File commands (ls, cp, mv, ln) Everyday file handling

CONCEPTS

> No trash can: rm and overwriting cp/mv are permanent. Add -i or -n while learning, and expand globs with echo first.

Directory entry name
Inode owner, mode, times
Data blocks contents

A directory entry names an inode; data lives until the last name goes

> A hard link is another name for the same inode (same filesystem, files only). A symlink is a small file holding a path and can dangle.

LISTING (LS)

ls
List the current directory

ls -l
Long format: mode, owner, size, date

ls -a
Include hidden dotfiles

ls -lh
Human-readable sizes

ls -lt
Newest first

ls -ltr
Oldest first, newest at the bottom

ls -d */
List only directories

ls -ld *<path>*
Show the directory itself, not its contents

COPYING (CP)

cp *<src> <dst>*
Copy a file

cp -r *<dir> <dst>*
Copy a directory recursively

cp -a *<src> <dst>*
Archive: recurse, keep links, mode, times

cp -i *<src> <dst>*
Ask before overwriting

cp *<src>*/. *<dst>*
Copy directory contents, including dotfiles

MOVING (MV)

mv *<src> <dst>*
Move or rename

mv *<a> <b> <dir>*
Move several items into a directory

mv -i *<src> <dst>*
Ask before overwriting

REMOVING (RM, RMDIR)

rm *<file>*
Delete a file

rm -i *<file>*
Confirm each deletion

rm -r *<dir>*
Delete a directory tree

rm -rf *<dir>*
Recursive, no prompts, ignore missing

rm -I *<files>*
One prompt for 4+ files or recursion

rm -- -<*name*>
Delete a file whose name starts with -

rmdir *<dir>*
Remove an empty directory only

> Prefer rm -I or a dry run (ls <glob>) before any recursive delete. Use rmdir when the directory should already be empty.

LINKS (LN, READLINK)

ln -s *<target> <link>*
Create a symbolic link

ln -sf *<target> <link>*
Replace an existing link

ln -sfn *<target> <link>*
Replace a link to a directory in place

ls -l *<link>*
Show where a link points

readlink -f *<path>*
Resolve fully to an absolute path

> A relative symlink target is resolved from the link's directory, not from where you ran ln.

DIRECTORIES AND FILES

mkdir *<dir>*
Create a directory

mkdir -p *<a/b/c>*
Create parents, no error if present

mkdir -p src/{app,lib}
Create several siblings with brace expansion

touch *<file>*
Create empty file or update its time

INSPECTING PATHS

stat *<file>*
Size, inode, mode, owner, times

stat -c '%a %U %s %n' *<file>*
Custom format: octal mode, owner, size, name

realpath *<path>*
Absolute path, symlinks resolved

basename *<path>*
Last path component

dirname *<path>*
Everything but the last component

TEMPORARY FILES (MKTEMP)

mktemp
Create a unique temp file, print its path

mktemp -d
Create a unique temp directory

mktemp /tmp/<*name*>**.XXXXXX**
Use your own template

> Always use mktemp rather than a fixed name like /tmp/out: it avoids clashes and symlink attacks. Remove it with a trap.

RECIPES

Safe scratch directory

```
tmp=$(mktemp -d) || exit 1
trap 'rm -rf "$tmp"' EXIT
cp -a <src>/. "$tmp"
```

Back up a file before editing

```
cp -a --backup=numbered <file>
<file>.bak
ls -l <file>*
```