Commodore

Amiga Unix
System V Release 4

Learning Amiga-Unix

Preface

This manual is written for people who have never used UNIX before. It includes a series of introductory tutorials that cover most basic user tasks. After completing all tutorials, readers should move on to the comprehensive user guide, **UsingAmiga UNIX**. Experience with any desktop computer may be helpful, but is not required.

Credits

Written by Carol Wahl
Directed by Richard Buck
Reviewed by Jan Carlson, Mike Ditto, Ken Farinsky, Mike Hall, Kendall Robinson and Rich Skrenta
Artwork by Cynthia Robinson
Additional contributions by Dave Ballman Don Bein Jesse Bornfreund, Paul Calkin, Keith Gabryelski, Sam Frederick and Randy Gort
Produced by Johann George
This manual was written and produced using Amiga UNIX and the X Window System. 11/1990

Table of contents

About this manual

Who should read this manual?
This tutorial is designed for people who have never used UNIX before. Each set of instructions is written simply so a novice user can perform the tutorial tasks. If you are an advanced UNIX user, you probably don't need to read this manual.

Why should you use this manual?
A series of hands-on exercises shows you how to perform basic Amiga UNIX functions, including:

- using virtual screens
- logging in and logging out
- changing your password
- finding help
- using basic desk tools (displaying the date and time, displaying a calendar, and using a calculator)
- creating, copying, and deleting directories
- creating and editing text files
- printing, copying, deleting, moving, and renaming files
- working with foreground and background processes
- using the talk and electronic mail utilities to communicate with other uscrs
- viewing terminal information
- changing the way your keyboard works
- changing your virtual screens
- creating your own commands
- working with OPEN LOOK and the X Window System

How much time will this take?
The entire tutorial Lakes about four hours to complete; however, you should not work through the book all at once. Complete the exercises in the order they are presented in the book and complete one chapter at a time. Take notes as you go and pause between chapters to review what you've learned.

What other books should you read?
Read *Using Amiga UNIX* to learn more about UNIX. It presents many concepts and commands in a textbook format, so you can study a specific task or look up a command. This tutorial assumes you are sitting at a computer and learning UNIX; *Using Amiga UNIX* is a reference for you to use at any time.
Another helpful source of information is an on-line reference available through UNIX called man (manual) pages. You will learn about man later in the tutorial.

If you need more detailed information, read the
UNIX System V Release 4 User's Guide (ISBN 0-13-947052-2) or
User's Reference Manual (ISBN 0-13-947037-9) published by UNIX Press.

Read reference material In this order:
Learning Amiga UNIX
Using Amiga UNIX
Using Amiga UNIX reference pages
man pages
AT&T manuals

Conventions

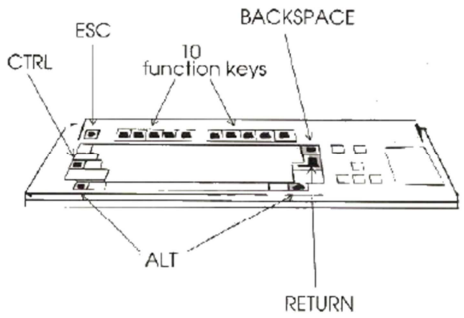
Four style conventions
We use four typographic conventions in this manual to highlight specific concepts.

Convention	What does it mean?
bold words	a UNIX command or command line: **ls -lt** **man ls**
italics	a value to be substituted, frequently as part of a command line: *filename* **cat *filename***
uppercase	a key on the keyboard, or a combination of keys if cmmected by a hyphen: RETURN CTRL-C
typewriter font	text that appears on your screen either as you type it or in response to your commands; values to be substituted by you are in italics

CTRL key combinations
Whenever you see a control key combination, hold down the CTRL key and press the character on the keyboard. For example, CTRL-C means hold down the CTRL key and press C.

RETURN
The RETURN key on a standard 3000UX keyboard is the large arrow key on the right of the typewriter keypad.

BACKSPACE
The BACKSPACE key is the left arrow key above the RETURN key.

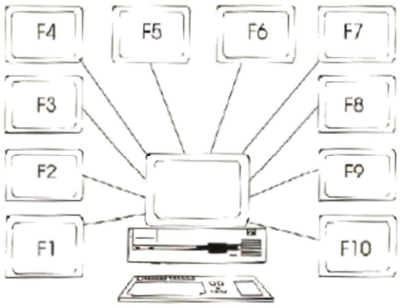


Special keys on your keyboard

If you use the X Window System
If you‘re using the X Window System, type the Window SYSTEM commands as they are written in this tutorial; you do not have to follow any special procedures. Refer to the *Changing your environment chapter* for information about using X.

Learning the basics

Virtual screens



You have up to ten virtual screens that you can use as individual terminals. You move between the screens by pressing ALT-*functionkey*.

Basic commands

passwd	change password
man	read on-line reference
date	display current date and time
cal	display a calendar
bc	use the calculator
sc	use the spreadsheet program

Log in and out

To log in, enter your *username* at the login prompt. **If** you have a password, enter it at the password prompt.

To log out, simply press CTRL-D.

What's in this chapter?

This chapter contains exercises to help you get started on UNIX. You'll learn how to:

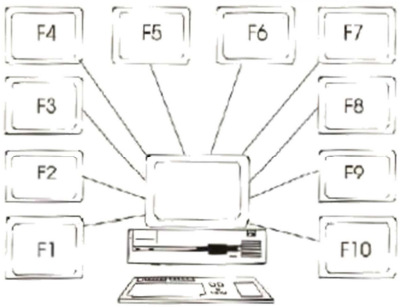
- use virtual screens
- log in and log out
- change your password
- get help
- use simple desk tools

Before you begin

You or the system administrator set up an account for you on the system. You need to know your username and password in order to log in.

Look at your virtual screens

What are "virtual screens"?
"Virtual screens" make your single computer work like eight or more terminals, all using the same monitor and keyboard. Let's look at your virtual screens.



Your computer can have up to 10 virtual screens

Press ALT-functionkey to move between screens
1. Hold down ALT and press the function keys (F2 through F10), one at a time. The ALT-*functionkey* combination moves you among your virtual screens.

If F9 and F10 don't work, those virtual screens haven't been turned on yet. You system is installed with F1 through F8 turned on; you can also turn on F9 and F10, but this uses more memory. Refer to *Using Amiga UNIX* for instructions.

As you can see, each screen has a login prompt; this allows you to log in to each screen just as if you had ten separate computers. The letter size and colors are set up differently for each virtual screen. You can change the screen settings to suit your needs. You'll practice changing your screens in an exercise later in this tutorial.

Don't work on the F1 console screen
2. Although you can use any of the virtual screens for the tutorial, you should avoid using the first screen (ALT-F1) because it's the console screen and is normally reserved for special processing.

Let's use virtual screen 2 for the following exercise; press ALT-F2.

Log in and log out

Type your username
1. Type your *username* next to the login prompt, and press RETURN.

UNIX is case sensitive
Be sure to pay attention to upper and lower case sensitive letters; UNIX is case sensitive. For example, if you type **Man** rather than **man**, UNIX responds with an error. If you make an error, just press RETURN until the login prompt appears again.

```
login: joe                          <- login prompt
Password:                           <- password prompt
UNIX System V Release 4.0 AT&T
utopia
Copyright (C) 1984, 1986, 1987, 1988 AT&T '
All Rights Reserved
Last loqin Tue Auq 14 16:31:05 on con2
           System V Release 4.0
Welcome to utopia                   <- message of the day
you have mail                       <- mail notice
$
```

Enter your password
2. If your account has a password, enter it and press RETURN.

Your password docsn't appear on the screen when you type it because it is supposed to remain confidential. Your password prevents other users from logging in to your account.

Read your system messages
3. Most likely, you won't have any system messages yet, but if any appear on your screen, read them. System messages include important notes from your system administrator or notices from UNIX processes. For example, if you have any mail messages, UNIX displays the "you have mail" notice. (You'll learn how to use the **elm** electronic mail utility later in the tutorial.)

Look at your prompt
4. Look at your shell prompt. Depending upon how your system is set up, either % (percent sign) or $ (dollar sign) appears as your shell prompt. $ is the Korn shell prompt; % is the C shell prompt.

What's a shell?
A shell is the interpreter that UNIX uses to understand the commands you type. Almost all commands work the same way with each of the shells; however, some shells have additional features. Refer to *Using Amiga UNIX* for descriptions of the various shells.

Once the shell prompt appears, you have successfully logged in to your system, and you can begin to use UNIX commands.

Log out
5. Log out of the system by pressing CTRL-D.
6. Press ALT-F2 to access the F2 screen again.

Log in again
7. Log back in to your system so you can perform the next exercises.

Change your password

To show you how easy UNIX is to use, let's change your password.

Change your password
1. At the shell prompt, type **passwd** and press RETURN.
2. If you already have a password, type it at the old password prompt. Ifyou don't have a password, this prompt doesn't appear.
3. When the **passwd** command asks you to type a new password, type change1 and press RETURN. Remember, you won't see the password on your screen because it's confidential.
4. The passwd command asks you to type the new password again; this step ensures you did not make a spelling mistake while typing the password the first time. Retype **change1** and press RETURN. Your new password is change1.

```
$ passwd
Old password:
New Password:
Re-enter new password:
$
```

Change password steps 1, 2, 3, and 4

Log out
5. To test your new password, log out of the system by pressing CTRI-D.
6. Press ALT-F2 to log in to the F2 virtual screen again.

Log in with new password
7. Log in again by typing your *username* at the login prompt.
8. Type **change1** for your password and press RETURN. Your shell prompt appears.

Change your password again
Change your password again back again or create a new one using these same steps.

NOTE: A password must be at least six characters long and must contain at least two alphabetic characters and at least one numeric or special character. For example, **test**, **test1**, and **123456** are not valid passwords; **testing1** is.

Find help

Use the man command for help
If you forget how to use a command or want to learn more about one, type **man *command*** and press RETURN to read an on-line description of the function. The on-line **man** (manual) pages describe each command in detail.

Read the man page for cp
1. Type **man cp** and press RETURN to view the on-line reference for the **cp** command

```
Reformatting page. Wait…done
Cp(1)            USER COMMANDS          cp(1)
NAME:
     cp – copy files
SYNOPSIS:
     Cp [-i][-p][-r]file1 [file2…] target
DESCRIPTION:
     The cp command copies files to target
                .
                .
                .
--More–(43%)
```

Sample man page

Try the man command keys
2. The prompt at the bottom of the screen indicates there's more of the file to read. Use the command keys in the table on the next page to move around the **man** pages. Wait until the prompt appears before trying another command.

Man command keys

Key	Description
RETURN	display next line
SPACE	display next screen
b	display previous screen
h	display help for man pages
q	quit

more
man pages are "piped" (sent) through the **more** command, which displays text files a page at a time; the keys listed in the chart above are display commands that work with **more**.

man passwd
3. Try another **man** page. Type **man passwd** and press RETURN. Press SPACE to read the next screen. Press **b** to go back to the previous screen.
Press **q** to quit.

man man
Find out more about **man**; type **man man** and press RETURN.

Use desk tools

Desk tools
These exercises introduce you to some basic desktop tools:

- clock
- calendar
- calculator
- spreadsheet program

Display date and time
1. First, let's look at the date and time. At your shell prompt, type **date** and press RETURN. The **date** command acts as an on-line clock; it displays the current day, date, and time.

NOTE: UNIX uses a 24-hour clock. (1PM is 13:00)

```
$ date
Thu Jul 12 09:23:29 EDT 1990
$
```

Display the date and time, step 1

Display calendar
2. Now, let's look at an on-line calendar. Type **cal** (calendar) and press RETURN. **cal** displays a calendar for the current month.
You can look at a calendar for any month or year by adding "arguments" to the **cal** comm and line. (Argument is the UNIX term for additional information typed after a command.) The **cal** ***month year*** command displays any month of any year.

Display year
3. Let's look at the calendar for 1991. Type **cal 1991** and press RETURN. The calendar rapidly scrolls down your screen.

The more command
4. Use the **more** command to display part of a calendar at a time. Separate the **cal** command from the **more** command with a I (pipe). The pipe key is next to the BACKSPACE key; it's the two vertical lines above the backslash.
Type **cal 1991 I more** to display part of the calendar at a time. Press the SPACEBAR or RETURN to display more lines.
You'll learn more about the **more** command later in the tutorial.

Display month
5. To look at March of 1991, type **cal 3 1991** and press RETURN.

```
$ cal 3 1991
          March 1991
S     M     Tu    W     Th    F     S
                                1     2
3     4     5     6     7     8     9
10    11    12    13    14    15    16
17    18    19    20    21    22    23
24    25    26    27    28    29    30
31
```

Display a calendar, step 5

Try the on-line calculator

6. Now, let's use the on-line calculator. Type **bc** (binary calculator) and press RETURN. Notice that the prompt disappears; this is because **bc** is waiting for you to start typing. (The shell prompt returns when you exit from **bc**.)

7. Type **2 + 2** and press RETURN. **bc** displays the answer to your equation (4).

```
$ bc
2+2
4
```

Try the on-line calculator, steps 6 and 7

You can perform several kinds of calculations with the on-line calculator. Here's a list of some of the symbols you use for basic mathematical operations.

Symbol	Operation
+	addition
-	subtraction
*	multiplication
/	division

Quit the calculator

8. To quit the calculator, press CTRL-D. Your shell prompt appears, and you're ready to perform more UNIX commands.

Look at the spreadsheet program

Amiga UNIX also includes a public domain spreadsheet program. To access the program, type **sc** (spreadsheet calculator) and press RETURN. A standard spreadsheet Format displays.

```
sc 6.1:    Type '?' for help
```

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							

Sample spreadsheet

Read sc help

10. We won't create a spreadsheet now, but you can read the on-line help available with the **sc** command. Press **?** to see the overview list and then press **b** through **n** to view the help selections.

Quit sc help

11. Quit out of the **sc** help by pressing **q**.

Quit sc

12. To quit the spreadsheet program, press **q**. Your shell prompt appears.

For more information

For more information about the desk tools, read the **man** pages for **more**, **date**, **cal**, **bc**, and **sc**.

Chapter review

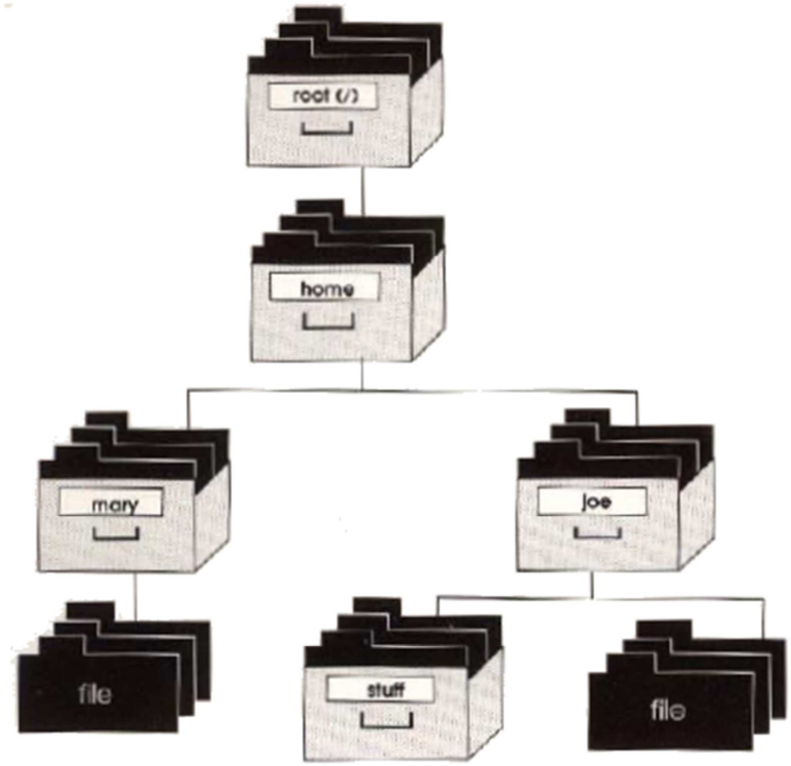
Here are some important points you learned in this chapter:

- You have up to ten virtual screens that you can use as individual terminals.
- You need to know your username and password (if you have one) to log in.
- To log out, press CTRL-D.
- Use the **passwd** command to change your password.
- Type **man *command*** to view the on-line reference or a specific command
- Amiga UNIX provides a variety of on-lino desk tools:

  date (clock)
  cal (calendar)
  bc (calculator)
  sc (spreadsheet program)

Working with directories

Change directories



You're in /home/joe and you want to move to /home/mary, type one of the following:
cd ../mary
cd /home/mary

You'‘re in /home/joe and want to move down to stuff type one of the following:
cd stuff
cd /home/joe/stuff

You'‘re in /ome/joe and you want to move up to /home, type one of the following:
cd ..
cd /home

Directory commands

pwd	display the current directory name
mkdlr	create a directory
cd	change the current directory
ls	list the contents of a directory
ls -l	list files in long format
ls -a	list all files
ls -al	list all files in long format
ls -lt	list files by time in long format
ls -lrt	list files by reverse time in long format
mv	move or rename a directory or file

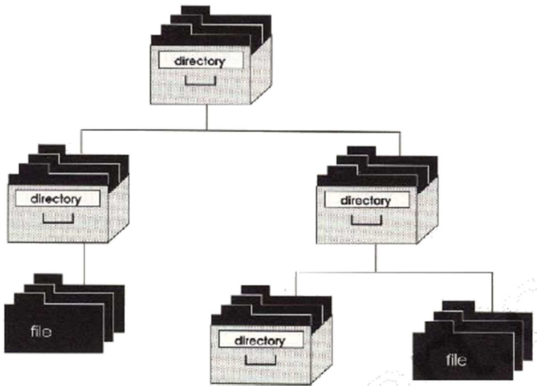
What's in this chapter?
This chapter contains exercises for working with directories, including such tasks as:

- showing the current directory
- creating new directories
- changing the current directory
- listing the contents of a directory
- renaming directories
- removing directories

Before you begin
Before you work with them, you need to understand what directories are, how they're named, and how to move among them.

The following pages briefly describe directories and directory structures. For a more complete explanation, refer to the *Working with files and directories* chapter in *Using Amiga UNIX*.

What's a directory?
Basically, directories are special types of files that contain files and other directories.



Directory structure

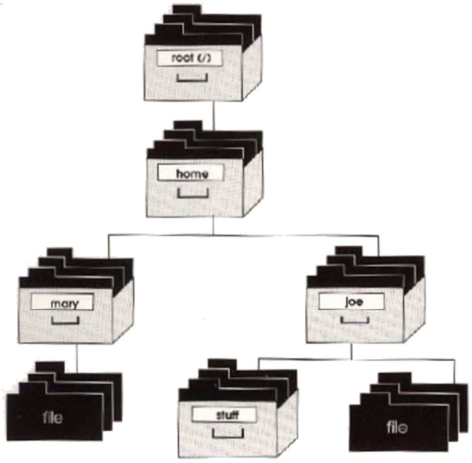
Pathname
A "path name" tells you the path through which you must go to reach a file or directory. The root directory is the first level of the directory structure (and is the first character of a pathname); all other directories and files branch out from the root directory.

Home directory
For example, your home directory (your login directory) might be called /home/joe. The first / indicates the root directory; root has a directory called home in it. Your directory (joe) is in the home directory.

Change and create directories

Changing the working directory
You can move up and down through directories, or go directly to at directory by using its pathname. Here are some ways to move among directories using the **cd** (change directory) command:

cd	move directly to your home directory from anywhere
cd ..	move up one level
cd stuff	move down one level
cd /home/mary	move directly to the specified directory
cd ../mary	move up, then down



Moving among directories

Home directory
1. Type **cd** and press RETURN to change to your home directory. A home directory is the directory you normally find yourself in when you log in to the system.

Show the current directory (pwd)
2. Make sure you're in your home directory. Type **pwd** (print working directory) and press RETURN. The **pwd** command displays the complete pathname of the current directory.

Create a directory (mkdir)
3. Let's create a directory. Type **mkdir mystuff** and press RETURN. The **mkdir** (make directory) command creates new directories.

Change current directory (cd)
4. Now, move to the mystuff directory. Type **cd mystuff** and press RETURN.

5. Type **pwd** to list your current directory.

```
$ cd
$ pwd                    <- display current directory
/home/joe
$ mkdir mystuff          <- create mystuff directory
$ cd mystuff             <- change to mystuff
$ pwd                    <- display current directory
/home/joe/mystuff
$
```

Changing and creating directories, steps 1 to 5

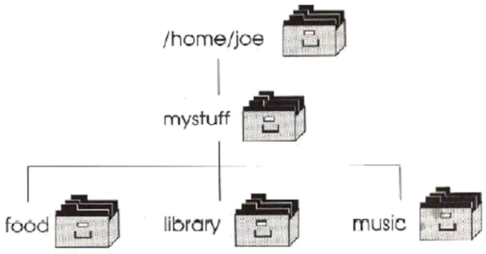
6. Let.'s make some more directories. Type **mkdir food library music** and press RETURN.

Create more directories

You can create more than one directory at a time by listing the names after the **mkdir** command. Don't put commas between the names.

When you look for these directories, they are listed in mystuff because you created them while mystuff was your current directory.

Your directory structure now looks like this:



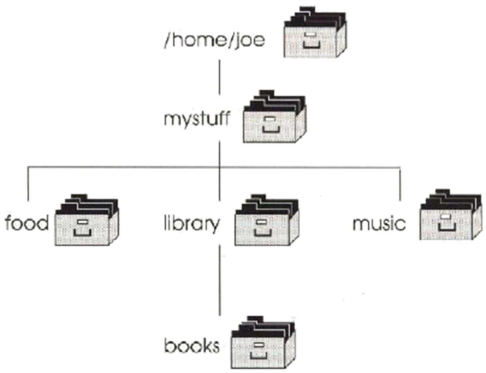
Your directory structure

Make library the current directory

7. Let's create a directory under library. Type **cd library** and press RETURN to make library your current directory. Remember that library is in mystuff; if you are not in mystuff, **cd library** will not work. Change to mystuff first, or go directly to library by typing **cd /home/*username*/mystuff/library**. (In our case, we used /home/joe/mystuff/library. Use your own name in place of joe.)

Your directory structure

Your directory structure now looks like this:



Your directory structure

List directory contents

List directory contents (ls)
You can list all the files in your directories and list file information such as size, type, time last modified, owner, and permissions.

ls (list)
1. Let's list the contents of your home directory. First type **cd** and press RETURN to make sure you are in your home directory. Then, type **ls** (list) and press RETURN to look at the contents of the directory.

ls -l (long list)
2. Now, type **ls -l** and press RETURN.

The ls command provides a list of file and directory names, in alphabetic order. Adding **-l** to the **ls** command creates an alphabetic list in a longer format. The **ls -l** command provides more information about the files, including the file type; the read, write, and execute permissions; the number of blocks in the file; the owner and group; the date and time the file was created or last modified; and the name of the file.

```
$ ls
mystuff
$ ls -l
1     2           3     4     5          6                  7
Drwxr-xr-x      5     joe   gp1   80    Aug 8 09:53      mystuff
```

List directory contents, steps 1 and 2

1: file type (d=directory)
2: Permissions (read, write, execute) dgfe/time
3: size
4: owner
5: group
6: date/time modified
7: name

ls -a (all files)
3. Let's try another **ls** command. Type **ls -a** and press RETURN. You see an alphabetic list of short names, just as if you typed **ls**; however, there are more file names listed. File names beginning with a period show up on the **ls -a** list. These files (called dot files) are normally hidden from users because they are special system configuration files that you won't access frequently.

```
$
$ ls -a
.     ..    .profile        mystuff
$
```

List directory contents, step 3

ls -al (all files, long list)

4. Type **ls -al** and press RETURN to look at the same list in the long format.

```
$ ls -al
drwxr-xr-x 5    joe   gp1   512   Aug 8 09:52     .
drwxr-xr-x 8    bin   bin   128   Jul 7 12:40     ..
-rw-r--r-- -    joe   gp1   191   Jul 7 12:36     .profile
-rw-r--r-- 1    joe   gp1   40    Jul 8 08:30     .rhosts
Drwxr-xr-x 5    joe   gp1   80    Aug 8 09:53     mystuff
$
```

List directory contents, step 4

5. Let's look at the contents of the mystuff directory. First, change the current directory to mystuff. Type **cd mystuff** and press RETURN.

ls lt (long list by time)

Type **ls -lt** and press RETURN. This list displays files in the order the files were created or modified, beginning with the most recent change.

```
$ cd mystuff
$ ls -lt
drw-r--r-- 1    joe   gp1   40     Aug 8 09:58    music
drwxr-xr-x 5    joe   gp1   512   Aug 8 09:53     library
drwxr-xr-x 5    joe   gp1   80    Aug 8 09:50     food
$
```

List directory contents, steps 5 and 6

ls -lrt (long list, reverse time)

To reverse the order of a listing, add **-r** to any of the **ls** commands. For example, type **ls -lrt** and press RETURN for a list of files displayed in order of the earliest date and time to the most recent date and time.

```
$ cd mystuff
$ ls -lt
drwxr-xr-x 5    joe   gp1   80    Aug 8 09:50     food
drw-r--r-- 1    joe   gp1   40     Aug 8 09:58    music
drwxr-xr-x 5    joe   gp1   512   Aug 8 09:53     library
$
```

List directory contents, step 7

Rename directories

Use the mv (move) command to rename

1. Renaming directories is very easy. Let's rename the mystuff directory. First, type **cd** to change to your home directory (/home/joe) and press RETURN.
2. Type **ls** and press RETURN. You see the mystuff directory listed in the contents of the home directory.
3. Now, type **mv mystuff junk** and press RETURN. You've just changed the name of the mystuff directory to junk.
4. Check the contents of your home directory. Type **ls** and press RETURN. You should see junk listed rather than mystuff.

```
$ cd                                                    <- move to home directory
$ ls -l                                                 <- list contents
drwxr-xr-x 5 joe gp1  80  Aug 8 09:53 mystuff           <- name is mystuff
$ mv mystuff junk                                       <- change directory name
$ ls -l
drwxr-xr-x 5 joe gp1  80  Aug 8 09:53 mystuff           <- name changed to junk
$
```

Renaming directories, steps 1-4

Remove directories

Delete empty directories

Removing empty directories is easy. However, if you want to delete directories that have files or directories in them, you first have to empty them. Removing directories with contents can be dangerous; refer to the *Using Amiga UNIX* manual for instructions.
Since the books directory is empty, let's delete it.

Start at your home directory

1. First, check your current directory (**pwd**), which should be your home directory. If your home directory is not your current directory, type **cd** and press RETURN.

Move to library

2. Move to the library directory (**cd junk/library**).

Check library contents

List the contents of the library directory (**ls -l**). (The library directory should include only the books directory.)

rmdir (remove directory)

Type **rmdir books** and press RETURN.
To make sure the books directory is gone, type **ls -l** to list the contents of library. The library directory is now empty.

```
$ pwd                                                    <- display current directory
/home/joe
$ cd junk/library                                        <- goto libary
$ ls -l                                                  <- list contents
drwxr-xr-x 1 joe gp1  10  Aug 14 14:30 books
$ rmdir books                                            <- remove books directory
$ ls -l                                                  <- list contents of library
total 0
$
```

Removing directories, steps 1-5

Chapter review

Here's a list of the commands you learned in this chapter:

pwd	display the current directory name
mkdlr	create a directory
cd	change the current directory
ls	list the contents of a directory
ls -l	list files in long format
ls -a	list all files
ls -al	list all files in long format
ls -lt	list files by time in long format
ls -lrt	list files by reverse time in long format
mv	move or rename a directory or file
rmdir	remove an empty directory

Editing files using vi

Special keys

ESC	cancel insert mode
CTRL-L	redraw screen
CTRL-D	scroll down
CTRL-U	scroll up

Insert and append

i	insert text
a	add text
o	add blank line

Quit and save

:q!	quit without saving
:wq	save and quit a file
:w	save your work

Change text

cw	change word
cc	erase a line and insert
C	change to end of line

Delete text

x	delete character
dw	delete word
dd	delete line
*n*dd	delete *n* lines

Undo and repeat

u	undo last change
U	undo changes to line
. (dot)	repeat last change

Move and copy

dd	cut
yy	yank (copy)
p	paste below

Search

/	search
N	repeat search
N	reverse search

Move in a file

w	to next word
0	back to the beginning of line
$	to end of line
G	to last line
*n*G	to the *n*th line
k	up
j	down

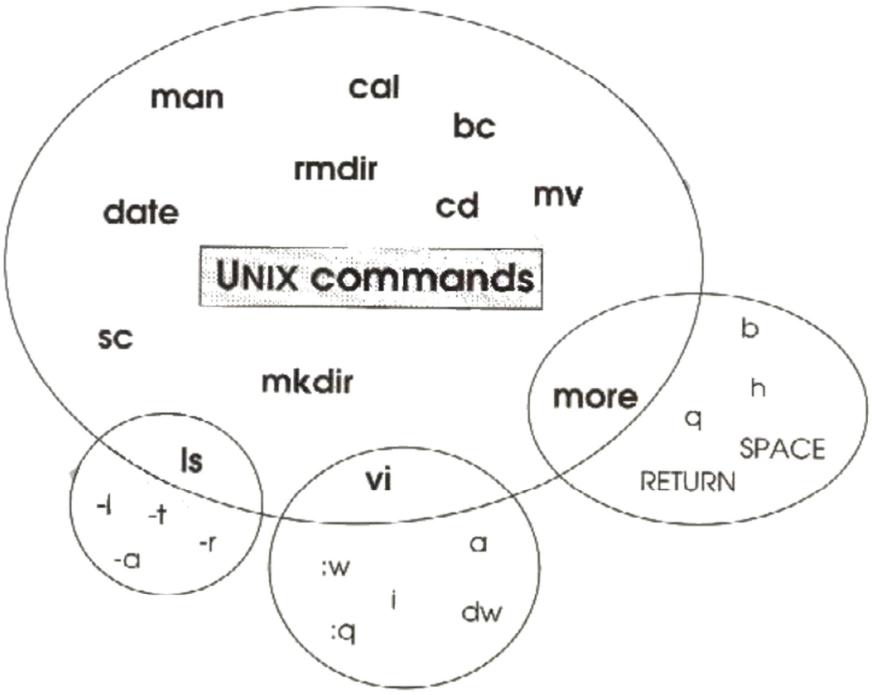
h left
l right

What's in this chapter?

Although you have a number of UNIX programs you can use to edit and format text, this tutorial focuses on the **vi** (visual) editor. ("vi" is pronounced "vee eye".) The exercises in this chapter show you how to create, save, quit, and edit files by using **vi** commands.

vi commands are different from UNIX commands. You can't use **vi** commands from a shell prompt, and you can't use UNIX commands inside **vi**.

Since a lot of material is covered in this chapter, each section has its own review.



Unix commands can have their own options and commands

Before you begin
You don't need to know **vi** to complete these exercises. This chapter is designed to give you an overview of the **vi** editor, but it doesn't provide a lot of detail. For a more complete list of **vi** commands, refer to the *Using the vi editor* chapter in *Using Amiga UNIX*.

Follow directions carefully
Follow the directions carefully; you could easily make a mistake and become confused if you press keys you aren't told to press.

vi is a modal program
vi is a modal program; that is, keys perform different tasks depending upon which mode you're in. If you're in the wrong mode, you may have trouble getting out to continue the exercise. If this happens, try pressing ESC.

The base mode is command mode, where most keys start a command. You can temporarily switch to either of the other modes (insert or file), then return to command mode. The three modes are described in the chart on the next page.

3 modes in vi
Here are the three modes you find in **vi**.

Mode	Description
insert	enter with an insert command (i, I, o, O, a, A); during insert mode, characters are entered as text; exit by pressing ESC
command	numbers and letters act as editing commands; they don't appear on the screen and are not inserted in the file
file	begin these commands with a colon (:w, :q, :q!) and execute by pressing RETURN

Moving within a vi file
Here are the commands you use to move around in **vi** file. Remember to use them only in command mode. You can also use the arrow keys to move the cursor.

Command	Description
k	move cursor up
j	move cursor down
H	move cursor to the left
l	move cursor to the right
w	move cursor by word
0 (zero)	move cursor to beginning of line

Create files in vi

Change to the food directory

1. Let's make some files to put into your food directory. First, move to your home directory (**cd**). Then move down to the food directory (**cd junk/food**).

2. Type **pwd** and press RETURN to view your current directory; make sure your current directory is food.

Create the snacks file

3. Now type **vi snacks** and press RETURN. The **vi *filename*** command either opens a file (if it exists in the current directory) or creates a new file.

```
$ cd
$ cd junk/food
$ pwd
/home/joc/junk/flood
$ vi snacks
```

Creating tiles, steps I-3

You've just created a file called snacks. Your screen fills with lines, each starting with a tilde (~). The filename is at the bottom of the screen.

```
~
~
~
"snacks" [New file]
```

Sample vi file

Press i to begin inserting text

4. Let's put some text into the snacks file. Press **i** to begin the text insertion mode of the **vi** editor. The letter doesn't appear on your screen because **vi** is expecting a command. Until you press an insert command key (**i,I,a,A,o,O**), nothing you type is added to your file.

5. Type **cake** and press RETURN. See how the text appears on your screen?

6. Type **cooks** on the next line, but don't press RETURN yet.

BACKSPACE and rekey to correct mistakes

7. You really need to type **cookies** rather than **cooks**. To correct your "mistake", press the BACKSPACE key. The „s" in cooks doesn't disappear, but you can type over it.

8. Type **ies** to make cookies. If you make a mistake, press the BACKSPACE key and retype. Press RETURN.

Add more lines

9. Type **fruit** on the next line and press RETURN.

10. Type **potato chips** on the next line and press RETURN.

11. Let's save and exit the snacks file. Type **:w** to save the file.

12. The **:w** appeared on the text line; that's because you're still in insert mode. BACKSPACE over the **:w**; it remains on your screen but is not actually part of your file.

Press ESC to quit insert mode
13. Exit text insert mode by pressing ESC.

Save a file (:w)
14. Now, type **:w** and press RETURN.

```
cake
cookies
fruit
potato chips
~
~
:w
„snacks“ [New file] 4 lines, 32 characters
```

Saving a file, step 14

The **:w** appears briefly on the bottom of your screen.
You just saved the snacks file.

File commands are preceded by a colon and require you to press RETURN before they execute.

Quit a file (:q)
15. Quit the file. Type **:q** and press RETURN to quit. Your shell prompt is back again.

```
cake
cookies
fruit
potato chips
~
~
~
:q
```

Quitting a file, step 15

Create more files

Create a new file
1 . Let's create another new file and practice editing it. Type **vi world** and press RETURN.

2. Type the following lines into the world file.

```
french
mexican
italian
```

Press i for text insertion
Remember, you need to start insert mode by pressing **i** before you can type text. (You can also start insert mode by using the other insert commands: **I, a, A, o, O**.)

Try to quit
3. Press ESC to quit insert mode, then quit the file by typing **:q** and pressing RETURN. What happens? Because you have not saved the file yet, **vi** displays a warning message at the bottom of the screen.

```
french
mexican
italian

:q
No write since last change (:quit! overrides)
```

Try to quit a file, step 3

Quit without saving (:q!)
4. Let's quit without saving this file. Typ **:q!** and press RETURN. The **!** (exclamation point) at the end of the **:q** command overrides the warning message so you can quit without saving the file.

Open a file
5. Let's add more text to the snacks file. First, open the file again; type **vi snacks** and press RETURN.

Start insert mode
6. Move the cursor down to the beginning of the word "cookies" with the arrow key, then press **i** (insert mode).

7. Type **chocolate** to make chocolate cookies.

8. Press ESC to quit insert mode.

Insert blank line and enter insert mode (o)
9. Move the cursor down to the last line with the arrow keys. Press **o** to begin inserting text on a new line below the cursor.

10. Type **pie** and press RETURN.

11. Type **tarts** and press RETURN.

```
cake
chocolate cookies
fruit
potato chips
pie
tarts
~
~
```

Add to a file, steps 7-11

12. Move the cursor up to the "e" in "chocolate".

Add text after cursor (a)

13. Let's add "chip" to make chocolate chip cookies. Since you're in command mode, press **a** to enter insert mode and type **chip**. The **a** command inserts characters after the cursor.

Combine save and quit (:wq)

14. Press ESC (to quit insert mode), then type **:wq** and press RETURN. (**:wq** combines the save and quit commands.)

```
cake
chocolate cookies
fruit
potato chips
pie
tarts
~
:wq
```

Add to a file, steps 12-14

REVIEW

Here's a list of concepts and commands you learned in this section.

Open a file

Type **vi *filename*** to open a file.

3 modes in vi

Mode	Description
command	keys perform editing commands
insert	insert command keys (**i,I,a,A,o,O**) add text until you press ESQ to quit
file	commands are preceded with a colon and are used to save and quit a file (**:w,:q,:wq**)

Commands learned in this section

Command	Description
i	insert text (start insert mode)
a	add text (Start insert mode)
o	insert a blank line and start insert mode
:w	save a file
:q	quit a file
:wq	save and quit a file
:q!	quit. without saving

Delete text and undo changes

Although you learned how to create and save files in the previous exercise, you haven't learned how to delete or undo your mistakes. This exercise shows you how to quickly and easily correct errors.

Create another file
1. Let's create another file. Type **vi groups** and press RETURN. Now type **meat**. What happens?

2. Only the "L" in meat appears. Why?

```
t
~
~
~
```

Create another file, steps 1 and 2

Remember to change modes
Because you are not yet in insert mode! The **vi** editor ignored the "m" and the "e" because they aren't valid **vi** commands, but read the „a" as an insert command.

3. BACKSPACE over the "t", type **meas** and press RETURN. The text appears this time because you started insert mode.

4. "Meas" isn't a real word. Let's delete the "**s**". Press ESC to quit insert mode, then use the arrow keys to move the cursor over to the "s".

Delete a character (x)
5. Press **x** to delete the letter. Remember, since you're back in command mode, **x** acts as a command key, not as a character.

6. Press **a** to enter insert mode, type **t**, and press RETURN.

Add more lines
7. Type the following new lines into the file. Remember to press RETURN after each line.

```
meat
fruits and vegetables
cereals and candy
dairy products
```

8. Let's get rid of the words "and candy" on the third line. Press ESC to exit insert mode and use the arrow keys to move the cursor to the word „and".

Delete a word (dw)
9. Type **dw** to delete the word "and."

```
meat
fruits and vegetables
cereals candy
dairy products
```

Delete 0 word, step 9

Undo (u)

10. Let's try to undo the delete. Press **u**. What happens?

```
meat
fruits and vegetables
cereals and candy
dairy products
```

Undo 0 change, step 10

The word "and" reappeared. **u** acts like an undo key when you're in command mode. It can undo everything you entered on a line since starting insert mode, but only if you press it as soon as you return to command mode.

Delete a line (dd)

11. Now, let's delete a whole line. Type **dd** to delete "cereals and candy".

Undo the delete

12. Since you still want to include the word "cereal", press **u** to undo your last delete.

Undo undo

13. Just for fun, press **u** again. What happens? The **u** command undid your last undo and deleted the line
again.

14. Press **u** again to make the line reappear.

Delete the end of line (D)

15. Now, get rid oft he words „and candy". Press **w** to move the cursor tot he word „and", then press SHIFT-D. The SHIFT-D (upper case D) combination deletes text to the end of the line.

```
meat
fruits and vegetables
cereals
dairy products
```

Delete, step I5

Save and quitt he file

16. Since you're in command mode (delete commands file only work in command mode), type **:wq** and press RETURN to save and quit the groups file.

REVIEW

Here's a list of commands you learned in this section.

Command	Description
x	delete character
dw	delete word
dd	delete a line
D	delete to end of line
u	undo the last command

Remember that all these commands work only in command mode. If you're in insert mode, press ESC to end insert, then use the command.

Edit a file

Create a new file
In previous exercises, you corrected mistakes by deleting the mistake and retyping; this isn't a very efficient way of changing text. The next exercise shows you how to use various **vi** commands to move around in a file and make corrections.

Press i for text insertion
1. Let's create the world file again. Type **vi world** and press RETURN. Because you did not save it the last time, **vi** creates a new file.

2. Press **i** for insert mode and type the lines shown below.

```
French
Mexican
Italian
German
These are the foods of the world.
```

Move to beginning of line (0)
3. Let's change the last sentence since it doesn't match the other lines. Press ESC to exit insert mode and press **0** (zero) to move the cursor to the beginning of the line (or move the cursor with the arrow keys).

Move by word (w)
4. Now, press **w** twice to move the cursor to "the". The **w** command moves the cursor forward one word.

Change a word (cw)
5. Type **cw**. Notice that this command replaces the last character of the word "the" with a $ (dollar sign). You are back in insert mode and can type text again.

```
French
Mexican
Italian
German
These are th$ foods of the world.
```

Change word, step 5

6. Type **types of**. Your sentence now reads: "These are types of foods of the world."

Change a line (cc)
7. Let's change the line again. Press ESC, then type **cc**.

8. The **cc** command deletes the current line and puts you back in insert mode.

9. Type **I like all kinds of food**.

Change a line (C)
10 .This line still doesn't fit with the rest of the file; let's change it one more time. Press ESC to quit insert mode, press **0** (zero) to move to the beginning of the line, then press **C** (SHIFT-C) to change the line.

Although **C** changes a line, it 's different from **cc** because the original line doesn't disappear. Like the **cw** command, a $ appears at the end of the text to be changed, and you can type over the line to change it.

11. Type Indian and press RETURN to make the remainder oft he line disappear

Save and quit the file (:wq)

12. Let's save and quit this file. Press ESC, type **:wq** and press RETURN. The shell prompt reappears.

```
French
Mexican
Italian
German
Indian

:wq

"world' [New file] 6 lines, 38 characters
$
```

Edit a file, steps 10-12

REVIEW

Here's a list of the commands you learned in this section.

Command	Description
0 (zero)	move to beginning of line
w	move ahead onc word
cw	change a word
cc	erase a line and insert
C	change to the end of a line

Copy and move text

Cut, copy and paste
You can use the **vi** editor for complex text editing tasks. The following exercise shows you how to rearrange text by copying, cutting, and pasting lines
in your file.

Open snacks
1. First, open your snacks file. (Type **vi snacks** and press RETURN.)

```
cake
chocolate chip cookies
fruit
potato chips
pie
tarts
```

2. Let's make some changes to the snacks file. Move the cursor down to the beginning of "pie" and press **i** to start insert mode.

Add apples
3. Type **apple** to make "apple pie", then press ESC to quit insert mode.

4. Move the cursor to the beginning of the next line (tarts) and press **i**. Type **apple** to make "apple tarts". Press ESC to quit insert mode.

```
cake
chocolate chip cookies
fruit
potato chips
apple pie
apple tarts
```

Change lines, steps 2-4

Cut lines (*n*dd)
5. The list was in alphabetical order, but it isn't any more. Move the two apple lines to the beginning of the file. Move the cursor to the beginning of the apple pie line, and type **2dd** to cut (delete) two lines.

Paste above cursor (P)
6. The two apple lines disappear and are placed in a temporary buffer. Move the cursor to the beginning of the file and press **P** (SHIFT-P) to paste the lines above cake.

```
apple pie
apple tarts
```

P to paste lines above

```
cake
chocolate chip cookies
fruit
potato
```

*n*dd to cut lines

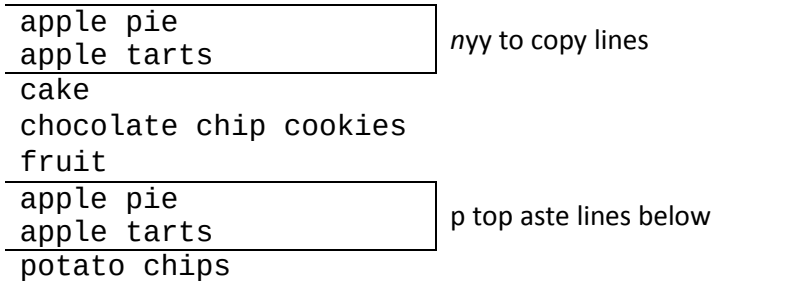
Cut and paste lines, steps 5 and 6

Copy lines (nyy)

7. Let's add two more lines to the file by copying the apple pie and apple tarts lines, pasting them after fruit, and changing apple to lemon. First, type **2yy** to copy the lines into the buffer.

Paste below cursor (p)

8. Move the cursor to the fruit line, then press **p** to paste the buffer below the cursor.



Copy and paste lines, steps 7 and 8

Change apple to lemon

9. Use the **cw** command to change apple to lemon on both lines.

```
apple pie
apple tarts
cake
chocolate chip cookies
fruit
lemon pie
lemon tarts
potato chips
```

Change word step 9

Save and quit the file (:wq)

10. Press ESC to quit insert mode, then type **:wq** to save and quit the file.

REVIEW

Here's a list of commands you learned in this section.

Command	Description
dd	cut (delete) a line
*n*dd	cut (delete) *n* lines
yy	copy (yank) a line
*n*yy	copy (yank) *n* lines
p	paste text below the cursor line
P	paste text above the cursor line

Search and repeat

These exercises show you how to search for words and repeat commands to quickly edit your files.

1. First, open the snacks file (**vi snacks**).

```
apple pie
apple tarts
cake
chocolate chip cookies
fruit
lemon pie
lemon tarts
potato chips
```

Use / to search
2. Press **/** (slash). The / appears at the bottom of your screen; type the text you want to find next to it.

3. Type **lemon** and press RETURN. The cursor moves to the first occurrence of lemon.

4. Type **cw** (change word) and type peach.

Use n to search again
5. Press ESC to quit insert mode and press **n** to seek the next occurrence of lemon.

Use dot command to repeat the last change
6. To repeat the last change easily, press the period key (**.**). The "dot" command repeats the last change. If your file was longer, you could continue to repeat the search and replace words until you ran out of lemon.

```
apple pie
apple tarts
cake
chocolate chip cookies
fruit
peach pie  1
peach tarts
potato chips
```

1
- **/lemon** and press RETURN to find lemon
- **cw** to change lemon to peach
- **n** to seek next lemon
- **.** (dot) to repeat change

Search and repeat change, steps 2-6

Save and quit (:wq)
7. Save and quit the snacks file (**:wq**)

You've just learned a complex process in a short amount of time, so let's do another search and repeat exercise.

1. Open the snacks file again (**vi snacks**).

Search (/)
2. Let's get rid of the word "peach" in the file. First, press **/** to begin the search.

3. Type **peach** next to the slash and press RETURN. The cursor moves to the first occurrence of peach.

Delete word
4. Type **dw** to delete the Word "peach".

Look for next (n)
5. Press **n** to look for the next peach.

Repeat (.)
6. Press **.** to delete peach again.

```
apple pie
apple tarts
cake
chocolate chip cookies
fruit
pie   1
tarts
potato chips
```

1
- **/peach** ond press RETURN to find peach
- **dw** to delete peach
- **n** to seek next peach
- **.** (dot) to repeat the deletion

Another search and repeat change, steps 1-6

Delete potato chips
7. The list is no longer in alphabetical order, so let‘s get rid of the „potato chips“. Move the cursor to the last line and type **dd**.

Save and quitt he file (:wq)
8. Type **:wq** to save and quit the file.

REVIEW
Here’s a list of the commands you learned in this section.

Command	**Description**
/*term*	search for *term*
.	repeat last change
n	repeat search (find next *term*)

Chapter review

Here's a list of the **vi** commands you learned in this chapter:

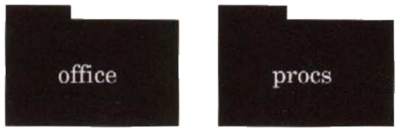
Command	Description
k	move cursor up
j	move cursor down
h	move cursor to the left
l	move cursor to the right
w	move cursor ahead one word
0 (zero)	move cursor to beginning of line
i	start insert mode after cursor
a	start insert mode before cursor
o	start insert mode with a blank line
:w	save a file
:q	quit a file
:wq	save and quit a file
:q!	quit without saving
x	delete a character
dw	delete a word
dd	delete a line
D	delete to end of a line
u	undo the last command
cw	change a Word
cc	erase a line and insert
C	change to the end of a line
dd	cut (delete) a line
*n*dd	cut (delete) *n* lines
yy	copy (yank) a line
*n*yy	copy (yank) *n* lines
p	paste below cursor
P	paste above cursor
/*term*	Search for *term*
. (dot)	repeat last change
n	repeat search (find next *term*)

Using UNIX file commands

File commands

cp	copy file
mv	move m rename a file
cp	remove a file

cp office procs



mv jobA job1

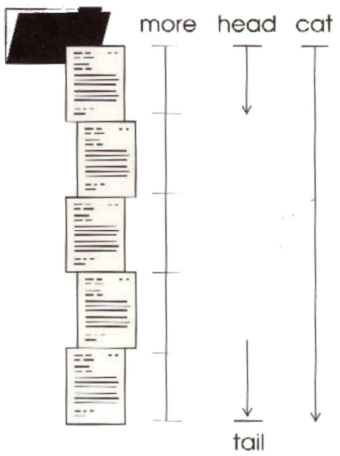
rm zoo

Print commands

lp	print a file
pr	format a file
lpstat -o	check print queue
cancel	cancel print request

File display commands

cat	display a file
more	display a file one screen at a time
head	display the beginning of a file
tail	display the end of a file

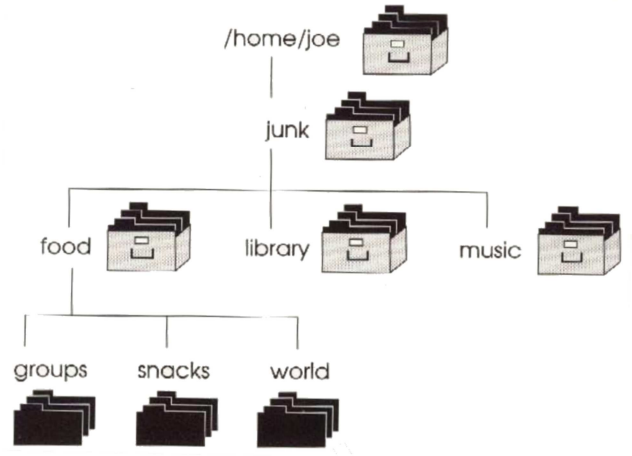


What's in this chapter?
You create and edit text with a text editor. You use UNIX commands to work with files in other ways, including:

- displaying a file
- printing a file
- copying a file to a new name or a new place
- renaming a file or directory
- moving a file to a new place
- deleting one or more files

Before you begin
In the *Working with directories* chapter, you learned about directory and file structures and about moving among directories. You learned that you can use pathnames to refer to directories (/home/joe/stuff) or you can use shortcuts (for example, .. indicates the parent directory).

Current directory structure
You should still have the directory structure you created in the previous exercises.



Your current file and directory structure

Wildcard characters
You can also use wildcard characters to represent several file names. For example, if you want to use a command with each of the files in the food directory, you simply type the command with an * (Asterisk) instead of a filename (food/*).
You can use * to replace a character or series of characters. For example, if you wanted to read all the files beginning with "p" in the office directory, you'd type **more office/p***.

For a complete list of wildcard characters, refer to *Using Amiga UNIX*.

Display a file

In an earlier exercise, you used the **ls** command to look at the contents of a directory. You use different commands to look at files. You can open a file through the **vi** program or you can view a file with UNIX commands such as **cat**, **more**, **head**, and **tail**. These commands enable you to see, but not edit, a file.

View a file (cat)
1. Let's look at a file with the cat command. Type **cat snacks** and press RETURN. The file displays on your screen and the shell prompt returns.

2. Now let's look at another file. Type **cat /etc/termcap** and press RETURN.

Notice how quickly the file scrolls to the end. You have no time to read information because it moves too fast.

View one page at a time (more)
3. Sometimes you need to display a file slowly so you can have time to read it. Type **more /etc/termcap** and press RETURN. The **more** command displays one screen of a file at a time.

NOTE: Don't worry about trying to understand the contents of the /etc/termcap file. We're using it only for display purposes.

4. You can move forward or backward in the file by using the keys listed below. Try to use them to view the /etc/termcap file.

more command keys

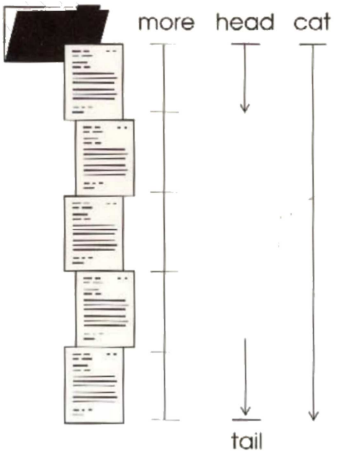
Key	Description
RETURN	display next line
SPACE	display next screen
b	skip backwards one screen
h	display help for the more command
q	quit

View the beginning of a file (head)
5. Sometimes you only need to look at the beginning of a file. In this case, use the head command. Practice using this command now; type **head /etc/termcap** and press RETURN. Only the first ten lines of the file display before your shell prompt returns.

View the end of a file (tail)
6. Now type **tail /etc/termcap** and press RETURN. The tail command displays only the last ten lines of a file.

7. You can also specify how many lines the **head** or **tail** commands should display. For example, type **tail -15 /etc/termcap** and press RETURN to display the last 15 lines of the file.

Display commands



Display a file with UNIX commands

Print a file

Now that you've created files through the **vi** editor, you can print them on a printer. Before you perform this exercise, make sure a printer has been set up for your system.

Print a file (lp)
1. Let's print the snacks file. Type **lp snacks** and press RETURN. The **lp** command sends the print request to the printer.

File permission
NOTE: In order to print a file, other users must have permission to execute the file. UNIX interprets **lp** as an "other user". If you own the file, you can add this permission by typing **chmod o+x *filename***.

Format and print a file (pr *file* I lp)
2. Request another printed copy of snacks; however, before you print the file, format it with headers and margins. Type **pr snacks I lp** and press RETURN. The **pr** command formats files.

Check the print queue
3. Most likely, you share your printer with other users. If this is the case, you may have to wait a while before your file prints. Your print requests wait in a list of similar jobs called a queue. Find where your requests are in the queue by typing **lpstat -o** and pressing RETURN. Look for your name and the ID assigned to your job.

```
$ lpstat –o
1                  2      3     4             5
Postscript-1165       joe   122   May 24 09:30    on pr1
Postscript-1167       joe   122   May 24 09:35
Postscript-1168       pat   435   May 24 09:45
```

Check print queue, step 3

1: ID
2: user
3: file size
4: date and time submitted
5: printer

Cancel a print request
4. You don't need two copies of the snacks file, so request cancel the first request. Type **cancel *ID***).

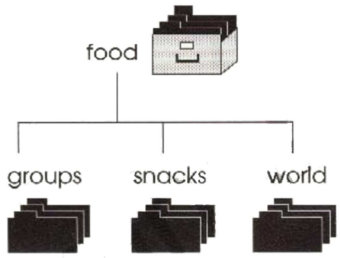
Unless you're logged in as root, you can only cancel your own requests.

Copy a file

Sometimes you need to duplicate your files, perhaps because someone wants a copy of your file to work with and you don't want to change the current file.
This exercise shows you how to copy files.

Let's look what you've created so for
1. Before we begin, lot's look at the files you've created so far. First, type **pwd** to make sure you're still in the food directory. If you're not, type **cd junk/food** from the home directory.

2. Type **ls** to list all the files in your food directory. A picture of your food directory looks like this.



Copy a file (cp)
3. Let's make a copy of the snacks file. At the shell prompt, type **cp snacks newsnacks** and press RETURN.

The **cp *file copyfile*** command makes a copy of a file.

Let's look at the food directory
4. Your copy is now in the same directory. Type **ls** and press RETURN; newsnacks should be listed.

```
$ pwd                                  check current directory
/home/joe/junk/food
$ ls                                   list directory contents
groups snacks world
$ cp snacks newsnacks                  copy file
$ ls
groups newsnacks snacks world          copy is listed in the directory
$
```

Copy a file, steps 1-4

Look at the contents of a file (cat)
5. Let's look at the contents of your newsnacks file. Type **cat newsnacks** and press RETURN. You see the newsnacks file contains the same lines as the snacks file.

Rename or move a file

Sometimes you need to rename files because names are so similar they are confusing or, perhaps, because you decide to standardize your file names. This exercise shows you how to rename and move files and directories so you can maintain and list your directories effectively.

Rename a file (mv)

1. The name "newsnacks" is too much like the name "snacks". Let's change newsnacks to goodthings. You should be in the food directory. If you're not, move there now (type **cd junk/food** from the home directory). At the prompt, type **mv newsnacks goodthings** and press RETURN. This changes the newsnacks file into the goodthings file.

2. Type **ls** to list the food directory and make sure goodthings is listed.

```
$ mv newsnacks goodthings          change a file's name
$ ls
goodthings groups snacks world
$
```

Rename a file, steps 1 and 2

Move a file into another directory (mv)

3. You can also use the **mv** (move) command to move files into other directories. Let's move the groups file (in the food directory) into the music directory.
Since you're in the food directory, type **cd ..** to move up one directory level to the junk directory.

List the contents of the junk directory

Type **pwd** to make sure you're in the junk directory and **ls** to verify that your current directory contains the food and music directories.

```
$ cd ..                            move up one directory level
$ pwd                              list the current directory
/home/joe/junk
$ ls                               list the junk directory's contents
food library music
$
```

Move a file, steps 3 and 4

Change to food directory

5. Type **cd food** and press RETURN to make food the current directory.

6. Type **ls** to list the contents of the food directory. It should include goodthings, groups, snacks, and world.

Move a file

7. Type **mv groups ../music/albums** and press RETURN.

The **mv *file directory*** command moves a file from one directory to another, using the same filename. The **mv *file directory/file*** command moves a file to another directory and gives it a new name.

List the contents of the food directory

8. Now, type **ls** and press RETURN to look for the contents of the food directory. Do you see groups'? No, you don't see the groups file because you moved it into the music directory under the name "albums".

Change to the music directory
9. Type cd .. /music and press RETURN to make music your current directory.

List the contents of the music directory
10. Look at the contents of the music directory; type **ls** and press RETURN. The albums file is in your music directory.

```
$ cd food                          make food the current directory
$ ls                               list the contents of food
goodthings groups snacks world
$ mv groups ../music/albums        move the groups file into the music directory and call it albums
$ ls                               list the contents of food again
goodthings snacks world
$ cd ../music                      make music the current directory
$ ls                               list the contents of music
albums
$
```

Move a file, steps 5-10

Look at the albums file
11. Look at the contents of the albums file; type **cat albums** and press RETURN.

Move the file back into the food directory
12. Meat, fruits and vegetables, cereals, and dairy products aren't music albums, so let's move the file back into the food directory and change its name back to groups.
Type **mv albums ../food/groups** and press RETURN.

List the contents of the food directory
13. Check the contents of food by making it the current directory (**cd ../food**) and listing the contents (**ls**).

```
$ cat albums                       display the albums file
meat
fruits and vegetables
cereals
dairy products
$ mv albums ../food/groups         move the albums file into food and call it groups
$ cd ../food                       make food the current directory
$ ls                               list the contents of food
goodthings groups snacks world
$
```

Move a file, steps 11- 13

Delete a file

Sometimes you need to remove unwanted files and directories that are taking up disk space. Be very careful when using remove commands; you can't undo a remove command or recover a file after it has been deleted.

Remove a file (rm –i)
1. Snacks and goodthings have the same contents. You don't need two copies of the same file, so let's get rid of goodthings. Make sure you're in the food directory (**pwd**), then type **rm -i goodthings** and press RETURN. What happens?

Why use the –I option?
rm (remove) is the delete command. The **–I** (interactive) option asks if you are sure you want to remove the file.

```
$ pwd
/home/joe/junk/food
$ rm –I goodthings
rm: remove goodthings: (y/n)?
```

Delete a file, step 1

Because any file you remove cannot he brought back (unless you have a backup), using the **rm** command alone can be dangerous. You should always type **rm -i** so you can make sure you"re deleting the correct file.

2. Type **y** in answer to the query and press RETURN.
Check the food directory to make sure the goodthings file is gone (**ls**).

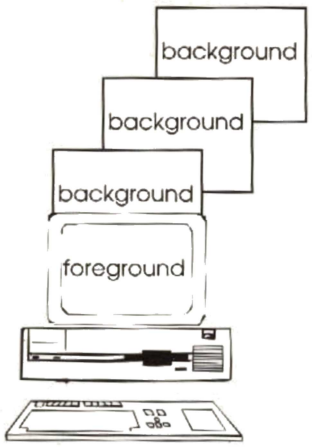
Chapter review

Here's a list of the commands you learned in this chapter:

Command	Description
cat	display a file
more	display a file one screen at a time
head	display the beginning of a file
tail	display the end of a file
lp	print a file
pr	format a file
lpstat –o	check the print queue
cancel	cancel a print job
cp	copy a file
mv	move or rename a file
rm -i	remove a file

Working with processes

Foreground and background processes



You can continue to work on a screen while a background process runs. The shell prompt remains on the screen. Type **&** at the end of a command line to put a process in the background.

A process automatically runs in the foreground, unless you put it in the background. The shell prompt disappears while a foreground process runs.

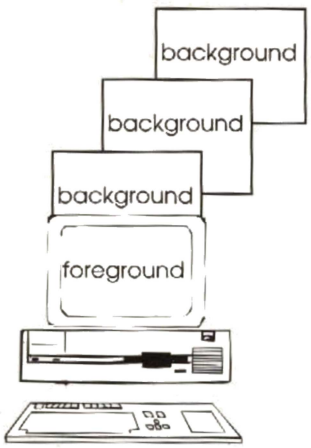
Cancel a process
To cancel a foreground process, press CTRL-C.

To cancel a background process,

1. find out the process ID (PID) by typing **ps -a**
2. cancel the process by typing **kill *PID***

What's in this chapter?
In these exercises, you'll learn how to list, cancel and kill foreground and background processes. These functions are useful when you need to know what programs you (or other users) are running and when you want to stop a running process.

Before you begin
These exercises are designed to show you the difference between foreground and background processes. The basic difference is that the shell prompt remains on the screen during background processes (although the process typically disappears), allowing you to continue working on the screen, and the prompt disappears during foreground processes. Rather than use background processes, you may choose to use another virtual screen instead.



Foreground and background processes

Create and cancel a foreground process

Create a foreground loop program
1. First, let's create a temporary program that loops continuously. At the shell prompt, type **while sleep 5; do echo "I'm ba-ack"; done** and press RETURN.

```
$ while sleep 5; do echo "I'm ba-ack"; done
I'm ba-ack"
```

Create a foreground process, step 1

This program displays "I'm ba-ack" on your screen every five seconds. Wait for the first occurrence. Wait a few more seconds; it happens again.

Notice that you do not get a shell prompt while this program is running. The program is running in the "foreground", which means you can't do any other work on this screen while the program is working.

Cancel a foreground process (CTRL-C)
3. Cancel this process by pressing CTRL-C. The shell prompt appears again.

Create and cancel a background process

You've just cancelled a foreground process. Processes can also run in the background; these processes may be hidden from you but are still active. Unlike foreground processes, you can use the same screen for other functions while background processes are running; you don't lose the shell prompt.

Since UNIX is a multi-tasking system, you can have many tasks running at the same time, but only one process runs in the foreground. Cancelling a foreground process is easy; you simply press CTRL-C. However, cancelling a background process is more complex. Since your screen is free, CTRL-C won't work. You have to list your processes, find the one you want, then issue a command to stop it.

Create a background loop program
1. Let's create another program and put it into the background.
At the shell prompt, type **while sleep 25; do echo "I'm ba-ack"; done&** and press RETURN. (The ampersand (&) moves the program
to the background.)

```
$ while sleep 5; do echo "I'm ba-ack"; done
[1] 405
```

Create a background process, step 1

2. Every twenty-five seconds "I'm ba-ack" appears.
Wait for the first occurrence.

View your active processes (ps)
3. Type **ps -f** and press RETURN to view a list of all running processes. (If you wanted to see the active processes on the system, you'd type **ps -e**.)

4. **ps -f** displays all your running processes. Many processes are running in your name: your login shell (**ksh**, **sh**, or **csh**), the **ps -f** command (which has since finished, but wass running a moment ago), and the **sleep** command you put in the background.

Look for PID
Look for the number near the beginning of the sleep line. This is the process ID (*PID*) you need to use in the next step.

```
$ ps –f
PID   TT    STAT  TIME       COMD
363   p2    R     08:42:48   sleep 25
$
```

List processes, steps 3 and 4

Kill a background process
5. Type **kill *PID*** (using the PID from the sleep process) and press RETURN to cancel the process.
If you successiully killed the process, the **kill** command responds with the message “terminated”.

```
$ kill 363
$ Terminated     Confirmation that process has been killed
$
```

Kill a background process, step 5

If the **kill** command fails, the **sleep** PID may have changed. Type **ps -f** to obtain the new PID, then type **kill *PID***, using the new PID number, to kill the **sleep** program. The number changes because, after each time the message is displayed, a new **sleep** process starts and the old one ends.

Chapter review

Here’s a list of important points you learned in this chapter:

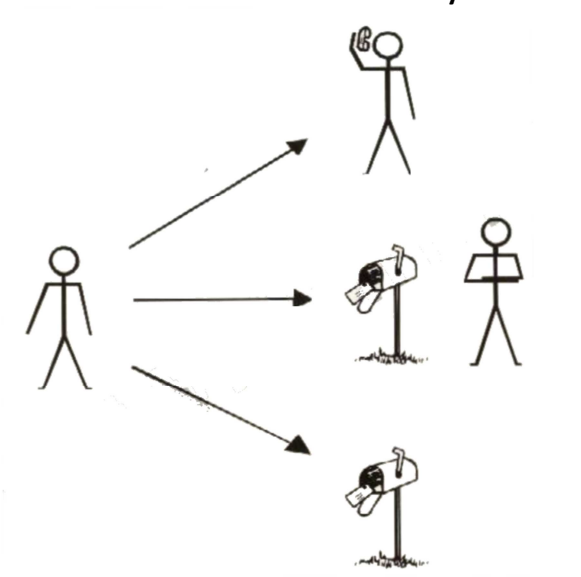
- A foreground process is visible on your screen. While it’s running, you cannot perform any other work on that screen.
- Press CTRL-C to cancel a foreground process.
- A background process ends with an & and gives you back your shell prompt. You can run as many background processes as you like and still be able to work on your screen.
- Use the **kill** command to cancel a background process.
- Find out the process ID number of a program with the **ps** (or **ps –f**) command.

Communicating with other users

Commands

who	list all active users on a system
who am i	list the identity of the current screen and user
finger	list active local and remote users
talk	send messages to active users on screen
elm	read and send mail

Ways to communicate with users



talk to activeusers

send mail (**elm**) to user

mail waits for user

Whats in this chapter?
If you share your system with co-workers, or if your system is part of a network, you can communicate with other users. This chapter shows you how to:

- find other users
- "talk" with them
- send mail to them

Before you begin
Before you begin these exercises, ask a friend to work with you. You'll need to know the friend's username (*username*) and the system name (*systemname*).

If you cannot find someone to work with, you can use the guest account on your system.
When you installed Amiga UNIX, the system automatically created a user account for guest. Find out the password, if there is one, for this account.

You can communicate with users on the same network

Find active users

You need to know who else is on the system if you want to talk to them. This exercise teaches you several commands you can use to find out about active users.

1. First, log in to several virtual screens. Log in to F3, F4, and F5. Remember, to change virtual screens press ALT-*functionkey*.

List all active users (who)

Now, let's find out who is currently on the system. Type **who** and press RETURN to produce a list of users on the system.

The list tells the username, virtual screen and the login date and time for each user. You are included on the user list; in fact, if no one else is using your system, you may be the only user listed.

```
$ who
joe   console    Jul   10    09:30
joe   term/con1  Jul   10    11:30
joe   term/con3  Jul   10    09:45
joe   term/con5  Jul   10    10:40
$
```

List active users, step 1

List your screen identification (who am i)

3. You've just logged in to several virtual screens. To find out which one you're currently using, type **who am i**. This command displays information about your current login session.

```
$ who am i
joe   term/con3  Jul   10    09:45
$
```

List user information, step 3

List active users (finger)

4. Another way you can find out who is logged in to the system is by typing **finger**. The **finger** command displays the username, the full name, the screen number (TTY), the amount of time the system has been idle, the login time, and the location (if known) for each user. Type finger and press RETURN to view this type of user list.

```
$ finger
```

```
Login Name       TTY        Idle  When       Where
joe   Joe Smith  term/con1  30    Fri 07:59
joe   Joe Smith  term/con2  9     Fri 08:30
joe   Joe Smith  term/con3  1d    Thu 09:45
$
```

List active users, step 4

Notice that you're listed at least twice. That's because you're working on more than one screen.

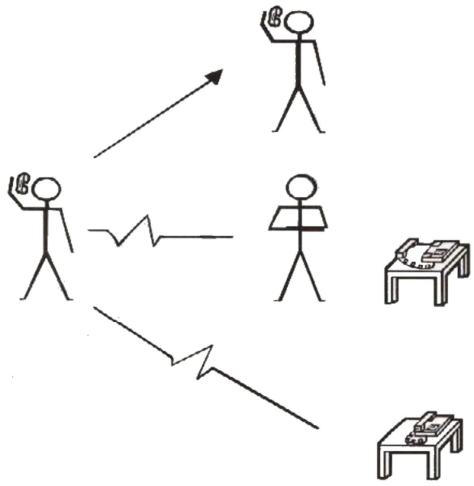
Talk to another user

"Talking" to other users

The **talk** command acts like a terminal telephone system. You "call" other users on their terrninals by typing **talk *username***. If they're on a different system, type ***username@systemname***.

List active users with who

You can only **talk** to "active" users (users who are logged in to a system) who want to talk to you (**mesg**=y). Use the **who** command to find out if the person you want to talk to is logged in to the system. The **who** command only lists active users, not all users who can use the system. You might be the only user listed if you are the only person logged in to the system.



You can only talk to active users who want to talk back

Log in as guest on another screen

1. To give yourself someone to talk to, change to another virtual screen (ALT-F7) and log in as guest. Each system includes an account called "guest".

Return to your screen; type who

2. Go back to your original working screen (ALT-F2). Type **who** and press RETURN to list all the active users on your system. Your *username* and **guest** should appear on the list.

Begin talk program

Type **talk guest** and press RETURN to begin the **talk** function. Your screen displays the following message:

```
[Waiting for your party to respond]
```

Message from talk program

Change to guest screen
You can't **talk** to guest until guest responds to your **talk** request. Change to the screen where you logged in as guest (ALT-F7). You see the following message:

```
Message from Talk_Daemon@utopia at 11:25
talk: connection requested by joe@utopia
talk: respond with talk joe@utopia
$
$ talk joe@utopia
```

Talk, steps 4 and 5

Guest responds to talk request
5. Type **talk**, followed by your *username*, and press RETURN.

6. The guest screen divides into two parts: the top part where you type and the bottom part where you read replies. Type **I'm sending a message as guest**.

```
I'm sending a message as guest.
```

You type your messages here
You read messages from other users her

Talk, step 6

Send guest a message
7. Go back to your original Working screen (ALT-F2). You see guest's message on the bottom half of your **talk** screen. The cursor remains there until you begin to type.

Read the message
8. Type a message to guest: **Hi guest! Here's my response.** Then go to the guest screen and read it (ALT-F7).

Quit talk
9. Exit the talk function by pressing CTRL-C.

Log out as guest
10. Log out of the guest screen (CTRL-D), and return to your original working screen (ALT-F2). A message at the top of your screen tells you the connection has been broken and your shell prompt returns.

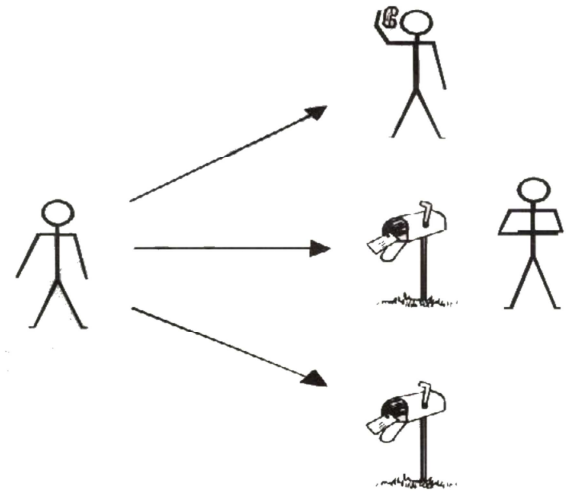
Refresh your screen
11. At the shell prompt, type **clear** to refresh your screen.

Use electronic mail (elm)

Comparing talk to electronic mail
Although the talk command is a good way to communicate with other users, it has a few drawbacks:

- you can only communicate with users who are logged in (and have **mesg**=y)
- you may interrupt the work of the person you talk to

You can send mail to active and inactive users
Another way to send information is through the electronic mail system. You can send mail at any time without interrupting other users' work. The mail waits in their mailboxes until they read it, just like when you send a letter through the post office.



You can talk only to active users, but you can send mail to active and inactive users

Types of electronic mail
Your Amiga UNIX system includes two types of electronic mail: **mail** and **elm**.

Help for mail
We'll be using **elm** in the tutorial, but you can use **mail** later if you like. If you have questions about how to use it, read the *Using electronic mail* chapter in the *Using Amiga UNIX* book.

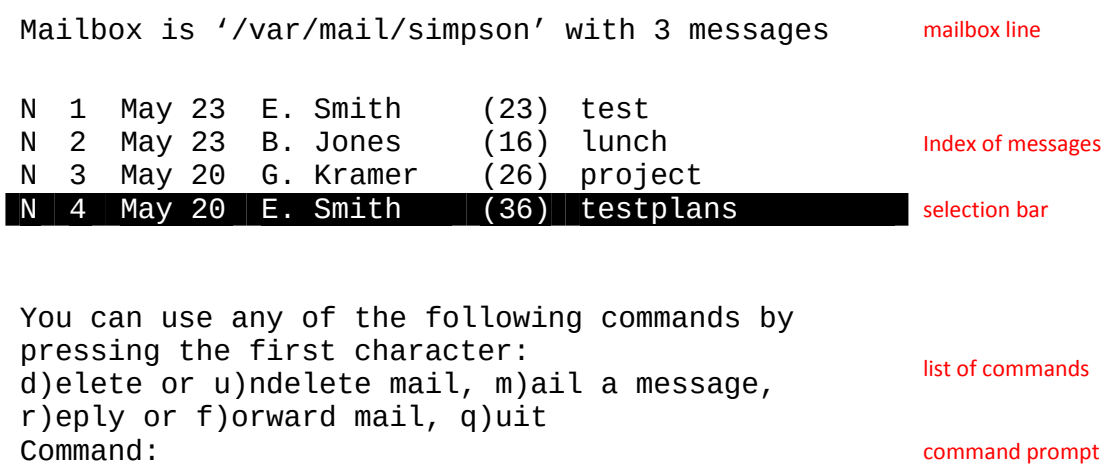
Access elm
1. At the shell prompt, type **elm** and press RETURN.

Preparing your system for elm
2. If you have not started **elm** before, the system sends you some notices about directories it needs to create. Press RETURN (for yes) after each question.

The **elm** mailbox screen appears. If you have any mail messages, they are listed in the top portion of the screen. The first message is highlighted with a reverse-video bar; this "bar" can he moved by pressing **j** (down) and **k** (up).

Some of the commands available through the **elm** mailbox screen are listed at the bottom of the screen. Refer to the *Using electronic mail (elm)* chapter in *Using Amiga UNIX* or information about additional commands.

elm mailbox screen



Sample of the elm mailbox screen

Send a message to yourself

4. For the first exercise, you can send a message to yourself. Press **m** for mail.

5. **elm** asks for the *username* of the person to whom you want to send mail. Type your *username* and press RETURN.

The system displays your full name as it is stored in the passwd file.

6. **elm** asks for the subject of your message. Enter **first test** and press RETURN.

7. **elm** prompts "Copies to:". Since you don't want to send any copies of this message, press RETURN.

When you want to send copies, type other *usernames* here.

```
Command; mail
Send message to: joe To: Joe Smith
Subject of message: first test
Copies to:
Invoking editor …
```

Using elm, steps 6 and 7

Use the editor (vi) to write your message

8. **elm** tells you it is invoking the **vi** editor and displays a blank file for you to edit.

vi uses a temporary name created by **elm**, the temporary file is automatically deleted when **elm** sends the message.

Enter the following message. (Remember to press **i** to enter insert mode.) If you make a mistake, use **vi** commands to correct it.

This is my first test. I'm just practicing.

```
This is my first test. I'm just practicing.
~
~
~
~
~
~
"/tmp/snd.380" 0 lines, 0
characters
```
temporary filename

Send a message, step 8

Store the message

9. Store the message by pressing ESC to quit insert message mode, typing **:wq**, and pressing RETURN.

10. **elm** asks you to choose an option. These options are slightly different from the ones on the **elm** mailbox screen.

```
Please choose one of the following options by parenthesized letter:
s
e)edit message, edit h)eaders, s)end it or f)orget it.
```
s (s (send) is the default)

Send a message, step 10

Send the message

Since **s** is the default value, press RETURN. **elm** sends your mail and brings you back to the **elm** mailbox screen.

Press CTRL-L to refresh the index

11. Don't be surprised if you don't see the new message listed in the index of messages; it may take a moment or two to reach your mailbox. Press CTRL-L to refresh the screen and update the index.

```
Mailbox is '/var/mail/smith' with 1 messages

N  4  Aug 14  Joe Smith    (72)  first test

You can use any of the following commands by
pressing the first character:
d)elete or u)ndelete mail, m)ail a message,
r)eply or f)orward mail, q)uit
Command:
```
new message status (N)

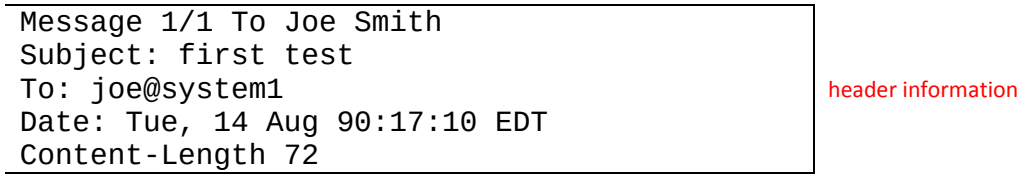
Read a message, steps 11 and 12

Read the message

12. The reverse video bar highlights your message line. Press RETURN to display the "first test" message.

Read the message

13. Read your message, then mark it for deletion by pressing **d** at the command prompt.



Read a message, step 13

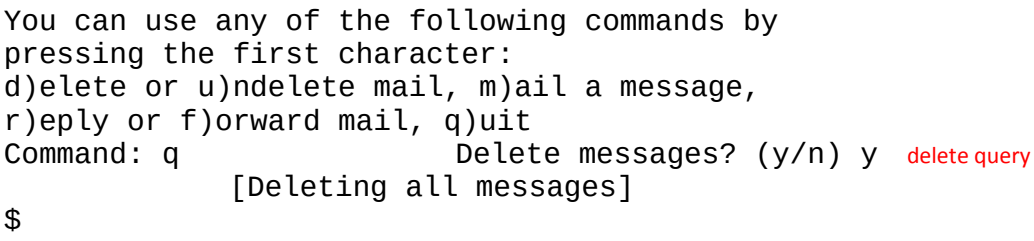
Message marked for deletion

The **elm** mailbox screen appears again and a "D" appears at the beginning of the first test message line. This indicates that you've marked the message for deletion; however, the message has not yet been deleted. Messages aren't deleted until you quit **elm**.

14. Press **q** to quit **elm**.

Delete query

15. **elm** asks "Delete messages?". Type **y** to delete any message with a delete status.



Quit elm steps 14-15

Mail another message

Let's send mail to another user. You can send mail to the guest account.

elm shortcut
1. At the shell prompt, type **elm guest** and press RETURN.

NOTE: You don't have to send messages through the **elm** mailbox screen; you can send them from your shell prompt with the **elm *username*** command.

2. Enter **second test** as the subject of the message and press RETURN.
3. **elm** displays the "Copies to:" line. Send a copy to yourself; type your *username* and press RETURN.

```
$ elm guest

To: System guest
Subject of message: second test
Copies to: joe
```

Send onother message, steps 1-3

Use an editor (vi) to write the message
4. When the **vi** screen appears, press **i** and type the following message:

Here's a message for guest. I'm also sending it to myself.

Send the message
5. Press ESC to quit insert mode, then type **:wq** and press RETURN to save the message and quit the editor. Press RETURN again to send it.

6. Now let's look at the message in guest's mail directory. Press ALT-F4 to reach another virtual screen.

Log in as guest
7. Log in as guest on this screen.

Note the mail notice
8. Note the login message "you have mail." This message appears whenever you log in and have mail. **elm** displays this message as long as you have active mail in your mail folder.

9. Let's look at the message (second test) you just sent to guest. Type **elm** and press RETURN. The **elm** mailbox screen displays and lists "second test" in the message index.

Read the message
10. If the reverse-video bar highlights the second test message, press RETURN. If second test isn't highlighted, move the bar (using the **j** and **k** keys) to select the message, and press RETURN to read the message.
11. After reading the message, press **i** to return to the index (mailbox screen).

Reply to the message
12. Press **r** to reply to the message.

13. **elm** asks "Copy message?". Type **y** to include the message as part of your reply.

Change the subject
14. Since you're replying to a message, **elm** automatically knows who to send the reply back to. However, let's change the subject; BACKSPACE over "second test" and type **Reply**. Press RETURN twice.

Write a reply
1 5. As you can see, your original message is repeated at the top of your reply. Use the arrow keys to move to a blank line, press **i**, and type the following message: **Guest is replying to my mail message.**

```
>Here's a message for my partner. I'm
>also sending it to myself.
Guest is replying to my mail message.
~
~
:wq
"/tmp/snd.380" 3 lines, 53 characters

Please choose one of the following options by parenthesized letter:
s
e)edit message, edit h)eaders, s)end it or f)orget it.
```

Send another message, step 15

Send the message
16. Save and quit the file (**:wq**), then send the message.

Delete the old message
17. The mailbox screen reappears. Delete the mail you sent to guest ("second test") by selecting the message, typing **d** at the command prompt and quitting **elm**. (Remember to answer yes to the delete inquiry.)

Log out guest
18. Log out of the virtual screen by pressing CTRL-D.

Read your mail
19. Now, let's look at the reply. Go back to your other screen (press ALT-F2) and read your mail through **elm**. (Press CTRL-L if the message lines don't appear on your mailbox screen.) You have two messages: a copy of "second test" and "Reply".

Delete the messages and quit elm
20. Delete the messages after you read them, then quit out of **elm**. Remember to answer yes to the "Delete messages?" query.

Chapter review

Here's a list of the commands you learned int this chapter:

Command	Description
who	list all active users on a system
who am i	list the indentity of the user and screen
finger	list active users
talk	write messages on your terminal to an active user
elm	send or read mail
elm *username*	send mail to a user

Changing your system

Commands you use to change your environment

stty	view terminal information and change command keys
color -bc *color* -fc *color*	change the screen colors
sioc setfont *fontfile*	change the screen font
echo *variable*	displays the contents of the shell variable ($SHELL, $HOME)

```
$ color -bc black -fc white
```

Working with aliases

alias	display current aliases
alias term="*command(s) and option(s)*"	create a command to replace a series of commands and options in the K shell; add this line to profile (in your home directory) to make the alias permanent
alias *term command(s) and option(s)*	create a command to replace a series of commands and options in the C shell; add this line to .login (in your home directory) to make the alias permanent

What's in this chapter?
This set of exercises shows you how to:

- view your terminal information
- change your command keys
- change your virtual screens
- customize commands

Before you begin
You don't need to know anything special to complete the exercises in this chapter. Simply follow the directions.

View terminal information

View terminal information (stty)

The **stty** command displays terminal information you might find useful. It shows the baud setting of your terminal, information about your monitor (rows, columns and pixels), and the command keys your system uses to erase lines, cancel commands, and suspend a screen.

1. Type **stty** and press RETURN to view information about your terminal. Pay particular attention to the settings highlighted below.

```
speed 38400 baud;
rows = 50; columns = 80; ypixels = 400; xpixels = 640
intr = ^c;         erase = ^h;     kill = ^u
```
interrupt (cancel command) erase (delete character) kill (erase line)

If a caret (^) appears next to another character, it indicates that you should hold down the CTRL key while typing that character. For example, to erase a line, you might hold down CTRL and press **u**.

However, special keys may automatically act as CTRL-*character* key combinations. For example, rather than pressing CTRL and **h** to erase a character, you can simply press the BACKSPACE key (<-). The BACKSPACE key acts like CTRL-H.

Discover your commands

2. Look for the commands listed in the table below; then write the key(s) used for that command in the key(s) colmnn. These are not all of the key commands listed by **stty**, but they are the ones you use most frequently.

Command	Description	Key(s)
erase	delete a character	
kill	erase the current command line	
intr	break (interrupt) a process	

Change command keys

The stty command also lets you change key commands. Let's try to change a key command.

Test the DEL key

1. First, try the current erase key. At the shell prompt, type **This is a test** and press the DEL key. Pressing the DEL key prints ^?; it doesn't delete the last "t" in test.

```
This is a test^?     DEL is not the erase key
```

Test the DEL key, step 1

2. Now, press BACKSPACE two times. (BACKSPACE is the default erase key.) The ^? and the last "t" in test disappear.

Change the erase key to DEL

3. Let's change your erase key to the DEL key. Press CTRL-U to erase this line and start over. Type **sty erase**, press SPACE, then press the DEL key (your screen displays ^? to indicate the DEL key). Press RETURN.

Check terminal information

4. Now check your terminal information by typing **stty** and pressing RETURN. Look for the word "erase". **DEL** is now the value for erase.

```
$ sty erase ^?
$ stty
.
.
intr    erase DEL;      kill ^u;
^c;
.       erase key is now DEL
.
$
```

Change the erase key; steps 3 and 4

Use DEL as the erase key

5. Let's try using DEL as the erase key. Type **This is another test** and press DEL. As you press DEL, the letters to the left of the cursor disappear.

6. Now, press the BACKSPACE key. Your screen displays ^H rather than deleting characters. You've successfully altered your erase key; BACKSPACE no longer works!

You are not confined to choosing only the BACKSPACE or DEL keys as your erase keys. You can select any key (except for the function keys) to act as your erase key.

Change the erase key to g

7. Let's change your erase key again. Press CTRL-U to clear the line and type **stty erase g**. Press RETURN again.

8. Now, type **This is a test** and press **g** several times. **g** deleted the last characters.

9. Press CTRL-U for another shell prompt and type **The boat is aground.** What happens?

The screen displays "The boat is round." The "g" did not print; pressing g deleted the “a”. As you can see, using a regular key as a special key isn’t a good idea.

Change your erase key back to BACKSPACE key

10. Switch your erase key back to the BACKSPACE key. First, press CTRL-U for another shell prompt. Then type **stty erase**, SPACE, then press the BACKSPACE key. Press RETURN.

11 . Type **stty** and make sure the value for erase is ^h (BACKSPACE).

Change screen colors

Refer for *Using AMIGA UNIX* for more details
In this exercise, you'll learn how to easily change the color of your virtual screens. You can change your virtual screen settings by editing the /etc/inittab file; however, the procedure is a little complex and you must be logged in as root. Refer to the *Special features of Amiga UNIX* chapter in the *Using Amiga UNIX* manual for instructions on editing inittab and for more details about the commands in this exercise.

You can use advanced color options to select a variety of colors. Refer to the **color man** pages for more information.

Monochrome monitors
If you have a monochrome monitor, your screen displays grey shades instead ofcolors like red and blue.

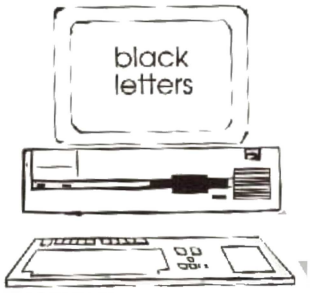
Change virtual screens
1. Let's change to another virtual screen. Press ALT-F5. The fifth virtual screen has a default setting of a black background with white letters in the foreground.

Default colors for ALT-F5

Change colors

2. Let's reverse the colors. Log in to the system on F5, then type **color -bc white -fc black** and press RETURN. The **-bc** option is the background color; the **-fc** option is the foreground (letters) color.

```
$ color -bc white -fc black
$
```



Change your virtual screen colors

Change colors again
3. Let's try it again. This time, type **color -bc black -fc white** and press RETURN. This command returns your screen to the original setting.

Create aliases

Create your own commands
An alias is a command you create to:

- change the name of a command
- use as a shortcut instead of typing an entire command and its options
- replace a series of commands you use frequently with a single command

The exercises below show you how to display your current aliases and how to create an alias for displaying an entire year's calendar with a single command.

The format you use to create an alias depends upon which of these shells you have. If your shell is /bin/sh, the Bourne shell, you can't use aliases.

Look at current aliases
1. First, type **alias** and press RETURN to view your current aliases. These aliases were installed automatically with Amiga UNIX; do not alter them.

What shell are you in?
2. Now, find out what shell you're in. Type **echo $SHELL** and press RETURN. The **echo** command displays the contents of the SHELL system variable.

echo $SHELL displays /bin/ksh for the Kern shell and /bin/csh for the C shell.

Create a temporary alias
3. At the shell prompt, type the appropriate **alias command**

Korn shell

`$ echo $SHELL`
`/bin/ksh`
`$ alias cyr "cal 1991"`

C shell

`% echo $SHELL`
`/bin/csh`
`% alias cyr cal 1991`
`%`

Create a temporary alias, steps 2 and 3

4. Now, type **cyr** and press RETURN. The alias displays the 1991 calendar.

Log in to another screen
5. Log in to another virtual screen (ALT-F6) and type **cyr**; press RETURN. What happens? The system tells you the **cyr** alias was not found.

NOTE: Aliases created from the command line only work during your current session.

Create permanent aliases
6. Let's make a permanent alias. To do this you need to edit a startup file in your home directory: .profile for the Kern shell or .login For the C shell. Type **vi .profile** (or **vi .login**) and press RETURN. Here's a sample of a .profile.

```
ENV=$HOME/.kshenv
export ENV
```

```
EDITOR=vi
export EDITOR
FCEDIT=vi
export FCEDIT
~
".profile" 6 lines 110 characters
```

Sample .profile

NOTE: Make sure you don't make any changes to this file other than the ones listed in the instructions. The profile and .login files are important files.

7. Use your arrow keys to move to the last line in the file.

8. Press **o** to add a line below the current line.

Add alias lines to a startup file

9. Type the appropriate lines into the startup file.

Korn shell
`alias status="who -u;date;pwd"` `alias cyr="cal 1990"`

C shell
`alias status (who -u;date;pwd)` `alias cyr (cal 1990)`

Creating permanent aliases, step 9

10. Press ESC to exit insert mode.

11. Type **:wq** and RETURN to save and quit the file.

Log out and in to use the updated startup file

12. Log out of your screen (CTRL-D). Press ALT-F6 and log in.

NOTE: You must log out and then log in to use the updated startup file.

Test the aliases

13. Type **status** and press RETURN to test your alias.

14. Type **cyr** and press RETURN to test this alias.

Chapter review

Here are some important points you learned in this section:

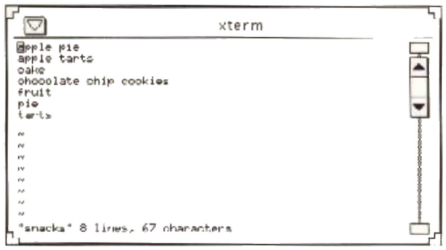
- You can use the **stty** command to view or change terminal information.
- You can change your screen colors with the **color -bc *color* -fc *color*** command.
- You can change the fonts on your screen with the **sioc setfont *fontfile*** command.
- You can use an alias to replace a series of commands. The alias can be temporary or permanent; to make it permanent you must add it to your .profile or .login startup file.

Using X windows

X Window System and OPEN LOOK

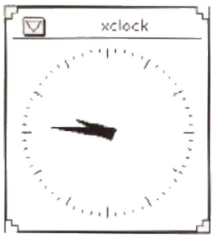
With OPEN LOOK and the X Window System, you can run many processes on a single screen

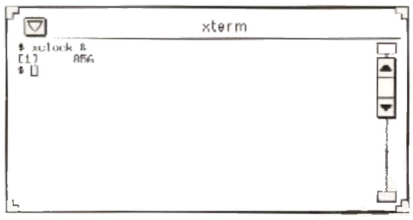
OPEN LOOK Workspace



vi

xclock





shell prompt

Steps to use X Window System and OPEN LOOK

Step	Procedure
1	The first time you use OPEN LOOK, type **/usr/X/adm/oladduser** at the shell prompt.
2	To start OPEN LOOK, type **olinit** at the shell prompt.
3	Select the programs option from the Workspace menu.
4	Select the terminal session (xterm) option from the programs pop-up window.
5	Enter commands into the xterm Window.

You can create many windows on a single screen
These exercises are designed to show you how to use the X Window System and OPEN LOOK with Amiga UNIX. These programs allow you to set up as many windows as you like on a single screen. Each window acts like a separate terminal screen; you can perform different functions in each window and switch among them with your mouse.

Set up OPEN LOOK
1. First, type **/usr/X/adm/oladduser *username*** at the shell prompt; press RETURN. This program automatically moves some OPEN LOOK startup files to your home directory.

```
$ /usr/X/adm/oladduser joe
$
```

Set up OPEN LOOK, step 1

Log out and in
2. Log out, then log in to use your changed startup file.

Run OPEN LOOK
At the shell prompt, type **olinit** and press RETURN.

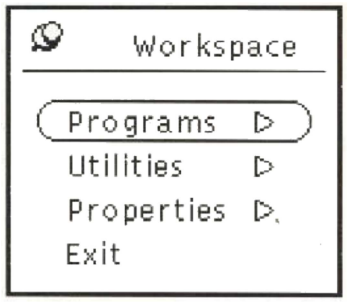
```
$ olinit
```

Run OPEN LOOK, step 3

The screen goes black, then grey, then white. The pointer changes to an X, then to an arrow. Finally, the Workspace menu pops up.

Select the programs option

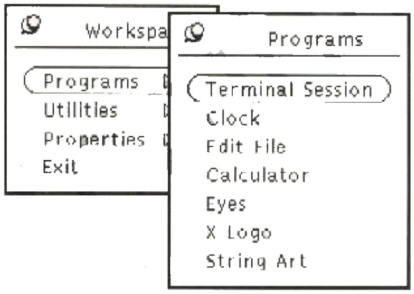
4. Select the programs option by moving the pointer over the option and clicking your left mouse button once.



OPEN LOCK Workspace window, step 4

Select Terminal Session

5. Another small window pops up. Select the terminal session (xterm) option by clicking on the word with your left mouse button.

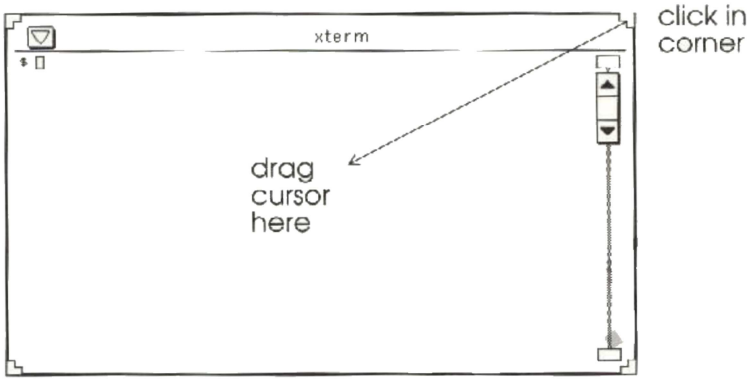


OPEN LOOK Workspace and Programs windows, step 5

The X Window System displays an xterm window where you can perform UNIX commands or start another application.

Shrink window

6. Let's make the window a little smaller. Move the cursor to the corner marker in the upper right corner of the window. Hold down the left mouse button and move the cursor down and to the left until the window is about five inches by three inches.



Reduce the size of the window, step 6

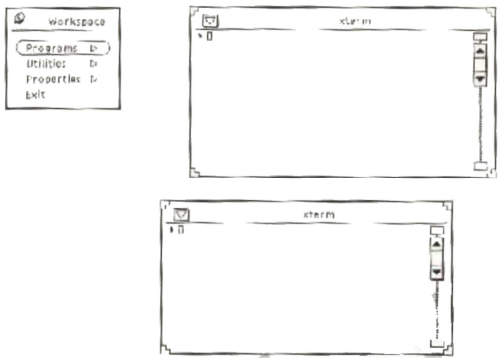
Move window
7. Move the window to the bottom of your screen by clicking (with the left mouse button) on the top bar of the menu, then holding down the button and moving the entire Window down.

You resize windows by grabbing the corner; you move windows by grabbing the border.

Add another window
8. Add nother window. Select the programs option window from the OPEN LOOK Workspace window.

9. Select the Terminal Session (xterm) option.

Shrink and move second window
10. Now make the new window smaller (drag the corner), and move it to the right side of your screen (drag the border).

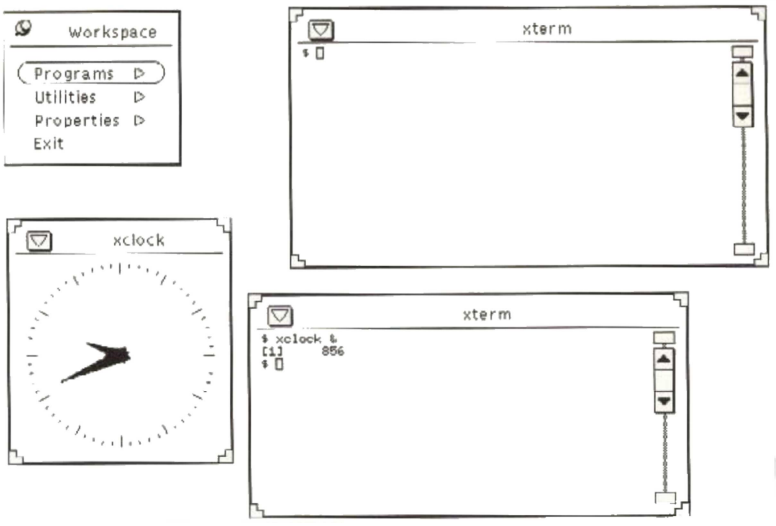


Working with X windows, steps 7-10

Active window
11. Select the bottom window by clicking the left mouse button on the top bar until the bar turns dark. The dark bar indicates the active window; anything you type appears only in this window.

Run Xclock
12. Type **xclock &** and press RETURN to make the X clock appear.

13. If you like, you can reshape and move the xclock window the same way you manipulated the xterm windows.

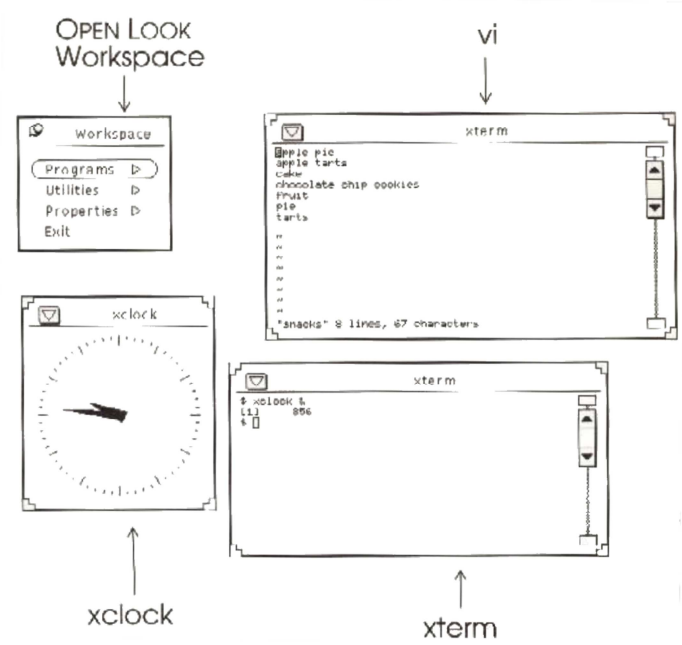


Working with X windows, steps 11- 13

Change active windows
14. Make the top xterm window the active window. Click the left mouse button once on the top bar until it turns dark.

Run vi
15. Type **vi snacks** and press RETURN. You're using the **vi** editor in this window.

You're now running two different programs (**vi** and **xclock**) on one screen. You can use the other xterm window to run more programs if you like.

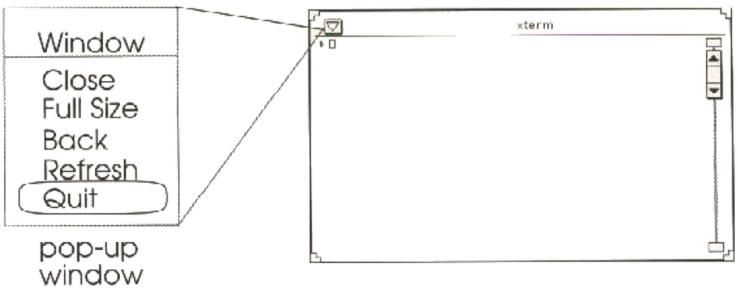


One screen with several windows

Quit the vi window with pop-up window
16. Quit out of the snacks file (**:q!**).

17. Quit out of the window by selecting a quit option on a pup-up menu.

To open the pop-up window, click the right mouse button on the arrow in the upper left corner and drag it down until the pop-up window displays. To select quit from this window, move the darkened bar over quit.



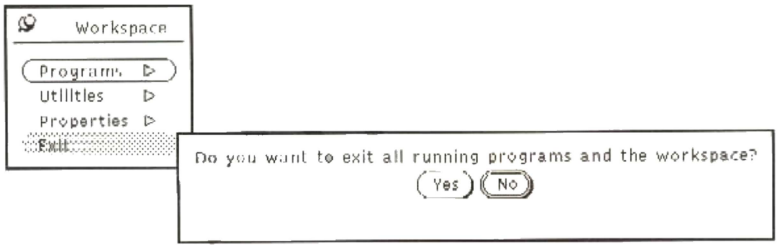
Quit an xterm window with the pop-up window, step 17

Quit xterm
18. Quit out of the other xterm window.

Exit X Window System

19. To exit the X Window System, click on the exit option in the OPEN LOOK Workspace window.

20. When OPEN LOOK asks if you want to exit all running programs and the workspace, click on yes.



Exit OPEN LOOK Workspace window, steps 19 and 20

Chapter review

Here are some important points you learned in this section:

- You can use the X Window System and OPEN LOOK to create as many windows as you want on a screen.
- Each window acts like a separate terminal screen; you can run different programs in each window.
- Before you use OPEN LOOK for the first time, type **/usr/X/adm/oladduser**; this sets up the OPEN LOOK default values in your home directory. Log out, then log in to use the changed startup file.
- Run **olinit** to start OPEN LOOK.
- You make windows active by clicking with the left mouse button on the top bar until it turns dark.
- You resize windows by grabbing the corner marker; you move windows by grabbing the border.
- To quit any OPEN LOOK (xterm) window, select the quit option from a pop-up menu. You display the pop-up window by clicking on the arrow icon in the left corner and dragging the cursor down with the right mouse button.