Lec-04-1 File System Permissions

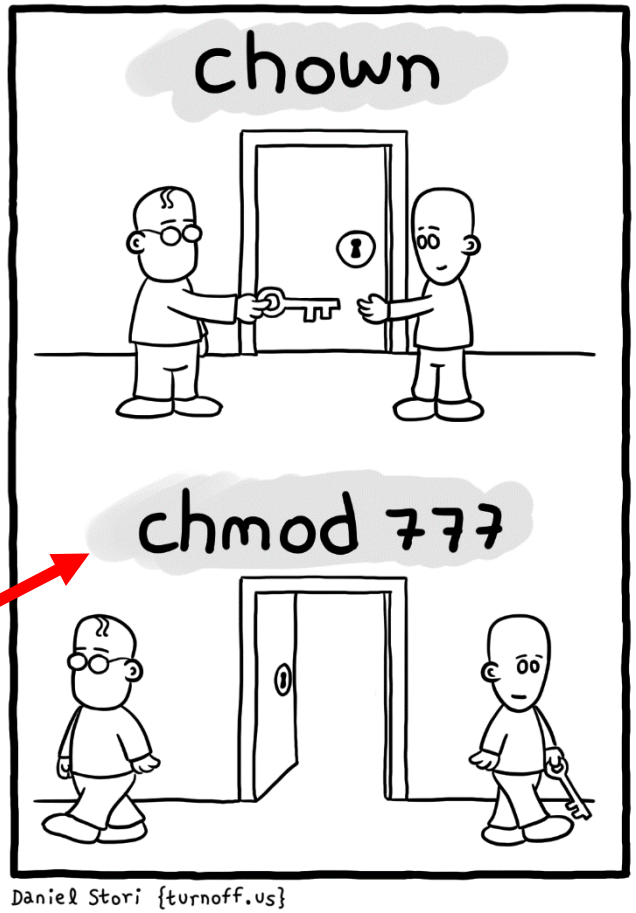
Dr Syed Faisal Hasan and Dr. Hymie Latif
Computing and Information Technology
College of Enterprise and Development
Otago Polytechnic
Dunedin, New Zealand

Bachelor of Information Technology
IN616 – Operating Systems Concepts
Semester 1, 2020

Schedule

- File system permissions
  - Levels of permission
  - Types of access
- Modifying permissions
  - chown
  - chmod

Worst method ever!



TOPIC:

Overview of File System Permissions

File System Permissions

- Controls who can access what (in the file system)

- Permissions can be divided into two principles

1. **Levels of permission**
2. **Types of access**

- We can view permissions using:

```
ls -l
ls -lisa
```

File System Permissions

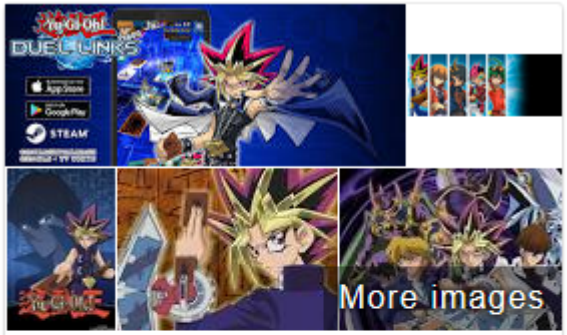
- **There are three types of permission levels**
  1. **U**ser (commonly referred to as the owner)
  2. **G**roup (usually the owners group)
  3. **O**thers (anyone else)

  – Together the make **ugo**
- **There are three types of access**
  1. **R**ead
  2. **W**rite
  3. **Ex**ecute

  – Together they make **`rwx`**

File System Permissions: Levels

- **There are three types of permission levels**
  - **U**ser (commonly referred to as the owner)
  - **G**roup (usually the owners group, not always)
  - **O**thers
  - Together the make **ugo**



File System Permissions: Access

- **There are three types of access**
  1. **R**ead
  2. **W**rite
  3. **Ex**ecute
  - Together they make **rwx**

Really **W**eird **X**ray

→ → →

File System Permissions

1. **Levels of permission**

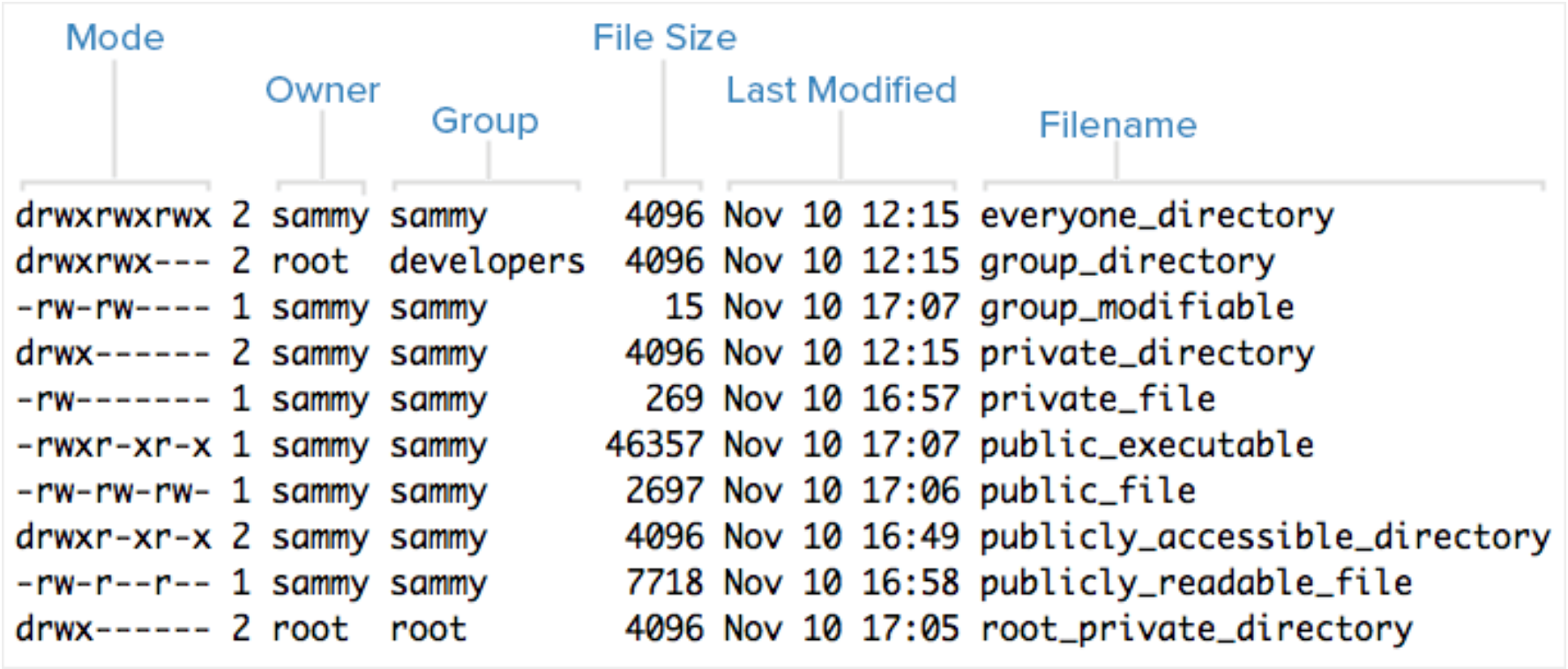
WHO CAN ACCESS THE FILE

2. **Types of access**

WHAT ACCESS DO THEY HAVE

File System Permissions: Visual

Command: `ls -l`



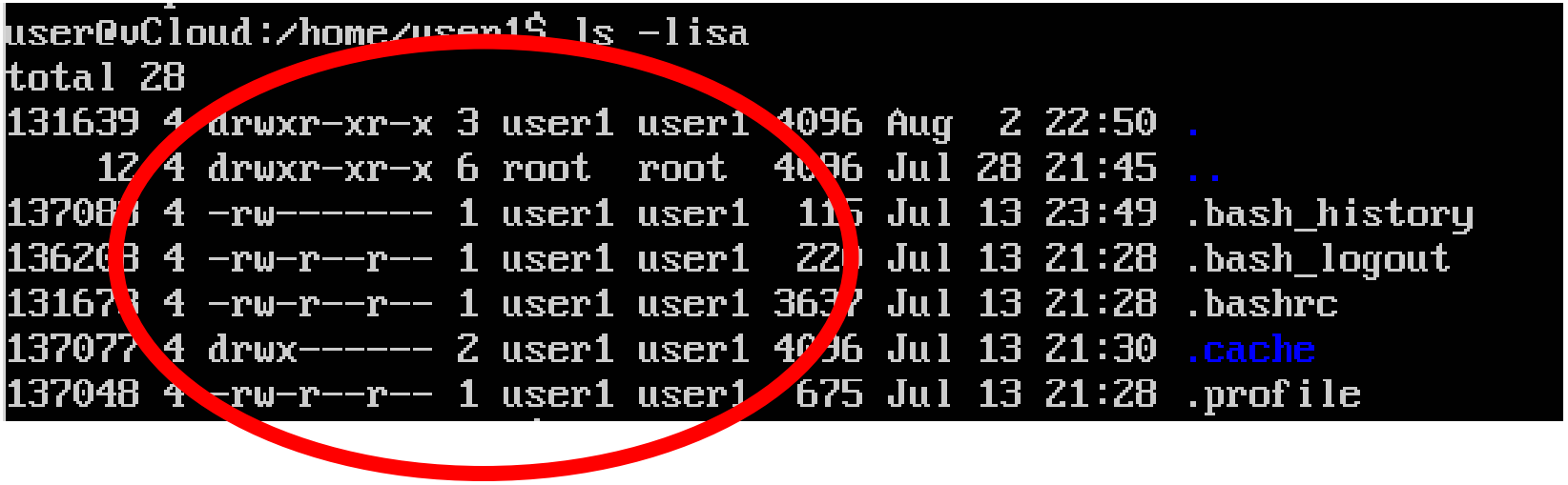
File System Permissions: Visual

User **Group** **Other**

Mode		Owner	Group	File Size	Last Modified	Filename
drwxrwxrwx	2	sammy	sammy	4096	Nov 10 12:15	everyone_directory
drwxrwx---	2	root	developers	4096	Nov 10 12:15	group_directory
-rw-rw----	1	sammy	sammy	15	Nov 10 17:07	group_modifiable
drwx------	2	sammy	sammy	4096	Nov 10 12:15	private_directory
-rw-------	1	sammy	sammy	269	Nov 10 16:57	private_file
-rwxr-xr-x	1	sammy	sammy	46357	Nov 10 17:07	public_executable
-rw-rw-rw-	1	sammy	sammy	2697	Nov 10 17:06	public_file
drwxr-xr-x	2	sammy	sammy	4096	Nov 10 16:49	publicly_accessible_directory
-rw-r--r--	1	sammy	sammy	7718	Nov 10 16:58	publicly_readable_file
dr[illegible]				4096	Nov 10 17:05	root_private_directory

Where is other?????

File System Permissions: Visual



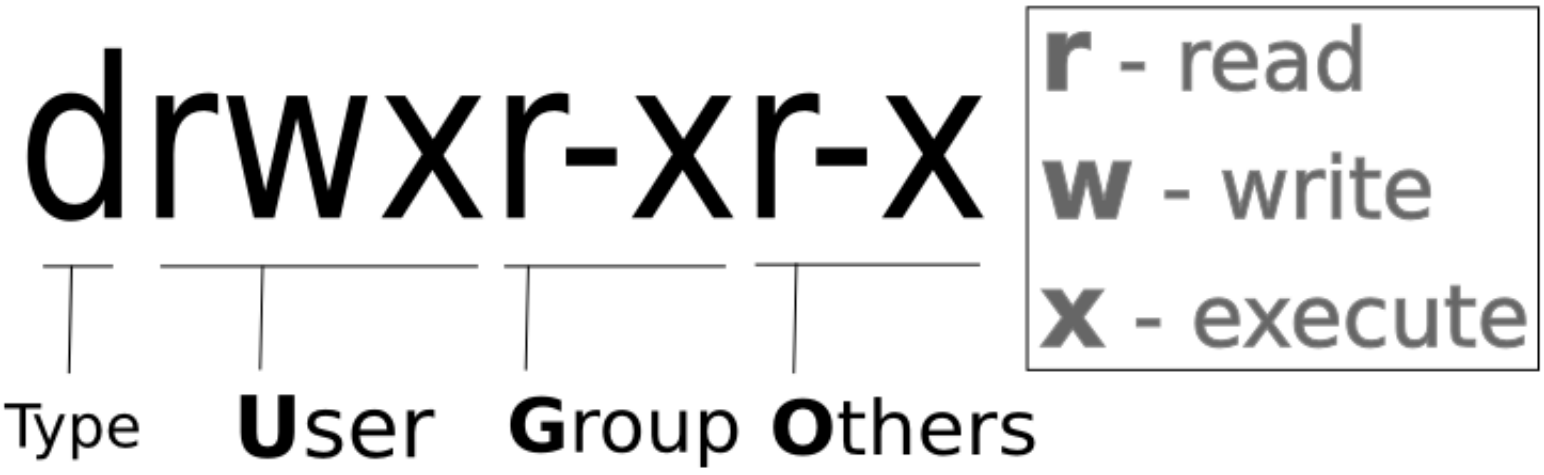
File System Permissions

```
manager@server:/home/student$ ls -lisa
total 36
   727 4 drwxr-xr-x 3 student student 4096 Aug  6 07:54 .
   792 4 drwxr-xr-x 4 root    root    4096 Jul 13 08:47 ..
  1113 4 -rw------- 1 student student  943 Jul 13 15:52 .bash_history
 38136 4 -rw-r--r-- 1 student student  220 Jul 13 08:22 .bash_logout
 36330 4 -rw-r--r-- 1 student student 3771 Jul 13 08:22 .bashrc
155078 4 drwx------ 2 student student 4096 Jul 13 08:31 .cache
 36328 4 -rw-r--r-- 1 student student  655 Jul 13 08:22 .profile
 38167 0 -rw-r--r-- 1 student student    0 Jul 13 08:31 .sudo_as_admin_successful
   312 4 -rw------- 1 root    root     875 Jul 13 08:50 .viminfo
```

1. inode
2. Size of file (in blocks)
3. Permissions set (Really Weird Xray)
4. Number of links (discussed in another lecture)
5. Owner (Yu-Gi-Oh – the user)
6. Group (Yu-Gi-Oh – the group)
7. Size (in bytes)
8. Modification date
9. Name of directory or file

Permissions in UNIX/Linux

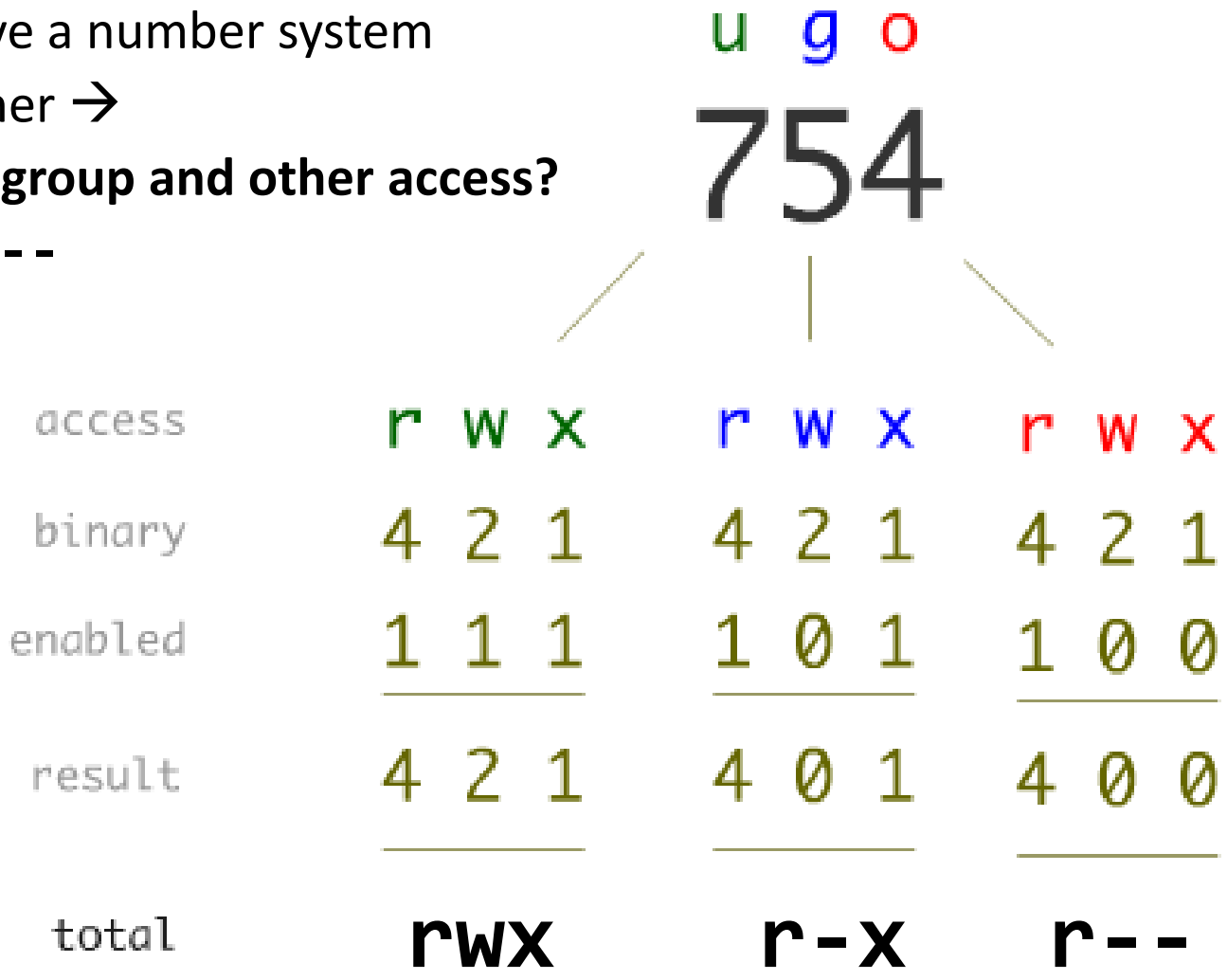
- A total of 10 characters (well 9 for permissions)
- The first character is the type (file, directory and others)



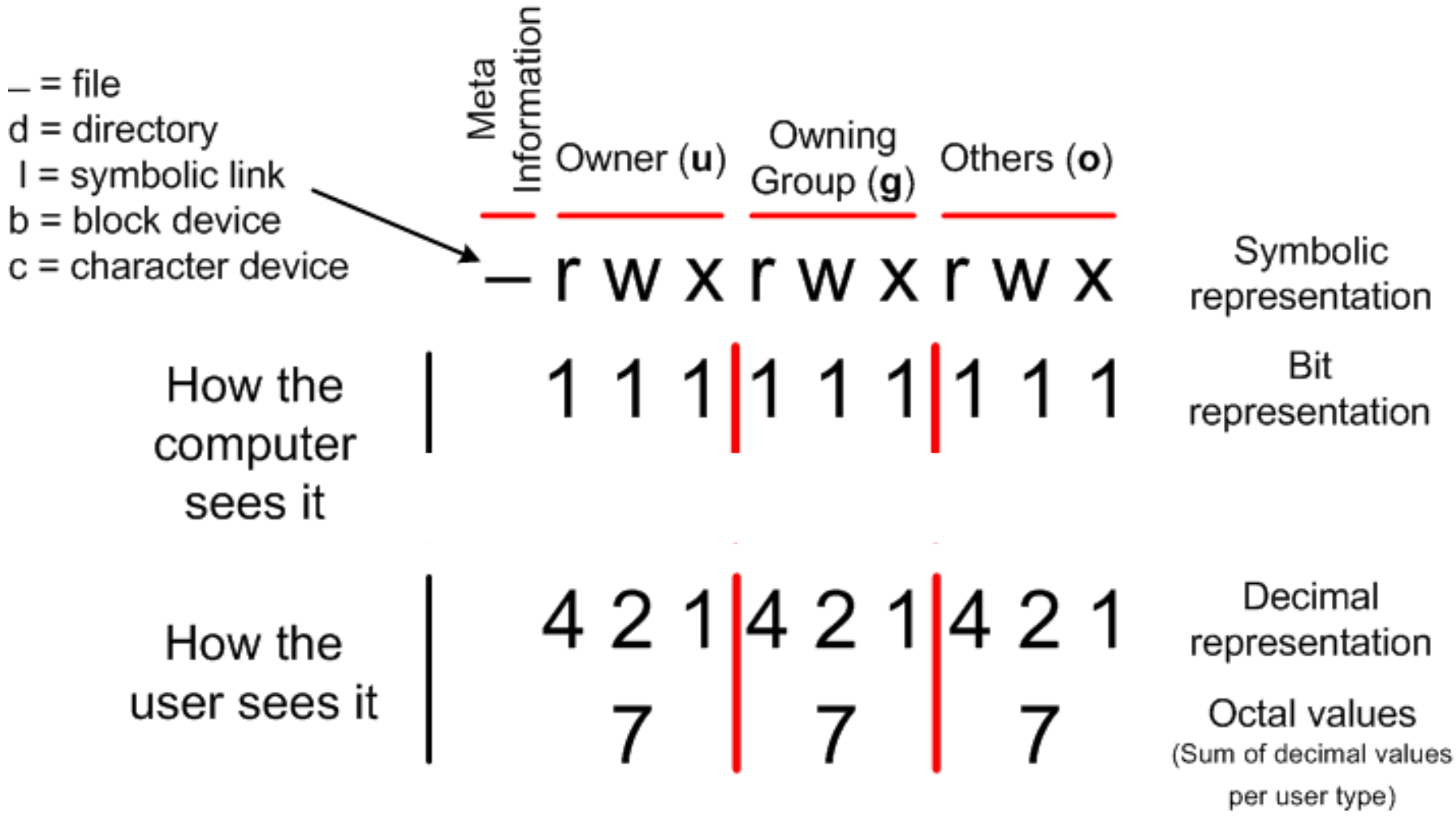
- **ENABLED**: A letter means it is enabled (can only be **r**, **w** or **x**)
- **DISABLED**: A dash **"-"** means it is disabled
- **What can user, group and other access?**

Permissions in Octal Representation

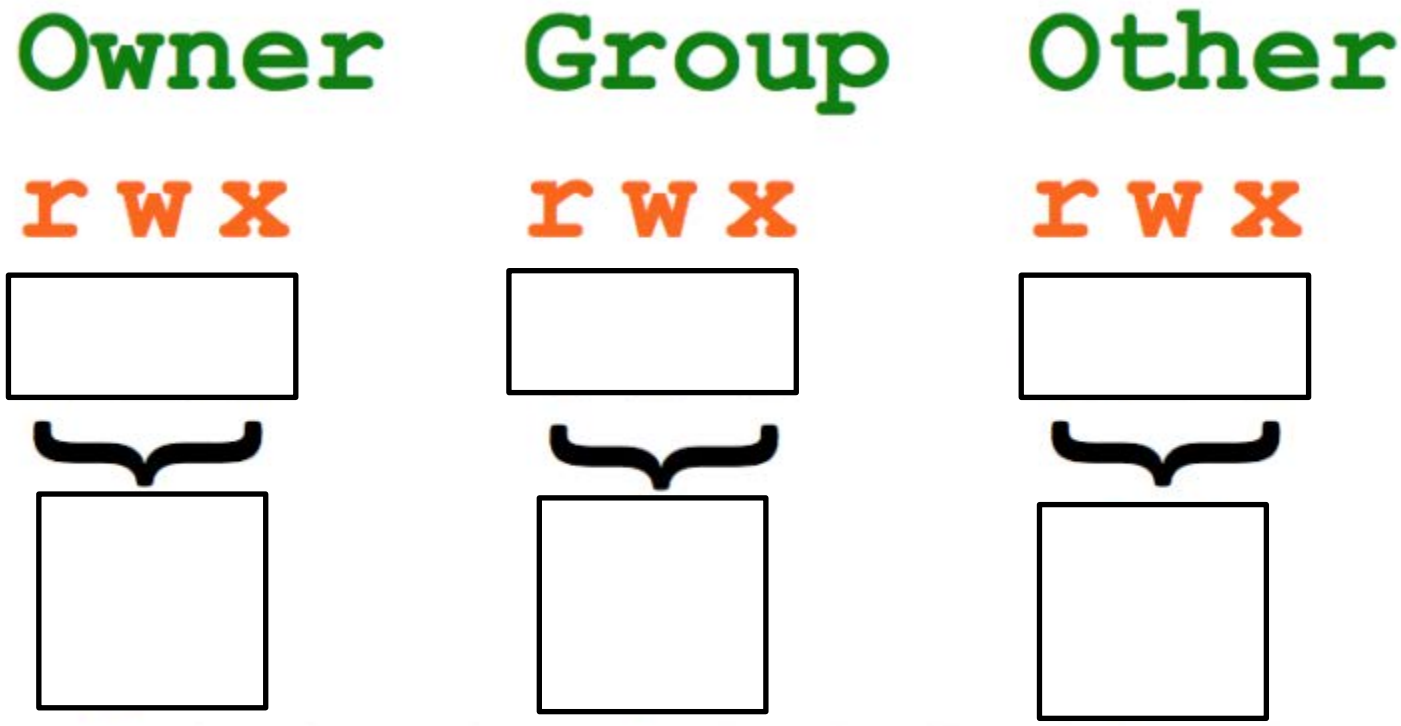
- Permissions have a number system
- User, group, other →
- **What can user, group and other access?**
- 754 = **rwxr-xr--**



Permissions in UNIX/Linux

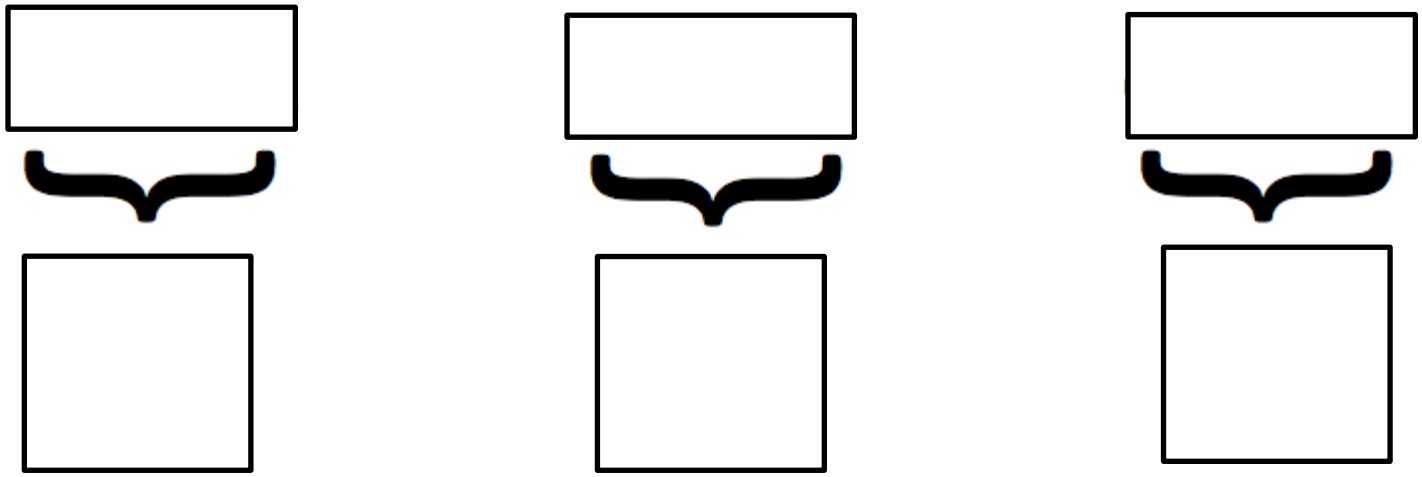


Permissions in UNIX/Linux



Permissions in UNIX/Linux

Owner	Group	Other
r w x	r - x	r - x



Permissions in Octal Representation

For each user, group, other in the permission …

There are 8 variations for each **R**ead, **W**rite, e**X**ecute (**RWX**)

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7

Permissions Table

BINARY	Decimal	Permission	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx

INTERESTING

IMPORTANT

IMPORTANT

IMPORTANT

Is Octal Important?

YES

- But there are online tools?!
  - http://permissions-calculator.org/
- It's still important
  - To fully understand permissions in Linux
  - To have better permission control
  - To understand Linux tutorials

Permissions on **Files**

- **`Read (r)`**
  - Permission to read a file
  - Example command?
  - **`cat TestFile`**
- **`Write (w)`**
  - Permission to write or modify a file
  - Deletion is managed by directory permissions
  - Example command?
  - **`touch TestFile`**
- **`Execute (x)`**
  - Permission to execute (run) a file
  - Very important for running BASH scripts (.sh files)
  - **`sh script.sh`** OR **`./script.sh`**

Permissions on Directories

- **`Read (r)`**
  - Ability to list a directory
  - Example command?
  - **`ls /etc/ssh`**
- **`Write (w)`**
  - Ability to add, delete and rename files in a directory
  - Example command?
  - **`touch /etc/ssh/TestFile`**
- **`Execute (x)`**
  - Ability to enter a directory and access files
  - Example command?
  - **`cd /etc/ssh`**

TOPIC:

Modifying:
1. Ownership
2. Permissions

Modifying Owners: **chown**

- **There are three types of permission levels**
  - **U**ser
  - **G**roup
  - **O**thers

- **chown**
  - **Ch**ange **own**ership for file or directory

Modifying Permissions: **chmod**

- **There are three types of access**
  1. **R**ead
  2. **W**rite
  3. **Ex**ecute
  - Together they make **rwx**

Really **W**eird **X**ray

→ → →

- **chmod**
  - **Ch**ange/**mod**ification of permission for file or directory

Modifying Owners: **chown**

- **`chown`**
  - Change ownership for file or directory
  - **`chown <username>:<groupname> <target>`**
  - **`chown <username> <target>`**
    - Change user ownership
  - **`chown :<groupname> target`**
    - Only change group ownership

- Examples:
  - **`chown frodo theonering.png`**
  - **`chown –R samwise recipes/`**
  - **`chown –R frodo:fellowship maps-of-Mordor/`**
  - **`chown :fellowship fellowship-members.csv`**

Modifying Permissions: **chmod**

- **chmod**
  - Change permissions for file or directory
  - **chmod <permission-set> <target>**

- Examples:
  - **chmod 600 theonering.png**
  - **chmod –R 770 recipes/**

Using chmod with octals

- **`chmod 212 theonering.png`**
  User = write, Group = execute, Other = write
  `-w---x-w-`
- **`chmod 401 youshallnotpass.txt`**
  User = read, Group = none, Other = execute
  `r-------x`
- **`chmod 777 /theShire`**
  User = full, Group = full, Other = full
  `rwxrwxrwx`
- **`chmod –R 644 maps-of-Mordor/`**
  User = read/write, Group = read, Other = read
  `rw-r--r--`
- **`chmod 511 recipes/`**
  User = read/execute, Group = execute, Other = execute
  `r-x--x--x`

Remember!

`r = 4`

`w = 2`

`x = 1`

Modifying Permissions: **chmod**

- **`chmod` can also modify single permissions**
  - *Symbolic notation* is finer-grained
  - Can modify user, group or other (ugo) or all (a)
  - Can modify read, write, execute (rwx)
  - Can add, remove or make exact permissions (+, -, =)

- **`chmod g+w painting-of-Mordor.png`**
  - Add group write permission to file
- **`chmod u=x script.sh`**
  - Only allow user execution permission to file
- **`chmod a-w do-not-edit.txt`**
  - Remove all (ugo) write permission to file

chmod and symbolic modes

Reference	Class	Description
u	owner	file's owner
g	group	users who are members of the file's group
o	others	users who are neither the file's owner nor members of the file's group
a	all	all three of the above, 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	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

Lab-04-1 – Start

- **TOPICS:**
- **Setting ownership**
- **Setting permissions**