

Linux Commands

ls - List Files

<code>ls</code>	List files in the current directory.
<code>ls file ...</code>	List the named files.
<code>ls -l ...</code>	Give a long listing with permissions, owner, etc.
<code>ls -a ...</code>	List all files, includes ones whose names start with a period.
<code>ls dir</code>	List files in directory <code>dir</code> .
<code>ls -d ...</code>	List directories - not the contents.
<code>ls -i ...</code>	List the inodes.
<code>ls -r ...</code>	Reverse order.
<code>ls -R ...</code>	Recursively list directories.
<code>ls -t ...</code>	Sort by time.

ls -l (1)

```
prompt> ls -l
total 8
drwx----- 2 sullivan staff 4096 Jan  9 13:45 dir
-rw----- 1 sullivan staff   51 Jan  9 13:45 file
```

File type

```
drwx----- 2 sullivan staff 4096 Jan  9 13:45 dir
-rw----- 2 sullivan staff   51 Jan  9 13:45 file
```

- d means directory
- - means plain file

ls -l (2)

Permissions for owner

```
drwx----- 2 sullivan staff 4096 Jan 9 13:45 dir
-rw----- 2 sullivan staff   51 Jan 9 13:45 file
```

Permissions for group

```
drwx----- 2 sullivan staff 4096 Jan 9 13:45 dir
-rw----- 2 sullivan staff   51 Jan 9 13:45 file
```

Permissions for everyone else

```
drwx----- 2 sullivan staff 4096 Jan 9 13:45 dir
-rw----- 2 sullivan staff   51 Jan 9 13:45 file
```

- r means read
- w means write
- x means execute

ls -l (3)

Number of hard links

```
drwx----- 2 sullivan staff 4096 Jan 9 13:45 dir
-rw----- 2 sullivan staff   51 Jan 9 13:45 file
```

Owner

```
drwx----- 2 sullivan staff 4096 Jan 9 13:45 dir
-rw----- 2 sullivan staff   51 Jan 9 13:45 file
```

Group owner

```
drwx----- sullivan staff 4096 Jan 9 13:45 dir
-rw----- sullivan staff   51 Jan 9 13:45 file
```

ls -l (4)

Size

```
drwx----- 2 sullivan staff 4096 Jan 9 13:45 dir
-rw----- 2 sullivan staff  51 Jan 9 13:45 file
```

Last modified time

```
drwx----- 2 sullivan staff 4096 Jan 9 13:45 dir
-rw----- 2 sullivan staff  51 Jan 9 13:45 file
```

Name

```
drwx----- sullivan staff 4096 Jan 9 13:45 dir
-rw----- sullivan staff  51 Jan 9 13:45 file
```

man - Print a manual page

man printf

Manual page for the printf command

man 3 printf

Manual page for the printf C library function

Manual Sections

- 1 User commands
- 2 System calls
- 3 Library functions
- 4 Special files
- 5 File formats
- 6 Games
- 7 Miscellaneous
- 8 System administration commands

echo - Print arguments to stdout

```
echo foo bar xyzy
```

Prints foo bar xyzy.

```
echo -n foo
```

Prints foo and omits the newline

mkdir - Make Directories

```
mkdir dir ...
```

Create the directories.

rmdir - Remove Directories

```
rmdir dir ...
```

Remove the directories, but only if they are empty.

rm - Remove Files

```
rm file ...
```

Remove files.

```
rm -r file ...
```

Remove files and recursively remove directories and their contents.

```
rm -r a* ...
```

Remove all files and directories starting with a.

```
rm -r a *
```

Remove a and all other files and directories. (Beware!!!)

mv - Move or Rename Files

```
mv old new
```

Rename old to new.

```
mv file ... dir
```

Move the files into the directory.

cat - Concatenate Files To Standard Output

```
cat file ...
```

Writes contents of files to stdout.

```
cat
```

Copies stdin to stdout.

```
cat -n ...
```

Numbers lines in file.

```
cat -e ...
```

Show ends of lines.

```
cat -T ...
```

Show tabs.

```
cat -v ...
```

Show nonprinting characters.

pwd - Print the Current Working Directory

pwd

Print the name of the current directory.

cd - Change Directory

```
cd dir
```

Changes the current directory to dir.

```
cd
```

Changes the current directory to the home directory.

cp - Copy Files and Directories

```
cp oldfile newfile
```

Copies oldfile to newfile.

```
cp file ... dir
```

Copies files into a directory.

```
cp -r olldir newdir
```

Copies a directory recursively.

head - Print the First Lines of a File

```
head file ...
```

Prints the first 10 lines of each file.

```
head
```

Prints the first 10 lines from stdin.

```
head -n N ...
```

Prints the first N lines of each file.

```
head -n -N ...
```

Prints all but the last N lines of each file.

tail - Print the Last Lines of a File

```
tail file ...
```

Prints the last 10 lines of each file.

```
tail
```

Prints the last 10 lines from stdin.

```
tail -n N ...
```

Prints the last N lines from each file.

```
tail -n +N ...
```

Prints lines beginning with line N from each file.

ssh - Secure Shell

```
ssh user@host
```

Logs user into host.

```
ssh -XY user@host
```

Logs user into host with X11.

```
ssh user@host command
```

Executes command on host as user.

- Also available on windows, mac.

scp - Secure Copy

```
scp file ... user@host:
```

Copies files to the user's home directory on host.

```
scp file ... user@host:dir
```

Copies files to user's directory dir on host.

```
scp user@host:file .
```

Copies file from the user's home directory on host.

```
scp user@host:file dir
```

Copies file from user's home directory to dir on local machine.

- Also available on windows, mac.

sort - Sort a File

```
sort file
```

Sorts file lexicographically to stdout.

```
sort -n file
```

Sorts file numerically to stdout.

```
sort
```

Sorts lines from stdin to stdout.

```
sort -r
```

Sorts in reverse.

tr - Transliterate (Replace) Characters

- tr reads from stdin and writes to stdout
- tr does not take file arguments

```
tr # %
```

Replaces # by %

```
tr A-Z a-z
```

Replaces A-Z by a-z

```
tr -c A-Za-z '\n'
```

(Complement) Replaces non-letters by newlines

```
tr -d '\r'
```

(Delete) Removes carriage returns

uniq - Remove Adjacent Duplicate Lines

```
uniq
```

Reads from stdin, writes to stdout

```
uniq infile outfile
```

Reads from infile, writes to outfile

less - Display Files Screen by Screen

```
less infile
```

Reads from infile.

```
less
```

Reads from stdin.

```
less file1 file2 file3
```

Reads from each file in turn.

Less Commands

```
q
```

Quit.

```
space
```

Move to the next screenful.

```
/regex
```

Searches for matches for a regular expression.

```
uparrow
```

Scroll up.

```
downarrow
```

Scroll down.

```
1G
```

Move to line 1 of file.

```
nG
```

Move to line n

```
$G
```

Move to end of file.

```
:n
```

Move to the next file.

seq - Print a Sequence of Numbers to Stdout

```
seq 10
```

Print the numbers from 1 to 10

```
seq 10 20
```

Print the numbers from 10 to 20

```
seq 10 2 20
```

Print the numbers from 10 to 20 incrementing by 2

wc - Count Lines, Words, and Characters in a File

```
wc file ...
```

Count lines, words, and characters in files

```
WC
```

Count lines, words, and characters from stdin

```
wc -l
```

Count lines from stdin

```
WC -w
```

Count words from stdin

```
WC -c
```

Count characters from stdin

shuf - Writes Lines to Standard Output in Random Order

```
shuf file
```

Randomizes lines from file

```
shuf
```

Randomizes lines from stdin

```
shuf -i 1000:9999
```

Randomizes the numbers from 1000 to 9999

```
shuf -i 1000:9999 -n
```

Randomizes numbers from 1000 to 9999 and prints n of them

```
shuf -i 1000:9999 -n 100
```

Prints 100 random numbers from 1000 to 9999 without repeats

```
shuf -i 1000:9999 -n 100 -r
```

Prints 100 random numbers from 1000 to 9999 allowing repeats

chmod - Change Permissions (1)

Syntax

```
chmod [ugoa]*[+ -=][rwxXst]+ file ...  
chmod octal file ...
```

Who is affected

u	owner
g	group
o	others
a	same as ugo

How are permissions changed

+	Add a permission
-	Take away a permission
=	Set the permissions exactly

chmod - Change Permissions (2)

Permissions

r	read
w	write
x	execute file or search directory
X	execute, but only if execute is already on for someone
s	setuid or setgid
t	sticky bit

- Can change multiple permissions with one command (separate by commas)
- Example: `chmod u=rw,g=r,o= file`

chmod - Change Permissions (3)

Octal Permissions

- Permissions are of the form UGO (3 octal digits)
- In each octal digit, each bit is one permission
 - $r = 4$
 - $w = 2$
 - $x = 1$
 - $rwx = 7$
 - $rw = 6$
 - $rx = 5$
- Example `chmod 640 file` is the same as `chmod u=rw,g=r,o= file`

Setuid, Setgid, and Sticky Bits

setuid	Program runs with owner permissions of file owner rather than user
setgid	Program runs with group permissions of file group rather than user
sticky bit	Directories - only owner can delete or rename files in the directory Regular files - keep program in swap space after completion (obsolete)

In - Create Links

```
ln file link-name
```

Create a hard link

```
ln -s file link-name
```

Create a symbolic link

```
ln file dir
```

Create a hard link to the file in the directory

```
ln -s file dir
```

Create a symbolic link to the file in the directory

sed - stream editor (1)

Syntax

sed [options] pattern(no space)command [file ...]

Reads from files or stdin and writes to stdout.

Patterns

m,n	Lines m through n
m,\$	Lines m to the end
/regex/	Lines matching regex
missing pattern	All lines

Commands

s/a/b/	Substitute b for first a
s/a/b/g	Substitute b for all a's
d	Delete line

sed - stream editor (2)

Options

-e	Command coming up (can have more than one)
-i	Edit in place (replace original file - use with caution)

Examples

```
sed -e 's/foo/bar/'
```

Replace the first occurrence of foo on each line by bar

```
sed -e 's/foo/bar/g'
```

Replace all occurrences of foo on each line by bar

```
sed -e '1,15s/foo/bar/g'
```

Replace only on lines 1-15

```
sed -e '15,$s/foo/bar/g'
```

Replace only on lines 15 to the end

```
sed -e 's/$/foo/'
```

Append foo to every line

```
sed -e '1,15d'
```

Delete lines 1-15

```
sed -e '/^[fc]/d'
```

Delete lines starting with f or c