UNIX for Beginning Users

Developed by:

User Liaison Section, D-7131
Denver Office

[Name and Phone number deleted at authors Request]

Revision Date: September 16, 1991

I. INTRODUCTION

A. Audience

This course is for individuals who will be using the UNIX operating system on a Reclamation computer platform. It is assumed that the student has a general understanding of data processing concepts.

B. Course Objectives

Upon successful completion of this course the student will be able to:

1. Demonstrate a knowledge of basic UNIX ideas.

2. Recognize the different types of files and the file

structure.

3. Log in and out of UNIX using an interactive terminal.

4. Change the password and be aware of other responsibilities of owning an account.

5. Demonstrate a knowledge of where to get help.

6. Use the appropriate UNIX commands to display/print files, copy/move files, change file access permissions, create/delete directories, and change the current working directory.

7. Transfer a file to another computer platform using File Transfer Protocol (FTP). Use FTP commands to do the following: initialize FTP, establish connection, local computer commands, remote computer commands, close connection, exit FTP, help command, and special functions.

8. Use an editor to create files, input text, insert/replace text, copy/move text, and exit/save changes.

9. Use the mail utility to send/receive/delete messages

10. Use basic Annex commands to reestablish connection to a disconnected process.

C. Course Handout Conventions

There are several conventions used in this handout for consistency and easier interpretation:

1. Samples of actual terminal sessions are single-lined boxed.

2. User entries are shown in bold print and are underlined.

   QUIT

3. All keyboard functions in the text will be bold.

(Ret)	Backspace
Tab	Ctrl-F6
Print (Shift-F7)	Go to DOS (1)

   NOTE: (Ret) indicates the Return or Enter key located above the right Shift key.

4. Examples of user entries not showing the computer's response are in dotted-lined boxes.

5. Command formats are double-lined boxed.

6. Three dots either in vertical or horizontal alignment mean continuation or that data is missing from the

diagram.

Multimax, Nanobus, and UMAX are trademarks of Encore Computer Corporation

Annex is a trademark of XYLOGICS, Inc

UNIX and Teletype are registered trademarks of AT&T Bell Laboratories

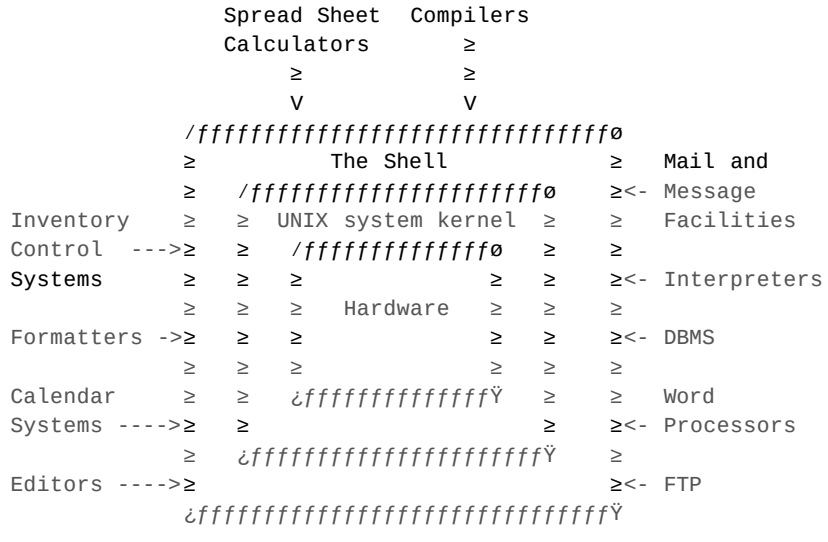
Ethernet is a trademark of Xerox Corporation

1. BASIC UNIX IDEAS

The UNIX operating system is a set of programs that act as a link between the computer and the user. The programs that allocate the system resources and coordinate all the details of the computer's internals is called the operating system or kernel.

Users communicate with the kernel through a program known as the shell. The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.

Here is a basic block diagram of a UNIX system.



The designers of UNIX used the following Maxims while writing the new operating system.

1. Make each program do one thing well. These simple programs would be called "tools."

2. Expect the output of every program to be the input to another program.

3. Don't stop building new "tools" to do a job. The library of tools should keep increasing.

1.1 The UNIX System

The main concept that unites all versions of UNIX is the following four basics:

Kernel

The kernel is the heart of the operating system. It schedules tasks and manages data storage. The user rarely interfaces with the kernel directly. This is the memory resident portion of the operating system.

Shell

The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell will support multiple users, multiple tasks, and multiple interfaces to itself. The shell uses standard syntax for all commands. There are two popular shells currently available, the BourneShell (standard System V UNIX) and the CShell (BSD UNIX). Because separate users can use different shells at the same time, the system can appear different to different users. There is another shell known as the KornShell (named after its designer), which is popular with programmers. This ability to provide a customized user interface is one of the most powerful features of UNIX.

Commands and Utilities

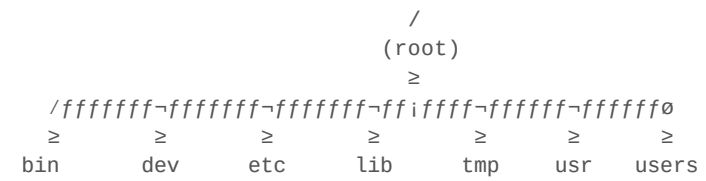
Separate utilities can be easily combined to customize function and output. They are flexible, adaptable, portable, and modular. They use pipes and filters. There are over 200 standard commands plus numerous others provided through 3rd party software.

Files and Directories

The directory system supports a multilevel hierarchy. Files and directories have access protection. Files and directories are accessed through pathnames. Files support multiple name links. Removable filesystems are also supported.

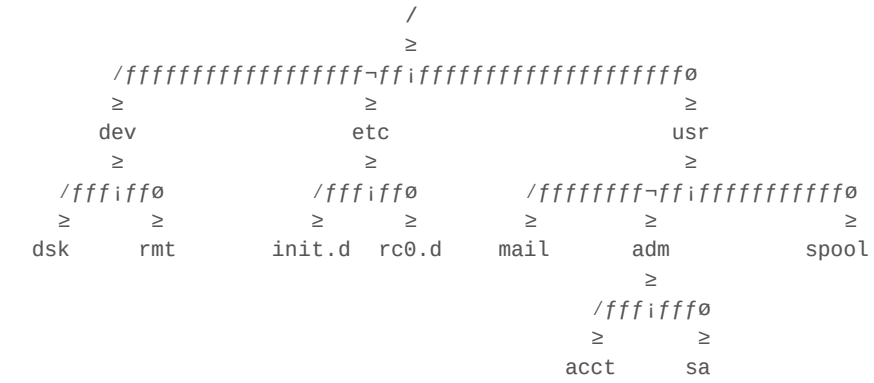
1.2 File Structure

All data in UNIX is organized into files. All files are organized into directories. These directories are organized into a tree-like structure called the filesystem. The following diagram describes the top level organization of the UNIX filesystem:



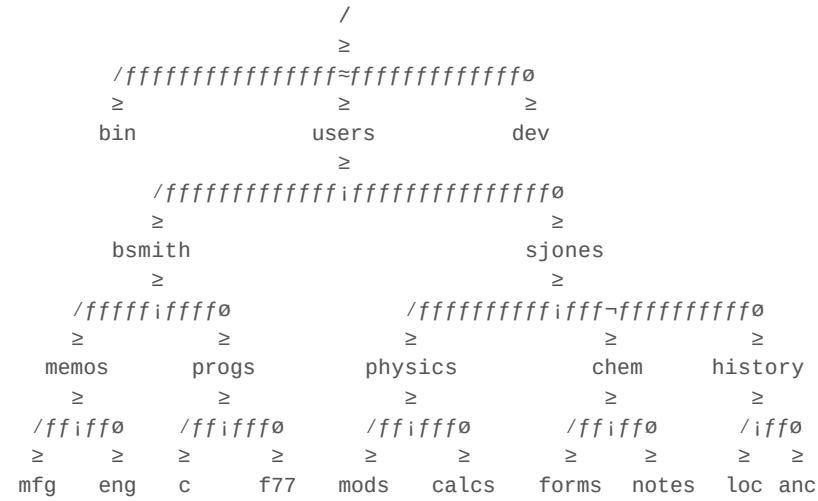
These directories, in turn, are also organized hierarchically.

For example:



In this example, dev, etc, usr, and adm are directories.
Directories contain other files or directories. Plain files
contain text or binary data and contain no information about
other files or directories.
Users can make use of this same structure to organize their
files.

For example:



Every file has a name. A filename is composed of one to fourteen
characters. Although you can use almost any character in a
filename, you will avoid confusion if you choose characters from
the following list.

1. upper case letters [A-Z]
2. lower case letters [a-z]
3. numbers [0-9]
4. underscore [_]
5. period [.]
6. comma [,]

The only exception is the root directory, which always uses the
symbol /. No other directory or file can use this symbol.

Like children of one parent, no two files in the same directory
can have the same name. Files in different directories, like
children of different parents, can have the same name.

The filenames you choose should mean something. Too often, a
directory is filled with important files with names like foobar,
wombat, and junk. A meaningless name won't help you recall the

contents of a file. Use filenames that are descriptive of the contents.

1.3 UNIX System Files

In order for you to have a basic understanding of the contents of some of the system directories, here is a partial list of those directories and what files they contain:

/bin	This is where the executable files are located. They are available to all user.
/dev	These are device drivers.
/etc	Supervisor directory commands, configuration files, disk configuration files, reboot files, valid user lists, groups, ethernet, hosts, where to send critical messages.
/lib	compiler libraries
/tmp	scratch processes, editors, compilers, and databases
/bsd	Berkeley commands
/mnt	empty, used for disks
/stand	boot information
/lost+found	orphans go here (look here after system crash)
/unix*	executable, bootable kernel

This is not an exhaustive list of directories that contain system information but it is intended to remove some of the mystery behind these directories and the types of files they contain.

1.4 Command Line Syntax

Users enter commands at the shell prompt. The default BourneShell prompt is the dollar sign ($). In general, the shell expects to see the following syntax:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  command  options  arguments                ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Command - This is the UNIX command. Sometimes the command is representative of the function. For example, the command to list the contents of a directory is ls. The first and third letters of the word "list" are used. Unfortunately, this is not always the case.

Options - These are also known as flags. The common form is:

-A

where A is the abbreviation of the optional function of the command. For example, the command ls lists the contents of a directory, while the command ls -l provides a long listing and ls -C provides output in columns. Several options can be combined following one '-'; for example -CF, or

they can be entered separately as -C -F.

Arguments - These can be file names, user names, or qualifiers to the command or one of its options.

Example:

```
. $ls -CF sjones                                                .
```

The UNIX command is ls list contents of directory the dash (-) indicates the options.

C = Multiple-column output with entries sorted down the columns

F = Put a slash (/) after each filename if that file is a directory and put an asterisk (*) after each filename that is executable.

sjones = name of the directory to list (it can be a relative or absolute pathname)

Example:

```
. $diff memo1 memo2                                             .
```

diff - differential file comparator command

memo1 - filename argument

memo2 - filename argument

This command will tell what lines must be changed in two files to bring them into agreement.
Here is another example that doesn't fit the general syntax for UNIX commands.

Example:

```
. $find . -atime +7 -print                                      .
```

find - find files

. - the current working directory

-atime - True if the file has been accessed in n days (n is the +7)

-print - always true; causes the current path name to be printed

So, this command will give a listing of all files in your current working directory that have been accessed in the past seven days.

Some commands have several options and/or arguments; while others, like passwd and mail, are interactive and will prompt the user for additional input.

1.5 Correcting Mistakes

Because the shell and most other utilities do not interpret the command line (or other text) until you press the (Ret) key, you can correct typing mistakes before you press (Ret). There are two ways to correct typing mistakes. You can erase one character at a time, or you can back up to the beginning of the command line in one step. After you press (Ret), it is too late to make a correction.

1.5.1 Erasing Characters

When entering characters from the keyboard, you can backspace up to and over a mistake by pressing the erase key (#) one time for each character you wish to delete. The # will appear on the screen, and the character preceding it will be discounted.

Example:

```
. $ls phajne#y                                                  .
```

In this example, the e will be ignored and ls phajny is sent to the Multimax. Multiple typos can be erased; simply press one # for each character to be erased. The erase key will back up as many characters as you wish, but it will not back up past the beginning of the line.

1.5.2 Deleting an Entire Line

You can delete an entire line you are entering any time before you press (Ret) by pressing the kill key (@). When you press the @ (kill key), the cursor moves down to the next line and all the way to the left. The shell doesn't give you another prompt, but it is as though the cursor is following a prompt. The operating system does not remove the line with the mistake but instead ignores it. Now enter the command (or text) again from the start.

1.5.3 Aborting Program Execution

Sometimes you may want to terminate a running program. UNIX might be performing a listing that is too long to display on your screen or for some other reason you want to terminate execution. To terminate program execution press the Delete key. The operating system sends a terminal interrupt signal to the shell. When the shell receives this signal, it displays a prompt and waits for another command.

1.5.4 Controlling Output to the Screen

There are several ways to control the flow of characters to the screen as a result of executing a command. Such as:

Ctrl-S	-	This keyboard function command will suspend the flow of characters to the screen as the result of executing a command. The screen will not continue until the keyboard function to resume output is given.
Ctrl-Q	-	This keyboard function command will resume the output to the screen.
Hold Screen	-	If your terminal has this key (i.e. VT200), you can press it once to stop output to the screen. To resume output to the screen, press the key again.

Denver BOR MULTIMAX

Each BOR Multimax 310 has four 15 Megahertz National Semiconductor 32-bit processors with 64 kilobytes of cache memory rated at 2 million instructions per second (MIPS) for a total of 8 MIPS. The main memory consists of 32 megabytes (million bytes). There can be a maximum of 14 disk drives. Each drive has a capacity of 600 megabytes for a total capacity of 8.4 gigabytes (a gigabyte is one thousand million bytes)

Connection to the Multimax is accomplished through one of several methods. Access is made through TCP/IP based Annex terminal servers. The two Annex II servers have 32 ports each and the Annex I has 16 ports. The Annex II servers will allow up to 64 users access to the two Multimax computers. The Annex I is used for access to the on-line printers. CDCnet and TELNET are other ways to gain access to the Multimaxes.

Printouts are handled on a 600-line-per-minute line printer and a 10-page-per-minute laser printer. Each Multimax has a hardcopy terminal and a CRT to serve as an operator console. There are two tape drives capable of 1600 or 6250 bits per inch (bpi) on each system. There is also a cassette tape drive.

Software available are FORTRAN, COBOL, C, and UNISOL (an accounting package). The database management system is INGRES by Relational Technology, Inc. PROCOMM+ will be the communication interface with IBM PC's and compatibles. The operating system for the Multimax is UMAX V. UMAX V is the name for the Encore implementation of UNIX System V.

1.6 Logging on the Annex

This sample session shows how the login process is displayed on the terminal screen and is uniform for all users. To bring the standard menu onto the screen, press the Space Bar. If you are using a PC, first start PROCOMM+. Then when you are in the Terminal-Mode Screen, press the Space Bar; and the MICOM menu will appear.

NOTE: Login procedures from the regions are included in the back of this manual

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ WELCOME TO THE B.O.R. NETWORK P/S:B                          ≥
≥ SYSTEMS PRESENTLY AVAILABLE ARE:                             ≥
≥                                                              ≥
≥  **SYSTEM**         **NAME**                                 ≥
≥                                                              ≥
≥ VAX 8300'S            VAX                                    ≥
≥ CYBER/CDCNET F.E.     CDC                                    ≥
≥ ENCORE/UNIX           MAX                                    ≥
≥ OUT DIAL              OD                                     ≥
≥                                                              ≥
≥ TO SELECT A SYSTEM, ENTER THE SYSTEM                         ≥
≥ NAME AND CARRIAGE RETURN AT NEXT                             ≥
≥ PROMPT.                                                      ≥
≥                                                              ≥
≥ CHANNEL 08/061. ENTER RESOURCE MAX                           ≥
≥                                                              ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

MAX is the resource name you must enter to be connected to the Annex, which is the Multimax front end processor. Some MICOM menus might not have the MAX selection; in this case, enter MAX to select the Annex. This is the same as if the menu showed the option.
After entering MAX you will see something similar to the following:

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ CONNECTED TO 06/011                                          ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

This indicates that you are connected to the port selector. Wait two seconds, press (Ret) twice, and the annex prompt will appear after a warning message.

Sample Session:

```
/fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥                                                                             ≥
≥ Annex Command Line Interpreter   *   Copyright 1988 Xylogics, Inc.         ≥
≥                                                                             ≥
≥  ***WARNING***Unauthorized access to U.S. Government computers             ≥
≥                is punishable by fine and/or imprisonment. ***WARNING***    ≥
≥ annex:                                                                      ≥
¿fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

1.7 Logging on the Multimax

To establish a connection between the Annex and the Multimax enter the following command at the Annex prompt:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  rlogin <host>                              ∫
∫                                                              ∫
∫     host - name of the Multimax                              ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

The Denver Multimaxes have been assigned the names domax0 and domax1. The names stand for the Denver Office Multimax System 0

and 1. The domax0 is used for production of Bureau-wide applications. The domax1 is used for training and application development and it is the one to use for exercises associated with this course.

To enter domax1 type:

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ annex:rlogin domax1                                            ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
                              or
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ annex:r domax1                                                 ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

NOTE: Abbreviations are allowed for the Annex commands, the only requirement is to type in enough characters to make it unique.

When the Annex has opened communications with the selected host, the following prompt will appear:

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ login:                                                         ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

To connect with the host, enter your login name at the prompt. Your login name is assigned to you by the system administrator and typically will be your first initial and last name, all one word with no spaces. Only 8 characters are allowed for the username so extra letters will be truncated.

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ login:rharding                                                 ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

Once the login name has been accepted, the next prompt will be for the password. The following prompt will appear on the screen.

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ Password:                                                      ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

Enter your password. For security reasons, the host will not display your password as you type it.

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ Password: secret                                               ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

Once you have entered the correct password. The login procedure will continue and the following will appear on the monitor screen.

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ UNIX System V Release ax.2.2o ns32332                                           ≥
≥ domax1                                                                          ≥
≥ Copyright (c) 1984 AT&T                                                         ≥
≥ All Rights Reserved                                                             ≥
≥  ***WARNING***Unauthorized access to/use of this U.S. Government                ≥
≥               computer is punishable by fine and/or imprisonment. ***WARNING***≥
≥ $                                                                               ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

At this point you are successfully signed on to the Multimax.
The dollar sign ($) is the default prompt for the BourneShell.

1.8 Logging Off the Multimax

At the shell prompt $, you can logout of the Multimax using one of the following methods:

1. Enter the keyboard function command Ctrl-D.

2. Type the UNIX command exit.

Once you have entered the command to logout the following will appear on the screen:

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ $exit                                                            ≥
≥ CLI: Connection closed.                                          ≥
≥ annex:                                                           ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

Once you are back at the Annex prompt, you can establish another connection or logout of the Annex.

1.9 Logging Off the Annex

When the Annex prompt (annex:) appears, you can enter the command to logout of the Annex. The command to logout of the Annex is as follows:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  hangup                                         ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

There is a 60 minute inactivity timeout programmed into the Annex; however, it is a waste of resources if you don't enter hangup. When you are finished with your session, be sure to enter hangup at the annex: prompt.

If you don't type anything for 60 minutes, the Annex will log you out of the system and display the following message:

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ *** Annex Port Reset Due to Inactivity Timeout ***            ≥
≥                                                               ≥
≥ Annex Command Line Interpreter                                ≥
≥ DISCONNECTED                                                  ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

When the hangup command has been entered, the following will appear on the screen:

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ annex: hangup                                                 ≥
≥                                                               ≥
≥ Resetting line and disconnecting.                             ≥
≥                                                               ≥
≥                                                               ≥
≥ Annex Command Line Interpreter * Copyright 1988 Xylogics      ≥
≥ annex:                                                        ≥
≥ DISCONNECTED                                                  ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

1.10 Changing the Password

The following command will change the password.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫    Command Format:     passwd                                 ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

You will be prompted to enter the existing password (this question is skipped if you don't have a password). Next you will be prompted to enter the new password. You will then be asked to enter the new password again. This will verify that you have not made a typographical error. If the two entries are the same, the password will be changed. The new password must meet the following criteria:

NOTE: Some of these items are configurable by the system administrator and these reflect the settings for the Denver Multimax only.

1. Each password must have at least six characters. Only the first eight characters are significant.

2. Each password must contain at least two alphabetic characters and at least one numeric or special character. Alphabetic characters can be upper or lower case.

3. Each password must differ from the login name and any reverse or circular shift of that login name. For comparison purposes, an upper case letter and its corresponding lower case letter are equivalent.

4. A new password must differ from the old by at least three characters. For comparison purposes,

an upper case letter and its corresponding lower case character are equivalent.

Passwords on the Multimax have a thirteen-week expiration period. At the end of the thirteen weeks, you will be required to change your password. Once you have changed the password, you cannot change it again for two weeks. This prevents you from immediately changing back to the old password and eliminates a possible security violation. If you try to change the password before two weeks have passed since the last change, a warning message will be displayed.

Sample Session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ $passwd                                                       ≥
≥ Changing password for teacher                                 ≥
≥ Old password: secret                                          ≥
≥ Sorry: < 2 weeks since the last change                        ≥
≥ $                                                             ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

NOTE: This is about as friendly as UMAX will ever get.

Try to choose a password that is not easy for someone else to guess. The increasing number of computer crimes involving thefts all point to a need for protecting the system from unauthorized access. Do not use words like your birthdate, telephone number, spouse's name, child's name, etc. for passwords. Although you may think passwords are an unnecessary nuisance, they are an important way to strengthen the security of the computer system.

1.11 On-line Manual

The major source of on-line help is in the form of documentation known as the on-line manual pages. The pages are divided into eight sections. Section 1 contains entries for UMAX user commands; the other sections describe administrative tools, library functions, games, and internal system structure and calls.

To gain access to the on-line manual pages enter the following command:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫      Command Format:   man <command>                          ∫
∫                                                               ∫
∫      command - the UNIX command you want information about    ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

NOTE: The name 'man' stands for manual.

Example:

```
.................................................................
. $man ls                                                       .
.................................................................
```

This command will display the on-line manual pages for the ls command.

The on-line manual pages entry begins with the command name and a

one line summary followed by a synopsis of the command line syntax. Optional flags and arguments are enclosed by square brackets []. A detailed description of the command and all of its options and arguments follow the synopsis. The description can include helpful examples. At the conclusion of the entry, related files and commands are listed.

NOTE: Most on-line manual pages will fill more than one screen. Be sure to control the output to your screen.

1.12 who and finger Commands

Once you have logged onto the Multimax, you can find out who is logged on the system with the following commands:

```
Command Format:  who [options]

   options - see man pages for a complete list
```

The default output (no options) of the who command lists the user's login name, terminal line, and the time that the user logged in.

Sample Session:

```
$who
jwheeler          ttyp0     Aug 15 10:26
mvlsdba           rt02190   Aug 15 09:25
teacher           rt020b0   Aug 15 11:07
eholderf          rt021c0   Aug 15 11:03
dbowman           rt01150   Aug 15 08:58
$
```

Options will display other information about the users that are currently logged onto the system. Some items available are the amount of time that has elapsed since activity occurred on that line, the process identifier (PID) of the login process, comments, and exit information.
A UNIX command that provides a little more information about users that are logged in the system is the finger command.

```
Command Format:  finger [options] [user1]

   options - see on line manual for complete list

    user1  - login name
```

The finger command with no options will list the login name, full name, terminal name, write status (an asterisk (*) before the terminal name indicates that write permission is denied), idle time, login time, office location, and phone number (if known) for each user that is currently logged in the system.

Sample Session:

```
$finger
Login        Name           TTY       Idle   When     Office
```

```
≥ Jwheeler     Jim Wheeler    ttyp0       16 Wed 10:26   MP     ≥
≥ mvlsdba      Motor Veh Lic  rt02190     16 Wed 09:25   d7160  ≥
≥ teacher      Teacher Acct  *rt020b0        Wed 11:07          ≥
≥ eholderf     Eileen Holder  rt021c0      1 Wed 11:03          ≥
≥ dbowman      Dale Bowman    rt01150         Wed 08:58          ≥
≥ $                                                             ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

Workshop 1

This workshop will reinforce your understanding of the ideas presented in Chapter 1. Each student is to complete the entire workshop.

DESK EXERCISES

1. What two organizations first developed UNIX?

2. In what high level programming language is UNIX written?

3. What are some characteristics of UNIX?

4. What is Encore Computer Corporations implementation of UNIX called?

5. What part of UNIX controls the details of the computer's internal operations?

6. What part of UNIX allows the user to communicate with the computer?

Continue on the next page

7. What is the name of the tree-like structure under which all data is stored?

8. What is the name of the highest level directory?

9. What symbol represents the highest level directory?

10. What is the general syntax of a UNIX command?

11. What is the most common form for listing options on a command line?

12. What character would you use to erase a character on the command line?

13. What character terminates the execution of a command?

14. What is the default BourneShell prompt?

15. How can you control the flow of output to your monitor screen?

1. What annex command is entered to make a connection to the Multimax?

2. What is the UNIX command to change the password?

3. How long is your password valid?

4. How long do you have to wait before changing your password again? Why?

5. What UNIX command is used to logout of the Multimax?

6. What is the command to logout of the annex?

COMPUTER EXERCISES

7. Login to the Multimax

   a. What did you notice when you entered the password?

   b. Can you see the password as you enter it?

   c. What happens if you make a mistake while entering the password?

8. What do you see once you have logged in? Write it here.

9. Enter the command which displays the man pages for the man command. (Don't forget to control output to the screen.)

   The first section is titled "NAME," what are the titles of the other sections?

10. What are the options for the man command?

11. Enter the command to find out who (hint) is logged into the system.

12. What command will give you more information about the current users? Try it.

13. Logout of the Multimax and the Annex.

2. FILES

In UNIX, all data is organized in files. An ordinary file is a memo, source code program or shell script. A shell script or program source code can be viewed or edited from your terminal. Other files contain binary data, like programs for the kernel; these files cannot be viewed or edited on the terminal.

Peripheral devices such as disks, tape drives, printers, and terminals are also assigned file names. Device files are considered to be special files. They have 'special' characteristics. Although input and output can be redirected to and from a special file, do not attempt to display the contents of a special file on your terminal.

3.1 File Access Modes

File access modes are the protections that can be assigned to files. This protection can protect your files from unauthorized reading or writing. You can even protect your files from yourself (you can prevent accidental deletion).

There are three access modes for files:

r (read) read, examine, copy data in a file

w (write) modify, delete a file

x (execute) use the file as a command

Users with access to a file fall into one of three groups:

u (user) the file's owner

g (group) users in the same group

o (other) everybody else

The first output field of the ls -l command is a ten character field. Characters two through ten describe the file access modes. A typical access mode listing looks like:

rwxr-xr-x

Of the nine columns, the first three describe modes for the file's owner, the next three for his group, and the last three for everyone else. Within each group of three, the first column describes read access mode, the second write, and the third execute. A letter in a column indicates access granted, a dash (-) indicates access denied.

Using the previous example, the user has r (read), w (write), and x (execute) permissions. Members of the user's logical group can read (r) or execute (x). Everyone else has read (r) and execute (x) permissions, too. The effect of these permissions is that the file's owner is the only one who can modify the file; but everyone can examine, copy, or execute the file.
To change access modes on a file or directory, use the chmod command.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  chmod <access> <file1[filen]>              ∫
∫                                                              ∫
∫        access    - access permissions                        ∫
∫     file1[filen] - one or more files to change permissions   ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Access can be expressed in either of two forms:

- with letters: [ugo] [+-=] [rwx]

- with numbers: [0-7] [0-7] [0-7]

Let's look at the method of changing the file permissions with letters. The letters u, g, and o represent user, group, and others, respectively. The + (plus) sign means to add the permission and the - (minus) sign means to remove the permission. The = (equal) sign means to set the permissions as shown. Of course, r,w, and x are read, write, and execute.

If, for illustration purposes, we created a file named file1 that had the following permissions:

rw-rwxrwx

and you want to give yourself (user) execute permission and take away others' (others' here means group and everyone else) write permissions.

Sample Session:

```
/ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßø
≥ $chmod u+x,g-w,o-w file1                                     ≥
≥ $                                                            ≥
¿ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßŸ
```

Now if we use the ls -la command, and look at the file permissions for file1, they will look like this:

rwxr-xr-x

If you want to set several protections at once use the equal sign. The following example will set the permissions for the user to read and execute.

Sample session:

```
/ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßø
```

```
≥  $chmod u=rx file1                                                  ≥
≥  $                                                                  ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

The second method of changing the permissions is to use the octal digits (0-7). The octal digits 0 through 7 are represented in binary in the following manner.

Octal	Binary	Corresponds to permissions
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

Notice that every time a one digit (1) occurs in the binary number the corresponding permissions are also set. Every time a zero (0) occurs, the corresponding permission is denied. So to change the file permissions in the previous example, this is the command to enter:

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ $chmod 755 file1                                                    ≥
≥ $                                                                   ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

The first octal digit assigns user permissions of read, write and execute. The second digit assigns the group permission to read and execute. The last digit sets the others permission to read and execute too.

3.2 Listing Contents of Directories

The ls command is used to display file names and their characteristics. Since file names are stored in directories, ls actually reads directory files. Executing ls with no flags or arguments simply lists the names of the files that exist in your current working directory. The initialization files will not be listed.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫   Command Format:  ls [options] [dir1[dirn]]                       ∫
∫                                                                     ∫
∫    options - see man pages for a complete list                      ∫
∫                                                                     ∫
∫    dir1[dirn] - one or more directory names                         ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

The -a flag will cause the hidden (initialization) and all other filenames to be displayed.

The -C flag causes the output to be changed from single-column to multi-column display.

The -F flag adds a character to the end of each displayed filename:

```
       /      indicates a directory
       *      indicates the file is executable.
   blank      indicates a plain or ordinary file
```

The -l flag causes detailed information to be printed for files in the directory. This information includes:

```
       file type (directory, block special, character special,
                fifo special, symbolic link, or ordinary file)
       access modes
       number of links
       ownership
     group affiliation
       size in bytes
       date and time of last modification
       filename
```

Without a filename argument, ls displays information about the current working directory. The output is automatically sorted alphabetically by default.

Example:

```
.................................................................
. $ls                                                           .
.................................................................
```

The following example provides a long listing of the current working directory.

Example:

```
.................................................................
. $ls -l                                                        .
.................................................................
```

This example shows the ls command with no arguments so it uses the default, the current working directory. The argument could be a relative or absolute directory name.

Sample session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ $ls -la                                                       ≥
≥ total 975                                                     ≥
≥ drwxrwxr-x  4 teacher   class   2048  Jul 16 17.56 .          ≥
≥ drwxr-xr-x 60 root              1536  Jul 13 14:18 ..         ≥
≥ -rwx------  1 teacher   class   4210  May  1 08:27 .profile   ≥
≥ -rwxr-xr-x  1 teacher   class   1948  May 12 13:42 memo       ≥
≥ $                                                             ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

3.3 File Classifications

The file command will classify files according to their contents.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  file [options] <file1[filen]>              ∫
∫                                                              ∫
∫    file1[filen] - one or more filenames to analyze           ∫
```

```
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

A few of the classifications that the file command displays are shown below. The results of using the file command are not always correct.

```
       English text
       ascii text
       c program text
       cannot stat
       commands text
       data
       directory
       empty
       executable
```

Sample Session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ $file speople                                                 ≥
≥ speople:                  commands text                       ≥
≥ $file test                                                    ≥
≥ test:                     directory                           ≥
≥ $file mail                                                    ≥
≥ mail:                     data                                ≥
≥ $                                                             ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

By convention, files beginning with a dot (.) are called initialization files or 'hidden files'. These files describe your environment to the shell. They are sometimes called 'dot files'.

By convention, files that end with:

```
                .c   are C source code programs
                .f   are Fortran source code programs
                .o   are object programs
                .a   are archive files
```

3.4 Displaying Files

The cat command displays the contents of a file. The command cat is an abbreviation for catenate. This command will read each file in sequence and write it to the monitor screen.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  cat [options] [file1[filen]]               ∫
∫                                                               ∫
∫    options - see man pages for a complete list               ∫
∫                                                               ∫
∫    file1[filen] - one or more file names                     ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

If no filename is given, or the argument - is encountered, cat reads from standard input.

Sample session:

```
.................................................................
.   $cat                                                        .
.................................................................
```

This is the simpliest example but not very exciting. The cat

command will get its input from the keyboard. Everything that is typed will be displayed on the monitor.

If an argument is given to the cat command that file will be displayed on the monitor.

Sample session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥   $cat main.c                                                    ≥
≥   main ()                                                       ≥
≥   {                                                             ≥
≥         printf ("hello from main!\n\n");                          ≥
≥         printf ("calling function1!\n\n");                        ≥
≥       funct1();                                                 ≥
≥       printf ("back from function1!\n\n");                      ≥
≥         printf ("calling function2!\n\n");                        ≥
≥         funct2();                                                 ≥
≥         printf ("that's it!\n\n");                                ≥
≥   }                                                             ≥
≥   $                                                             ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

Several files can be displayed on the monitor one after the other by separating the filenames with a space.

Sample session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥    $cat main.c main.f                                           ≥
≥    main ()                                                      ≥
≥    {                                                            ≥
≥           printf ("hello from main!\n\n");                        ≥
≥           printf ("calling function1!\n\n");                      ≥
≥           funct1();                                               ≥
≥           printf ("back from function1!\n\n");                    ≥
≥           printf ("calling function2!\n\n");                      ≥
≥           funct2();                                               ≥
≥           printf ("that's it!\n\n");                              ≥
≥     }                                                           ≥
≥             program calling                                     ≥
≥         write(6,100)                                            ≥
≥     100     format('Hello from main!',/)                          ≥
≥           write(6,110)                                          ≥
≥     110     format(' Calling subroutine1!',/)                     ≥
≥             call sub1                                             ≥
≥           write(6,120)                                          ≥
≥     120     format(t15' Back from subroutine1!',/)                ≥
≥           write(6,130)                                          ≥
≥     130     format(' Calling subroutine2!',/)                     ≥
≥           call sub2                                             ≥
≥           write(6,140)                                          ≥
≥     140     format(t15' Back from subroutine2!',/)                ≥
≥           write(6,150)                                          ≥
≥     150     format(' Thats all, folks!')                          ≥
≥           end                                                   ≥
≥     $                                                           ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

If the file contains more lines than can be displayed on the screen the display will continue to scroll until the last line has been displayed then the prompt will be redisplayed. This can be a problem if you intend to read the text. Be prepared to stop the screen so it can be read.
The pg command displays the contents of a file one screen at a time. It allows the user to perform string searches and to scroll backwards.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  pg [options] [file1[filen]]                ∫
∫                                                               ∫
∫    options - see man pages for a complete list                ∫
∫                                                               ∫
∫    file1[filen] - one or more files to paginate               ∫
∫                                                               ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥                                                                            ≥
≥   $pg memo                                                                 ≥
≥   What's Happening                                                         ≥
≥   by Pam Hajny                                                             ≥
≥   Denver Office                                                            ≥
≥                                                                            ≥
≥   With IRM Training:                                                       ≥
≥                                                                            ≥
.   A Reclamation-wide workshop was held in early October to                 .
.                                                                            .
.                                                                            .

≥   three groups; CYBER, VAX, and other (PC/LAN, scientific,                 ≥
≥   :                                                                        ≥
≥                                                                            ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

Twenty three lines of the file will appear and the : (colon) prompt will appear on the last line. To have the next twenty three line of the file appear, simply press (Ret). If you don't want to see anymore of the file, enter a q (for quit) and the shell prompt will be redisplayed.
The following UNIX command is useful for viewing the end of a file without having to display the entire file.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  tail [options] [file1]                      ∫
∫                                                               ∫
∫      options - see man pages for a complete list                ∫
∫                                                               ∫
∫    file1 - the file to display, if none is given use          ∫
∫            standard input                                     ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

The tail command displays the last 10 lines of file by default.
The tail command accepts a -N flag to display the last N lines.

Sample Session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ $tail memo                                                                 ≥
≥ data communication between the ASC IBM and other Reclamation computers.    ≥
≥ Asynchronous communication can be accomplished with the same terminals     ≥
≥ we use for other computer tasks, over the same lines and through the MICOM ≥
≥ port selectors.  Currently, host-to-host communications is accomplished    ≥
≥ over a line between the IBM and the CYBERs.  The software that supports    ≥
≥ this communication is called NJEF.  Although the capability has been there ≥
≥ for some time, we have recently been working with ASC personnel to         ≥
≥ improve its reliability and accessibility.  For CYBER users, there is      ≥
≥ an NJEF Users' Guide available which can be requested through the Hotline  ≥
≥ (303) 236-4567.                                                            ≥
```

```
≥ $                                                                            ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

3.5 Removing Files

The rm command will remove the entries for one or more files from a directory. If an entry was the last link to the file, the file will be destroyed. Removal of a file requires write permission to the directory itself, but neither read nor write permission to the file itself. The format for the rm command is:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫    Command Format:  rm [options] <file1[filen]>             ∫
∫                                                             ∫
∫    options - see man pages for a complete list              ∫
∫                                                             ∫
∫    file1[filen] - one or more files to remove               ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ $ls                                                                  ≥
≥ memo                                                                 ≥
≥ tdata                                                                ≥
≥ subdir                                                               ≥
≥ $rm memo                                                             ≥
≥ $ls                                                                  ≥
≥ tdata                                                                ≥
≥ subdir                                                               ≥
≥ $                                                                    ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

The file memo has been deleted from the current working directory.
Multiple files can be deleted by separating the filenames with a space.

Sample session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ $ls                                                                  ≥
≥ memo                                                                 ≥
≥ tdata                                                                ≥
≥ subdir                                                               ≥
≥ $rm memo tdata                                                       ≥
≥ $ls                                                                  ≥
≥ subdir                                                               ≥
≥ $                                                                    ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

3.6 Printing Files

The lp command routes a file to a printer.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  lp [-d<dest>] [-n<number>] [file1[filen]]   ∫
∫                                                               ∫
∫     dest    - destination (default set by administrator)      ∫
∫                                                               ∫
∫     number  - number of copies (default is 1)                 ∫
∫                                                               ∫
∫     file1[filen] - one or more files to be printed            ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

If no file name is mentioned the standard input is assumed. The filename dash (-) stands for standard input and may be supplied in conjunction with named files. The order in which the filenames appear is the order in which they will be printed.

The printers in Denver have the following destination names:

Mannesman 910 laser printer - mtlzr

Mannesman 600 line printer - mt_600 (Denver default)

If no specific printer is given the default printer will be selected. The following example will print one copy (default) of the file called test_285 to the line printer (default).

Sample session:

```
$lp test_285
request id is mt_600-1271 (1 file)
$
```

It is possible to specify the printer as shown in the following example. In this case, we specified the default printer.

Sample Session:

```
$lp -dmt_600 test_286
request id is mt_600-1272 (1 file)
$
```

To print two copies of a file called test_287 on the laser printer in Building 53 in Denver, enter the following command:

Sample Session:

```
$lp -dmtlzr -n2 test_287
request id is mtlzr-1273 (1 file)
$
```

3.7 Print Status

The lpstat command will print information about the current status of the printer system.

Command Format: lpstat [options]

options - see man pages for a complete list

If no options are given, the lpstat command will print the status of all requests made to lp by the user.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $lpstat                                                              ≥
≥ mtlzr-1274    teacher     22560    Jul 16 09:05 on mtlzr      ≥
≥ $                                                                    ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

The first field is the remote id of the print job. The username is next and the size (in bytes) of the print file. The date and time are next and finally the name of the printer.

One of the options available is -t. This option will print all of the printer status information.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $lpstat -t                                                           ≥
≥ scheduler is running                                                 ≥
≥ system default destination: mt_600                                   ≥
≥ device for mt_600: /dev/rlp000                                       ≥
≥ device for mtlzr: /dev/rt0002                                        ≥
≥ mt_600 accepting requests since Sep 19 16:09                         ≥
≥ mtlzr accepting requests since Sep 19 16:43                          ≥
≥ printer mt_600 is idle. enabled since Jul 3 16:52                    ≥
≥ printer mtlzr is idle. enabled since Jul 3 16:51                     ≥
≥ $                                                                    ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

This is an example of the kinds of information available from the lpstat command.

3.8 Canceling Print Jobs

The cancel command will cancel printer requests made by the lp command. The command line arguments can be either request id's (these are returned by the lp command) or the printer name. If you specify the request id, the cancel command will stop the job even if it is currently printing. If you specify the printer name, the job currently being printed will be canceled. In either case, the cancellation of a request that is currently printing will free the printer to print the next request.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫     Command Format:     cancel <[ids] [printer]>             ∫
∫                                                              ∫
∫      ids - request ids (returned by lp command)              ∫
∫                                                              ∫
∫      printer - printer name                                  ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $lp -dmt_600 contest                                                 ≥
≥ request id is mt_600-1280 (1 file)                                   ≥
≥ $cancel mt_600-1280                                                  ≥
≥ request "mt_600-1280" canceled                                       ≥
≥ $                                                                    ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

3.9 Copying Files

A user may make a copy of a file if he has read access to that

file. The cp command can be used to copy the contents of one
file to another.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  cp <file1[filen]> <target>                 ∫
∫                                                              ∫
∫    file1[filen] - one or more source files                   ∫
∫                                                              ∫
∫    target       - file or dirname                            ∫
∫                                                              ∫
∫                   file1 and target cannot be the same and    ∫
∫                   if the target is a file its' contents are  ∫
∫                   destroyed.                                 ∫
∫                                                              ∫
∫                   If target is a directory, then the contents ∫
∫                   of the source file(s) is copied to that    ∫
∫                   directory.                                 ∫
∫                                                              ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $cp contest memo                                             ≥
≥ $                                                            ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

This will cause a copy of the file contest to be made into a file
named memo. If memo doesn't exist, it will be created. If it
already exists, it will be written over. The cp command is
nondestructive; that means that the source file will remain
intact.

The cp command can also be used to copy several files into
another directory.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $cp file1 file2 /user0/teacher                               ≥
≥ $                                                            ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

A copy of file1 and file2 has been sent to the directory (in this
case, the target directory) /user0/teacher. The user of cp will
own the newly copied files.

3.10 Moving Files

A user may move a file only if he has write access to that file.
The mv (move) command can be used to rename one file.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  mv <file1[filen]> <target>                 ∫
∫                                                              ∫
∫    file1[filen] - one or more source files                   ∫
∫                                                              ∫
∫    target       - file or dirname                            ∫
∫                                                              ∫
∫                   file1 and target cannot be the same and    ∫
∫                   if the target is a file its' contents are  ∫
∫                   destroyed.                                 ∫
∫                                                              ∫
∫                   If target is a directory, then the contents ∫
```

```
∫                  of the source file(s) are moved to that      ∫
∫                  directory.                                   ∫
∫                                                               ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
⁄ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ $mv contest memo                                              ≥
≥ $                                                             ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

This will have the effect of changing the name of the file contest into memo. The permissions on the file will remain the same. The move command is destructive. That means the source file no longer exists.

The mv command can also be use to move files from one directory to another.

Sample Session:

```
⁄ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ $mv file1 file2 /user0/teacher                                ≥
≥ $                                                             ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

The files, file1 and file2, have been sent to the directory /user0/teacher. They have been "moved" and no longer reside in the current directory. The owner remains the same when a file is moved.

Workshop 3

This workshop will reinforce your understanding of the topics presented in this chapter. Login to the Multimax using the username and password given to you by the instructor. Each student should complete the entire workshop. You might need to work in a team on the computer exercises.

DESK EXERCISES

1. List four types of files.

2. What does the file command do?

3. The ls command will display the contents of the current working directory. What does the -F option do?

4. What command is used to display the contents of an ordinary file?

5. What command would you use to append one file to the end of another?

6. What is the lp command?

Continue on the next page

7. How can you find out the status of your print job?

8. What command would you enter to cancel a print job called mt_600-1131?

9. What command will copy the contents of one file to another?

10. What does mv do?

11. What do the following file protections indicate?

rwx------

rwxr-xr-x

----

rwxr--r--

Continue on the next page

COMPUTER EXERCISES

12. Log into the Multimax.

13. Execute the file command on the files listed below. Record the output in the space provided.

    a. .profile

    b. /bin/vax

    c. /dev/console

14. Which of the above files is readable?

15. Enter the command to display the contents of the current working directory. Hint: ls

    a. How many files are listed?

    b. Type ls -a

    c. How many entries are listed?

Continue on the next page

    d. Which entries were not listed in your original output of ls?

16. How does the output of ls -a and ls -Ac differ?

Try it.

17. How many fields are displayed for each entry when you execute ls -l? What are the fields?

18. What are the current permissions on .profile?

19. Change permissions on .profile so that no one (including you) has any access to the file.
(Hint: Use the chmod command)

20. Without changing the permissions, list the contents of the file named .profile to the screen.
What happened? Why?

Continue on the next page

21. Change the permissions on .profile to

    u - read, write, execute
    g - read
    o - read

22. Type cat .profile. What happened? Do you know why?

23. Enter pg memo. What does this command do?

24. Send one copy of the file called memo to the laser printer.

25. Logout of the Multimax and the Annex.

NOTES

‹‹

4. DIRECTORIES

A directory is a file whose sole job is to store file names and related information. All files, whether ordinary, special, or directory, are contained in directories.

The directory in which you find yourself when you first login is called your home directory. You will be doing much of your work in your home directory and subdirectories that you'll be creating to organize your files.

4.1 Absolute/Relative Pathnames

As we saw earlier, directories are arranged in a hierarchy with root (/) at the top. The position of any file within the hierarchy is described by its pathname. Elements of a pathname are separated by a /. A pathname is absolute if it is described in relation to root, so absolute pathnames always begin with a /. These are some example of absolute filenames.

```
/etc/passwd
/users/sjones/chem/notes
/dev/rdsk/Os3
```

A pathname can also be relative to your current working directory. Relative pathnames never begin with /. Relative to user sjones' home directory, some pathnames might look like this:

```
chem/notes
personal/res
```

To determine where you are within the filesystem hierarchy at any time, enter the command to print the current working directory.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  pwd                                                ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ $pwd                                                             ≥
≥ /user0/teacher                                                   ≥
≥ $                                                                ≥
¿fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

Notice that this is an absolute pathname. This is the pathname of the current working directory.

4.2 Creating Directories

Directories are created by the following command:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  mkdir [options] <dirname>                    ∫
∫                                                                ∫
∫      options - see man pages for a complete list               ∫
∫                                                                ∫
∫      dirname -  name of the new directory (absolute or         ∫
∫                    relative pathname).                         ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

If the option to change permission mode is not given, the directory will have default permissions set to read,write,execute for the user and read and execute for group and others. The files . (dot) and .. (dot dot) are created automatically. In order to create a sub-directory, you must have write permission on the parent directory. The owner id and the group id are set to the real users id and group id, respectively.

4.3 Removing Directories

Directories can be deleted using the rmdir command.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  rmdir  [options]  <dirname>                  ∫
∫                                                                ∫
∫    options - see man pages for a complete list                 ∫
∫                                                                ∫
∫    dirname - the directory to remove, it must be empty.        ∫
∫                                                                ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $pwd                                                           ≥
≥ /user0/teacher                                                 ≥
≥ $ls -la                                                        ≥
≥ total 5                                                        ≥
≥ drwxr-xr-x 2 teacher  class   512  Jul 18 08:12 .              ≥
≥ drwxrwxr-x 5 root     root    2048 Jul  1 13:14 ..             ≥
≥ $rmdir teacher                                                 ≥
≥ rmdir:teacher:Directory does not exist                         ≥
≥ $                                                              ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

Normally, directories are deleted using the rmdir command. Before the directory can be removed, it must be empty; that is, it must not contain any files. Notice that in the above example two files are present, . (dot) and .. (dot). Remember, these refer to the current working directory and its parent. They cannot be removed.
Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $rmdir .                                                       ≥
≥ rmdir: .: Can't remove current directory or ..                 ≥
≥ $                                                              ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

For the purposes of deleting a directory, the directory is empty if it contains only two files, namely . (dot) and .. (dot dot).

4.4 Changing Directories

To "move around" in the filesystem, use the cd (change directory)
command.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  cd [dirname]                                  ∫
∫                                                                 ∫
∫  dirname  -    If not specified, the value of the $HOME         ∫
∫                shell variable will be used as the new           ∫
∫                current working directory.                       ∫
∫                                                                 ∫
∫                If the directory given is an absolute pathname   ∫
∫                that directory is the new current working        ∫
∫                directory.  A relative pathname can also be      ∫
∫                given.                                           ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $cd /user0/teacher                                              ≥
≥ $pwd                                                            ≥
≥ /user0/teacher                                                  ≥
≥ $                                                               ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

The current working directory is now /user0/teacher.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $cd memos                                                       ≥
≥ $pwd                                                            ≥
≥ /user0/teacher/memos                                            ≥
≥ $                                                               ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

This command will look for a subdirectory called memos under the
current working directory. If it is found, it will become the
new working directory; otherwise, an error will occur.

Error messages beginning with "cannot access file..." often
indicate that the pathname is incorrect or misspelled.

4.5 Renaming Directories

The mv (move) command can also be used to rename a directory.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  mv <dirname> <target>                         ∫
∫                                                                 ∫
∫    dirname   - name of the source directory                     ∫
∫    target    - target directory name                            ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $mv users newusers                                              ≥
≥ $                                                               ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

This will have the effect of changing the name of the directory

users into newusers. The permissions on the directory will
remain the same.

NOTE: All files and subdirectories in the directory newusers
now have new absolute pathnames.

4.6 The directories . (dot) and .. (dot dot)

The filename . (dot) represents the current working directory;
and the filename .. (dot dot) represent the directory one level
above the current working directory, often referred to as the
parent directory. If we enter the command to show a listing of
the current working directories files and use the -a option to
list all the files and the -l option provides the long listing,
this is the result.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $ls -la                                                          ≥
≥ total 975                                                        ≥
≥ drwxrwxr-x  4 teacher   class   2048  Jul 16 17.56 .             ≥
≥ drwxr-xr-x 60 root              1536  Jul 13 14:18 ..            ≥
≥ ----------  1 teacher   class   4210  May  1 08:27 .profile      ≥
≥ -rwxr-xr-x  1 teacher   class   1948  May 12 13:42 memo          ≥
≥ $                                                                ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

The ls -la command displays access modes, number of links, the
owner, the group, size, etc. of files in a directory; but also
displays the characteristics of the current working directory and
its parent. The first entry is the entry for the current
directory. The owner is teacher and the group is class. The
second entry is the parent directory. It is one level up from
the current working directory. It is owned by the root
directory.

Instead of asking for information on all of the files in a
directory, you can request just the information on the current
working directory.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $ls -ld                                                          ≥
≥ drwxrwxr-x  4  teacher   class  2048 Jul 16 17:56 .           ≥
≥ $                                                                ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

The response from the command simply shows the long information
for the current working directory . (dot).
Information can also be obtained for the parent of the current
working directory by using its name as an argument.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $ls -ld ..                                                       ≥
≥ drwxr-xr-x  60  root  root  1536  Jul 13 14:18 ..               ≥
≥ $                                                                ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

Here's the long list of the current working directories parent. (.. is the shorthand representation of the current working directories parent)

Both of the directory names . (dot) and .. (dot dot) can be used as arguments to commands. To change the parent of the current working directory into the current working directory, the command is:

Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ $pwd                                                      ≥
≥ /user0/teacher                                           ≥
≥ $cd ..                                                   ≥
≥ $pwd                                                     ≥
≥ /user0                                                   ≥
≥ $                                                        ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

The current working directory is the former parent.

This is all very interesting but what good is it? You can specify the current working directory or its parent without typing the entire absolute pathname. It can also be handy when giving arguments to UNIX commands.

Why are the pathnames sjones/chem and ./sjones/chem equivalent?

4.7 Directory Access Modes

Directory access modes are listed and organized in the same manner as any other file. There are a few differences that need to be mentioned.

4.7.1 Read

Access to a directory means that the user can read the contents. The user can look at the filenames inside the directory.

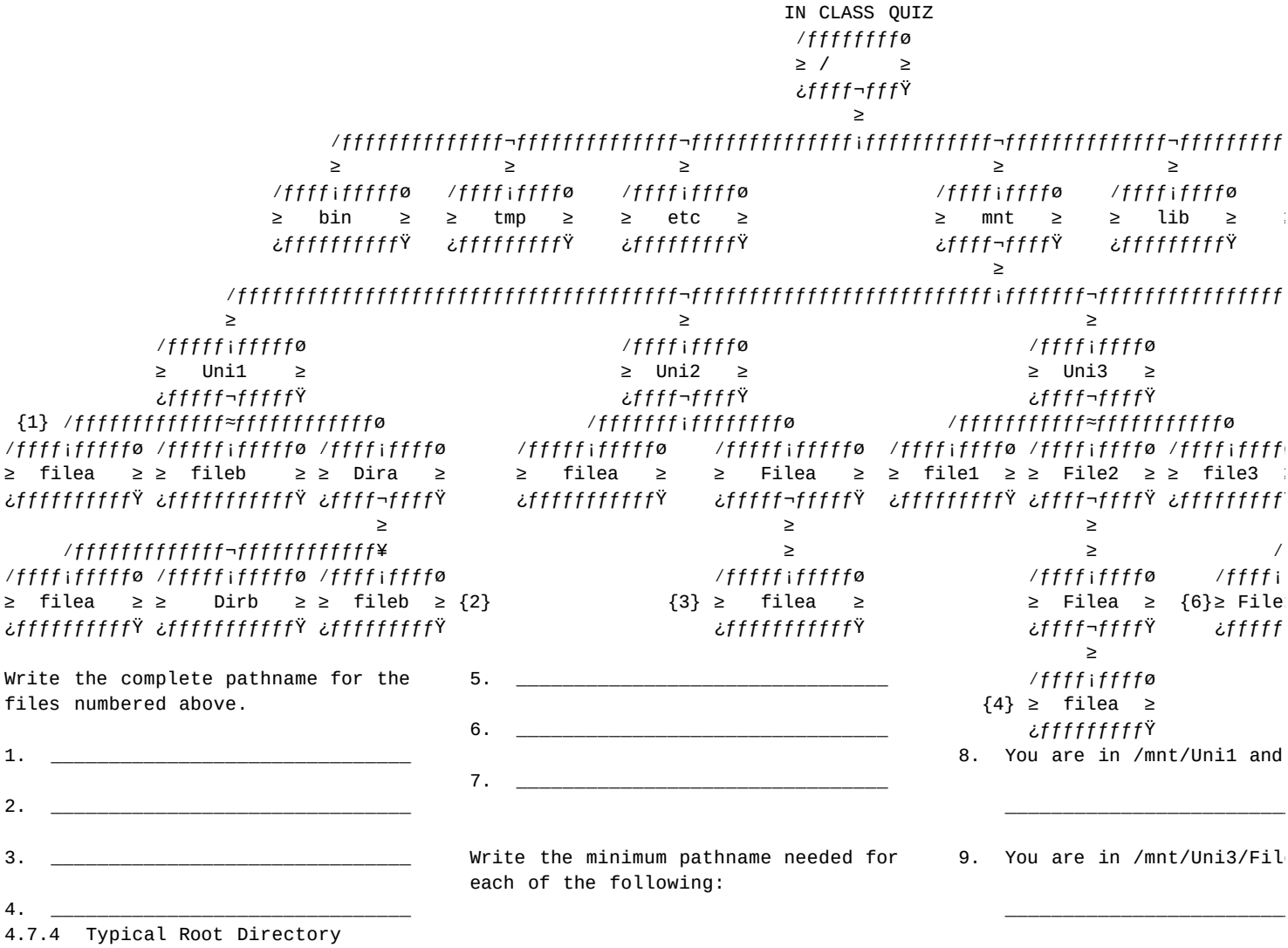
4.7.2 Write

Access means that the user can add or delete files to the contents of the directory.

4.7.3 Execute

Executing a directory doesn't really make a lot of sense so think of this as a traverse permission. This access allows the user to reference the directory name in a command. The reference is not necessarily explicit, since the shell deduces the absolute pathname of a command from the user's environment. For example, the shell knows that the full pathname of the ls command is /bin/ls. A user must have execute access to the bin directory in order to execute ls.

If traverse permissions are denied, others cannot change to it or through it. Another user can't do a cd to the protected directory or any subdirectory beneath it.



4.7.4 Typical Root Directory

```
$ ls -FC /
Student/          bin/            lib/          stand/          u2/         user2/
Students/         bad/            lisp/         tmp/            unix*       usr/
Support/          dev/            lost+found/   tmp.sh          unix.bak*   usr2/
etc/              mnt/            tmp1/         unix.test*      usr3/       a.out*
foo               rel_notes       tmp2/         user0/          install/    shlib/
u1/               user1/
```

NOTES

Workshop 4

This workshop will reinforce your understanding of the topics presented in this chapter. Login to the Multimax using the username and password from the previous workshop. All students

should complete the entire workshop. You may need to work in a team on the computer exercises.

DESK EXERCISES

1. What is a directory?

2. What is an absolute path name?

3. What is a relative path name?

4. What command will create a directory?

5. What command will remove a directory?

6. What command is used to change from one directory to another?

7. How would you change the name of a directory?

Continue on the next page

8. What do the files . (dot) and .. (dot dot) represent?

9. What does execute permission on a directory mean?

COMPUTER EXERCISES

10. Login to the Multimax.

11. What is the absolute pathname of your current working directory? Hint: pwd

12. Type cd etc

    What message do you get? Can you explain why?

13. Type cd /etc

    What is your current working directory? Why did this happen?

14. Enter the command that will return you to your home directory.

Continue on the next page

15. Enter the command that will change to your current working directories parent.

16. List the contents of your current working directory

17. List the permissions, ownership, size, etc. of your current directories parent.

18. Enter the command to change to your home directory. Create a new subdirectory with a name of your choice.

19. Change the current working directory to the subdirectory you just created.

20. Rename the subdirectory to Student. Is this the same subdirectory as everyone else in the class? Why?

21. Change to your home directory and delete the subdirectory Student.

22. Logout of the Multimax and the Annex.

NOTES

---

5. COMMUNICATION UTILITIES

This chapter will deal with the utilities that allow one user to communicate with another. Some of these utilities require the other user to be logged in and others do not.

The mail utility can be used to send messages to one or more users. It is not necessary for the user that is receiving the message to be logged in. The mail utility delivers the message to a file belonging to the recipient. The user will be notified that a mail message exists. Messages can be saved or deleted and a reply sent.

The talk utility is an interactive session that allows each user to send message simultaneously to each other. Both users must be currently logged in for this utility to work.

The write utility is a one-way communication. It allows you to send a message to another user. The user must be logged in and no reply is possible.

5.1 Sending Electronic Mail

The basic command line format for sending mail is:

```
  Command Format: mailx [options] [user1[usern]]

    options - see man pages for a complete list

    user1[usern] - one or more users to get the mail
                   message
```

The username is the name assigned by the system administrator to a user on the UNIX system (for example, rharding). The username can also include a system name if the recipient is on another UNIX system that can communicate with the sender's (for example, sys2!rharding). Let's assume that the recipient is on the local UNIX system.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $mailx rharding(Ret)                                                  ≥
≥ Subject:                                                              ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

Now enter the subject of your message followed by a (Ret). The cursor will appear on the next line. Simply start typing the message. There is no limit to the length of a message. When you have finished, send it by typing Ctrl-D on a new line.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $mailx rharding(Ret)                                                  ≥
≥ Subject: Work schedule(Ret)                                           ≥
≥ Please check the bulletin board(Ret)                                  ≥
≥ for the new work schedule.(Ret)                                       ≥
≥ Ctrl-D                                                                ≥
≥ $                                                                     ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

The shell prompt on the last line indicates that the message has been queued (placed in a waiting line) and will be sent.

5.2 Reading Mail

To read your mail enter:

Example:

```
.............................................................
. $mailx                                                     .
.............................................................
```

Executing this command places you in the command mode of mailx. If there are no mail messages waiting to be read, you will see the following message on the screen:

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $mailx                                                                ≥
≥ No mail for teacher                                                   ≥
≥ $                                                                     ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

Of course, your username will appear instead of 'teacher'.

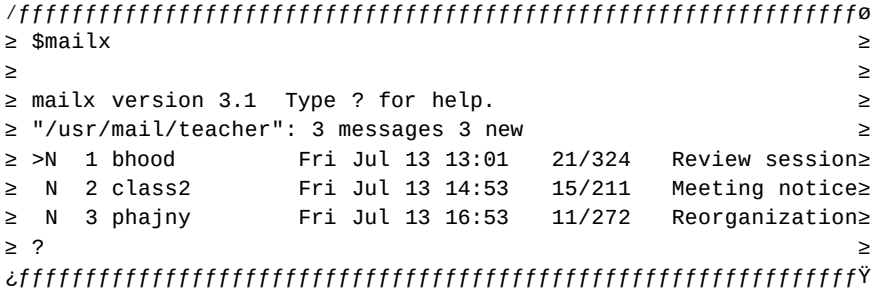
When a mail message appears in the recipient's mailbox, the following message will appear on the screen.

Example:

```
.............................................................
. you have mail                                              .
.............................................................
```

This notice will appear when you login to the system or upon return to the shell from another procedure.
When you have been notified of mail waiting to be read, enter the command to enter mail. The screen will look something like this:

Sample Session:



This first line indicates the version of mailx that is being used. In this case, version 3.1. There is a reminder that help is available by typing the ?. The second line shows the path name of the file used as input (usually the same as the username) and a count of the total number of messages and their status. The messages are numbered in sequence with the latest one received on the bottom of the list. To the left of the sequence numbers, there may be a status indicator; N for new, U for unread. The > symbol points to the current message. The other fields in the header line show the login of the sender, day, date, and time it was delivered. The next field has the number of lines and characters in the message. The last field is the subject of the message; it might be blank.

To read the mail messages you can do any of the following steps:

(Ret)	-	This will cause the current message to be displayed. The current message is the once indicated by the > sign.
p (Ret)	-	This is equivalent to pressing the (Ret) key with no argument. The current message will be displayed.
p 2 (Ret)	-	You can press p (for print) or t (for type) followed by the message number(s).
p teacher (Ret)	-	This will print all messages from user teacher.

5.3 Saving Mail

All messages that are not specifically deleted are saved when quitting mailx. Messages that have been saved are placed in a file in the home directory called mbox. The mbox file is the default. It is possible to save them in a file of the users choice. Messages that have not been read are held in the mailbox. The command to save messages comes in two forms.

Command Format: S [msglist]

msglist =

n	message number n the current message
^	the first undeleted message
$	the last message
*	all messages

```
∫                                                                ∫
∫     n-m        an inclusive range of message numbers           ∫
∫                                                                ∫
∫     user       all messages from user                          ∫
∫                                                                ∫
∫     /string    All messages with string in the subject line    ∫
∫                (case is ignored)                               ∫
∫                                                                ∫
∫     :c         all messages of type c where c is:              ∫
∫                                                                ∫
∫                    d - deleted messages                        ∫
∫                    n - new messages                            ∫
∫                    o - old messages                            ∫
∫                    r - read messages                           ∫
∫                    u - unread messages                         ∫
∫                                                                ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Messages specified by the msglist argument are saved in a file in the current directory named for the author of the first message in the list. If the username 'teacher' sent the message and you entered:

Sample Session:

```
⁄ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ ? S *                                                          ≥
≥ "teacher" [New file] 11/268                                    ≥
≥ ?                                                              ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

The mail message has been saved into a file in your current directory called 'teacher'. If you want to save the file in another filename, you can do that with the second method of saving mail. Basically, it works the same as S; but it allows you to save the mail to a file you specify.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  s  [msglist]  [file1]                        ∫
∫                                                                ∫
∫     msglist - same arguments as before                         ∫
∫                                                                ∫
∫     file1 - filename which will receive the saved mail         ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

5.4 Deleting Mail

To delete a message, enter a d at the command mode prompt followed by a msglist argument. An msglist argument can be any one the following:

n	message number n the current message
^	the first undeleted message
$	the last message
*	all messages
n-m	an inclusive range of message numbers
user	all messages from user
/string	All messages with string in the subject line (case

is ignored)

:c all messages of type c where c is:

d - deleted messages
n - new messages
o - old messages
r - read messages
u - unread messages

For example, suppose you wanted to delete all of your mail messages. Enter the following command at the command mode prompt. The command mode prompt for mailx is the question mark (?).

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $mailx                                                                      ≥
≥                                                                             ≥
≥  mailx version 3.1  Type ? for help.                                        ≥
≥  "/usr/mail/teacher": 3 messages 3 new                                      ≥
≥  >N  1 bhood              Fri Jul 13 13:01   21/324   Review  session      ≥
≥   N  2 class2             Fri Jul 13 14:53   15/211   Meeting notice       ≥
≥   N  3 phajny             Fri Jul 13 16:53   11/272   Reorganization       ≥
≥ ? d *                                                                       ≥
≥ ? q                                                                         ≥
≥ $                                                                           ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

All of the messages have now been deleted. The messages are not actually deleted until the mailbox is exited. Until that happens the u (for undelete) command is available. Once the quit command (q) is entered, however, the deleted messages are gone.

5.5 Undeliverable Mail

If there has been an error in the recipient's username, the mail command will not be able to deliver the message. For example, let's say you misspelled the username. It will return the mail in a message that includes the system name and username of the sender and recipient. It also includes a message stating the reason for the failure.

The sender of the message would get a message from mailx indicating that an error had occurred.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $mailx                                                                      ≥
≥                                                                             ≥
≥ mailx version 3.1  Type ? for help.                                         ≥
≥ "/usr/mail/teacher": 1 message 1 new                                        ≥
≥ >N  1 teacher            Fri Jul 13 13:45   25/655 Returned mail:User unkno≥
≥ ?                                                                           ≥
≥ Message  1:                                                                 ≥
≥ From teacher Fri Jul 13 13:45:57 1990                                       ≥
≥ Received: by domax1.UUCP (5.51/)                                            ≥
≥      id AA01997; Fri, 13 Jul 90 13:45:54 mdt                                ≥
≥ Date: Fri, 13 Jul 90 13:45:54 mdt                                           ≥
≥ From: Mail Delivery Subsystem <MAILER-DAEMON>                               ≥
≥ Subject: Returned mail: User unknown                                        ≥
```

```
≥ Message-Id: <9007131945.AA01997@domax1.UUCP>                              ≥
≥ To: teacher                                                               ≥
≥ Status: R                                                                 ≥
≥                                                                           ≥
≥    ----- Transcript of session follows -----                              ≥
≥ 550 snoopy... User unknown: No such file or directory                     ≥
≥                                                                           ≥
≥    ----- Unsent message follows -----                                     ≥
≥ Received: by domax1.UUCP (5.51/)                                          ≥
≥      id AA01995; Fri, 13 Jul 90 13:45:54 mdt                              ≥
≥ Date: Fri, 13 Jul 90 13:45:54 mdt                                         ≥
≥ From: Teacher Account D-7130 <teacher>                                    ≥
≥ Message-Id: <9007131945.AA01995@domax1.UUCP>                              ≥
≥ To: snoopy                                                                ≥
≥ Subject: Meeting notice                                                   ≥
≥                                                                           ≥
≥ Meeting will be held at Charlie Brown's house.                            ≥
≥ July 13, 1990                                                             ≥
≥ 7:30 p.m.                                                                 ≥
≥                                                                           ≥
≥ ?                                                                         ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

The ? is the mailx command mode prompt. Mailx is asking for
input.

A list of commands available can be shown by entering a ?.

Sample Session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ ? ?                                                                       ≥
≥                         mailx commands                                    ≥
≥ type [msglist]          print messages                                    ≥
≥ next                    goto and type next message                        ≥
≥ edit [msglist]          edit messages                                     ≥
≥ from [msglist]          give header lines of messages                     ≥
≥ delete [msglist]        delete messages                                   ≥
≥ undelete [msglist]      restore deleted messages                          ≥
≥ save [msglist] file     append messages to file                           ≥
≥ reply [message]         reply to message, including all recipients        ≥
≥ Reply [msglist]         reply to the authors of the messages              ≥
≥ preserve [msglist]      preserve messages in mailbox                      ≥
≥ mail user               mail to specific user                             ≥
≥ quit                    quit, preserving unread messages                  ≥
≥ xit                     quit, preserving all messages                     ≥
≥ header                  print page of active message headers              ≥
≥ !                       shell escape                                      ≥
≥ cd [directory]          chdir to directory or home if none given          ≥
≥ list                    list all commands (no explanations)               ≥
≥ top [msglist]           print top 5 lines of messages                     ≥
≥ z [-]                   display next [last] page of 10 headers            ≥
≥                                                                           ≥
≥ [msglist] is optional and specifies messages by number, author,           ≥
≥ or type.                                                                  ≥
≥ The default is the current message.                                       ≥
≥ ?                                                                         ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

This is a partial list of mailx commands available to you. We
will not discuss all of them. If you are interested in the other
features, you can use the on-line manual pages to find out how to
use them.

5.6 Talk Utility

Talk is a visual communication program which copies lines from one terminal to that of another user. This is similar to the phone utility on VMS. Once communication is established between two users, they can both type simultaneously with their output appearing in separate windows.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  talk <user1> [ttyname]                                     ∫
∫                                                                              ∫
∫     user1 -     If you are talking to someone on the same machine,          ∫
∫                 then this is just the person's username.  If                ∫
∫                 you want to talk to a user on another host, then            ∫
∫                 user1 is of the form:                                       ∫
∫                                                                              ∫
∫                 host!user     or                                            ∫
∫                 host.user     or                                            ∫
∫                 host:user     or                                            ∫
∫                 user@host                                                   ∫
∫                                                                              ∫
∫                 user@host being preferred                                   ∫
∫                                                                              ∫
∫     ttyname   - If the person you want to talk to is logged on              ∫
∫                 more than once, you can use the ttyname argument            ∫
∫                 to indicate the terminal name.                              ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

For illustration, let's assume we want to talk with the user student on the same machine. The command is:

Example originator:

```
.................................................................
. $talk student                                                 .
.................................................................
```

Example recipient:

```
.................................................................
. Message from Talk_Daemon@domax1 at 17:36 ...                  .
. talk: connection requested by teacher@domax1.                 .
. talk: respond with:  talk teacher@domax1                      .
.................................................................
```

When the recipient has typed in talk teacher@domax1, the following message will appear on the originators screen:

Sample Session originator:

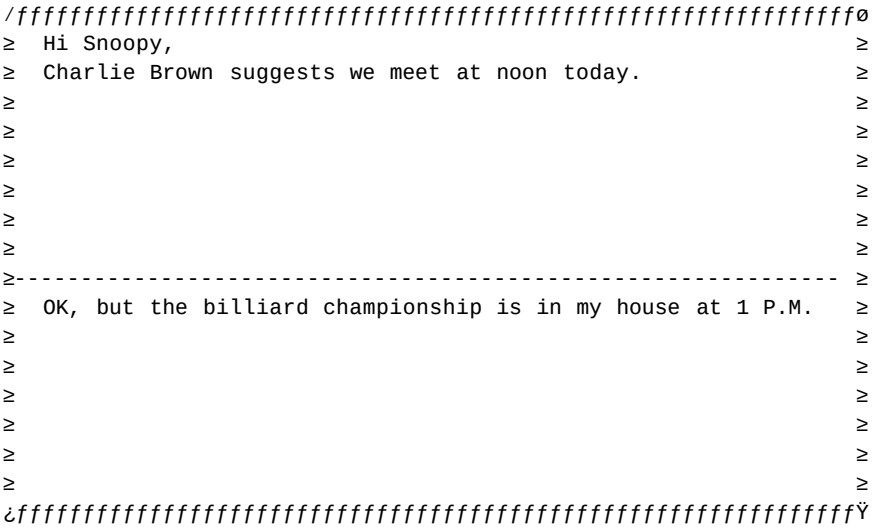
```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥  Connection established.                                      ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

The screen will be divided in half by a row of dash characters. The originator will type a message on the top half, and the same message will appear on the lower half of the screen on the recipient's screen.

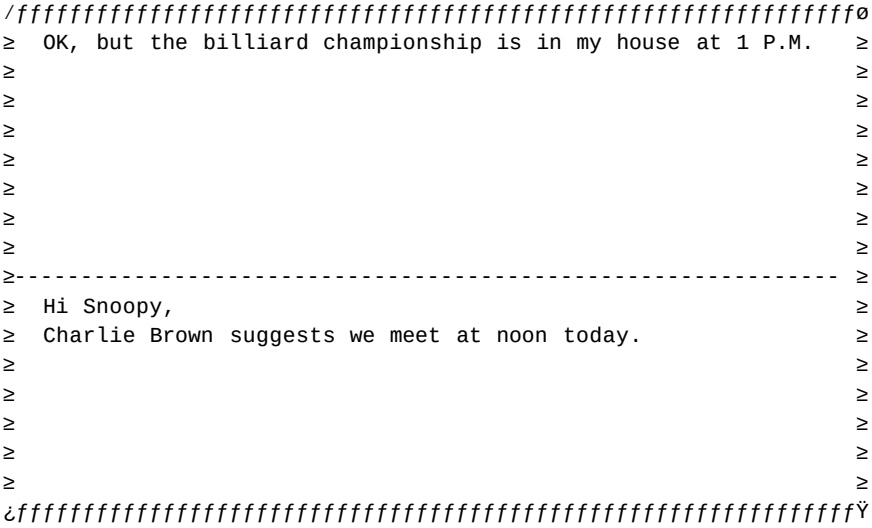
Likewise, everything the recipient types on the top of his screen the same message will appear on the bottom of the originators screen. Once this communication is established, the parties may type simultaneously with their output appearing in different windows. While in talk, Ctrl-L will cause the screen to be reprinted, and the erase and kill characters work as you would

expect.

Sample Session originator:



Sample session recipient:



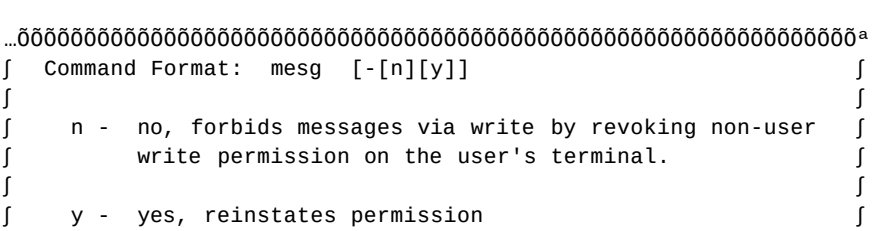
When the communication is finished, the interrupt character will cause the talk utility to exit.

Example:

```
.....................................................................
.     [Connection closing. Exiting]                                 .
.....................................................................
```

5.7 Talk Permission Denied

If you don't wish to have your work interrupted by a request to establish a talk connection, you can deny messages.



```
∫                                                                    ∫
∫                                                                    ∫
∫     mesg with no arguments will report the current state           ∫
∫     without changing it.                                           ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßø
≥ $mesg                                                              ≥
≥ is y                                                               ≥
≥ $mesg -n                                                           ≥
≥ $mesg                                                              ≥
≥ is n                                                               ≥
≥ $mesg -y                                                           ≥
≥ $mesg                                                              ≥
≥ is y                                                               ≥
≥ $                                                                  ≥
¿ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßŸ
```

The default permission is enabled. Some UNIX commands, however, disallow messages in order to prevent messy output.

5.8 Write Utility

This command will write a message to the screen of another user.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  write <user1> [ttyname]                          ∫
∫                                                                    ∫
∫   user1  - username of the user                                    ∫
∫                                                                    ∫
∫   ttyname - which terminal to send (i.e. tty00)                   ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session originator:

```
/ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßø
≥ $write lucy                                                        ≥
≥ Hello Lucy,                                                        ≥
≥ What's the latest from the Psychology Department?                  ≥
≥ (interrupt character)                                              ≥
≥ $                                                                  ≥
¿ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßŸ
```

Sample Session recipient:

```
/ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßø
≥ $                                                                  ≥
≥                                                                    ≥
≥ Message from teacher on domax1 (rt021d0) [ Thu Jul 19 13:43:12 ] ..  ≥
≥ Hello Lucy,                                                        ≥
≥ What's the latest from the Psychology Department?                  ≥
≥ <EOT>                                                              ≥
≥                                                                    ≥
≥ $                                                                  ≥
¿ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßŸ
```

Here's a suggestion for using write to communicate a little easier.

When the user first 'writes' to another user, wait for the recipient to 'write' back before starting to send. Both users

should agree on a signal to indicate to the other person that they can reply. How about 'o' for over. The signal 'oo' could be used for "over and out," which would mean that the communication is finished.

NOTES

‹‹‹

Workshop 5

This workshop will reinforce your understanding of the topics covered in this chapter. Login to the Multimax with the username and password given to you by the instructor. Each student is to complete the entire workshop. Computer exercises might need to be worked as a team.

DESK EXERCISES

1. What is the command to send an electronic mail message to another user on the Multimax?

2. Once you have entered the mail utility what command can you enter to get help?

3. What does the command d 5-9 accomplish?

4. What is the command to exit the mail utility and return to the UNIX system prompt?

5. What is the mailx command mode prompt?

6. How would you create a "talk" session to user Student2 on the host domax0?

7. What time does the billiard championship start?

Continue on the next page

8. What UNIX command will prevent interruption of your work by someone wishing to "talk"?

9. Regarding "write", does the recipient need to be logged in?

Regarding "talk", does the recipient need to be logged in?

Continue on the next page

COMPUTER EXERCISES

10. Login to the Multimax.

11. Send a mail message to another student in the class.

How can you find out who is logged in? (who?)

Does the recipient need to be logged in?

12. Send a mail message to username lucy. (lucy does not exist)

    What happened? Why?

13. Read your mail and save one message to the current working directory.

    Delete all other mail messages.

Continue on the next page

14. Establish a talk connection with another student.

15. What UNIX command do you enter to deny permission for a talk connection? Try it!

16. Send a message to another student using the write command.

    How is this different from "talk?"

17. Logout of the Multimax and the Annex.

6. SHELL BASICS

There have been several shells written for UNIX. They have different features and each is in use through out the world. The BourneShell is the accepted standard for System V UNIX. Another shell is called the Cshell, named for "C" which is the high-level programming language. Another shell is the KornShell; it is named after the person who developed it, David Korn. It has more features than the BourneShell and is of special interest to programmers.

The purpose of this chapter is to give you some idea as to the functions available through the shells and their general function. Details of shell programming are discussed in another class, "UNIX Bourne Shell Programming".

UMAX makes full use of the ASCII character set. Unlike operating system command languages like VMS or NOS, UNIX is case sensitive. In addition, several characters have special meanings to the shell. We have already seen that a slash (/) by itself indicates the root directory and is used with directory, subdirectory, and filenames to indicate an absolute or relative pathname.

Other special characters that have meaning to the shell include:

` ' $ { } || && ;

Input to a command is usually taken from your keyboard, and the output of a command is normally displayed on your monitor screen. Keyboard input is referred to as "standard input" or "stdin," and screen output as "standard output" or "stdout."

6.1 Input Redirection

It is possible to instruct UNIX to get data from a file rather than from the keyboard. This is called input redirection. To indicate that input to a command is to come from a file rather than the keyboard, use the input redirection character (<).

```
Command Format:  command < input-file1

   command   - a command

 input-file1 - input file that supplies input
                to the command
```

A Memory Trick: The less-than symbol looks like a funnel. If you pour liquid into the wide end, it flows to the narrow end. The input-file "pours" its contents into the command.

Sample Session:

```
$mailx phajny < report
$
```

The file named report will be sent to the login name phajny. Mail normally expects the input to come from standard input, the keyboard. The input redirection symbol causes the input to mail to come from the file called report.

6.2 Output Redirection

It is also possible to instruct UNIX to send data to a file rather than sending it to the default monitor screen. This is called output redirection. To indicate that the output from a command is to go into a file rather than be displayed on the monitor screen, use the output redirection character >.

```
Command Format:  command > output-file1

     command    - a command
```

```
∫                                                                ∫
∫    output-file1 - output file that will receive the output    ∫
∫                   from the command                            ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

The memory trick still works; only now the funnel points toward the file that will receive the output.

Sample Session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ $ls -l > listing                                              ≥
≥ $                                                             ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

The output of the ls command will not be displayed on the screen, instead it will be in the file named listing. If the file does not exist, the shell will create it. If it already exists, it will be overwritten.

WARNING: The shell will NOT issue a warning about overwriting the original file.

It is possible to use the cat command to create a file and input text into that file using output redirection. The following example shows how this can be done.

Sample session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥    $cat > file1                                               ≥
≥      This is a line of text.                                    ≥
≥    This is another line of text.                              ≥
≥    (Ctrl-D)                                                   ≥
≥    $cat file1                                                 ≥
≥    This is a line of text.                                    ≥
≥    This is another line of text.                              ≥
≥    $                                                          ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

6.3 Output Redirection with Append

The following shell command will also redirect the output to a file but instead of overwriting the existing file, it will append the output to the end of output-file.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  command >> output-file1                     ∫
∫                                                                ∫
∫    command      - a command                                    ∫
∫                                                                ∫
∫    output-file1 - receives the output from command            ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Believe it or not, the memory trick still works; only in this case, one funnel feeds onto another. So the output is fed onto the end of output-file. Okay, it's a little far fetched; but it can help you remember. Try it.

Sample Session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ $ls -l subdir >> listing                                      ≥
≥ $                                                             ≥
```

```
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

This will append the output of the ls command to the file listing
without destroying any existing data. If the file does not
exist, the shell will create it.

Again, it's possible to append text to the end of an existing
file using the cat command. Note the following example.

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥    $cat >> file1                                              ≥
≥      This is a third line of text.                              ≥
≥    This is a fourth line of text.                             ≥
≥    (Ctrl-D)                                                   ≥
≥    $cat file1                                                 ≥
≥    This is a line of text.                                    ≥
≥    This is another line of text.                              ≥
≥    This is a third line of text.                              ≥
≥    This is a fourth line of text.                             ≥
≥    $                                                          ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

If the file does not exist it will be created and the text added.

6.4 Input and Output Redirection

Input and output redirection can occur on the same command line.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  command  < input-file1 >  output-file1      ∫
∫                                                               ∫
∫    command       - A command                                  ∫
∫    input-file1   - supplies input to command                  ∫
∫    output-file1  - receives the output from command           ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ $cat command_file                                                            ≥
≥ p                                                                            ≥
≥ $mailx < command_file > result_file                                          ≥
≥                                                                              ≥
≥ $cat result_file                                                             ≥
≥ mailx version 3.1  Type ? for help.                                          ≥
≥ "/usr/mail/teacher": 1 message 1 new                                         ≥
≥ >N  1 teacher            Mon Dec 31 10:16   57/3171                          ≥
≥ Message  1:                                                                  ≥
≥ From teacher Mon Dec 31 10:16:30 1990                                        ≥
≥ Received: by domax1.UUCP (5.51/)                                             ≥
≥         id AA18976; Mon, 31 Dec 90 10:16:28 mst                              ≥
≥ Date: Mon, 31 Dec 90 10:16:28 mst                                            ≥
≥ From: Teacher Account D-7130 <teacher>                                       ≥
≥ Message-Id: <9012311716.AA18976@domax1.UUCP>                                 ≥
≥ To: teacher                                                                  ≥
≥ Status: R                                                                    ≥
≥                                                                              ≥
≥ What's Happening                                                             ≥
≥ by Pam Hajny                                                                 ≥
≥ Denver Office                                                                ≥
≥                                                                              ≥
≥ With IRM Training:                                                           ≥
```

≥   ≥
≥ A Reclamation-wide workshop was held in early October to discuss information ≥
≥ resources management training. Trainers from each region and the Denver Offic≥
≥ shared training techniques, ideas and course materials. We met one afternoon ≥
≥ with the personnel training officers to discuss broad IRM training needs and ≥
.   .
.   .
.   .

6.5 Pipes

The output of a command can be used as the input to a second command by using the "pipe" symbol (|) without using any temporary files. On some terminals the pipe symbol is a vertical bar and on others it is a broken vertical bar. Both will work exactly the same. The following command format shows how to use the pipe symbol:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  command1 | command2                          ∫
∫                                                                 ∫
∫    command1  - a command                                        ∫
∫                                                                 ∫
∫    command2  - a second command                                 ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Example:

```
.................................................................
. $man acct | pg                                                .
.................................................................
```

The output from the command man are processed by the pg command before appearing on your screen. Normally the output from the man command will appear on the monitor line after line until the end is reached. In this case, the output is "piped" to the pg command; and the screen will stop scrolling after 23 lines so you can read them.

6.6 Wildcards

Wildcards are special characters that cause the shell to search over a range of possible values.

? represents any one character, while

* stands for any number of characters including none.

Example:

```
jo?eph
```

This indicates that the third letter of the string "jo eph" could be any single character. Any character could be substituted for the ? character, including numeric and special characters.

To limit the range of possible values, enclose the possibilities in brackets [ ].

Example:

```
jo[a-z]eph
```

This example limits the range of characters to the set lowercase a through lowercase z. Uppercase characters, numeric, or special characters would not make a match. Notice that only one charater will make a match.

Using a comma as a separator between choices we can further restrict the range.

Example:

```
jo[s,m,5]eph
```

The only set of characters that will make a match are lowercase s, lowercase m, and the number 5. No other character will make a match.

The string jos* causes the shell to look for every string that begins with the letters "jos," regardless of their length while [i-k]*h finds every string that begins with "i", "j", or "k" and ends with an "h".

Wildcards are extremely useful in wide variety of applications. For example, if you want to use the man pages, but do not know the exact command names on the subject of system accounting, try

Sample Session:

```
$man acc*
```

All of the commands that begin with the letters acc followed by any string (including none) will be passed to the man command as arguments.

If you wanted to get a listing of all the files in your current working directory that ended in .c (these are the C source code programs). You could enter the following command:

Sample Session:

```
$ls *.c
```

In order for the shell to stop interpretation of a special character (i.e., use it as a normal character), it must be preceded by a backslash (\) or enclosed in single quotes.

Example:

```
jo\?eph
  or
'jo?eph'
```

Both of these examples represent the string jo?eph. The shell will not interpret the question mark character as a wildcard metacharacter.

6.7 Reestablishing a Background Job

Processes in UNIX can run in the foreground or the background. Foreground processes are interactive; the input is read from the keyboard or standard in, and the out goes to the monitor screen or standard out. Background jobs run with no interaction with an interactive terminal. Your current interactive process can be suspended by typing the break character at the shell prompt.

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $ <break>                                                                 ≥
≥ annex:                                                                    ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

The jobs command displays information on all current jobs (sessions). The most recent job is marked with a plus sign (+), and the next previous is marked with a dash or minus sign (-). A job begins when you execute a command to connect to a host (or another Annex). A job ends when you logout from the host or terminate the job at the Annex with the kill or hangup command.

The number of possible jobs allowed per user is determined by the network administrator. The number of jobs can range from 1 to 16 with a default of 3.
The Annex command to display the information about the current job(s) is:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  jobs                                                   ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

If there are no jobs, the annex: prompt will be displayed. If there are some 'suspended' jobs the following will appear:

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ annex: jobs                                                               ≥
≥ +1      rlogin domax1                                                     ≥
≥ -2      rlogin domax1                                                     ≥
≥ annex:                                                                    ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

This shows that there are two jobs in suspension. Both of these sessions did a remote login to domax1. This is just for illustration.
The fg (foreground) command returns to a suspended job. The command displays the job number and the Annex command that created it. When no arguments are provided, fg will return to the most recent job. With a numeric argument, fg returns the specified job.

To connect with a suspended job (session) enter the following Annex command:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  fg [n]                                                 ∫
∫                                                                          ∫
∫     (none) -   most recent job (+) to foreground                         ∫
∫       n    -   job "n" to foreground                                     ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ annex: jobs                                                  ≥
≥ +1      rlogin domax1                                        ≥
≥ -2      rlogin domax1                                        ≥
≥ annex:fg 1                                                   ≥
≥ 1       rlogin domax1                                        ≥
≥ (Ret)                                                        ≥
≥ $                                                            ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

NOTES

‹‹‹

Workshop 6

This workshop will reinforce your understanding of the topics covered in this chapter. Login to the Multimax with the username and password given to you by the instructor. Each student is to complete the entire workshop. Computer exercises might need to be worked as a team.

DESK EXERCISES

1. What is the meaning of the term "case sensitive?"

2. What is a wildcard?

3. How does the shell interpret the following wildcards?

   a. ?

   b. [0-9]

   c. *

4. How does the shell interpret the following strings?

   a. M[i,r]*

   b. b?ll

   c. me??[1,2]

   d. '*special*'

   e. anyone\?

Continue on the next page

5. What is "standard input?"

6. What symbol causes a command to take its input from a file?

7. What is "standard output?"

8. What symbol causes the output of a command to be redirected to a file?

9. What symbol causes the output of a command to be redirected to the input of another command?

10. What symbol is used to indicate input is to be from a file instead of the keyboard?

11. How can the output from a command be saved in an ordinary file?

Continue on the next page

12. What is a pipe? No, it's not something you smoke.

COMPUTER EXERCISES

13. Login to the Multimax

14. How many different on-line manual entries are displayed by executing the command man ca*?

15. Execute man ls | pg. What is the purpose of the | character?

16. Save the on-line manual pages on the cat command in a file called mp0. (hint: output redirection)

17. Save the on-line manual pages on the assist command in a file called mp1. (no hint this time)

18. Type cp mp0 man

    Does file mp0 still exist after this command is executed?

    Why?

Continue on the next page

19. Type mv mp1 assist

    Does file mp1 still exist after this command is executed?

    Why?

20. Type cp mp3 man

    What error message do you get?

21. Logout of the Multimax and the Annex.

7. UMAX FILE TRANSFER PROTOCOL (FTP)

File Transfer Protocol (FTP) is a utility which can transfer files to and from TCP/IP networked computers. TCP/IP stands for Transmission Control Protocol/Internet Protocol and consists of a suite of defacto standard protocols for networking computers. FTP is one protocol in that suite. (Other significant protocols

within TCP/IP are TELNET, Simple Mail Transfer Protocol (SMTP), and Network File Systems (NFS).) The Client portion of UNIX FTP lets users on the Multimax access file systems on a remote computer. The Server portion of UNIX FTP lets users on remote computers access Multimax files. For Reclamation, these remote computers would be VAXes, CYBERs, IBMs, and Sun workstations.

Using FTP, you can access directories and files on a remote computer and perform common operations, such as list and change working directories, transfer files, create directories, delete working directories, delete files and directories, and rename files and directories. Once you have entered the FTP utility, you make a connection to the desired remote computer and then work with the remote computer's files using FTP commands. The connection to the remote computer's FTP remains in effect until terminated by the user. Multimax FTP supports both local help for FTP commands and remote help, which displays FTP elements available on the remote computer.

Throughout this chapter, the term "local computer" will refer to the Multimax. The term "remote computer" will refer to the CYBER mainframe or the VAX minicomputer. Please be aware that these procedures will work for any computer connected to the Ethernet that has an FTP server installed. The messages that appear may be different, but the process will be the same.

7.1 Initializing FTP on UMAX

FTP can be invoked on the Multimax using the following syntax:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫                                                                    ∫
∫       Command Format:      ftp [options] [host]                    ∫
∫                                                                    ∫
∫            options - see man pages for a complete list               ∫
∫                                                                    ∫
∫           host   - the name of the remote computer                 ∫
∫                                                                    ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

NOTE: UNIX is case sensitive. The commands and options must be entered as shown.

7.2 Establishing Connection with the Remote Computer

There are two ways to make a connection with the remote computer.

7.2.1 Calling FTP with no hostname

The first way is to invoke FTP using no options, simply enter the ftp command at the shell prompt. UMAX will respond with the ftp prompt: ftp>

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $ftp                                                               ≥
≥ ftp>                                                               ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

FTP commands can now be entered. The utility has its own set of commands, and we will discuss about 12 of them in this chapter. A complete list of the FTP commands can be obtained by entering help at the FTP prompt.

The command to establish a connection with remote computer is:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫                                                                    ∫
∫     Command Format: open <host> [port]                             ∫
∫                                                                    ∫
∫     host - hostname, this host must have an FTP server.            ∫
∫                                                                    ∫
∫     port - port number (optional)                                  ∫
∫                                                                    ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

This command will establish a connection to the remote computer's FTP server. The hostname for the VAX is ERC830. The following FTP command will establish a connection with the VAX (ERC830):

VAX Sample Session:

```
/ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄø
≥ ftp>open erc830                                                    ≥
≥ Connected to erc830.                                               ≥
≥ 220 erc830 Wollongong FTP Server (Version 5.0) at Mon Dec 4        ≥
≥ Name (ERC830:rharding):                                            ≥
¿ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄŸ
```

The cursor will stop after the colon. FTP is waiting for you to enter the login name to use when signing on to the remote computer. FTP tries to help you out by giving you a default login name. In the above example, the default login name is rharding. To select the default name, press (Ret). You can enter any login name you want and then press (Ret). After you have selected the login name, either by choosing the default or entering a new name, you will be asked for the password.

VAX Sample Session:

```
/ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄø
≥ 331 Password required for rharding.                                ≥
≥ Password:                                                          ≥
≥ 230 User logged in, default directory D_1131:[RHARDING]             ≥
¿ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄŸ
```

Enter the password required for the login name that you specified. Echoing is disabled and the password you enter will not be displayed on the screen. If you entered the correct password, message number 230 will show you are logged in and the default directory on the remote system. You are now logged into the remote computer system and can proceed to transfer files.

CYBER Sample Session:

```
/ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄø
≥ $ftp                                                               ≥
≥ ftp>open cy2                                                       ≥
≥ Connected to cy2.                                                  ≥
≥ 220 SERVICE READY FOR NEW USER.                                    ≥
≥ Name (cy2:rharding): class8                                        ≥
≥ 331 USER NAME OKAY, NEED PASSWORD.                                 ≥
≥ Password: secret                                                   ≥
≥ 230 USER LOGGED IN, PROCEED.                                       ≥
≥ ftp>                                                               ≥
¿ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄŸ
```

This example for the CYBER is similar to the VAX example. Notice

that there a few differences. The login name was changed from rharding and the username class8 was entered instead.

7.2.2 Calling FTP with a hostname

The second method of signing on to the remote computer is to specify the name of the remote computer on the call to ftp.

VAX Sample Session:

```
$ftp erc830
220 erc830 Wollongong FTP Server (Version 5.0) at Fri Dec
Name (ERC830:rharding):
```

You can now enter the username for the remote system, and you will then be prompted for the password. The effect of specifying the hostname on the ftp command line is to do an "automatic" open command.

NOTE: The messages are slightly different from the VAX login. The login for the CYBER works in a similar manner.

7.3 Local Computer Commands

From the FTP prompt, you can issue commands to the local computer to display files or show the contents of a directory. The commands you enter are FTP commands; and although they might resemble UNIX commands, they are NOT UNIX commands.

The FTP command to transfer file(s) from the remote computer to the local computer is as follows:

```
Command Format:  get <remote-file> [local-file]

    remote-file - the filename on the remote computer

    local-file  - the filename on the local computer
```

This FTP command will retrieve the remote-file and store it on the Multimax. If the local-file name is not specified, the name of the file on the Multimax will be the same as it was on the remote computer. The current settings for type, form, mode, and structure will be used during the file transfer.

VAX Sample Session:

```
ftp>get overview.dat
200 PORT Command OK.
125 File transfer started correctly
226 File transfer completed ok
local: overview.dat remote: overview.dat
884 bytes received in 0.04 seconds (22 Kbytes/s)
ftp>
```

Messages 200, 125, and 226 let you know that the file transferred properly. The next line shows the local-filename, in this case we didn't specify the local-filename, so the remote-filename and

the local-filename are the same. The next line shows the number of bytes transferred and the amount of time it took to transfer the file.

CYBER Sample Session:

```
⁄ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ ftp>get prolog8                                               ≥
≥ 220 COMMAND OKAY.                                             ≥
≥ 150 FILE STATUS OKAY; ABOUT TO OPEN DATA CONNECTION.          ≥
≥ 226 CLOSING DATA CONNECTION.                                  ≥
≥ local: prolog8 remote: prolog8                                ≥
≥ 41 bytes received in 0.8 seconds (0.05 Kbytes/s)              ≥
≥ ftp>                                                          ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

7.3.1 Changing the Local Directory

The directory on the local computer can be changed to any directory you desire. This is called the working directory. This is the directory where files that are transferred from the remote computer will be stored.

The syntax of the command to change local working directory is as follows:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫                                                               ∫
∫  Command Format:  lcd [dirname]                               ∫
∫                                                               ∫
∫     dirname - the name of the new local working directory     ∫
∫                                                               ∫
∫    if directory is omitted, the home directory is assumed     ∫
∫                                                               ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
⁄ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ ftp>lcd /user0/student0                                       ≥
≥ Local directory now /user0/student0                           ≥
≥ ftp>                                                          ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

Absolute or relative pathnames can be specified for directory.

7.3.2 Listing the Contents

Any UNIX command can be entered from the FTP utility. You must preface the command with the FTP command that invokes the interactive shell.

The syntax to invoke the interactive shell is as follows:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫                                                               ∫
∫  Command Format:  ! [command [arguments]]                     ∫
∫                                                               ∫
∫     command - any valid UNIX command, if omitted the          ∫
∫               interactive shell is invoked                    ∫
∫                                                               ∫
∫    arguments - if supplied are arguments to the UNIX command  ∫
∫                                                               ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

If arguments are provided, the first argument is considered to be

the UNIX command and the remaining arguments are considered to be arguments to that command.

Example:

```
. ftp>!ls -la                                                  .
```

This command will display the contents of the local working directory. The l option specifies the 'long' listing, and the a option requests all files including the initialization files.

7.4 Remote Computer Commands

From the FTP prompt, you can issue commands to the remote computer to display files or show the contents of the remote directory. Recall that the commands you enter are FTP commands; and although they look like UNIX commands, they are not.

Transferring file(s) from the Multimax to the remote computer is accomplished with the following command:

```
∫                                                                ∫
∫  Command Format:  put <local-file> [remote-file]               ∫
∫                                                                ∫
∫       local-file   - the filename on the local computer        ∫
∫                                                                ∫
∫       remote-file  - the filename on the remote computer       ∫
∫                                                                ∫
```

This FTP command will retrieve the local-file, transfer it to the remote computer, and store it in the remote directory. If the remote-file is not specified, the name of the file on the remote computer will be the same as it was on the Multimax. The current settings for type, form, mode, and structure will be used during the file transfer.

VAX sample sessions:

```
≥ ftp>put memo                                                 ≥
≥ 200 PORT Command OK.                                         ≥
≥ 125 File transfer started correctly                          ≥
≥ 226 File transfer completed ok                               ≥
≥ local: memo remote: memo                                     ≥
≥ 2299 bytes sent in 0.08 seconds (28 Kbytes/s)                ≥
≥ ftp>                                                         ≥
```

Messages 200, 125, and 226 let you know that the file transferred properly. The next line shows the local-filename. In this case, we didn't specify the local-filename, so the local-filename and the remote-filename are the same. The next line shows the number of bytes sent and the amount of time for the transfer.

CYBER Sample Session:

```
≥ ftp>put memo                                                 ≥
≥ 200 COMMAND OKAY.                                            ≥
≥ 150 FILE STATUS OKAY; ABOUT TO OPEN DATA CONNECTION.         ≥
```

```
≥ 226 CLOSING DATA CONNECTION.                                    ≥
≥ local:memo remote:memo                                          ≥
≥ 2299 bytes sent in 0.08 seconds (28 Kbytes/s)                   ≥
≥ ftp>                                                            ≥
```

7.4.1 Changing the Remote Directory

The directory on the remote computer can be changed to any directory you want. This is called the remote working directory. This is the directory where files that are sent from the Multimax will be stored.

The syntax for the command to change remote working directory is as follows:

```
Command Format:  cd <remote-dirname>

      remote-dirname - the name of the new remote working
                       directory
```

VAX Sample Session:

```
≥ ftp>cd d_1131:[gholdaway]                                       ≥
≥ 200 Working directory changed to D_1131:[GHOLDAWAY]             ≥
≥ ftp>                                                            ≥
```

You must specify a valid directory on the remote computer.

CYBER Example:

```
. 502 COMMAND NOT IMPLEMENTED.                                    .
```

The reason this command is not implemented on the CYBER is because NOS does not support the idea of directories.

7.4.2 Listing the Contents

```
Command Format:  ls [remote-dirname] [local-file]

      remote-dirname - working directory on remote computer

      local-file  - local file where the remote-directory
                    contents will be written.  If omitted,
                    the output is sent to the screen.
```

VAX Sample Session:

```
≥ ftp>ls                                                          ≥
≥ 200 PORT Command OK.                                            ≥
≥ 125 File transfer started correctly                             ≥
≥ login.com;13                                                    ≥
≥ jeff.;1                                                         ≥
```

```
≥ test.com;1                                                          ≥
≥ 226 File transfer completed ok                                      ≥
≥ 228 bytes received in 0.06 seconds (0.34 Kbytes/s)                  ≥
≥ ftp>                                                                ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

Since no remote directory was specified, the contents of the current working directory is transferred and no local file was specified, so the output is displayed on the screen.

CYBER Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ ftp>ls                                                              ≥
≥ 200 COMMAND OKAY.                                                   ≥
≥ 150 FILE STATUS OKAY; ABOUT TO OPEN DATA CONNECTION.                ≥
≥ PROLOG8                                                             ≥
≥ FSEP1A                                                              ≥
≥ FSEP1                                                               ≥
≥ FSEP2                                                               ≥
≥ 226 CLOSING DATA CONNECTION.                                        ≥
≥ 52 bytes received in 1 seconds (0.05 Kbytes/s)                      ≥
≥ ftp>                                                                ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

7.5 Closing the Connection

The current FTP session with the remote server can be terminated without leaving FTP. When the current session is terminated a session to another remote FTP server can be initiated.

```
…õõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõª
∫  Command Format:  close                                             ∫
»õõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõº
```

This command will terminate the current FTP session with the remote server and return to the FTP command interpreter.

VAX Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ ftp>close                                                           ≥
≥ 221 Goodbye.                                                        ≥
≥ ftp>                                                                ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

CYBER Sample Session:

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ ftp>close                                                           ≥
≥ 221 SERVICE CLOSING CONTROL CONNECTION. LOGGED OUT.                 ≥
≥ ftp>                                                                ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

7.6 Exiting FTP

When you have finished using FTP, the following command will terminate FTP and return control to the shell.

```
…õõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõª
∫  Command Format:  quit                                              ∫
»õõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõõº
```

This command will terminate the current FTP session and exit FTP.

VAX Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ ftp>quit                                                                ≥
≥ 221 Goodbye.                                                           ≥
≥ $                                                                      ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

CYBER Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ ftp>quit                                                                ≥
≥ 221 SERVICE CLOSING CONTROL CONNECTION. LOGGED OUT.                     ≥
≥ $                                                                      ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

7.7 Special FTP Commands

This section will discuss some FTP commands that are useful in using FTP. They include an on-line help, status, and the ! character.

The help command will display all of the FTP commands on the screen.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫                                                                        ∫
∫  Command Format:  help [command]                                       ∫
∫                                                                        ∫
∫      command  - an FTP command                                         ∫
∫                                                                        ∫
∫     if omitted, prints a list of all known commands                    ∫
∫                                                                        ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ ftp>help get                                                            ≥
≥ get         receive file                                               ≥
≥ ftp>                                                                   ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

There is a synonym for the help command. It works in the same way as the help command.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  ? [command]                                          ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ ftp>? put                                                              ≥
≥ put          send one file                                             ≥
≥ ftp>                                                                   ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

FTP status can be displayed on the screen by entering the following command:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  status                                          ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ ftp>status                                                        ≥
≥ Connected to ERC830.                                              ≥
≥ No proxy connection.                                              ≥
≥ Mode: stream; Type: ascii; Form: non-print; Structure: file       ≥
≥ Verbose: on; Bell: off; Prompting: on; Globbing: on               ≥
≥ Store unique: off; Receive unique: off                            ≥
≥ Case: off; CR stripping: on                                       ≥
≥ Ntrans: off                                                       ≥
≥ Nmap: off                                                         ≥
≥ Hash mark printing: off; Use of PORT cmds: on                     ≥
≥ ftp>                                                              ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

These are the default settings. The meaning of these settings and how to change them are found in the supplemental material at the end of this manual.

There are a few "bugs" in FTP.

Correct execution of many FTP commands depends upon the remote server. The VAX server is supplied by The Wollongong Group, Inc. If you encounter problems transferring files to/from the Multimax, please bring them to the attention of the User Support Branch or call the Hotline (FTS 776-4688 or 6-HOTT).

7.8 Introducing UMAX TELNET

TELNET protocol will allow communication with another host. The TELNET protocol can be invoked from either the Annex prompt or from the shell prompt while you are logged into the Multimax. If you invoke TELNET while logged into the Multimax, that session will continue to be charged at the appropriate rate. The new session to another host will also charge the account. This means you are paying connect charges on both systems.

The syntax to invoke TELNET is as follows:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫                                                                   ∫
∫  Command Format:  telnet [host [port]]                            ∫
∫                                                                   ∫
∫      host  - the host name                                        ∫
∫                                                                   ∫
∫      port  - the port number, if not given, use default           ∫
∫                                                                   ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ $telnet                                                           ≥
≥ telnet>                                                           ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

The telnet> prompt indicates that telnet commands can now be entered. If no parameters are given, telnet enters the command mode.

In order to create a connection to another host from command mode, use the open command.

```
Command format:  open <host> [port]

    host - host name

    port - port number, optional
```

Sample session:

```
telnet>open erc830
Trying...
Connected to erc830.
Escape character is '^]'.

(Warning message from VAX)

Username:
```

If you enter the host name on the same command line as telnet, the open command will be done for you.

Sample Session:

```
$telnet erc830
Trying...
Connected to erc830.
Escape character is '^]'.

( Warning message from VAX)

Username:
```

When you logout of the destination host, you will be automatically brought back to the originating host.

Sample Session:

```
$lo
Connection closed by foreign host .L-1990 15:57:42.19
$
```

The first $ prompt is the VMS prompt. The lo command logs you out of the VAX. Notice that we get the connection closed message, and the next $ prompt is back to the Multimax.
The connection that was created was closed. There is a TELNET command to close the connection as well.

```
Command Format:  close
```

This TELNET command will close the connection and return to the TELNET command mode.

To exit TELNET, enter the following command at the telnet>

prompt.

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  quit                                          ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

This command will close any open TELNET session and exit TELNET. An end-of-file (in command mode) will also close a session and exit.

The current status of TELNET can be shown by entering the following command:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  status                                        ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ telnet>status                                                   ≥
≥ Connected to erc830.                                            ≥
≥ Operating in character-at-a-time mode.                          ≥
≥ Escape character is '^]'.                                       ≥
≥                                                                 ≥
≥ telnet>                                                         ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

A listing of TELNET commands can be displayed by entering the following command at the TELNET command mode prompt telnet>:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  help                                          ∫
∫                                                                 ∫
∫                    ?                                            ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Sample Session:

```
/ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍø
≥ telnet>help                                                     ≥
≥ Commands may be abbreviated. Commands are:                      ≥
≥                                                                 ≥
≥ close     close current connection                              ≥
≥ display   display operating parameters                          ≥
≥ mode      try to enter line-by-line or char-at-a-time mode      ≥
≥ open      connect to a site                                     ≥
≥ quit      exit telnet                                           ≥
≥ send      transmit special characters ('send ?' for more)       ≥
≥ set       set operating parameters ('set ?' for more)           ≥
≥ status    print status information                              ≥
≥ toggle    toggle operating parameters ('toggle ?' for more)     ≥
≥ z         suspend telnet                                        ≥
≥ ?         print help information                                ≥
≥ telnet>                                                         ≥
¿ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍŸ
```

Workshop 7

This workshop will reinforce your understanding of the topics covered in this chapter. Login to the Multimax with the username and password given to you by the instructor. Each student is to complete the entire workshop. Computer exercises might need to be worked as a team.

COMPUTER EXERCISES

1. Log into the Multimax.

Questions 2 through 11 have to do with a connection between the local computer (Multimax) and the remote computer (VAX).

2. Initialize FTP on the Multimax and create a connection to the VAX. (Hint: open)

   What is the remote computer default username?
   How can you enter a different username?

3. What files are on the remote computer's directory?
   (Hint: If you can't remember the FTP command, how can you find out?)

4. What is the default type? (Hint: status)

Continue on the next page

5. Transfer the file "memo" from the Multimax to the VAX.
   Change the name of the file on the VAX to "memo.doc".

6. Transfer the file "DATA.MAY" from the VAX to the Multimax.
   Keep the same filename on both platforms.

7. Without entering it, what FTP command would you enter to change the remote computer working directory to D_1131:[STUDENT]?

8. Enter the FTP command to list the contents of the local computer working directory. What files are present?

9. Enter the FTP command to list the contents of the remote computer working directory. What files are present?

10. Without entering the command, how would you change the remote working directory to D_1131:[STUDENT1]?

11. What changes would you have to make in order to transfer a binary file from the Multimax to the VAX?

Continue on the next page

** NOTE **

Questions 12 through 20 have to do with a connection between the local computer (Multimax) and the remote computer (CYBER).

12. Close the connection with the VAX and then open a connection to the CYBER.

13. What files are on the remote computer's directory?

14. What is the default type? (Hint: status)

15. Transfer the file "memo" from the Multimax to the CYBER. Change the name on the CYBER to a filename of your choice.

16. Transfer the file "MAYDATA" from the CYBER to the Multimax. Keep the same filename on both platforms.

17. Without entering it, what FTP command would you enter to change the remote computer working directory?

18. Enter the FTP command to list the contents of the local computer working directory.

Continue on the next page

19. Enter the FTP command to list the contents of the remote

computer working directory.

20. Close the connection with the CYBER and exit FTP.

Continue on the next page

** NOTE **

The following questions have to do with your understanding of the Telnet communications protocol.

21. Enter the command to invoke the Telnet protocol.

22. Open a connection to the VAX.

23.   Enter a valid username and password.

24.   Are you logged into the VAX or the Multimax?

25.   Enter the command to exit the VAX. (Hint: logoff)

26.   Are you logged into the VAX or the Multimax?

27.   Are you confused? Logout of the Multimax and the Annex.

NOTES

‹‹‹

8. INTRODUCTION TO vi

The vi editor was developed at the University of California, Berkeley. It was originally included as part of BSD UNIX. It became an official part of AT&T UNIX with the release of System V. Before vi was invented, the standard UNIX editor was ed. The ed editor was line oriented and made it difficult to see the context of the file being edited.

The next progression was an editor called ex. The ex editor had some distinct advantages over ed. It allowed you to display an entire screen of text instead of just one line at a time. While in the ex editor, you could give the command vi (for visual mode). Users used the visual mode so much that developers of ex made it possible to use the display editing feature without having to enter ex and then vi. They called the new facility simply vi.

The vi editor does its work in a work buffer. When you start vi, it copies the disk file into the work buffer. During the editing session, changes are made to this copy. The contents of the disk file are not changed until you write the contents of the work buffer to the disk file.

The command to enter the vi editor is:

```
…ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕª
∫  Command Format:  vi <file1>                                   ∫
∫                                                                ∫
∫      file1 - the filename to edit                              ∫
»ÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕÕº
```

Your screen is cleared, then the first lines of the file are displayed, and the cursor is positioned at the top of the screen. The bottom line of your screen is reserved for certain command

mode activities and for error and status messages and does not contain any of the file's text. If the file already exists, the bottom line lists the filename in quotes and the number of lines and characters it contains. If the file is new, "New file" is displayed next to the filename. If the file does not fill an entire screen, a tilde (~) character appears in the leftmost column of any blank lines.

By default, you are always in command mode at the start of a vi session. The most common command mode activities are:

- cursor positioning
- entering text mode
- moving, copying, and deleting text
- storing changes
- quitting

Whenever you wish to return to command mode, or are unsure of what mode you are in, press the Esc key.

Esc can be entered any number of times without harm. The Esc key on the VT terminals is the Ctrl-3 combination. On the PC, it is the key marked Esc.

8.1 vi: Cursor Positioning

Below is a list of cursor positioning commands. Characters are not echoed on your screen when one of these commands is executed. The cursor simply moves to the desired location. If a command is not accepted, the cursor remains where it is. The current line is defined as the line on which the cursor currently resides. The letter N is a repeat factor.

N+	move down N lines from current line. The cursor can be anyplace on the current line. When complete, the cursor will be located at the first character on the line N lines down from the current line.
N-	move up N lines from current line. The cursor can be anyplace on the current line. When complete, the cursor will be located on the first character on the line located N lines up from the current line.
(Ret)	The cursor can be located anyplace on the current line. The will be on the first character of the next line.
$	The cursor will move to the end of the current line
NG	This command will move the cursor to line N. Default is to move to the last line.
Ctrl-D	move down 1/2 screen (11 lines)
Ctrl-U	move up 1/2 screen (11 lines)
NOTE:	Words are delimited by spaces (ie., a word begins and ends with a space).
Nw	The cursor will be on the first character of the word located N words from the current word. The current word is the word where the cursor is located. The default is to skip to the beginning of the next word.

Nb The cursor will be on the first character of the word
    located N words back from the current word. The
    default is to skip back to the beginning of the
    previous word.

e   The cursor will skip to the end of the current word.

The following keys are also defined for moving around the screen:

h   back one space

j   down one line

k   up one line

l   forward one space

The arrow keys will also work.

CAUTION NOTE: If you hold the arrow key down to move quickly to another area of the text, a line might be inserted into your file.

8.2 vi: Text Mode

Several commands in command mode allow you to enter text. Once the command is entered, all other characters that you type are inserted in your text until you press the Esc key.

To add text, use:

I enter text mode, additional text appears at the beginning of the current line.

i enter text mode, additional text appears before the current cursor position.

A enter text mode, additional text appears at the end of the current line.

a enter text mode, additional text appears after the current position.

O enter text mode, open a line above the current line.

o enter text mode, open a line below the current line.

To replace text, use:

R replace characters until Esc

r replace one character at current cursor position, then return to command mode

To substitute text, use:

Ns substitute character for the current N characters until Esc. Default is to substitute for the current character until Esc.

8.3 vi: Deleting Text

vi commands for deleting text take effect relative to the

cursor's current position. Text deletion commands are not echoed on your screen.

Ndd delete N lines starting at the current line. The default is to delete the current line.

Ndw delete N words starting with the current word. The default is to delete the current word.

Nx delete N characters starting at the current cursor position. The default is to delete one character.

D delete remainder of line

8.4 vi: Copying Text

Copying text is performed using one of the "yank and put" command pairs. The most straight forward command sequence for copying is:

1. Yank a word, line, or number of lines. A copy of the yanked text is stored invisibly. The original text is not disturbed.
2. Move the cursor to the desired location.
3. Put the yanked copy into place.
4. Move the cursor to the next block of text you want to copy, then go to step 1.

Here are some yank and put commands:

NY yank N lines. Default is to yank one line.

Nyw yank N words. Default is to yank one word.

P put yanked lines above current cursor position
or
put yanked words before current cursor position

p put yanked lines below current cursor position
or
put yanked words after current cursor position

8.5 vi: Moving Text

Moving text from one area to another can be accomplished in several different ways. You can use whichever method is the easiest for you to remember.

1. Yank, put, and delete:
   a. Yank the desired text.
   b. Move the cursor to the new location and then "put" the "yanked" text into its new location.
   c. Move the cursor back to the original text and delete it.

2. Delete and put:

a. Delete the desired text

b. Move the cursor to the new location

c. Use a put command to add the text.

NOTE: The delete command stores an invisible copy of the deleted text in a buffer. This is done so the undo command is capable of restoring the previous command. That's why it is possible to move that deleted text to another area.

8.6 vi: Restoring the Last Change

The Undo command will reverse the last command you just entered. It will restore text that you have changed or deleted by mistake. The undo command will undo only the most recently changed text.

```
Command Format:  u

    u - undo the last change

      U - restore the current line to the way it was before you
        started changing it, even if several changes were made
```

If you delete a line and then change a word, undo will restore the changed word but will not restore the line.

8.7 vi: Recovering Text After a Crash

You can often recover text that would have been lost because of a system crash. When the system has been brought back up enter the following command to see if the system saved a copy of your work buffer:

Example:

```
$vi -r filename
```

If your work buffer was saved, you will be editing a recent copy of the work buffer. Use the w command to write the edited version to the disk file.

The -r option will recover the version of filename that was in the buffer when the crash occurred. If no buffer was saved, the editor will assume you are going to edit a new empty file called filename.

8.8 vi: Saving Text and Quitting

Commands to save (write) text and to quit are entered from the Last Line Mode. The Last Line Mode is entered by entering a colon (:) character from the command mode.

To save changes without exiting vi, enter:

Example:

```
.............................................................
. :w                                                        .
.............................................................
```

This command is displayed on the status line as it is typed in. The commands are executed by pressing the Enter key. The file's name and number of lines and characters are displayed on the status line. With no option, the work buffer will be written back to the original disk file. If, for some reason, you don't have write permission to the working directory, you can copy the work buffer to another file by specifying the complete pathname of a temporary file.

Example:

```
.............................................................
. :w /user0/rharding/temp                                   .
.............................................................
```

Now you can exit vi and not lose any of your work. The editing session is saved in the file /user/rharding/temp.

To exit vi without saving any of the changes since the last :w (or to discard all changes if no :w), enter:

Example:

```
.............................................................
. :q!                                                       .
.............................................................
```

The exclamation mark (!) (in slang, it's a bang) indicates to quit the current editing session, regardless. If you just enter q alone, the editor will warn you that existing changes were not saved. It is difficult to get out of this mode. Use the exclamation mark to indicate do the exit no matter what and not save the changes since the last w command.
To save and quit, enter:

Example:

```
.............................................................
. :wq                                                       .
.............................................................
```

The w command will write the work buffer to the disk file. The q command will exit the editor. The shell prompt ($) will be displayed after the file has been saved and the editor exited.
8.9 Other vi Commands

To save the file you are editing under a different name, use:

Example:

```
.............................................................
```

```
. :w newfile                                                    .
.................................................................
```

To copy in the contents of another file, position the cursor on the last line you want to be above the new text, then execute:

Example:

```
.................................................................
. :r filename                                                   .
.................................................................
```

The contents of filename will appear on your screen below the last cursor position. The existing text will be moved down.

To include the output of a shell command (i.e., date) in the file you are editing, position the cursor as described above, then enter:

Example:

```
.................................................................
. :r !shell-cmd                                                 .
.................................................................
```

To execute a shell command without including its output in your file, enter:

Example:

```
.................................................................
. :!shell-cmd                                                   .
.................................................................
```

This feature enables you to check man pages or the contents of other files without exiting vi.

NOTES

‹‹

Workshop 8

This workshop will reinforce your understanding of the topics covered in this chapter. Login to the Multimax with the username and password given to you by the instructor. Each student is to complete the entire workshop. Computer exercises might need to be worked as a team.

1. Login to the Multimax.

2. Edit the file rocket.sh .
   (Hint: vi rocket.sh)

3. Position the cursor at the beginning of line 10.

4. Move the cursor up five lines.

5. Move the cursor to the end of the current line.
   What vi command did you use?

6. Move the cursor to the first line of the file.
   What vi command did you use?

7. Move to the end of the file and insert a new line after it
   that contains the following text:

   fi

8. Remove all the blank lines from this file.

Continue on the next page

9. Locate the word grop and change it to grep .

10. Add the following text after the last line of the file.

    rm ./temp$$

11. Now execute the script by typing rocket.sh

    (Hint: What are the permissions on this file?)

    If you did the editing correctly fireworks should appear. If
    not, compare your script to /user0/teacher/rocket.sh

    To stop the fireworks enter the interrupt character (CTRL-C)

12. Create a file with a name of your own choice. Insert the
    output from the UNIX command ls -la . Save your change and
    exit vi.

13. Edit the file you just created. Go to the end of the file
    and without leaving vi, display a listing of the directory
    /user0/teacher. How do you return to the editing session?
    Did the listing get inserted into your editing session?

13. What is the option to recover your changes after a system crash?

14. Logout of the Multimax and the Annex.

9. GETTING HELP

9.1 Assist

The assist command is a menu driven utility that can provide information on the following topics:

1. Information on a variety of UNIX topics
2. Tutorials
3. The ability to construct and execute command lines
4. A "pop up" menu for advanced users

Assist is set up so you do not have to know the exact command name in order to get information or use the command. To execute assist enter:

```
Command Format:  assist  [name]
                 assist  [-s]
                 assist  [-c name]

name  - invoke an assist-supported UNIX system or
        walkthru for name.

-s    - reinvoke the assist setup module to check or
        modify the terminal variable.

-c name  - invoke the version of name that is in the
           current directory.
```

Sample session:

```
. $assist                                                    .
```

The first time assist is executed, assist will automatically check your terminal capabilities and then runs a brief tutorial. You can run the tutorial again by entering:

Sample session:

```
. $assist -s                                                 .
```

This command will also allow you to recheck your terminal setup.

The following is a list of useful assist commands:

Ctrl-A	-	assist help
Ctrl-O	-	help with current menu
Ctrl-Y	-	help with current menu item
Ctrl-T	-	call top level menu
Ctrl-F	-	call "pop up" menu
Ctrl-R	-	go back to previous menu
(Ret),Ctrl-N	-	move cursor to next menu item
Ctrl-P	-	return cursor to previous item
Ctrl-G	-	select (execute) current menu item
Ctrl-V	-	clear help message or prompt
Ctrl-D	-	exit

Assist contains information on many, but not all, of the UMAX commands. In addition, not all options and possibilities for each command are covered. For complete information about a UMAX command, please use the on-line manual pages.

9.2 UNIX Primer Plus

This manual is intended to be the reference manual for UNIX. It has several handy features. The inside of the front cover has a listing of UNIX command and the page number on which a description of the command and its options can be found. In addition, there are some quick reference sheets that can be removed from the book and used at your terminal. The book is well written, humorous, and contains a lot of information about UNIX. There might be subtle differences between generic UNIX and UMAX.

Another manual that is a good reference for UNIX is "A Practical Guide to UNIX System V" by Mark G. Sobell.

9.3 TAB (Technical Assistance Bulletin)

The TAB is published monthly and contains current articles and helpful hints for the Multimax minicomputers and UNIX in general. To be added to the mailing list to receive a FREE subscription, contact Gloria Armstrong (FTS) 776-4433 or (303) 236-4433.

9.4 Local Support

If you have a local technical person that is available, try them. Some regional offices have a hotline that you can call for assistance.

9.5 CCS Hotline

The is a technical Hotline service available in the Denver office. This service is available to the entire Bureau. This is the fastest way to get your questions answered. The Hotline number is (FTS) 776-HOTT (4688) or Commercial (303) 236-HOTT (4688).

9.6 CBT (DOS based training for UNIX)

There is a Computer Based Training course available on a PC in the Denver training room. It runs under DOS and doesn't need to be connected to a UNIX machine. It is easy to use and has lessons for the beginning and advanced UNIX user, as well as courses in C programming and UNIX system administration. It can also give you instruction about a particular command or topic that interests you.

Workshop 9

Lucky you! No workshop

Please complete the...

Summary Workshop

and

Course Evaluation

NOTES

‹‹‹

APPENDIX A: DENVER OFFICE LOGIN SEQUENCE

PRESS Space Bar

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥ WELCOME TO THE B.O.R. NETWORK P/S:B                              ≥
≥ SYSTEMS PRESENTLY AVAILABLE ARE:                                 ≥
≥                                                                  ≥
≥  **SYSTEM**          **NAME**                                    ≥
≥                                                                  ≥
≥ VAX 8300'S            VAX                                        ≥
≥ CYBER/CDCNET F.E.     CDC                                        ≥
≥ ENCORE/UNIX           MAX                                        ≥
≥ OUT DIAL              OD                                         ≥
≥                                                                  ≥
≥ TO SELECT A SYSTEM, ENTER THE SYSTEM                             ≥
≥ NAME AND CARRIAGE RETURN AT NEXT                                 ≥
≥ PROMPT.                                                          ≥
≥                                                                  ≥
≥ CHANNEL 04/010. ENTER RESOURCE MAX                               ≥
≥ CONNECTED TO 04/052                                              ≥
¿ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒŸ
```

Wait 2 seconds then PRESS (Ret) TWICE

```
/ƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒƒø
≥                                                                                   ≥
≥ Annex Command Line Interpreter   *   Copyright 1988 Xylogics, Inc.                ≥
≥                                                                                   ≥
```

```
≥  ***WARNING***Unauthorized access to U.S. Government computers                ≥
≥               is punishable by fine and/or imprisonment. ***WARNING***       ≥
≥ annex: c domax1                                                              ≥
≥ login: your username(Ret)                                                    ≥
≥ Password: your password(Ret)                                                 ≥
≥ UNIX System V Release ax.2.2o ns32332                                        ≥
≥ domax1                                                                       ≥
≥ Copyright (c) 1984 AT&T                                                      ≥
≥ All Rights Reserved                                                          ≥
≥  ***WARNING***Unauthorized access to/use of this U.S. Government             ≥
≥              computer is punishable by fine and/or imprisonment. ***WARNING***≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

NOTES

‹‹

APPENDIX B: GREAT PLAINS LOGIN SEQUENCE

PRESS (Ret)

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ WARNING ****** WARNING ****** WARNING ****** WARNING ****** WARNING        ≥
≥ PUBLIC LAW 99-474 PROHIBITS UNAUTHORIZED USE OF THIS U.S. GOVERNMENT        ≥
≥ COMPUTER SYSTEM AND/OR SOFTWARE.  PUNISHMENT INCLUDES FINES AND UP TO       ≥
≥ 10 YEARS IN PRISON.  REPORT VIOLATIONS TO THE SYSTEM SECURITY OFFICER.      ≥
≥ WARNING ****** WARNING ****** WARNING ****** WARNING ****** WARNING        ≥
≥                                                                             ≥
≥                                                                             ≥
≥ ENTER RESOURCE  A - BIL640, B - BIL751, OA - BIL630, DEN - DENVER CYBERS    ≥
≥ FOR STATUS OF COMPUTER SYSTEMS CALL   (406) 657-6828 OR FTS 585-6828        ≥
≥ FOR EMERGENCY AND AFTER HOURS CALL    (406) 255-6932                        ≥
≥                                                                             ≥
≥ CHANNEL 02/035.  ENTER RESOURCE DEN(Ret)                                    ≥
≥ CONNECTED TO 02/079                                                         ≥
≥                                                                             ≥
≥ WELCOME TO THE B.O.R. NETWORK P/S:B                                         ≥
≥ SYSTEMS PRESENTLY AVAILABLE ARE:                                            ≥
≥                                                                             ≥
≥  **SYSTEM**          **NAME**                                               ≥
≥                                                                             ≥
≥  VAX 8300'S           VAX                                                  ≥
≥  CYBER/CDCNET F.E.    CDC                                                  ≥
≥  CENTER               ASC                                                  ≥
≥  ENCORE/UNIX          MAX                                                  ≥
≥  TO SELECT A SYSTEM, ENTER THE SYSTEM                                       ≥
≥  CARRIAGE RETURN AT NEXT                                                    ≥
≥  PROMPT.                                                                    ≥
≥                                                                             ≥
≥  CHANNEL 02/079. ENTER RESOURCE MAX(Ret)                                    ≥
≥  CONNECTED TO 06/025                                                        ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

PRESS (RET) TWICE

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥                                                                             ≥
≥ Annex Command Line Interpreter   *   Copyright 1988 Xylogics, Inc.          ≥
≥                                                                             ≥
≥  ***WARNING***Unauthorized access to U.S. Government computers              ≥
≥               is punishable by fine and/or imprisonment. ***WARNING***      ≥
≥ annex: c domax1                                                             ≥
≥ login: your username(Ret)                                                   ≥
≥ Password: your password(Ret)                                                ≥
≥ UNIX System V Release ax.2.2o ns32332                                       ≥
≥ domax1                                                                      ≥
≥ Copyright (c) 1984 AT&T                                                     ≥
```

APPENDIX C: LOWER COLORADO LOGIN SEQUENCE

The following operating procedures show how a user gets to Denver using the Local Area Network (LAN) in Boulder City, starting with the PC prompt: M:\USERNAME>

ENTER PCPLUS(Ret)

```
≥                          |                                          ≥
≥ COMMUNICATION SERVICES   |     PROCOMM PLUS ADD SERVICES MENU       ≥
≥     ON NETWORK           |                                          ≥
≥                          |                                          ≥
≥ GENERAL SPECIFIC SERVER  | UP/DOWN ARROW ..Highlight Services       ≥
≥__________________________|                                          ≥
≥ MICOM       *       *    | ENTER ....Connect Highlighted Services   ≥
≥ VAX_19.2    *       *    |                                          ≥
≥ MI24        *       *    | PgPd .....Scroll Up One Page             ≥
≥ ADMICOM     *       *    |                                          ≥
≥                          | PgPn .....Scroll Down One Page           ≥
≥                          |                                          ≥
≥                          | Home .....First Service                  ≥
≥                          |                                          ≥
≥                          | End ......Last Service                   ≥
≥                          |                                          ≥
≥                          | Alt-E ....Expand/Contract Services       ≥
≥                          |                                          ≥
≥                          | Alt-M ....Manual Connect                 ≥
≥                          |                                          ≥
≥                          | Alt-X ....Exit PROCOMM PLUS              ≥
≥                          |                                          ≥
≥                          | Alt-Z ....Help                           ≥
```

SELECT MICOM. PRESS (Ret) SEVERAL TIMES

```
≥   THIS IS THE LOWER COLORADO REGIONAL OFFICE INSTANET 6600    ≥
≥   RESOURCES AVAILABLE                                        ≥
≥   BLD460                                                     ≥
≥   BLD732                                                     ≥
≥   BLDT50                                                     ≥
≥   DEN   (1200BPS)                                           ≥
≥   DEN2  (2400BPS)                                           ≥
≥   OUTDIAL (1200 BPS)                                        ≥
≥   TELEBIT (1400 BPS OUTDIAL)                                ≥
≥   VAX  (19.2 lines only)                                    ≥
≥   CHANNEL 02/008. ENTER RESOURCE                            ≥
```

ENTER DEN(Ret)

```
≥ You are accessing the Denver MICOM through the Boulder City   ≥
≥ MICOM.  Please remember to hit the break key three times      ≥
≥ after logging off.  The first DISCONNECTED comes from.  The   ≥
≥ second DISCONNECTED comes from Boulder City.  This will assure≥
≥ that other users can connect when you are finished.           ≥
```

```
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

PRESS (Ret) SEVERAL TIMES

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥   **SYSTEM**                  **NAME**                        ≥
≥                                                               ≥
≥   VAX 8300'S                   VAX                            ≥
≥   CYBER/CDCNET F.E.            CDC                            ≥
≥   ASC/CORP. CENTER             ASC                            ≥
≥   ENCORE/UNIX                  MAX                            ≥
≥                                                               ≥
≥   TO SELECT A SYSTEM, ENTER THE SYSTEM                        ≥
≥   NAME AND CARRIAGE RETURN AT NEXT                            ≥
≥   PROMPT.                                                     ≥
≥                                                               ≥
≥   CHANNEL 02/079. ENTER RESOURCE MAX(Ret)                     ≥
≥   CONNECTED TO 06/025                                         ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

PRESS (Ret) TWICE

```
/fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥                                                                                ≥
≥ Annex Command Line Interpreter   *   Copyright 1988 Xylogics, Inc.            ≥
≥                                                                                ≥
≥  ***WARNING***Unauthorized access to U.S. Government computers                ≥
≥                is punishable by fine and/or imprisonment. ***WARNING***        ≥
≥ annex: c domax1                                                                ≥
≥ login: your username(Ret)                                                      ≥
≥ Password: your password(Ret)                                                   ≥
≥ UNIX System V Release ax.2.2o ns32332                                          ≥
≥ domax1                                                                         ≥
≥ Copyright (c) 1984 AT&T                                                        ≥
≥ All Rights Reserved                                                            ≥
≥  ***WARNING***Unauthorized access to/use of this U.S. Government              ≥
≥               computer is punishable by fine and/or imprisonment. ***WARNING***≥
¿fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

APPENDIX D: MID-PACIFIC LOGIN SEQUENCE

NETWORK LOGIN PROCEDURE

TYPE PCOMN(Ret)

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥         /fffffffffffffffffffffffffffffffffffffffffffø          ≥
≥ /ffffff¥                 Sacramento Connect Menu         √ffffffffø ≥
≥ ≥       ¿fffffffffffffffffffffffffffffffffffffffffffŸ        ≥ ≥
≥ ≥                                                            ≥ ≥
≥ ≥                                                            ≥ ≥
≥ ≥       1)  Connect to the Sacramento VAX 8300 (USR)         ≥ ≥
≥ ≥                                                            ≥ ≥
≥ ≥       2)  Connect to the Sacramento VAX 780 (CVOCO)        ≥ ≥
≥ ≥                                                            ≥ ≥
≥ ≥       3)  Connect to the Sacramento ENCORE                 ≥ ≥
≥ ≥                                                            ≥ ≥
≥ ≥       4)  Connect to the Sacramento (TCP/IP) NETWORK       ≥ ≥
≥ ≥                                                            ≥ ≥
≥ ≥       5)  Manual Setup/Connections                         ≥ ≥
≥ ≥                                                            ≥ ≥
≥ ≥       D)  Connect to the DENVER Computers                  ≥ ≥
```

```
≥ ≥                                                             ≥ ≥
≥ ≥       E)  EXIT to DOS                                       ≥ ≥
≥ ¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

PRESS D(Ret)

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥         /fffffffffffffffffffffffffffffffffffffffffffffø          ≥
≥ /ffffff¥                    Denver Connect Menu          √ffffffffø ≥
≥ ≥       ¿ffffffffffffffffffffffffffffffffffffffffffffŸ        ≥ ≥
≥ ≥                                                             ≥ ≥
≥ ≥                                                             ≥ ≥
≥ ≥       1)  Connect to the Denver VAX 8300 (USR)              ≥ ≥
≥ ≥                                                             ≥ ≥
≥ ≥       2)  Connect to the Denver CYBER AA & EE               ≥ ≥
≥ ≥                                                             ≥ ≥
≥ ≥       3)  Connect to the Denver ENCORE                      ≥ ≥
≥ ≥                                                             ≥ ≥
≥ ≥       4)  Connect to the Denver IBM (FFS)                   ≥ ≥
≥ ≥                                                             ≥ ≥
≥ ≥       5)  Connect to Sacramento Computers                   ≥ ≥
≥ ≥                                                             ≥ ≥
≥ ≥       E)  EXIT to DOS                                       ≥ ≥
≥ ¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

PRESS 3(Ret)

```
/fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥ hosts                                                                        ≥
≥    Host Name                System Status  Load Factor    Inet Addr          ≥
≥    ====================================================================      ≥
≥    domax0                   up             0.46           137.77.1.2         ≥
≥    domax1                   up             1.23           137.77.1.3         ≥
≥    dosun0                   up             1.28           137.77.1.5         ≥
≥    erc830                   up             0.36           137.77.1.4         ≥
≥ annex: c domax0                                                              ≥
≥ login: your username(Ret)                                                    ≥
≥ Password: your password(Ret)                                                 ≥
≥ UNIX System V Release ax.2.2j ns32332                                        ≥
≥ domax0                                                                       ≥
≥ Copyright (c) 1984 AT&T                                                      ≥
≥ All Rights Reserved                                                          ≥
≥  ***WARNING***Unauthorized access to/use of this U.S. Government             ≥
≥          computer is punishable by fine and/or imprisonment. ***WARNING***≥
≥                                                                              ≥
¿fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

DEDICATED LINE LOGIN

TYPE PCOM(Ret)

PRESS (Ret)

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥   NAME OF RESOURCE:  DEN(Ret)                                   ≥
≥                                                                 ≥
≥   **SYSTEM**                  **NAME**                          ≥
≥                                                                 ≥
≥   VAX 8300'S                   VAX                              ≥
≥   CYBER/CDCNET F.E.            CDC                              ≥
≥   ASC/CORP. CENTER             ASC                              ≥
≥   ENCORE/UNIX                  MAX                              ≥
```

```
≥                                                                    ≥
≥   TO SELECT A SYSTEM, ENTER THE SYSTEM                            ≥
≥   NAME AND CARRIAGE RETURN AT NEXT                                 ≥
≥   PROMPT.                                                          ≥
≥                                                                    ≥
≥   CHANNEL 02/079. ENTER RESOURCE MAX(Ret)                          ≥
≥   CONNECTED TO 06/025                                              ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

PRESS (Ret) TWICE

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥                                                                                           ≥
≥ Annex Command Line Interpreter   *   Copyright 1988 Xylogics, Inc.                       ≥
≥                                                                                           ≥
≥  ***WARNING***Unauthorized access to U.S. Government computers                           ≥
≥                is punishable by fine and/or imprisonment. ***WARNING***                   ≥
≥ annex: c domax1                                                                           ≥
≥ login: your username(Ret)                                                                 ≥
≥ Password: your password(Ret)                                                              ≥
≥ UNIX System V Release ax.2.2o ns32332                                                     ≥
≥ domax1                                                                                    ≥
≥ Copyright (c) 1984 AT&T                                                                   ≥
≥ All Rights Reserved                                                                       ≥
≥  ***WARNING***Unauthorized access to/use of this U.S. Government                         ≥
≥              computer is punishable by fine and/or imprisonment. ***WARNING***≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

NOTES

‹‹

APPENDIX E: PACIFIC NORTHWEST LOGIN SEQUENCE

PRESS (Ret) OR Space Bar

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥  ******************************  NOTICE  *******************************  ≥
≥  USE OF GOVERNMENT  COMPUTER RESOURCES AND DATA IS RESTRICTED TO OFFICIAL  ≥
≥  GOVERNMENT BUSINESS.  FAILURE TO COMPLY COULD RESULT IN DISCIPLINARY      ≥
≥  ACTION OR PROSECUTION UNDER FEDERAL LAW.  REPORT UNAUTHORIZED USE OR      ≥
≥  ACCESS TO THE ADP SECURITY OFFICER AT (208)334-1746 OR (FTS)554-1746.     ≥
≥                                                                           ≥
≥  C  =  CYBER                                                             ≥
≥  H  =  HYDROMET                                                          ≥
≥  P  =  OUT DIAL                                                          ≥
≥  V  =  VAX BOISE                                                         ≥
≥  Y  =  YAKIMA VAX                                                        ≥
≥                                                                           ≥
≥  CHANNEL 02/014. ENTER RESOURCE:  C(Ret)                                  ≥
≥  CONNECTED TO CHANNEL 03/094                                             ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

PRESS (Ret) TWO OR THREE TIMES

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥    **SYSTEM**                  **NAME**                              ≥
≥                                                                     ≥
≥    VAX 8300'S                  VAX                                   ≥
≥    CYBER/CDCNET F.E.           CDC                                   ≥
≥    ASC/CORP. CENTER            ASC                                   ≥
≥    ENCORE/UNIX                 MAX                                   ≥
≥                                                                     ≥
≥    TO SELECT A SYSTEM, ENTER THE SYSTEM                             ≥
≥    NAME AND CARRIAGE RETURN AT NEXT                                 ≥
```

```
≥    PROMPT.                                                    ≥
≥                                                               ≥
≥    CHANNEL 02/079. ENTER RESOURCE MAX(Ret)                    ≥
≥    CONNECTED TO 06/025                                        ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
       PRESS (Ret) TWICE

/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥                                                                                       ≥
≥ Annex Command Line Interpreter   *   Copyright 1988 Xylogics, Inc.                    ≥
≥                                                                                       ≥
≥  ***WARNING***Unauthorized access to U.S. Government computers                       ≥
≥                is punishable by fine and/or imprisonment. ***WARNING***               ≥
≥ annex: c domax1                                                                       ≥
≥ login: your username(Ret)                                                             ≥
≥ Password: your password(Ret)                                                          ≥
≥ UNIX System V Release ax.2.2o ns32332                                                 ≥
≥ domax1                                                                                ≥
≥ Copyright (c) 1984 AT&T                                                               ≥
≥ All Rights Reserved                                                                   ≥
≥  ***WARNING***Unauthorized access to/use of this U.S. Government                     ≥
≥               computer is punishable by fine and/or imprisonment. ***WARNING***       ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

APPENDIX F: UPPER COLORADO LOGIN SEQUENCE

```
       PRESS (Ret)

/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥    Server>  C MICOM2400(Ret)                                  ≥
≥    Server -010- Session 1 connected.                          ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ


       PRESS (Ret)

/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥    SLC PORT SELECTOR                                          ≥
≥    CHANNEL 01/091.  ENTER HOST:   DEN(Ret)                    ≥
≥    CONNECTED TO 01/014.                                       ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ


       PRESS (Ret) TWO OR THREE TIMES

/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥    **SYSTEM**                  **NAME**                       ≥
≥                                                               ≥
≥    VAX 8300'S                   VAX                           ≥
≥    CYBER/CDCNET F.E.            CDC                           ≥
≥    ASC/CORP. CENTER             ASC                           ≥
≥    ENCORE/UNIX                  MAX                           ≥
≥                                                               ≥
≥    TO SELECT A SYSTEM, ENTER THE SYSTEM                       ≥
≥    NAME AND CARRIAGE RETURN AT NEXT                           ≥
≥    PROMPT.                                                    ≥
≥                                                               ≥
≥    CHANNEL 02/079. ENTER RESOURCE MAX(Ret)                    ≥
≥    CONNECTED TO 06/025                                        ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
       PRESS (Ret) TWICE

/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥                                                                                       ≥
≥ Annex Command Line Interpreter   *   Copyright 1988 Xylogics, Inc.                    ≥
```

```
≥                                                                              ≥
≥  ***WARNING***Unauthorized access to U.S. Government computers               ≥
≥               is punishable by fine and/or imprisonment. ***WARNING***       ≥
≥ annex: c domax1                                                              ≥
≥ login: your username(Ret)                                                    ≥
≥ Password: your password(Ret)                                                 ≥
≥ UNIX System V Release ax.2.2o ns32332                                        ≥
≥ domax1                                                                       ≥
≥ Copyright (c) 1984 AT&T                                                      ≥
≥ All Rights Reserved                                                          ≥
≥  ***WARNING***Unauthorized access to/use of this U.S. Government             ≥
≥              computer is punishable by fine and/or imprisonment. ***WARNING***≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

APPENDIX G: WASHINGTON OFFICE LOGIN SEQUENCE

PRESS Space Bar ONCE OR TWICE

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥    CONNECTED TO 01/044                                                       ≥
≥    WELCOME TO THE B.O.R. NETWORK P/S:C                                       ≥
≥    SYSTEMS PRESENTLY AVAILABLE ARE:                                          ≥
≥                                                                              ≥
≥    **SYSTEM**           **NAME**                                             ≥
≥                                                                              ≥
≥    CYBER SYSTEMS                                                             ≥
≥     (AA OR EE)                                                               ≥
≥    VAX CLUSTER           DEN                                                 ≥
≥                                                                              ≥
≥    OUT-DIAL MODEM        OD                                                  ≥
≥                                                                              ≥
≥    TO SELECT A SYSTEM,ENTER THE SYSTEM                                       ≥
≥    NAME AND CARRIAGE-RETURN AT NEXT                                          ≥
≥    PROMPT.                                                                   ≥
≥                                                                              ≥
≥    CHANNEL 02/026. ENTER RESOURCE  DEN(Ret)                                  ≥
≥    CONNECTED TO 01/051                                                       ≥
≥                                                                              ≥
≥    **SYSTEM**                 **NAME**                                       ≥
≥                                                                              ≥
≥    VAX 8300'S                  VAX                                           ≥
≥    CYBER/CDCNET F.E.           CDC                                           ≥
≥    ASC/CORP. CENTER            ASC                                           ≥
≥    ENCORE/UNIX                 MAX                                           ≥
≥                                                                              ≥
≥    TO SELECT A SYSTEM, ENTER THE SYSTEM                                      ≥
≥    NAME AND CARRIAGE RETURN AT NEXT                                          ≥
≥    PROMPT.                                                                   ≥
≥                                                                              ≥
≥    CHANNEL 02/079. ENTER RESOURCE MAX(Ret)                                   ≥
≥    CONNECTED TO 06/025                                                       ≥
¿ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffŸ
```

PRESS (Ret) TWICE

```
/ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffø
≥                                                                              ≥
≥ Annex Command Line Interpreter   *   Copyright 1988 Xylogics, Inc.           ≥
≥                                                                              ≥
≥  ***WARNING***Unauthorized access to U.S. Government computers               ≥
≥               is punishable by fine and/or imprisonment. ***WARNING***       ≥
≥ annex: c domax1                                                              ≥
≥ login: your username(Ret)                                                    ≥
≥ Password: your password(Ret)                                                 ≥
≥ UNIX System V Release ax.2.2o ns32332                                        ≥
≥ domax1                                                                       ≥
```

APPENDIX H: UNIX COMMANDS QUICK REFERENCE

a > b	put the output of command a into file b
a >> b	append the output of command a onto file b
a < b	take the input of command a from file b
a | c	pipe the output of command a to the input of command c
a &	run command a in the background
assist	call up the assist menu for information on UMAX commands
at time < script	run script at time
at -l	list your at jobs waiting to be executed
at -r xx	remove at job xx
awk '/str1/,/str2/' file	display all lines between those containing str1 and str2
awk '{print $n,$m}' file	display fields n and m of file
call host	connect to a Multimax from an Annex
cat file	display file on the screen
cat file1 >> file2	append file1 onto file2
cd	return to your home directory
cd dir	work in directory dir
chmod perms file	change permissions on file to perms
cp file1 file2	copy file1 to file2
cp f1 f2 f3 dir	copy files f1, f2, and f3 into directory dir
csh	the C shell
cu options host	dial up a remote host
cut -fx file	display field x of file
cut -da -fx file	use a as a field separator
diff file 1 file 2	display differences between file1

	and file2
echo string	display string on the terminal
file file1	describe file1's type (data, text, binary, etc)
finger user	display information on user
ftp	interactive remote file transfer
grep string file	search for string in file
grep -c string file	display only the number of occurrences of string
grep -l string files	list file names that contain string
kill %x	kill background job x
ksh	the KornShell
lp -ddest file	Print file on the printer dest
ls	list the files in the current working directory
ls dir	list the files in directory dir
ls -a	include files that begin with a . (period)
ls -l	long listing including permissions, size and ownership
ls -C	list in columns
ls -ld	display detailed information on a directory, not its contents
mailx	read mail via interactive mail program
mailx user	send mail to user
man command	display the man pages for command
mkdir dir	create directory dir
mv file1 file2	move file1 to file2
mv f1 f2 f3 dir	move files f1, f2, and f3 into directory dir
nsh host commands	execute commands on a remote host
passwd	change your password
pg file	display file on screen at a time
ps	display process status of your current session
ps -u user	display process for user

pwd	print (current) working directory
rcp host1:file host2:file	copy files from one host to another
rlogin host	login to a remote host
rm file	remove file
rm -rdir	remove directory dir and contents
rmdir dir	remove directory dir
ruptime	display status of hosts on the network
rwho	display information on network users
sed -e "action" file	use stream editor on file
sh	Bourne shell
shl	the Shell Layer Manager
sort file	perform an alphabetic sort based on the first field of file
sort -n file	perform a numeric sort based on the first field of file
sort +x file	sort on field x+1
sort -ta file	use a as a field separator
spell file	check file for correct spelling
stty	display current stty settings
stty intr	set interrupt character to
stty kill	set kill character to
talk	talk with user on your terminal
talk file	display the last 10 lines of file
telenet host	connect to a remote host
telenet annex	connect to an Annex for use of an outbound port
tr a b file	in file, change every a to b
vi file	edit file with a full screen editor
wc file	list the number of lines, words and characters in file
write user	send a message to user's terminal
uucp file hostpath	remote copy

APPENDIX I: vi COMMANDS QUICK REFERENCE

Special Commands

```
Esc                  return to command mode
 u                   undo last command
 .                   repeat last insert, delete or put command

Saving Text and Quitting

:w                   write (save) text
:w newfile           save text to file newfile
:x,yw newfile        save lines x to y into newfile
:q!                  quit without saving changes
:wq                  save text and quit

Cursor Positioning

N                    move to line N
N+                   down N lines
N-                   up N lines
^D                   down one screen
^U                   up one screen
k                    up one line
j                    down one line
^                    beginning of line
$                    end of line
Nw                   N words ahead
Nb                   back N words
w                    word ahead
b                    back one word
e                    end of word
h                    backspace
l                    forward one space
arrow keys           space left or right, go up or down one line


Searches

/pattern             search forward for pattern
?pattern             search backward for pattern
 ? or /              repeat the last search

Deleting Text

Ndd                  delete N lines
dd                   delete current line
D                    delete remainder of line
Ndw                  delete N words
dw                   delete current word
Nx                   delete N characters
x                    delete one character

Copying Text

NY                   yank N lines
Y                    yank one line
Nyw                  yank N words
yw                   yank one word

P                    put yanked lines above current cursor position, or
                     put yanked words before current cursor position

p                    put yanked lines below current cursor position, or
                     put yanked words after current cursor position

Entering Text Mode

I                    enter text mode, additional text appears at the
                     beginning of the current line
```

```
i                   enter text mode, additional text appears before
                    the current cursor position

A                   enter text mode, additional text appears at the
                    end of the current line

a                   enter text mode, additional text appears after the
                    current cursor position.

O                   enter text mode, open a line above the current
                    line

o                   enter text mode, open a line below the current
                    line

Replacing and Substituting Text

r                   replace one character at current cursor position,
                    then return to command mode

R                   replace characters until Esc

s                   substitute characters for the current character
                    until Esc

Ns                  substitute characters for the current N characters
                    until Esc
Reading in Text

:r filename         append the contents of filename below the current
                    cursor position

:r !shell-cmd       append the output of shell-cmd below the current
                    cursor position

Global Operations

:x,ys/old/new/g
                    on lines x through y, change old to new

:x,yg/pattern/d
                    delete any line from x toy that has the string
                    pattern
                                        NOTES
‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹‹
APPENDIX J:  vi COMMANDS REFERENCE


NAME
     vi - screen-oriented (visual) display editor based on ex

SYNOPSIS
     vi [ -t tag ] [ -r file ] [ -L ] [ -wn ] [ -R ] [ -x ] [ -C
     ] [ -ccommand ] file ...
     view [ -t tag ] [ -r file ] [ -L ] [ -wn ] [ -R ] [ -x ] [
     -C ] [ -ccommand ] file ...
     vedit [ -t tag ] [ -r file ] [ -L ] [ -wn ] [ -R ] [ -x ] [
     -C ] [ -ccommand ] file ...

DESCRIPTION
     vi (visual) is a display-oriented text editor based on an
     underlying line editor ex(1).  It is possible to use the
     command mode of ex from within vi and vice-versa. The visual
     commands are described on this manual page; how to set
     options (like automatically numbering lines and
```

automatically starting a new output line after a carriage return) and all ex(1) line editor commands are described on the ex(1) manual page.

When using vi, changes made to the file are reflected in what is displayed on the terminal screen. The position of the cursor on the screen indicates the position within the file.

INVOCATION

The following invocation options are interpreted by vi:

-t tag Edit the file containing the tag and position the cursor at its definition. The file (tags) containing the tag is found in the current directory or in /usr/lib/tags. Below is an example of a tags file:

```
line /tmp/vi.file   /line/
this /tmp/vi.file   /this/
```

Using "vi -t line", the edited file will be "/tmp/vi.file". The file will be searched for the first occurrence of "line", and the cursor will be placed at "line".

-r file Edit file after an editor or system crash. (Recovers the version of file that was in the buffer when the crash occurred.)

-L List the name of all files saved as the result of an editor or system crash.

-wn Set the default window size to n. This is useful when using the editor over a slow speed line.

-R Readonly mode; the readonly flag is set, preventing accidental overwriting of the file.

-x Encryption option; when used, vi simulates the X command of ex(1) and prompts the user for a key. This key is used to encrypt and decrypt text using the algorithm of crypt(1). The X command makes an educated guess to determine whether or not text read in is encrypted. The temporary buffer file is encrypted also, using a transformed version of the key typed in for the -x option. See crypt(1). Also, see the WARNING section at the end of this manual page.

-C Encryption option, same as the -x option, except that vi simulates the C command of ex(1). The C command is like the X command of ex(1), except that all text read in is assumed to have been encrypted.

-c command Begin editing by executing the specified editor command (usually a search or positioning command).

The file argument indicates one or more files to be edited.

The view invocation is the same as vi except that the readonly flag is set.

The vedit invocation is intended for beginners. It is the same as vi except that the report flag is set to 1, the

showmode and novice flags are set, and magic is turned off.
These defaults make it easier to learn vi.

VI MODES

Command Normal and initial mode. Other modes return to command mode upon completion. ESC (escape) is used to cancel a partial command.

Input Entered by setting the following options: a i A I o O c s R. Arbitrary text may then be entered. Input mode is normally terminated with ESC character, or abnormally with interrupt.

Last line
Reading input for : / ? or !; terminate with CR to execute, interrupt to cancel.

COMMAND SUMMARY

In the descriptions, CR stands for carriage return and ESC stands for the escape key.

Sample Commands

<- | | ->	arrow keys move the cursor
h j k l	same as arrow keys
itextESC	insert text
cwnewESC	change word to new
easESC	pluralize word (end of word; append s; escape from input state)
x	delete a character
dw	delete a word
dd	delete a line
3dd	delete 3 lines
u	undo previous change
ZZ	exit vi, saving changes
:q!CR	quit, discarding changes
/textCR	search for text
U D	scroll up or down
:ex cmdCR	any ex or ed command

Counts Before vi Commands

Numbers may be typed as a prefix to some commands. They are interpreted in one of these ways:

line/column number	z G |
scroll amount	D U
repeat effect	most of the rest

Interrupting, Canceling

ESC	end insert or incomplete cmd
DEL	(delete or rubout) interrupts
L	reprint screen if DEL scrambles it
R	reprint screen if L is -> key

File Manipulation

ZZ	if file is modified, write and exit; otherwise, exit
:wCR	write back changes
:w!CR	forced write, if permission originally not valid
:qCR	quit
:q!CR	quit, discard changes
:e nameCR	edit file name
:e!CR	reedit, discard changes
:e + nameCR	edit, starting at end
:e +n filename CR	edit starting at line n
:e #CR	edit alternate file

```
  :e! #CR             edit alternate file, discard changes
  :w nameCR           write file name

  :w! nameCR          overwrite file name
  :shCR               run shell, then return
  :!cmdCR             run cmd, then return
  :nCR                edit next file in arglist
  :n argsCR           specify new arglist
  G                   show current file and line
  :ta tagCR           to tag file entry tag

  In general, any ex or ed command (such as substitute or
  global) may be typed, preceded by a colon and followed by a
  CR.

Positioning Within File
  F                   forward screen
  B                   backward screen
  D                   scroll down half screen
  U                   scroll up half screen
  Ng                  go to the beginning of the specified
                      line (end default), where n is a line
                      number
  /pat                next line matching pat
  ?pat                prev line matching pat
  n                   repeat last / or ? command
  N                   reverse last / or ? command
  /pat/+n             nth line after pat
  ?pat?-n             nth line before pat
  ]]                  next section/function
  [[                  previous section/function
  (                   beginning of sentence
  )                   end of sentence
  {                   beginning of paragraph
  }                   end of paragraph
  %                   find matching ( ) { or }

Adjusting The Screen
  L                   clear and redraw
  zCR                 clear and redraw window if ^L is -> key
  ZCR                 redraw screen with current line at top
                      of window
  z-CR                redraw screen with current line at
                      bottom of window
  z.CR                redraw screen with current line at
                      center of window
  /pat/z-CR           move pat line to bottom of window
  zn.CR               use n line window
  E                   scroll window down 1 line
  Y                   scroll window up 1 line
Marking and Returning
  ``                  move cursor to previous context
  ''                  move cursor to first non-white space in
                      line
  mx                  mark current position with the ACSII
                      lower-case letter x
  `x                  move cursor to mark x
  'x                  move cursor to first non-white space in
                      line marked by x

Line Positioning
  H                   top line on screen
  L                   last line on screen
  M                   middle line on screen
  +                   next line, at first non-white
  -                   previous line, at first non-white
```

```
  CR                  return, same as +
  | or j              next line, same column
  | or k              previous line, same column

Character Positioning
  first non-white-space character
  0                   beginning of line
  $                   end of line
  l or ->             forward
  h or <-             backwards
  H                   same as <- (backspace)
  space               same as -> (space bar)
  fx                  find next x
  Fx                  find previous x
  tx                  move to character prior to next x
  Tx                  move to character following previous x
  ;                   repeat last f F
  ,                   repeat last t T
  n|                  to specified column
  %                   find matching () { or }

Words, Sentences, Paragraphs
  w                   forward a word
  b                   back a word
  e                   end of word
  )                   to next sentence
  }                   to next paragraph
  (                   back a sentence
  {                   back a paragraph
  W                   forward a blank-delimited word
  B                   back a blank-delimited word
  E                   to end of a blank-delimited word
Corrections During Insert
  H                   erase last character (backspace)
  W                   erase last word
  erase               erase, same as H
  kill                kill, erase this line of input
  \                   quotes H, erase and kill characters
  ESC                 ends insertion, back to command mode
  DEL                 interrupt, terminates insert mode
  D                   backtab one character; reset left margin
                      of autoindent
  |D                  caret () followed by control-d (D);
                      backtab to beginning of line; do not
                      reset left margin of autoindent
  0D                  backtab to beginning of line; reset left
                      margin of autoindent
  V                   quote non-printable character

Insert and Replace
  a                   append after cursor
  A                   append at end of line
  i                   insert before cursor
  I                   insert before first non-blank
  o                   open line below
  O                   open above
  rx                  replace single char with x
  RtextESC            replace characters

Operators
  Operators are followed by a cursor motion, and affect all
  text that would have been moved over.  For example, since w
  moves over a word, dw deletes the word.  Double the
  operator, e.g., dd to affect whole lines.

  d                   delete
```

```
     c                   change
     y                   yank lines to buffer
     <                   left shift
     >                   right shift
     !                   filter through command

   Miscellaneous Operations
     C                   change rest of line (c$)
     D                   delete rest of line (d$)
     s                   substitute chars (cl)
     S                   substitute lines (cc)
     J                   join lines
     x                   delete characters (dl)
     X                   ... before cursor (dh)
     Y                   yank lines (yy)
   Yank and Put
     Put inserts the text most recently deleted or yanked;
     however, if a buffer is named (using the ASCII lower-case
     letters a - z), the text in that buffer is put instead.

     3yy                 yank 3 lines
     3yl                 yank 3 characters
     p                   put back text after cursor
     P                   put back text before cursor
     "xp                 put from buffer x
     "xY ("xyy)          yank to buffer x
     "xD ("xdd)          delete into buffer x

   Undo, Redo, Retrieve
     u                   undo last change
     U                   restore current line
     .                   repeat last change
     "dp                 retrieve d'th last delete

AUTHOR
     vi and ex were developed by The University of California,
     Berkeley California, Computer Science Division, Department
     of Electrical Engineering and Computer Science.

FILES
     /tmp                     default directory where temporary
                              work files are placed; it can be
                              changed using the directory option
                              (see the ex(1) set command)
     /usr/lib/terminfo/?/*    compiled terminal description
                              database
     /usr/lib/.COREterm/?/*   subset of compiled terminal
                              description database, supplied on
                              hard disk

NOTES
     Two options, although they continue to be supported, have
     been replaced in the documentation by options that follow
     the Command Syntax Standard (see intro(1)).  A -r option
     that is not followed with an option-argument has been
     replaced by -L and +command has been replaced by -c command.

SEE ALSO
     ed(1), ex(1).
     "Screen Editor Tutorial (vi)" in the UMAX V User's Guide.

WARNING
     The encryption options are provided with the Security
     Administration Utilities package, which is available only in
     the United States.
     Tampering with entries in /usr/lib/.COREterm/?/* or
```

/usr/lib/terminfo/?/* (for example, changing or removing an entry) can affect programs such as vi(1) that expect the entry to be present and correct. In particular, removing the "dumb" terminal may cause unexpected problems.

BUGS

Software tabs using T work only immediately after the autoindent.

Left and right shifts on intelligent terminals do not make use of insert and delete character operations in the terminal.

APPENDIX K: ftp COMMANDS REFERENCE

NAME

ftp - Internet file transfer program

SYNOPSIS

ftp [-v] [-d] [-i] [-n] [-g] [host]

DESCRIPTION

ftp is the user interface to the DARPA File Transfer Protocol. The program transfers files to and from a remote network site.

The client host with which ftp is to communicate can be specified on the command line. In this case, ftp immediately attempts to establish a connection to an FTP server on that host; otherwise, ftp enters its command interpreter and waits for instruction, displaying the prompt ftp>.

ftp recognizes the following commands:

! [command [args]]
Invoke an interactive shell on the local machine. If there are arguments, the first is taken to be a command to execute directly, with the rest of the arguments as its arguments.

$ macro-name [args]
Execute the macro-name that was defined with the macdef command. Arguments are passed to the macro unglobbed.

account [passwd]
Supply a supplemental password required by a remote system for access to resources once a login has been successfully completed. If no argument is included, the user will be prompted for an account password in a non-echoing input mode.

append local-file [remote-file]
Append a local file to a file on the remote machine. If remote-file is left unspecified, the local file name is used to name the remote file after being altered by any ntrans or nmap setting. File transfer uses the current settings for type, format, mode, and structure.

ascii Set the file transfer type to network ASCII. This is the default type.

bell Sound a bell after each file transfer command is completed.

binary Set the file transfer type to support binary image

transfer.

bye
Terminate the FTP session with the remote server and exit ftp.

case
Toggle remote computer file name case mapping during mget commands. When case is on (default is off), remote computer file names with all letters in upper case are written in the local directory with the letters mapped to lower case.

cd remote-directory
Change the working directory on the remote machine to remote-directory.

cdup
Change the remote machine working directory to the parent of the current remote machine working directory.

close
Terminate the FTP session with the remote server, and return to the command interpreter. Any defined macros are erased.

cr
Toggle carriage return stripping during ASCII type file retrieval. Records are denoted by a carriage return/linefeed sequence during ASCII type file transfer. When cr is on (the default), carriage returns are stripped from this sequence to conform with the UNIX single linefeed record delimiter. Records on non-UNIX remote systems may contain single linefeeds; when an ASCII type transfer is made, these linefeeds may be distinguished from a record delimiter only when cr is off.

delete remote-file
Delete the file remote-file on the remote machine.

debug [debug-value]
Toggle debugging mode. If an optional debug-value is specified, it is used to set the debugging level. When debugging is on, ftp prints each command sent to the remote machine, preceded by the string --> .

dir [remote-directory] [local-file]
Print the contents of directory, remote-directory, and, optionally, place the output in local-file. If no directory is specified, the current working directory on the remote machine is used. If no local file is specified, or local-file is -, output comes to the terminal.

disconnect
A synonym for close.

form format
Set the file transfer form to format. The default format is file.

get remote-file [local-file]
Retrieve the remote-file and store it on the local machine. If the local file name is not specified, it is given the same name it has on the remote machine, subject to alteration by the current case, ntrans, and nmap settings. The current settings for type, form, mode, and structure are used while transferring the file.

glob Toggle filename expansion for mdelete, mget and mput. If globbing is turned off with glob, the file name arguments are taken literally and not expanded. Globbing for mput is done as in csh(1). For mdelete and mget, each remote file name is expanded separately on the remote machine and the lists are not merged. Expansion of a directory name is likely to be different from expansion of the name of an ordinary file: the exact result depends on the foreign operating system and FTP server, and can be previewed by doing "mls remote-files -". Note: mget and mput are not meant to transfer entire directory subtrees of files. That can be done by transferring a tar(1) archive of the subtree (in binary mode).

hash Toggle number-sign (#) printing for each data block transferred. The size of a data block is 1024 bytes.

help [command]
Print a description of command. With no argument, ftp prints a list of the known commands.

lcd [directory]
Change the working directory on the local machine. If no directory is specified, changes to the user's home directory.

ls [remote-directory] [local-file]
Print an abbreviated listing of the contents of a directory on the remote machine. If remote-directory is left unspecified, the current working directory is used. If no local file is specified, the output is sent to the terminal.

macdef macro-name
Define a macro. Subsequent lines are stored as the macro macro-name; a null line (consecutive newline characters in a file or carriage returns from the terminal) terminates macro input mode. There is a limit of 16 macros and 4096 total characters in all defined macros. Macros remain defined until a close command is executed. The macro processor interprets "$" and "\" as special characters. A "$" followed by a number (or numbers) is replaced by the corresponding argument on the macro invocation command line. A "$" followed by an "i" signals that macro processor that the executing macro is to be looped. On the first pass "$i" is replaced by the first argument on the macro invocation command line, on the second pass it is replaced by the second argument, and so on. A "\" followed by any character is replaced by that character. Use the "\" to prevent special treatment of the "$".

mdelete [remote-files]
Delete the specified files on the remote machine.

mdir remote-files local-file
Like dir, except multiple remote files may be specified. If interactive prompting is on, ftp will prompt the user to verify that the last argument is indeed the target local file for

receiving mdir output.

mget remote-files
Expand the remote-files on the remote machine and do a get for each file name thus produced. See glob for details on the filename expansion. Resulting file names will then be processed according to case, ntrans, and nmap settings. Files are transferred into the local working directory, which can be changed with "lcd directory"; new local directories can be created with "! mkdir directory".

mkdir directory-name
Make a directory on the remote machine.

mls remote-files local-file
Like ls, except multiple remote files may be specified. If interactive prompting is on, ftp will prompt the user to verify that the last argument is indeed the target local file for receiving mls output.

mode [mode-name]
Set the file transfer mode to mode-name. The default mode is stream.

mput local-files
Expand wild cards in the list of local files given as arguments and do a put for each file in the resulting list. See glob for details of filename expansion. Resulting file names will then be processed according to ntrans and nmap settings.

nmap [inpattern outpattern]
Set or unset the filename mapping mechanism. If no arguments are specified, the filename mapping mechanism is unset. If arguments are specified, remote filenames are mapped during mput commands and put commands issued without a specified remote target filename. If arguments are specified, local filenames are mapped during mget commands and get commands issued without a specified local target filename. This command is useful when connecting to a non-UNIX remote computer with different file naming conventions or practices. The mapping follows the pattern set by inpattern and outpattern. inpattern is a template for incoming filenames (which may have already been processed according to the ntrans and case settings). Variable templating is accomplished by including the sequences "$1", "$2", ..., "$9" in inpattern. Use "\" to prevent this special treatment of the "$" character. All other characters are treated literally, and are used to determine the nmap inpattern variable values. For example, given inpattern $1.$2 and the remote file name mydata.data, $1 would have the value mydata, and $2 would have the value data. The outpattern determines the resulting mapped filename. The sequences "$1", "$2", ..., "$9" are replaced by any value resulting from the inpattern template. The sequence "$0" is replaced by the original filename. Additionally, the sequence "[seq1,seq2]" is replaced by seq1 if seq1 is not a null string; otherwise it is replaced by seq2. For example, the command "nmap $1.$2.$3

[$1,$2].[$2,file]" would yield the output filename myfile.data for input filenames myfile.data and myfile.data.old, myfile.file for the input filename myfile, and myfile.myfile for the input filename .myfile. Spaces may be included in outpattern, as in the example:

```
nmap $1 | sed "s/  *$//" > $1
```

Use the "\" character to prevent special treatment of the "$", "[", "]", and "," characters.

ntrans [inchars [outchars]]
Set or unset the filename character translation mechanism. If no arguments are specified, the filename character translation mechanism is unset. If arguments are specified, characters in remote filenames are translated during mput commands and put commands issued without a specified remote target filename. If arguments are specified, characters in local filenames are translated during mget commands and get commands issued without a specified local target filename. This command is useful when connecting to a non-UNIX remote computer with different file naming conventions or practices. Characters in a filename matching a character in inchars are replaced with the corresponding character in outchars. If the character's position in inchars is longer than the length of outchars, the character is deleted from the file name.

open host [port]
Establish a connection to the specified host's FTP server. An optional port number can be supplied, in which case, ftp attempts to contact an FTP server at that port. If the auto-login option is on (default), ftp also attempts to automatically log the user in to the FTP server (see below).

prompt
Toggle interactive prompting. Interactive prompting occurs during multiple file transfers to allow the user to selectively retrieve or store files. If prompting is turned off (default), any mget or mput transfers all files and mdelete will delete all files.

proxy ftp-command
Execute an ftp command on a secondary control connection. This command allows simultaneous connection to two remote FTP servers for transferring files between the two servers. The first proxy command should be an open, to establish the secondary control connection. Enter the command "proxy ?" to see other ftp commands executable on the secondary connection. The following commands behave differently when prefaced by proxy: open will not define new macros during the auto-login process, close will not erase existing macro definitions, get and mget transfer files from the host on the primary control connection to the host on the secondary control connection, and put, mput, and append transfer files from the host on the secondary control connection to the host on the primary control connection. Third party file transfers depend upon support of the FTP protocol PASV

command by the server on the secondary control connection.

put local-file [remote-file]
Store a local file on the remote machine. If remote-file is left unspecified, the local file name is used in naming the remote file, after processing according to any ntrans or nmap settings. File transfer uses the current settings for type, format, mode, and structure.

pwd Print the name of the current working directory on the remote machine.

quit A synonym for bye.

quote arg1 arg2 ...
The arguments specified are sent, verbatim, to the remote FTP server.

recv remote-file [local-file]
A synonym for get.

remotehelp [command-name]
Request help from the remote FTP server. If a command-name is specified, it is supplied to the server as well.

rename [from] [to]
Rename, on the remote machine, the file from to the file to.

reset Clear reply queue. This command re-synchronizes command/reply sequencing with the remote FTP server. Resynchronization may be necessary following a violation of the FTP protocol by the remote server.

rmdir directory-name
Delete a directory on the remote machine.

runique Toggle storing of files on the local system with unique filenames. If a file already exists with a name equal to the target local filename for a get or mget command, a ".1" is appended to the name. If the resulting name matches another existing file, a ".2" is appended to the original name. If this process continues up to ".99", an error message is printed, and the transfer does not take place. The generated unique filename will be reported. Note that runique will not affect local files generated from a shell command (see below). The default value is off.

send local-file [remote-file]
A synonym for put.

sendport Toggle the use of PORT commands. By default, ftp attempts to use a PORT command when establishing a connection for each data transfer. The use of PORT commands can prevent delays when performing multiple file transfers. If the PORT command fails, ftp uses the default data port. When the use of PORT commands is disabled, no attempt is made to use them for each data transfer. This is useful for certain FTP implementations that do ignore PORT commands but wrongly indicate they

have been accepted.

status Show the current status of ftp.

struct [struct-name]
Set the file transfer structure to struct-name. The default structure is stream.

sunique Toggle storing of files on remote machine under unique file names. Remote FTP server must support the FTP protocol STOU command for successful completion. The remote server will report a unique name. Default value is off.

tenex Set the file transfer type to that needed to talk to TENEX machines.

trace Toggle packet tracing.

type [type-name]
Set the file transfer type to type-name. If no type-name is specified, the current type is printed. The default type is network ascii.

user user-name [password] [account]
The user identifies him/herself to the remote FTP server. If the password is not specified and the server requires it, ftp prompts the user for it (after disabling local echo). If an account field is not specified, and the FTP server requires it, the user is prompted for it. If an account field is specified, an account command will be relayed to the remote server after the login sequence is completed if the remote server did not require it for logging in. Unless ftp is invoked with "auto-login" disabled, this process is done automatically on initial connection to the FTP server.

verbose Toggle verbose mode. In verbose mode, all responses from the FTP server are displayed to the user. In addition, if verbose is on, when a file transfer completes, statistics regarding the efficiency of the transfer are reported. By default, verbose is on.

? [command]
A synonym for help.

Command arguments that have embedded spaces can be quoted with double quote (") marks.

ABORTING A FILE TRANSFER

To abort a file transfer, use the terminal interrupt key (usually <ctrl>C). Sending transfers will be immediately halted. Receiving transfers will be halted by sending a FTP protocol ABOR command to the remote server, and discarding any further data received. The speed at which this is accomplished depends upon the remote server's support for ABOR processing. If the remote server does not support the ABOR command, an ftp> prompt will not appear until the remote server has completed sending the requested file.

The terminal interrupt key sequence will be ignored when ftp has completed any local processing and is awaiting a reply from the remote server. A long delay in this mode may result from the ABOR processing described above, or from

unexpected behavior by the remote server, including violations of the FTP protocol. If the delay results from unexpected remote server behavior, the local ftp program must be killed by hand.

FILE NAMING CONVENTIONS

Files specified as arguments to ftp commands are processed according to the following rules.

1. If the file name is -, the standard input (for reading) or the standard output (for writing) is used.

2. If the first character of the file name is a bar |, the remainder of the argument is interpreted as a shell command. ftp then forks a shell, using popen(3S) with the argument supplied, and reads (writes) from the stdout (stdin). If the shell command includes spaces, the argument must be quoted; for example, "| ls -lt". A particularly useful example of this mechanism is "dir | more".

3. Failing the above checks, if globbing is enabled, local file names are expanded according to the rules used in the csh(1); see the glob command. If the ftp command expects a single local file (e.g., put), only the first filename generated by the globbing operation is used.

4. For mget commands and get commands with unspecified local file names, the local filename is the remote filename, which may be altered by a case, ntrans, or nmap setting. The resulting filename may then be altered if runique is on.

5. For mput commands and put commands with unspecified remote file names, the remote filename is the local filename, which may be altered by a ntrans or nmap setting. The resulting filename may then be altered by the remote server if sunique is on.

FILE TRANSFER PARAMETERS

The FTP specification identifies many parameters that can affect a file transfer. The type can be one of ascii, image (binary), ebcdic, and local byte size (for PDP-10's and PDP-20's mostly). ftp supports the ascii and image types of file transfer, plus local byte size 8 for tenex mode transfers.

ftp supports only the default values for the remaining file transfer parameters: mode, form, and struct.

OPTIONS

Options can be specified at the command line, or to the command interpreter.

The -v (verbose on) option forces ftp to show all responses from the remote server, as well as report on data transfer statistics.

The -n option restrains ftp from attempting "auto-login" upon initial connection. If auto-login is enabled, ftp checks the netrc file in the user's home directory for an entry describing an account on the remote machine. If no entry exists, ftp will prompt for the remote machine login name (default is the user identity on the local machine), and, if necessary, prompt for a password and an account with which to login.

The -i option turns off interactive prompting during multiple file transfers.

The -d option enables debugging.

The -g option disables file name globbing.

THE .netrc FILE

The .netrc file contains login and initialization information used by the "auto-login" process. It resides in the user's home directory. The following tokens are recognized; they may be separated by spaces, tabs, or new-lines:

machine name
: Identify a remote machine name. The auto-login process searches the .netrc file for a machine token that matches the remote machine specified on the ftp command line or as an open command argument. Once a match is made, the subsequent .netrc tokens are processed, stopping when the end of file is reached or another machine token is encountered.

login name
: Identify a user on the remote machine. If this token is present, the "auto-login" process will initiate a login using the specified name.

password string
: Supply a password. If this token is present, the "auto-login" process will supply the specified string if the remote server requires a password as part of the login process. Note that if this token is present in the .netrc file, ftp will abort the "auto-login" process if the .netrc is readable by anyone besides the user.

account string
: Supply an additional account password. If this token is present, the "auto-login" process will supply the specified string if the remote server requires an additional account password, or the "auto-login" process will initiate an ACCT command if it does not.

macdef name
: Define a macro. This token functions like the ftp macdef command functions. A macro is defined with the specified name; its contents begin with the next .netrc line and continue until a null line (consecutive new-line characters) is encountered. If a macro named init is defined, it is automatically executed as the last step in the "auto-login" process.

SEE ALSO

csh(1).
ftpd(1M) in the UMAX V Administrator's Reference Manual.

BUGS

Correct execution of many commands depends upon proper behavior by the remote server.

An error in the treatment of carriage returns in the 4.2BSD UNIX ASCII-mode transfer code has been corrected. This correction may result in incorrect transfers of binary files to and from 4.2BSD servers using the ascii type. Avoid this problem by using the binary image type.

APPENDIX L: telnet COMMANDS REFERENCE

NAME

telnet - user interface to the TELNET protocol

SYNOPSIS

telnet [host [port]]

DESCRIPTION

The telnet command communicates with another host using the TELNET protocol. If telnet is invoked without arguments, it enters command mode, indicated by its prompt (for example, telnet>). In this mode, it accepts and executes the commands listed below. If it is invoked with arguments, it performs an open command (see below) with those arguments. Once a connection has been opened, telnet enters input mode. The input mode entered will be either character at a time or line by line depending on what the remote system supports.

In character at a time mode, most text typed is immediately sent to the remote host for processing.
In line by line mode, all text is echoed locally, and (normally) only completed lines are sent to the remote host. The local echo character (initially ^E) may be used to turn off and on the local echo (this would mostly be used to enter passwords without the password being echoed).

In either mode, if the localchars toggle is TRUE (the default in line mode; see below), the user's quit, intr, and flush characters are trapped locally, and sent as TELNET protocol sequences to the remote side. There are options (see toggle autoflush and toggle autosynch below) which cause this action to flush subsequent output to the terminal (until the remote host acknowledges the TELNET sequence) and flush previous terminal input (in the case of quit and intr).

While connected to a remote host, telnet command mode may be entered by typing the telnet escape character (initially ^]). When in command mode, the normal terminal editing conventions are available.

COMMANDS

The following commands are available. Only enough of each command to uniquely identify it need be typed (this is also true for arguments to the mode, set, toggle, and display commands).

open host [port]
: Open a connection to the named host. If no port number is specified, telnet attempts to contact a TELNET server at the default port. The host specification can be either a host name (see hosts(4)) or an Internet address specified in "dot notation" (see inet(3N)).

close
: Close a TELNET session and return to command mode.

quit
: Close any open TELNET session and exit telnet. An end-of-file (in command mode) will also close a session and exit.

<ctrl>Z
: Suspend telnet. This command only works when the user is using the csh(1) or the BSD application environment version of ksh(1).

status Show the current status of telnet. This includes
        the peer one is connected to, as well as the
        current mode.

display [argument ...]
        Displays all, or some, of the set and toggle values
        (see below).

? [command]
        Get help. With no arguments, telnet prints a help
        summary. If a command is specified, telnet will
        print the help information for just that command.

send arguments
        Sends one or more special character sequences to
        the remote host. The following are the arguments
        which may be specified (more than one argument may
        be specified at a time):

        escape
             Sends the current telnet escape character
             (initially ^]).

        synch
             Sends the TELNET SYNCH sequence. This
             sequence causes the remote system to discard
             all previously typed (but not yet read) input.
             This sequence is sent as TCP urgent data (and
             may not work if the remote system is a 4.2 BSD
             system -- if it doesn't work, a lower case r
             may be echoed on the terminal).
        brk
             Sends the TELNET BRK (Break) sequence, which
             may have significance to the remote system.

        ip
             Sends the TELNET IP (Interrupt Process)
             sequence, which should cause the remote system
             to abort the currently running process.

        ao
             Sends the TELNET AO (Abort Output) sequence,
             which should cause the remote system to flush
             all output from the remote system to the
             user's terminal.

        ayt
             Sends the TELNET AYT (Are You There) sequence,
             to which the remote system may or may not
             choose to respond.

        ec
             Sends the TELNET EC (Erase Character)
             sequence, which should cause the remote system
             to erase the last character entered.

        el
             Sends the TELNET EL (Erase Line) sequence,
             which should cause the remote system to erase
             the line currently being entered.

        ga
             Sends the TELNET GA (Go Ahead) sequence, which
             likely has no significance to the remote
             system.

nop
Sends the TELNET NOP (No operation) sequence.

?
Prints out help information for the send command.

set argument value
Set any one of a number of telnet variables to a specific value. The special value off turns off the function associated with the variable. The values of variables may be interrogated with the display command. The variables which may be specified are:

echo
This is the value (initially ^E) which, when in line by line mode, toggles between doing local echoing of entered characters (for normal processing), and suppressing echoing of entered characters (for entering, say, a password).

escape
This is the telnet escape character (initially ^[) which causes entry into telnet command mode (when connected to a remote system).

interrupt
If telnet is in localchars mode (see toggle localchars below) and the interrupt character is typed, a TELNET IP sequence (see send ip above) is sent to the remote host. The initial value for the interrupt character is taken to be the terminal's intr character.

quit
If telnet is in localchars mode (see toggle localchars below) and the quit character is typed, a TELNET BRK sequence (see send brk above) is sent to the remote host. The initial value for the quit character is taken to be the terminal's quit character.

flushoutput
If telnet is in localchars mode (see toggle localchars below) and the flushoutput character is typed, a TELNET AO sequence (see send ao above) is sent to the remote host. The initial value for the flush character is taken to be the terminal's flush character.

erase
If telnet is in localchars mode (see toggle localchars below), and if telnet is operating in character at a time mode, then when this character is typed, a TELNET EC sequence (see send ec above) is sent to the remote system. The initial value for the erase character is taken to be the terminal's erase character.

kill
If telnet is in localchars mode (see toggle localchars below), and if telnet is operating in character at a time mode, then when this character is typed, a TELNET EL sequence (see send el above) is sent to the remote system. The initial value for the kill character is taken to be the terminal's kill character.

eof
If telnet is operating in line by line mode, entering this character as the first character on a line will cause this character to be sent to the remote system. The initial value of the eof character is taken to be the terminal's eof character.

toggle arguments ...
Toggle (between TRUE and FALSE) various flags that control how telnet responds to events. More than one argument may be specified. The state of these flags may be interrogated with the display command. Valid arguments are:

localchars
If this is TRUE, then the flush, interrupt, quit, erase, and kill characters (see set above) are recognized locally, and transformed into (hopefully) appropriate TELNET control sequences (respectively ao, ip, brk, ec, and el; see send above). The initial value for this toggle is TRUE in line by line mode, and FALSE in character at a time mode.

autoflush
If autoflush and localchars are both TRUE, then when the ao, intr, or quit characters are recognized (and transformed into TELNET sequences; see set above for details), telnet refuses to display any data on the user's terminal until the remote system acknowledges (via a TELNET Timing Mark option) that it has processed those TELNET sequences. The initial value for this toggle is TRUE if the terminal user had not done an stty noflsh, otherwise FALSE (see stty(1)).

autosynch
If autosynch and localchars are both TRUE, then when either the intr or quit characters is typed (see set above for descriptions of the intr and quit characters), the resulting TELNET sequence sent is followed by the TELNET SYNCH sequence. This procedure should cause the remote system to begin throwing away all previously typed input until both of the TELNET sequences have been read and acted upon. The initial value of this toggle is FALSE.

crmod
Toggle carriage return mode. When this mode is enabled, most carriage return characters received from the remote host will be mapped into a carriage return followed by a line feed. This mode does not affect those characters typed by the user, only those received from the remote host. This mode is not very useful unless the remote host only sends carriage return, but never line feed. The initial value for this toggle is FALSE.

debug
Toggles socket level debugging (useful only to the super-user). The initial value for this

toggle is FALSE.

options
Toggles the display of some internal telnet protocol processing (having to do with TELNET options). The initial value for this toggle is FALSE.

netdata
Toggles the display of all network data (in hexadecimal format). The initial value for this toggle is FALSE.

?
Displays the legal toggle commands.

SEE ALSO
csh(1), ksh(1), rlogin(1N).
inet(3N), services(4), hosts(4) in the UMAX V Programmer's Reference Manual.
telenetd(1M) in the UMAX V Administrator's Reference Manual.

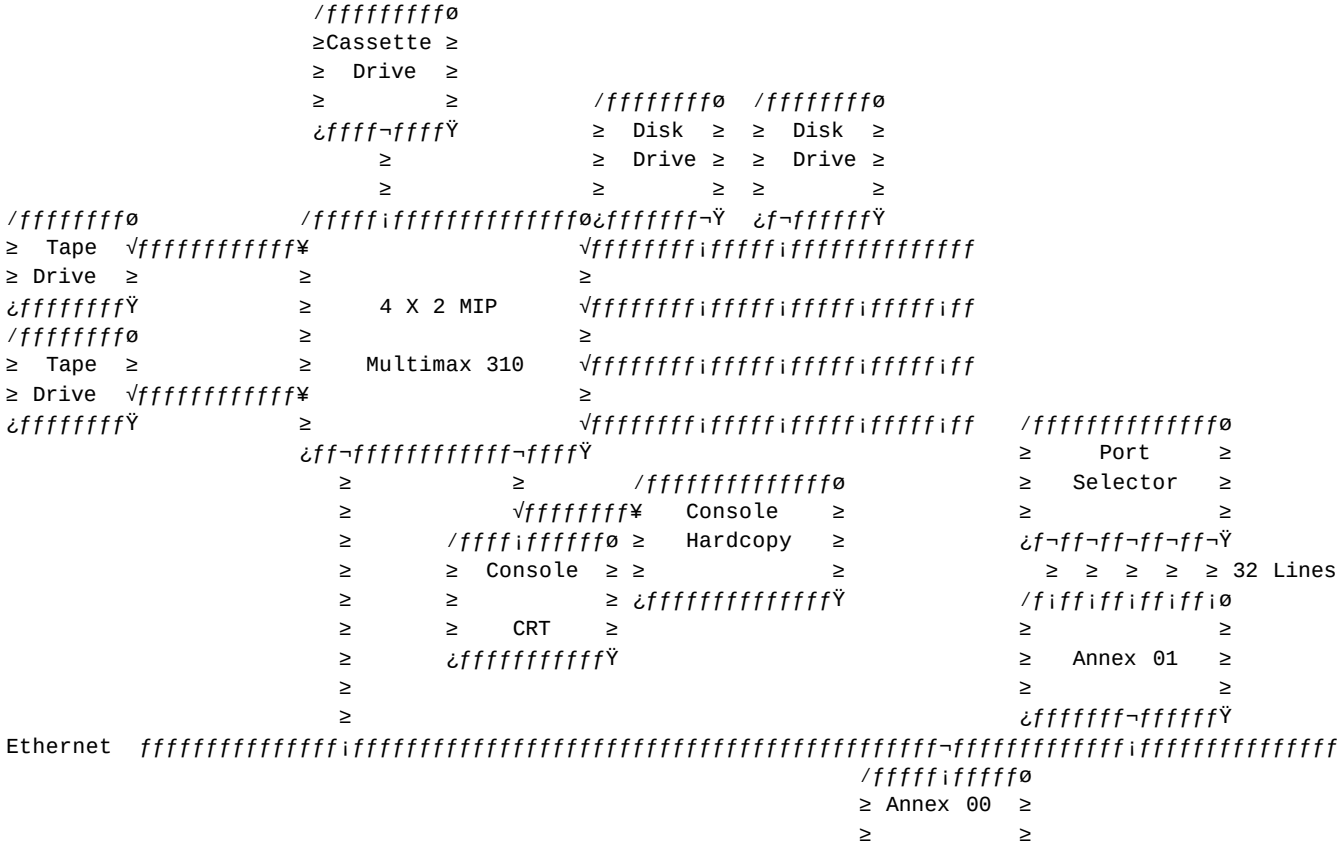
BUGS
There is no adequate way for dealing with flow control.
On some remote systems, echo has to be turned off manually when in line by line mode.

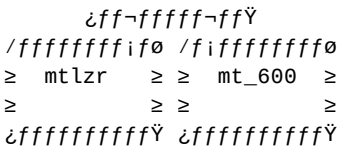
There is enough settable state to justify a .telnetrc file.

No capability for a .telnetrc file is provided.

In line by line mode, the terminal's eof character is only recognized (and sent to the remote system) when it is the first character on a line.

APPENDIX M: domax1 AND domax0 HARDWARE CONFIGURATION





NOTES

‹‹

APPENDIX N: BASIC UNIX REVIEW

Write the letter(s) of the UNIX component that best fit each description.

K = Kernel S = Shell U = Utilities D = Directory

_____ 1. Uses standard syntax for all commands.

_____ 2. Schedules tasks and manages data storage.

_____ 3. Memory resident code.

_____ 4. Main interface between UNIX and users.

_____ 5. Heart of the operating system.

_____ 6. Can be easily combined to perform the exact function which the user desires.

_____ 7. Path name concept.

_____ 8. Written mostly in the "C" programming language.

_____ 9. Multi-level directory structure.

_____ 10. Uses pipes and filters.

_____ 11. Supports control structures.

_____ 12. Includes text processing, electronic mail, file manipulation, and program generation.

NOTES

‹‹

INDEX