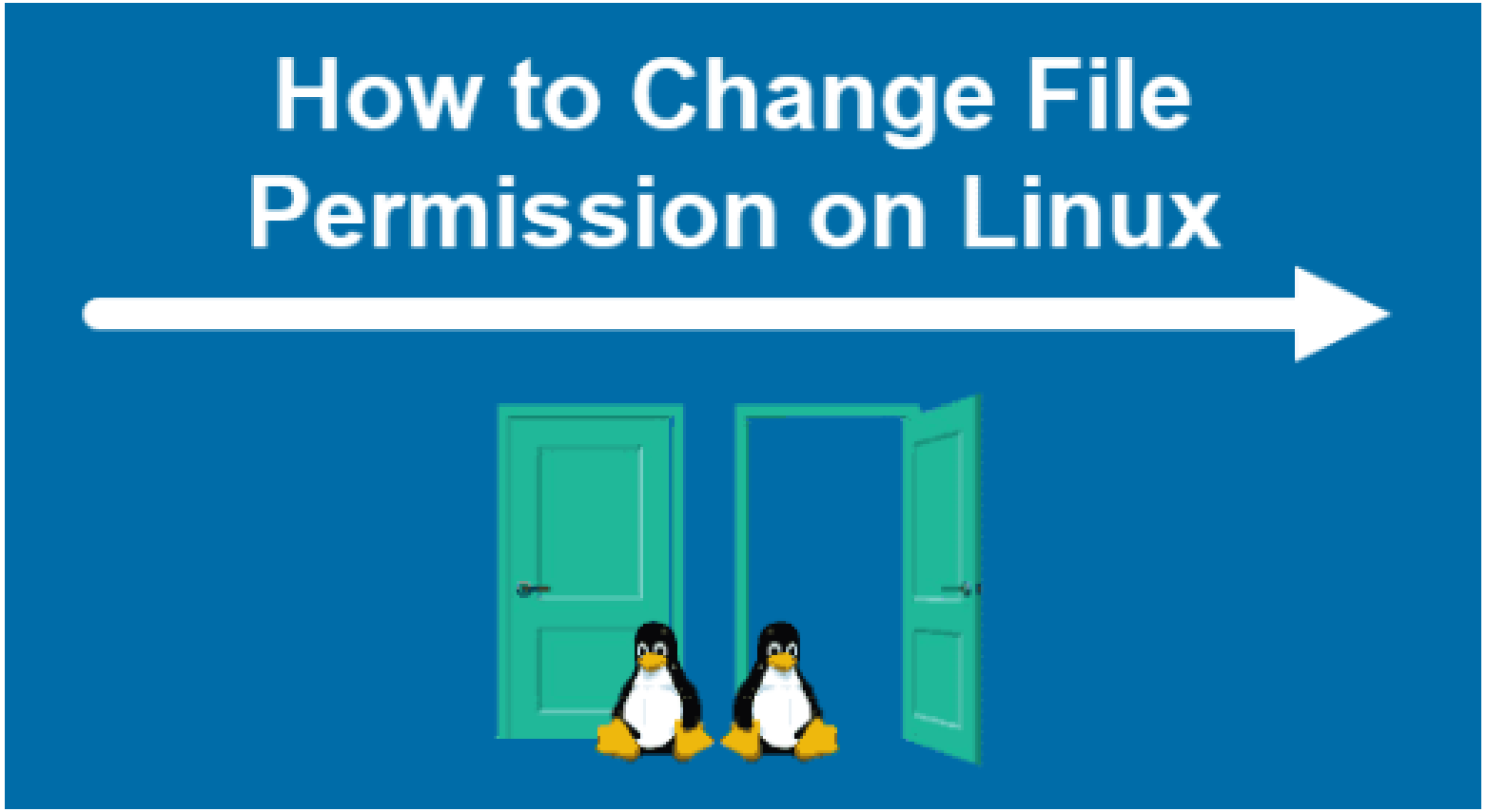
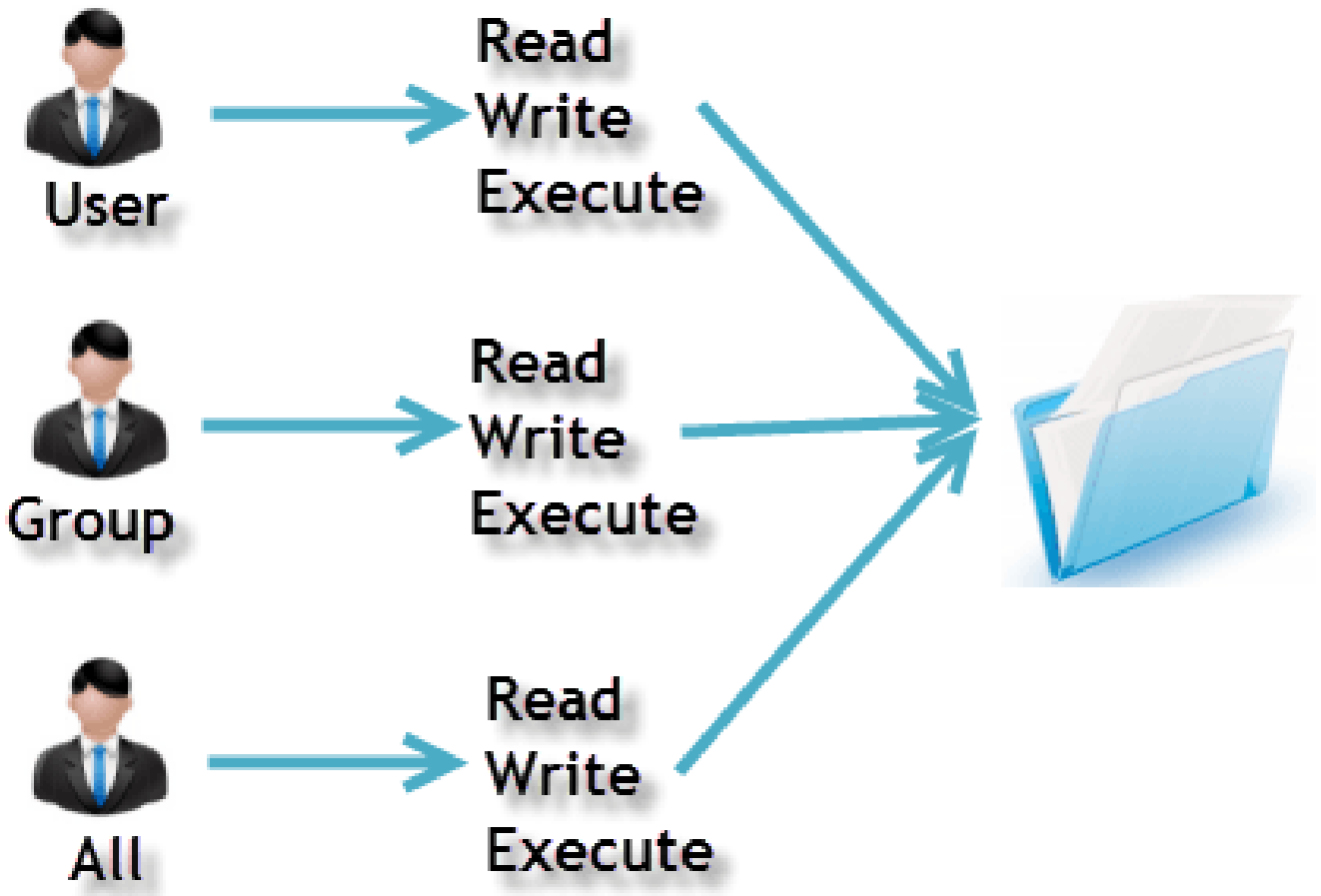
# *Lecture 6*
# *Permissions*

# File permissions

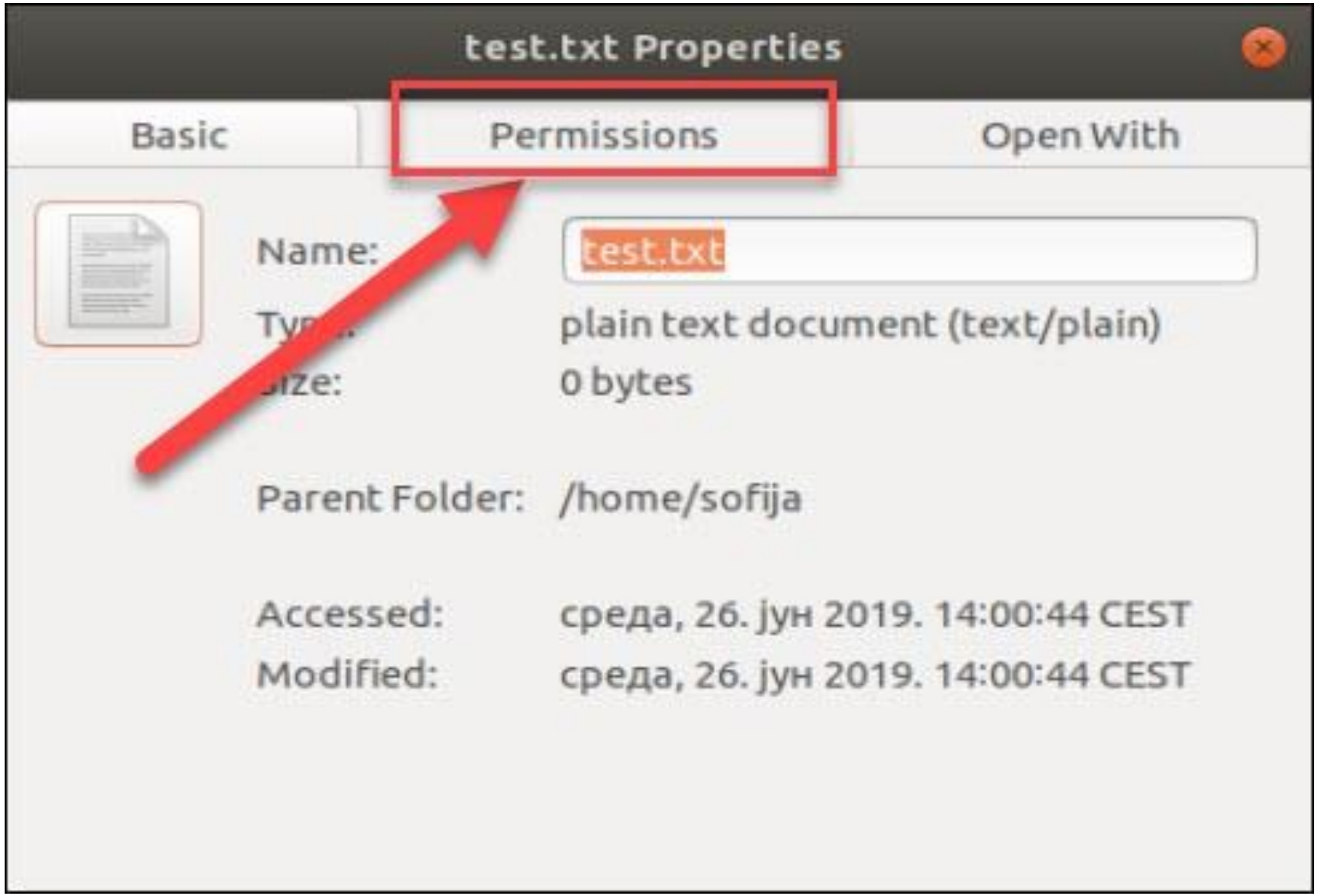


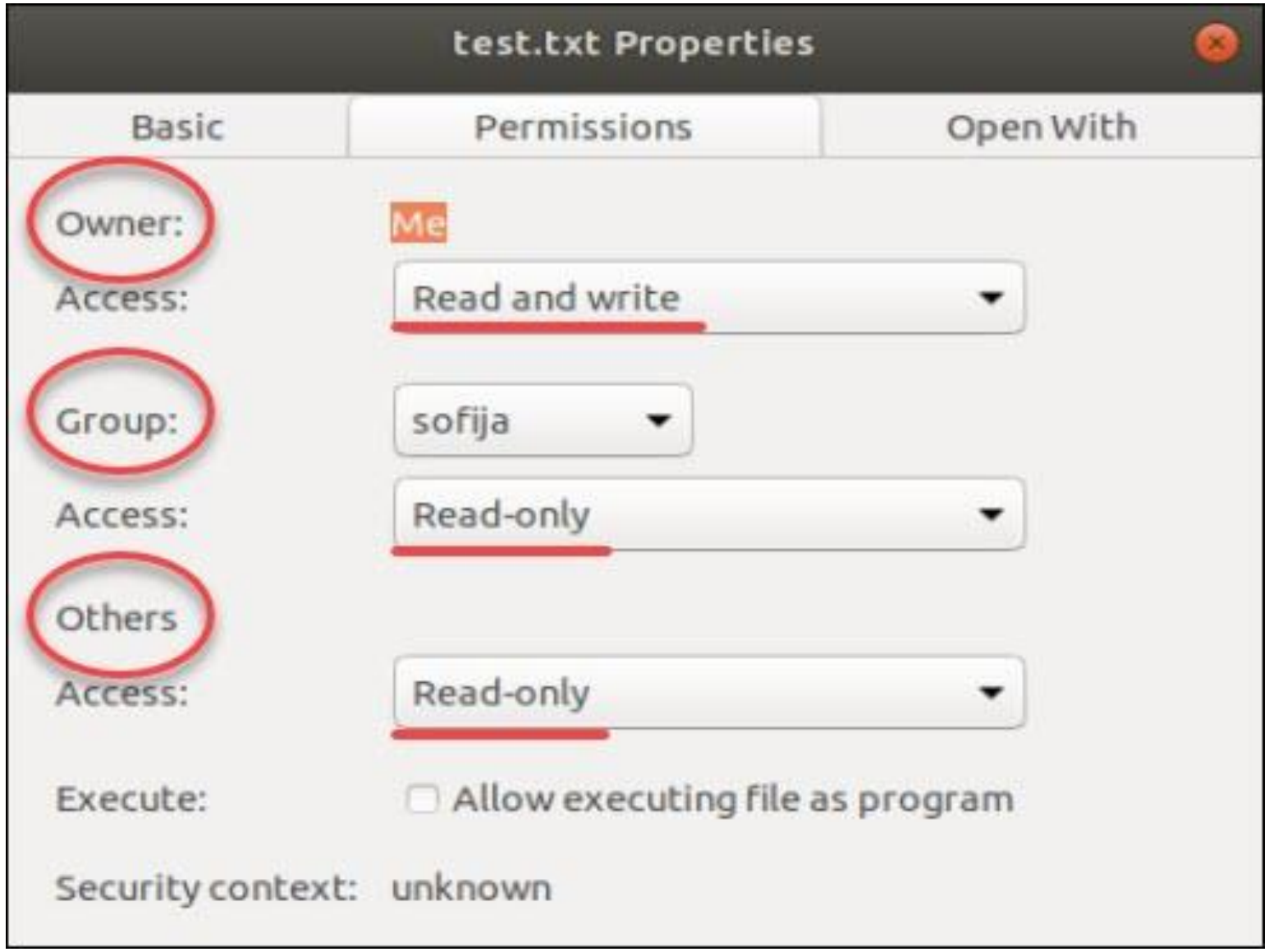
## Owners assigned Permission On Every File and Directory



# Permissions by GUI

- Finding the file (directory) permission via the graphical user interface (GUI) is simple.
- 1. Locate the file you want to examine, right-click on the icon, and select **Properties**.
- 2. This opens a new window initially showing **Basic** information about the file. Navigate to the second tab in the window, labeled **Permissions**.


test.txt Properties
Basic
Permissions
Open With
Name:
test.txt
plain text document (text/plain)
0 bytes
Parent Folder:
/home/sofija
Accessed:
среда, 26. јун 2019. 14:00:44 CEST
Modified:
среда, 26. јун 2019. 14:00:44 CEST


test.txt Properties
Basic
Permissions
Open With
Owner:
Me
Access:
Read and write
Group:
sofija
Access:
Read-only
Others
Access:
Read-only
Execute:
Allow executing file as program
Security context:
unknown

# Permission by command lines

- We have three "actors" that can perform operations on a file or directory:
- users
- groups
- other

# User – Group – Owner

- There are three levels of permission in Linux:
- 1. owner (user): is the user own the file or folder
- 2. group: means the users in the group of owner
- 3. others: all other users who aren’t the owner or in the group

# Permissions

- Users, groups, and anyone else ("other") have specific things they can do to a file or directory. These are the three permissions a user/group/other can do on a file/directory:
- read (r)
- write (w)
- execute (x)

# Files:

- Read is the ability to read the contents of a file, including open in an editor in a read-only format
- Write is the ability to modify or delete a file
- Execute is the ability to run the file as a program (e.g. a shell script, python script, php script)

# Directories

- Read is the ability to investigate a directory (ls)

- Write is the ability to add to a directory, or delete
- the directory

- Execute is the ability to `cd` into directory
- `-rwx-rw-r--` - A file. U: rwx, G: rw, O: r
- `drwx-rwx-rx` - A directory. U: rwx, G rwx, O: rx

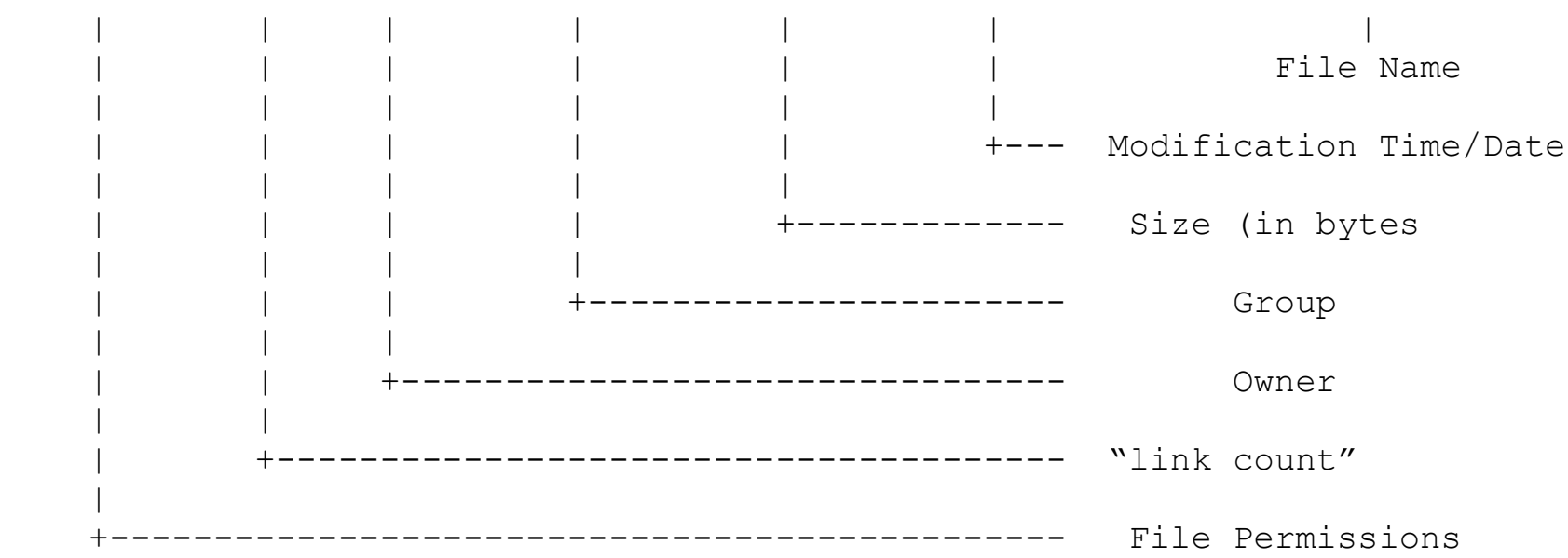
# Example

- When we type:

```
ls -l /usr/bin/top
```

- We'll see:
- -rwxr-xr-x 1 root root 68524 2011-12-19 07:18 /usr/bin/top
- What does all this mean?

**-r-xr-xr-x  1  root  root  68524  2011-12-19 07:18 /usr/bin/top**




**Group**

The name of the group that has permissions in addition to the file's owner.

**Owner**

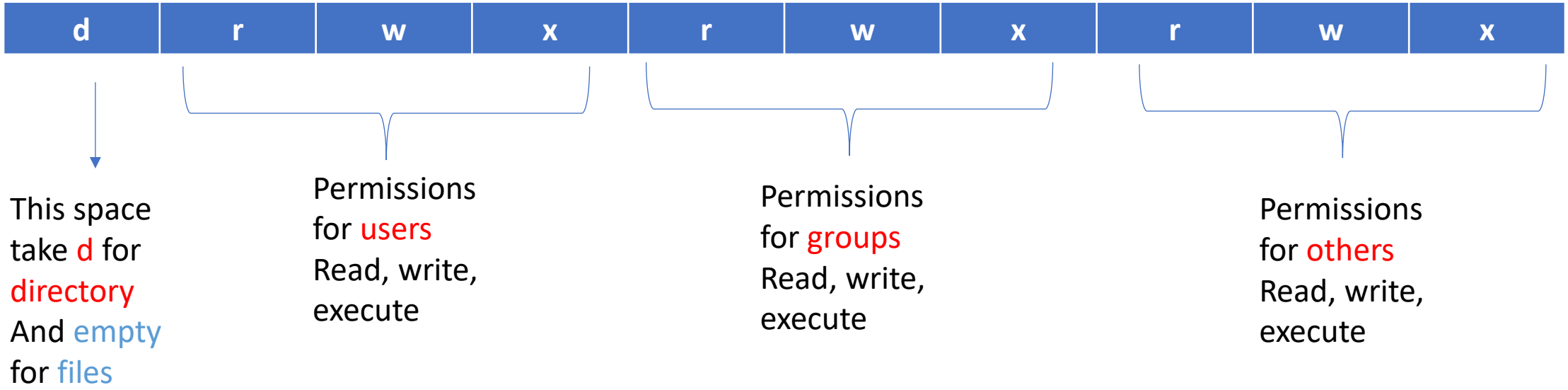
The name of the user who owns the file.

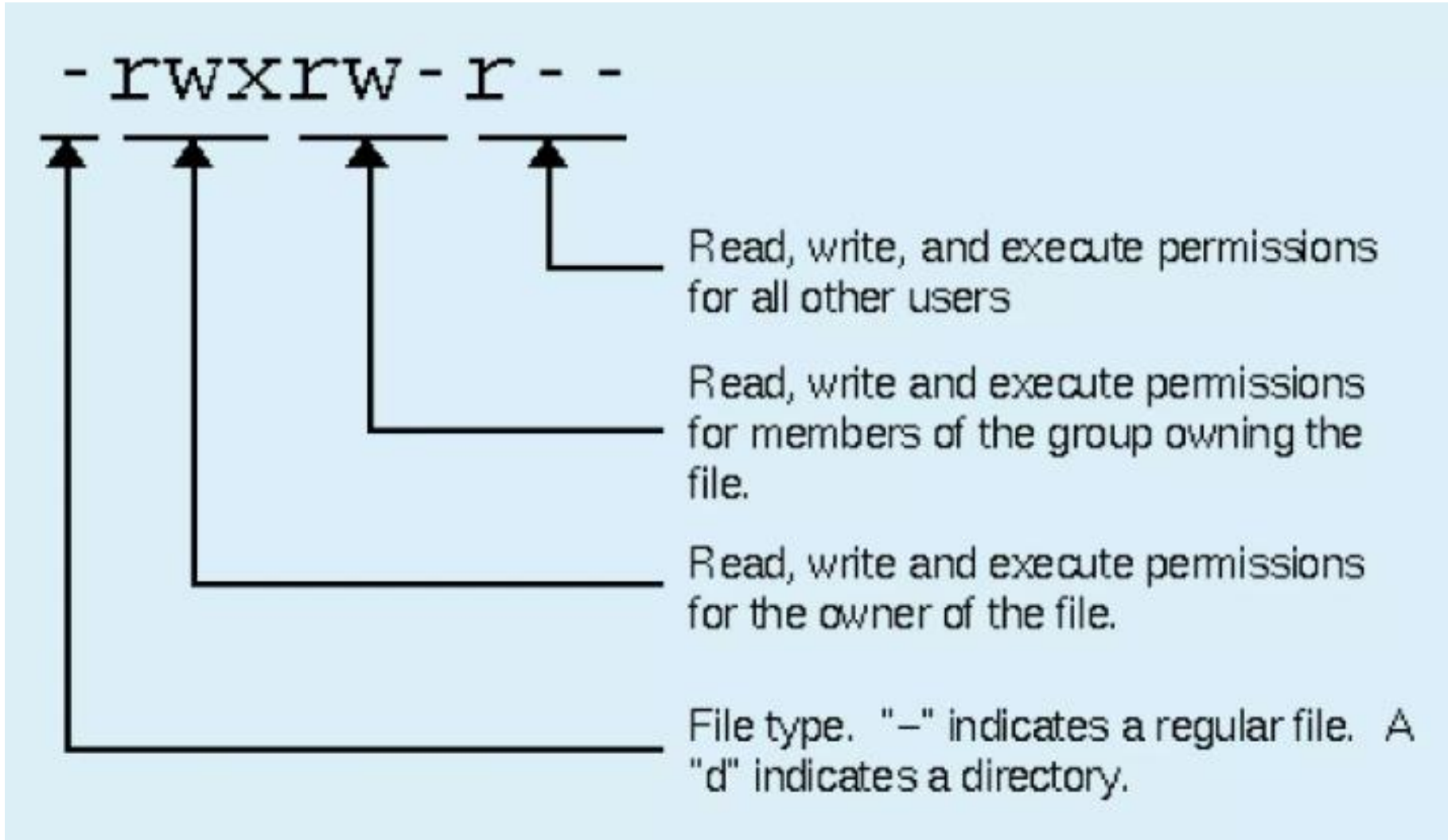
**File Permissions**

The first character is the type of file. A "-" indicates a regular (ordinary) file. A "d” indicate a directory. Second set of 3 characters represent the read, write, and execution rights of the file's owner. Next 3 represent the rights of the file's group, and the final 3 represent the rights granted to everybody else.

(Example modified from **http://www.linuxcommand.org/lts0030.php**)

# The permissions RWX




-rwxrw-r--
Read, write, and execute permissions for all other users
Read, write and execute permissions for members of the group owning the file.
Read, write and execute permissions for the owner of the file.
File type. "–" indicates a regular file. A "d" indicates a directory.

# Change the permission

- There are two ways to set permissions when using the **`chmod`** command:

1. Symbolic mode:
2. Absolute mode:

# File permissions

## 1. Symbolic mode:

*testfile* has permissions of `-r--r--r--`

```
                                    U   G   O*
$ chmod g+x testfile        ==>  -r--r-xr--
$ chmod u+wx testfile       ==>  -rwxr-xr--
$ chmod ug-x testfile       ==>  -rw--r--r--
U=user, G=group, O=other (world)
R read  , W write, X execute
```

Example
The download folder contains 3 files cap1, drop and shell
Type the commands to achieve the following

1. Add execute permission for user file: cap1 ……
2. Delete permission write for group file: drop ……
3. Add read permission for users, groups and others file: shell…….

Example

The download folder contains 3 files cap1, drop and shell

Type the commands to achieve the following

1. Add execute permission for user file: cap1 ……
   `chmod U+x cap1 …`
2. Delete permission write for group file: drop ……
   `chmod g-W drop …`
3. Add read permission for users, groups and others file: shell
   `chmod ugo+r shell`

# File permissions cont.

## Absolute mode:

We use octal (base eight) values represented like this:

| Letter | Permission | Value |
|---|---|---|
| R | read | 4 |
| W | write | 2 |
| X | execute | 1 |
| - | none | 0 |

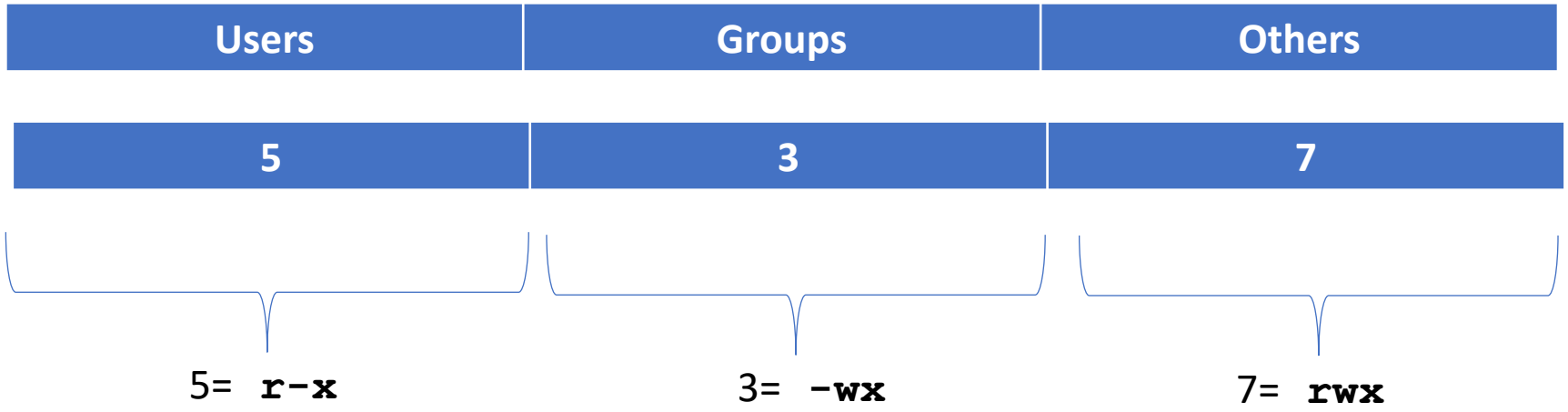
For each column, User, Group or Other you can set values from 0 to 7. Here is what each means:

0= **---** 1= **--x** 2= **-w-** 3= **-wx**

4= **r--** 5= **r-x** 6= **rw-** 7= **rwx**

0= `---`  1= `--x`  2= `-w-`  3= `-wx`

4= `r--`  5= `r-x`  6= `rw-`  7= `rwx`



# File permissions cont.

## Numeric mode cont:

Example index.html file with typical permission values:

```
$ chmod 755 index.html
$ ls -l index.html
-rwxr-xr-x  1 root  wheel  0 May 24 06:20 index.html

$ chmod 644 index.html
$ ls -l index.html
-rw-r--r--  1 root  wheel  0 May 24 06:20 index.html
```

Example
The download folder contains file cap1
Type the commands (symbolic mode) to achieve the following

1. Add execute permission for user file: cap1 ……
2. Add write permission for others file: cap1 ……
3. Delete permission write, execute for group file: cap1 ……
4. Add read permission for users, groups and others file: cap1
   ……

Solution

chmod 546 cap1