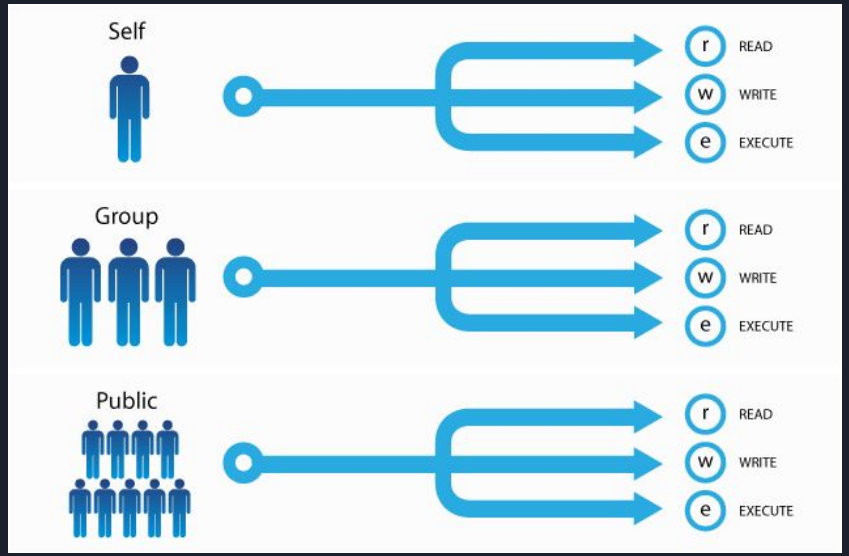
File perms, ownership, attributes

I swear its importante

Why bother?

- Restrict your information
- Only correct users should be able to view certain things

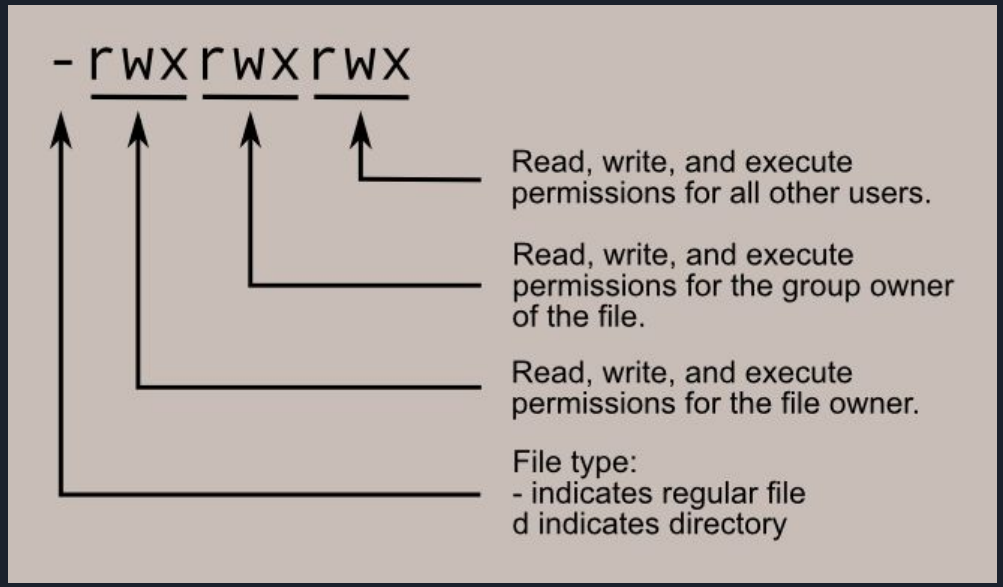


Example

```
joseph@pop-os:~/linux_examples/permissions$ ls -l
total 8
-rwxrwxrwx 1 christo christo    0 Jun 23 14:20 arch_linux
drwxr-xr-x 2 joseph  joseph  4096 Jun 23 14:21 crypto_books
drwxr-xr-x 2 christo christo  4096 Jun 23 14:21 ctf_stuffs
-rw-r--r-- 1 joseph  instructors    0 Jun 23 14:20 how_to_teach_for_dummies.pdf
joseph@pop-os:~/linux_examples/permissions$
```

What's this stuff →

```
-rwxrwxrwx 1
drwxr-xr-x 2
drwxr-xr-x 2
-rw-r--r-- 1
```



How is this represented?

- Ex: RWXR_XR_X
  - Let's take a look at the first trio: RWX
  - 1 is on, 0 is off
  - RWX → 111
    - 111 base 2 = 7 base 10
  - R_X → 101
    - the write perm is not set, so it's a 0
    - 101 base 2 = 5 base 10
  - put it all together
    - RWX = 7
    - R_X = 5
    - R_X = 5
    - → 755

Practice exercise

```
-rw--wx--x 1 joseph  joseph          0 Jun 23 14:33 file1
-r--r---w- 1 joseph  joseph          0 Jun 23 14:33 file2
-rwxr-x-wx 1 joseph  joseph          0 Jun 23 14:33 file3
---------- 1 joseph  joseph          0 Jun 23 14:33 file4
```

- What are the numerical values for these 4 permissions?

Chmod command

- Syntax: chmod [permissions] [file]
- Try:
  - Create chmod_ex.txt
  - Owner: can read, write, execute
  - Group: can read, execute
  - Everyone: can’t do anything
- Verify with ls -l

Example problem

- touch chmod_ex.txt; chmod 750 chmod_ex.txt
- ls -l
- Verified result:

```
-rwxr-x--- 1 joseph  joseph          0 Jun 23 14:38 chmod_ex.txt
```

Other chmod method

- u: user
- g: group
- o: other
- +: add
- -: remove
- Ex: chmod ugo+rwx file1
  - is the equivalent of chmod 777

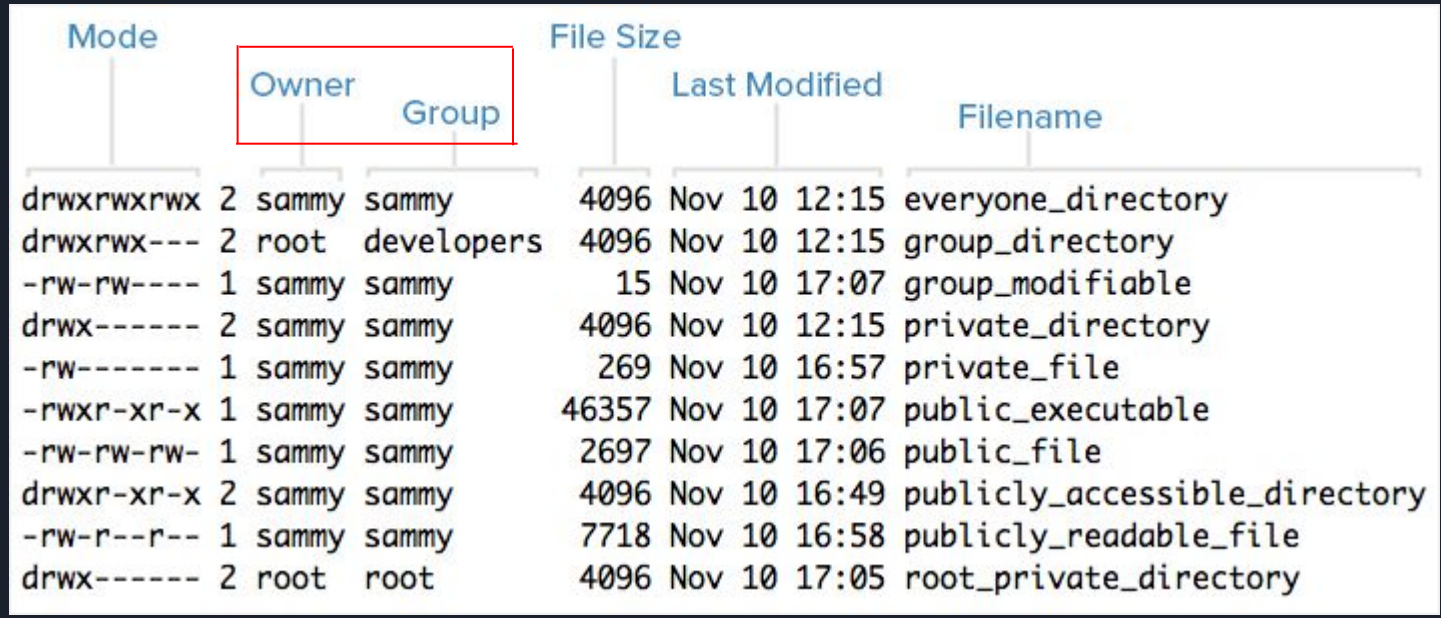
Special permissions

- SUID, SGID, Sticky bit
  - SUID: inherit permission of the owner
  - SGID: inherit permission of the group
  - Sticky bit: only owner can delete or rename
- These are also set through the chmod command

ID	File	Directory
SUID	Run program as owner of the file	-
SGID	Assign authority to run program as owner of the file	Inherit group ownership of all of the item created beneath that directory
Sticky Bit	-	Only owner of the file can delete the file e.g. /tmp

File ownership

- Files are controlled by an owner and a group



Chown and Chgrp

- Chown: change ownership
  - Can also be used to change the group
  - Ex: chown root:root file
- Chgrp: change groupo
  - Can only be used to change group. Cannot change the owner
  - Ex: Chgrp root file

Attributes

- Allows further customization of files
  - Some common ones:
    - a: append only
    - e: extent format
    - i: immutable
    - u: undeleteable

Chattr and lsattr

- chattr: change attribute
  - ex: chattr +i file
    - \+ means add the attribute
    - \- would remove it
- lsattr: list attribute
  - basically, ls but for attributes