File and Directory Permissions in UNIX/Linux

Key points with respect to permissions on files and directories

- Categorise users into **three** types:

  user-owner: person who owns the file

  group to which the user belongs

  others: also referred to as “rest of the world”

- We can refer to each of these by the first letter; i.e.

  user-owner = u

  group = g

  others = o

- Each category has **three** possible permissions:

  r = read

  w = write

  x = execute for a file; or access for directory (think aXcess!)

- Files and directories also have types, indicated by the first letter in a long listing (“ls -l”); for example:

  d = directory

  - = file

  l = link

  b = block device (for eg: hard-disk drive)

  c = character device (for eg: mouse)

Examples:

-r w -	r - -	r - -	rfilo	users	myfile.txt
File read/write for user-owner	read-only for group	read-only for others	user	group	filename

Things to note:

- **Three** letters are used to indicate permissions, r, w, x
- Each letter has its own position; i.e. “r” is always most significant, “w” is always in the middle, “x” is always least-significant
- We can represent “r w x” permissions as bits (binary-digits) in base 8 (Octal); for example:

  r w - = 1 1 0 = 6
  r - - = 1 0 0 = 4
  r - x = 1 0 1 = 5
  So, permissions: “-rw- r- - r - -” = 6 4 4
- There are two ways to set permissions on a file: use an example: Say we want to make myfile.txt **read-only** also for the user-owner:
- Change from “-rw- r- - r - -” to “-r- - r- - r - -”

  Way 1: chmod u-w myfile.txt
  Way 2: chmod 444 myfile.txt
- To change myfile.txt back to read/write for the user-owner:
- Change from “-r - - r- - r - -” to “-r w - r- - r - -”

  Way 1: chmod u+w myfile.txt
  Way 2: chmod 644 myfile.txt