

# Operating systems fundamentals - B10

David Kendall

Northumbria University

- Basics of protection and security in UNIX
  - Users
  - Groups
  - Files
  - Permissions
- `adduser`, `addgroup`, `groups`
- `umask`
- `chmod`, `chown`, `chgrp`
- `sudo`

# UNIX users and groups

- Usually, in order to use a UNIX system, you need a *user account* on the system
- A user account can be created by a system administrator using the `adduser` command, e.g.

```
$ sudo adduser fred
```

will create a new user with user name `fred`

- The new user will be given
  - a *home directory* containing files copied from `/etc/skel`
  - a unique *user id (uid)*, usually in the range 1000-9999
  - a unique *group id (gid)*, usually the same as the uid
  - The details are determined by `/etc/adduser.conf`

# UNIX users and groups

- A user can belong to more than one group
- You can find out which groups a user belongs to using the `groups` command, e.g.  

```
$ groups cgdk2
```

```
cgdk2 : cgdk2 cdrom sudo plugdev lpadmin
```
- You can add a new group like this  

```
$ addgroup nufc
```

which will add a new group called `nufc`
- Users can be added to this group like this  

```
$ adduser cgdk2 nufc
```
- Check that the user has been added to the group  

```
$ groups cgdk2
```

```
cgdk2 : cgdk2 cdrom sudo plugdev lpadmin nufc
```

# Files and permissions

- Any files or directories created by a user are *owned* by the user, i.e. associated with user's uid, and are associated with the primary group (gid) of the user, e.g.

```
$ cat >fred.txt <<<"Hello world"
```

```
$ ls -l fred.txt
```

```
-rw-rw-r-- 1 cgdk2 cgdk2 12 Mar 20 07:19 fred.txt
```

- There are 9 permission bits associated with the file

User (owner)	Group	Other
rw-	rw-	r--

which mean that the *user* that owns the file can read and write it, any user in the *cgdk2 group* can read and write it, anyone who is not the owner and not in the *cgdk2 group*, i.e. an *other*, can only read the file

# Files and permissions

- In addition to the read and write bits, there is an *execute* bit, e.g.  
\$ ls -l hello  
-rwxrwxr-x 1 cgdk2 cgdk2 12 Mar 20 07:19 hello  
the x in the third bit position of each class indicates that the file is *executable* by members of that class, i.e. user, group and other
- The permissions associated with a file can be changed using the **chmod** command, e.g.  
\$ chmod g-w hello  
\$ ls -l hello  
-rwxr-xr-x 1 cgdk2 cgdk2 12 Mar 20 07:19 hello  
Notice that now only the user (owner) of the file can write it
- u, g and o are used to indicate the class, and + and - are used to add or take away the permission, e.g.  
\$ chmod go-x hello  
\$ ls -l hello  
-rwxr--r-- 1 cgdk2 cgdk2 12 Mar 20 07:19 hello

# Files and permissions

- The permission bits have slightly different meanings for directories
  - `r` - can list directory
  - `w` - can create and remove files from directory
  - `x` - can `cd` to directory

- By default, a command to create a file, e.g. `open`, requests

```
rw-rw-rw-
```

as the permissions

- A command to create a directory, e.g. `mkdir`, requests

```
rwxrwxrwx
```

as the permissions

- The defaults can be modified with a global mask created by the `umask` command, e.g.

```
$ umask 002
```

```
$ mkdir bert
```

```
$ ls -ld bert
```

```
drwxrwxr-x 2 cgdk2 cgdk2 4096 Mar 20 08:32 bert
```

# Files and permissions

- File permissions can also be indicated using 3 digits between 0 and 7, e.g.

```
$ chmod 754 hello
```

```
$ ls -l hello
```

```
-rwxr-xr-- 1 cgdk2 cgdk2 12 Mar 20 07:19 hello
```

- The permissions are determined as follows

User (owner)	Group	Other
7	5	4
111	101	100
rwx	r-x	r--

- Each digit corresponds with a class of user (user, group, other) and each bit in the digit corresponds with a permission (read, write, execute)
- A 1 bit indicates that the permission is granted, a 0 bit indicates that the permission is not granted



# Files and permissions

- The owner of a file can be changed using the `chown` command, e.g.

```
$ sudo chown fred hello
```

```
$ ls -l hello
```

```
-rw-rw-r-- 1 fred cgdk2 12 Mar 20 07:19 hello
```

makes `fred` the owner of the `hello` file

- Permissions are determined in the order 1. User 2. Group and 3. Other, e.g. assume the file properties for `hello` are

```
-r--rw-r-- 1 fred nufc 12 Mar 20 07:19 hello
```

and both `fred` and `cgdk2` are in the group `nufc`

- When logged in as `fred` we cannot write `hello` (user (owner) has no write permission and even though `fred` is in a group (`nufc`) that does have write permission, that is trumped by the lack of owner permission)
- When logged in as `cgdk2` we can write the `hello` (we're not the owner and we're in a group (`nufc`) that has write permission)