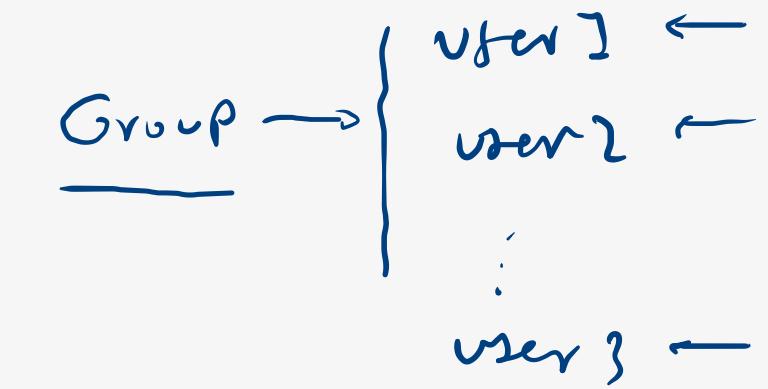




# Linux File Permissions

**File permissions** in Linux are a **set of rules that define who can read, write, or execute a file**. They are essential for maintaining system security and ensuring that users can only access files they are authorized to interact with. The file permissions are usually represented in a triplet format and are set for three different types of users: the **owner (user)**, **the group**, and **others**.

## Components of File Permissions



- Owner (User) Permissions: The permissions granted to the user who owns the file.
- Group Permissions: The permissions granted to the group that owns the file.
- Other Permissions: The permissions granted to all other users.

# Types of Permissions

There are three types of permissions in Linux:

- ✗ • **Read (r)**: Permission to read the contents of the file.
- ✗ • **Write (w)**: Permission to modify the contents of the file.
- ✗ • **Execute (x)**: Permission to execute the file, if it is a script or a program.

لیز

تغیر محتوا

لئارن

# Representation of Permissions

$755 \Rightarrow rwxr-xr-$

Permissions are typically represented in two ways:

## Symbolic (rwx) Notation:

- r for read
- w for write
- x for execute
- - indicates the absence of a permission

## Octal (Numeric) Notation:

Permissions are represented by a three-digit octal number, where each digit ranges from 0 to 7. Each digit is a sum of binary values: read (4), write (2), and execute (1)

directory  
rw -

## Example:

- rwx (read, write, execute) for the owner
- r-x (read, execute) for the group
- r-- (read) for others

## Example:

755

- 7 ( $4+2+1$ ) for the owner (read, write, execute)
- 5 ( $4+0+1$ ) for the group (read, execute)
- 5 ( $4+0+1$ ) for others (read, execute)

- **Viewing Permissions:**

You can view the permissions of a file using the `ls -l` command.

```
ls -l filename
```

This command will output a detailed list including the file's permissions.

- **Changing Permissions:**

You can change file permissions using the `chmod` command.

### Symbolic mode:

```
chmod u+rwx,g+rx,o+r filename
```

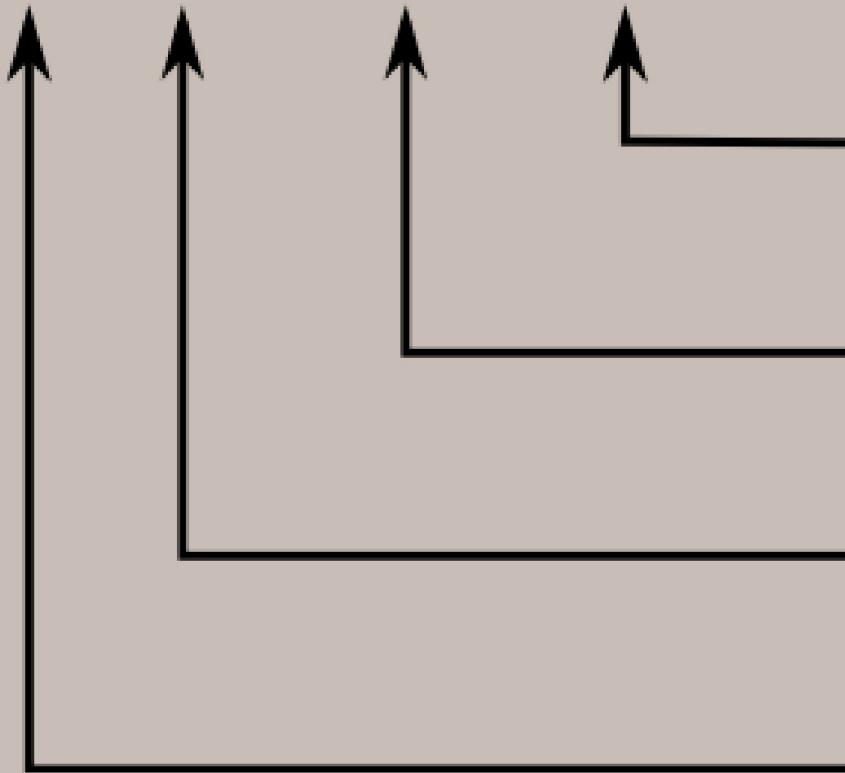
A hand-drawn diagram below the command shows three blue arrows pointing from the letters 'u', 'g', and 'o' in the permissions string to the words 'user', 'group', and 'other' respectively. The 'rwx' part of the string is also circled in blue.

### Octal mode:

```
chmod 755 filename
```

A hand-drawn diagram below the command shows a blue bracket under the number '755', with the word '755' written above it in blue ink.

- rwx rwx rwx



Read, write, and execute  
permissions for all other users.

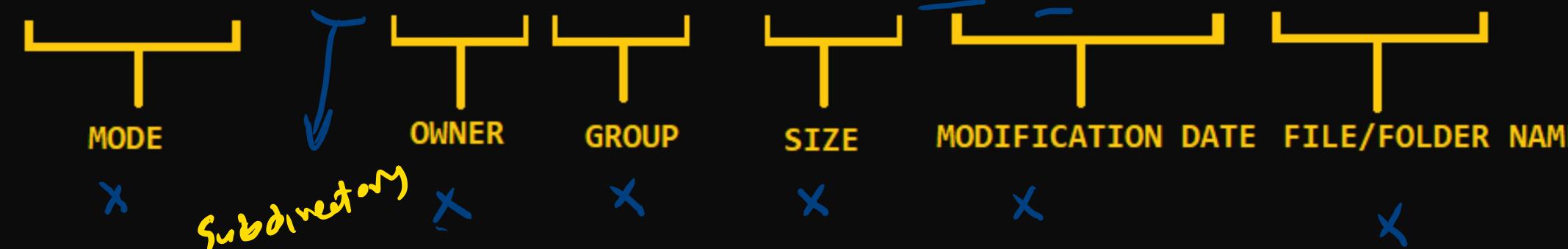
Read, write, and execute  
permissions for the group owner  
of the file.

Read, write, and execute  
permissions for the file owner.

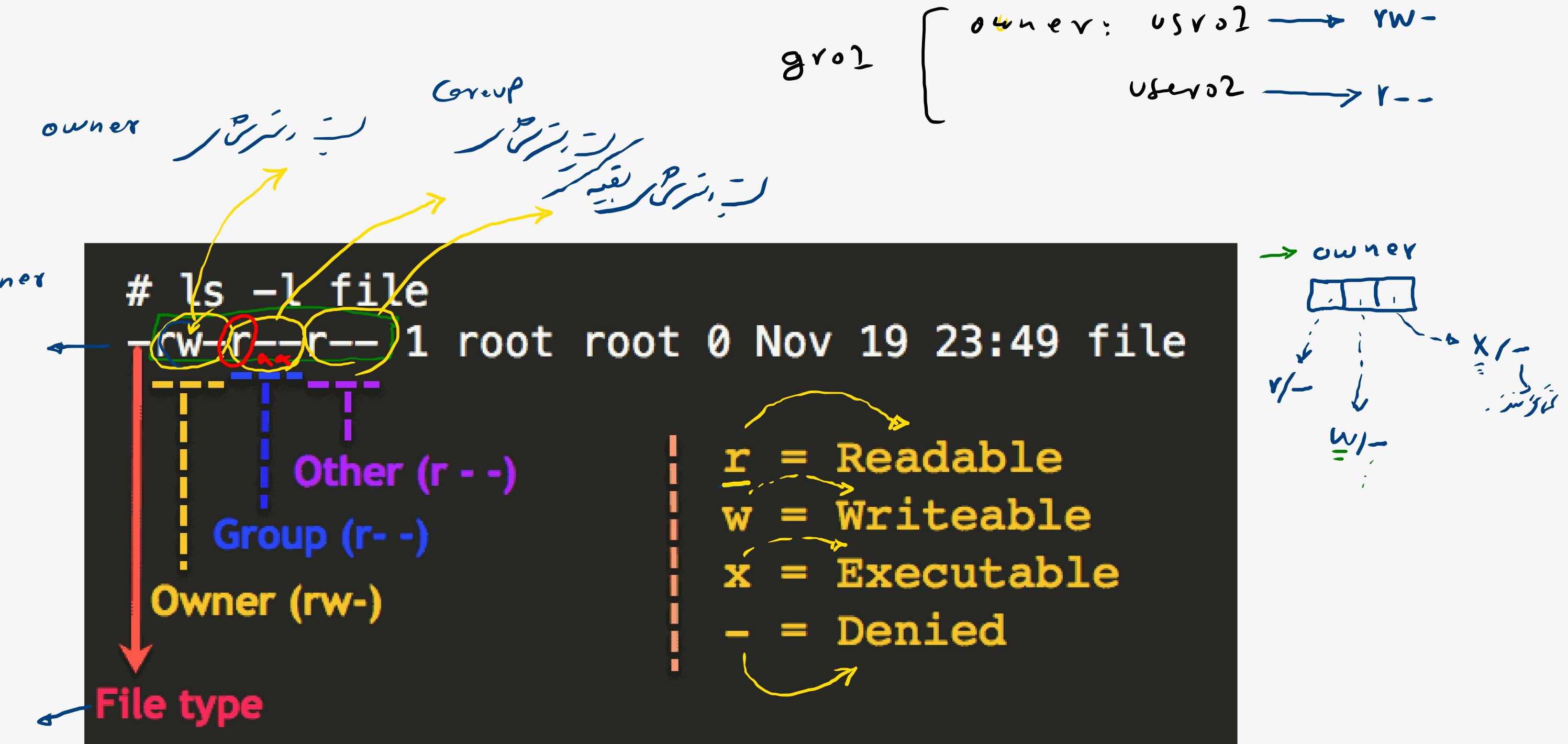
File type:  
- indicates regular file  
d indicates directory

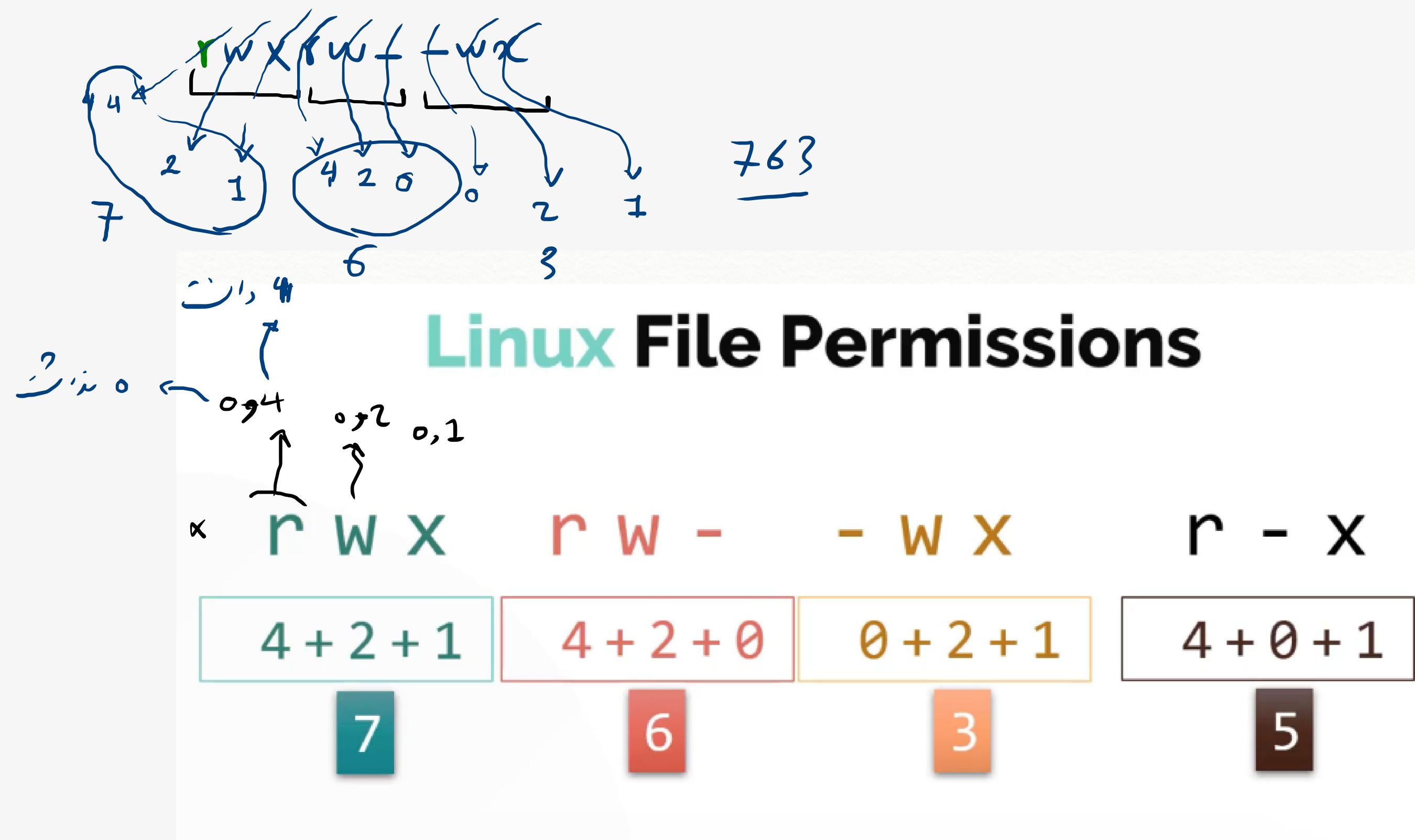


```
zaira@Zaira:~/freeCodeCamp$ ls -l
total 3856
-rw-r--r--    1 zaira zaira      89 Apr  5 20:46 CODE_OF_CONDUCT.md
-rw-r--r--    1 zaira zaira     210 Apr  5 20:46 CONTRIBUTING.md
-rw-r--r--    1 zaira zaira   1513 Apr  5 20:46 LICENSE.md
-rw-r--r--    1 zaira zaira  19933 Apr  5 20:46 README.md
drwxr-xr-x    4 zaira zaira    4096 Apr  6 22:45 api-server
-rw-r--r--    1 zaira zaira      67 Apr  5 20:46 babel.config.js
drwxr-xr-x   10 zaira zaira    4096 Apr  6 22:55 client
drwxr-xr-x    5 zaira zaira    4096 Apr  6 22:54 config
```



کاربران را کو آنده (rw)  
کاربران فرآورانه (r--)  
کاربران را لواسترا





# Compression

In Linux, compression refers to the **process of reducing the size of files or directories to save space or to facilitate easier and faster transfer over a network**. Compression works by eliminating redundancy in data, encoding it more efficiently, or both.

# Common Compression Tools

**gzip**

**bzip2**

**xz**

**zip**

**tar**

## gzip

- Commonly used for compressing individual files.
- Uses the DEFLATE compression algorithm.
- Compressed files have a .gz extension.

```
gzip file.txt # Compress file.txt to file.txt.gz
```

```
gunzip file.txt.gz # Decompress file.txt.gz to file.txt
```

## bzip2

- Similar to gzip but often achieves better compression ratios.
- Uses the Burrows-Wheeler algorithm.
- Compressed files have a .bz2 extension.

**bzip2 file.txt** # Compress file.txt to file.txt.bz2

**bunzip2 file.txt.bz2** # Decompress file.txt.bz2 to file.txt

## **xz**

- Offers high compression ratios and supports multi-threading.
- Uses the LZMA2 algorithm.
- Compressed files have an .xz extension.

**xz file.txt #** Compress file.txt to file.txt.xz

**unxz file.txt.xz #** Decompress file.txt.xz to file.txt

# zip

- Widely used for creating compressed archives that can contain multiple files and directories.
- Compressed files have a .zip extension.

**zip archive.zip file1.txt file2.txt # Compress file1.txt and file2.txt into archive.zip**

**unzip archive.zip # Extract files from archive.zip**

.

# **tar**

- Often used in conjunction with gzip, bzip2, or xz to create compressed archive files.
- A .tar file is an uncompressed archive, often called a tarball.

**tar -cvf archive.tar file1.txt file2.txt # Create an uncompressed archive**

**tar -xvf archive.tar # Extract an uncompressed archive**

**# With compression**

**tar -czvf archive.tar.gz file1.txt file2.txt # Create a gzip-compressed archive**

**tar -xzvf archive.tar.gz # Extract a gzip-compressed archive**

**tar -cJvf archive.tar.xz file1.txt file2.txt # Create an xz-compressed archive**

**tar -xJvf archive.tar.xz # Extract an xz-compressed archive**

**tar -cJvf archive.tar.xz file1.txt file2.txt # Create an xz-compressed archive**

**tar -xJvf archive.tar.xz # Extract an xz-compressed archive**

# Compression Algorithms

- **DEFLATE:** Used by gzip, combines LZ77 and Huffman coding.
- **Burrows-Wheeler Transform (BWT):** Used by bzip2, transforms data for better compression.
- ✓ • **LZMA2:** Used by xz, a high-ratio algorithm suitable for large files.
- **Lempel-Ziv (LZ77 and LZ78):** Basis for many modern compression methods.

# Usage Scenarios

- **Space Saving:** Reducing file size to save disk space.
- **Backup and Archiving:** Creating compressed archives for efficient storage and retrieval.
- **Data Transfer:** Compressing files for faster and more reliable network transfer.
- **Packaging:** Creating compressed packages for software distribution.