FUNDED BY NIHR | National Institute for Health Research

https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode

File and Folder Manipulation in UNIX

Moving, copying and changing file permissions in UNIX can be just as fast as point and click/drag and drop but takes a little getting used to so here's a quick guided tour of the basics:

One thing to note as good practice is to **not** spaces in filenames but replace them with the _ underscore character

Moving files

mv - Move or rename a file or folder.

Please note

- If a file already exists under the new name/location, it will be replaced with the moved file without any warning and no opportunity to undo
- If a folder exists under the new name/location, the file will be placed inside that directory, with the same name as it had originally.

Examples

```
mv foo bar
```

rename a file in the current directory from "foo" to "bar"

```
mv foo ~/documents
```

move the file "foo" from the current directory into a directory called documents in your home directory

```
mv foo ~/documents/bar
```

move the file "foo" from the current directory to a directory called documents in your home directory and rename it "bar"

```
mv *.jpg ~/documents
```

move all files with names ending in ".jpg" from the current directory to a directory called documents in your home directory

Copying Files

cp - Copy a file (or directory).

Please note

- If a file already exists with the name given for the copy, it will be replaced with the copied file without any warning undo option
- If a folder exists with the name given for the copy, the copy will be placed inside that directory, with the same name as the original file.

Examples

https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode

```
cp foo bar
```
copy a file named "foo" in the current directory to a file in the same directory called bar

```
cp foo ~/documents
```
copy a file named "foo" in the current directory into a directory called documents in your home directory. Note ~ stands for your home directory (/home/<YOUR USERNAME> on a Linux machine, /Users/<YOUR USERNAME> on a mac.

```
cp foo ~/documents/bar
```
copy a file named "foo" in the current directory into the documents directory in your home directory and name the copy "bar"

```
cp *.jpg ~/documents
```
copy all files with names ending in ".jpg" into the documents directory in your home directory

```
cp -r documents documents_backup
```
copy an entire directory named "documents" and name the copy "documents_backup". When ever copying a directory the -r is required (it stands for recursive)

Removing Files

rm - Remove (delete) a file or directory.

Please note

- There is no Trash bin or Waste basket. If you delete a file it is gone, there is no undo

Examples

```
rm foo
```
delete the file named "foo"

```
rm a*
```
delete all files with names beginning with "a"

```
rm *.jpg
```
delete all files with names ending in ".jpg"

```
rm -R temp
```
delete the directory named "temp", and all of its contents

rmdir - Delete an empty directory. If you want to delete a directory that isn't empty, you either need to delete the contents first, or use the `rm -R` command instead.

Global Health Research Unit

FUNDED BY NIHR | National Institute for Health Research

https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode

Making Directories

mkdir - Make directory

Example

`mkdir foo`

Create a new directory named "foo"

`mkdir /home/user1/bar`

Create a new directory named "bar" in the `/home/user1` directory

Changing file and directory permissions

chmod - Change protection mode on files and folders. You must own the files (or be root) to change their permissions.

Every file has a owner and a group and these can be seen by typing the `ls -l` command
In the example below the first column shows the file permissions. The 'd' designates that the file is a directory. The next three letters show the permissions for the owner of the file (here in red). The second group of three letters show the permissions for the group of the file (here in blue) and the last three letters show the permissions for everybody else (others). This file can be read, written and executed by the owner and group but only read and executed by others.

```
drwxrwxr-x  5 biouser biouser     4096 Feb  5 18:54 assembly
```

Examples

`chmod u+w foo`

Allow the user (file owner) write access to the file or folder named "foo"

`chmod u-r foo`

Disallow the user (file owner) read access to "foo"

`chmod ug+x foo`

Allow the user (file owner) and group members execute access to "foo"

`chmod o+r *`

Allow "other"s read and access to all files in the current directory.

`chmod o=rw foo`

Allow "other"s read and write access, but disallow execute to "foo"

`chmod -R ugo+r ~/documents`

https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode

Allow everyone (user, group, and other) read access to the documents directory in your home directory and everything in it (the -R means change permissions for the folder's entire contents, not just the folder itself)

Changing file and directory owner and group

chown - Change the owner and/or group of a file or folder. You must be root to use this command.

Examples

```
sudo chown user1 foo
```

Switch to root and assign user1 as the owner of the file name "foo"

```
sudo chown -R user1:staff /home/user1/documents
```

Assign user1 as the owner and staff as the group for the "documents" directory, and everything in it (the -R means change ownership/group for the folder's entire contents, not just the folder itself)

chgrp - Change the group of a file or folder. You must own the file, and can only assign to groups you're a member of (unless you're root).

Examples

```
chgrp staff foo
```

Assign staff as the group of the "foo" file

```
chgrp -R staff /home/user1/documents
```

Assign staff as the group for the "documents" directory, and everything in it (the -R means change group for the folder's entire contents, not just the folder itself)

References

- Comprehensive tutorial
- Cheatsheet