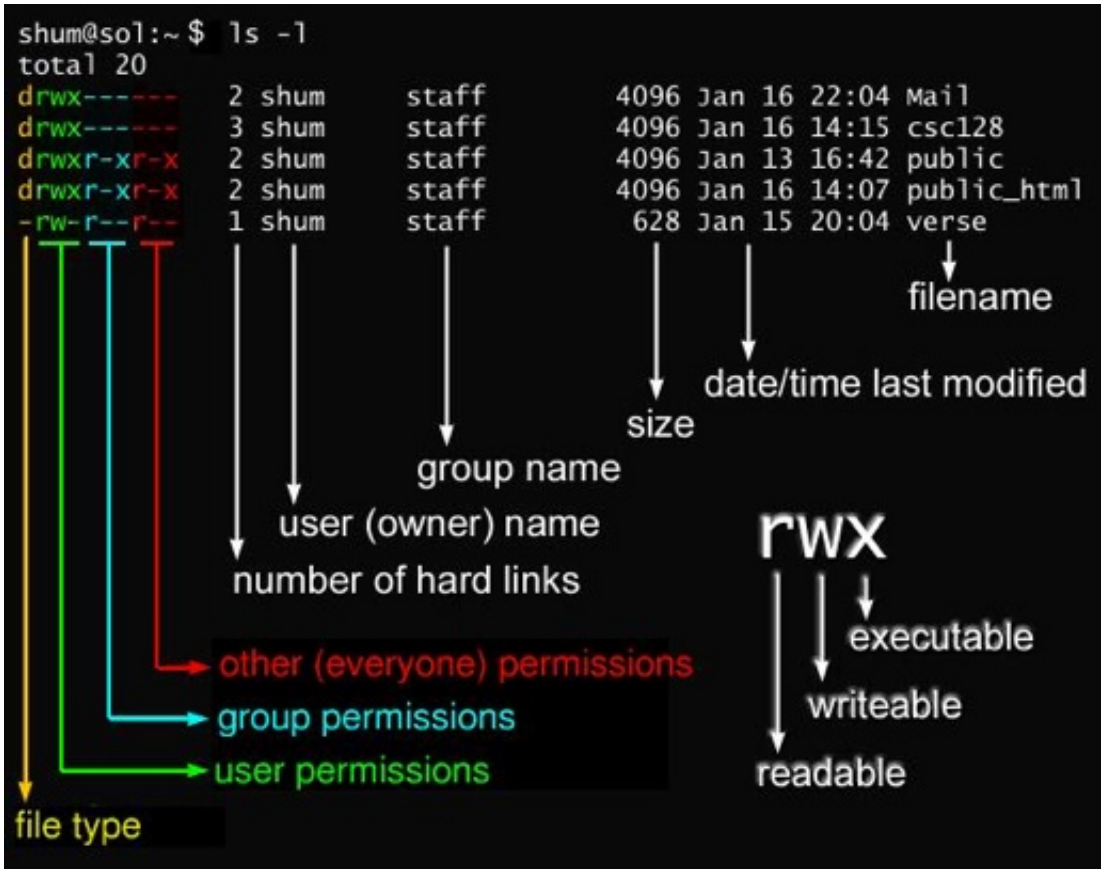
Week 1: Linux Permissions

linux,permissions

- **can view file permissions via** `ls -l`
- permissions are split in sets of 3
- **r**: read
- **w**: write
- **x**: executable
- `usr | grp | otr`
- `rwx | rwx | rwx`
- if any permission has a dash, it is not permitted

`filetype | permissions | # of hard links | user | group | other | file attributes`



Special Permissions

sticky bit (o+t)

- on shared directories, it locks files within the directory from being modified/deleted by users other than the file creator, owner of the directory, or root, even if others have write permissions
    - Ex: /tmp/firefox/...

- Set sticky bit on this directory so that user2 cannot modify files created/used by user1 (prevent conflicts/collisions)

setuid, setgid (u+s, g+s)

- “set user/group ID upon execution”
- run an executable with the permissions of the executable’s owner or group

chmod

- modify permissions for a file
  - only owner of a file or root user have permission to modify permissions of a file
- **syntax**: `chmod a-x filename`
  - removes executable permission for all users

Symbolic

Reference	Class	Description
u	user	the owner of the file
g	group	users who are members of the file's group
o	others	users who are not the owner of the file or members of the group
a	all	all three of the above, is the same as *ugo*

Operator	Description
+	adds the specified modes to the specified classes
-	removes the specified modes from the specified classes
=	the modes specified are to be made the exact modes for the specified classes

Mode	Name	Description
r	read	**r**ead a file or list a directory's contents
w	write	**w**rite to a file or directory
x	execute	e**x**ecute a file or recurse a directory tree

Numeric

#	Permission
7	full
6	read and write
5	read and execute
4	read only
3	write and execute
2	write only
1	execute only
0	none

- the # corresponds to the bit display of the rwx field
- 7 -> rwx
- 6 -> rw-
- 5 -> r-x
- …etc.