LINUX FUNDAMENTALS

OVERVIEW

History of Linux?

What is kernel

Linux System architecture

Linux file systems

Linux Folders

Linux common commands

File Permissions

History of Linux

- Before talking of Linux, we first have to go back in time to learn about another name, which is Unix.

1969 -Ken Thompson and Dennis Ritchie

- Unix is an operating system that has been around for a long time, at AT&T Bell Labs. The project was led by Ken Thompson and Dennis Ritchie, two famous computer scientists.

- **Motivation**-That time were very few operating systems those that were available often highly specific to particular hardware architectures.

- Unix is multi-tasking, multi-user operating system but is not free to use and is not open source.

1983-Richard Stallman ,GNU (GNU’s Not Unix) project

- Main Goal-create a free, Unix-like operating system, where people have the freedom to copy, develop, modify and distribute software

Linus and Linux

- Linus Torvalds, a student studying computer science at the university of Helsinki, he wanted to make a free and open operating system that anyone could use and improve with the help of other developers.

1991

- Finally Linus Torvalds introduced a personal product, which later became the Linux Kernel.

- The combination of the Linux kernel and the GNU(**GNU's Not Unix**) software created the first completely free operating system. It is named **GNU/Linux**.

Important things of Linux

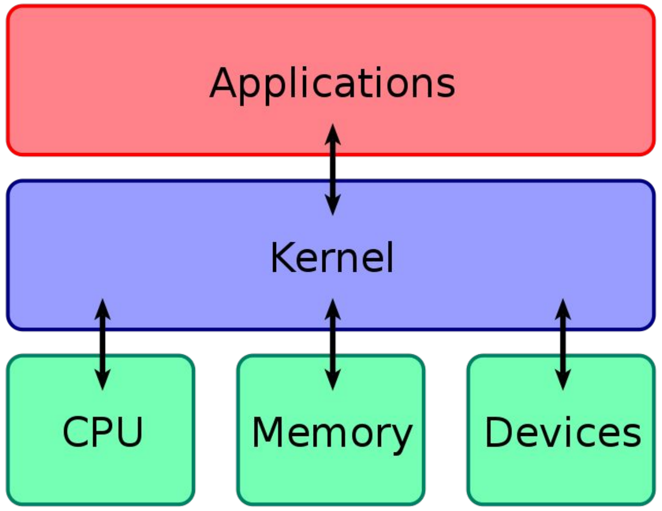
- Linux itself is just a kernel, it is not a complete operating system.
-
- The operating system that we still using on our computer is called GNU / Linux,
- Linux operating system does not use or share any part of Unix,It was built entirely new by Linus and the GNU Project .

What is Kernel

A kernel is the core component of an operating system. It is also a system program. It is the part of Operating System which translates the application commands in to hardware command

It provides an interface between application and hardware.

The main purpose of a kernel is to manage memory, disk and task.



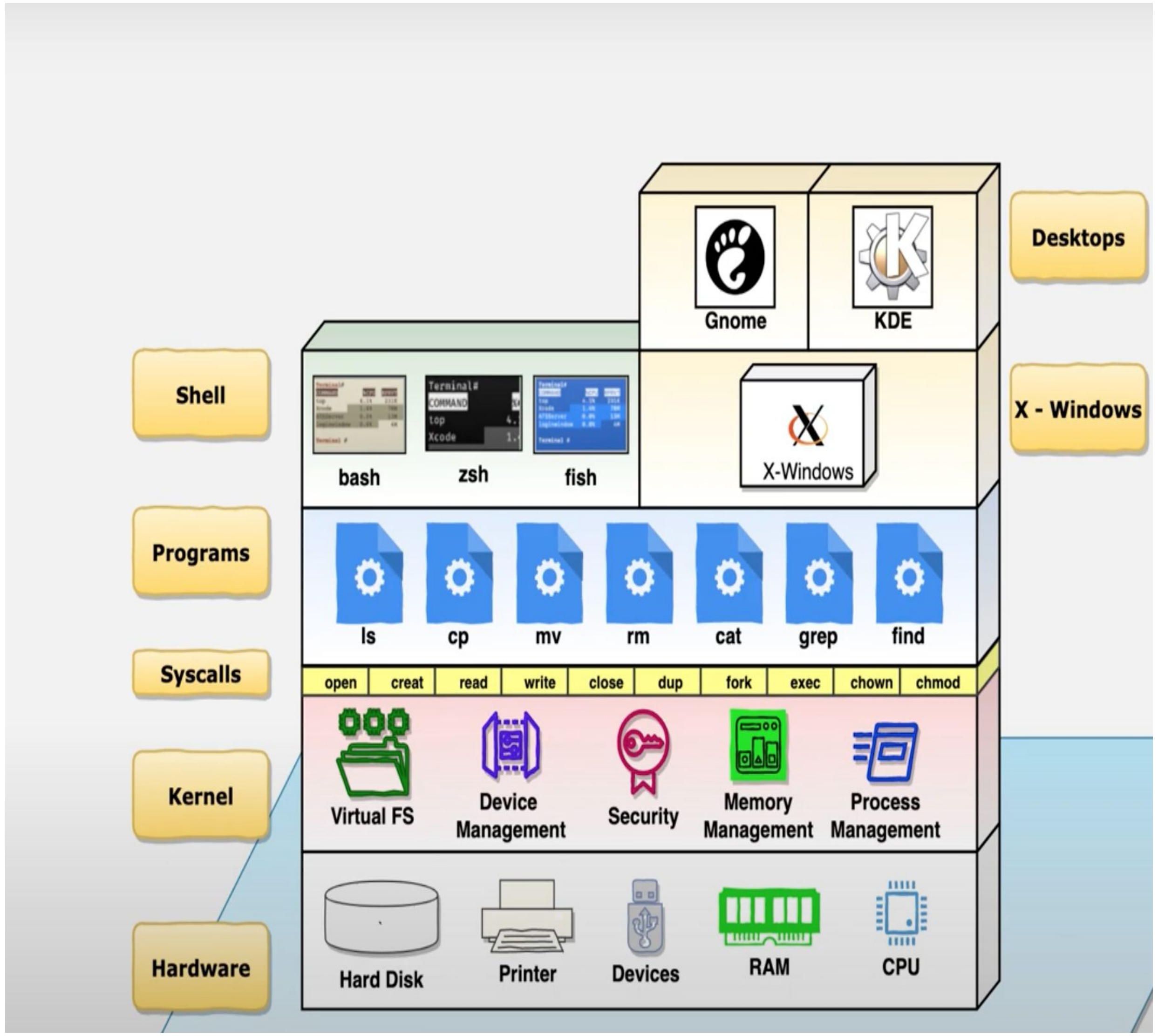
Linux System Architecture

Divided in to two levels

- User space
- Kernel space

user space is the area of memory where applications and user-level programs run. It contains the code and data that are executed by user-level processes, and it provides access to system resources through **system calls.**

Kernel space It is responsible for managing hardware resources and providing services to user-level programs through **system calls**.



system calls-It **is a request for the kernel to access a resource**
open() - opens a file and returns a file descriptor
read() - reads data from a file descriptor
write() - writes data
close() - closes a file descriptor

System calls provide an interface between the user-level application and the kernel

Programs-This includes the user application and utilities(commands that are used to perform various tasks on the system)

Shell-This is a program that provides a command-line interface for users to interact with the operating system

Gnome-It is a desktop environment in Linux. It is a graphical user interface (GUI) that provides an user-friendly interface for users to interact with the operating system

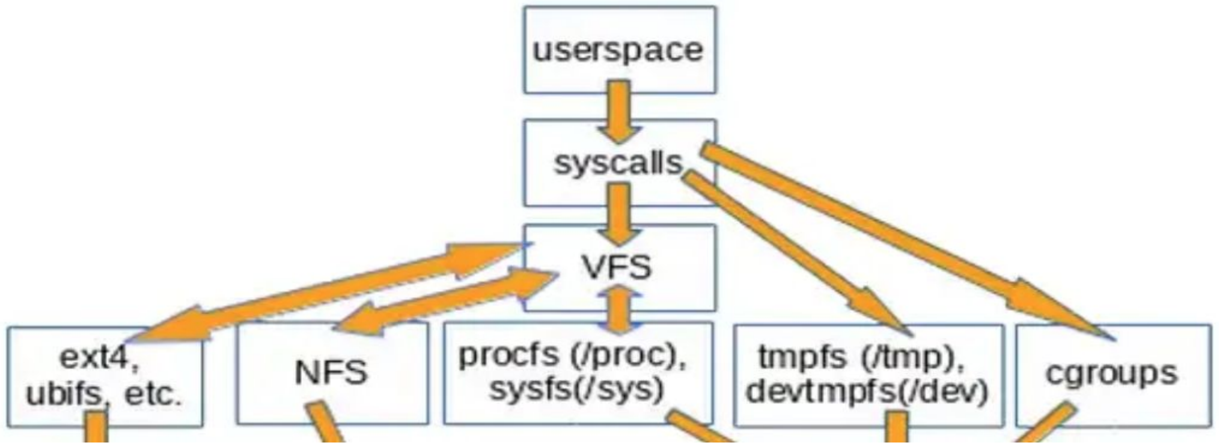
When a user types a command like **cat example.txt** in the terminal, the shell decodes the command and uses **system calls** to execute it, from the system calls this request will be sent to the **kernel**. In this case, the shell would use the **open() ,read() ,write() and close()** system calls to perform this task(cat) **example.txt**

Linux File System

In Linux, a file system is a way of organizing and storing files and folders on a storage device such as hard disk drive (HDD) .

Linux abstracts file systems operations through the Virtual File System (VFS),This provides a standard way of interacting with different file systems

It provides a uniform view of the file system to user applications.

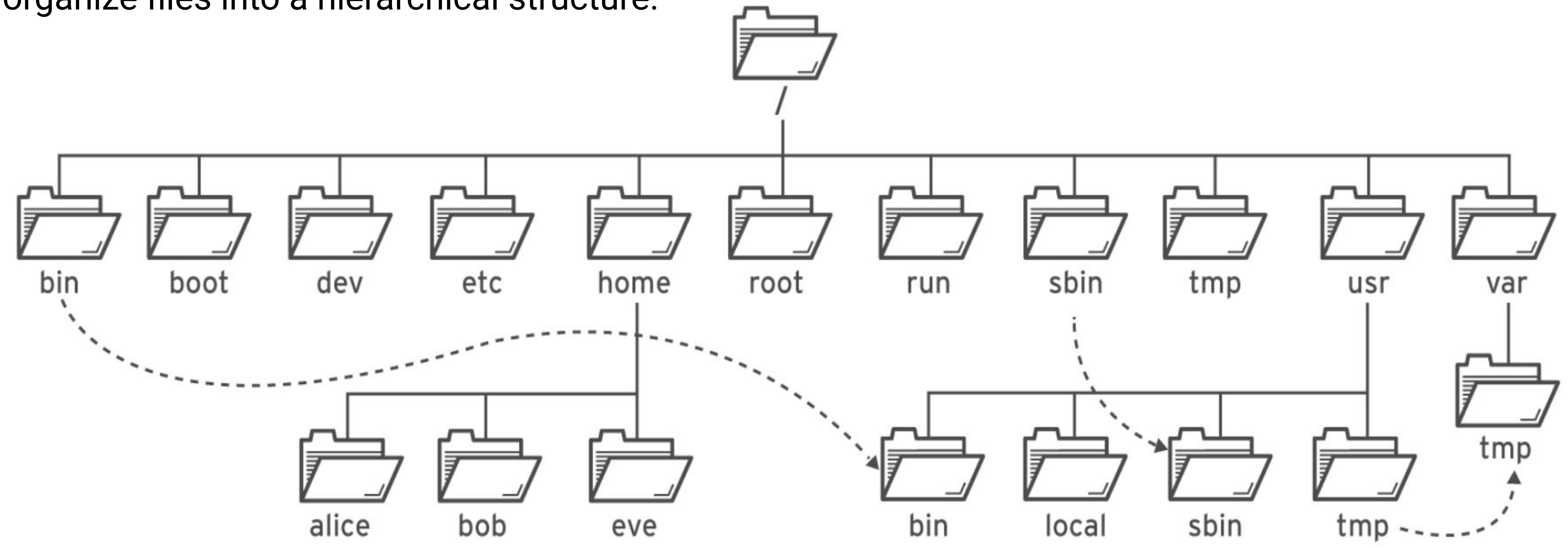


So a user program which uses the **write() syscall** doesn’t execute a **syscall** in the kernel immediately,the VFS handles the request to the appropriate file system based on the file descriptor, Each file system has its own features and limitations, and the choice of file system depends on factors such as the type of storage device

- Ext
- Ext2
- Ext3
- Ext4
- JFS
- XFS
- btrfs
- swap

Folder/Directory Structure

In Linux, a folder is a special type of file that is used to organize files into a hierarchical structure.



•**(/)** folder: also known as the root directory, is the top-level directory in the file system hierarchy. All other directories and files in the file system are located within the root directory or one of its subdirectories

/root : The administrative user's home folder. (cd ~)

/bin : binary files and executable programs that are required for basic system functionality, such as the **shell (ls,cp,mkdir,rm and etc)** commands.

• **/boot** : contains files required for booting the system, including the kernel, initial ramdisk, and boot loader configuration files..

• **/dev** : contains device files that represent physical and virtual devices connected to the system, such as disks, terminals, and printers..

• **/etc** : Most important system configuration files are in /etc, this directory contains data similar to those in the Control Panel in Windows.

• **/home** : Home directories of the common users.

/lib: Library files, includes files for all kinds of programs needed by the system and the users.

- /usr : contains user-related files and directories, such as application binaries, libraries, and documentation.

- /tmp: contains temporary files that are created by applications and the system, and are typically deleted when system reboot.

Common Commands

- **Useradd**- add a new user
- **Passwd**-change the current password or add a password for new user
- **ls** : List the contents of a directory
- **pwd** : Present working directory
- **cd** : Change directory
- **mkdir** : Make a directory
- **cp** : Copy
- **cp -r** :Copy a directory and its contents

Common Commands

mv : Move

rm : Remove

rm -r directory: Remove a directory containing files

rmdir directory: Remove an empty directory

Shell Shortcuts for bash

Ctrl-A (jump to start of line)
Ctrl-E (jump to end of line)
Ctrl-K (delete (kill) everything from the cursor onwards
Ctrl-W (delete the previous word only)
Ctrl-Y (paste whatever was just deleted)
Ctrl-C (kill/exit a running process)
Ctrl-L (clear the screen)
Ctrl-R (search for previously executed commands)
Tab (auto-complete command or file/directory name)
↑ / ↓ (scroll back / forwards through previously entered commands)

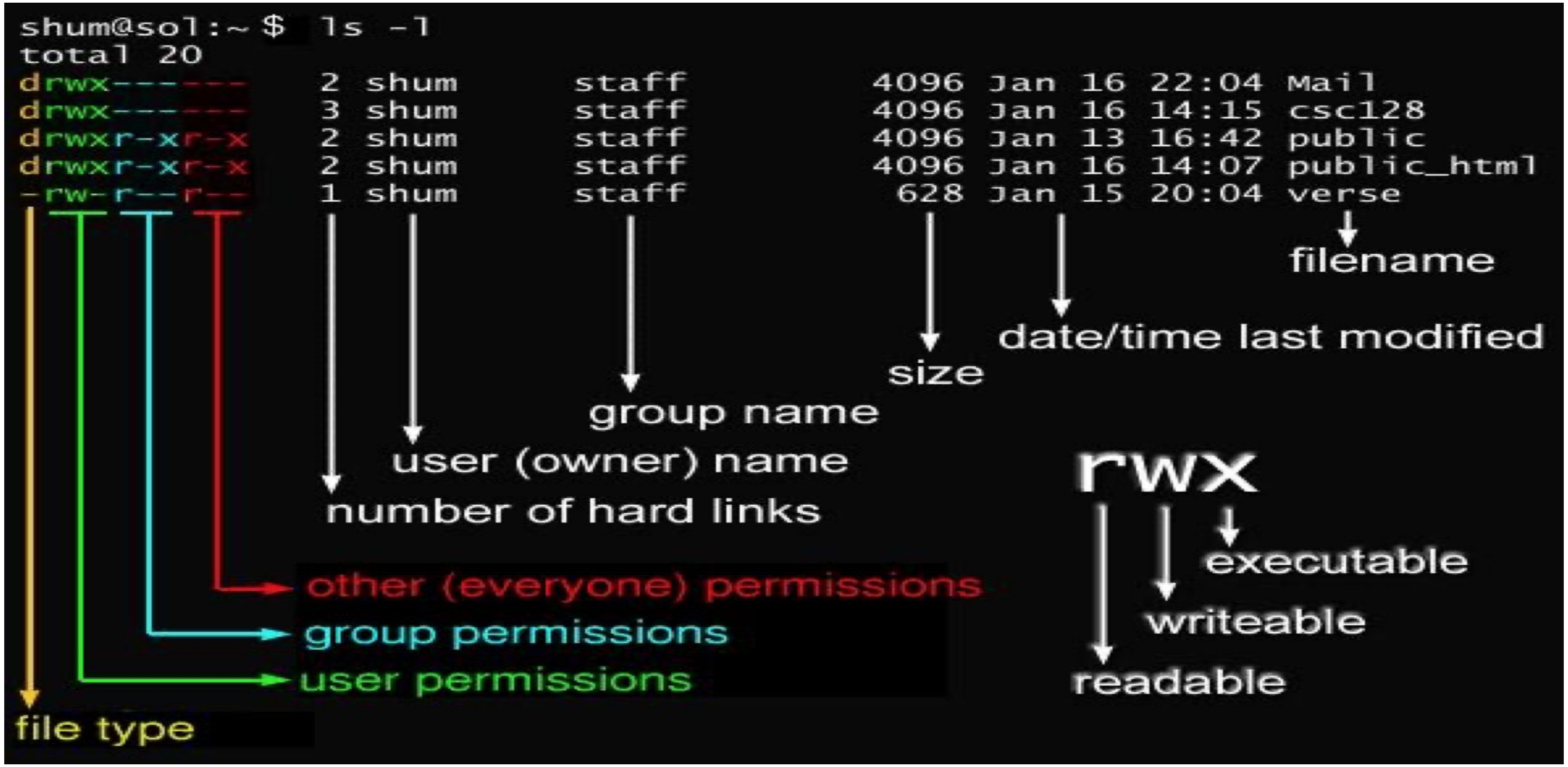
Types of Users

Root User-The root user is also known as the superuser and has complete control over the system

System users: System users are created by the system for running specific services or processes. These users do not have login privileges, and their accounts are locked by default.

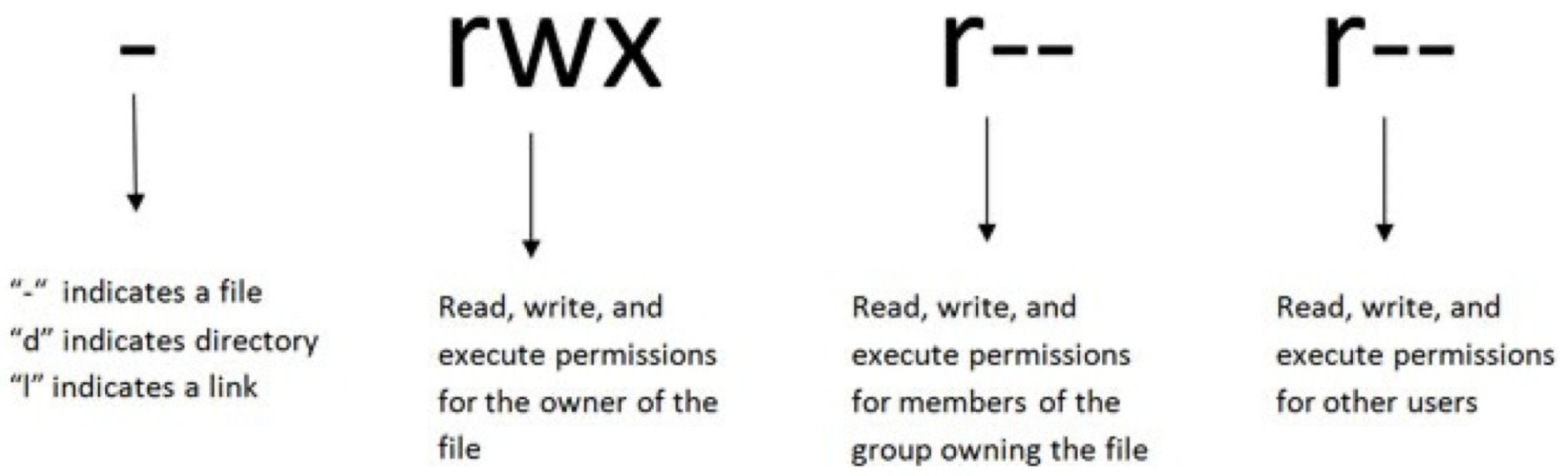
Regular users: Regular users are created by the system administrator or by other regular users. These users have limited privileges and cannot perform tasks that require root access, such as modifying system files or installing software

File Permission



```
shum@sol:~ $  ls -l
total 20
drwx------    2 shum      staff        4096 Jan 16 22:04 Mail
drwx------    3 shum      staff        4096 Jan 16 14:15 csc128
drwxr-xr-x    2 shum      staff        4096 Jan 13 16:42 public
drwxr-xr-x    2 shum      staff        4096 Jan 16 14:07 public_html
-rw-r--r--    1 shum      staff         628 Jan 15 20:04 verse
```

File Permission in Detail



-

"-" indicates a file
"d" indicates directory
"l" indicates a link

rwx

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

r--

Read, write, and execute permissions for members of the group owning the file

r--

Read, write, and execute permissions for other users

Access Rights

- Files are owned by a user and a group (ownership)
- Files have permissions for the user, the group, and other
- “Other” permission is often referred to as “world”
- The permissions are Read, Write and Execute (r, w, x)
- The user who owns a file is always allowed to change its permissions

Changing File Permissions

File permissions can be change using “**chmod**” command There are two ways to use this command

- *Symbolic mode*
- *Absolute mode*

Symbolic Mode

Uses letters and “+” , ”-” to give permissions

- Letters are used as following

u	User
g	Group
o	Other
r	Read
w	Write
e	Execute
+	To add permission
-	To remove a permission

Symbolic mode (Examples)

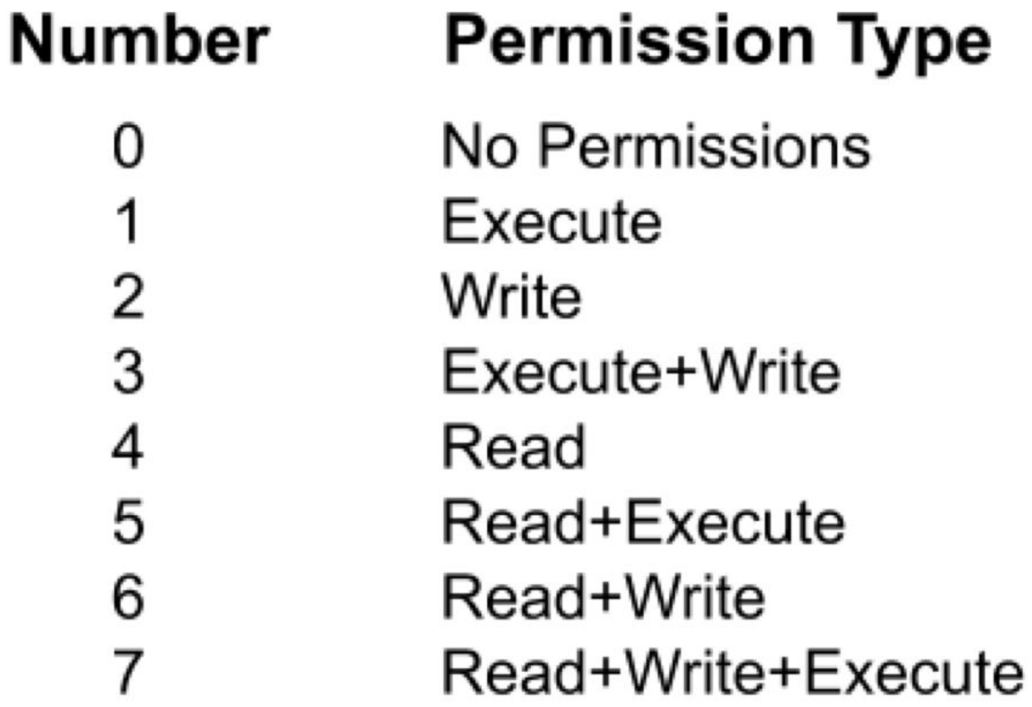
- $ chmod g+x testfile ==> -r--r-xr—
- $ chmod u+wx testfile ==> -rwxr-xr—
- $ chmod ug-x testfile ==> -rw--r--r—

u=user, g=group, o=other (world)

Absolute Mode

We use octal (base eight) values represented like this

For each column, User, Group or Other you can set values from 0 to 7



Number	Permission Type
0	No Permissions
1	Execute
2	Write
3	Execute+Write
4	Read
5	Read+Execute
6	Read+Write
7	Read+Write+Execute

Symbolic Mode (Example)

- $ chmod 445 testfile ==> -r--r-xr—
- $ chmod 754 testfile ==> -rwxr-xr—
- $ chmod 644 testfile ==> -rw--r--r—