CS10120 - Setting Web Permissions

Hannah Dee (based in part on work by Angharad Shaw)

February 2026

1 Putting pages on the WWW

To put something on the WWW involves putting it on a webserver and setting the permissions appropriately.

You have access to a webserver within Aberystwyth University. This is at `http://users.aber.ac.uk/`. This is set up so that:

- If you have a folder called public_html in your home directory, and
- If you have web pages in that directory, and
- If those web pages have the right permissions ...
- Then that website will be visible at `http://users.aber.ac.uk/xyz25/` where xyz25 is your aber userid.

This practical talks you through how permissions work (this may be a recap for those who are familiar with unix/linux), and asks you to create a web page on the Aberystwyth University users server.

2 Setting public_html web permissions

2.1 Logging on to the central UNIX machine

You can use a standard workstation PC to log in to the central UNIX machine, using Putty. Select "Central" from the machines that you're given in the list or enter central.aber.ac.uk in the text field.

This will start a login 'shell', or terminal session, on Information Service's central computer. You should enter your userid (e.g. xyz25) when asked. You will then be asked for your password. This is the same password as you normally enter when you log into any workstation around the campus.

From another UNIX (or Linux) machine, you can access central (from anywhere in the world) by typing:

- `ssh userid@central.aber.ac.uk`

(where `userid` is your Aberystwyth userid).

2.2 Finding the directory where your web pages are

Type:

- `cd public_html`

to change your directory to where your web pages are. You can look at the contents of this directory from the command line by typing

- `ls`

which will show you the same files as if you looked at the directory using the Windows explorer. If it does not, you've probably done something wrong!

2.3 Setting the permissions

Use the Unix command `chmod` to set file permissions. In case you aren't familiar with chmod, there is a brief introduction at the end of this guide.

To set the permissions on ALL files under this directory, and on subdirectories of this directory, so that they will be visible to anyone in the world (this is what you want!), type:

- `chmod -R o+rx *`

Don't do the above if you have any PHP scripts though - that would cause them to fail.

Directories require execute permissions, and are commonly given read permissions (which enables them to be browsed if there is no index.html or similar file in place). Files need only read permission, in order to be visible to the world. So if you have no subdirectories, then

- `chmod -R o+r *`

will do. Replace `*` with a file name to set permissions on an individual file.

2.4 PHP scripts

PHP works differently. The web server will refuse to serve the page if it has read permissions for anyone other than the owner. The owner must have full permissions.

The easiest way to do this is:

- `chmod 700 *.php`

Using the symbolic method, we could do it like this:

- `chmod go-rwx u+rwx *.php`

This removes read, write and execute permissions for (g)roup and (o)ther, whilst ensuring that (u)ser (owner) has full rwx permissions.

The command:

- `fixwebperms`

will also set the permissions correctly, but you have no control over what it does, and there might well be files in your public_html that you don't want others to see, so be careful using this command!

3 A brief introduction to `chmod`

The Unix command `chmod` lets you set access permissions for three classes of user:

- the file owner or **user** (u)
- the owner's **group** (g)
- and all **others** (o).

Note that o means others (everybody), *not* owner! You can set permissions to

- read (r)
- write (w)
- execute (x).

There are two alternative ways of telling `chmod` what permissions you want to set, symbolic and numeric.

Using the symbolic approach, you use the letters given above in the parameters to add (+), remove (-) or set (=) permissions. So giving the command to set permissions for the PHP files from earlier,
`chmod go-rwx u+rwx *.php`,
we are removing all three permissions for group and others, and adding them for the user.
This is a wee bit cumbersome, so the numerical alternative is available to save typing. We give each permission a number, so read (r) = 4, write, (w) = 2, execute (x) = 1 and no permission = 0. We add these to get the comination of permissions we want so read and write is 6, for example. We then define user, group and others' permissions using three figures in that order, so
`chmod 644`
gives the user read and write permissions and everyone else (group and others) read permission only.

3.1 Making your first webpage

3.1.1 Create an empty file

(Leaving putty/terminal open so you can fix permissions later)

Using a text editor of your choice, create a file called 'index.html' in your *public_html* directory.

3.1.2 Put some HTML in it

In this file type the following:

```
<html>
<head>
<title>My first webpage</title>
</head>
<body>
<h1>Hello, world</h1>
</body>
</html>
```

and then save your work.

3.1.3 Set the permissions

In your putty or terminal window set the file index.html so that it is world readable.

3.1.4 Test your work

IF this has worked ok you should be able to view your new webpage.

Using a web browser of your choice go to:
`http://users.aber.ac.uk/\emph{xyz25}`
(replacing xyz25 with your username).

3.2 Logging out

That's it. You can now log out of central by typing

- `logout`

3.3 All done!

This should have been a quick practical - if you're finished early, you can play with the website a bit, or just do the quiz and leave early.