ALX LESSON Ubuntu Shell Commands P2

Shell - Bash

TABLE OF CONTENTS

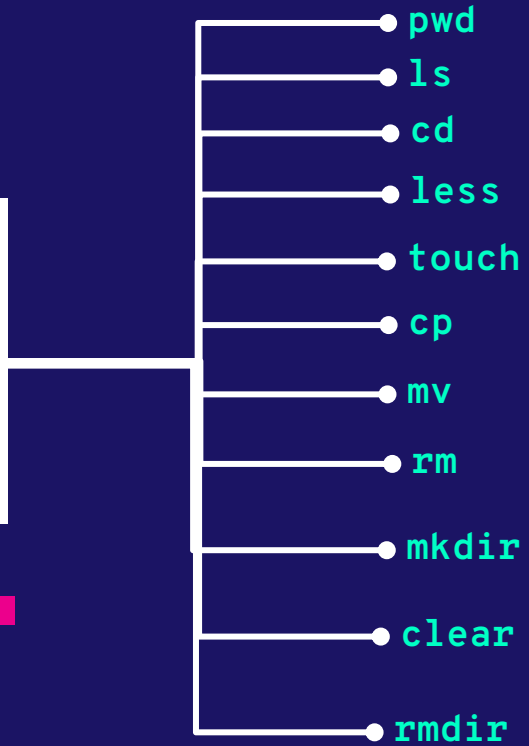
01

OVERVIEW topics

OVERVIEW Topics

Ubuntu Shell Commands

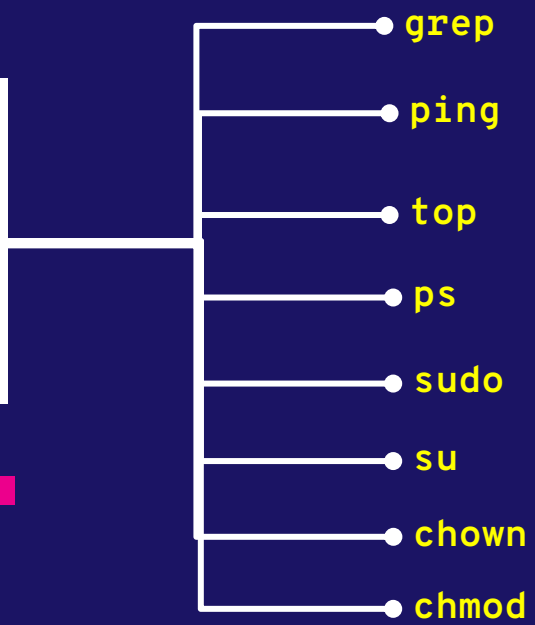
Topics



OVERVIEW Topics

Ubuntu Shell Commands

Topics



Slides On Telegram

https://t.me/alx_2023

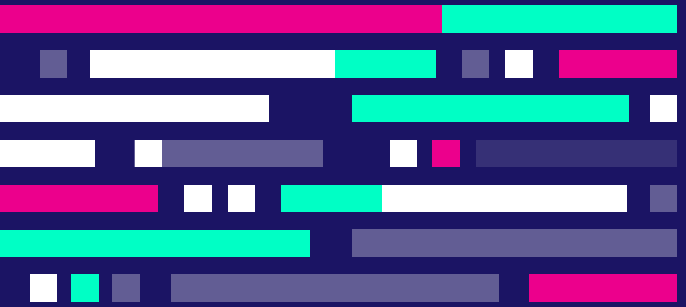
Ubuntu Shell Commands

Topics

02

Learning Objectives

pwd



This command stands for "print working directory."

When executed, it displays the current directory (folder) you are in within the terminal.

pwd

```
current_dir=$(pwd)
echo "Current directory: $current_dir"

cp some_file.txt $(pwd)/../some_other_directory/
```

ls

This command stands for "list." When executed, it displays a list of files and directories in the current directory (folder) you are in within the terminal.

`ls -a`

This will list all files and directories in the current directory, including hidden files.

`ls -l example.txt`

display additional information about each file, such as permissions, ownership, size, and modification date

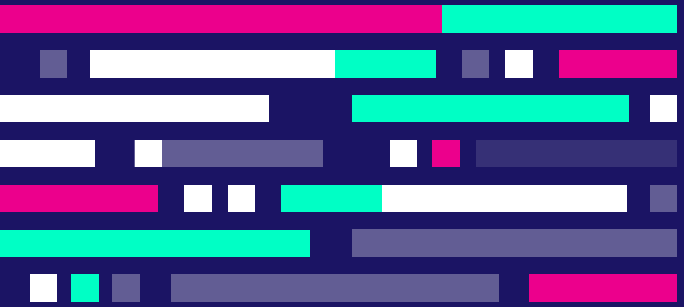
`ls -lt`

sort files by modification time

`ls -ls`

sort files by modification time (in reverse order) or by size

cd



This command stands for "change directory." When executed, it allows you to navigate to a different directory (folder) within the terminal.

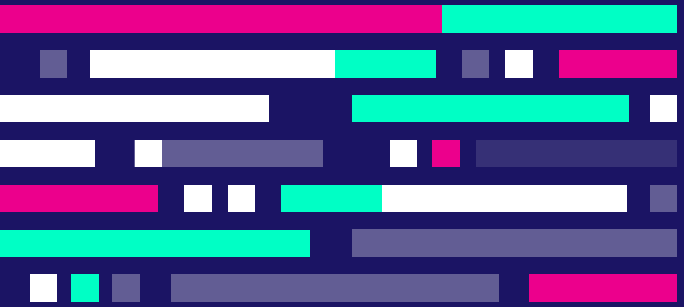
`cd ~`

This command takes you to your home directory. Your home directory is a default directory created for each user on the system, and it's usually located at /home/yourusername/.

`cd /`

This command takes you to the root directory of the file system. The root directory is the top-level directory in the file system hierarchy and contains all other directories and files on the system.

less



This command is a program used to view text files within the terminal. It displays the contents of a file one screenful at a time and allows you to scroll through the file.

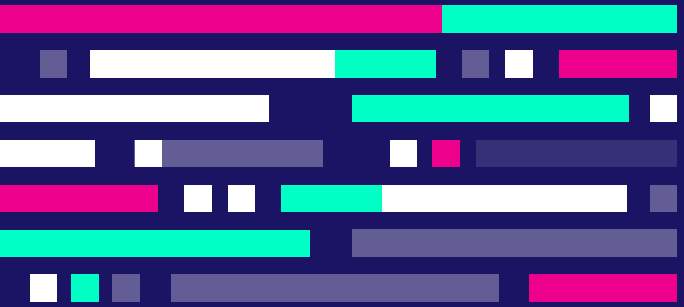
less /path/to/file

/search_term
This will search for the text "search_term" in the file.

less file.gz
it will show you the contents of the compressed file without extracting it, another formats supported ".bz2" ".xz"

you can use tar to extract files from a .tar.gz archive, or unzip to extract files from a .zip archive.

touch



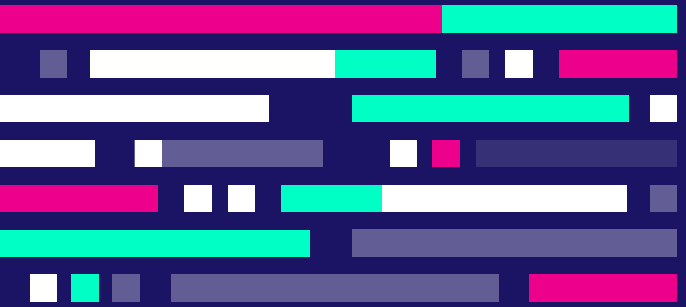
This command creates a new file in the current directory or updates the modification time of an existing file.

```
touch myfile.txt
```

```
touch file1.txt file2.txt file3.txt
```

This will create three new empty files called "file1.txt", "file2.txt", and "file3.txt" in the current directory.

cp



This command stands for "copy." When executed, it copies one or more files from one location to another.

```
cp file.txt /path/to/new/location/
```

```
cp file1.txt file2.txt file3.txt /path/to/new/location/
```

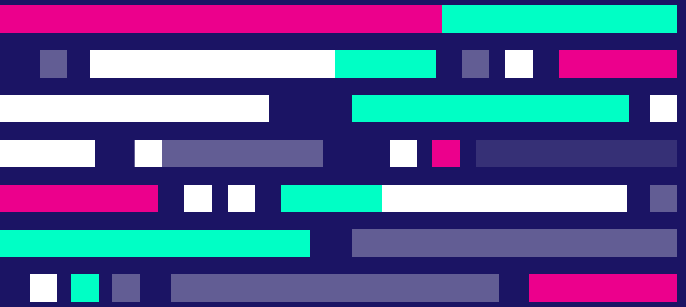
```
cp -r /path/to/source/ /path/to/destination/
```

This will copy the "source" directory and all of its contents to the "destination" directory recursively.

```
cp file.txt newfile.txt
```

This will create a copy of "file.txt" with the new name "newfile.txt" in the same directory as the original file.

mv

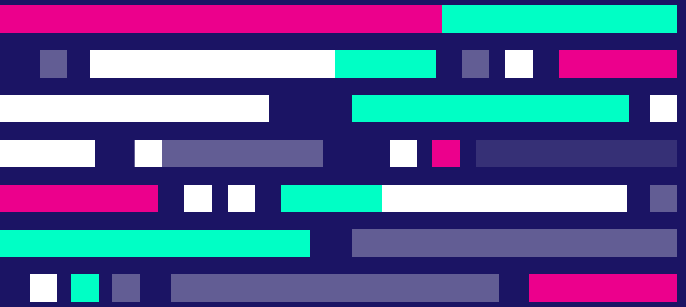


This command stands for "move." When executed, it moves one or more files from one location to another. It can also be used to rename files.

```
//Moving files
mv ~/example.txt ~/documents/

mv file1.txt file2.txt file3.txt /path/to/new/location/

//Renaming files
mv example.txt exam.txt

mv /path/to/source/ /path/to/destination/
```

This will move the "source" directory and all of its contents to the "destination" directory.

rm

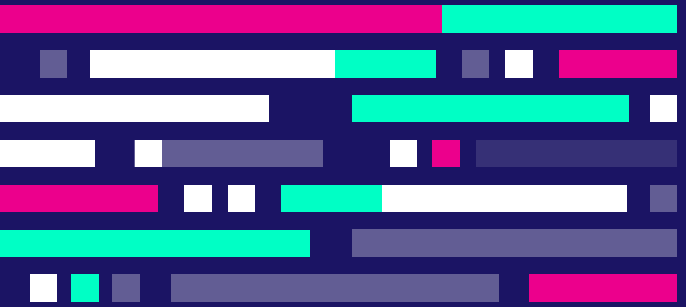


This command stands for "remove." When executed, it deletes one or more files or directories from the system. Be careful when using this command, as it permanently deletes files and directories.

```
rm ~/example.txt
rm ~/*.txt
rm -rf  ~/directory
```

This will forcefully remove the "directory" directory and all of its contents without prompting for confirmation.

mkdir



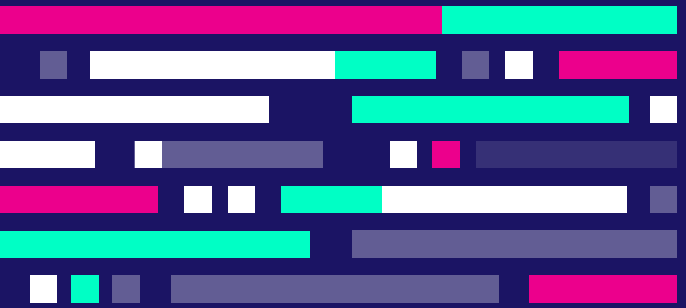
This command stands for "make directory." When executed, it creates a new directory (folder) in the current directory (folder) you are in within the terminal.

```
mkdir ~/documents
```

```
mkdir -p my_parent_directory/my_directory
```

This will create the directory my_directory within the directory my_parent_directory. If my_parent_directory does not exist, it will be created along with my_directory.

rmdir

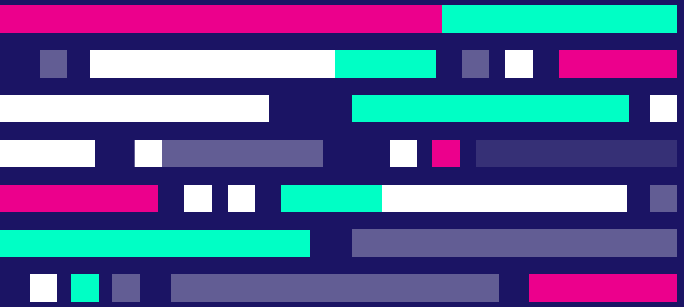


This command stands for "remove directory." When executed, it deletes an empty directory (folder) from the system. If the directory contains files or subdirectories, you must first remove those before using rmdir.

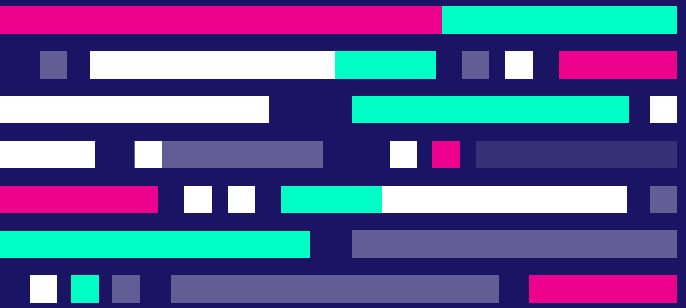
```
rmdir ~/documents
```

clear

This command empty the console



grep



This command searches for a specific pattern within a file or files. It can be used to search for specific text or to filter results.

```
grep "example" file.txt
```

```
grep "example" *.txt
grep -r "example" docs/*.txt
```

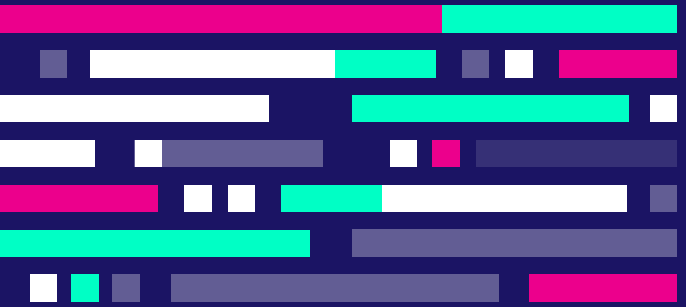
```
grep -i "example" file.txt
```

This would search for the word "example" in the file.txt file, ignoring the case of the pattern.

```
grep "[0-9]" file.txt
```

You can use regular expressions with the grep command to search for more complex patterns. For example, if you want to search for all lines in a file that contain a number

ping



This command is used to test the connectivity between your computer and another computer or server on the network.

ping google.com

ping -c 5 google.com

This would send 5 packets to the google.com host

ping -s 1000 google.com

This would send packets with a size of 1000 bytes to the google.com

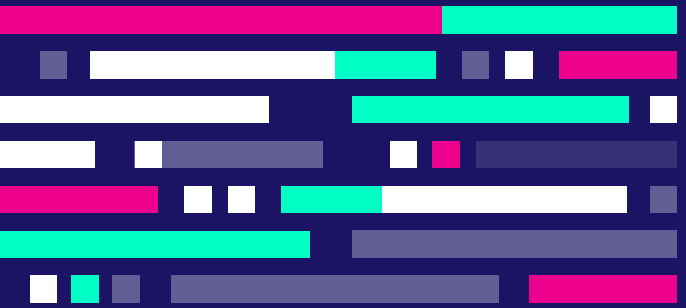
ping -i 2 google.com

This would send packets to the google.com host with an interval of 2 seconds between them

ping -6 2001:4860:4860::8888

This would send packets to the IPv6 address

top



This command is a program that displays a real-time view of the processes running on the system, including CPU usage, memory usage, and more.

if you want to sort processes by memory usage, you can press the M key. Similarly, if you want to sort processes by process ID, you can press the P key.

top -p 1234

This will display information about the process with ID 1234.

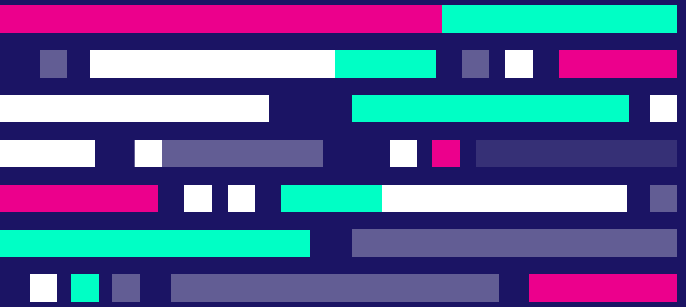
top -n 5

This will display the top 5 processes by CPU usage.

top -d 5

This will update the display every 5 seconds.

ps



This command stands for "process status." It displays information about the currently running processes on the system.

`ps` command without any options, it will display a list of running processes for the current terminal session

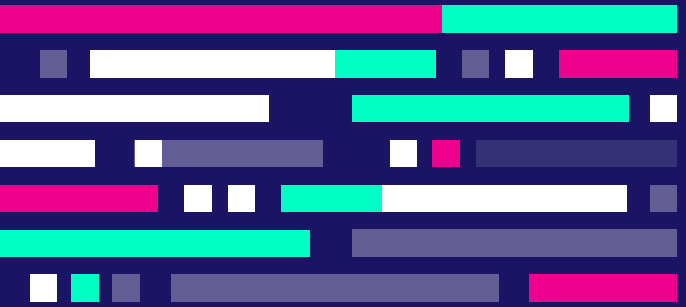
`ps -e` display processes for all users

`ps -f` display full information about each process.

`ps -H` Displaying processes in a tree format

`ps -e --sort=-pcpu --format='pid,comm,%cpu,%mem'`
This will display all processes, sorted in descending order by CPU usage, and the process ID, command name, and percentage of CPU and memory used.

sudo

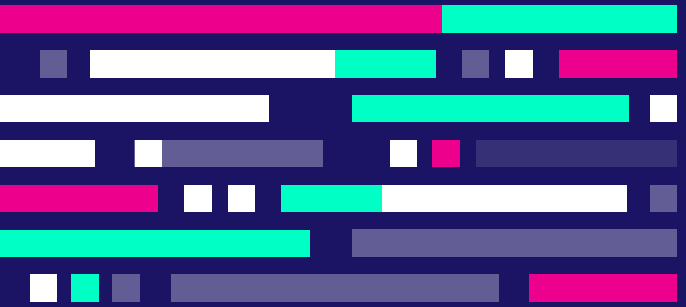


It allow users to execute commands with elevated privileges or permissions. It stands for "superuser do" and allows a regular user to temporarily elevate their privileges to those of the system administrator or root user. This is necessary for carrying out certain system-level tasks that require special permissions.

If you want to install software on your Unix-based system, you will often need root-level access to do so. For example, you can install a package in Ubuntu using
`sudo apt-get install <package name>`

Updating your operating system requires elevated privileges. You can use the command on Ubuntu to update the system.
`sudo apt-get update && sudo apt-get upgrade`

sudo

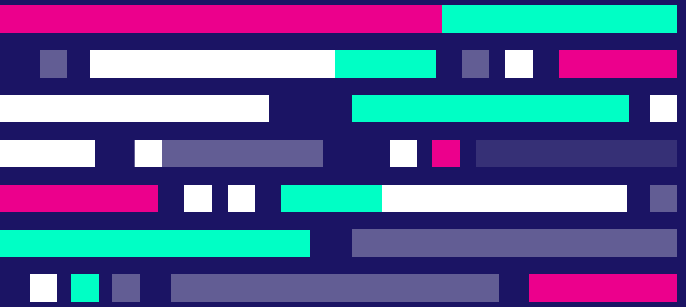


`sudo service <service name> start`
Start a service or stop it

`sudo vi /etc/fstab`
open and edit the system file.

When you run `sudo -i` and enter your password, you will be switched to the root user's environment and will have full root-level access to the system until you `exit` the shell by typing exit or pressing `Ctrl-D`.

su



The su command is used to switch to another user account or become a superuser.

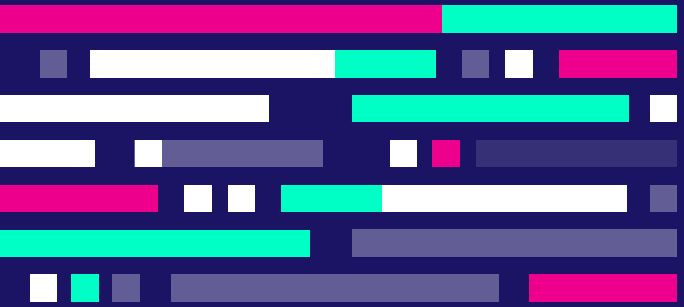
```
su <username>
```

Then it will ask you to enter the password to login.

Switch to root user:

```
su
```

chown



This command stands for "change owner." It allows you to change the ownership of files and directories to a different user or group.

chown jane example.txt

This will change the ownership of example.txt to the user jane.

chown -R jane:users example

You can use the -R option to change the ownership recursively, and the : character to change the group ownership. For example, to change the ownership of a directory named example and all its contents to the user jane and the group users

chown --reference=reference.txt example.txt

This will change the ownership of example.txt to match the ownership of reference.txt.

chown

chown 1000 example.txt
instead of using the username to change the ownership, you can use the numeric user ID (UID) or group ID (GID) to change the ownership. For example, to change the ownership of a file named example.txt to the user with UID 1000

how to get id of group(GID):
cat /etc/group
The first field is the group name, followed by a colon (:). The second field is the group password (which is usually empty), followed by another colon. The third field is the group ID (GID), followed by a colon. The fourth field is a comma-separated list of the usernames that belong to the group

how to get id of user(UID):
id -u jane
grep jane /etc/passwd

chmod

Symbol	Description
r	Read
w	Write
x	Execute

Symbol	Description
+	Add
-	Remove
=	Set

Symbol	Category	Description
`u`	Owner	The user who owns the file
`g`	Group	A group of users who have been granted access to the file
`o`	Others	All other users who are not the owner or part of the group
`a`	All	All users, including the owner, group, and others

chmod

Octal Notation	Symbolic Notation	Permission
0	---	No permissions
1	--x	Execute only
2	-w-	Write only
3	-wx	Write and execute
4	r--	Read only
5	r-x	Read and execute
6	rw-	Read and write
7	rwx	Read, write, and execute

chmod

chmod 644 example.txt

This sets the file example.txt to have read and write permissions for the owner of the file, and read-only permissions for group and other users.

6 4 4

(owner "u", group "g", others "o")

chmod g+w,o-rx example_dir

This adds write permission for the group owner of the directory example_dir, and removes the read and execute permissions for other users.

chmod -R u=rwX,g=rX,o=rX example_dir

the capital X option means that directories and files that already have execute permission will retain it, while those that don't have execute permission will not be granted it. This is a useful way to ensure that only directories (and certain files) that are meant to be executed can be executed, without changing the execute permissions of all files in the directory.

03

Quiz questions

04

Hands on lab Practice


Have a Question
Leave a Comment!

Share

To let the others know more

Subscribe

To stay updated with latest videos

Thanks