Linux Commands

— continuation

File permissions & ownership

Permission Structure

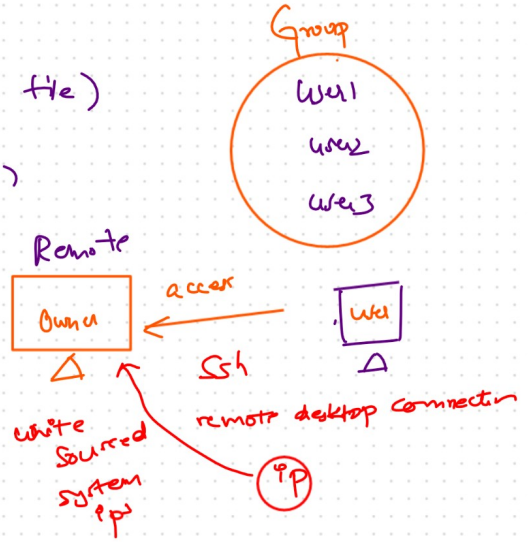
format: drwxrwxrwx

drwxrwxrwx
① ②

* first character : file type (d = directory
  - = file)
* Next 9 characters : permissions in groups of 3

Permission Groups

(1) Owner (user who owns the file) — root
(2) Group (users in the same group)
(3) others (everyone else)



Permission Types :

* r (read) = 4
* w (write) = 2
* x (execute) = 1

$ chmod — Change File Permissions

purpose : modify file & directory permissions

Syntax : chmod [permissions] [file/directory]

Examples

① # owner : rwx , Group rx , others : rx

4+2+1 4+1 4+1

7 5 5

```
-rw-rw-r--  1 nandyalapsai3443 nandyalapsai3443    0 Jun 30 19:16 code.py
```

$ Chmod 755 code.py

```
-rwxr-xr-x  1 nandyalapsai3443 nandyalapsai3443    0 Jun 30 19:16 code.py*
```

② # Owner : rw , Group : r , others : r

4+2 4 4

6

$ chmod 644 code.py

③ # Full permissions for everyone

· 4+2+1 =>7

$ Chmod 777 Code.py

④ # Read only for everyone

$ chmod 444 code.py

⑤ + add execute permisim

$ Chmod +x script.sh

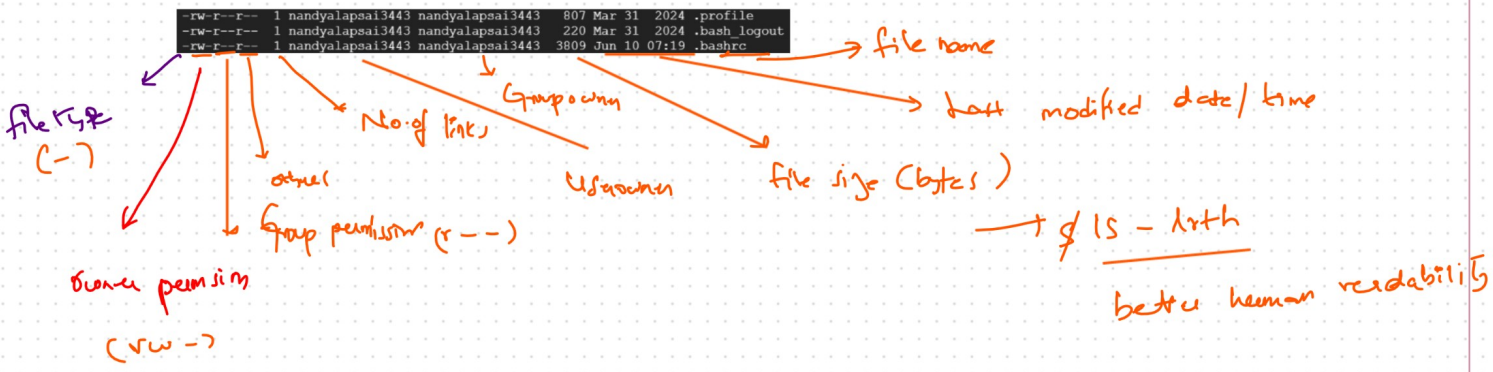
* r (read) = 4
* w (write) = 2
* x (execute) = 1

Common Permission Combinations

* 755 : Standard for executable files
* 644 : Standard for regular file
* 777 : Full access (use carefully)
* 600 : private file (owner only)

$ ls -l output ↓

System Information : file & directory Details



→ $ ls -lrth

better human readability

Navigation Shortcuts

Command History

Up Arrow : Navigate through previous commands

Down Arrow : Navigate forward through command history

Ctrl + R : Search command History