cat

Allows you to look, modify or combine a file.

Syntax
cat filename [-n] [-b] [-u] [-s] [-v]

filename	The name of the file or files that you wish to look at or perform tasks on.
-n	Precede each line output with its line number.
-b	Number the lines, as -n, but omit the line numbers from blank lines.
-u	The output is not buffered. (The default is buffered output.)
-s	cat is silent non-existent files.
-v	Non-printing characters (with the exception of tabs, new-lines and form-feeds) are printed visibly
-e	A $ character will be printed at the end of each line (prior to the new-line).
-t	Tabs will be printed as ^I's and formfeeds to be printed as ^L's.

*If the -v is used -e and -t will be ignored.
Examples
cat file1.txt file2.txt > file3.txt
Reads file1.txt and file2.txt and combines those files to make file3.txt.

cd [or] chdir

Changes the directory.

Syntax
cd [directory] [or] chdir [directory]

Directory	Name of the directory user wishes to enter.
cd ..	Used to go back one directory on the majority of all Unix shells. It is important that the space be between the cd and the ..
cd -	When in a Korn shell to get back one directory used to go back one directory.

Examples
cd hope
The above example would go into the hope directory if it exists.
cd ../home/users/computerhope
The above example would go back one directory and then go into the home/users/computerhope directory.
cd ../../
Next, the above example would go back two directories.
cd
Finally, typing just cd alone will move you into the home directory. If you're familiar with MS-DOS and how typing cd alone prints the working directory. Linux and Unix users can print the working directory by using the pwd command.

chmod

Changes the permission of a file.

Syntax

chmod [OPTION]... MODE[,MODE]... FILE...
chmod [OPTION]... OCTAL-MODE FILE...
chmod [OPTION]... --reference=RFILE FILE...

Permissions
u - User who owns the file.
g - Group that owns the file.
o - Other.
a - All.
r - Read the file.
w - Write or edit the file.
x - Execute or run the file as a program.

Examples

The above numeric permissions can be added to set a certain permission, for example, a common HTML file on a Unix server to be only viewed over the Internet would be:

chmod 644 file.htm

This gives the file read/write by the owner and only read by everyone else (-rw-r--r--).

Files such as scripts that need to be executed need more permissions. Below is another example of a common permission given to scripts.

chmod 755 file.cgi

This would be the following 400+040+004+200+100+010+001 = 755 where you are giving all the rights except the capability for anyone to write to the file.cgi file(-rwxr-xr-x).

chmod 666 file.txt

Finally, another common CHMOD permission is 666, as shown below, which is read and write by everyone.

Additional information

Below is an example of how a file may be listed when typing (ls -l) at the prompt as well as information on how to interpret it.

-rw-rw-r-- 1 hope 123 Feb 03 15:36 file.txt

-	rw	rw-	r--	1	hope	123	Feb 03 15:36	file.txt
File	owner	group	everyone else	links	owner	size	mod date	file name

clear

Clears the screen.

Syntax

clear

Examples

clear

Clear the screen.

Additional information

Many newer Unix systems and Unix variants will also accept the **cls** command or have cls as an alias for clear.

cmp

Compares two files and tells you what line numbers are different.

Syntax

cmp [-c] [-i N] [-l] [-s] [-v] firstfile secondfile

-c	Output differing bytes as characters.
-i N	Ignore differences in the first N bytes of input.
-l	Write the byte number (decimal) and the differing bytes (octal) for each difference.
-s	Write nothing for differing files; return exit statuses only.
-v	Output version info.
firstfile	First file that you wish to compare.
secondfile	Second file that you wish to compare to.

Examples

cmp file1.txt file2.txt

Compares file1 to file2 and outputs results. Below is example of how these results may look.

```
file.txt file2.txt differ: char 1011, line 112
```

cp

Copies files from one location to another.

Syntax

cp [OPTION]... SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... --target-directory=DIRECTORY SOURCE...

Examples

cp file1.txt newdir

Copies the file1.txt in the current directory to the newdir directory.

cp /home/public_html/mylog.txt /home/public_html/backup/mylog.bak

Copies the mylog.txt file in the public_html directory into the public_html/backup directory as mylog.bak. The files are identical however have different names.

cp *.txt newdir

Copy all files ending in .txt into the newdir directory.

cp -r /home/hope/files/* /home/hope/backup

Copies all the files, directories, and subdirectories in the files directory into the backup directory.

yes | cp /home/hope/files/* /home/hope/files2

Copies all the files and subdirectories in files into the files2 directory. If files with the same name exist or it's prompted to overwrite the file it answers yes.

Additional information

Many new versions of Linux/Unix or their variants may also be able to use copy in place of cp or have an alias setup for cp as copy.

date

Tells you the date and time in Unix.

Syntax

date [-a] [-u] [-s datestr]

-a	Slowly adjust the time by sss.fff seconds (fff represents fractions of a second).
-u	Display (or set) the date in Greenwich Mean Time (GMT-universal time),
-s datestr	Sets the time and date to the value specfied in the *datestr*. The datestr may contain the month names, timezones, 'am', 'pm', etc.

Examples

date

List the date and time of the server. Below is an example of the output.

```
Thu Feb 8 16:47:32 MST 2001
```

date -s "11/20/2003 12:48:00"

Set the date to the date and time shown.

date '+DATE: %m/%d/%y%nTIME:%H:%M:%S'

Would list the time and date in the below format.

```
DATE: 02/08/01
TIME:16:44:55
```

time

Used to time a simple command.

Note: If you are trying to display the time in Unix / Linux or a variant use the date command.

Syntax

time [-p] utility [argument]

-p	Write the timing output to standard error in the following format: real %f\nuser %f\nsys %f\n < real seconds>, <user seconds>, <system seconds>
utility	The name of the utility that is to be invoked.
argument	Any string to be supplied as an argument when invoking utility.

echo

Echo's to the screen what you type after echo. Echo is useful for producing diagnostics in command files, for sending known data into a pipe, and for displaying the contents of environment variables.

Syntax

echo [-n] text

-n	On BSD and some variants derived from BSD does not begin a new line after the echoed text.
text	The text that you want to echo to the screen.

Examples

echo Hello world

The above example would return "Hello world" to the console

echo * | wc

The above example would list a count of all the files and directories in the current directory.

exit

Allows you to exit from a program, shell or log you out of a Unix network.

Syntax

exit

Examples

exit

If supported would exit you from the program, shell or log you out of network.

Note: If exit does not log you out you can also do logout, lo, bye, quit, also Ctrl-D may work.

grep

Finds text within a file.

Syntax

grep [options] PATTERN [FILE...]
grep [options] [-e PATTERN | -f FILE] [FILE...]

Patterns for searching.

.	Matches single character.
*	Wild character Example C* if found would pull up CC or CAT...
{}	Matches any character contained within the bracket.
^	Represents the beginning of the line, so if you did ^T it would search for any sentence starting with a T.
$	Represents the end of the line, so if you did $. then it would pull up any lines that ended with .
\	Means to take the next character serious so you could search for C\ C.

Note: Be careful using the characters $, *, [, ^, |, (,), and \ in the pattern list because they are also meaningful to the shell. It is safest to enclose the entire pattern list in single quotes '... '.

Examples

grep "unix" *.htm

search all .htm files in the current directory for any reference of unix and give results similar to the below example text.

```
asoftwar.htm: href="win95.htm">Windows 95</a>, <a href="unix.htm">Unix</a>, <a
href="msdos.htm">MS-DOS</a>,
asoftwar.htm: <td><font face="Times New Roman"><a name="U"></a><a
href="unix.htm"><strong>Unix</strong></a></font></td>
learnhtm.htm: <a href="unix.htm">Unix help</a><br>
os.htm: <a href="unix.htm">Unix</a><br>
```

As seen above the grep command has found references of unix in some of the HTML files in our home directory. The file name that contains unix is listed at the beginning of the line followed by a colon and the text continuing unix.

ls -1 | grep ^a | wc -l

head

Displays the first ten lines of a file, unless otherwise stated.

Syntax

head [-number | -n number] filename

-number	The number of the you want to display.
-n number	The number of the you want to display.
filename	The file that you want to display the x amount of lines of.

Examples

head -15 myfile.txt

Display the first fifteen lines of myfile.txt.

ls

Lists the contents of a directory.

Syntax

ls [-a] [-A] [-b] [-c] [-C] [-d] [-f] [-F] [-g] [-i] [-l] [-L] [-m] [-o] [-p] [-q] [-r] [-R] [-s] [-t] [-u] [-x] [pathnames]

Examples

ls -l

In the above example this command would list each of the files in the current directory and the files permissions, the size of the file, date of the last modification, and the file name or directory. Below is additional information each of the fields this command lists.

Permissions	**Directories**	**Group**	**Size**	**Date**	**Directory or file**
drwx------	2	users	4096	Nov 2 19:51	mail/
drwxr-s---	35	www	32768	Jan 20 22:39	public_html/
-rw-------	1	users	3	Nov 25 02:58	test.txt

Below is a brief description of each of the above categories shown when using the ls -l command.

Permissions - The permissions of the directory or file.

Directories - The amount of links or directories within the directory. The default amount of directories is going to always be 2 because of the . and .. directories.

Group - The group assigned to the file or directory

Size - Size of the file or directory.

Date - Date of last modification.

Directory of file - The name of the file or file.

ls ~
List the contents of your home directory by adding a tilde after the ls command.
ls /
List the contents of your root directory.
ls ../
List the contents of the parent directory.
ls */
List the contents of all sub directories.
ls -d */
Only list the directories in the current directory.

mkdir

Short for make directory this command is used to create a new directory.

Syntax

mkdir [option] directory

-m mode	Set permission mode (as in chmod), not rwxrwxrwx - umask.
-p	No error if existing, make parent directories as needed.
-v	Print a message for each created directory
-Z	(SELinux) set security context to CONTEXT
directory	The name of the directory that you wish to create.

Examples
mkdir mydir
The above command creates a new directory called mydir.
mkdir -m a=rwx mydir
This next example would use the -m option to not only create the mydir directory but also set the permissions to all users having read, write, and execute permissions.

mv

Renames a file or moves it from one directory to another directory.

Syntax

mv [-f] [-i] oldname newname

oldname	The oldname of the file renaming.
newname	The newname of the file renaming.
filename	The name of the file you want to move *directory* - The directory of were you want the file to go.

Examples

mv myfile.txt newdirectory/
Moves the file myfile.txt to the directory newdirectory.
mv myfile.txt ../
Moves the file myfile.txt back one directory (if available).
mv computer\ hope.txt computer_hope.txt
Moves (renames) the file "computer hope.txt" to computer_hope.txt. When working with a file or directory with a space you must escape that space with a backslash or surround the filename or directory with quotes.

pwd

Short for print working directory the pwd command displays the name of the current working directory.

Syntax

pwd

Examples

pwd

Typing pwd at the prompt would give you something similar to:

```
/home/computerhope/public_html
```

Users who are familiar with MS-DOS or the Windows command prompt may type cd alone to print the working directory. However, typing cd alone in Linux and Unix will return you to the home directory.

rm

Deletes a file without confirmation (by default).

Syntax

rm [-f] [-i] [-R] [-r] [filenames | directory]

-f	Remove all files (whether write-protected or not) in a directory without prompting the user
-i	Interactive. With this option, rm prompts for confirmation before removing any files.
-R	Same as -r option.
-r	Recursively remove directories and subdirectories in the argument list.
filenames	A path of a filename to be removed.

Examples

rm myfile.txt

Remove the file myfile.txt without prompting the user.

rm -r directory

Remove a directory, even if files existed in that directory.

rmdir

Deletes a directory.

Syntax

rmdir [OPTION]... DIRECTORY...

--ignore-fail-on-non-empty	ignore each failure that is solely because a directory is non-empty.
-p, --parents	Remove DIRECTORY and its ancestors. E.g., `rmdir -p a/b/c' is similar to `rmdir a/b/c a/b a'.
-v, --verbose	output a diagnostic for every directory processed.
--version	output version information and exit.

Examples
rmdir mydir
Removes the directory mydir
rm -r directory
Remove a directory, even if files existed in that directory.

sort

Sorts the lines in a text file.

Syntax

sort [-b] [-d] [-f] [-i] [-m] [-M] [-n] [-r] [-u] [+fields] filename [-o outputfile]

-b	Ignores spaces at beginning of the line.
-d	Uses dictionary sort order and ignores the punctuation.
-f	Ignores caps
-i	Ignores nonprinting control characters.
-m	Merges two or more input files into one sorted output.
-M	Treats the first three letters in the line as a month (such as may.)
-n	Sorts by the beginning of the number at the beginning of the line.
-r	Sorts in reverse order
-u	If line is duplicated only display once
+fields	Sorts by fields , usually by tabs
filename	The name of the file that needs to be sorted.
-o outputfile	Sends the sorted output to a file.

Examples
sort -r file.txt
Sort the file, file.txt in reverse order.

wc

Short for word count, wc displays a count of lines, words, and characters in a file.

Syntax

wc [-c | -m | -C] [-l] [-w] [file ...]

-c	Count bytes.
-m	Count characters.
-C	Same as -m.
-l	Count lines.
-w	Count words delimited by white space characters or new line characters.
file	Name of file to word count.

Examples

wc myfile.txt

Displays information the file myfile.txt. Below is an example of the output.

```
5 13 57 myfile.txt
```

5 = Lines
13 = Words
57 = Characters

whoami

Print effective userid.

Syntax

whoami

Examples

whoami

Would display the name of the current userid. For example, may list root if you're logged in as root.

passwd

Allows you to change your password.

Syntax

passwd [-r | files | -r nis | -r nisplus] [-a] [-d | -l] [-e] [-f] [-g] [-h] [-n min] [-s] [-w warn] [-x max] [-D domainname][name]

-r	Specifies the repository to which an operation is applied.
-a	Show password attributes for all entries.
-d	Deletes password for name.
-l	Locks password entry for name.
-e	Change the login shell.
-f	Force the user to change password at the next login
-g	Change the gecos (finger) information
-h	Change the home directory.
-n min	Set minimum field for name.
-s	Displays information
-w warn	Set warn field for name
-x max	Set maximum field for name.
-D domainname	Consult the passwd.org_dir table in domainname
name	Login ID of user

Examples

passwd - entering just passwd would allow you to change the password. After entering passwd you will receive the following three prompts:

Current Password:
New Password:
Confirm New Password:

Each of these prompts must be entered and entered correctly for the password to be successfully changed.

passwd newperson

If you have just setup an account using the useradd command you would then type a command similar to the above command to set the password for the "newperson" account. Only root users have the ability to set passwords for other accounts.