Introduction to Linux environment

Sarai Varona

BU-ISCIII
Unidades Comunes Científico Técnicas – SGSAFI-ISCIII

29 Octubre - 3 Noviembre 2025, 1ª Edición
Programa Formación Continua, ISCIII

Overview

1. Linux OS
2. Linux file system
3. Basic commands
4. Command line syntax
5. Linux users and privileges

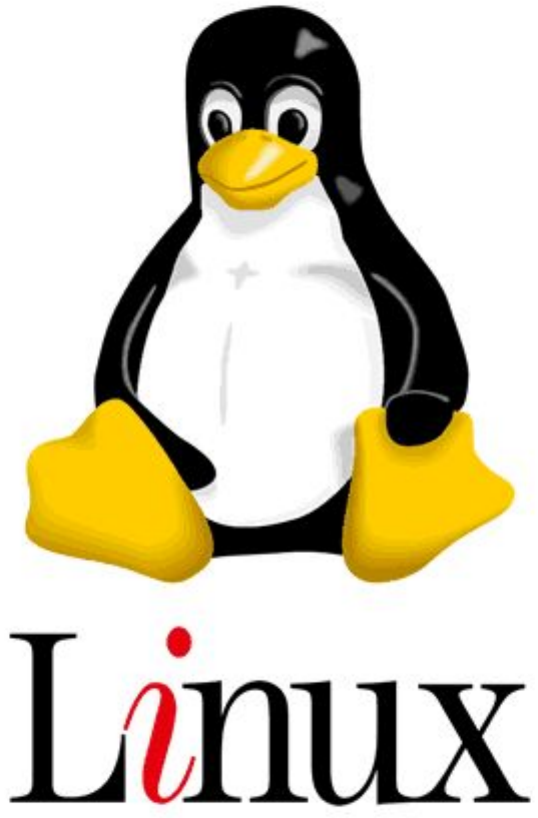
Overview

1. Linux OS
2. Linux file system
3. Basic commands
4. Command line syntax
5. Linux users and privileges

1. Linux Operating System (OS)

Features:

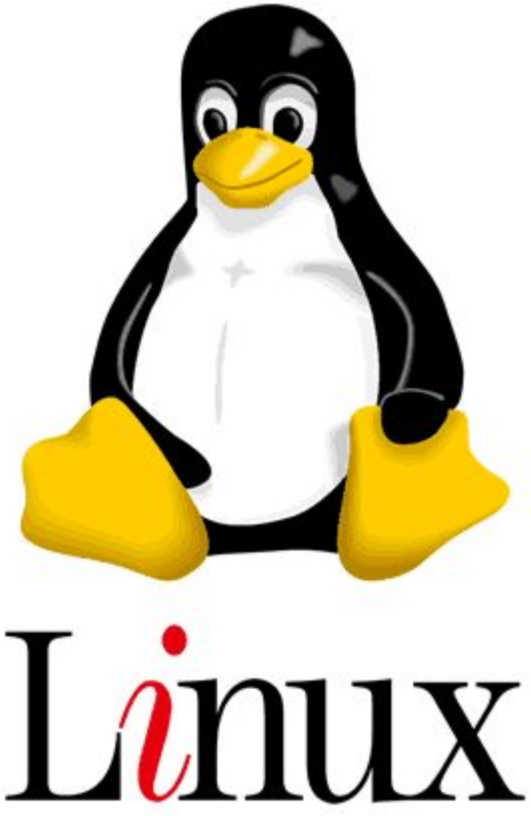
1. Operating System
2. Open source
3. Linux distributions
4. Multitasking
5. Multi-user



1. Linux Operating System (OS)

Features:

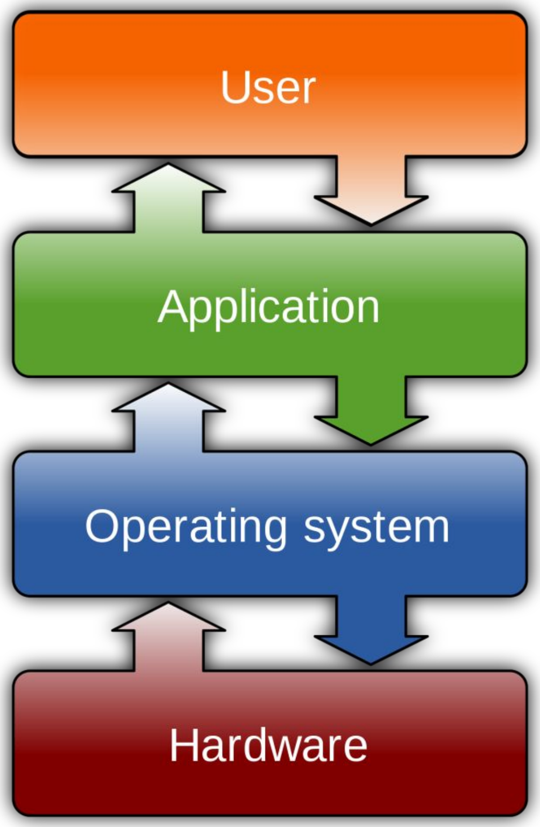
1. Operating System
2. Open source
3. Linux distributions
4. Multitasking
5. Multi-user

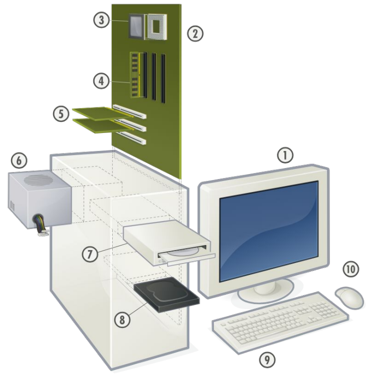


1.1. Operating System

Operating System (OS):

- **Software** that manages computer hardware and software resources and provides common services for computer programs





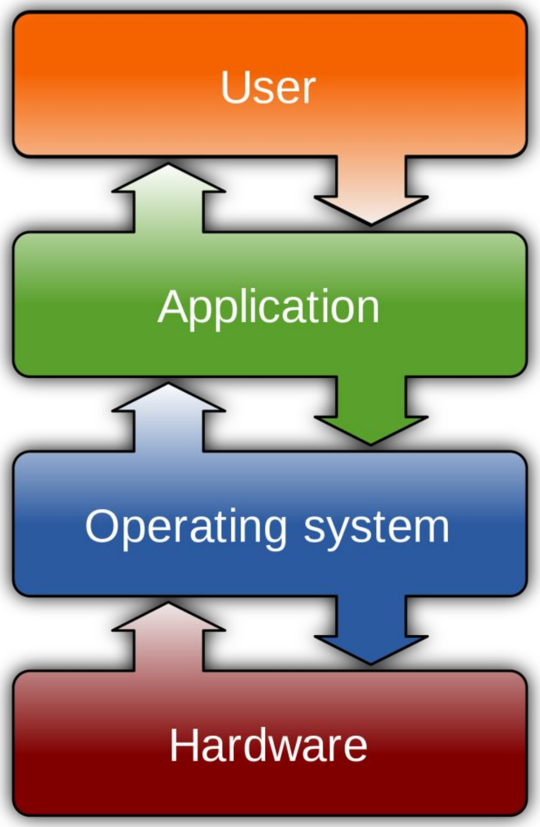
1.1. Operating System

Operating System (OS):

- Software that manages

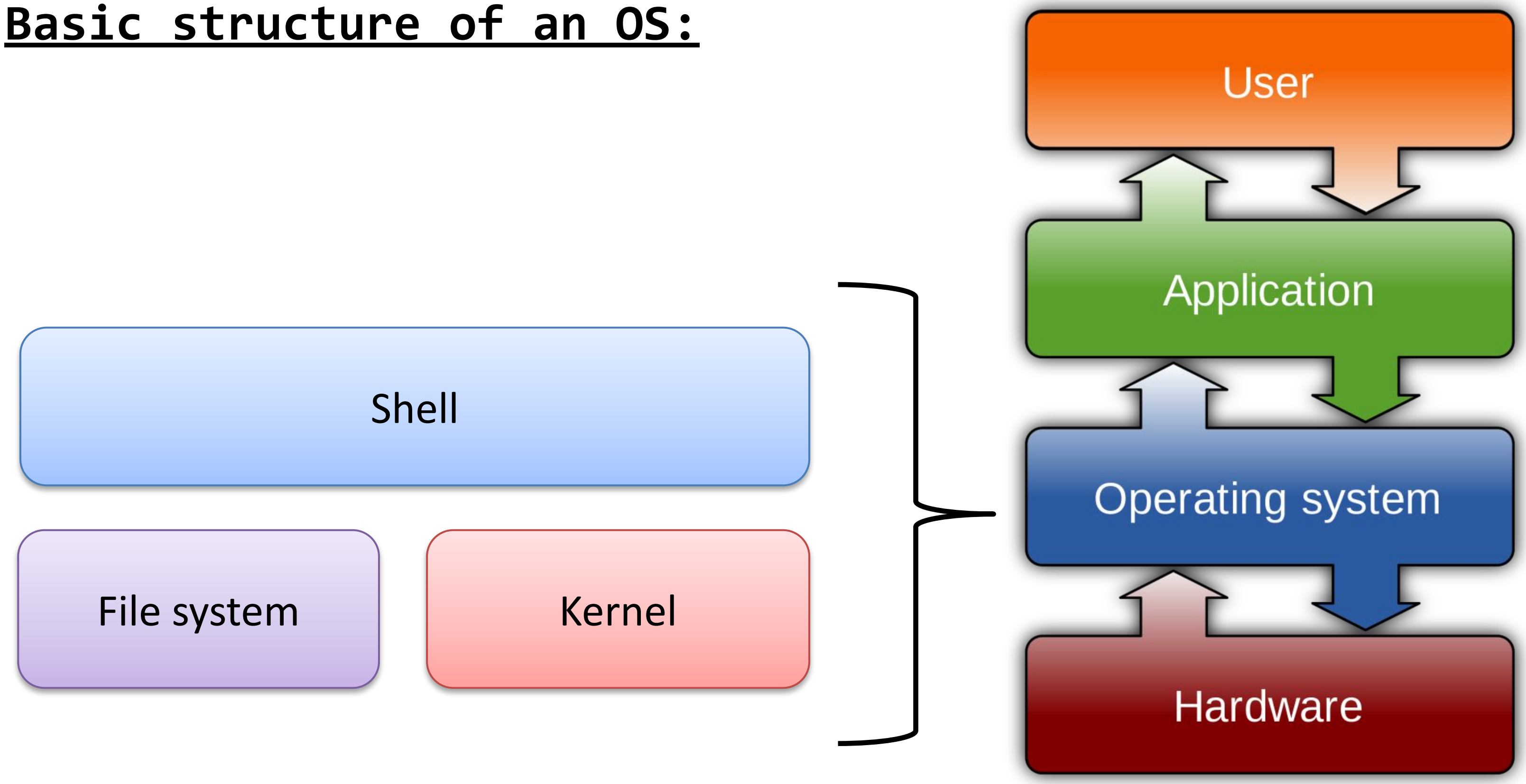
Functions:

- Program execution and control (launch, manage, and supervises running programs)
- Peripheral management (controls devices like keyboard, mouse, screen, printer, USB…)
- User and permission management (creates users, assigns roles, and controls access to files and system functions)
- Error and security management (detects system or hardware failures and protects against unauthorized access)



1.1. Operating System

Basic structure of an OS:



1.1. Operating System

Kernel

It is the foundational layer of an OS. It functions at a basic level, communicating with hardware and managing resources.

1.1. Operating System

File system

It controls how data is stored, manipulated and retrieved. (Part 2)

Kernel

It is the foundational layer of an operating system (OS). It functions at a basic level, communicating with hardware and managing resources.

1.1. Operating System

Shell

<u>Program</u> that provides the traditional, text-only user interface. (Part 3)

File system

It controls how data is stored, manipulated and retrieved. (Part 2)

Kernel

It is the foundational layer of an operating system (OS). It functions at a basic level, communicating with hardware and managing resources.

1.1. Operating System

<u>Operating System (OS):</u>

Windows

Mac OS

GNU/Linux

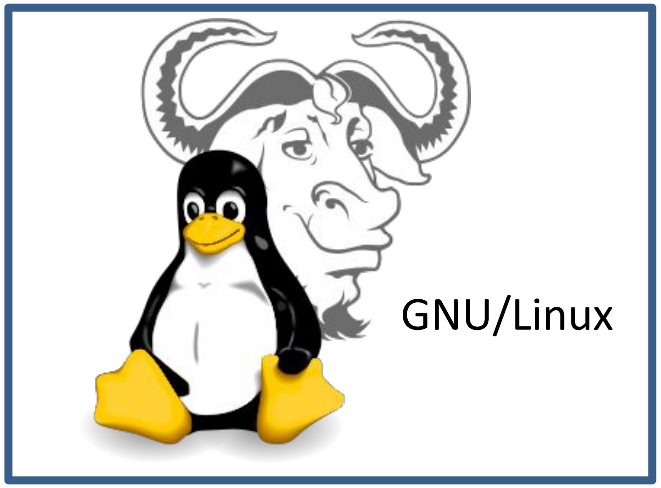
1.1. Operating System

GNU/Linux:

GNU = **G**NU's **N**ot **U**nix project
(Richard Stallman, 1983)
- Shell
- Compiler
- Libraries
- Other utils

+

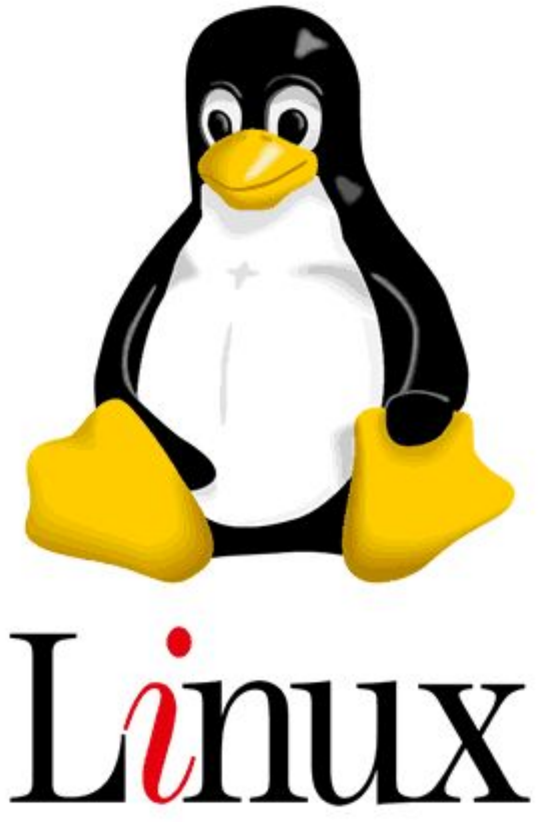
Kernel (Linus Torvalds, 1991)



1. Linux Operating System (OS)

Features:

1. Operating System
2. Open source
3. Linux distributions
4. Multitasking
5. Multi-user



2. Open source

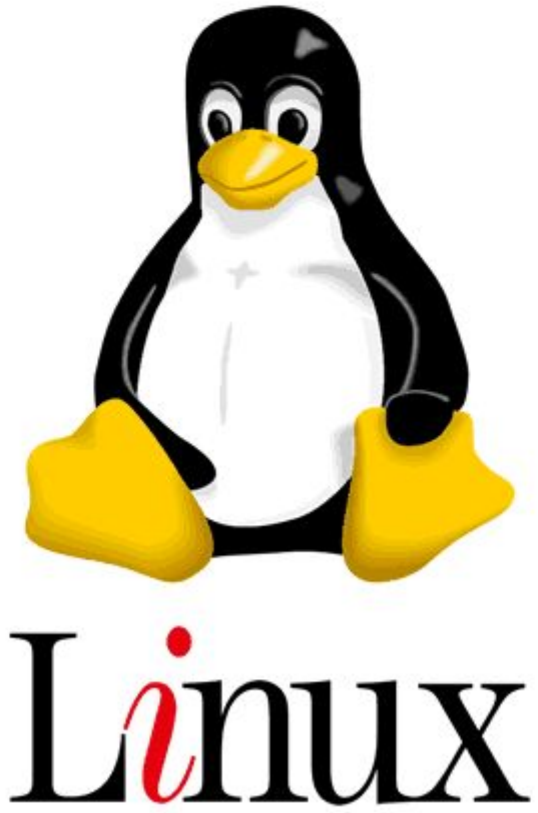
<u>The distribution terms of open-source software must comply with the following criteria:</u>

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of the Author’s Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavour
7. Distribution of license
8. License Must Not Be Specific to a Product
9. License Must not Restrict Other Software
10. License Must Be Technology-Neutral

1. Linux Operating System (OS)

Features:

1. Operating System
2. Open source
3. Linux distributions
4. Multitasking
5. Multi-user



3. Linux Distributions

<u>Linux Distributions</u>

- A distro is a Linux kernel based operating system made from a software collection and sometimes a package management system.
- There are distros for a wide variety of platforms.
- A typical Linux distro comprises a Linux kernel, GNU tools and libraries, additional software, documentation, a window system, a window manager, and a desktop environment.
- Most of the included software is free and open-source software made available both as compiled binaries and in source code form, allowing modifications to the original software. This means that you can see how the software works, and even modify it, which is especially useful in science, where reproducibility is key.

3. Linux Distributions

Linux distributions:

- Ubuntu: Beginner-friendly.
- Debian: Known for its stability.
- Fedora: Sponsored by Red Hat, cutting-edge technology.
- Arch Linux: Designed for advanced users, full control over system configuration.
- CentOS / Rocky Linux / AlmaLinux: Oriented towards server environments.

3. Linux Distributions

“Distro” - Version Kernel + programs:

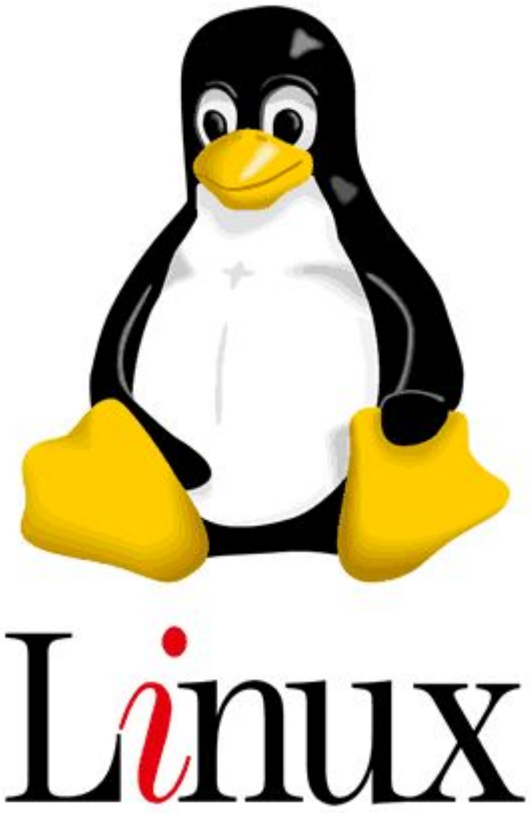
- Ubuntu: Beginner-friendly.
- Debian: Known for its stability.
- Fedora: Sponsored by Red Hat, cutting-edge technology.
- Arch Linux: Designed for advanced users, full control over system configuration.
- **CentOS** / Rocky Linux / AlmaLinux: Oriented towards server environments.

CentOS Stream release 8

1. Linux Operating System (OS)

Features:

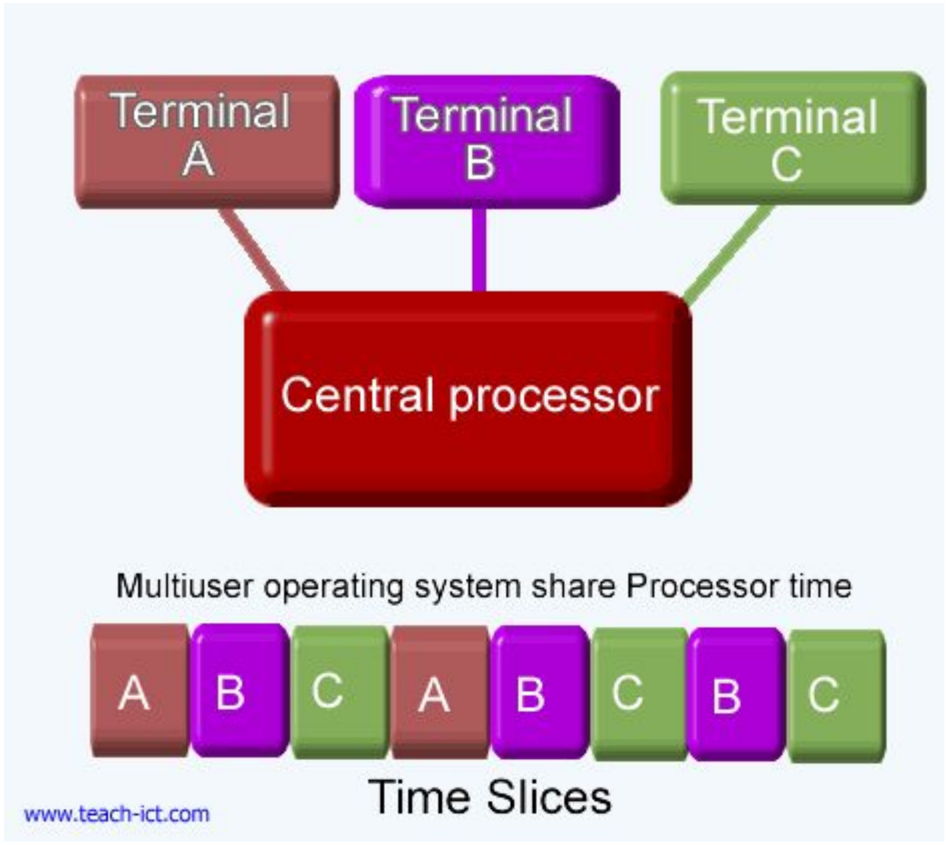
1. Operating System
2. Open source
3. Linux distributions
4. Multitasking
5. Multi-user



4. Multi-task

A multitasking operating systems allows a user to perform more than one computer task (such as the operation of an application program) at a time.

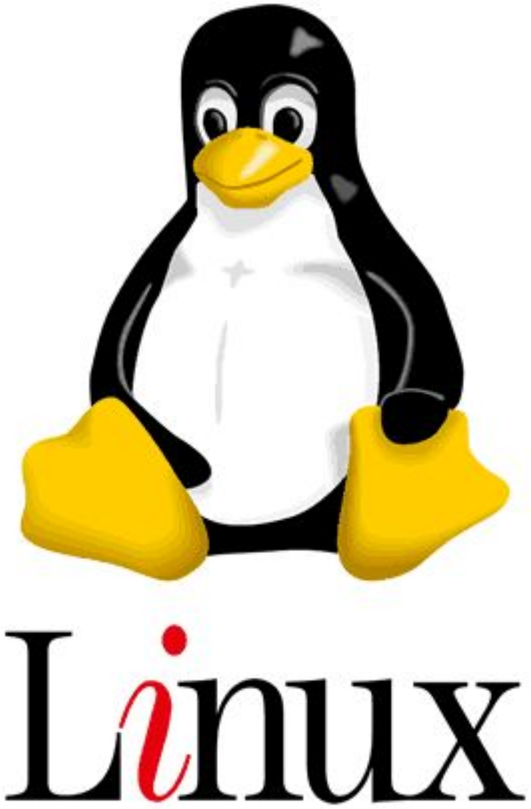
The operating system is able to keep track of where you are in these tasks and go from one to the other without losing information.



1. Linux Operating System (OS)

Features:

1. Operating System
2. Open source
3. Linux distributions
4. Multitasking
5. Multi-user



5. Multi-user

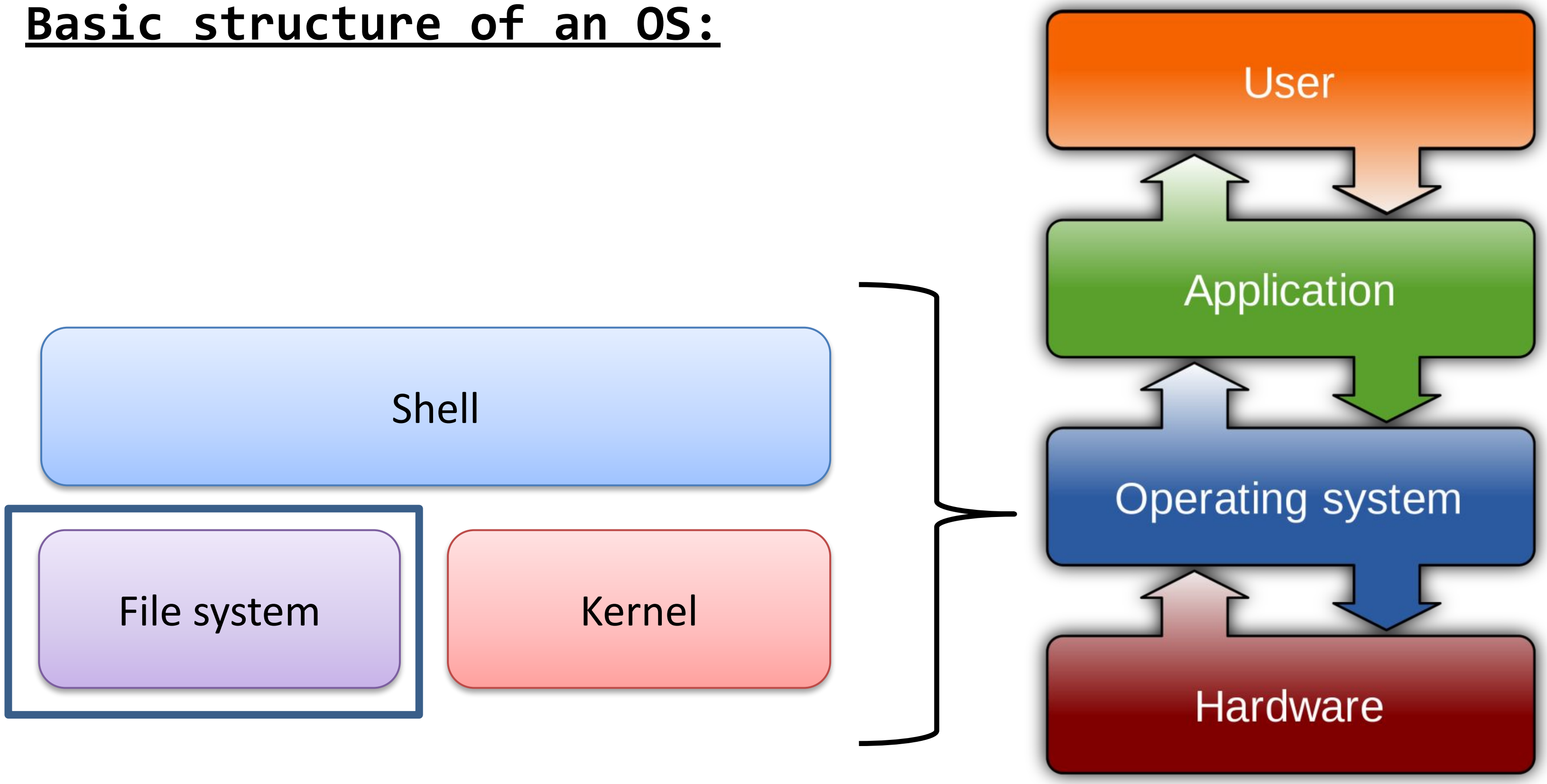
- A multi-user operating system is software that allows access by multiple users of a computer.
- An example is a server where multiple remote users have access (such as via a serial port or Secure Shell) to the Unix shell prompt at the same time.

Overview

1. Linux OS
2. Linux file system
3. Basic commands
4. Command line syntax
5. Linux users and privileges

1.1. Operating System

Basic structure of an OS:



2. Linux File System

1. Features
2. Structure
3. Comparison with Windows
4. Paths

2.1. Linux File System Features

The file system is what gives structure to the data. It **allows**:

- Storing files and folders in an organized way.
- Accessing them by name, path, and permissions.
- Controlling who can read, modify, or execute each file.

When you **save a file** the file system:

- Divides it into physical blocks stored on the disk.
- Associates metadata (name, date, permissions, size, owner).
- Places it in a folder hierarchy to make it easier to locate.

It also **manages**:

- Permissions and ownership
- Access/modification times
- Free and used disk space
- Integrity and error recovery

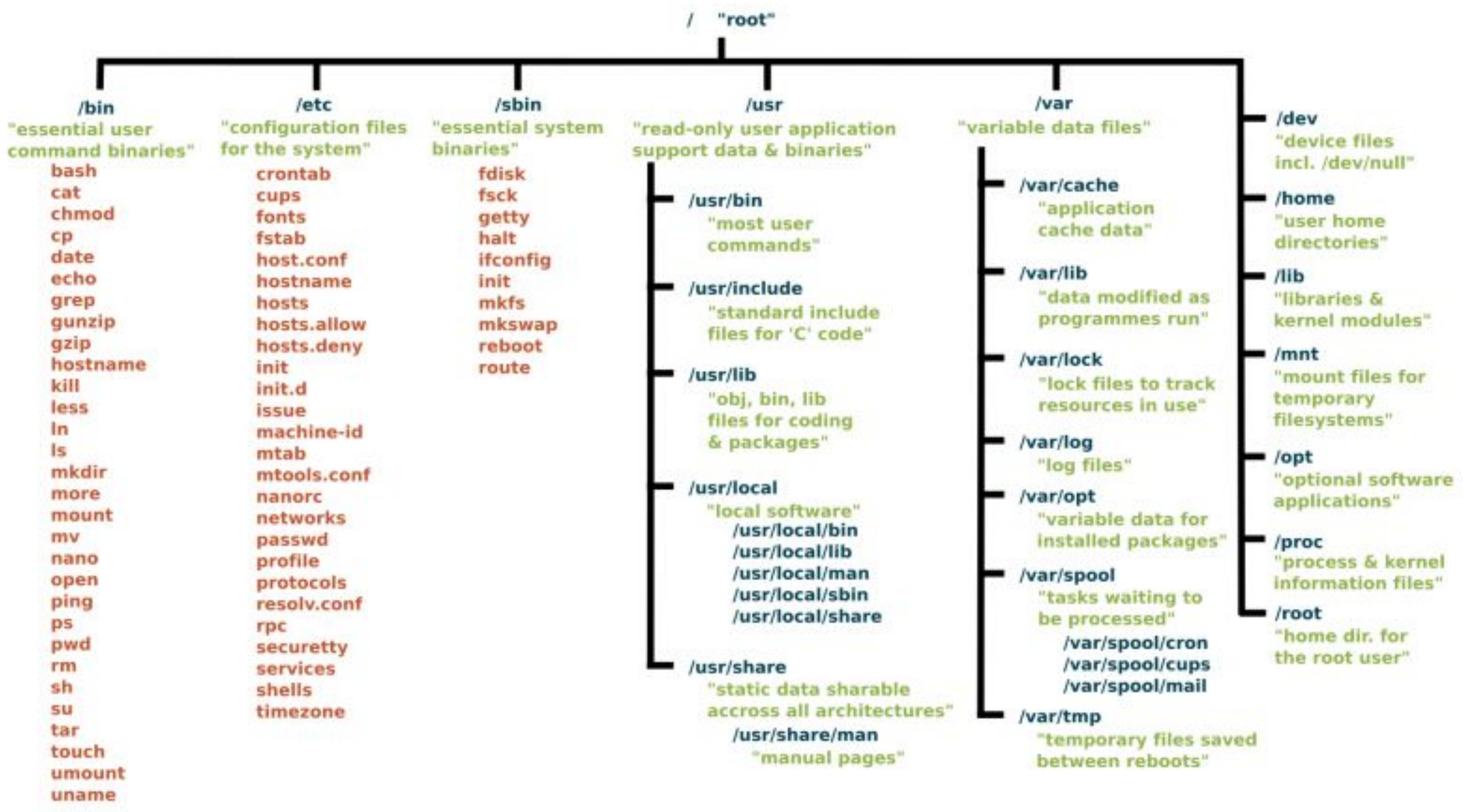
2.1. Linux File System Features

- Root directory (/)
- Everything is a File (devices, directories, and processes)
- Specifying Paths
- Case-Sensitive
- File Extensions and Hidden Files

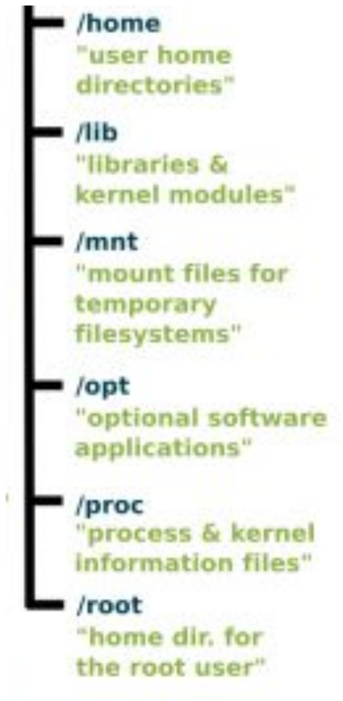
2. Linux File System

1. Features
2. Structure
3. Comparison with Windows
4. Paths

2.2. Linux File System Structure



2.2. Linux File System Structure


/home
"user home directories"
/lib
"libraries & kernel modules"
/mnt
"mount files for temporary filesystems"
/opt
"optional software applications"
/proc
"process & kernel information files"
/root
"home dir. for the root user"

User data directory

Or /media, directory where external disks are mounted.

Directory containing additional software. In our case is where bioinformatic software is installed.

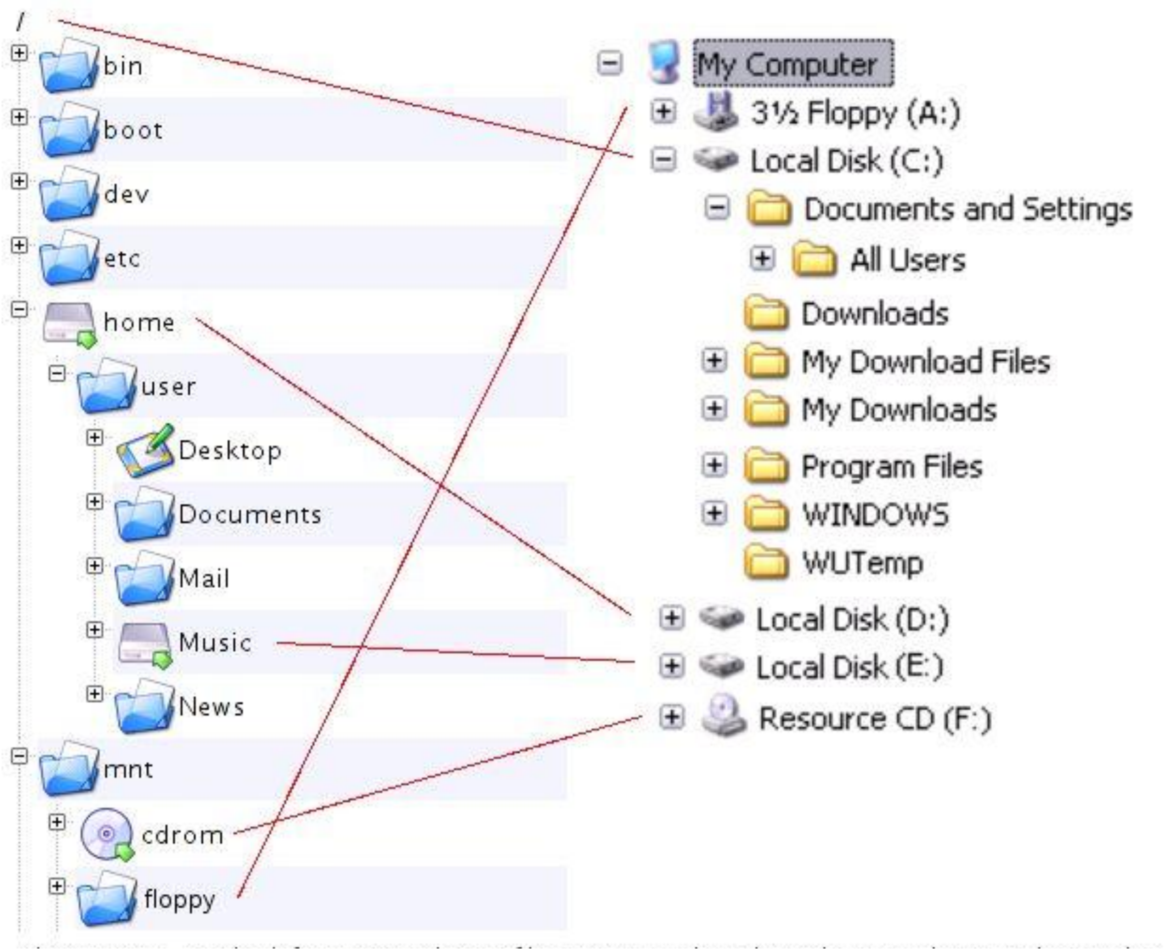
2. Linux File System

1. Features
2. Structure
3. Comparison with Windows
4. Paths

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES Instituto de Salud Carlos III >BU-ISCIII

2.3. Linux File System Comparison

- Everything "hangs" from root
- Files are classified by type / role instead of unit location
- Files locations in disks are invisible for users



2. Linux File System

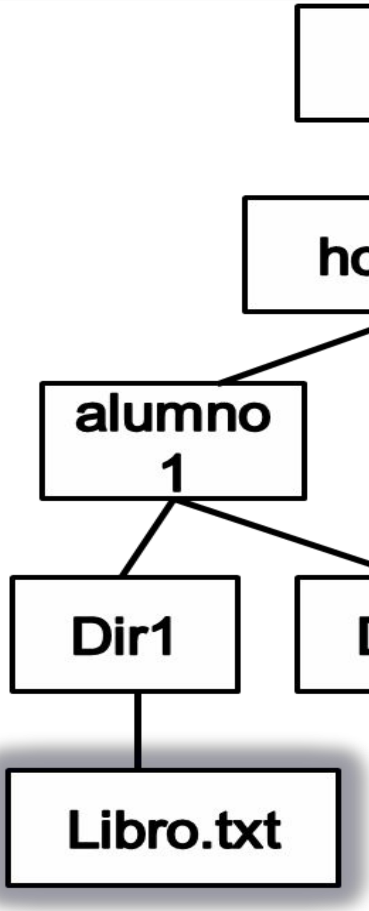
1. Features
2. Structure
3. Comparison with Windows
4. Paths

2.4. Linux File System Paths

- PATH:
  - **Absolute path:**
    - Location of a file or directory from the root directory (/).
    - Static.
    - Ej: /home/alumno1/dir1/book.txt
  - **Relative path:**
    - Path related to the present working directory (pwd).
    - Variable
    - Actual pwd = “.”
    - Parent directory = “..”
    - Ej:Path to book.txt
      - From /home: ./alumno1/dir1/book.txt
      - From /home/alumno1: dir1/book.txt
      - From /home/alumno1/dir1: book.txt

2.4. Linux File System Paths

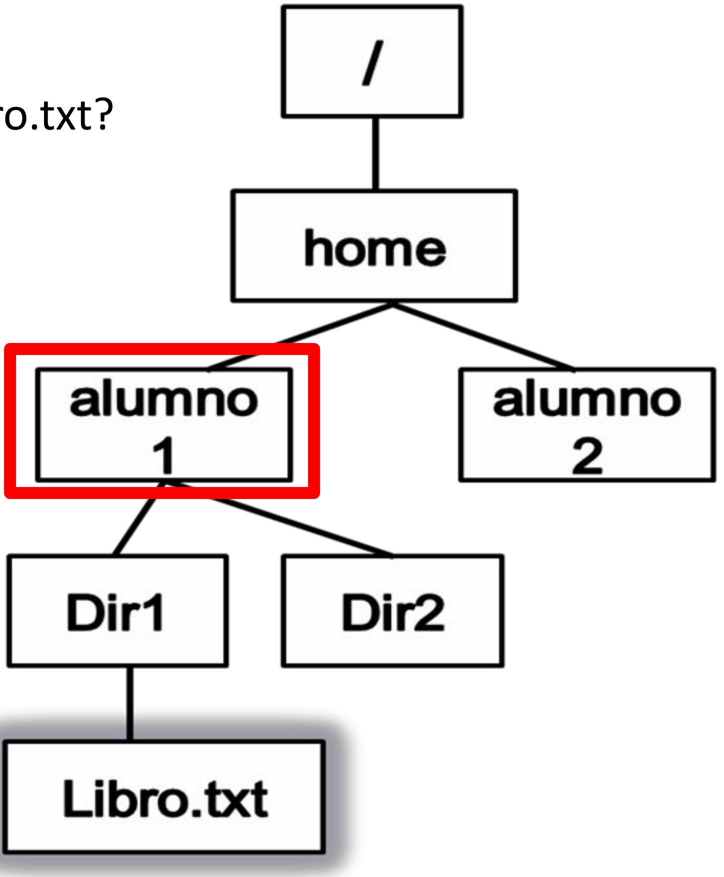
- Example
  - ../../ To go *home* from *Dir1* or *Dir2*
  - **../../alumno2** To go *alumno2* from *Dir1* o *Dir2*.
  - **../Dir2** To go from *Dir1* to *Dir2*.



2.4. Linux File System Paths

Which is the absolute path to Libro.txt?

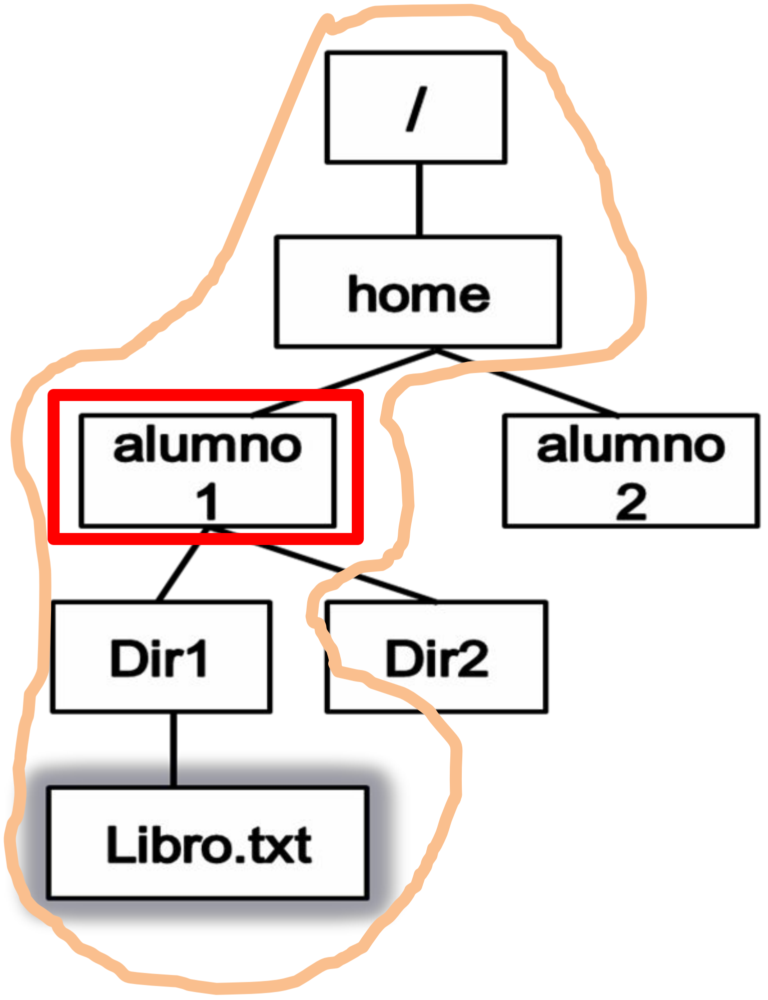
We are here ->



2.4. Linux File System Paths

No matter were we are

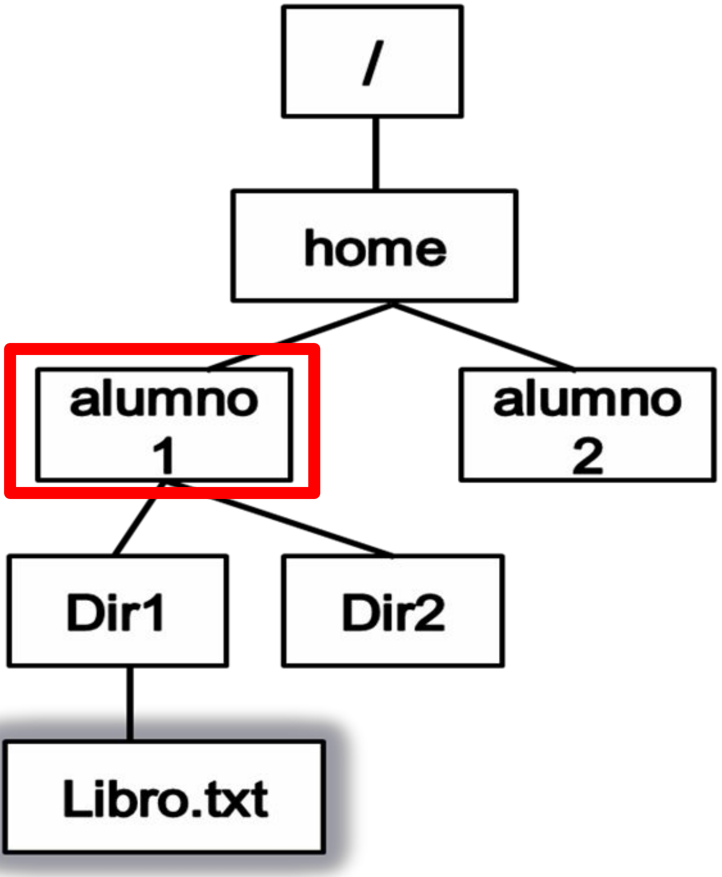
Absolute path
/home/alumno1/Dir1/Libro.txt



2.4. Linux File System Paths

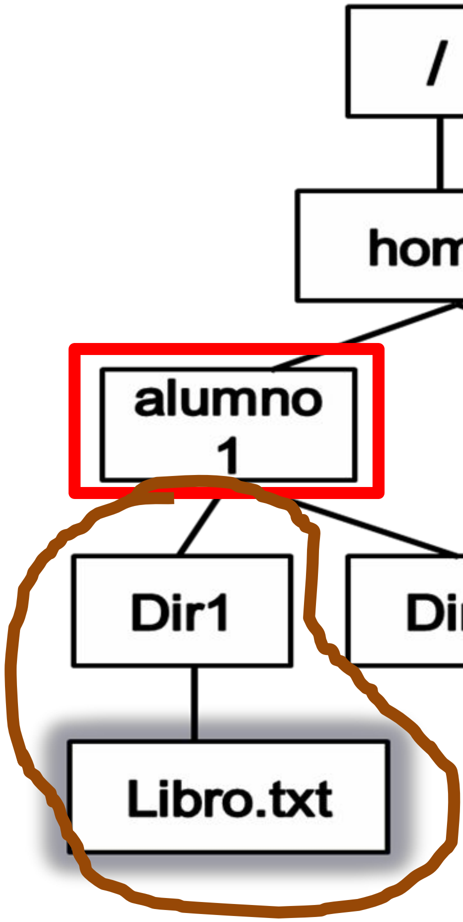
Which is the the relative path to Libro.txt? ?

We are here ->



2.4. Linux File System Paths

Important pwd or "."



Relative paths:
./Dir1/Libro.txt
or Dir1/Libro.txt

Overview

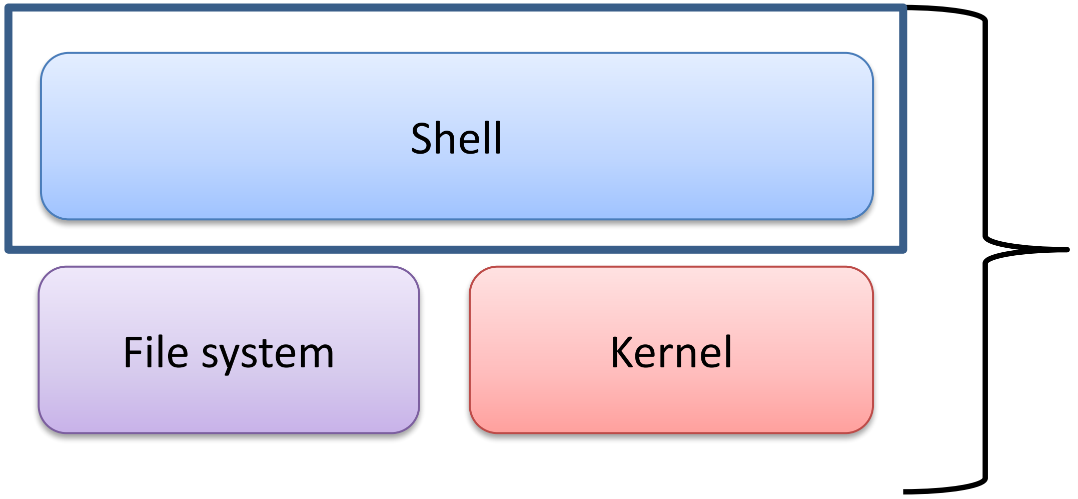
1. Linux OS
2. Linux file system
3. Basic commands
4. Command line syntax
5. Linux users and privileges

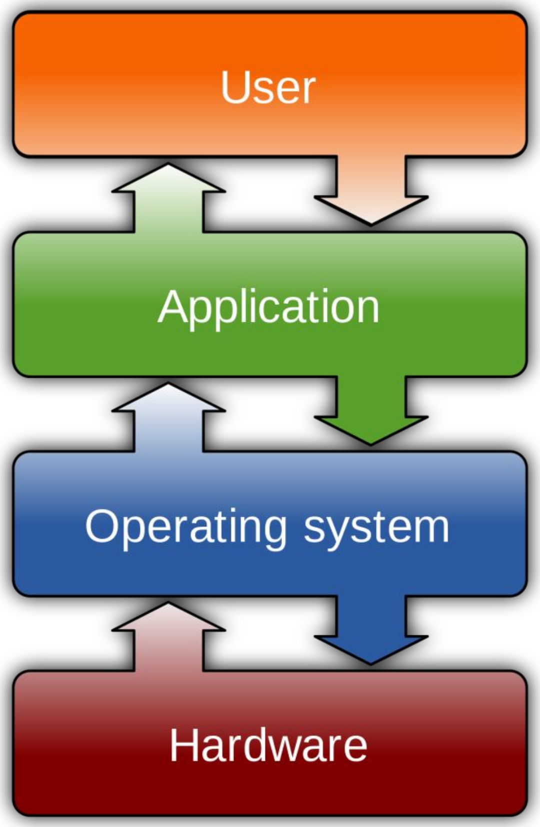
3. Basic Commands

1. Shell
2. Prompt
3. Basic Commands

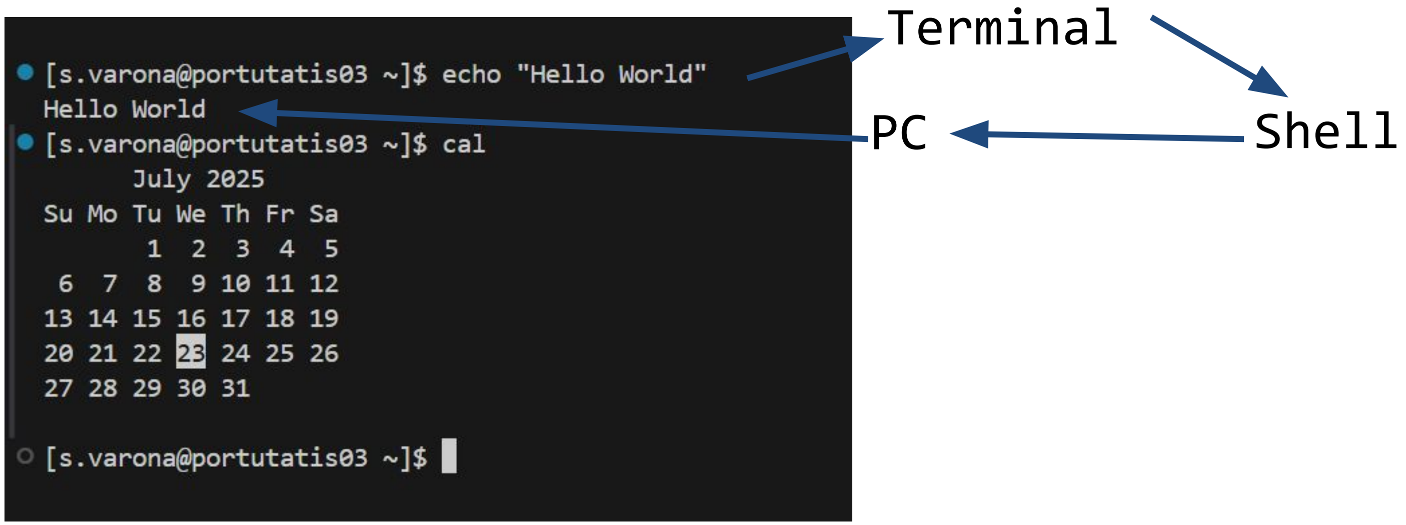
3.1. Shell

Basic structure of an OS:



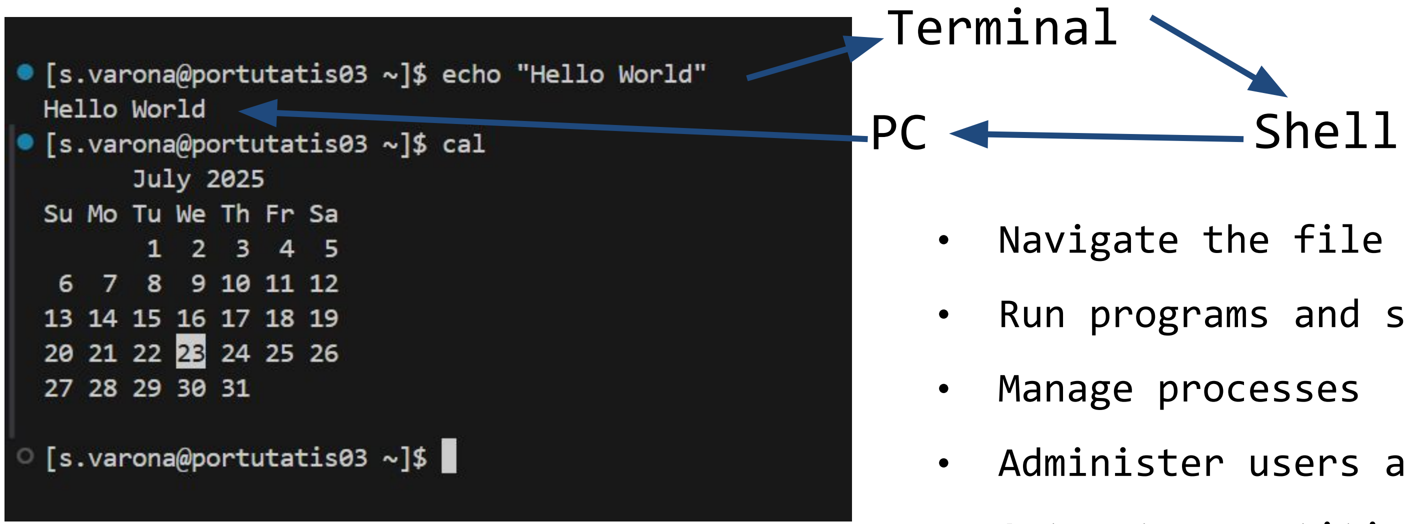


3.1. Shell


```
[s.varona@portutatis03 ~]$ echo "Hello World"
Hello World
[s.varona@portutatis03 ~]$ cal
      July 2025
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[s.varona@portutatis03 ~]$ 
```

3.1. Shell



```
[s.varona@portutatis03 ~]$ echo "Hello World"
Hello World
[s.varona@portutatis03 ~]$ cal
      July 2025
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[s.varona@portutatis03 ~]$
```

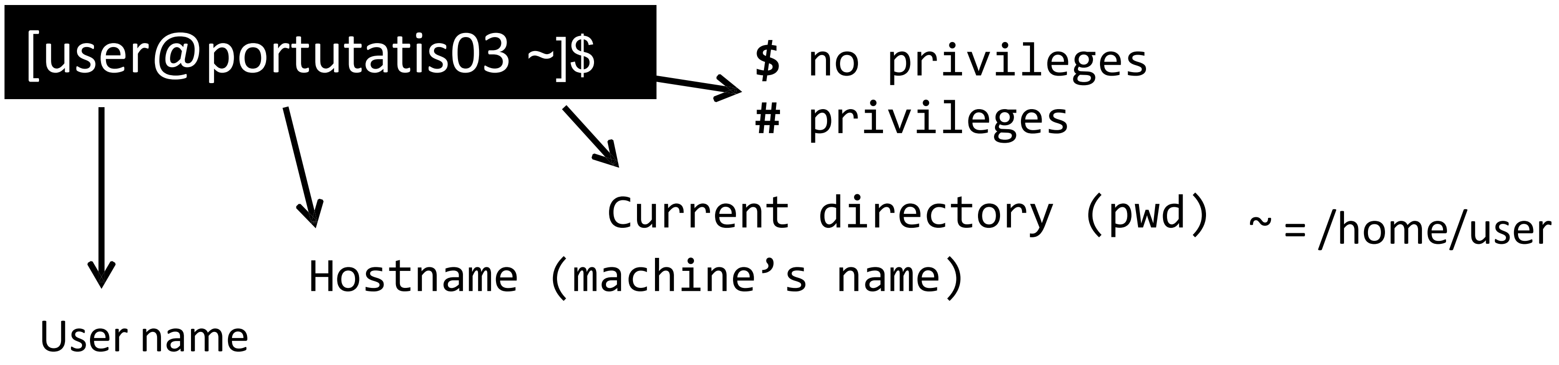
- Navigate the file system
- Run programs and scripts
- Manage processes
- Administer users and permissions
- Automate repetitive tasks

3. Basic Commands

1. Shell
2. Prompt
3. Basic Commands

3.2. Shell’s prompt

- Shell waits for the user to write commands in a line called prompt.
- Prompt line gives some important information that can be easily understood:



```
[user@portutatis03 ~]$
```

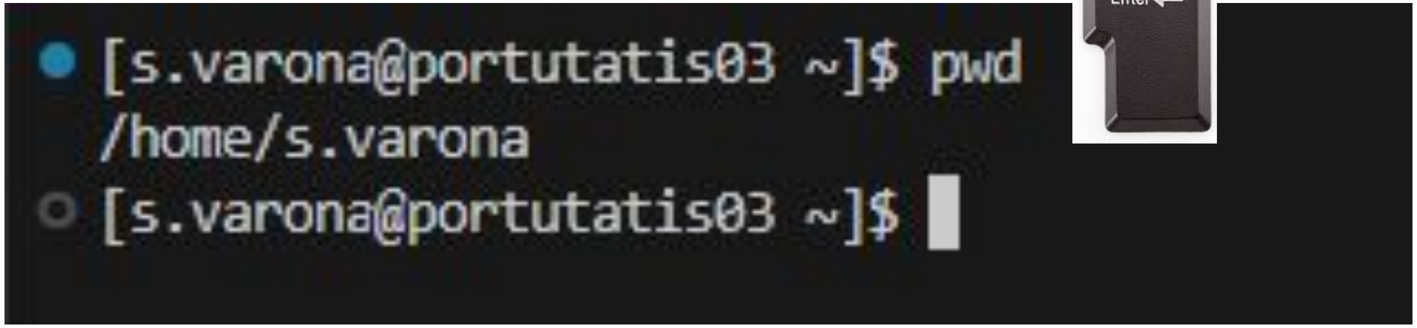
- This example prompt gives the information that the user is user, which has no admin privileges, which is connected to portutatis03 node in the HPC, and the directory where user is located is the /home/user folder.

3. Basic Commands

1. Shell
2. Prompt
3. Basic Commands

pwd (Print Working Directory)

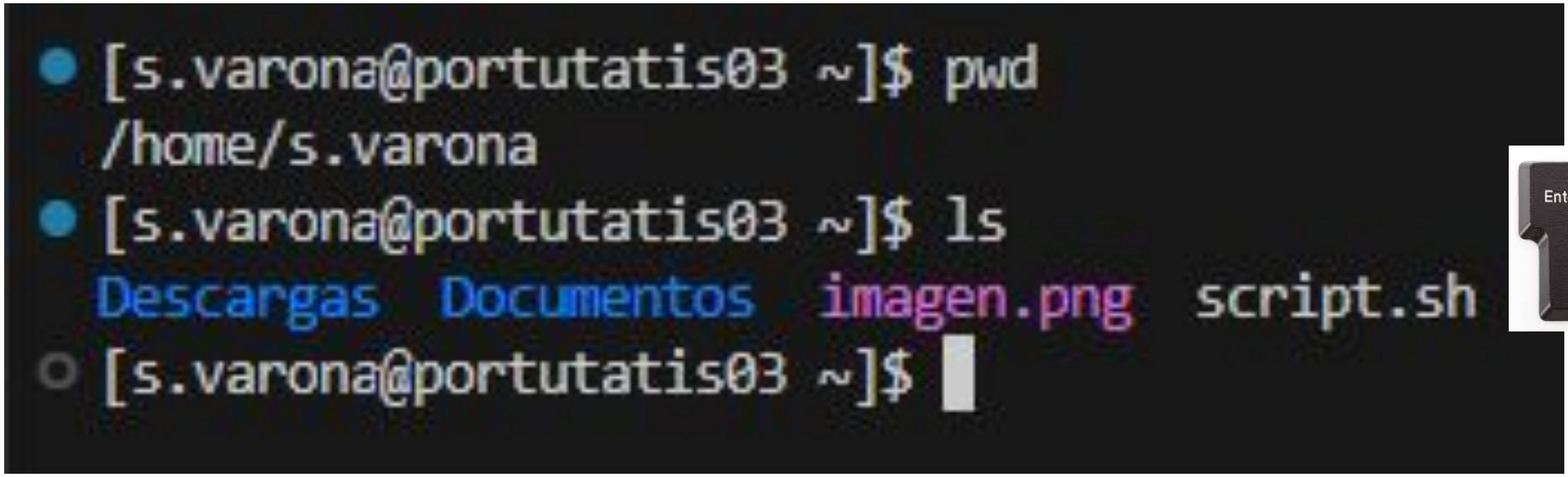
Shows the current directory



```
[s.varona@portutatis03 ~]$ pwd
/home/s.varona
[s.varona@portutatis03 ~]$
```

ls (List)

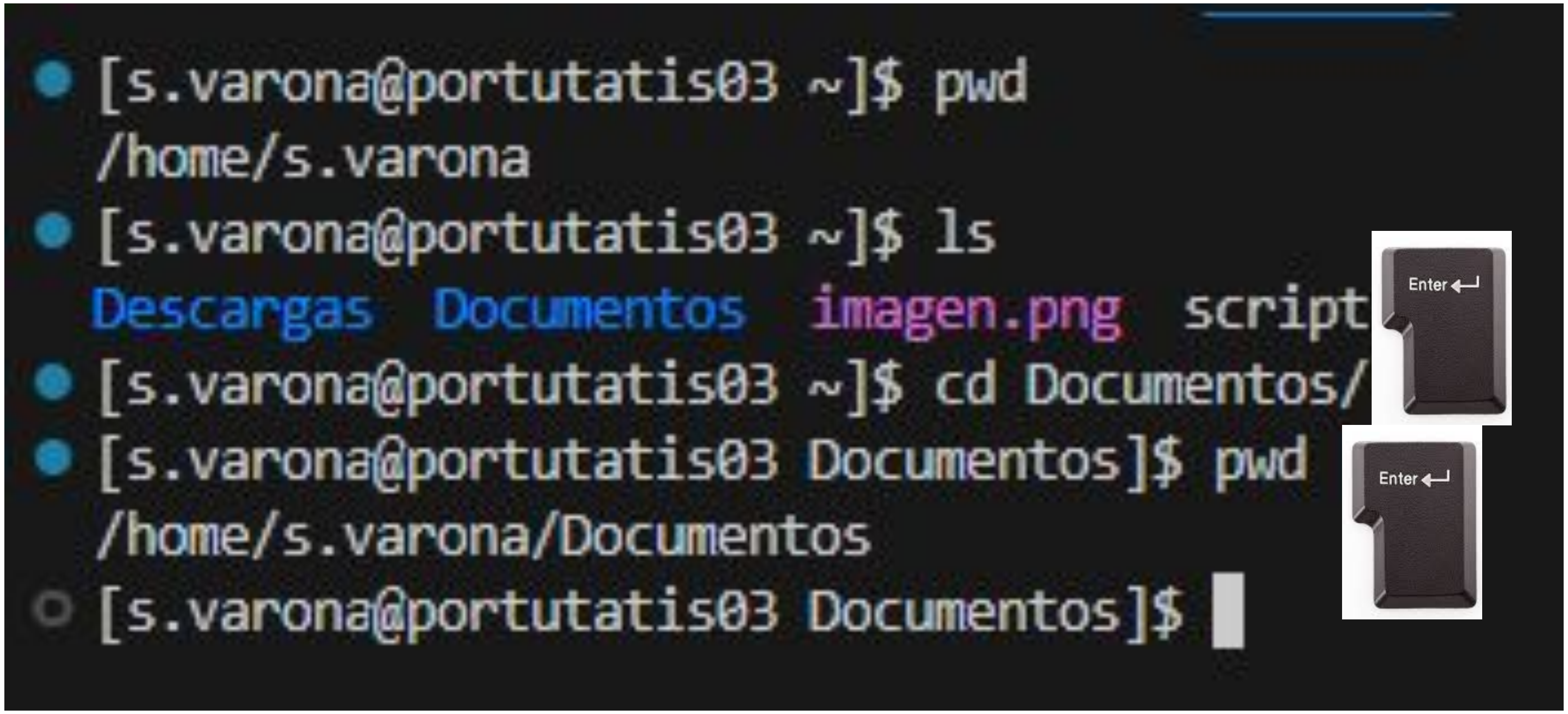
List the content of a directory



```
[s.varona@portutatis03 ~]$ pwd
/home/s.varona
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  imagen.png  script.sh
[s.varona@portutatis03 ~]$
```

cd (Change Directory)

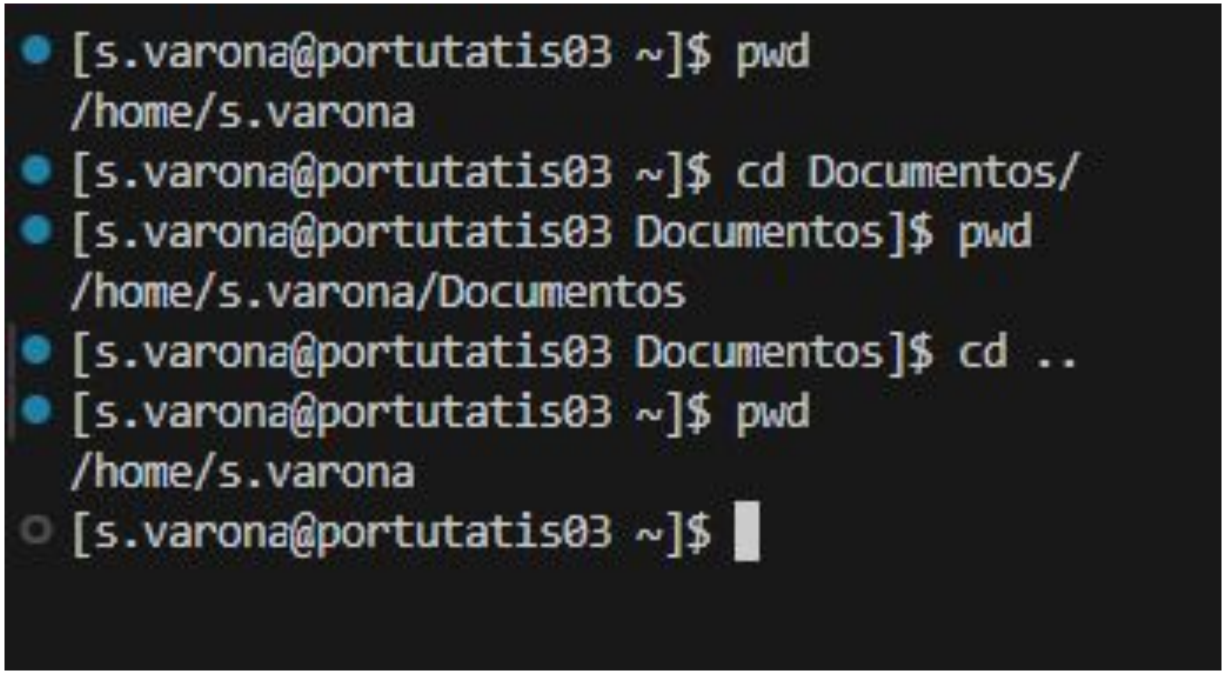
Move from one directory to another



```
[s.varona@portutatis03 ~]$ pwd
/home/s.varona
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  imagen.png  script
[s.varona@portutatis03 ~]$ cd Documentos/
[s.varona@portutatis03 Documentos]$ pwd
/home/s.varona/Documentos
[s.varona@portutatis03 Documentos]$
```

cd (Change Direcotry)

Move from one directory to another



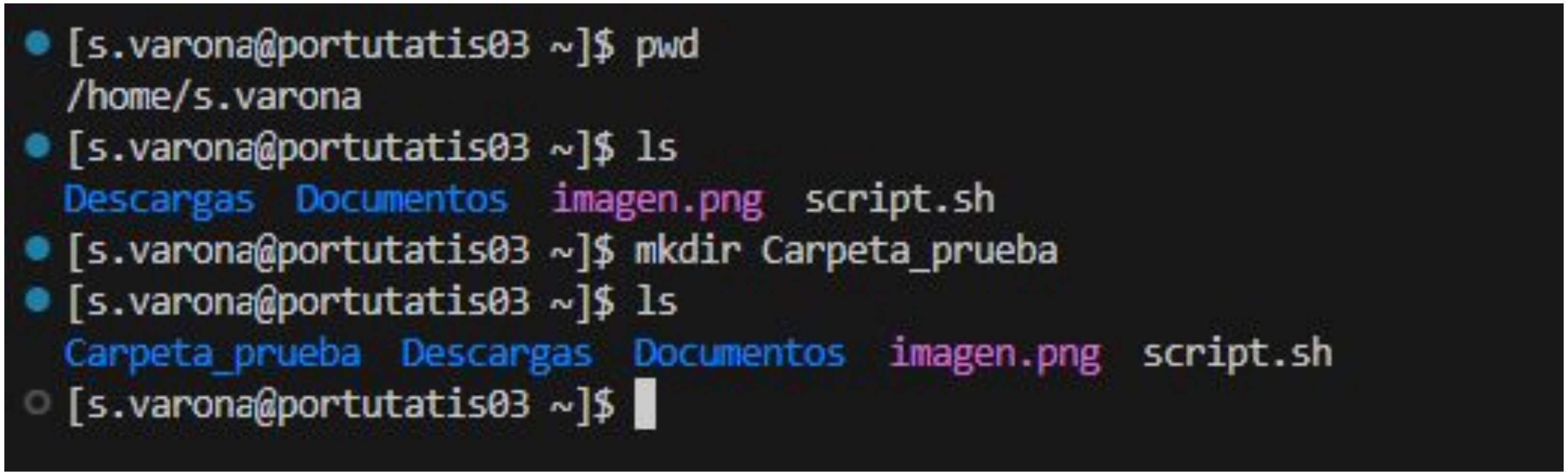
```
[s.varona@portutatis03 ~]$ pwd
/home/s.varona
[s.varona@portutatis03 ~]$ cd Documentos/
[s.varona@portutatis03 Documentos]$ pwd
/home/s.varona/Documentos
[s.varona@portutatis03 Documentos]$ cd ..
[s.varona@portutatis03 ~]$ pwd
/home/s.varona
[s.varona@portutatis03 ~]$
```

. = Current pwd

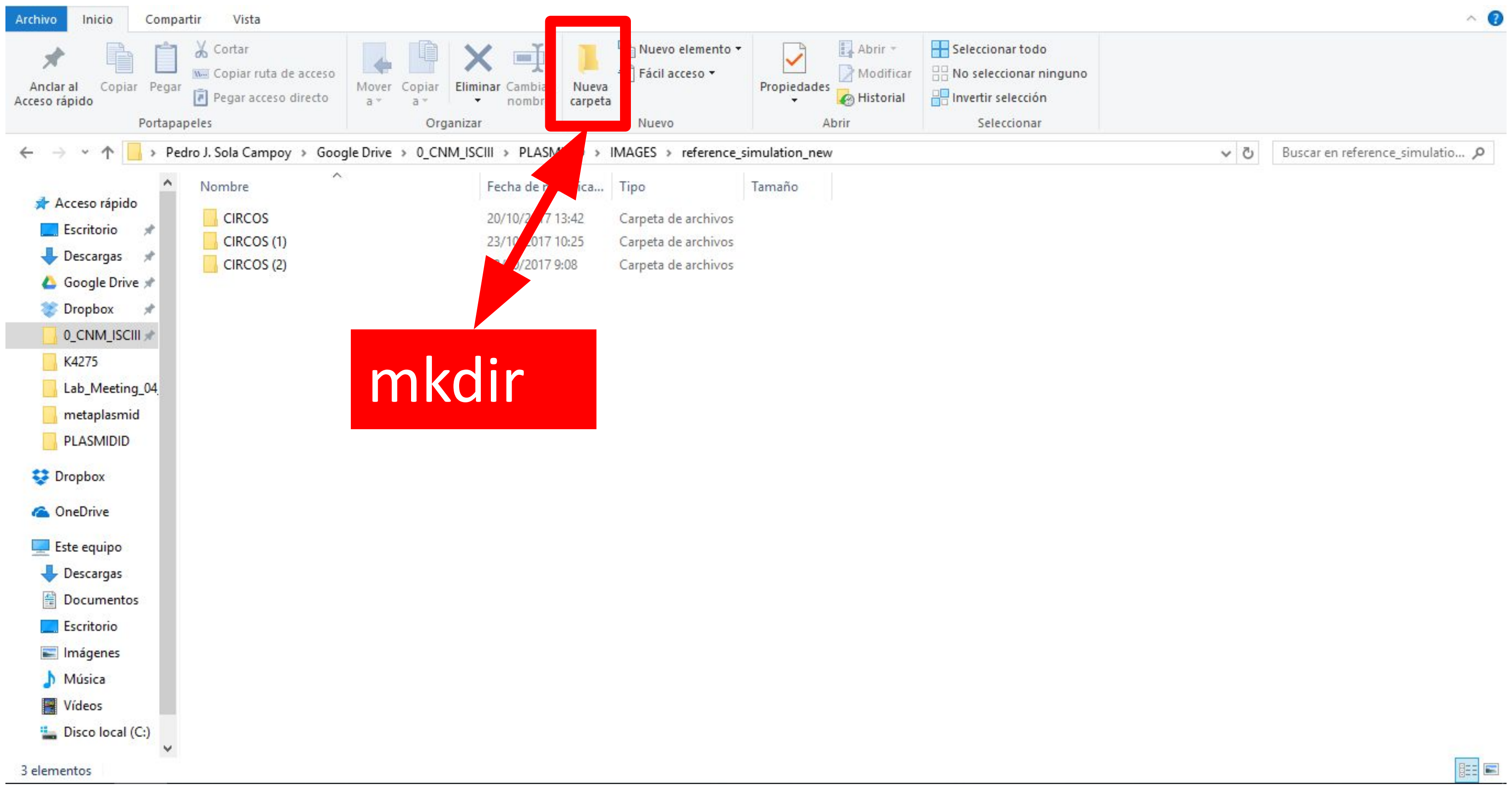
.. = Parent directory

mkdir (Make Directory)

Create an empty directory


```
[s.varona@portutatis03 ~]$ pwd
/home/s.varona
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  imagen.png  script.sh
[s.varona@portutatis03 ~]$ mkdir Carpeta_prueba
[s.varona@portutatis03 ~]$ ls
Carpeta_prueba  Descargas  Documentos  imagen.png  script.sh
[s.varona@portutatis03 ~]$ 
```

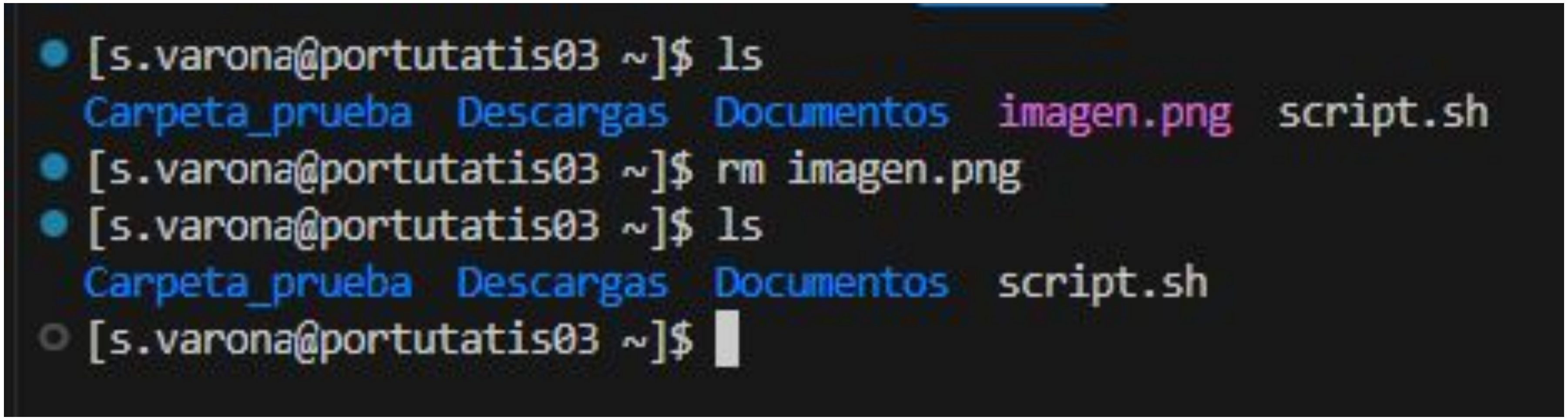
mkdir (Make Directory)



Nombre	Fecha de modifica...	Tipo	Tamaño
CIRCOS	20/10/2[illegible]7 13:42	Carpeta de archivos	
CIRCOS (1)	23/10[illegible]017 10:25	Carpeta de archivos	
CIRCOS (2)	[illegible]/2017 9:08	Carpeta de archivos	

rm (Remove)

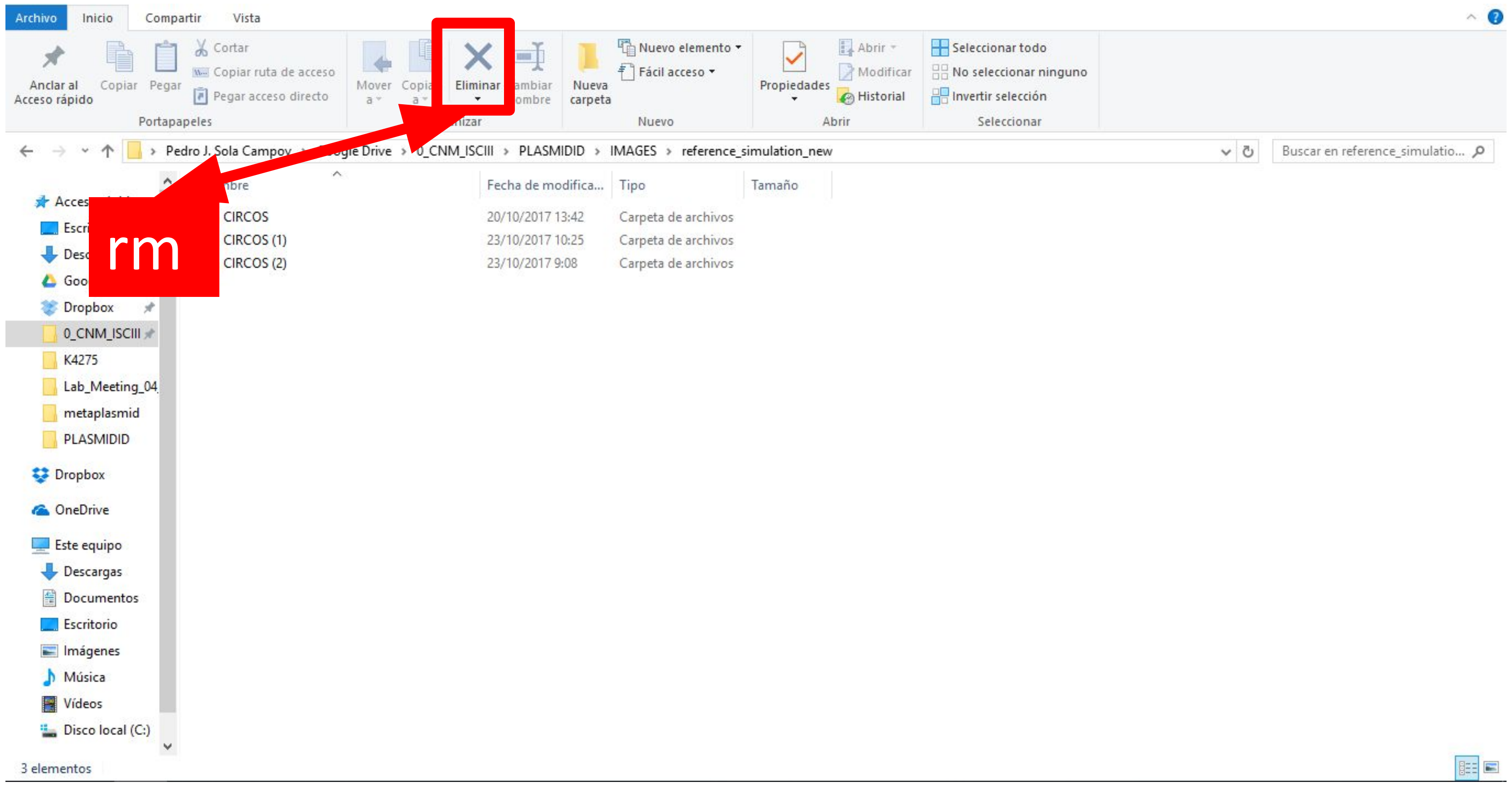
Remove files



```
[s.varona@portutatis03 ~]$ ls
Carpeta_prueba  Descargas  Documentos  imagen.png  script.sh
[s.varona@portutatis03 ~]$ rm imagen.png
[s.varona@portutatis03 ~]$ ls
Carpeta_prueba  Descargas  Documentos  script.sh
[s.varona@portutatis03 ~]$
```

It doesn’t ask for a confirmation

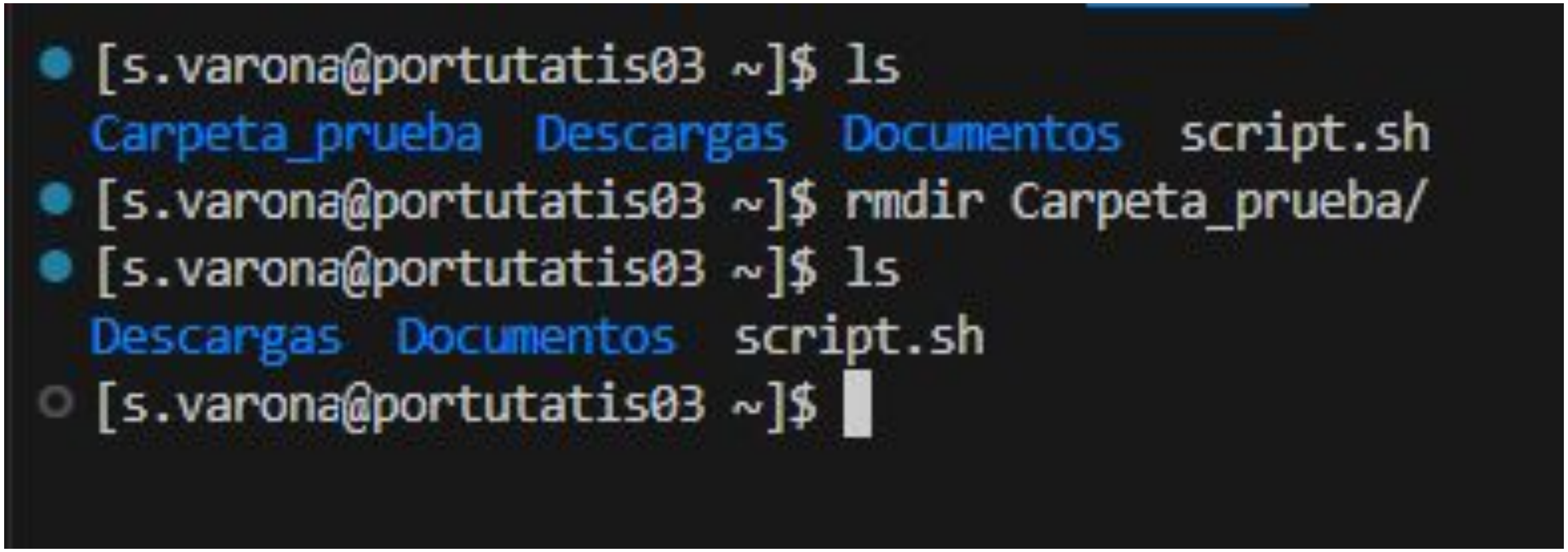
rm (Remove)



Nombre	Fecha de modifica...	Tipo	Tamaño
CIRCOS	20/10/2017 13:42	Carpeta de archivos	
CIRCOS (1)	23/10/2017 10:25	Carpeta de archivos	
CIRCOS (2)	23/10/2017 9:08	Carpeta de archivos	

rmdir (Remove Directory)

Remove empty directories

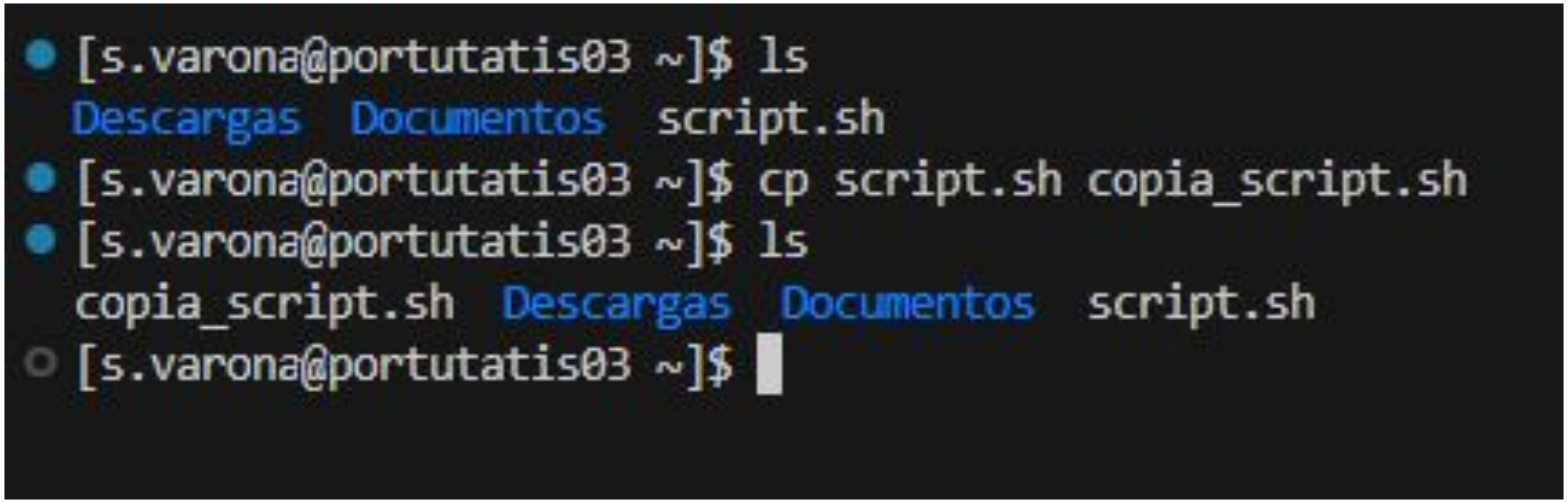


```
● [s.varona@portutatis03 ~]$ ls
  Carpeta_prueba  Descargas  Documentos  script.sh
● [s.varona@portutatis03 ~]$ rmdir Carpeta_prueba/
● [s.varona@portutatis03 ~]$ ls
  Descargas  Documentos  script.sh
○ [s.varona@portutatis03 ~]$ █
```

It doesn’t ask for a confirmation

cp (Copy)

Copy a file



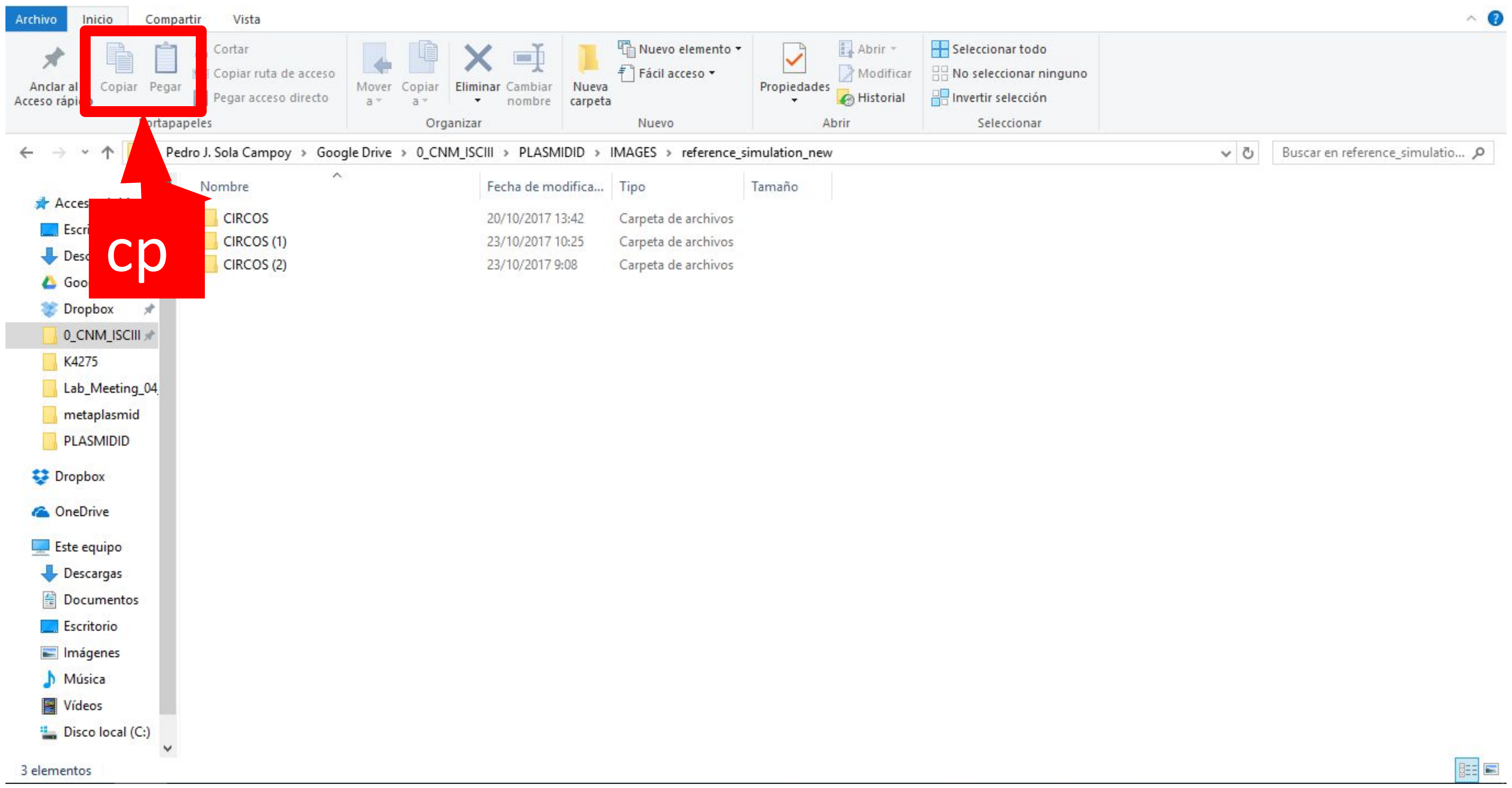
```
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  script.sh
[s.varona@portutatis03 ~]$ cp script.sh copia_script.sh
[s.varona@portutatis03 ~]$ ls
copia_script.sh  Descargas  Documentos  script.sh
[s.varona@portutatis03 ~]$ 
```

GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

Instituto de Salud Carlos III

BU-ISCIII

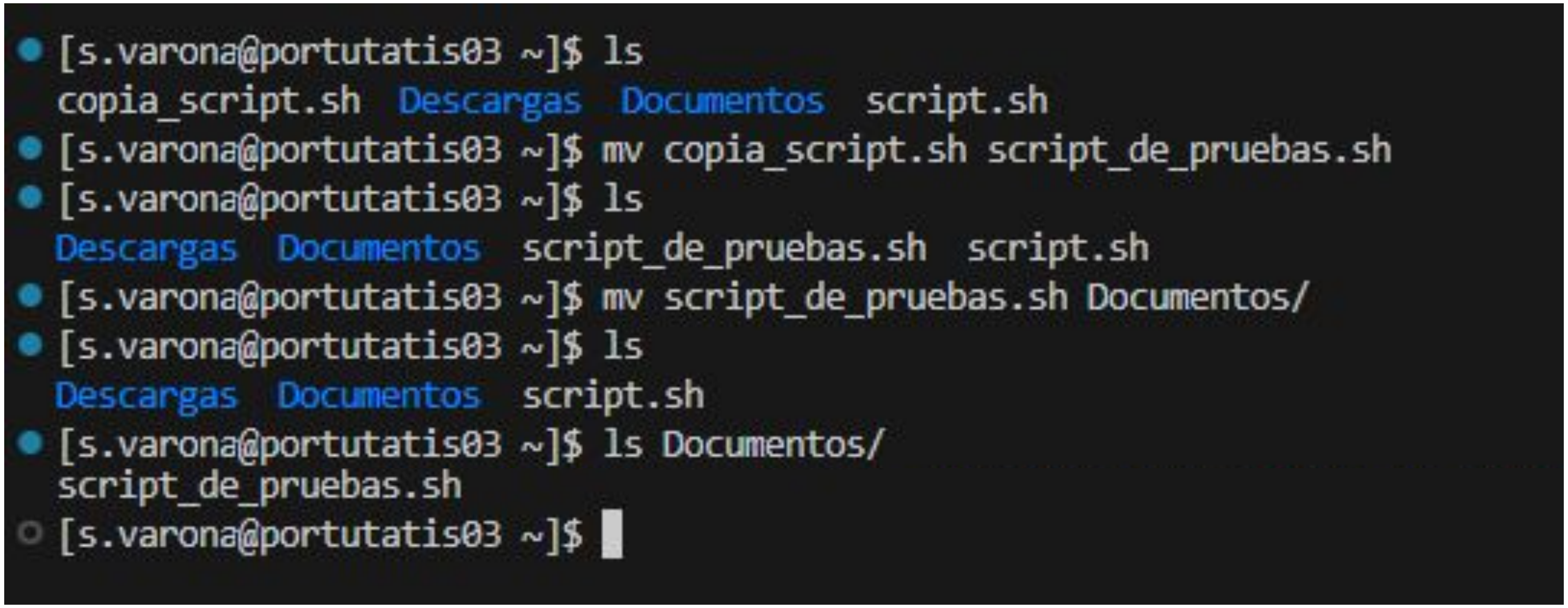
cp (Copy)



Nombre	Fecha de modifica...	Tipo	Tamaño
CIRCOS	20/10/2017 13:42	Carpeta de archivos	
CIRCOS (1)	23/10/2017 10:25	Carpeta de archivos	
CIRCOS (2)	23/10/2017 9:08	Carpeta de archivos	

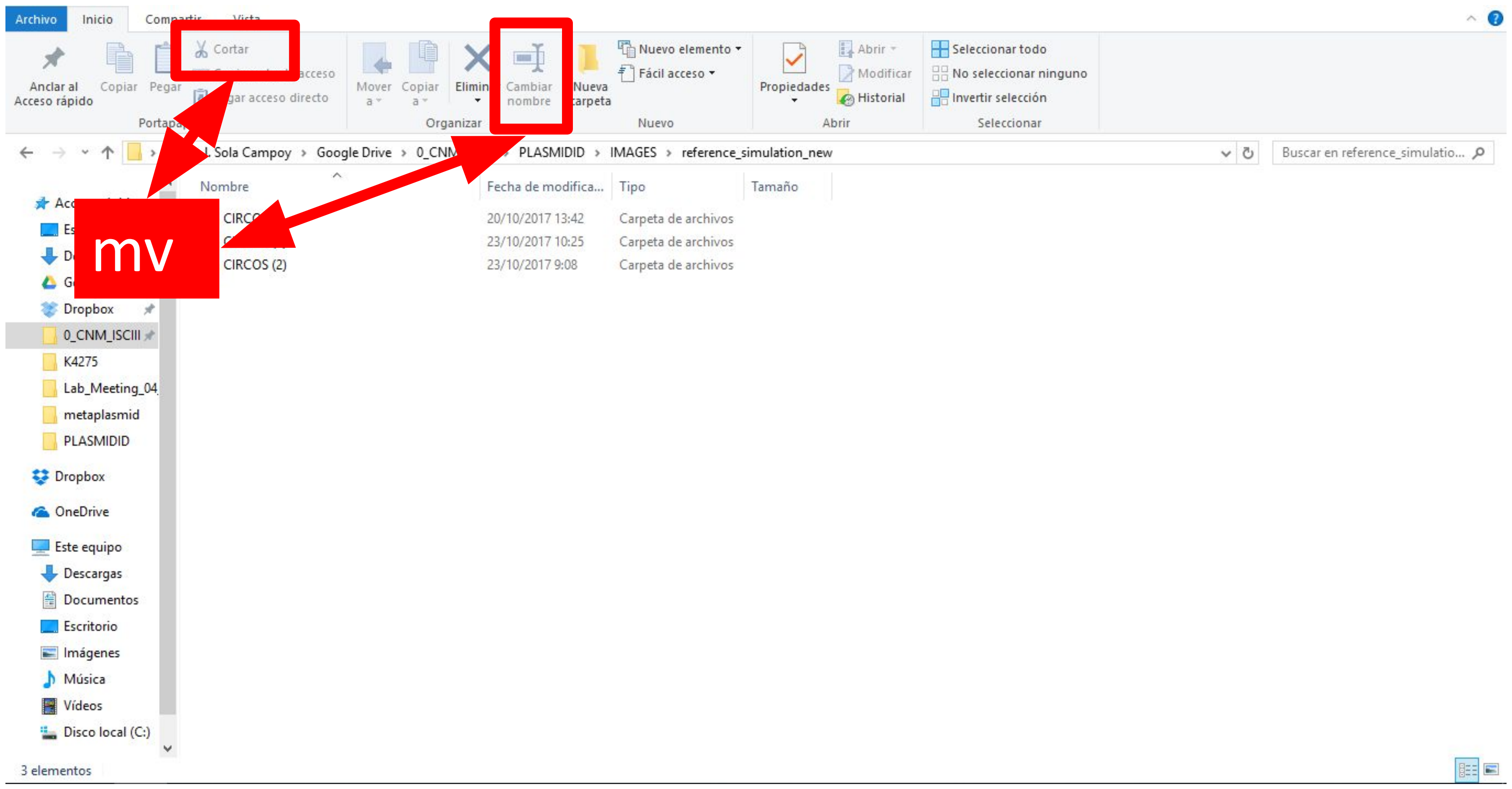
mv (Move)

Rename a file or move it to another directory



```
[s.varona@portutatis03 ~]$ ls
copia_script.sh  Descargas  Documentos  script.sh
[s.varona@portutatis03 ~]$ mv copia_script.sh script_de_pruebas.sh
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  script_de_pruebas.sh  script.sh
[s.varona@portutatis03 ~]$ mv script_de_pruebas.sh Documentos/
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  script.sh
[s.varona@portutatis03 ~]$ ls Documentos/
script_de_pruebas.sh
[s.varona@portutatis03 ~]$
```

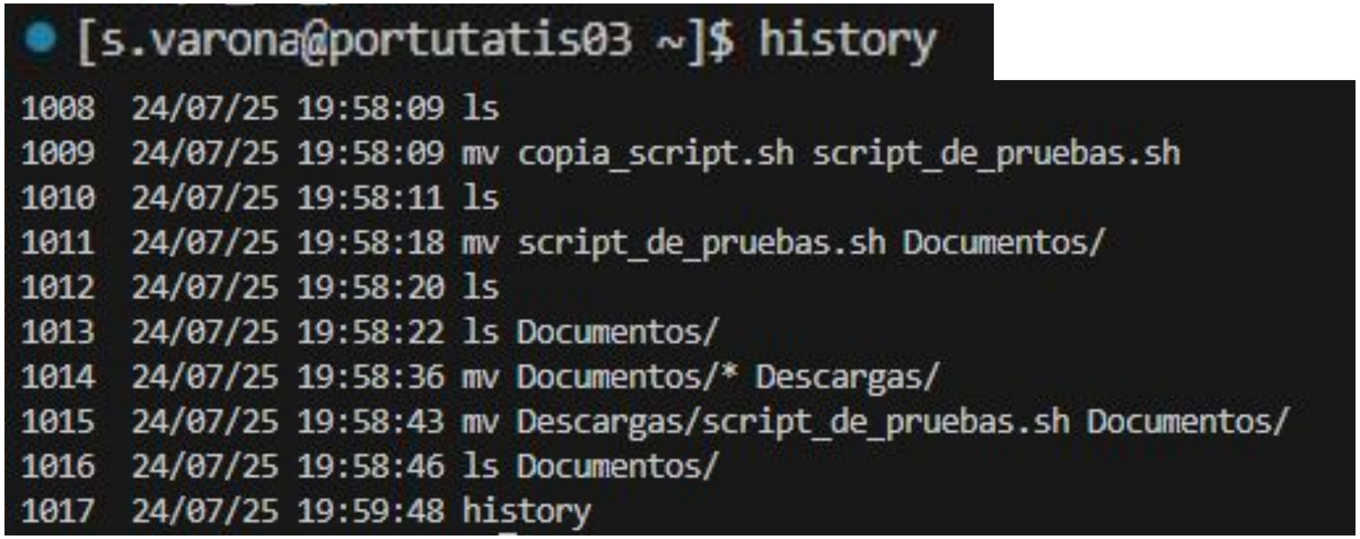
mv (Move)



Nombre	Fecha de modifica...	Tipo	Tamaño
CIRC	20/10/2017 13:42	Carpeta de archivos	
C	23/10/2017 10:25	Carpeta de archivos	
CIRCOS (2)	23/10/2017 9:08	Carpeta de archivos	

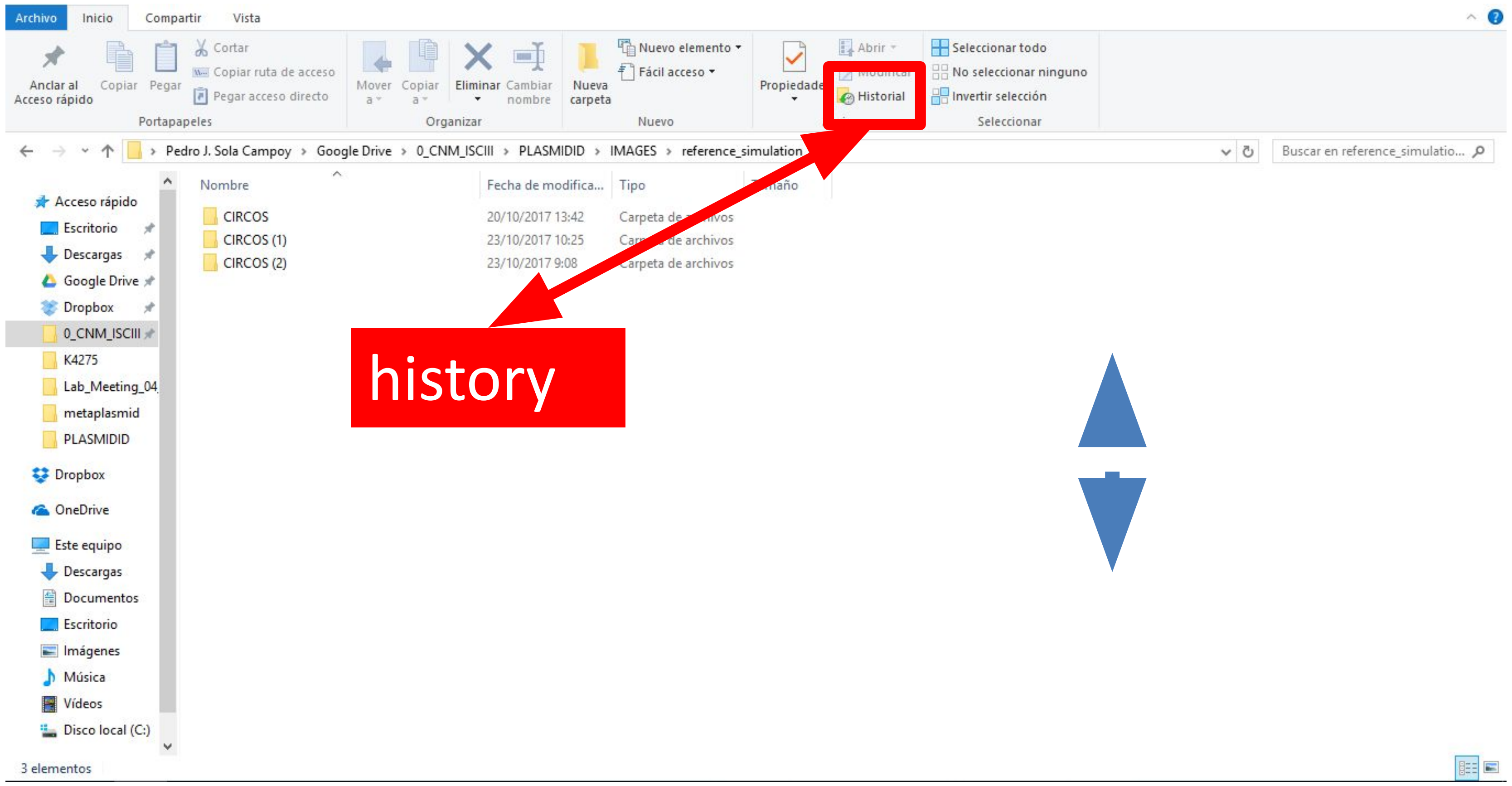
history

Show history of commands used



```
● [s.varona@portutatis03 ~]$ history
1008  24/07/25 19:58:09 ls
1009  24/07/25 19:58:09 mv copia_script.sh script_de_pruebas.sh
1010  24/07/25 19:58:11 ls
1011  24/07/25 19:58:18 mv script_de_pruebas.sh Documentos/
1012  24/07/25 19:58:20 ls
1013  24/07/25 19:58:22 ls Documentos/
1014  24/07/25 19:58:36 mv Documentos/* Descargas/
1015  24/07/25 19:58:43 mv Descargas/script_de_pruebas.sh Documentos/
1016  24/07/25 19:58:46 ls Documentos/
1017  24/07/25 19:59:48 history
```

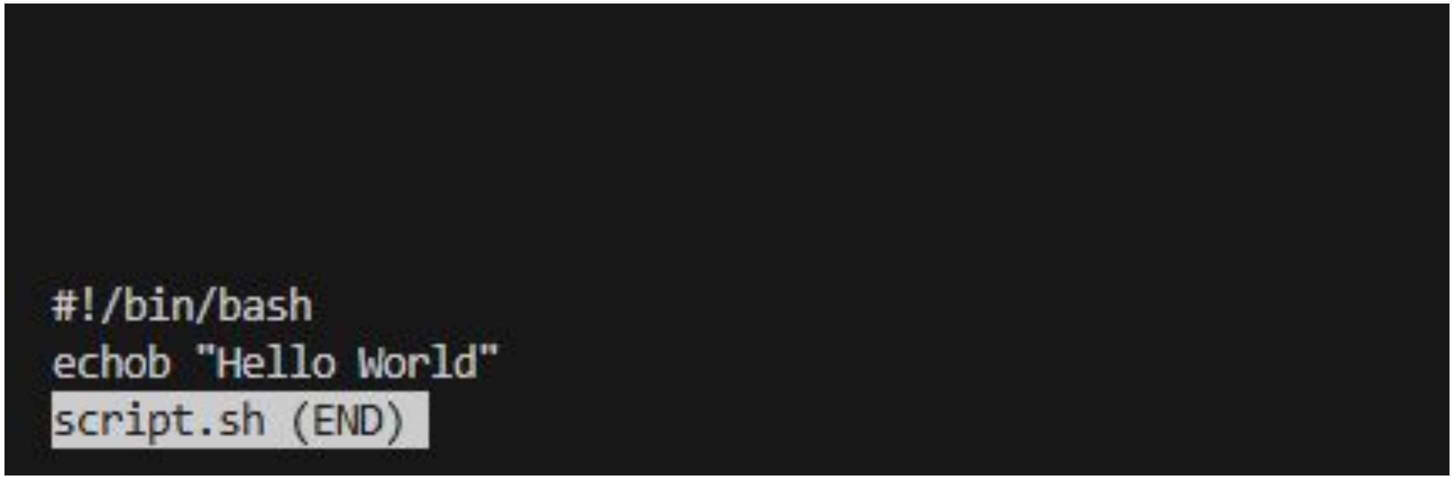
history



Nombre	Fecha de modifica...	Tipo	Tamaño
CIRCOS	20/10/2017 13:42	Carpeta de archivos	
CIRCOS (1)	23/10/2017 10:25	Carpeta de archivos	
CIRCOS (2)	23/10/2017 9:08	Carpeta de archivos	

less (Historial)

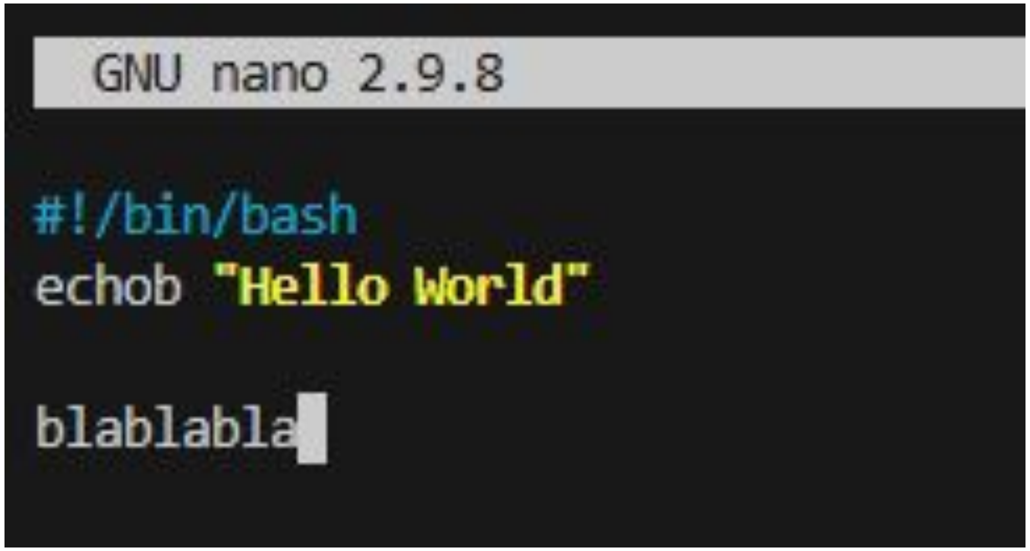
Show content of a file



```
#!/bin/bash
echob "Hello World"
script.sh (END)
```

nano

Text editor



```
GNU nano 2.9.8

#!/bin/bash
echob "Hello World"

blablabla
```

Save: ctrl+o > Intro

Exit: ctrl+x

Basic Commands

REMEMBER:

TAB is your friend!

Hit it to autocomplete a command, file, path or get suggestion to do it

Overview

4. Command Line Syntax

1. Syntax
2. Input
3. Output
4. Redirect
5. Pipes
6. WildCards and Regular expressions

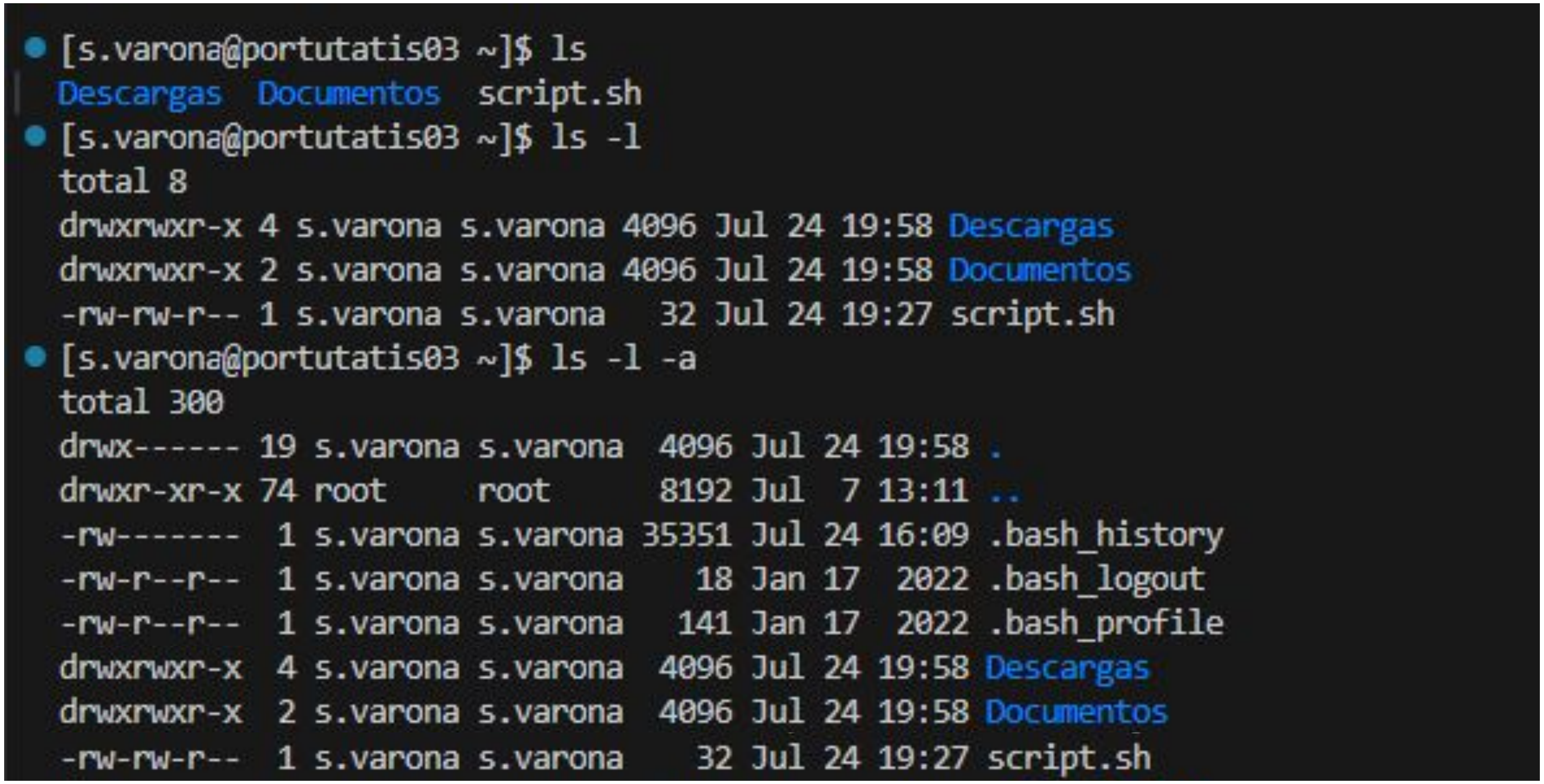
4.1. Command Line Syntax

Linux command line follows a simple syntax common to every command and program you can execute on it:

```
Command [options o parameters] [arguments]
```

- Options are characters or words preceded by a dash (`ls -la`). They change the way a program works by default.
- Arguments are other parameters that the program may need to run. The most common ones usually are the input files.
- REMEMBER: most programs have a –h or --help option which displays a short description and usage guide.

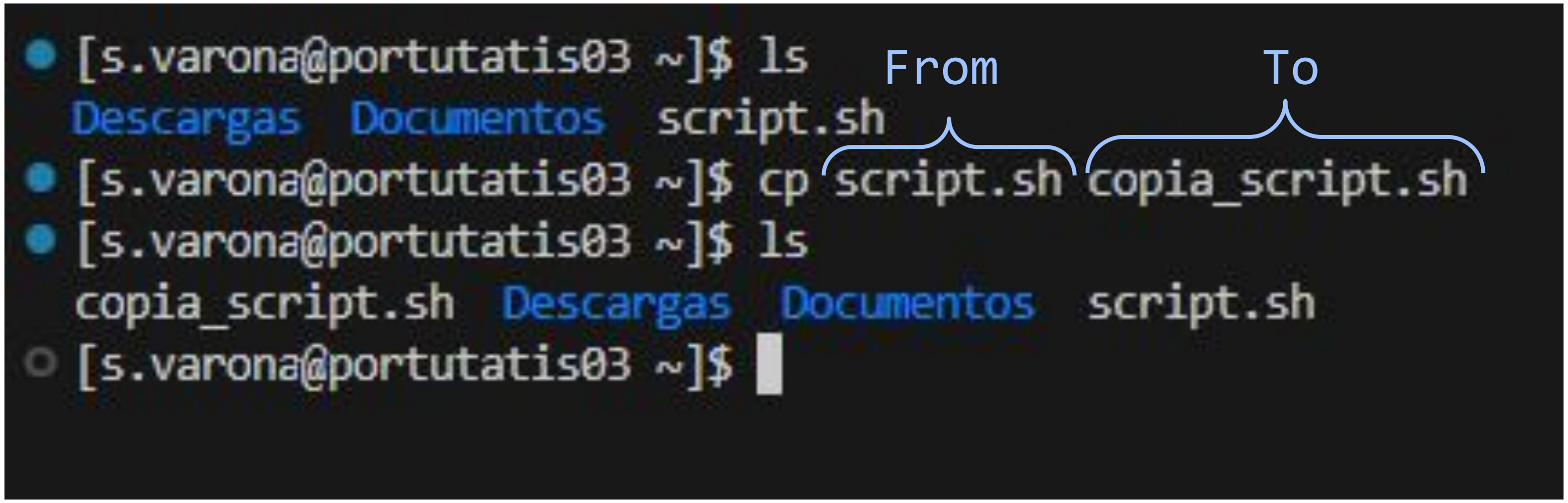
4.1. Command Line Syntax



```
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  script.sh
[s.varona@portutatis03 ~]$ ls -l
total 8
drwxrwxr-x 4 s.varona s.varona 4096 Jul 24 19:58 Descargas
drwxrwxr-x 2 s.varona s.varona 4096 Jul 24 19:58 Documentos
-rw-rw-r-- 1 s.varona s.varona   32 Jul 24 19:27 script.sh
[s.varona@portutatis03 ~]$ ls -l -a
total 300
drwx------ 19 s.varona s.varona  4096 Jul 24 19:58 .
drwxr-xr-x 74 root     root      8192 Jul  7 13:11 ..
-rw-------  1 s.varona s.varona 35351 Jul 24 16:09 .bash_history
-rw-r--r--  1 s.varona s.varona    18 Jan 17  2022 .bash_logout
-rw-r--r--  1 s.varona s.varona   141 Jan 17  2022 .bash_profile
drwxrwxr-x  4 s.varona s.varona  4096 Jul 24 19:58 Descargas
drwxrwxr-x  2 s.varona s.varona  4096 Jul 24 19:58 Documentos
-rw-rw-r--  1 s.varona s.varona    32 Jul 24 19:27 script.sh
```

Instituto de Salud Carlos III

4.1. Command Line Syntax



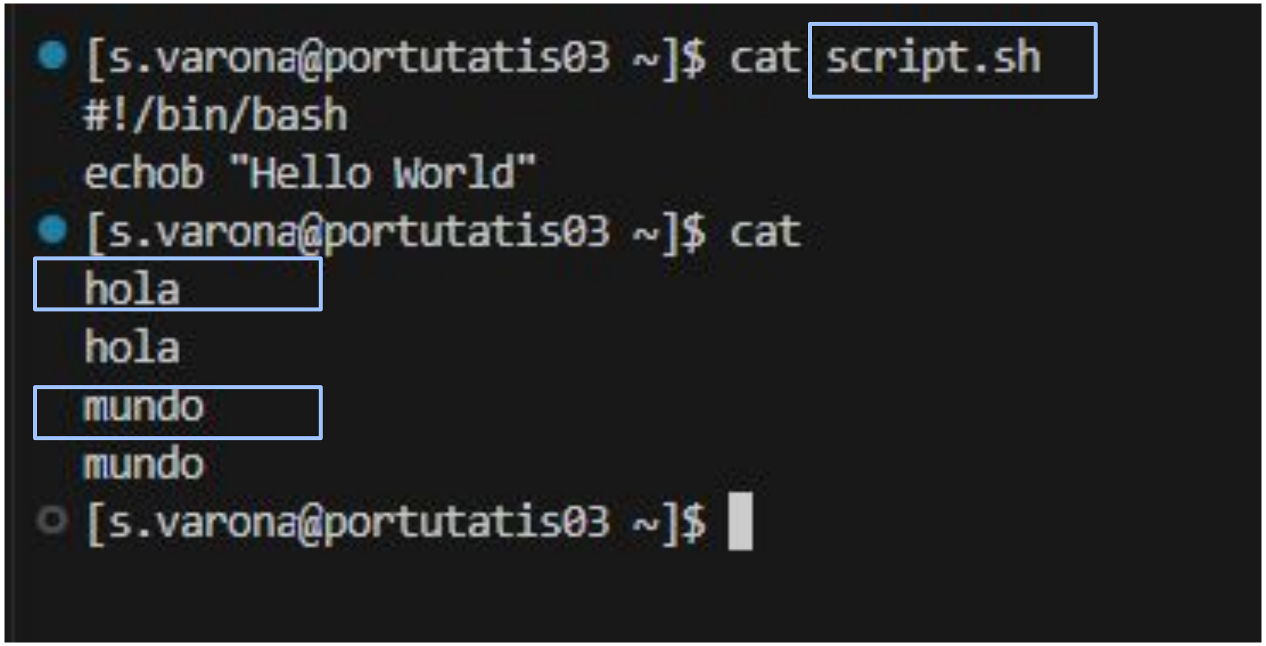
```
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  script.sh
[s.varona@portutatis03 ~]$ cp script.sh copia_script.sh
[s.varona@portutatis03 ~]$ ls
copia_script.sh  Descargas  Documentos  script.sh
[s.varona@portutatis03 ~]$
```

4. Command Line Syntax

1. Syntax
2. Input
3. Output
4. Redirect
5. Pipes
6. WildCards and Regular expressions

4.2. Input

Standard Input (stdin): The data a command receives.



```
[s.varona@portutatis03 ~]$ cat script.sh
#!/bin/bash
echob "Hello World"
[s.varona@portutatis03 ~]$ cat
hola
hola
mundo
mundo
[s.varona@portutatis03 ~]$
```

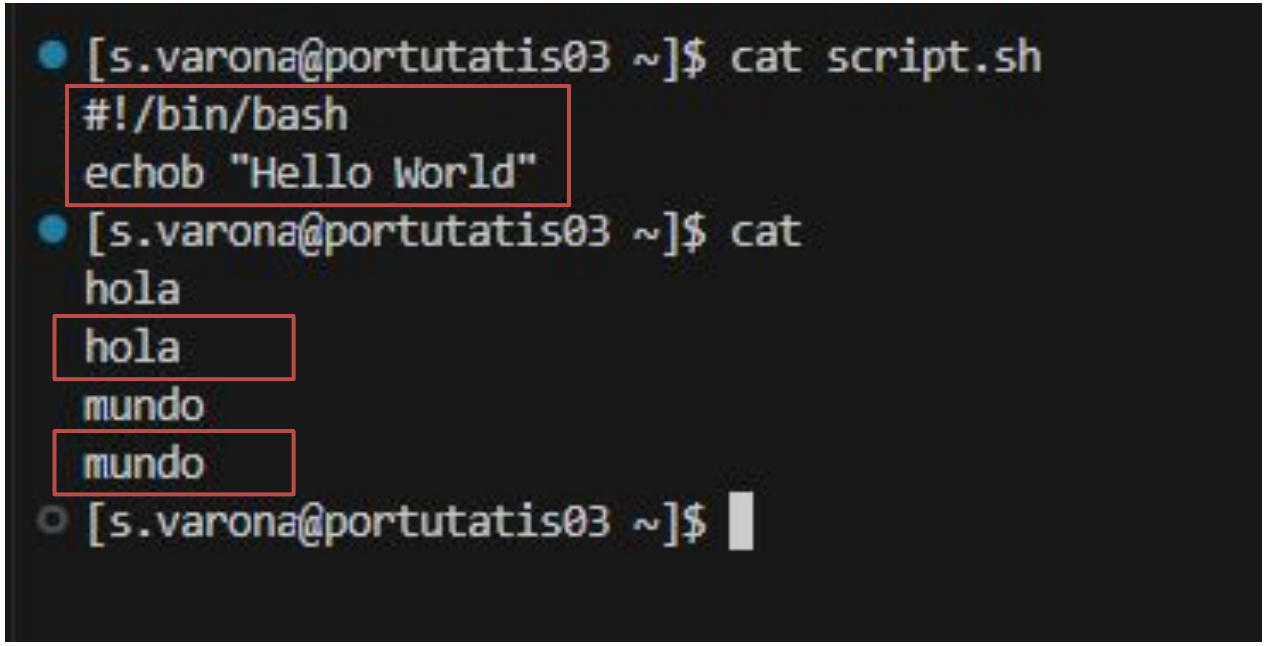
4. Command Line Syntax

1. Syntax
2. Input
3. Output
4. Redirect
5. Pipes
6. WildCards and Regular expressions

4.3. Output

Standard Output (stdout): data a command sends back as a result.

By default, it is printed to the screen.

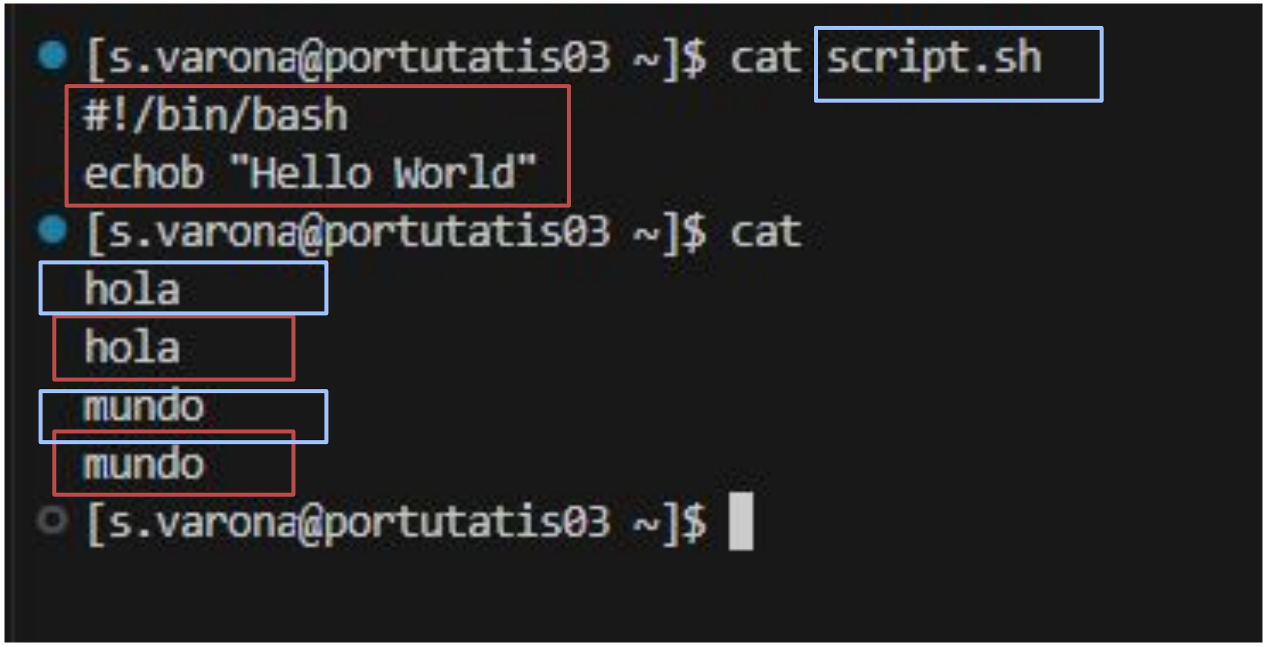


```
● [s.varona@portutatis03 ~]$ cat script.sh
  #!/bin/bash
  echob "Hello World"
● [s.varona@portutatis03 ~]$ cat
  hola
  hola
  mundo
  mundo
○ [s.varona@portutatis03 ~]$ 
```

Input/Output

stdin

stdout



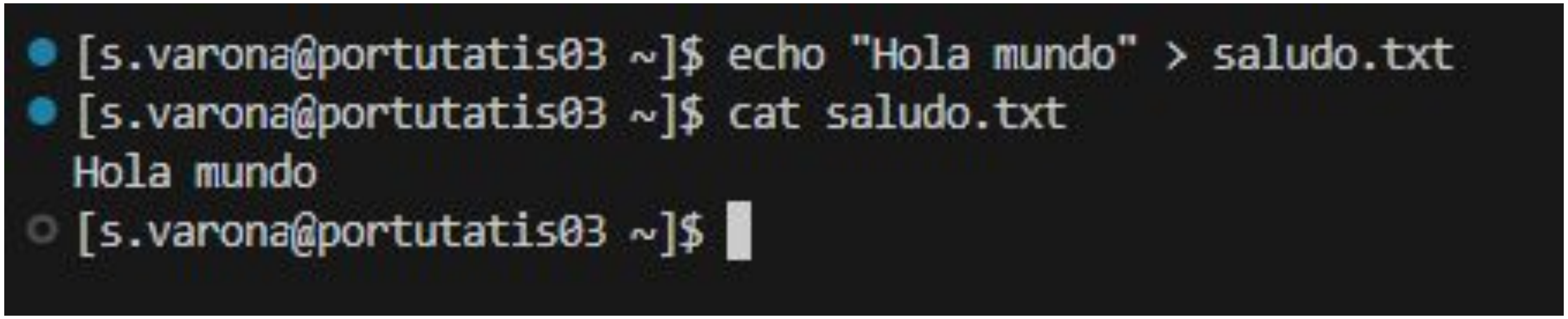
```
[s.varona@portutatis03 ~]$ cat script.sh
#!/bin/bash
echob "Hello World"
[s.varona@portutatis03 ~]$ cat
hola
hola
mundo
mundo
[s.varona@portutatis03 ~]$
```

4. Command Line Syntax

1. Syntax
2. Input
3. Output
4. Redirect
5. Pipes

4.4. Redirect

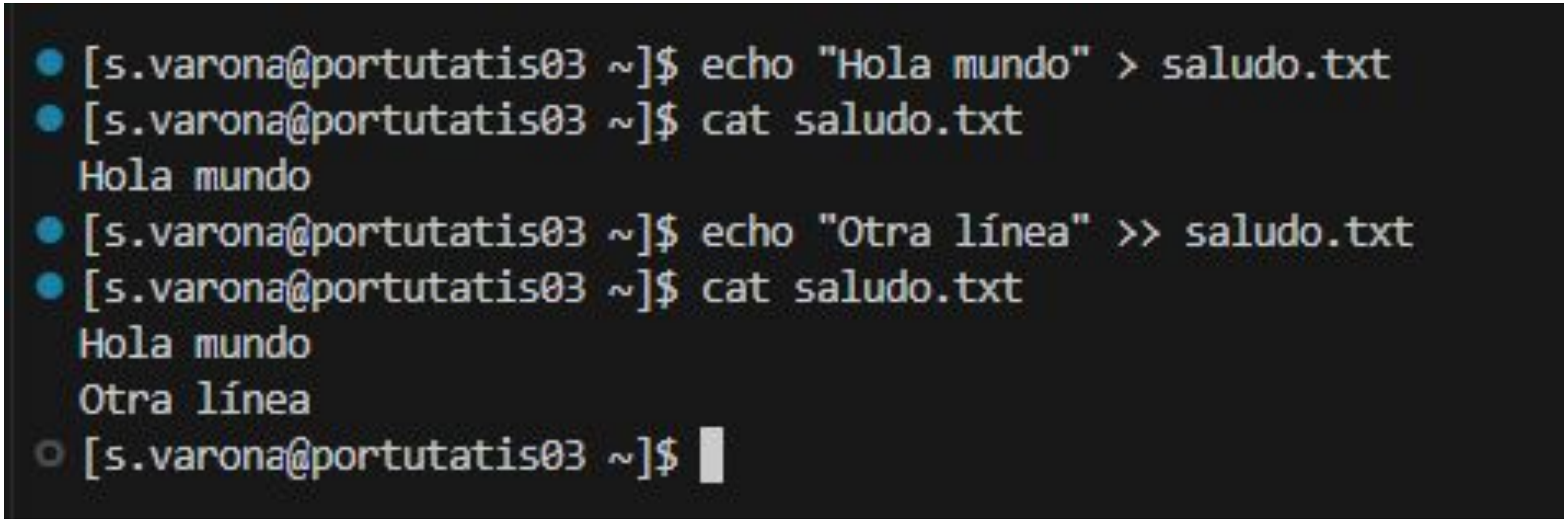
Redirect stdout (>)



```
[s.varona@portutatis03 ~]$ echo "Hola mundo" > saludo.txt
[s.varona@portutatis03 ~]$ cat saludo.txt
Hola mundo
[s.varona@portutatis03 ~]$ 
```

4.4. Redirect

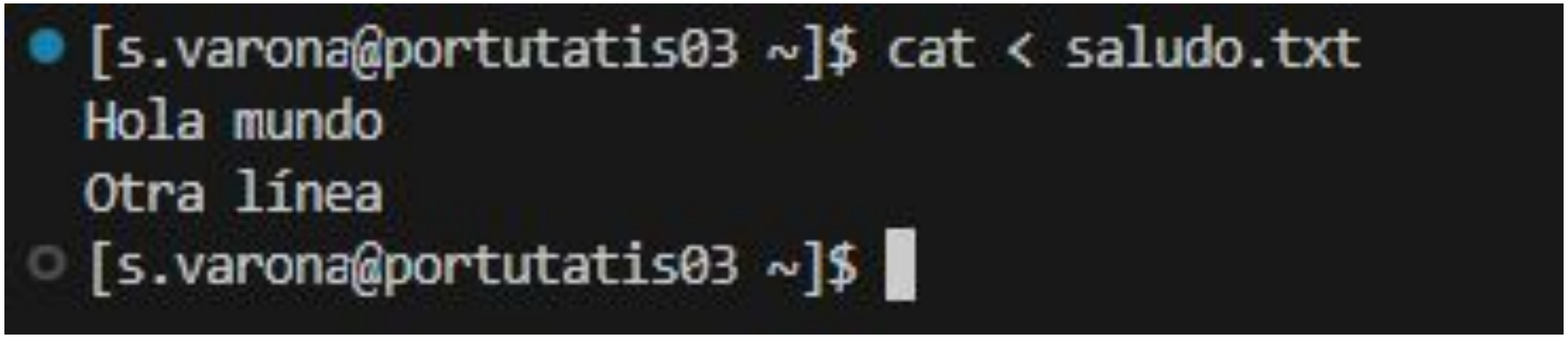
Redirect stdout at the end (>>)



```
[s.varona@portutatis03 ~]$ echo "Hola mundo" > saludo.txt
[s.varona@portutatis03 ~]$ cat saludo.txt
Hola mundo
[s.varona@portutatis03 ~]$ echo "Otra línea" >> saludo.txt
[s.varona@portutatis03 ~]$ cat saludo.txt
Hola mundo
Otra línea
[s.varona@portutatis03 ~]$
```

4.4. Redirect

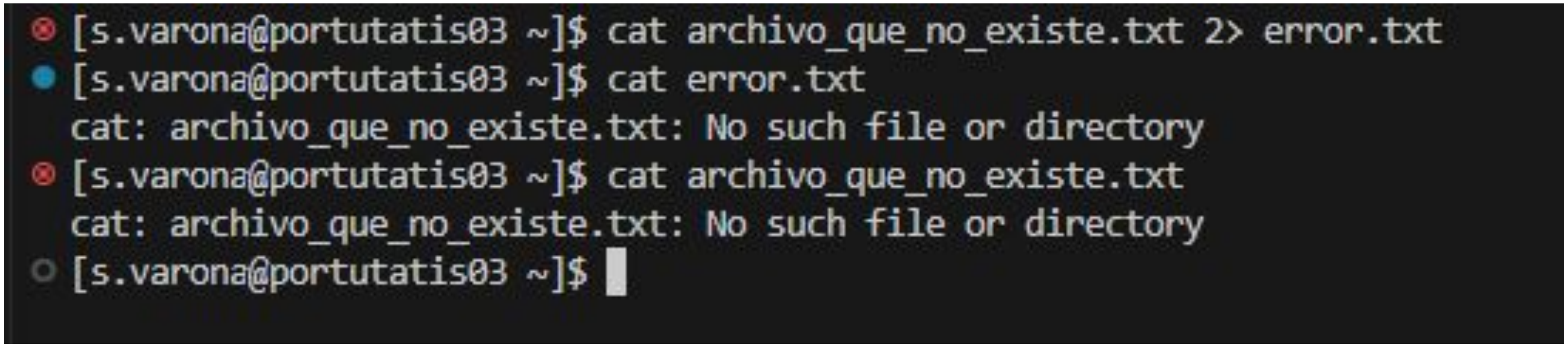
Input from file (<)



```
[s.varona@portutatis03 ~]$ cat < saludo.txt
Hola mundo
Otra línea
[s.varona@portutatis03 ~]$ 
```

4.4. Redirect

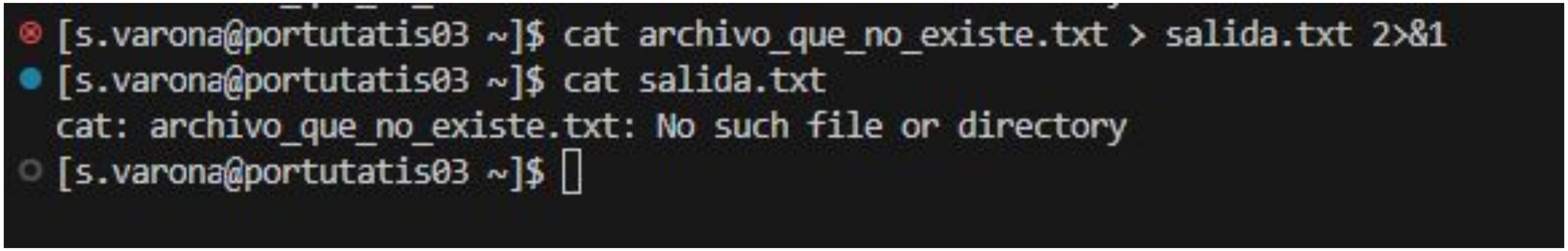
Redirect error log (2>)



```
[s.varona@portutatis03 ~]$ cat archivo_que_no_existe.txt 2> error.txt
[s.varona@portutatis03 ~]$ cat error.txt
cat: archivo_que_no_existe.txt: No such file or directory
[s.varona@portutatis03 ~]$ cat archivo_que_no_existe.txt
cat: archivo_que_no_existe.txt: No such file or directory
[s.varona@portutatis03 ~]$
```

4.4. Redirect

Redirect error log and output (2>&1)



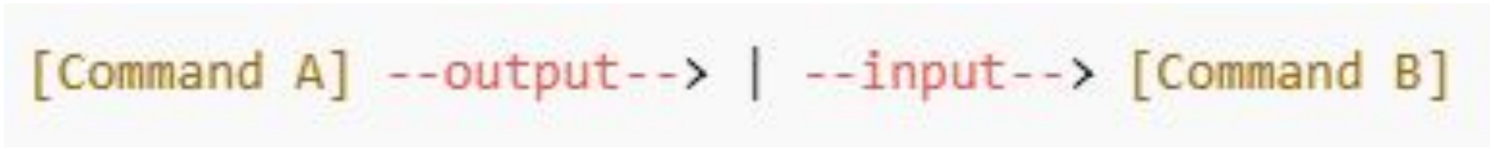
```
[s.varona@portutatis03 ~]$ cat archivo_que_no_existe.txt > salida.txt 2>&1
[s.varona@portutatis03 ~]$ cat salida.txt
cat: archivo_que_no_existe.txt: No such file or directory
[s.varona@portutatis03 ~]$ 
```

4. Command Line Syntax

1. Syntax
2. Input
3. Output
4. Redirect
5. Pipes
6. WildCards and Regular expressions

4.5. Pipes

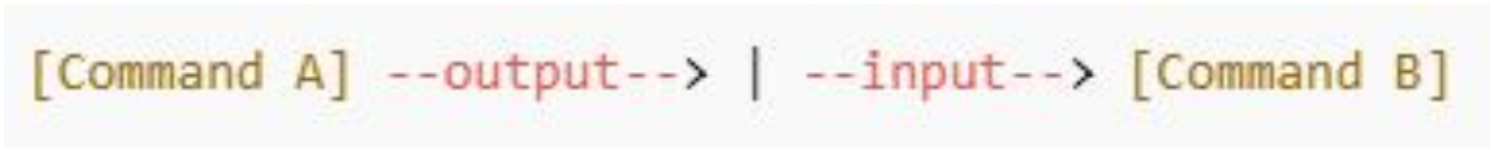
Connect stdout from a command to stdin of another command



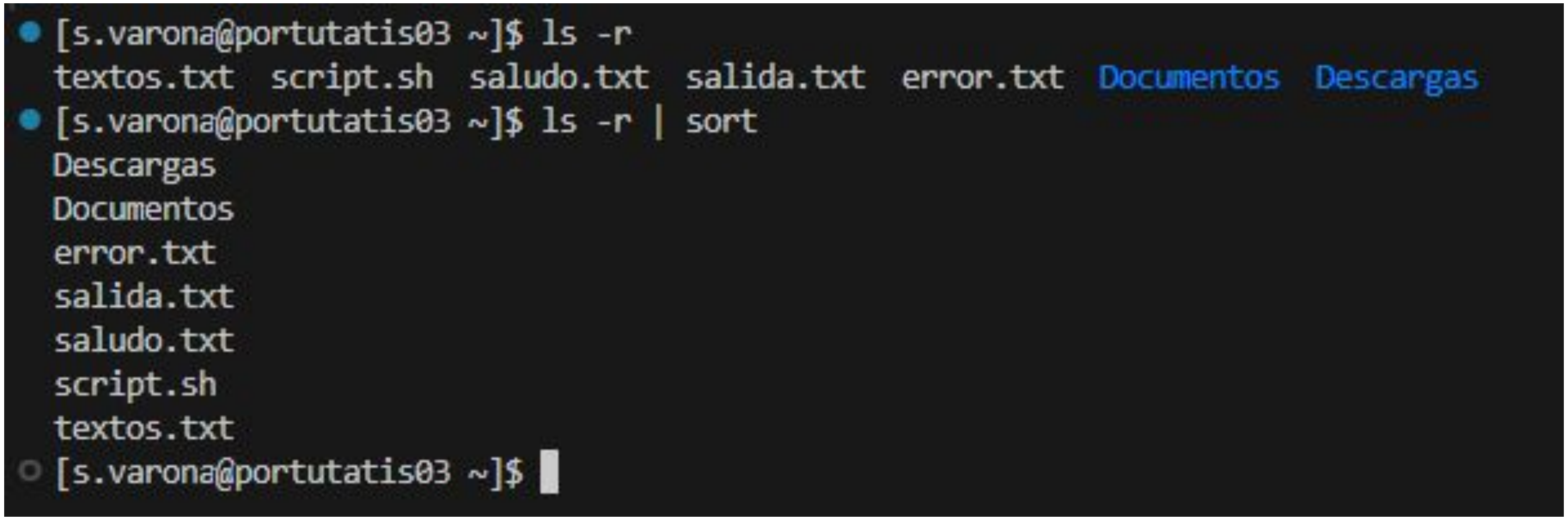
```
[Command A] --output--> | --input--> [Command B]
```

4.5. Pipes

Connect stdout from a command to stdin of another command



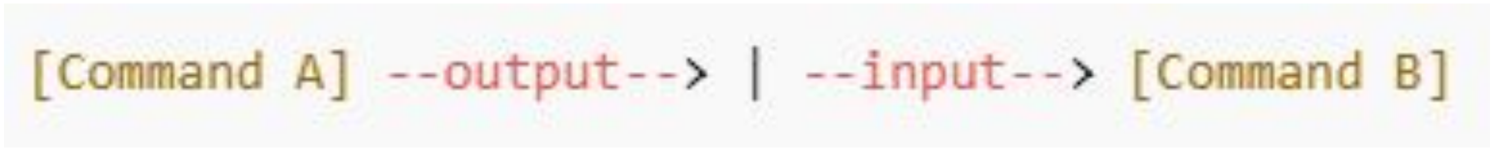
```
[Command A] --output--> | --input--> [Command B]
```



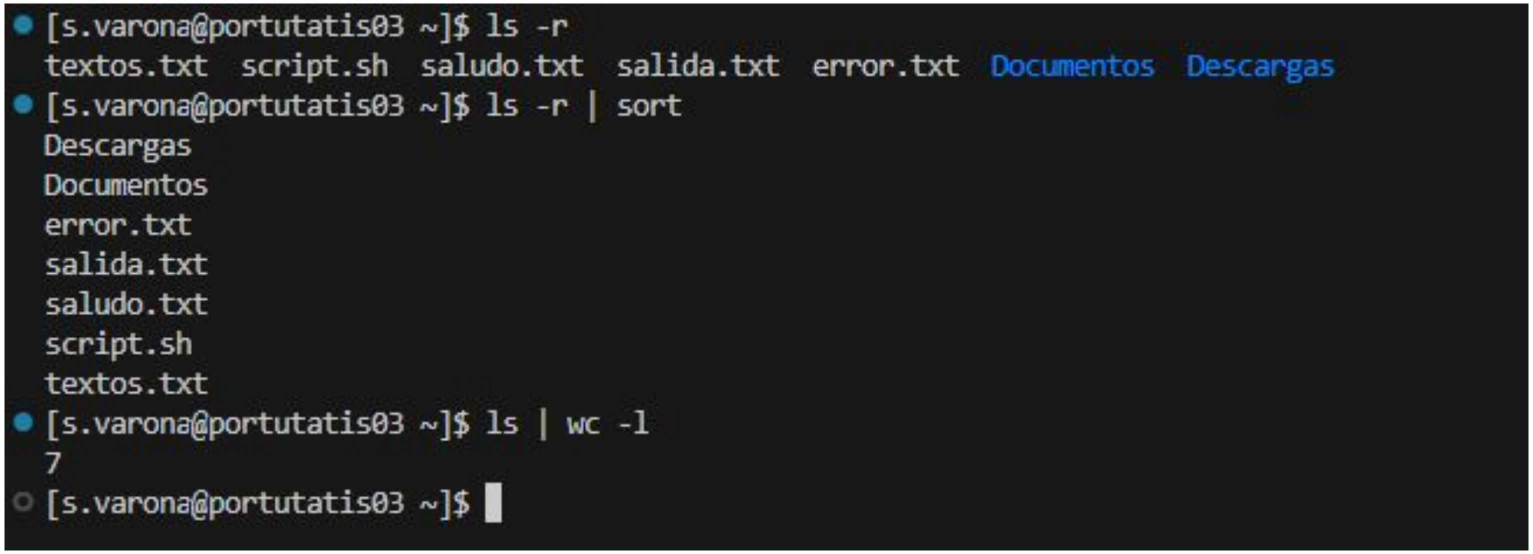
```
[s.varona@portutatis03 ~]$ ls -r
textos.txt  script.sh  saludo.txt  salida.txt  error.txt  Documentos  Descargas
[s.varona@portutatis03 ~]$ ls -r | sort
Descargas
Documentos
error.txt
salida.txt
saludo.txt
script.sh
textos.txt
[s.varona@portutatis03 ~]$
```

4.5. Pipes

Connect stdout from a command to stdin of another command

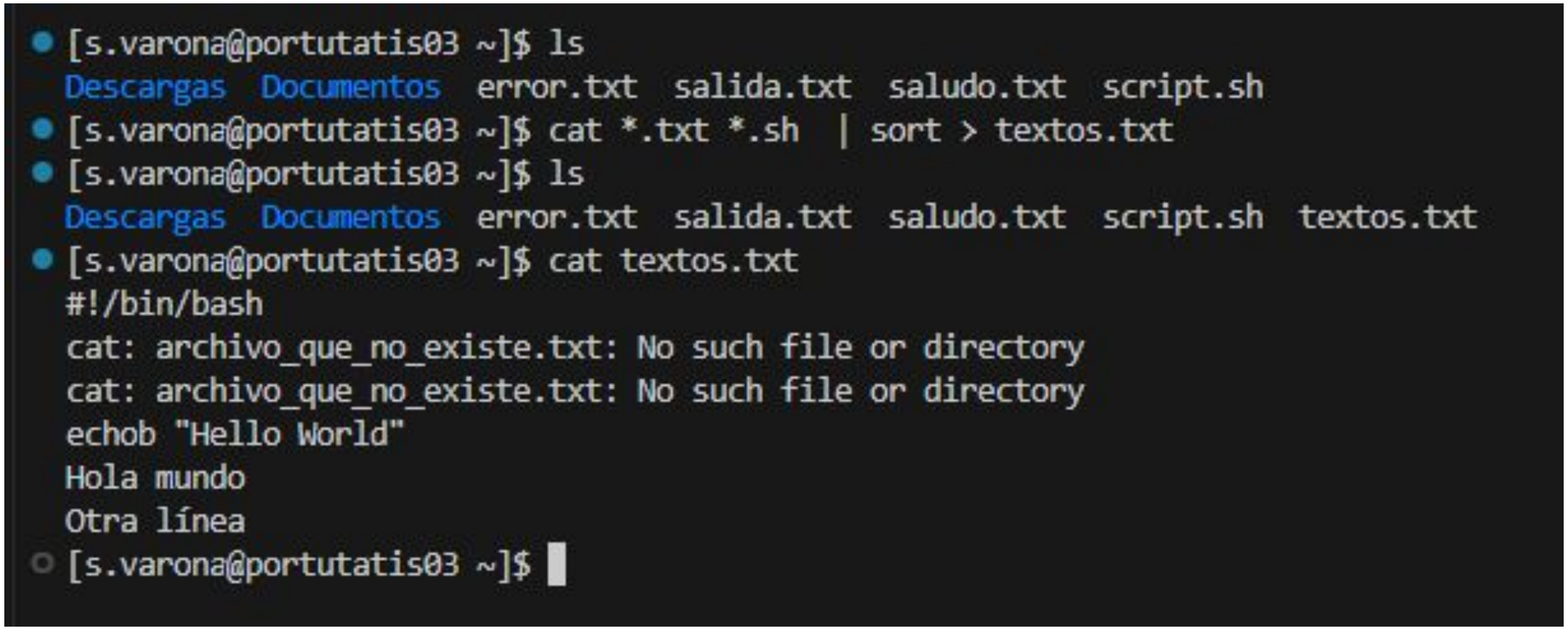


```
[Command A] --output--> | --input--> [Command B]
```



```
[s.varona@portutatis03 ~]$ ls -r
textos.txt  script.sh  saludo.txt  salida.txt  error.txt  Documentos  Descargas
[s.varona@portutatis03 ~]$ ls -r | sort
Descargas
Documentos
error.txt
salida.txt
saludo.txt
script.sh
textos.txt
[s.varona@portutatis03 ~]$ ls | wc -l
7
[s.varona@portutatis03 ~]$
```

4.5. Pipes



```
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  error.txt  salida.txt  saludo.txt  script.sh
[s.varona@portutatis03 ~]$ cat *.txt *.sh  | sort > textos.txt
[s.varona@portutatis03 ~]$ ls
Descargas  Documentos  error.txt  salida.txt  saludo.txt  script.sh  textos.txt
[s.varona@portutatis03 ~]$ cat textos.txt
#!/bin/bash
cat: archivo_que_no_existe.txt: No such file or directory
cat: archivo_que_no_existe.txt: No such file or directory
echob "Hello World"
Hola mundo
Otra línea
[s.varona@portutatis03 ~]$
```

4. Command Line Syntax

\
s

4.6. WildCards and Regular expressions

Patterns used to search, match, or validate text in strings

Pattern	Meaning	Use example	What finds
*	Zero or more repetitions	file*.txt	fileee.txt file1.txt files.txt
^	Start of the string	^file	file.pdf file.txt files.docx
$	End of the string	.txt$	file.txt document.txt
\t	Tabular	Name\tAge John\t25	Name Age John 25
\s	White space	John\sSmith	John Smith

Overview

1. Linux OS
2. Linux file system
3. Basic commands
4. Command line syntax
5. Linux users and privileges

5. Linux users and privileges

1. Users
2. Permissions
3. Change permissions

5.1. Linux Users

- Access entity to system resources
- Each user has:
  - A username
  - A user identifier (UID)
  - A primary group (GID)
  - A home directory (for example, /home/user)
  - A command-line interpreter (shell, such as /bin/bash)
- Type of entities:
  - Normal user:
    - Limited access
    - Only modify owner files
  - Root user:
    - Sysadmin
    - Total access
    - Un/Install software
  - Groups:
    - Group of users to manage collective permissions.

5.1. Linux Users

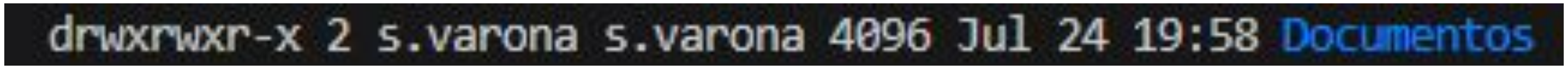
- Users can be linked to a person or computer process
- Every user may belong to one ore more groups
- Every user may has a home folder inside /home
- Users own the files they create, directly or indirectly
- Users can change permissions on files they own
- Users also own processes they execute
- Root rules over them all
- Root home folder is in /root

5. Linux users and privileges

1. Users
2. Permissions
3. Change permissions

5.2. Linux Permissions

To view file permissions and ownership on files and directories, use the **ls -l** command. For example:



```
drwxrwxr-x 2 s.varona s.varona 4096 Jul 24 19:58 Documentos
```

`drwxrwxr-x` are the permissions
`2` is the number of files or directories
`user` is the owner
`user` is the group
`4096` is the size in bytes
`Jan  9 10:11` is the date/time of last access
`documents` is the directory

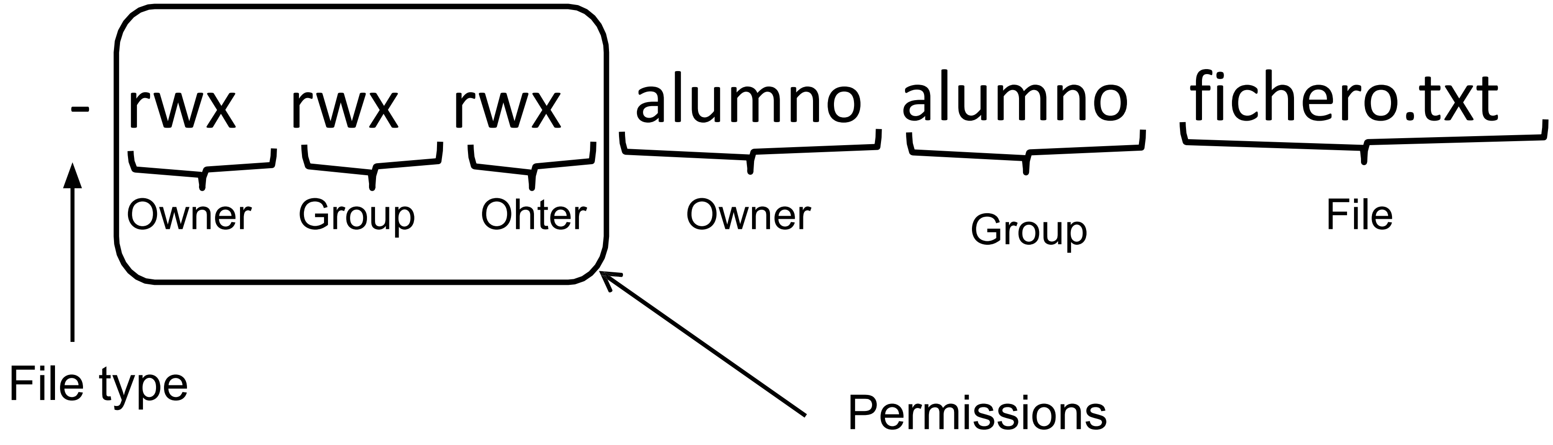
5.2. Linux Permissions

Permissions are the “rights” to act on a file or directory. There are only 3 basic permissions:

- **Read (r)** - allows the contents of the file to be viewed. A read permission on a directory allows you to list the contents of a directory.
- **Write (w)** - allows you to modify the contents of that file. For a directory, the write permission allows you to edit the contents of a directory.
- **Execute (x)** - for a file, the executable permission allows you to run the file and execute a program or script. For a directory, the execute permission allows you to enter the directory and make it your current working directory (pwd or “.”).

5.2. Linux Permissions

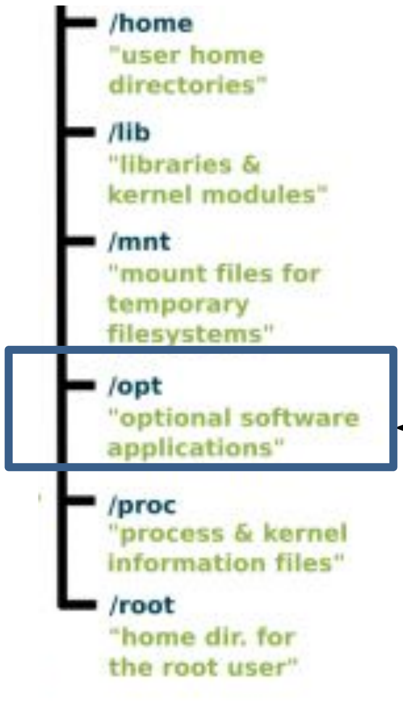
- A file's rights can only be modified by the owner of the file, the group owning the file and the root
- The system stores this right information in a 9 bits sequence.
- This sequence has a sequence of 3 elements for each 3 groups:



5.2. Linux Permissions

- Examples:
  - Example 1: /opt directory

5.2. Linux Permissions


/home
"user home directories"
/lib
"libraries & kernel modules"
/mnt
"mount files for temporary filesystems"
/opt
"optional software applications"
/proc
"process & kernel information files"
/root
"home dir. for the root user"

Directorio que contiene software adicional. En nuestro caso aquí se realiza la instalación de software bioinformático

5.2. Linux Permissions

- Examples:
  - Example 1: /opt directory:

```
drwxr-xr-x    root    root    opt
```

    - Owner (root) can read, write and execute
    - Group (root) and rest can only read and execute

5.2. Linux Permissions

- Examples:
  - Example 2: personal directory :   /home/alumno

```
drwx------   alumno   clase   alumno
```

- Owner (alumno) can read, modify and access the directory
- Group (clase) and rest can’t do anything

5.2. Linux Permissions

- Examples:
  - Example 3: /tmp :

```
drwxrwxrwx    root    root    tmp
```

  - ???

5.2. Linux Permissions

- Examples:
  - Example 3: tmp :

```
drwxrwxrwx   root   root   tmp
```

    - Everybody has permissions for everything

5.2. Linux Permissions

To view file permissions and ownership on files and directories, use the **ls -al** command. For example:

```
    drwxr-xr-x 2 user user 4096 Jan  9 10:11 documents
```

`drwxr-xr-x` are the permissions
`2` is the number of files or directories
`user` is the owner
`user` is the group
`4096` is the size in bytes
`Jan  9 10:11` is the date/time of last access
`documents` is the directory

5.2. Linux Permissions

Following previous example:

```
drwxrw-r-- 2 user user 4096 Jan  9 10:11 documents
```

?????

5.2. Linux Permissions

Following previous example:

```
    drwxrw-r-- 2 user user 4096 Jan  9 10:11 documents
```

Permissions are listed in the first 10 characters-dash section. The section can be read as follows:

`d` is a directory (`-` for files)
`rwx` the user has read, write, and execute permissions
`rw-` the group has read and write permissions
`r--` all others have read only permissions

5. Linux users and privileges

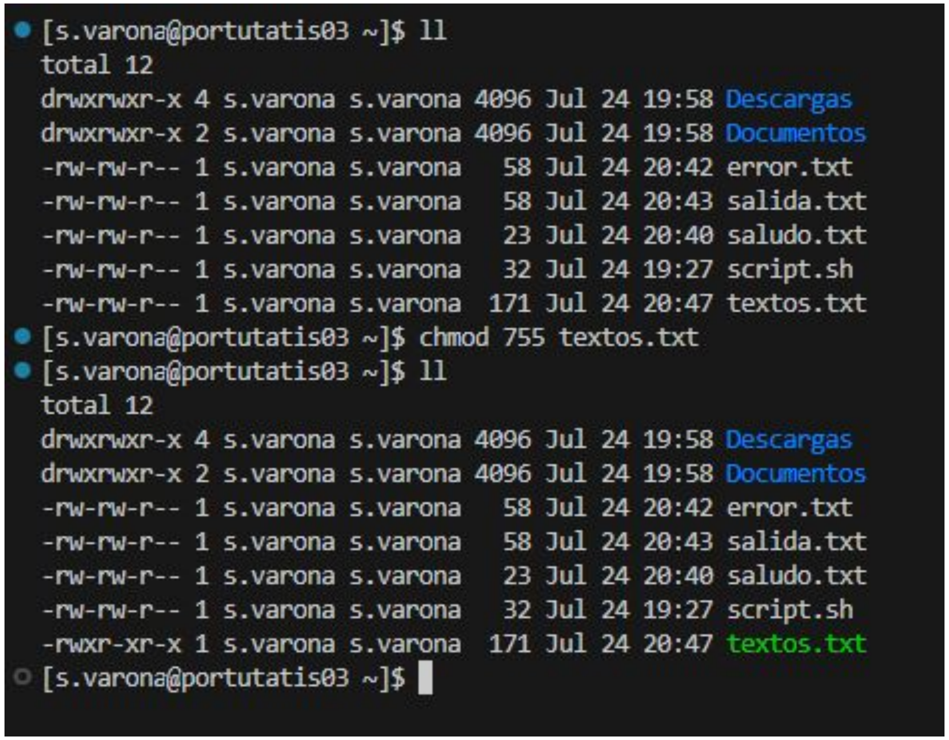
1. Users
2. Permissions
3. Change permissions

5.3. Modify Permissions

You can only change permission on files you own, while root can change permissions on any file of the system.

To change permissions, use the commands:

- `chmod` – change permissions



```
[s.varona@portutatis03 ~]$ ll
total 12
drwxrwxr-x 4 s.varona s.varona 4096 Jul 24 19:58 Descargas
drwxrwxr-x 2 s.varona s.varona 4096 Jul 24 19:58 Documentos
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:42 error.txt
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:43 salida.txt
-rw-rw-r-- 1 s.varona s.varona   23 Jul 24 20:40 saludo.txt
-rw-rw-r-- 1 s.varona s.varona   32 Jul 24 19:27 script.sh
-rw-rw-r-- 1 s.varona s.varona  171 Jul 24 20:47 textos.txt
[s.varona@portutatis03 ~]$ chmod 755 textos.txt
[s.varona@portutatis03 ~]$ ll
total 12
drwxrwxr-x 4 s.varona s.varona 4096 Jul 24 19:58 Descargas
drwxrwxr-x 2 s.varona s.varona 4096 Jul 24 19:58 Documentos
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:42 error.txt
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:43 salida.txt
-rw-rw-r-- 1 s.varona s.varona   23 Jul 24 20:40 saludo.txt
-rw-rw-r-- 1 s.varona s.varona   32 Jul 24 19:27 script.sh
-rwxr-xr-x 1 s.varona s.varona  171 Jul 24 20:47 textos.txt
[s.varona@portutatis03 ~]$
```

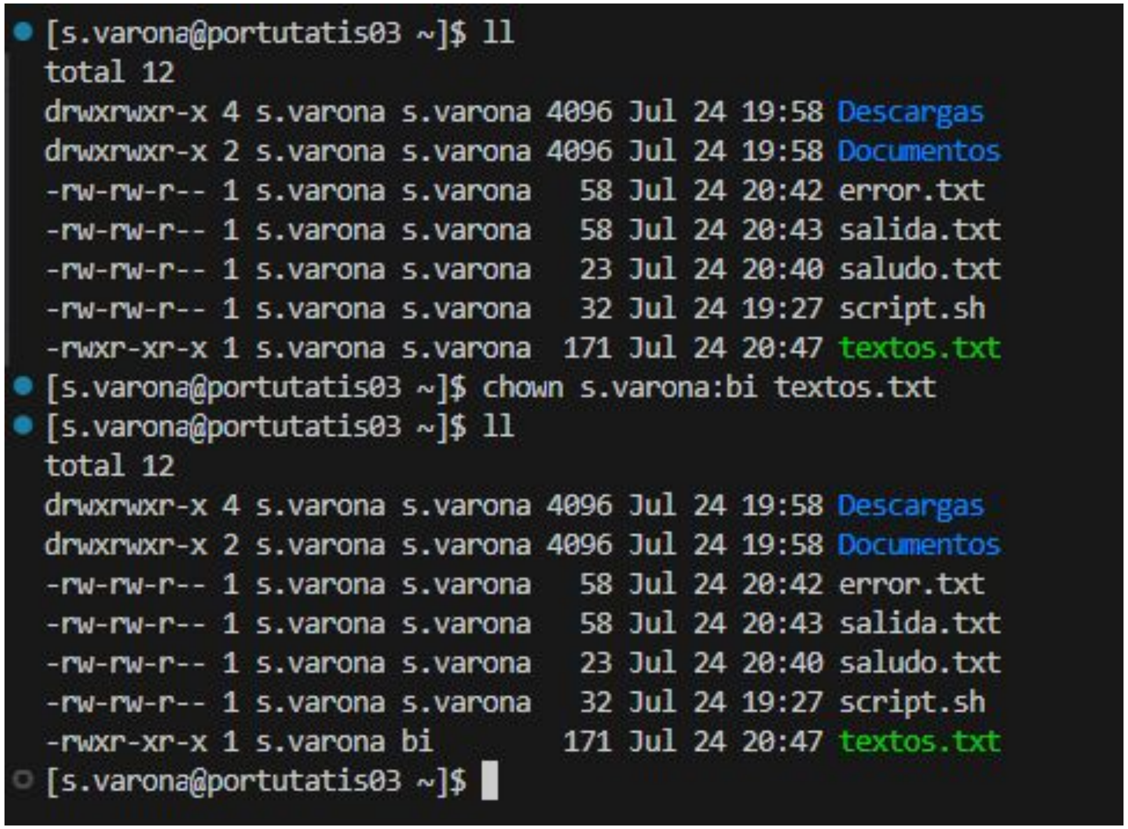
`rwx` to owner (7 = 4 + 2 + 1)

`rx` to group (5 = 4 + 0 + 1)

`rx` to others (5 = 4 + 0 + 1)

5.3. Modify Permissions

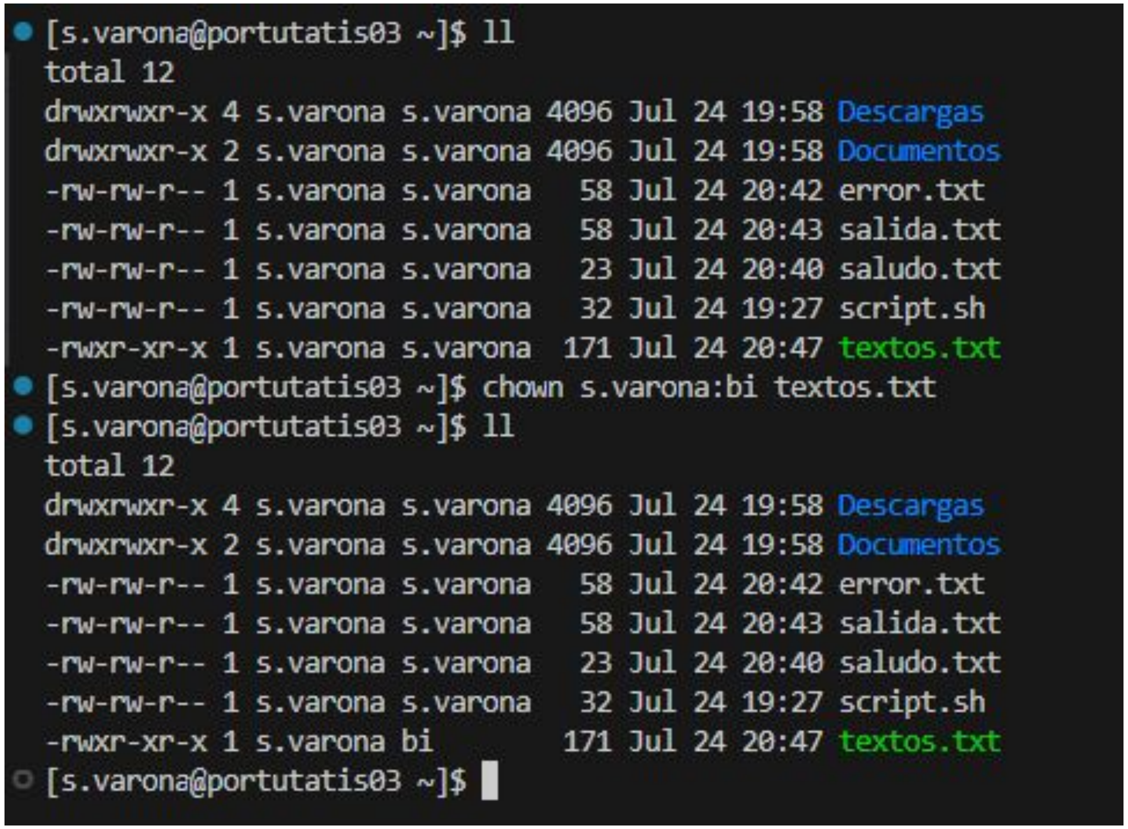
- `chown` - change owner



```
[s.varona@portutatis03 ~]$ ll
total 12
drwxrwxr-x 4 s.varona s.varona 4096 Jul 24 19:58 Descargas
drwxrwxr-x 2 s.varona s.varona 4096 Jul 24 19:58 Documentos
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:42 error.txt
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:43 salida.txt
-rw-rw-r-- 1 s.varona s.varona   23 Jul 24 20:40 saludo.txt
-rw-rw-r-- 1 s.varona s.varona   32 Jul 24 19:27 script.sh
-rwxr-xr-x 1 s.varona s.varona  171 Jul 24 20:47 textos.txt
[s.varona@portutatis03 ~]$ chown s.varona:bi textos.txt
[s.varona@portutatis03 ~]$ ll
total 12
drwxrwxr-x 4 s.varona s.varona 4096 Jul 24 19:58 Descargas
drwxrwxr-x 2 s.varona s.varona 4096 Jul 24 19:58 Documentos
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:42 error.txt
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:43 salida.txt
-rw-rw-r-- 1 s.varona s.varona   23 Jul 24 20:40 saludo.txt
-rw-rw-r-- 1 s.varona s.varona   32 Jul 24 19:27 script.sh
-rwxr-xr-x 1 s.varona bi        171 Jul 24 20:47 textos.txt
[s.varona@portutatis03 ~]$
```

5.3. Modify Permissions

- `chown` - change owner



```
[s.varona@portutatis03 ~]$ ll
total 12
drwxrwxr-x 4 s.varona s.varona 4096 Jul 24 19:58 Descargas
drwxrwxr-x 2 s.varona s.varona 4096 Jul 24 19:58 Documentos
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:42 error.txt
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:43 salida.txt
-rw-rw-r-- 1 s.varona s.varona   23 Jul 24 20:40 saludo.txt
-rw-rw-r-- 1 s.varona s.varona   32 Jul 24 19:27 script.sh
-rwxr-xr-x 1 s.varona s.varona  171 Jul 24 20:47 textos.txt
[s.varona@portutatis03 ~]$ chown s.varona:bi textos.txt
[s.varona@portutatis03 ~]$ ll
total 12
drwxrwxr-x 4 s.varona s.varona 4096 Jul 24 19:58 Descargas
drwxrwxr-x 2 s.varona s.varona 4096 Jul 24 19:58 Documentos
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:42 error.txt
-rw-rw-r-- 1 s.varona s.varona   58 Jul 24 20:43 salida.txt
-rw-rw-r-- 1 s.varona s.varona   23 Jul 24 20:40 saludo.txt
-rw-rw-r-- 1 s.varona s.varona   32 Jul 24 19:27 script.sh
-rwxr-xr-x 1 s.varona bi        171 Jul 24 20:47 textos.txt
[s.varona@portutatis03 ~]$
```

5.3. Modify Permissions

Permissions are usually managed in octal format

Every 3 characters group belonging to a set of permissions translates to a number ranging from 0 (---) to 7 (rwx), where

- r=4
- w=2
- x=1

Example:

```
rwx = 4 + 2 + 1 = 7
rw- = 4 + 2 + 0 = 6
r-- = 4 + 0 + 0 = 4
```

Thanks for your attention!