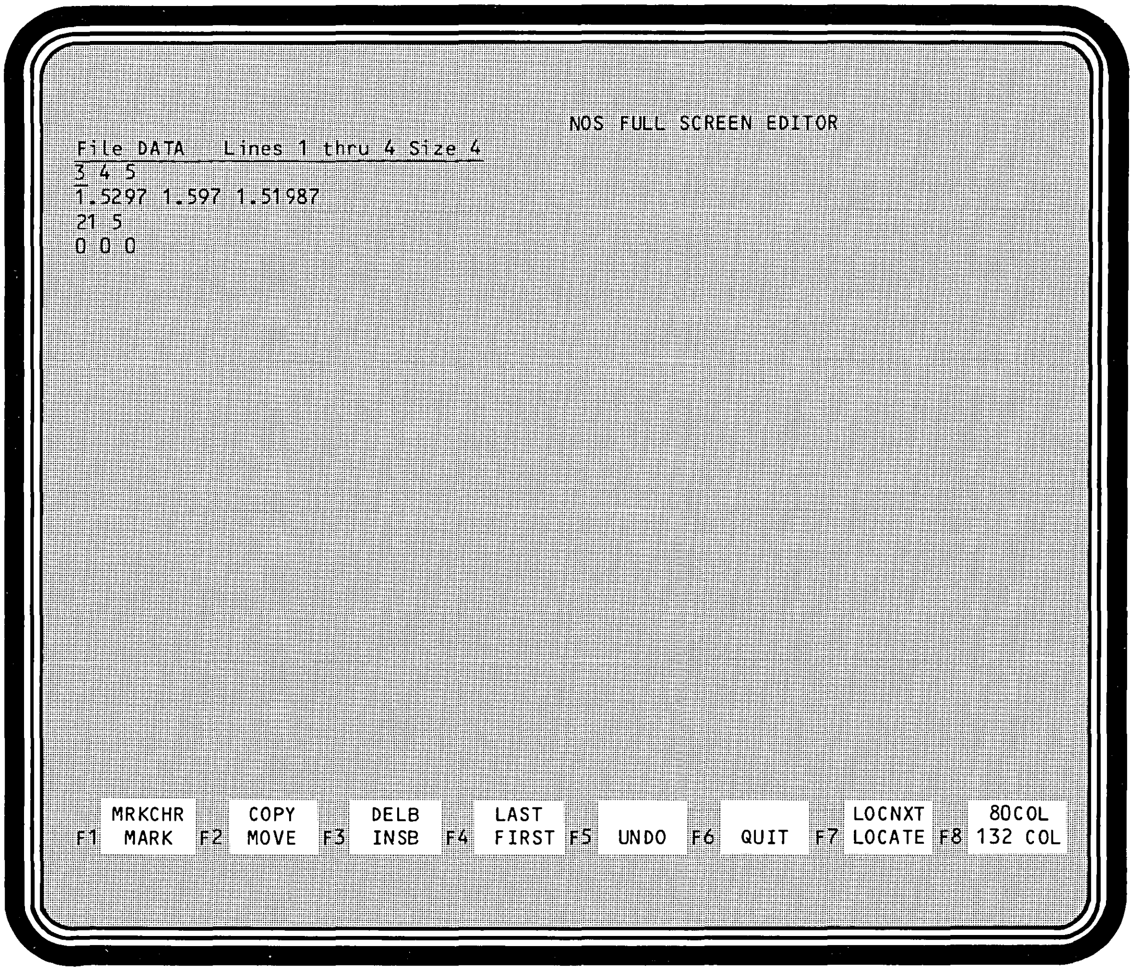
As in the earlier example, the file DATA

```
3 4 5
1.5297 1.597 1.51987
21 5
0 0 0
```

will be changed to

```
3 4 5
2 1 5
1.5197 1.5197 1.5197
0 0 0
```

After entering the FSE command, the following screen is displayed.



```
                                    NOS FULL SCREEN EDITOR
File DATA   Lines 1 thru 4 Size 4
3 4 5
1.5297 1.597 1.51987
21 5
0 0 0
```

To move the cursor while using screen editing, press one of the arrows on the 2, 4, 6, or 8 keys to the right of the main keyboard.

On some terminals, like the CDC 722, you must press the shift key along with the arrow.

The rest of this example shows partial screens with comments to the right.

Terminal Session **Comments**

```
File DATA   Lines 1 thru 4 Size 4
3 4 5
1.5297 1.597 1.51987
21 5
0 0 0
```

The line under 3 represents the current position of the cursor.

```
File DATA   Lines 1 thru 4 Size 4
3 4 5
1.5297 1.597 1.51987
21 5
0 0 0
```

Pressing the down arrow (that is, the number 2 key) moves the cursor to the second line.

```
File DATA   Lines 1 thru 4 Size 4
3 4 5

1.5297 1.597 1.51987
21 5
0 0 0
```

Pressing the shift key ⇧ and the insert key INSRT (on the upper righthand side of the keyboard) inserts a blank line where the cursor is positioned.

```
File DATA   Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5297 1.597 1.51987
21 5
0 0 0
```

The user can then type in the new line. When the carriage return is pressed, the line at the top is updated to show 5 lines.

```
File DATA   Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5297 1.597 1.51987
21 5
0 0 0
```

The cursor is positioned to the next line and the character to be edited, 2.

```
File DATA   Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5197 1.597 1.51987
21 5
0 0 0
```

The character in error is simply typed over and the correct character, 1, replaces it.

Terminal Session | **Comments**

```
File DATA    Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5197 1.597 1.51987
21 5
0 0 0
```

The cursor is again moved, this time to the position where a character should be inserted.

```
File DATA    Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5197 1.5 97 1.51987
21 5
0 0 0
```

Pressing the (INSRT) key inserts a blank where the new character can be inserted.

```
File DATA    Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5197 1.5197 1.51987
21 5
0 0 0
```

The new character is simply typed in.

```
File DATA    Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5197 1.5197 1.51987
21 5
0 0 0
```

The cursor is moved to the 8, a character to be deleted.

```
File DATA    Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5197 1.5197 1.5197
21 5
0 0 0
```

Pressing the (DLETE) key (on the upper righthand side of the keyboard) deletes the 8 and automatically moves the rest of the line over.

```
File DATA    Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5197 1.5197 1.5197
21 5
0 0 0
```

The cursor is moved down to the fourth line. This line should be deleted.

```
File DATA    Lines 1 thru 5 Size 5
3 4 5
2 1 5
1.5197 1.5197 1.5197
0 0 0
```

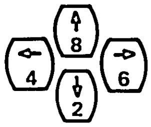
Pressing the shift key (⇧) and the (DLETE) key deletes the entire line. The file is now correct.

Pressing the function key labeled F6 at the top of the keyboard stops the Full Screen Editor. The system responds

```
FILE: DATA
FILE: FSEPROC  (NO CHANGES)

 READY.
```

Remember, the changes are made only on the copy of the file at your terminal. To keep them, you must save or replace the file.

In summary, for screen editing:

Entry	Description
FSE,filename	Starts editing on the file specified.
	Moves the cursor in the direction indicated by the arrow (up, down, left, or right). The arrows are located on the numbers 2, 4, 6, and 8 to the right of the main keyboard.
INSRT	Inserts a blank at the current position of the cursor.
⇧ INSRT	Inserts a blank line at the current position of the cursor.
DLETE	Deletes the character at the current position of the cursor.
⇧ DLETE	Deletes the line at the current positon of the cursor.
F6 function key	Ends editing on the file.

There are many more advanced features available on the Full Screen Editor. You can find out more by reading the Full Screen Editor user's guide mentioned in the preface.

INDEX

NOS 2 Reference Set Volume 1 Introduction to Interactive Usage 60459660 C

We would like your comments on this manual. While writing it, we made some assumptions about who would use it and how it would be used. Your comments will help us improve this manual. Please take a few minutes to reply.

Who Are You?

- □ Manager
- □ Systems Analyst or Programmer
- □ Applications Programmer
- □ Operator
- □ Other ______________

How Do You Use This Manual?

- □ As an Overview
- □ To Learn the Product/System
- □ For Comprehensive Reference
- □ For Quick Look-up

Do You Also Have?

- □ Volume 2, Guide to System Usage
- □ Volume 3, System Commands
- □ NOS Full Screen Editor User's Guide

What programming languages do you use? ______________________________

Which are helpful to you? □ Index on inside cover □ Full Screen Editor (Appendix C) □ Glossary □ Other: ______________________________

How Do You Like This Manual? Check those that apply.

Yes	Somewhat	No	
□	□	□	Is the manual easy to read (print size, page layout, and so on)?
□	□	□	Is it easy to understand?
□	□	□	Is the order of topics logical?
□	□	□	Are there enough examples?
□	□	□	Are the examples helpful? (□ Too simple □ Too complex)
□	□	□	Does the manual tell you what you need to know about the topic?
□	□	□	Is there enough information about terminals?
□	□	□	Are our assumptions about the login procedure on your system correct? (If not, please comment below.)

Comments? If applicable, note page number and paragraph.

Would you like a reply? □ Yes □ No

Continue on other side

From:

Name ______________________ Company ______________________

Address ______________________________ Date ______________

______________________________ Phone No. ______________

____

Please send program listing and output if applicable to your comment.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112

FOLD FOLD

Comments (continued from other side)