UNIX Commands Guide

This is a guide to some basic UNIX commands. It is intended for people with little or no experience with UNIX. It will teach you how to manipulate files and directories, communicate with other UNIX users, as well as some other useful and fun stuff that UNIX can do. Some of the commands are local to Brown, while others are supported on all UNIX systems.

```
*note: The man page links are only available from within the Brown cs department. If
you are local and still getting errors, use http://localhost/whatever instead of
http://www.cs.brown.edu/whatever.
```


What is a Shell?

A shell is basically the user interface in UNIX. It is a program that allows the system to understand your commands. The shell has three main uses:

- **Interactive Use**
  When used interactively, a shell waits for you to type a command at the prompt
- **Customizing your UNIX session**
  The shell lets you define environment variables that control the behavior of your UNIX session, such as setting the appearance of the screen.
- **Programming**
  The UNIX shell provides a number of commands that can be used to create programs called **shell scripts**. Scripts are useful for executing a series of individual commands. They can also execute commands in loops (repeatedly) or conditionally (if-else). There are three main types of shells: the **Bourne** shell, the **Korn** shell, and the **C** shell. Most people at Brown use the **tcsh**, which is an enhanced C shell.

xterm

The command xterm starts up a new shell. If you are using this command from a shell, however, you will want to put it in the background using the '&' operator. Otherwise you will be running the shell from the shell in which it was executed from and won't be able to use that shell anymore. A useful option for the xterm is -e which allows you to specify the program to be run in the xterm window. If I want a login shell, for example, I type `xterm -e /bin/login &` . This provides a new shell with a login prompt which allows someone to login to the computer.

setenv

In UNIX, the system has to have a way to store certain settings that the shell and various programs use. This information is stored in "environment variables" each of which has a name (like DISPLAY, PATH, TIMEZONE, HOME) and a value. There is no standard UNIX command to set these environment variables, but the C-Shell (which you are probably using a form of) has a built-in command called setenv which you can use to set one of these variables.

printenv

The C-Shell command (again, not a real UNIX command) printenv will show you the contents of an environment variable.

Some Useful Operators for the csh or tcsh:

- * match any characters in a filename (wild-card)
- | take output from one command and use it as input for a second command (pipe)
- > redirect output (over-writes)
- >> append output
- & run process in the background

back to the top

Simple Commands

man

This command displays information from online documentation. It will give you a manual page for virtually any command (or programming function) available on the system, which describes what the command does and how it is used. Each command in this document is linked to its man page for easy reference.

finger

Locate any user using the finger command. This command provides useful user information such as login name, the person's real name, what computers the user is currently logged in on, when the users last login was, and information about when the user last received and checked e-mail. It also includes the information listed in the files **.plan** and **.project**. These files will be explained later, but here is an example of what you would see if you typed in a shell finger

dmb. Alternatively, you could use finger with a person's first or last name, and it will find all the login names for that name. If fingering a username, you can abbreviate this command to f [username]. You can also finger at a machine and get back who is logged into that machine and from where. Typing `finger @cslab0a` will return all the users logged on to cslab0a. Typing finger by itself will tell you who is logged on to the machine you are on.

back to the top

Files and Directories

cd

The cd command lets you navigate the filesystem. The UNIX filesystem is organized as a heirarchy where each level is separated by a "/". The "root" directory is simply called /. So, if I am in / I can type `cd u/ck` and I will be in /u/ck, ck's home directory, since it will change directories relative to my current path. If I codefix the directory name with a /, it will jump to an absolute location so typing `cd /u/ljc` from /u/ck will put you into /u/ljc. Also, the directory ".." is one level up in the heirarchy, so from /u/adt typing `cd ..` will move you up one directory into /u.

ls

This command lists all directories and files contained inside the current directory, when typed alone. The ls commands has a number of options that let you display a variety of information about what is contained in a directory. Some useful options include:

-a
: Lists all the files in a directory, including the normally hidden . files.

-F
: Flag filenames by adding **/** to directories, ***** to executable files, and **@** to symbolic links.

-l
: The long format listing includes permissions, owner, size and modification time.

-R
: lists subdirectories as well as directories recursively.

So, for example, if i want to flag filenames when using the ls command, I can type ls -F.

more
less

These commands allow you to display the contents of a file, which is useful if you want to scroll through a file that is too long to fit within one screen. Two simple ways to do this are to use the more and less commands. These commands are similar in that you can read, but not edit a file using this command.

Using more, press the return key to view the next line of text, the spacebar to read the next page of text, **q** to quit, and / to search.

You can use the same commands in less, but in addition you can use the arrow keys to navigate by line, and **u** to move up a page. A useful option is **-N**, which numbers the line in the files.

To use more or less, type more or less and the name of the file you want to read. Type any option you want to use before the filename. As with any command in this guide, for more options or more information about a command, see the man pages.

mv

This allows you to change the name of a file, or move files or directories around. To create a new file in UNIX we simply try to open a file that does not exist in a text editor, described below. UNIX will then create a new file with this name. If we want to change the name of a file or move it to a directory other than the one it was created in, we use the mv command. To use this command, we type mv then the name or names of the files we want to move and then the directory or new file name we want to move to. For example, if we have a file named foo and we want to rename it foo.html, we type `mv foo foo.html`. Now if we want to move foo.html to dmb's home directory, we type `mv foo.html /u/dmb`.

cp

To copy a file into a file with a different name, or into another directory, we use the cp command. Above, we moved foo to foo.html, but if we want to copy foo into a file called foo.html, but don't want to remove foo, we type `cp foo foo.html`. Likewise, we can copy a file into another directory: `cp foo.html /u/dmb .` If we want to copy a file from a different directory into a current directory, we can use a '.' to mean the current directory. So to copy foo.html from dmb's home directory into the current directory, type `cp /u/dmb/foo.html .`

rm

To remove a file we no longer need, we use the rm command. This command has a number of useful options:

-i

This makes rm interactive, and asks the user if it wants to rm the file or files named. The user responds with **y** or **n**.

-r

In UNIX, an ***** can be used as a wild card. Typing `rm *.html` removes all files whose name ends in ".html". Likewise, typing `rm *` removes all the contents of a directory. However, to recursively remove the contents of all subdirectories, you must use the -r option. The command can also be used to remove a not-empty directory and its contents by specifying a directory name. Be warned! This is a dangerous command to use, because if the directory contains files, you may be unknowingly removing them when you remove the entire directory.

mkdir

To create a new directory, use the mkdir command. This will create a new directory with the specified name in the current directory. So to create the directory JavaWork in the current directory, we type `mkdir JavaWork`.

rmdir

To remove an empty directory, or to remove the name of a directory, but not its contents type rmdir and the directory name. For a directory that is not empty, the rm -r command can be

used.

pwd

This commands prints out the full pathname of the current directory. So if I type `pwd` while I am in ljc's home directory, UNIX responds `/home/ljc` .

chmod

The chmod command is a little more complicated than some of the other commands, but extremely important. Everything you create has permissions for the people who can read, write to, or execute the file or directory you create. The command ls -l lists the permissions of a file. If you want to change who can read a file you use the command chmod. You must then specify who you are changing permission for and what permission you are changing it to. Who and the permission are connected by either a '+' (to add a permission), a '-' (to remove a permission), or an '=' (to assign a permission and remove other permissions). Who can be either 'u' for the user, 'g' for the group the user belongs to (all users belong to at least one group), 'o' for everyone else, and 'a' all. The common permissions include r (read), w (write), and x (execute - for directories and executable files). Here are some examples of how to change permissions on the file foo.html:

Add read permission for all:

```
chmod a+r foo.html
```

Add read and write permissions for the group:

```
chmod g+rw foo.html
```

remove execute permission for others:

```
chmod o-x foo.html
```

chgrp

Often users belong to more than one group. To change the group that has permission on a certain file, use the chgrp command. To use this command, type chgrp, the name of the new group you want to assign the file to, and the file name. For example, to assign foo.html to the cs015 group, type `chgrp cs015 foo.html`.

grep

The grep command allows you to search one or more files for lines that match an expression. Two useful options of this command are:

-i

Ignores uppercase and lowercase distinctions

-n

Prints lines and their line numbers.

To use the command, type grep, any options you want, the expression to search for and the files to search in. For example to search for the word UNIX in foo.html, type `grep UNIX foo.html` .

cat

The command cat simply takes a file or multiple files and outputs them. While it may seem simple, it allows you to perform many powerful operations. UNIX has notions of standard

input and standard output. Your screen is typically standard output, so when you run a command, the output appears on your screen. However, you can redirect standard output, so it goes to a file. The > operator will redirect standard output to a file, and the >> operator will append standard output to a file. So, `cat a b > foo.html` will create a file, foo.html that consists of the contents of a followed by the contents of b. If we then type `cat c >> foo.html` foo.html will contain the contents of a then b then c. Another redirection operator, called pipe (the '|' key), takes the output of one operation and uses it as the input to another operation. So `cat foo.html | mail dmb` will take the output from foo. html and use it as the input for and e-mail to dmb.

du

This command prints the disk usage (without the -k option, it prints out the number of 512-byte blocks used). The -k option writes the files sizes in units of 1024 bytes, rather than 512. If you name a directory, it will print out the size of that directory and all sub-directories, or the default is the home directory.

gzip/gunzip

The gzip command reduces the size of the named files by compressing it. To read a gzipped file, you must first uncompress it by using the gunzip command.

compress/uncompress

Basically the same as gzip, compress reduces the size of the named files. To read the files, use uncompress.

touch

The touch command updates the access and modification times of the named files to the current time and date. The command is useful because some commands rely on a file's access and modification times.

head
tail

The head command prints only the first ten lines of the named file and the tail command prints only the last ten lines. If you want head or tail to print out a different number of lines, you can specify the length as `-<the number of lines>` . For example, `head -20` will print out the first 20 lines of a file.

sort

To sort the lines of a file, typically in alphabetical or numerical order, use the sort command. One useful option is **-r**, sort in reverse order.

ln

The ln command creates links for files. For example, if you type `ln foo.html foo` , foo becomes a psuedonym for foo.html. If foo is an already existing file, it is overwritten. If foo

is a directory, however, a link named foo.html is created in foo. The **-s** option creates a symbolic link, which lets you link across file systems and also lets you see the name of the link when you run ls -l.

back to the top

Communication Commands

zwrite

The command zwrite sends a message to another user by using the zephyr notification service, by displaying a window with a message on the recipient's screen. Zephyr is not available on all UNIX systems, but zwrite is a commonly used command at Brown.

zwgc

Zephyr Windgowgram Client program, or zwgc, is the main zephyr client used for zwrites. To receive zwrites, a user must be running zwgc. This can be done by using the command zwgc, or if you want to receive zwrites in a shell, type `zwgc -ttymode` .

write

The write command is used when zwrite is not available. It allows you to write to the shell (usually the console) of another user. In order to write to a user, however, you must first remotely log on (rsh) to their machine. To end a message, type control-d on a line by itself.

talk

The command talk creates a two-way, screen-oriented communication program. It allows users to type simultaneously, with their output displayed in separate regions of the screen. To send a talk request, type talk and then <user name>@<their machine>. So, for example, to send a talk request to dmb, who is logged on to cslab0a, type `talk dmb@cslab0a` . The recipient of the request is then prompted for a response.

mesg

The mesg command changes the ability of others to send write or zwrite messages or talk requests to a user. Typing `mesg n` or `mesg -n` forbids messages, while `mesg y` or `mesg -y` allows messages.

mail

The mail command is a quick and easy way to send an email to someone. Just type mail and the address of the person you want to mail. You will then be prompted for a subject and any cc's. Then just type your message and control-d on a line by itself to send the mail.

pine

There are a number of ways to read e-mail in UNIX. One option is pine. It a screen-oriented tool with limited functions for message-handling. You can also use pine to send a quick email without opening your mail directory and inbox (which pine does when you type the pine

alone) by typing pine followed by an e-mail address.

tin

There are also a number of options to use to read newsgroups. One popular news reader is tin, which like pine is screen-oriented with functions to read and compose news. When you start up tin, it will show a list of the newsgroups found in you .newsrc file in your home directory. You can also specify a newsgroup for tin to open by typing tin and then the name of the newsgroup.

back to the top

Process Commands

Everything you run from your account is called a process. So when you are running a shell, netscape and emacs, you are running three processes. There are a number of useful commands for dealing with processes.

ps

Without specifying options, ps lists all the active processes running from a given shell. A more useful way of using the ps command is to use the **-u** option, which you specify a user, and this lists **all** the processes that user is running. Each process listed has a 'PID' (process id) number associated with it. This is an important number!

kill

Use kill with a PID number to terminate a process. Use the ps command (above) to find out the PID for a process. Then type `kill <PID>` to stop the process.

nice

All commands that you run have a priority. To give a command a lower priority, use nice with the command name. You can specify how "nice" you want to be by using the **-n** option with a number from 1 to 19. A higher number means a lower priority. This command is often used if you are running something on a machine and want to give others priority when they are running processes that are more important.

top

The command top displays the top 15 processes on the system and periodically updates this information. If you specify a number, then that number processes will be displayed instead of the default.

fg

If a process is running in the background, fg runs it in the foreground.

bg

If a process is running in the foreground, bg runs it in the background.

back to the top

Text Editors

There are a number of text editors available when using the UNIX operating system. Some work from with in the shell that you start it in. Start up these editors by using their name and then the name of the file you wish to edit. A new file is indicated by a name that does not already exist in the directory you are in. Other editors start up in their own windows when you start them from a shell. Read the man pages for more information about how to use the different text editors.

Text Editors in a Shell:

- emacs
- pico
- vi

Text Editor Applications:

- xemacs
- emacs-19

back to the top

Using Printers

In UNIX, you can print a file without opening it in a text editor using various commands. When you print to one of Brown's printers, your file will be printed with a "banner" sheet that identifies who the print job is for. The CS department has various locations to print to. The printer for undergraduates is cis. The department also has three printers for staff, namely ps1, ps2 and ps3. The 'ps' is short for POSTSCRIPT, which is what our printers use to understand and print the files.

lpr

The command lpr prints the named files to either the user's default printer (for most undergraduates, cis) or to the named printer using the -P option. For example, to print foo.html to cis (and cis is not the default printer), type `lpr -Pcis foo.html .`

lpq

To print out the printer queue to check the status of a print job, use the lpq command. To specify a printer, use the -P option as used above. So, if I want to print out the queue for cis, I type `lpq -Pcis .`

lprm

The command lprm allows you to remove a job from the printer queue. To use the command, you can specify the job number by using the lpq command to determine what the number is. So if I find out from lpq that the job I want to remove is 787, I can type `lprm 787` to remove the job from the printer queue. Using the command alone removes the first of your jobs in the printer queue.

pr

The pr command is most commonly used at brown with the **-f** option to print multiply files without banner sheets between the files. For example, if I want to print foo1, foo2 and foo3, I type `pr -f foo1 foo2 foo3 | lpr` . The pipe (|) operation takes the output from the first command and uses it as the input for the second command. The pr command is basically used to apply options to, and then sent to the printer by piping the output to the lpr command. For more options, see the man pages.

enscript

The printer at Brown reads mainly POSTSCRIPT files. The enscript command converts text files to POSTSCRIPT format for printing.

back to the top

Logging On to Machines

Logging on to a machine on console (at the terminal) is simple. But what if you want to use someone else's machine to log into your account, or log into another machine remotely. This is useful when you are running a number of processes on your machine and it would be faster to run some on another machine.

su

The su command allows you to become the "super-user" or another user in a shell. In order to use su, the password of the super-user must be supplied when prompted. If the first argument to su is a dash (-), for example `su - dmb` , the computer will go through the entire login sequence, where as without the dash, the environment will remain unchanged.

rsh

The rsh command creates a remote shell by connecting to the specified computer. If you specify a command, the command will be executed on the specified computer. A useful option is -l, which allows you to log in as a specified user. So, typing `rsh cslab0a -l dmb` will allow dmb to remotely log on to cslab0a from the computer she is on. She will have to type her password at the prompt for this to be successful, however. If dmb is already logged on the computer she is trying to rsh another machine from, she does not have to use the -l option. She can simply type `rsh cslab0a` .

rlogin

The command rlogin creates a remote login session from your terminal to the machine specified. Once again, you can specify a user name using the -l option. For dmb to remotely log on to another machine from someone else's account she would type `rlogin cslab0a -l dmb` and type her password at the prompt. To remotely log on from her own account to another machine she would simply need to type `rlogin cslab0a` .

back to the top

X server and DISPLAY

When you are logged onto console on a machine, you own the X server, or what can be displayed on your screen. In order for other users or other machines to use your DISPLAY, you must give them permission, and in order for a user to use the DISPLAY of another machine, he or she must set the DISPLAY to be that computer.

xhost

The xhost command is used to add and delete computers or users to the list allowed to make connections to the x server, which enables something to appear on the screen. Use `xhost + <name>` to add permission, or `xhost - <name>` to remove a permission. Using xhost + without a name allows anyone to use your xserver. The command `xhost + localhost` can be used to allow any user logged on to your machine to use the display.

setenv DISPLAY

The command setenv is used to set the various environment variables. A common variable that is often changed is the 'DISPLAY' variable, which determines where something will be displayed. You can set DISPLAY to any computer that has given your computer, or a user, permission to use the x server. So, for example, if you are a super-user on a computer and you want to set DISPLAY to localhost, you would type `setenv DISPLAY localhost:0`

back to the top

Miscellaneous Commands

yppasswd

Everyone has a password to get into their account for security purposes. You can change your network password using the yppasswd command. After typing the command, it will prompt you for your old password (for security reasons) and then ask you the new password twice to prevent mistakes.

cal

The cal command prints a calendar to standard output. You can specify a month (using the number for the month) and a year, or just a year to print a calendar for an entire year. If you do not specify anything, a calendar for the current month is printed. So to print the calendar for December 1999, type `cal 12 1999 .`

date

The date command simply returns the current date and time.

whoami

The whoami command displays the login name of the current effective user. If you have used su to adopt another user, whoami wil report the login name associated with that user ID. For a command similar to whoami, but with more options, see the man pages for who.

which

The command which takes a list of specified names and looks for what would be executed had these names been given as commands. UNIX allows you to alias commands to a shorter name in your .alias file. If you have aliased a command, the which expands the name. For example, if I have the command 'zwrite' aliased to 'z' in my .alias file, and I type `which z` , the command returns `z: aliased to zwrite .`

loc

The command loc returns the address of a user. This command is extremely useful when used with the -g option, which locates all users in the specified group. For example, if i wanted to locate all of the cs015 tas, I would type `loc -g cs015ta` .

groups

The groups command will list all of the groups a specified user belongs to. If a user is not specified, the groups for the current user will be listed.

grplist

The grplist command will list all the users in a specified group.

telnet

To create a connection to a remote system, use telnet.

keyinit

In order to telnet into the Brown Computer Science department's telnet host, you must use the S/Key authentication system. The keyinit command is used to initialize the system or change your password so you can use S/Key one-time passwords to login. This lets you avoid sending a real password over the network.

lw, sunfree, sunlist

These are commands used for the sun lab at Brown University. They provide information in various formats about who is logged in on console on what computer in the sunlab. Sorry, there are no man pages for these commands.

xstartfm

Framemaker is a word processing and desktop publishing package. It is often used to make lecture slides in the Brown cs department. To start framemaker, use the xstartfm command. It is often helpful to run framemaker in the background by using the & operator.

xcolorsel

xcolorsel displays the colors available (usually the contents of the rgb.txt file). Each color will consist of three intergers (usually 0-255) representing the RGB values and a nickname for the color. You may want to run this command in the background using the & operator.

back to the top

Fun Stuff and Games

There are lots of fun things and games you can use in UNIX. Most of the ones listed below are local to Brown University. Try each of these commands. Check the man pages or the links below if you have trouble, but note, some of the commands do not have man pages. Have fun!

banner
figlet
WhatsForDinner
food
dilbert
forecast
fortune
say
xteddy
xdeady
BattleTris
nethack
netris
xbill
xblast
xboing
xroach

back to the top

HOME