

CIS241

System-Level Programming and Utilities

Linux Permissions

Erik Fredericks, frederer@gvsu.edu

Fall 2025

Based on material provided by Erin Carrier, Austin Ferguson, and Katherine Bowers

First, aliases!

Or, how to make your life easier in the Linux world

- Less keystrokes are good for everybody
- Short command for a longer series of commands/keys

Add in `~/.bashrc` to make it persist!

- `.bashrc` runs when you login (`source ~/.bashrc` if you make changes and don't want to logout/login again)
- Example: `alias la='ls -a'`
- Depending on your distro, may go in `~/.profile` or `~/.bash_profile`
 - Lots of discussion: <https://stackoverflow.com/questions/415403/whats-the-difference-between-bashrc-bash-profile-and-environment>

Examples from the internet!

Sourced from

https://www.reddit.com/r/linuxquestions/comments/16tgbh9/what_are_your_favorite_aliases_to_use/

Useful:

- `alias up="sudo apt update && sudo apt upgrade"`
- `alias clearswap="sudo swapoff -a && sudo swapon -a"`
- `alias untar='tar -zxvf '`
- `alias eos=ssh username@eos18.cis.gvsu.edu`

Oops:

- `alias bim=vim`

File permissions

Three aspects to permissions:

- Owner - file owner
- Group - members of group associated with file
- Everybody/thing else

Three types of access:

- Read
- Write
- Execute

Check permissions:

```
ls -l
```

- Char 0: filetype (directory (**d**) or file (**-**) or link (**l**))
- Char 1-9: file permissions
- ACL flag: # links, owner, group, size, mod date, filename

File permissions:

- **rwX**: Read, Write, eXecute
- 3 characters for each (owner, group, other)
 - Example: **-rwxr-x---**
 - File where owner has full permissions, group has read/execute, other has none

File permissions

Also can be written numerically (octal):

- Read: 4
- Write: 2
- Execute: 1

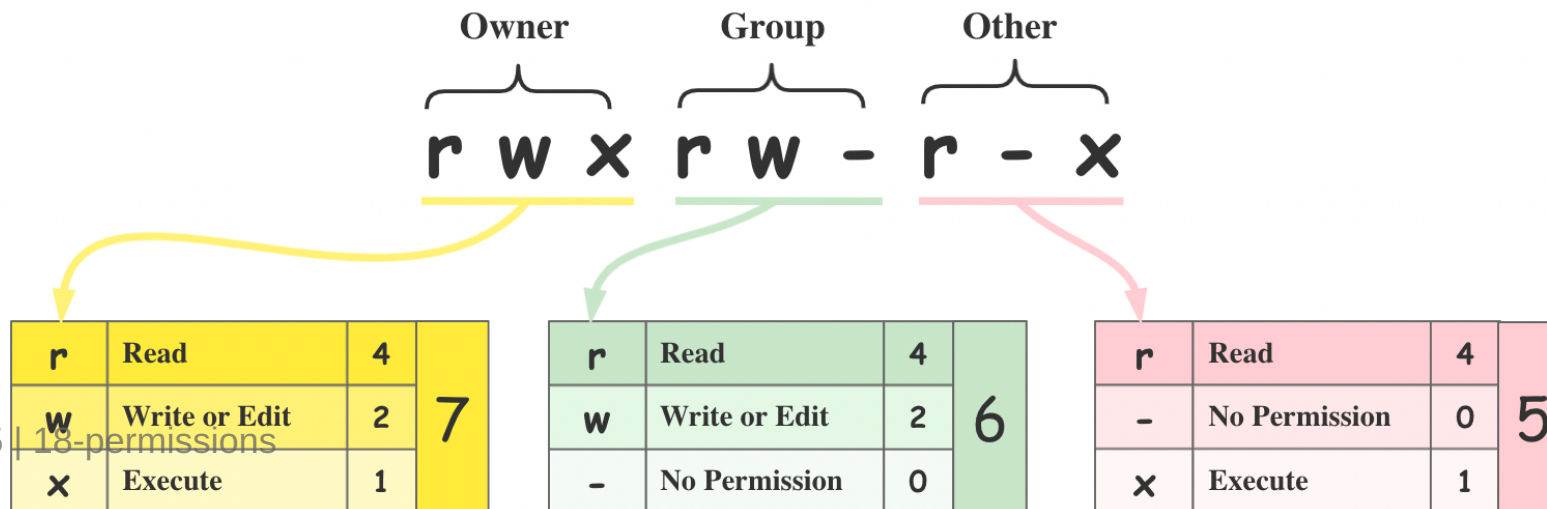
Sum to get the combination (min of 0, max of 7)

`rwX` : $4 + 2 + 1 = 7$

`r-x` : $4 + 0 + 1 = 5$

Linux File Permissions

Binary	Octal	String Representation	Permissions
000	0 (0+0+0)	---	No Permission
001	1 (0+0+1)	--x	Execute
010	2 (0+2+0)	-w-	Write
011	3 (0+2+1)	-wx	Write + Execute
100	4 (4+0+0)	r--	Read
101	5 (4+0+1)	r-x	Read + Execute
110	6 (4+2+0)	rw-	Read + Write
111	7 (4+2+1)	rwX	Read + Write + Execute



{
d = Directory
- = Regular File
l = Symbolic Link
}

File Type

{
r = Readable
w = Writeable
x = Executable
- = Denied
}



-

rwX



User

rwX



Group

rwX



Others

Modifying permissions

chmod

- `+` : add permission
- `-` : remove permission

Examples:

- `chmod u+x filename`
 - Give `filename` owner execute permission
- `chmod o+rw filename`
 - Give `filename` other (not owner/group) read/write permission
- `chmod 755 filename`
 - Give `filename` `rwX r-X r-X` permission

Environment variables

PATH : where shell looks for programs to execute

- You may have had to modify **PATH** in the past for things like, 'where is Java?' (or Python)

To modify:

- Open `~/.bashrc`
- add `export PATH="newpath:$PATH"`
 - Adds your **newpath** onto the existing **\$PATH** string

Searched in the order listed

How to check **\$PATH specifically?**

Print all environment variables - `printenv`

Installing software from source

Why do this?

- Working through an example: `sl`
- First, let's clone the repo:
 - `git clone git@github.com:mtoyoda/sl.git`
- If you don't have ssh keys setup on EOS:
 - `git clone https://github.com/mtoyoda/sl`
- Build the application, if necessary
- Modify `PATH` to include this directory