

Basic UNIX for CMS Users

[File Attributes](#)

[File Commands](#)

[Process Commands](#)

[Miscellaneous but useful commands](#)

[Examples](#)

*Beware: almost everything about UNIX is case-sensitive, including filenames and commands. Also, if the **BackSpace** key doesn't backspace, try the **Delete** key, **Control-BackSpace** or **Shift-BackSpace**.*

Files

Everything in UNIX is a "file" -- real files, directories, device drivers, and so forth. This makes it easy to combine files and programs in many different and new ways.

Directories are special files that hold the names of other files or other directories. In this way, the UNIX file system looks like a hierarchical tree, similar to DOS. Note that directories are separated by a forward slash (/) not a backslash (\) as in DOS.

File Attributes

<i>file name</i>	A file name is a string of characters. UNIX does not give special relevance to periods or other characters, although some programs expect specific types of filenames.
<i>full path</i>	The full path of a file specifies all directories, such as <code>/usr/local/bin/foo.bar</code> . The filename <code>foo.bar</code> can be used for this file when <code>/usr/local/bin</code> is the current directory.
<i>file permissions</i>	Each file has an associated owner and group. The owner is a logon account, and the group is a possibly empty group of logon accounts. The read, write, and execute permissions (which can be set by the owner) can be different for the owner (user), group, and public (other). Permissions apply to directories as well as files.

Enter `ls -l` to find the permissions on the files and subdirectories in the current directory. This returns lines that look something like the following:

```
drwx----- 2 bobg comp 512 Jun 7 09:49 mydir
-rwxr-xr-x 1 bobg comp 321 May 30 14:36 myscript
```

Each line describes one file. From left to right: the permissions, the number of links, the owner, the group owner, the size in bytes, the date and time of the last modification, and the file's name.

The first character of the permissions tells what kind of "file" it is; `d` for directory, hyphen (-) for regular file. The remaining nine characters are three triplets. The triplets give the read, write, and execute permissions for that file or directory for that file for, respectively, the file's owner, its group owner, and for the public. The `r` (read), `w` (write), and `x` (execute), indicate the presence of read, write and execute permissions; the hyphen (-) indicates their absence. For directories: `r` permission allows you to list the files in the directory, `w` permission allows you to create or remove files from the directory, and `x` permission allows you to `cd` to the directory (enter it).

Thus, in this example, both files are owned by `bobg` and have the group `comp` as group owner. `mydir` is a subdirectory in which only `bobg` can read, write, and execute; and `myscript` is a file which `bobg` can read, write, execute, while everyone else, including those in group `comp`, can read and execute it, but not write or delete it.

The `chmod` command changes file permissions. For example `chmod u+x file` adds execute permission to `file` for the owner (user); use `u-x` to remove execute permission for the owner. `u` indicates the file's owner (user), `g` the owner's group, `o` the public (other), or `a` for all three; and `r` is read permission, `w` is write permission, and `x` is execute permission.

File Commands:

CMS Command	UNIX Command	Function
listfile filelist	ls ls -al	gives a list of filenames in the current directory displays a long list, including system files, with date, time, size (and Unix permissions)
rename	mv	rename a file: mv oldfile newfile or: mv oldfile newdir
copyfile	cp	Copy a file: cp oldfile newfile or: cp oldfile newdir
acf	chmod	change permissions; see "File Attributes" above for examples
erase	rm	remove a file
access	cd	change directories
print	lpr	print a file: lpr -P printer-name file For more information, see <i>Using UNIX: Printing</i> http://www.uic.edu/depts/adn/infwww/inform/unixprt.html
id	pwd	"print" working directory; returns the current directory
q disk/limits	quota	to see how much disk space you've used
create dir	mkdir	create a new directory
erase	rmdir	remove a directory
vmarch	gzip	compress a file into one with a .gz extension; gunzip reverses the process
package	tar	package a group of files into one file for moving or archiving; also extracts tar files

Processes

The shell (Korn shell, or ksh, by default on the ADN UNIX systems) is a command line interpreter (as well as programming language). The shell reads the command line, interprets any special characters, and then runs the specified command, usually as a new process (which itself can spawn new processes).

Process Commands:

CMS Command	UNIX Command	Function
	ps	list your current processes; note the PID (process id); you need the PID to change attributes of the processes
force	kill	send a signal to a process; kill -9 pid terminates the process with process id <i>pid</i>
batch	&	run a command in the background; put the & at the end of your command
cp disc	nohup	put this command at the beginning to make a background process continue after you break the logon connection
hi, hx	Ctrl-c	get command prompt back
/*, <null>	Ctrl-d	end of input; same as logout when in shell

Miscellaneous but useful commands:

CMS Command	UNIX Command	Function
compare	diff	compare two files
listfile	find	recursively search for files
grep, search	grep	search a bunch of files for a string
logoff	logout	end your UNIX session
help	man	look up a manual page; man ls will tell about the options to the ls command, for example; main online documentation for UNIX
	more	make the output stop after each screenful; Spacebar displays next screenful, Ctrl-b displays the previous screenful, q quits
altpasswd	passwd	change your password; passwords on the ADN UNIX systems may be up to 8 characters long and must have at least three letters and two non-letters
q disk/limits	quota	check your own disk space usage and quota
sort	sort	sort the lines of a file
pipe	cmd cmd	connect program output to input

Examples:

who am i	To do the obvious; just who lists everyone who is logged on; likewise hostname tells you the name of the machine you're using.
pwd	To display current directory.
mkdir foo	To make a new directory called <i>foo</i> .
cd	To change to your home directory from anywhere.
cd bin	To change to the <i>bin</i> directory under the current directory.
cd /usr/local/bin	To change to the <i>/usr/local/bin</i> directory, regardless of what your current directory is; much of the public software is stored in the subdirectories of the <i>usr</i> directory
ls -l -a -R more	To get a list of all files in a directory. -l to get a long listing, with permissions; -a to include files with filenames beginning with a period (.); and -R (note the R is uppercase) to also list the files in all subdirectories of the current directory. And since this is likely to be a long listing, it's "piped" into more so the display pauses at the end of each screen.
ls -l a*.c	To get a long listing of all files with filenames starting with <i>a</i> and ending in <i>.c</i> .
more .profile	To display your <i>.profile</i> file, when your current directory is your home directory. For more information about <i>.profile</i> and <i>.env</i> files, see <i>Customizing UNIX</i> at: http://www.uic.edu/depts/adn/infwww/inform/unixcust.html
more ~/.profile	To display your <i>.profile</i> file when your current directory is not your home directory.
cp ../foobar .	To copy the file <i>foobar</i> from the parent of the current directory into the current directory..
rm -i ??	To remove (erase) all files in the current directory with exactly 2 characters in the filename, verifying each erase with a y or n .

<code>mv foobar ../newdir/fubar</code>	To rename the file <i>foobar</i> to <i>fubar</i> and place it in the directory <i>newdir</i> , that is a child of the parent directory of your current directory.
<code>chmod u+x foobar</code>	To make the file <i>foobar</i> executable by its owner.
<code>chmod -R a+r *</code>	To make all files the current directory all subdirectories (-R) readable by everyone.
<code>man ls</code>	To get information on the command ls and its flags.
<code>ps -e more</code>	To get a list all processes piped into more; without the -e only the processes associated with your session are listed
<code>nohup cmd < foo.in >> foo.out &</code>	To run the command <i>cmd</i> , taking input from <i>foo.in</i> , appending output to <i>foo.out</i> , and run the command in the background so that you can log off and have the command continue to run. (Without the >> , nohup puts output in the file nohup.out .)
<code>grep double *.c</code>	To search all the C source files for the string "double".
<code>ls -l grep ^d wc -l</code>	To find the number of subdirectories in the current directory.
<code>find ~ -name foobar -print</code>	To search for all files named <i>foobar</i> in your home directory tree.
<code>uncompress foo.Z</code>	To uncompress the compressed file <i>foo.Z</i> in the current directory.
<code>tar -xf foo.tar</code>	To extract the contents of the file <i>foo.tar</i> in the current directory.
<code>diff foobar fubar diff.bar</code>	To find the differences between <i>foobar</i> and <i>fubar</i> and record the differences in a file called <i>diff.bar</i> .
<code>pine consult@uic.edu</code>	To send a note to <i>consult@uic.edu</i> ; enter just: pine for a complete mail management menu.

Possible .profile commands

<code>set -o ignoreeof</code>	Logout only with exit not with <i>Ctrl-d</i>
<code>set -o vi</code>	Allow vi-style command history recall similar to CMS F6
<code>export PATH=list of directories</code>	Change PATH for executable search
<code>alias mycmd=unixcmd(s)</code>	Set up alternate command name
<code>set -o vi</code>	Allow vi-style backup for commands similar to CMS F6
<code>rm -f .pine-debug* rm -f core</code>	Remove annoying files that get left
<code>export TERM=termtyp</code>	Set terminal type, e.g. vt220
<code>PS1="\\$PWD"</code>	Set command line prompt string to current directory
<code>tset -e^?</code>	Set delete key as erase
<code>alias rm='rm -i' alias cp='cp -i' alias mv='mv -i'</code>	Set up "safe" commands (they inquire, i.e. ask for a confirmation)