
Unix & Linux Tutorial

[Passwords, Ownership, Permissions](#)

[File Manipulation](#)

[Using Unix Mail](#)

[Networking](#)

[List of Unix/Linux Commands](#)

Overview

There are many different flavors of Unix. Linux is a popular variant that is freely distributed. This is a general description of how to get around in most Unix systems. The availability of the commands listed here depends on the system you are using. To get online help from any Unix system, type "man programname" for local documentation on the program you specify.

Several conventions are followed in this document. Unix is case sensitive. If a capital letter is shown in a command, it is required. An empty set of angle brackets (<>) represents the <return> or <enter> key. A set of brackets with a label inside (<label>) represents the key with that label. Single quotation marks are used to indicate variable characters and file names. The single quotation marks are not to be typed in the shell window. Parentheses are used to set off instructions and are not to be included in the commands. Brackets around a [path] are not typed but indicate that the path to a file or directory must be included in the command.

Many Unix commands are *base commands* that can be followed by a series of *switches*, or modifiers, with no space between them. Most Unix shells require a dash before the switches. The format is as follows:

```
command -switch1switch2
```

Passwords, Ownership, Permissions

As a multitasking, multiuser operating system, Unix requires a security system. The three methods of security employed by Unix are individual passwords for users, file ownership by users, and permissions granted by file owners.

Passwords

passwd

Changes a user's password at any time. The system asks for the old password, then the new one, then a confirmation of the new password. A password should be 6-8 characters long, with at least one numeric character. Non-alphanumeric characters are acceptable.

Ownership

chown

The creator of a file is automatically the owner. Change ownership with the chown command, as follows:

```
chown newowner 'fname' <>
```

Permissions

```
chmod
```

Used by a file owner to change permissions. This command uses modes to set permissions: r (read), w (write), and x (execute). Multiple permissions may be set with one chmod command by adding together modes to give o (owner), u (user group), or a (all) permissions.

File Manipulation

File Names

File names in any of the following commands may include wildcards. ? allows any one character, whereas * allows no character or any number of characters. All Unix switches can be combined; thus, "ls -al" would produce an expanded listing including any hidden files.

```
ls
```

Lists the files in the current directory; if followed by 'fname' it indicates whether that file exists.

-l Shows expanded listing, identifying files and directories.

-d Lists directories only.

-a Includes hidden files.

```
cat 'fname'
```

Displays the content of 'fname' on the screen, with scrolling. Multiple files can be displayed by concatenating.

```
cat 'fname1' >> 'fname2'
```

Appends (concatenates) 'fname1' at the end of 'fname2.'

```
more 'fname'
```

Displays 'fname' one screen at a time; subsequent screens are displayed by pressing <spacebar>, subsequent lines with <>. Press q to end the display. Paging backward is not possible on most systems.

```
diff 'fname1' 'fname2'
```

Compares two files and displays the differences. Lines from 1 are preceded by <, and lines from 2 are preceded by >.

Copying, Moving, and Deleting Files

```
cp [path]/'fname1' [path]/'fname2'
```

Copies 1 to 2. If no path is specified, the current directory is used.

```
mv [path]/'fname1' [path]/'fname2'
```

Changes the name of the file from 1 to 2.

With both cp and mv, Unix will overwrite any existing files named 'fname2' without warning.

```
rm [path]/'fname'
```

Removes (deletes) 'fname.' Concatenation is possible; wildcards may be used. Unix does not allow recovery of deleted files.

-I Requires confirmation for each specific file deleted.

Compressing, Backing Up, and Restoring Files

Compressing a file in Unix typically applies the Ziv-Lempel algorithm. Related files can be combined and compressed as one file using the tar utility.

```
compress 'fname'
```

Encodes 'fname,' reduces its size, and adds the extension .Z.

-v The percentage of the compression will be listed upon completion.

```
uncompress 'fname'
```

Restores 'fname.Z' to its unencoded state.

```
gzip 'fname'
```

Encodes 'fname' using a different compression algorithm and adds the extension .gz.

```
gunzip 'fname'
```

Restores 'fname.gz' to its unencoded state.

```
tar -cvf 'fname1.tar' [path] /'fname2' 'fname3'
```

Copies 'fname2' and 'fname3' to the archive 'fname1.tar.'

-c Specifies a new archive.

-v Displays a list of the files being archived (verbose mode).

-f Creates the archive as a file.

```
tar -xvf [path]/ 'fname.tar'
```

Restores files from an archive file and displays a list of them on the screen.

-x Restores files from an archive file.

Directory Manipulation

```
mkdir [path] 'dirname'
```

Creates the directory 'dirname' in the location specified by [path]. If [path] is not given, 'dirname' is created in the current directory.

```
cd [path] 'dirname'
```

Changes to the directory 'dirname' at the location [path].

Type "\$home" at the beginning of [path] to start at the root directory; \$home as 'dirname' returns the user to the home directory.

cd

Returns the user to the home directory on most systems, or moves one directory up the tree.

pwd

Displays the path of the working directory.

rmdir [path] 'dirname'

Removes the specified directory, but only if it is empty.

rm -r 'dirname'

Removes the specified directory and its contents recursively through all subdirectories.

On most systems, the rm command does not ask for confirmation unless the -i switch is set.

Monitoring the System and Disk Usage

The following commands allow a user to check whether a process is running in the background and to shut down a program without waiting for it to finish.

ps -a

Provides a list of current processes. -ef gives more details than -a.

top -i #

Lists the running processes and the percentage of cpu usage by each, updating the list every # seconds. Press q to end the display.

kill [PID]

Kills the process with the given process identification.

kill -HUP [PID]

Causes the process specified to reread its configuration file, allowing it to be restarted with different parameters. To check the available disk space:

df -k

Displays a list of directories, their sizes, and the percentage of the disk in use. The -k option displays the size in kilobytes rather than blocks.

du -k [path]

Provides a listing of the files in the directory specified by [path] and their sizes.

dtree

Displays the directory structure.

Using Unix Mail

Reading and Processing Received Mail in Unix

The following are command-line options for sending mail:

`mail`

Invokes the mail client. If mail has been received, headers are displayed.

`p`

Displays the current message. To make a message current, enter its number.

`+`

Displays the next message.

`-` (dash)

Displays the previous message.

`M`

Displays the current message, page by page.

`d`

Deletes the current message. To delete a specific message, add its number.

`u`

Restores (undeletes) the most recently deleted message.

`s 'fname'`

Saves the current message as 'fname' with headers.

`w 'fname'`

Saves the current message with no headers.

Sending Mail

In compose mode Mail will prompt for a subject. Type a subject and press <>; then type the message.

`~v :`

Switches to vi to compose the message (see the Editors section below).

`<Ctrl>C`

Aborts composition without sending the mail.

`.` (period)

Alone on a line, the `.` command terminates composition and sends the mail.

`~r 'fname'`

Incorporates 'fname' in the current message, at the current cursor position.

`R`

Replies to the current message; enters compose mode.

`q`

Quits Mail (the command `x` may also be used).

Editors

The most common Unix text editor is `vi`. It is a line editor, meaning that its functions normally are executed line by line or word by word. It operates in either command or insert mode. Some editing functions may be carried out from command mode.

Starting vi

`vi 'fname'`

If 'fname' does not already exist, it will be created. A number of files may be loaded simultaneously by concatenating file names. Move to the next file by typing "n."

Command Mode

`:` (colon)

Type a colon (`:`) to enter command mode. Then type a command and press `<>`.

`:q`

Leaves `vi`.

`:q!`

Leaves without saving the file buffer.

`:w 'fname'`

Writes the current file to a disk as 'fname'; if a file of that name already exists, it will be overwritten.

`:lnumber1, lnumber2 w 'fname'`

Writes the portion of the current file that lies between `lnumber1` and `lnumber2` to 'fname.'

`:x`

Saves the file and quits (`:wq` may also be used).

Moving Around in a File

The following commands are not prefaced with a colon. They are used to move the cursor within a file for viewing or to reach the point where editing is to begin.

- (dash) or <up arrow>

Move up one line.

<> or <down arrow>

Move down one line.

<backspace> or <left arrow>

Move left one character.

<spacebar> or <right arrow>

Move right one character.

\$

Move to the end of the line.

^

Move to the beginning of the line.

H

Move to the top of the screen.

L

Move to the bottom of the screen.

M

Move to the middle of the screen.

<Ctrl>D

Move down half a page.

<Ctrl> U

Move up half a page.

<Ctrl> F

Move down a whole page.

<Ctrl> B

Move up a whole page.

'lnumber' G

Go to a specific line (:set nu displays line numbers).

G

Go to the end of the file.

`/'text'`

Search forward for 'text' in a file.

`?'text'`

Search backward for 'text' in a file.

`n`

Continue the search for the next instance of 'text' (when using `/'text'` and `?'text'`).

`N`

Go back to the previous instance of 'text.'

Editing in Command Mode

Editing commands that can be used without entering insert mode:

`x`

Delete the character under the cursor; `nx` (`n` = a number) deletes `n` characters.

`dd`

Delete the current line; `ndd` deletes `n` lines.

`p`

Enter the most recently deleted text after the cursor; `p` pastes in front of the cursor.

`YY`

Copy the line to the buffer (yank).

`O <esc>`

Insert a blank line above the cursor.

`o <esc>`

Insert a blank line below the cursor.

`r 'newchar'`

Replace the character at the cursor with 'newchar.'

`R 'newchars'`

Replace multiple characters.

`:r 'fname'`

Insert the file 'fname' in the current file at the cursor.

Note: All replace commands put vi in insert mode; type `<esc>` to return to command mode.

u

Undo the most recent change.

. (period)

Repeat the last command.

Editing in Insert Mode

In insert mode vi behaves like a standard ASCII text editor. After entering insert mode, vi remains in insert mode until <esc> is pressed; commands that belong to command mode are not executed. Typed characters are entered as text.

i :

Enter insert mode; text fills to the left of the cursor.

a :

Enter insert mode; text fills to the right of the cursor.

I :

Insert text at the beginning of the line.

A :

Insert text at the end of the line.

<Esc> :

Exit insert mode.

Networking

Networking capabilities are built into the Unix operating system. Telnet provides remote command capabilities, and FTP allows the transfer of files between machines. All networking activities require a daemon program running on the server and a client program running on the remote computer.

Telnet

The telnet protocol allows remote computers to log in and execute standard Unix commands on a Unix server. The server may be accessed from a PC or Macintosh computer running a telnet client. From a Unix command line, type "telnet machinename" to begin using telnet.

After the client connects to the daemon on the server, the daemon will ask for a username and password. Once authenticated, the user can execute all standard Unix commands. To end a telnet session, type "exit" at the command line.

FTP

File transfer protocol allows machines to pass files back and forth. Upon the opening of an FTP connection, the prompt is ftp>, not the standard %. If the remote machine supports anonymous FTP,

you can enter the word "anonymous" in response to the request for your name and enter your email address as the password.

Other than commands for transferring files, only three commands work under an FTP connection: `ls` lists directories, `pwd` gives the path name of the working directory, and `cd` allows you to change directories.

These commands function only on the host system when a user is logged in. To begin an FTP session, open the desired local directory where data will be transferred.

Individual files are transferred with the `put` or the `get` command. To transfer 'filename' from the local machine to the host, type:

```
put 'filename'
```

To retrieve 'filename' from the host and store it on the local machine, type:

```
get 'filename'
```

The FTP daemon alerts the user when a file transfer is completed.

Transferring multiple files is accomplished with the `mget` and `mput` commands. These commands will accept arguments consisting of either a list of file names or a generic name using wildcards to specify a group of files. The daemon asks before transferring each file when a group of files are sent together. Enter a "y" to send a file or an "n" to skip over a file in the group. The following command transfers the three specified files one after another from the local machine to the host:

```
mput filename1 filename2 filename3
```

The following command retrieves all files beginning with the characters filename and seeks confirmation before each file is transferred.

```
mget filename*
```

Binary or ASCII File Transfers

The FTP program on the server must be set to binary file transfer mode to transfer compressed files. Because binary mode works for both ASCII and executable files, it is the preferred default setting. Enter "ascii" or "binary" at the FTP prompt to select the type of files to be transferred.

To terminate an FTP session and shut down the client, enter the command "bye" or "quit." Enter the command "close" to terminate the current connection and leave the client running. Initiate a new connection by typing "open machinename."

Getting Help

The "man" pages in Unix provide an online help system. To get detailed help on any command, type "man" followed by the command. For help on `chmod`, enter the following at a shell prompt:

```
man chmod
```

The online documentation for `chmod` will be formatted and displayed, and the more command may be used to page forward.

List of Unix/Linux Commands

Please refer to the document [Unix/Linux Commands](#).