

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

The image shows a terminal window with the command `ls -l` executed. The output is a list of files with their permissions, number of hard links, owner, group, size, date, and filename. Annotations with arrows point to specific parts of the output to explain their meaning:

- file type**: Points to the first character of the permission string (e.g., 'd' for directory, '-' for file).
- user (owner) name**: Points to the user name in the permission string (e.g., 'r' for read).
- group permissions**: Points to the group permissions in the permission string (e.g., 'w' for write).
- other (everyone) permissions**: Points to the other permissions in the permission string (e.g., 'x' for execute).
- number of hard links**: Points to the number of hard links to the file.
- user name**: Points to the user name in the output.
- group name**: Points to the group name in the output.
- size**: Points to the file size in bytes.
- date/time last modified**: Points to the date and time the file was last modified.
- filename**: Points to the filename.
- rwX**: A detailed breakdown of the permissions 'rwx'. 'r' is labeled 'readable', 'w' is labeled 'writeable', and 'x' is labeled 'executable'.

file type	permissions	# of hard links	user	group	size	date/time last modified	filename
d	rwX----	2	shum	staff	4096	Jan 16 22:04	Mail
d	rwX----	3	shum	staff	4096	Jan 16 14:15	csc128
d	rwXr-Xr-X	2	shum	staff	4096	Jan 13 16:42	public
d	rwXr-Xr-X	2	shum	staff	4096	Jan 16 14:07	public_html
-	rw-r--r--	1	shum	staff	628	Jan 15 20:04	verse

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 <i>ugo</i>

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	read a file or list a directory's contents
w	write	write to a file or directory
x	execute	execute 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.