
User's Guide to Solaris (Sun Unix 7)

1 INTRODUCTION	1
2 GETTING ON AND OFF THE SYSTEM	1
2.1 LOGGING ON TO THE SYSTEM:	1
2.2 LOGGING ON	2
2.2.1 <i>At the Login Prompt</i>	2
2.2.2 <i>Errors at Login</i>	2
2.3 CHANGING YOUR PASSWORD	3
2.4 RE-SETTING YOUR TERMINAL TYPE	4
2.5 LOGGING OUT	4
2.6 CLEARING THE SCREEN	4
3 TYPING COMMANDS	5
3.1 THE COMMAND LINE PROMPT	5
3.2 EXECUTING COMMANDS.	5
3.3 TYPING MORE THAN ONE COMMAND ON A LINE	7
3.4 PREVIOUS COMMANDS	7
3.5 FILE COMPLETION	8
4 FILES	8
4.1 CREATING FILES	8
4.1.1 <i>PICO</i>	9
4.1.2 <i>JOVE</i>	9
4.1.3 <i>EMACS</i>	9
4.1.4 <i>VI</i>	9
4.2 NAMING FILES	9
4.3 LISTING FILES	10
4.4 VIEWING FILES	10
<i>MORE</i>	10
4.5 MANIPULATING FILES	11
4.5.1 <i>Copying files</i>	11
4.5.2 <i>Renaming files</i>	11
4.5.3 <i>Removing files</i>	11
4.5.4 <i>Creating File Links</i>	12
4.5.5 <i>Protecting Files</i>	12
4.5.6 <i>Comparing files</i>	13
4.5.7 <i>Searching files</i>	14
5 ORGANIZING FILES- DIRECTORIES	15
5.1 THE STRUCTURE OF THE SYSTEM	15

5.2	PATHNAMES	15
5.3	CREATING DIRECTORIES	16
5.4	DELETING DIRECTORIES	16
5.5	CHANGING DIRECTORIES	17
5.6	MOVING FILES BETWEEN DIRECTORIES	18
5.7	WILDCARD CHARACTERS	18
6	THE SHELL	19
6.1	WHAT IS THE SHELL?	19
6.1.1	<i>Perl Script</i>	19
6.2	CUSTOMIZING YOUR ENVIRONMENT	19
6.2.1	<i>History</i>	20
6.2.2	<i>Aliases</i>	20
6.2.3	<i>Line Prompt</i>	21
6.2.4	<i>Setting Your Search Path</i>	21
6.3	MANAGING JOBS	21
6.4	RUNNING BACKGROUND PROCESSES	22
6.5	REDIRECTING STANDARD INPUT, OUTPUT, AND ERROR	22
6.6	PIPES	23
7	PROGRAMMING	24
7.1	NAMING PROGRAMS	24
	<i>Programname.f</i>	24
7.2	COMPILING PROGRAMS	24
7.2.1	<i>Java</i>	24
7.2.2	<i>Pascal</i>	24
7.2.3	<i>Fortran</i>	24
7.2.4	<i>C and C++</i>	25
7.3	RUNNING PROGRAMS	25
8	COMMUNICATING WITH OTHER USERS AND OTHER SITES	26
8.1	MAIL AND ADDRESSING	26
8.2	PINE	26
8.2.1	<i>Creating/Sending Mail</i>	27
	<i>Commands to use when composing a message.</i>	27
	Mark, Cut and Paste in Composer (block moves)	28
	ATTACHING A FILE TO A MAIL MESSAGE	28
8.2.2	<i>Reading Your Mail</i>	29
8.2.3	<i>Receiving Attachments</i>	29
8.2.4	<i>Replying to an e-mail message.</i>	29
8.2.5	<i>Exporting a Mail Message to a File in your Home Directory</i>	31
8.2.6	<i>Viewing Messages You Have Sent</i>	31
8.2.7	<i>Printing Your E-mail</i>	31
8.2.8	<i>Address Book</i>	32
	<i>To Add a Name:</i>	32

<i>To Create a List:</i>	32
<i>To Edit an Address:</i>	32
8.2.9 <i>Spamming/Mass Mailing</i>	32
8.2.10 <i>Conserving Disk Space</i>	33
8.3 INTERNET: FTP, TELNET	33
8.3.1 <i>TELNET</i>	33
8.3.2 <i>FTP</i>	33
8.3.3 <i>The World Wide Web</i>	35
9 GETTING HELP	35
9.1 THE MAN COMMAND	35

1 Introduction

Throughout this document, the characters the user types are written in **courier bold face type** and computer responses are written in *typewriter type*.

The general format for a command is centered with the command in **bold face** and variable information, such as filenames to be used with the command, written in *italics*.

When a word is surrounded by < and > symbols, it is referring to the typewriter key of that name. For instance, <ENTER> means to press the key that says " ENTER ".

2 Getting On and Off the System

2.1 Logging On to the System:

From a modem off campus: --

Western Suffolk:

Students 762-1000

Faculty/Staff 762-2000

Nassau

Students 417-1000

Faculty/Staff 417-2000

New York City

Students 718-807-1000

Faculty/Staff 718-807-2000

** You *may* get a window prompt asking for your login information. Here you will enter in your userid and password:

When you see a prompt similar to: `NOC-AS1-2>`

Type: `ms.cc.sunysb.edu`

You will then be prompted to enter your `userid` and your `password`.

Telnetting from another machine:

`telnet ms.cc.sunysb.edu`

2.2 Logging On

2.2.1 At the Login Prompt

When you see the login prompt:

```
login:
```

Type your username and press <ENTER>. Next, it will ask you for your password. While entering your password, the characters will not be visible on the screen. This is so that your password remains secret.

Note: The first time you sign on, you may be asked to change your password to a new one.

If so, you will be asked to type your old password and your new password twice. Be sure that your new password has these characteristics:

- 6 to 8 characters long.
- It's not your name or id or a common word.
- At least 1 character is a number or punctuation.
- Cannot have more than 3 numbers
- Use a combination of letters and numbers.
- Upper case and lower case letters are interpreted as different characters.

For more information about password guidelines, please see the section "*Changing Your Password.*"

If you successfully log on, you will receive the messages of the day. **It is very important to read the messages!** These messages provide system information regarding shutdowns and related information.

Hit the space bar to move down the screen, then press <ENTER> when prompted to.

As the log in process is running, you may be asked to specify a terminal type. If you get the prompt:

```
TERM = (vt100)
```

Simply press <ENTER> your terminal will be set to the default in the parenthesis: (vt100) in this case). If a mistake occurs, see "*Setting the Terminal Type*", in the section on shells, to change.

Most of the time your terminal should be set to: vt100

2.2.2 Errors at Login

Several errors may occur when you log in. If you get the message:

```
Login incorrect  
login:
```

This means that either your username is not valid, you made a typing error in entering either your username or password, or your password is incorrect. If it was only a typing error, you can re-enter your username and password correctly. (Also, note that your username should be in lower case letters and not uppercase). If you have the wrong password, you will need to contact the Computer account office at 632-8011.

```
Your password has expired. Choose a new one
Changing password for (your username)
```

```
New password:
```

You must choose a new password. The procedure is similar to that in the section on changing passwords.

2.3 Changing Your Password

You want to change your password periodically to prevent anyone else from using your account. If you keep the same password for more than 90 days (approximately 3 months), the system will force you to change it.

If you would like to change your password before the 90 days, you need to use the password command:

```
ms.cc.sunysb.edu.3 passwd username
```

```
Changing password for (your username)
Current password:
```

Now, enter your old password (which will not be visible on the screen). You will be prompted to enter your new password:

```
New password:
```

Then, re-enter your new password.

```
Re-Enter your new password:
```

This is to make sure you entered it correctly, since you cannot see if you made a mistake. (If it was misspelled, you would not be permitted to log in again unless you made the same spelling error.)

You should select a password which you will remember, but someone else won't guess. Be sure to follow these guidelines when choosing your password:

- 6 to 8 characters long.
- It's not your name or id or a common word.
- At least 1 character is a number or punctuation.
- Cannot have more than 3 numbers
- Use a combination of letters and numbers.
- Upper case and lower case letters are interpreted as different characters.

After running the **passwd** command, several error messages may appear. If you receive an error message, you may execute the **passwd** command and try again. If you get this message:
Too many failures - try later.

You have made more than two mistakes when entering your new password.

2.4 Re-setting Your Terminal Type

If your terminal type is set incorrectly, refer to the section on “*Customizing Your Environment*” to reset it.

2.5 Logging Out

At the end of your session, before you leave your computer, **you must log out.**

The logout command will take you off of the system.

[Disconnected from ms.cc.sunysb.edu'] means that you have successfully logged out.

```
ms.cc.sunysb.edu.1 logout
```

2.6 Clearing the screen

Whenever you would like to clear the screen, type:

Clear

3 Typing Commands

3.1 The Command Line Prompt

The command line prompt means that the computer is ready for an instruction. Normally, the prompt looks like `ms.cc.sunysb.edu.5`. The number after the prompt is the command number in your current shell. If the prompt is not to your tastes it may be changed. You can locate the prompt by pressing `<ENTER>` several times. The prompt will be displayed every time you press `<ENTER>`

```
ms.cc.sunysb.edu.3 <ENTER>
ms.cc.sunysb.edu.3 <ENTER>
ms.cc.sunysb.edu.3
```

For clarity, this guide will use the `ms.cc.sunysb.edu.3` prompt to show what the user would type.

If you make a mistake typing a command, a backspace will erase the previous character and a `<CTRL>-u` will erase the whole line.

To change backspace to another key (such as `<DELETE>`):

```
stty erase <KEY TO BE USED>
```

For example:

```
ms.cc.sunysb.edu.3 stty erase <DELETE> <ENTER>
```

3.2 Executing Commands.

To execute a command, type the command's name after the prompt and press `<ENTER>`. The command will begin to run. When the command is finished, the prompt reappears to show that it is ready for a new instruction. Try executing the **whoami** command now. This command tells you your username. If your username was `mork`, this is what would happen:

```
ms.cc.sunysb.edu.3 whoami <ENTER>
mork
```

(From here on, the `<ENTER>` reminder will not be shown.)

To see who else is on the system, execute the **users** command:

```
ms.cc.sunysb.edu.3 users
mork mindy beepo
```

To find out more about the other users, try the `who` command. This will show which terminal they are on and what time they logged on:

```
ms.cc.sunysb.edu.3 who
```

```
elee      Pts/39      Jul 18 09:12      (as6.dialup.sunysb.edu)
ghui      Pts/143     Jul 17 09:56     (libibm06.sinc.sunysb.edu)
asattar   Pts/41      Jul 20 08:58     (csibm08.sinc.sunysb.edu)
```

To see the date and time, use the `date` command:

```
ms.cc.sunysb.edu.3 date
```

```
Fri Feb 26 12:13:40 EST 1999
```

For a calendar of the current month, use the `cal` command:

```
ms.cc.sunysb.edu.4 cal
```

```
      July 1998
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
```

For a calendar of any other month, use the `cal` command this way:

cal *month year*

For example, try September of 1752:

```
ms.cc.sunysb.edu.3 cal 9 1752
```

```
      September 1752
S  M Tu W Th F S
      1  2 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

For a calendar of the whole year, use the `cal` command like this:

cal *year*

For example, to make it show the year 1989:

```
ms.cc.sunysb.edu.3 cal 1989
```

The **expr** command allows you to calculate expressions. One example of its use is like this:

```
ms.cc.sunysb.edu.3 expr 2 + 2  
4
```

3.3 Typing More Than One Command On a Line

You can instruct the system to run several commands on the same line if you separate them with a semicolon:

command1; command2; command3

For instance, you may type:

```
ms.cc.sunysb.edu.3 whoami;date;users  
  
mork  
Fri Jun 30 13:59:52 EDT 1989  
mork mindy beepo
```

3.4 Previous Commands

The **history** command allows you to see the last *n* commands you ran. '*n*' is a number that can be changed and will be discussed in the section on shells. Each command is numbered beginning with the earliest.

```
ms.cc.sunysb.edu.8 history  
1. whoami  
2. users  
3. who  
4. date  
5. cal  
6. cal 10 1367  
7. cal 1989  
8. whoami;date;users
```

Previous commands can be re-executed by referring to the number next to them or the first distinct letters of the command. The numbers by the commands are the command numbers (the numbers you see in the shell prompts). The format is an exclamation point followed by the number or letters. For example, if you wish to run the second command again, type this:

```
ms.cc.sunysb.edu.9 !2  
users  
mork mindy beepo
```

If you wish to execute the eighth command you could type:

```
ms.cc.sunysb.edu.10 !wh
```

```
whoami;date;users  
mork  
Fri Jun 30 12:36:45 EDT 1989  
mork mindy beepo
```

Previous commands may also be referred to by using relative numbers. That is, the last command would be -1, the one before it would be -2, etc. So, instead of using **!2** you could have used **!-7**.

3.5 File Completion

First, you must edit your .cshrc file. On the last line of your .cshrc file, type:

```
set filec
```

The key for command completion depends on the shell that you are running. If you are using csh, you should press the <ESC> key. If you are using bash, press the <TAB> key.

If you enter the first few letters of a filename and press <ESC>, the computer will try to complete the word for you. If the letters you entered are not unique to one filename, it cannot complete the rest.

The computer will complete the word as far as it can and let you type the rest. In this case it does not know which program you wish to compile, but it will fill in the rest of the letters up to the point where the two files differ. If you are not sure what the rest of the name is you can then press <CTRL>-D . This will give you a list of files that have the beginning that you have specified and then leave you where you were in typing your command..

NOTE: Make sure when you press <CTRL>-D you are not at the first character of the command. If you press <CTRL>-D by itself it will act as if you typed **logout**.

4 Files

4.1 Creating Files

Files are created in an editor.

The mail server has a few text editors that you can use, they include:

- pico
- jove
- emacs

- vi

4.1.1. PICO

To use pico, type:

pico *filename*

If the *filename* does not exist, you will get a blank screen and you can create your file. At the bottom of the screen will be your command keys. To use the commands, please note that the caret character ^ represents the <ctrl> key.

4.1.2. JOVE

To invoke the editor, use the **jove** command this way (Jove is a version of the Emacs editor):

jove *filename*

Where *filename* is the name of your new file. For a tutorial on Jove, run the **teachjove** command:

```
ms.cc.sunysb.edu.3 teachjove
```

4.1.3. EMACS

If you wish to use gnuemacs for your editor instead of jove you would first have to make sure your path includes **/usr/local/bin** (this will be discussed in the section on the shell). Then you would just type:

emacs *filename*

4.1.4. VI

vi *filename*

There are jove, emacs, and vi reference sheets available at the SINC Sites.

4.2 Naming Files

File names can contain up to 14 characters, which can be any combination of letters, numbers or special characters (like =, - , _). **Caution:** Do not start any file names with a period (.) or a dash (-) since a file that starts with a period is a system file, and files with dashes in front are hard to delete.

The following are valid names:

Stuff	Bill_Ted	321-Contact
Important.bak	Readme	Ernie+Bert

Since Solaris Unix interprets upper case and lower case letters as different, the following are all separate files:

```
Ernie+Bert          ernie+bert          ERNIE+BERT
```

4.3 Listing Files

The **ls** command allows you to see what files are in your directory.

```
ms.cc.sunysb.edu.3 ls
113.hw          a.out letters-ernie          startrek          stuff
```

File names beginning with a dot are invisible. The **ls** command does not normally display these files. In order to list the invisible files, you must use the **-a** option for the **ls** command:

```
ms.cc.sunysb.edu.3 ls -a
.cshrc          .login          113.h letters-Ernie
stuff a.out startrek
```

The **ls -l** command will give you a long listing of your files. This will include the owner of the file, the time it was last modified and information on how it is protected.

4.4 Viewing Files

MORE

The **more** command allows you to see the contents of a file. The format for this command is:

more filename

For instance, if you wish to read the contents of the file *startrek*, type this:

```
ms.cc.sunysb.edu.3 more startrek
```

These are the voyages of the starship Enterprise.

Its five year mission:

To explore strange, new worlds,

To seek out new life and new civilizations,

To boldly go where no man has gone before.

If the file was more than one screen long, it would show at the bottom of each page how much of the file has been read.

Press:

<space bar>	To see the next page
q	To stop the listing
b	To see the previous page
h	For help on additional commands

4.5 Manipulating Files

4.5.1 Copying files

The **cp** command is used to copy a file. The format is like this:

```
cp oldfile newfile
```

If you wish to copy the `startrek` file and name it *Enterprise*, the command would be:

```
ms.cc.sunysb.edu.3 cp startrek Enterprise
```

Be careful when naming the new file because if a file by that name already exists, it will be erased and replaced with the copy.

4.5.2 Renaming files

The **mv** command (for 'move') can be used to rename files. The format is similar to that of the **cp** command:

```
mv oldname newname
```

The following would change the name of the `113.hw` file to `hw1.113`

```
ms.cc.sunysb.edu.3 mv 113.hw hw1.113
```

As with the **cp** command, if the new name is already the name of an existing file, that file will be lost.

4.5.3 Removing files

The **rm** command removes files from your directory. The structure is simple:

```
rm filename
```

To erase the file `hw1.113`, type the following:

```
ms.cc.sunysb.edu.3 rm hw1.113
```

Now, a listing of your files will show that *hw1.113* no longer exists. The **rm** command with the **-i** option is very useful. With this option, the computer first asks to make sure you want to erase the file.

```
ms.cc.sunysb.edu.3  rm -i  hw1.113
hw1.113: ?
```

The computer is asking for a confirmation. Type **"y"** to erase and anything else to abort.

4.5.4 Creating File Links

It may be the case that there are files in other directories (such as the shareware directory) that you want to have a copy of in another directory of yours. Making a copy with **cp** is undesirable in this case as it will use extra disk space when it is not necessary to do so (if the file is already on the disk why make it there twice).

Instead, you would use the **ln** command. This command makes an entry in your directory as though a file were there, but it is really just a note for the computer to use the original copy.

For example, to create a "copy" of **/usr/local/bin/emacs** in your current directory as **emacs** you would type:

```
ms.cc.sunysb.edu.3  ln -s /usr/local/bin/emacs emacs
```

Now a directory listing via **ls** will show a "file" emacs.

4.5.5 Protecting Files

Only the owner of a file can change the mode of that file. If you want to protect certain files from being read, executed, or modified by you or other users, use the **chmod** (change mode) command. The command is used this way:

chmod *code filename*

where the code is a three digit number. The first digit refers to the user, the second refers to the user's group, and the third refers to others.

The following numbers allow these permissions:

- 0—Cannot access file
- 1—Can execute file
- 2—Can write to file
- 4—Can read file
- 7—Can read, write, and execute file

These numbers can be added to each other to make more combinations. The **4** for reading files and the **2** for writing to files can be added to make **6** to allow reading and writing to the file.

The code **751** means that you (the owner) can read, write and execute the file, the members of the group can only read and execute the file, and all other people can only execute the file.

If you wanted to protect your *startrek* file so that nobody else can touch it, and you can only read it:

```
ms.cc.sunysb.edu.3 chmod 400 startrek
```

This method of protecting files can also be used to protect directories.

4.5.6 Comparing files

If you want to compare two different files, the **cmp** command will show you the first instance where they differ. If both files are identical, it does not say anything. The command works like this:

```
cmp file1 file2
```

For instance, if the file *poem1* contained this:

```
Jack and Jill went up the hill  
To fetch a pail of water
```

And the file *poem2* contained this:

```
Jack and Jill went up the hill  
To look for Bo Peep's sheep  
Down the hill fell Jack and Jill  
And frightened Miss Muffet away
```

If you wish to compare these two **files**, type this:

```
ms.cc.sunysb.edu.3 cmp poem1 poem2  
poem1 poem2 differ: char 35, line2
```

It says this because the first difference occurs at the 35th character- which appears on line 2.

The **diff** command shows the difference between the files. First, it lists the lines of the first file that are not in the second, then the lines of the second not in the first. For example the following command will produce:

```
ms.cc.sunysb.edu.3 diff poem1 poem2  
  
2c2  
< To fetch a pail of water  
---  
> To look for Bo Peep's sheep
```

```
> Down the hill fell Jack and Jill  
> And frightened Miss Muffet away
```

4.5.7 Searching files

The **grep** command searches for a word or group of words in a file. It prints the lines containing the word or group of words. Groups of words should be put inside single quotes:

grep *'group of words' filename*

If you wanted to find the lines of the file *poem2* containing the phrase "Jack and Jill", the command would look like this:

```
ms.cc.sunysb.edu.3 grep 'Jack and Jill' poem2
```

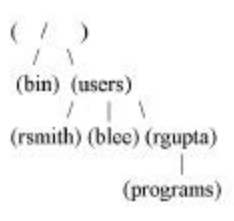
```
Jack and Jill went up the hill  
Down the hill fell Jack and Jill
```

Since that phrase was present more than once, all lines containing it are printed. If the phrase does not appear in the file, nothing is printed.

5 Organizing Files- Directories

5.1 The Structure of the System

The system is organized in a tree structure where directories can have subdirectories, called *children*. The directory a child directory is under is called its *parent* directory. The root directory has no parent and all other directories are its children or grandchildren or great-grandchildren.



In the system represented by the diagram above, the backslash indicates the root. The **users** directory is a child of the root, and a parent to subdirectories **rsmith**, **blee**, and **rgupta**. The subdirectories can, in turn, have children of their own.

5.2 Pathnames

When you first log on, you are put into your *home directory*; but you can move around to other directories. The directory that you are in at any particular time is your *current directory*.

The **pwd** command will show you the pathname of your current directory. The pathname begins with the root (indicated by the first backslash) and shows you the lineage down to your current directory. Names are separated with backslashes ("/"). If you were in directory **blee** and executed the **pwd** command, you would see this:

```
ms.cc.sunysb.edu.3 pwd  
/users/blee
```

This is so because **users** is the parent of **blee** and the root is the parent of **users**. Pathnames such as this are called *absolute pathnames* because they mean the same thing any place on the system.

Relative pathnames depend on where you are. A single dot (".") is a relative pathname indicating your current directory. Two dots ("..") indicate the parent of your current directory.

Users' home directories can be referred to by using the tilde ("~"). For example to get a listing of **dnovell's** home directory you would type:

ls ~dnovell

In **dnovell's** home directory, there is a file called: **core**

Core files are created when a program crashes, they are very big and should be deleted immediately when created.

If you wanted to remove a "core" file in your home directory you could just type:

```
ms.cc.sunysb.edu.3 rm ~/core
```

A tilde followed immediately by a slash or a space refers to your home directory.

5.3 Creating Directories

You can create subdirectories to your directory using the **mkdir** command.

mkdir *directoryname*

Directory names follow the same rules as filenames. If you wish to make a directory called **stuff** you can type:

```
ms.cc.sunysb.edu.3 mkdir stuff
```

5.4 Deleting Directories

The **rmdir** command removes a directory.

rmdir *directoryname*

You can only delete an empty directory.

To delete your directory **stuff** type:

```
ms.cc.sunysb.edu.3 rmdir stuff
```

If there are files in the directory you are trying to delete, you will be told that it is not empty and it will not be deleted.

5.5 Changing Directories

The **cd** command allows you to change your current directory.

cd *pathname*

To go to a child of your current directory, the pathname is simply the name of the subdirectory. If you were in **rgupta** and wanted to go to **programs** then the command is:

```
ms.cc.sunysb.edu.3 cd programs
```

"**cd**" by itself will take you back to your home directory:

```
ms.cc.sunysb.edu.3 cd
```

Relative pathnames can be used to go up to a parent directory. If you are now in **rgupta**, the following will take you to **users**:

```
ms.cc.sunysb.edu.3 cd ..
```

By executing the **pwd** command, you can see that you are now in the users directory.

```
ms.cc.sunysb.edu.3 pwd
/users
```

From here, you can use another relative pathname to go to **programs**.

```
ms.cc.sunysb.edu.3 cd rgupta/programs
```

From anywhere on the system, you can specify the absolute pathname for a particular directory.

```
ms.cc.sunysb.edu.3 cd /users/blee
```

Notice that absolute pathnames begin with a backslash while relative pathnames do not.

5.6 Moving Files between Directories

The `mv` command was previously used to rename files, but it can also be used to move them to other directories. By renaming a file to include the pathname of another directory, you have moved the file.

If you want to move your *startrek* file into your subdirectory *stuff* do this while in your home directory:

```
ms.cc.sunysb.edu.3 mv startrek stuff/startrek
```

Pathnames here follow the same rules as above with the `cd` command.

5.7 Wildcard Characters

Filenames and directories can be referred to by using wildcard variables. The asterisk (*) is used to refer to any number of characters in a name. Used by itself, it would just refer to all of your files and directories. For instance, to list your files:

```
ms.cc.sunysb.edu.3 ls *  
  
homework          hooks          morework  
  
book:  
chapter 1  chapter2      chapter2b     chapter3      foreword
```

This shows the files in your directory, as well as files in your subdirectories. (In this case, *book* is a subdirectory containing files concerning a book you are writing.)

If you want to list only files beginning with an *ho*, use the asterisk to refer only to the end of the name:

```
ms.cc.sunysb.edu.3 ls ho*  
  
homework          hooks
```

You can list all files ending in certain letters by using the wildcard to fill in the beginning:

```
ms.cc.sunysb.edu.3 ls *work  
  
homework morework
```

The question mark is used to refer to only one character. This is useful when files are set up the way the chapters are in the subdirectory *book*. If you wanted to list these files:

```
ms.cc.sunysb.edu.3 ls book/chapter?
```

book/chapter1 book/chapter2 book/chapter3

The file *chapter2b* is not listed because there is more than one character after the letters *chapter*.

Wildcard characters can be used with any command requiring an existing filename or directory name. (such as **cp**, **mv**, **rm**, **rmdir** and **lp**)

6 The Shell

6.1 What is the Shell?

The shell is a command interpreter. It acts as a go-between between you and the operating system.

To find out the name of the shell you are using, type:

```
ms.cc.sunysb.edu.3 echo $shell
/bin/csh
```

This shows that you are in what is called the **C shell**. The syntax of this shell resembles the **C** programming language. This shell has many features, some of which are a command history buffer and command aliases

6.1.1 Perl Script

In order to use Perl, you need to have the following in your path:
`/usr/local/bin`

The first line of your PERL Program must be:
`#!/usr/local/bin/perl`

The permissions on your program must be set so that the program is executable.

6.2 Customizing Your Environment

By running the **printenv** command, you can view current information about your account, such as your home directory, your login name, which shell you are put into, your terminal type, etc.

```
ms.cc.sunysb.edu.3 printenv
```

Terminal settings are shown with the command:

```
ms.cc.sunysb.edu.3 stty
```

This information comes from two hidden files in your home directory called `.cshrc` and `.login`. By changing or adding certain lines of these files, you can customize your account. These are invoked whenever you log on.

Changes to the `.cshrc` file will not take effect until you either execute the command. You can either use the **source** command or you can log off and then back on again.

The source command is used in the following way:

```
source <filename>
```

```
ms.cc.sunysb.edu.3 source .cshrc
```

6.2.1 History

There is a line in your `.cshrc` file that looks like this:

```
set history = 20
```

If you type:

```
ms.cc.sunysb.edu.3 more .cshrc
```

And you do not see **set history** in your file, you should edit the file and add the history command. The format of the command is:

```
set history = #
```

is how many commands you want it to remember at one time

This shows how many commands will be "remembered" in the command history buffer. When you execute the **history** command, you are viewing the contents of this buffer. The number can be changed to whatever you want.

6.2.2 Aliases

The **alias** command allows you to call commands by a different name. This function is useful for shortening long commands that you use often. Alias commands can be added onto the end of your `.cshrc` file. When the command is placed here, the alias will work every time you log on. The format for the function is:

alias newname command

For instance, if you want to shorten the **history** command to "h", add this to your `.cshrc` file:

```
alias h history
```

Now, instead of typing the entire command **history**, you can just enter an "h" to run the command.

6.2.3 Line Prompt

Your line prompt can be changed to something else by changing the line that begins with `set prompt`.

For instance, the default prompt, which tells you the machine you are on and the number of the next command, is set with the command:

```
set prompt = "hostname'.!> "
```

6.2.4 Setting Your Search Path

Your search path is a list of directories that can have commands in them. When you type a command the shell searches each of the directories, in order, looking for your command. If it is found, the command is executed. If not, an error message is returned. To see what directories are in your path, you can type:

```
ms.cc.sunysb.edu.3 echo $path  
/bin /usr/bin /usr/local/bin /usr/bin/X11 .
```

This shows that there are five directories in your path, the four with "bin" in them and "." (your current directory).

6.3 Managing Jobs

When you run a command or a program, the shell creates what is known as a *process*. A process is a copy of the program that is currently running. Each process is given a number called a process-id (PID). To see the processes and PID's currently being executed, type the `jobs` command with the `-l` option:

```
ms.cc.sunysb.edu.3 jobs -l
```

If you wish to abort a process, use the **kill** command:

```
kill PID
```

This is useful if your keyboard is locked up because of a certain process. You could log into another machine and kill the process causing the problem.

If you wanted to abort process number 8270,

```
ms.cc.sunysb.edu.3 kill 8270
```

If you are totally stuck and nothing else works, log into another machine and execute the **ps -u** command with your own username:

```
ps -u username
```

This will list the running processes and their PID's. Kill them using the **kill** command with the **-9** option:

```
kill -9 PID
```

You can only kill your own processes.

6.4 Running Background Processes

Most commands take only a few seconds to run, but when you run something that takes longer, you may want to put it into the *background* so you can do something else in the *foreground* while it is running. To put a process in the background, type an ampersand after the command.

```
command &
```

Another way to put a process in the background is, while it is running, type **<CTRL>-z** (which stops the process) and enter **bg** to put it in the background. After entering a background command, it will show you the PID of that process. You will receive a message when the process is done. To put a background process into the foreground, type **fg**. If more than one process is running in the background, you can specify which one you want in the foreground by including the job number of that process in the **fg** command:

```
fg %n
```

where *n* is the job number (not process id).

6.5 Redirecting Standard Input, Output, and Error

When you run a command such as **date** the output automatically goes to the screen. This is because the screen is the standard output. If you wish to redirect the output to a file, use the ">" sign.

```
command > filename
```

If you want to put the output from the **date** command into the file *today*:

```
ms.cc.sunysb.edu.3 date > today
ms.cc.sunysb.edu.3 more today
Fri Feb 26 12:28:24 EST 1999
```

This created a file called *today* and put the output into this file. If you wish to append an existing file, use two of the ">>" signs:

command >> existing filename

If you want to add the output of the **users** command to the file *today*

```
ms.cc.sunysb.edu.3 users >> today
ms.cc.sunysb.edu.3 more today
Fri Feb 26 12:28:28 EST 1999
bill ted ernie orson
```

To redirect standard input, use the "<" sign.

command < datafile

If you wish to mail somebody a file, you could use the file as standard input to the mailx command. If you want to send Ted your startrek file:

```
ms.cc.sunysb.edu.3 mailx ted < startrek
```

Standard error is where error messages are sent (normally the screen). To redirect standard error:

command >& filename

If a particular command produces an error message, it will be sent to a file along with the other output. For instance, if you compile a program and get error messages, they will be sent to the file.

```
ms.cc.sunysb.edu.3 f77 prog.f >& results
```

The error message can be found in the file *results*.

6.6 Pipes

You can use the shell to create pipelines. Pipelines are used to send the output from one command as the input for another. Commands in a pipeline are separated by a vertical line <|> (<shift> \)

command1 | command2 | | commandN

The output from the first command is used as input for the second, then output from the second can be used as input for a third command and so on. For instance, the command **wc -c** counts the number of characters in its input. If you connect this with the output from the **whoami** command, you can get the number of letters in your username (plus one for the line feed at the end):

```
ms.cc.sunysb.edu.3 whoami | wc -c
```

7 Programming

7.1 Naming Programs

Programs are written using an editor. You should name the file with an extension after the "." which indicates the programming language you are using.

Language	Filename
Java	Programname.java
Pascal	Programname.p
FORTTRAN	Programname.f
C or C++	Programname.c

7.2 Compiling Programs

7.2.1 Java

To compile the Java program, use the **javac** command:

javac *filename*

7.2.2 Pascal

To compile and link a Pascal program, use the **pc** command.

pc *filename*

7.2.3 Fortran

The **f77** or **f90** commands are used for FORTRAN programs:

f77 *filename*

f90 *filename*

NOTE: For more information about Fortran, please refer to: <<http://www.sinc.sunysb.edu/Help/fortran.html>>

7.2.4 C and C++

Either the **gcc** command or the **g++** command can be used to compile and link C programs:

```
gcc filename  
g++ filename
```

The **CC** command is to compile and link C ++ programs:

```
CC filename
```

Whenever you compile a program written in either: C, C++, Fortran, or Pascal, if there are no errors, this will create an executable file called *a.out*.

If you wrote a Pascal program called *progl* and stored it in file *progl.p*, to compile and link it, type:

```
ms.cc.sunysb.edu.3 pc progl.p
```

To redirect the standard error to a file when compiling and linking a program, (let's say a pascal program called *progl.p*):

```
ms.cc.sunysb.edu.3 pc progl.p >& errorfile
```

The **-L** option for the **pc** and **fc** commands will give a listing of the program with any embedded syntax errors.

```
ms.cc.sunysb.edu.3 pc -L progl.p > errorfile
```

The program listing with the errors will be in **errorfile**.

7.3 Running Programs

To run a successfully compiled program, type this:

```
ms.cc.sunysb.edu.3 a.out
```

If input is from the keyboard, a **<CTRL>-d** represents an EOF.

If you do not have **.** in your path, you will have to type:

```
ms.cc.sunysb.edu.3 ./a.out
```

By using the ideas in the section on redirecting standard input and output, you can use datafiles as input for your programs and have the output (including the error messages) redirected to a file. If you have information in the file *data1* for the program you compiled, you can run this:

```
ms.cc.sunysb.edu.3 a.out < data1
```

If you wish to redirect the output to a file that you can later print out to hand in:

```
ms.cc.sunysb.edu.3 a.out < data1 >& output
```

The file *output* will be created and will contain the output from the program.

Now, if you wish to use the input file *data2* for your program and have the output *appended* to the existing file *output*:

```
ms.cc.sunysb.edu.3 a.out < data2 >>&output
```

8 Communicating With Other Users and Other Sites

8.1 Mail and Addressing

When sending mail to someone on this system you may simply mail to their username.

To mail to someone with an Internet address, the format is:

username@internetaddress

An Internet address is usually in the form xxx.yyy.zzz where *zzz* signify the type of organization (.edu for educational, .gov for governmental, .mil for military, .com for commercial, .uk for installations in the United Kingdom, .ca for Canadian ones, etc.)

If you wish to send mail to another user on the either on our system or on a remote system, there are several different mail programs you can use: **mail**, **mailx**, **elm**, or **pine**. In general, users seem to prefer pine for its ease of use. **Pine** will be discussed below.

8.2 Pine

At Stony Brook, we have an easy-to-use e-mail program called PINE.

After you log on to your account, you will see the system prompt. (ms.cc.sunysb.edu.15). Type:

pine

Please note: You **cannot use** the mouse in this program. You **must** use the arrow keys, tab key or the specified command keys to move around in this program.

The *first* time you use this program, it will ask if you would like to request a document, **Press N**.

The following menu is then displayed

To move to a menu selection, use either the arrow keys or type in the appropriate letter.

```
?  HELP          - Get help using Pine
C  COMPOSE MESSAGE - Compose and send/post a message
I  FOLDER INDEX  - View messages in current folder

A  ADDRESS BOOK  - Update address book
S  SETUP         - Configure or update Pine
Q  QUIT         - Exit the Pine program

Copyright 1989-1997.  PINE is a trademark of the University of Washington.

Help      PreVCmd      RelNotes
OTHER CMDS [ListFldrs] NextCmd      KBLock
```

8.2.1 Creating/Sending Mail

To type an e-mail message, choose **COMPOSE**, or press **c** The following screen is displayed:

```
1.To      :
2.Cc      :
3.Attachmnt :
4.Subject :
5 -----Message Text -----
```

1. **To:** This field is where you put the person's e-mail address. (To send the same message to several people, separate each complete address with a comma)
2. **Cc:** Carbon copy field The system will send a copy of the letter to any e-mail addresses are entered in this field.
3. **Attachmnt:** This field is for sending a file along with your e-mail message.
4. **Subject:** This field is for a brief description of the message.
5. **Message Text:** Type your message here.

Commands to use when composing a message.

- The bottom portion of the screen presents commands that you can use:



- The caret character (^) means to press the Ctrl key.

To **send** your e-mail message, press **Ctrl** and **x** (^x)

- ^g** Displays a menu of the different composer commands that can be used when creating and editing a mail message. (For the most part, they are similar to, if not identical to, the keystrokes used in PICO.)
- ^c** *Cancel* allows you to abandon creating this message. Pine will next ask you to verify that you want to cancel; type **Y** or **N**.
- ^r** *Read file* allows you to add an external file to your message. (Different from *Attachment* in the header of the mail program. This command actually copies the external file into the message area.)
- ^o** *Postpone* allows you to suspend typing or sending a message until a later time. You may only postpone one message. The next time you use Pine, it will ask:
Continue postponed composition (answering ``No" won't erase it)? y/n/^c [y]:
Press **y** to continue, **n** to start a new message, or **^c** to cancel this message entirely.

Mark, Cut and Paste in Composer (block moves)

In the message composer, if you would like to **move** (cut and paste) a block of text :

- Using the arrow keys, move the cursor to the left of the first word that you would like to edit.
- Next, press: **Ctrl, shift-6** (^^) This enables you to select your text.
- To select the text, move the arrow keys over or down, and you will notice a white block going over the text.
To **cut** the text, press **^k**
- Move the cursor to where you would like to insert the text.
To **paste (uncut)** the text, press **^u**

Note: If you use the cut key sequence before you specify a mark, it will cut one line of text.

Attaching a file to a mail message

There are two ways to attach a file to a mail message:

1. using the **^r ReadFile** from the menu **OR**
2. using *Attachmnt* in the message header

Note that:

Read File will only work with text only (ASCII) files.

Attach File can be used to send binary files (such as documents saved as MS word files or Word Perfect) only if the person who is receiving the message has a program capable of translating “MIME” format.

1. Using the **^r** *ReadFile* menu option at the bottom of the screen reads an ASCII file (text only) directly into the mail message. Pine will ask you for the name of the file. This is the easiest way to send a file to someone.

2. Using *Attachmnt* in the message header.

- (a) Type the file name on the line. This attaches it as a binary file.
- (b) When it arrives at its destination, it will advise the recipient the following:

[Cannot display this part]
[Use the V command to save in a file]

8.2.2 Reading Your Mail

From the Main Menu select either:

1. **I Folder Index** or press **enter**. This places you in another screen where the different folders are listed.
2. **INBOX** should be highlighted; if not, use the arrow keys or tab key to move to this folder.
3. When selected, press **enter**. This will display a list of mail messages.
4. Move the cursor to the message you wish to read and press **enter**

Note: The menu at the bottom of the screen displays the key sequences used to reply to a message, delete a message, or scroll to other pages of a long message.

8.2.3 Receiving Attachments

If you receive a message that says it has attachments, then:

1. Press **v** to choose which attachment you would like to see
2. To **choose an attachment**, use the arrow keys (down or up).
3. To **save an attachment** in your account, press **s**
4. Pine gives the file a default file name, similar to the one it had when it was sent, and then saves it to a directory on your account once you press **enter**

If you would like to save the attachment to your disk, please see the **FTP** section.

8.2.4 Replying to an e-mail message.

1. To reply to a message, press **r**
This will prepare you to send a response to a message that you have received.
2. The computer will ask if you would like to *Include Original Message in reply?* Answer **y** or **n**
3. Next, the Compose Message screen appears. Notice that the **To:** and **Subject:** fields are already completed.

4. When you have finished typing your reply, use **^x** to send the message.

8.2.5 Exporting a Mail Message to a File in your Home Directory

While viewing a mail message, you can extract the message and place it in a file:

1. Press **e** for *export*. (This option is displayed on the menu at the bottom of the screen when you enter **o** for *OTHER CMDS*.)
2. You will be asked for the name of the file (in your home directory) where you want to place it. If the file already exists, Pine will ask whether you wish to *append* to that file. Note that exporting a message to a file does not delete it from your mail folder.
3. If you would like to *save* the message to your disk, see **ftp** on the last page of this document.
4. While viewing a mail message, you can *save* the message to your mail folder by typing **s**
5. to save. You will be asked to name the message, and then it will be saved.

8.2.6 Viewing Messages You Have Sent

Messages that you have sent are saved in a folder called *sent-mail* which can be found by:

1. Pressing **I Folder List**
2. Press **Enter** on [*Select Here to See Expanded List*].
3. With the arrow keys, move to the *sent-mail* folder you would like to view.
4. Press **Enter** to view the messages in that folder.

8.2.7 Printing Your E-mail

To print your e-mail, you must do the following.

From the *Main Menu*:

1. Press **s** for Setup
2. Press **p** for Printer
3. Move with the arrow keys to: Standard UNIX Print command
4. The print command should list: **lp -dliblaser1**
5. Press Enter to select it.
6. Press E to exit config
7. Commit Changes: Press **Y**
8. Go back to the message that you want to print.
9. Press **y**
10. You will be asked if you would like to print; press **y**

8.2.8 Address Book

The Address Book is a convenient tool that allows you to store frequently used email addresses and mailing lists. Selecting **a** from the *Main Menu* to access the address book.

Choose **A** to Add New Entry to the address book. Your screen will look like:

```
PINE 3.96  ADDRESS BOOK (Edit)  Folder: INBOX  0 Messages
Nickname : 
Fullname : 
Fcc      : 
Comment  : 
Addresses: 

Fill in the fields just like you would in the composer.
To form a list, just enter multiple comma-separated addresses.
To add to a list, use the View/Edit cmd instead of the AddNew cmd.
It is ok to leave fields blank. Press ^X to save the new entry.

^G Get Help  ^X eXit/Save  ^R RichView  ^Y PrvPg/Top
^C Cancel    ^V NxtPg/End  ^T To AddrBk
```

To Add a Name:

Type in the Nick Name, Full Name, and e-mail address. Press Ctrl X to save it

To Create a List:

Type in the Nick Name of the List, a full name. In the Addresses: type each e-mail address separated by a comma.

To Edit an Address:

Move to that address and press the Enter key. Make the appropriate changes and Press CTRL X when you are done.

8.2.9 Spamming/Mass Mailing

These practices are not permitted on our system. If you do send mass mailings and people report you, you risk having your account revoked. To find out more about mass mailing/spamming, please refer to <http://www.sinc.sunysb.edu/UNIX/SPAM>

8.2.10 Conserving Disk Space

In order to stay within your disk quota, it is important to weed out unnecessary messages from your folders. Go to the *I Folder Index* and, once inside a folder, press **d** to *delete* any unnecessary mail message. Note: You can *undelete* a message by pressing **u**.

At the end of every month, Pine attempts to move *sent-mail* for that month into a file called *sent-mail-month-year*. It will ask whether you want to do this. Type y or n

To find out your disk usage and quotas, you can use the **quota -v** command.

8.3 Internet: FTP, TELNET

The Internet is comprised of networks which use the Transmission Control Protocol/Internet Protocol or TCP/IP. This protocol includes SMTP, the Simple Mail Transfer Program, Telnet and FTP. SMTP is grafted onto our local mail system and allows users to use all of the mail commands discussed above.

8.3.1 TELNET

Telnet allows you to remotely log into another computer system. The format of the command is:

telnet hostname

where **hostname** is the Internet name or number of the host computer. For example, to use telnet to log on to CS UG system, the command is

telnet public.ug.cs.sunysb.edu

Once you are connected to the host computer, a login screen will appear and you can login using your valid userid and password. To log off of TELNET, use the **logout** command. This will return you to the system from which you issued the TELNET command.

For further information and commands, type

man telnet

8.3.2 FTP

FTP, the File Transfer Program, provides for file transfer between two host computers on the Internet. This also involves logging in to remote sites, but only for the purpose of either "putting" files or "getting" files.

The command is:

ftp hostname

where **hostname** is either the name or number associated with the computer you are going to.

Once the connection is made, you will be asked to enter your userid and password.

On the Internet there are a myriad of sites with public domain software, documentation, and other information which you can retrieve by what is known as "anonymous ftp". To do this you ftp to the host computer. At the login prompt you supply the user name "anonymous" and a password - your userid and Internet address is recommended for record keeping purposes on the other end.

- The commands available for FTP include such operations as displaying a directory, changing directories, getting files or putting files. To list possible commands type help or ? .
- Use either **ls** or **dir** to display the contents of a directory.
- Files and directories are listed in the format of the system they reside on. Files on a VAX/VMS or a VM file system will appear somewhat differently than Unix files.
- To get a file from the remote machine or archive, the get command is used. The format of the command is:

get *sourcefile destinationfile*

- To transfer all files in the directory or all files that match a particular string the mget command is used. For example,

mget *x**

will get all files beginning with x.

- To transfer a file from your home machine to the remote machine, the put command is used:
put *source.file destination.file*
- To transfer binary files, specify **binary** before issuing the **get** or **put** command
- Use the command **quit** to terminate an ftp session.

With both the **put** and the **get** commands, if no destination name is specified, the destination file name is the same as its origin.

8.3.3 The World Wide Web

The Lynx browser allows you to view the World Wide Web on the Unix system *without* the graphics. To use lynx, the command is

lynx

You can put web pages in your `~/public_html` subdirectory and they will be visible over the web at <http://ms.cc.sunysb.edu/~username> where username is your login id.

9 Getting Help

9.1 The MAN Command

The man command will give you a listing of the manual section concerning the specified command. The format looks like this:

man *command*

where *command* is the name of the command you **want more information** about.

man -k *keyword*

will print a one line synopsis of each manual section whose table of contents listing includes the keyword.

For more on the man command, try:

```
ms.cc.sunysb.edu.3 man man
```

This guide was originally prepared by Instructional Computing and has been modified for the Faculty/Staff Unix mail server.

You can call client support at 632-9800 during regular hours for help.