

# Get to Linux

## Session 2

*Your ultimate guide to mastering linux!*

# Session Agenda

01

Basic commands:  
zip, tar, find, nano

04

Packages

02

Shell environment

05

Mounting

03

Intro to Bash Script

06

Open source programs

07

How to search about your  
problems

01



# Basic Commands

zip/unzip

tar

find

nano



# Zip

A tool for both compression and archiving.

Syntax:

`zip [options] zipname.zip files`

- To compress everything inside a folder:

```
sondoskhali@pop-os:~/Documents$ zip -r test.zip test
adding: test/ (stored 0%)
adding: test/testfile.txt (stored 0%)
```

- To compress multiple files into a zip folder:

```
sondoskhali@pop-os:~/Documents$ zip fruits.zip banana.txt apple.txt
adding: banana.txt (stored 0%)
adding: apple.txt (stored 0%)
sondoskhali@pop-os:~/Documents$ ls
apple.txt  banana.txt  fruits.zip
```

01

# unzip

For decompressing a zip file.

Syntax:

`unzip file.zip`

- To decompress a file:

```
sondoskhali@pop-os:~/Documents$ unzip fruits.zip
Archive: fruits.zip
extracting: banana.txt
extracting: apple.txt
```

# Tar

A tool for archiving.

Syntax:

`tar -options -f archivename.tar FilesToBeArchived`

- To archive files:

```
zaytoon@ZaytoonsComputer:~/session2tartest$ tar -cv -f archivedfile.tar file1.txt file2.txt file3.txt
file1.txt
file2.txt
file3.txt
zaytoon@ZaytoonsComputer:~/session2tartest$ ls
archivedfile.tar  file1.txt  file2.txt  file3.txt
zaytoon@ZaytoonsComputer:~/session2tartest$ tar -tv -f archivedfile.tar
-rw-rw-r-- zaytoon/zaytoon 0 2026-02-01 14:08 file1.txt
-rw-rw-r-- zaytoon/zaytoon 0 2026-02-01 14:09 file2.txt
-rw-rw-r-- zaytoon/zaytoon 0 2026-02-01 14:09 file3.txt
```

Note: -f must be followed by the archived file name.

## tar - Important options:

Option	What It Does
<b>-c</b>	Create
<b>-x</b>	Extract
<b>-r</b>	Append
<b>-t</b>	List contents
<b>-z</b>	Compress (gzip)
<b>-j</b>	Compress (bzip2)
<b>-v</b>	Verbose (مطّول) (Optional)

01

# find

to search for files & directories based on name, type, size, or other conditions

Syntax:

`find [path] [options] [expression]`

- To find a file in a specific directory with -name option:

```
sondoskhalid@pop-os:~$ find ./Documents -name apple.txt  
./Documents/apple.txt
```

- to find all .txt files in current directory:

```
sondoskhalid@pop-os:~/Documents$ find *.txt  
apple.txt  
banana.txt
```

# find - Important options:

Option	What It Does	Example
<b>-iname</b>	Case-insensitive name search	<code>find ~ -iname "notes.*"</code>
<b>-type f</b>	Finds only files	<code>find /var/log -type f</code>
<b>-type d</b>	Finds only directories	<code>find ~ -type d</code>
<b>-perm -u+x</b>	Finds executable files	<code>find . -perm -u+x</code>
<b>-empty</b>	Finds and lists all empty files and directories	<code>find . -empty</code>

01

# nano

simple, terminal-based text editor in Linux.

Syntax:

`nano filename.txt`

Nano file handling:

Action

Save current

Save as / offer to write

Exit Nano

Undo

Shortcut

`Ctrl + S`

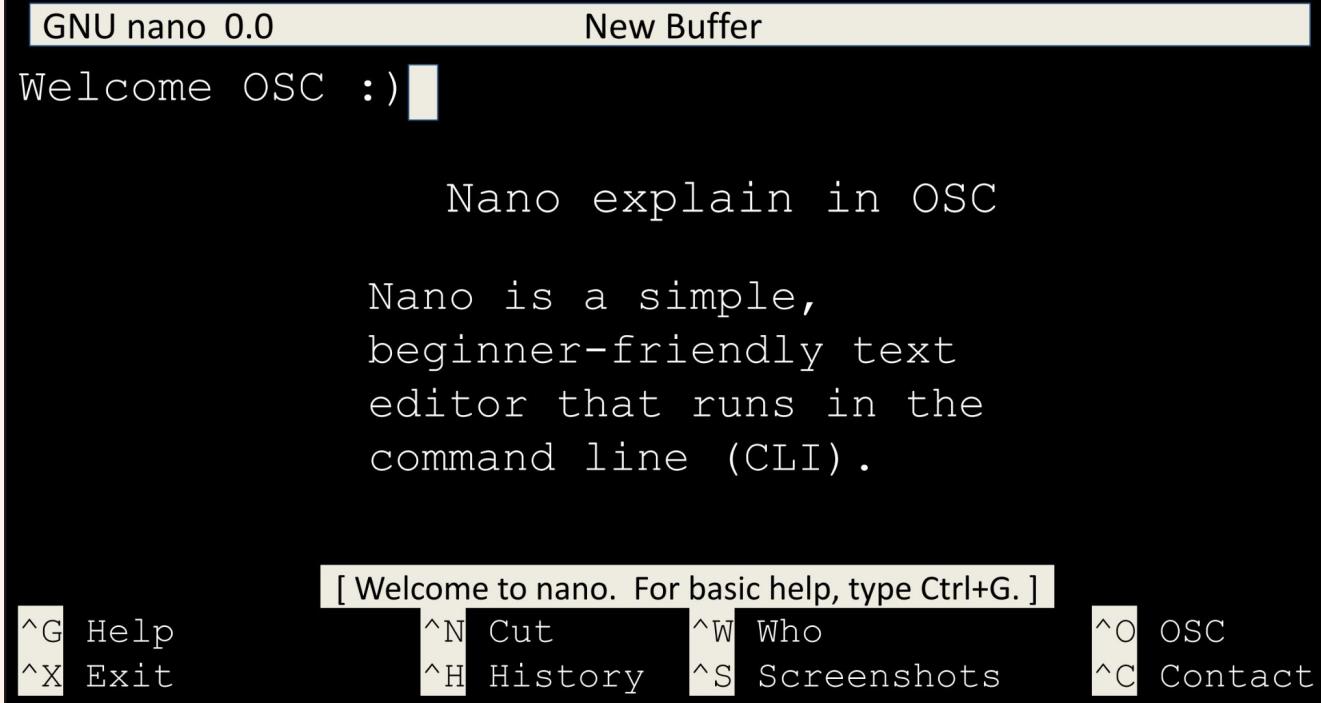
`Ctrl + O filename`

`Ctrl + X`

`Alt + U`

01

# nano



# HANDS ON (1)

- 1)Search for a specific set of files in your working directory.
- 2)Compress the found files into a single zip archive.
- 3)Create a new directory for extraction.
- 4)Extract the zip archive into the new directory.
- 5)Inside the extracted directory, create or open a text file using nano.
- 6)Write the following text inside the file: MISSION ACCOMPLISHED

02



# Shell environment



## Shell environment

A shell is a **command-line interface** program that allows users to interact with the operating system by typing commands.

So the shell environment is **The context in which the shell runs**.

# Shell

Write **sh** to switch to your shell:

```
sondoskhali@pop-os:~$ sh
$ echo "Hello from sh"
Hello from sh
$ bash
sondoskhali@pop-os:~$ echo "back to defult bash shell"
back to defult bash shell
sondoskhali@pop-os:~$
```

# Shell Variables

A shell variable is used to **store some value**. It could be an integer , filename, string or some shell command itself.

## Purpose:

Are often used for **temporary storage** of data within a script or for managing settings specific to the current interactive shell session.

## Scope:

These variables are **local** and apply only to the current shell instance in which they are defined. They are not inherited by child processes.

# Shell Variables

- Declaring a shell var syntax `varname=value` & Printing a shell var syntax `$varname`:

```
sondoskhaliid@pop-os:~$ exampleVar="sondos"
sondoskhaliid@pop-os:~$ echo $exampleVar
sondos
```

03



# Introduction to Bash Script





# Introduction to Bash Script

The script is a **text file** containing a series of commands or even block of code that can be **executed by the shell**.

- used to automate repetitive tasks.
- Have extension .sh but its not mandatory.



# Introduction to Bash Script

**script file example :**

```
$ scriptV.sh ×  
home > sondoskhalid > Desktop > $ scriptV.sh  
1 #!/bin/bash  
2 echo "Hello this is my first script ;)"  
3
```



# How to execute scripts?



# How to execute scripts?

first way :

- make your script file executable:

`chmod +x scriptFile.txt`

- run your script by specifying its path:

`pathToFile/scriptFile.txt`

```
sondoskhalid@pop-os:~/Desktop$ chmod +x scriptV.sh
sondoskhalid@pop-os:~/Desktop$ ./scriptV.sh
Hello this is my first script ;)
```

# How to execute scripts?

**second way :**

- **by specifying the shell explicitly:**

**bash pathToFile/scriptFile.txt**

```
sondoskhalid@pop-os:~/Desktop$ ls -l scriptV.sh
-rwxrwxr-x 1 sondoskhalid sondoskhalid 53 Dec 21 08:03 scriptV.sh
sondoskhalid@pop-os:~/Desktop$ bash scriptV.sh
Hello this is my first script ;)
sondoskhalid@pop-os:~/Desktop$
```

## HANDS ON (2)

- 1)Create a new script file.
- 2)Write a command inside the script that prints the following text to the terminal: Hello World
- 3)Make the script executable.
- 4)Execute the script from the terminal.
- 5)Confirm that the output is displayed correctly

04



# Packages



## What is a Package Manager?

A package manager is a tool that helps you install, update, and remove software easily and it:

- **Fetches the right files for you**
- **Installs them in the right folders**
- **Knows what dependencies to keep and what to toss**

## package managers Examples:

PackageManager	Distribution /Type	Command
APT	Debian, Ubuntu	<code>sudo apt install firefox</code>
DNF	Fedora, RHEL	<code>sudo dnf install gimp</code>
Pacman	Arch Linux	<code>sudo pacman -S vim</code>
Snap	Universal	<code>sudo snap install spotify</code>
Flatpak	Universal	<code>flatpak install flathub org.gimp.GIMP</code>

## 04

# What if the package is not in a package manager?

You can use other methods:

## 1. **appImage**:

- just download and run:

```
chmod +x example.AppImage  
./example.AppImage
```

## What if the package is not in a package manager?

### 2. Install tar.gz archive:

- usually contains source code or precompiled binaries:

```
tar -xvzf package.tar.gz  
cd package_folder  
.configure  
make  
sudo make install
```

## What if the package is not in a package manager?

### 3. debian file (Debian/Ubuntu)

- Install with dpkg: `sudo dpkg -i package.deb`
- using apt: `sudo apt install ./package.deb`

others:

install instruction for specific program is included in its repo.

05



# Mounting



01

## What is mounting ?

Mounting is the process of connecting storage devices or file systems (like EXT4, NTFS, or FAT32) to directories known as **mount points**. Once mounted, everything inside that mount point reflects the contents of the attached storage.

Done using mount command.

# Mount command

Syntax:

```
sudo mount [options] <device> <mount_point>
```

Here:

- **device**: the partition you want to mount.
- **mount point**: the directory where the device will be mounted
- **options**: different mount options that modify the command behaviour.

Mounting is like plugging a USB into a computer and opening it to see the files.

# Mount command

- First **list block devices with lsblk**:  
to see all device names and their mount point before mounting anything.

```
sondoskhalid@pop-os:~$ lsblk
NAME      MAJ:MIN RM    SIZE RO TYPE MOUNTPOINTS
zram0     251:0   0    16G  0 disk [SWAP]
nvme0n1   259:0   0  476.9G  0 disk
└─nvme0n1p1 259:1   0   100M  0 part
└─nvme0n1p2 259:2   0    16M  0 part
└─nvme0n1p3 259:3   0  148.7G  0 part
└─nvme0n1p4 259:4   0   735M  0 part
└─nvme0n1p5 259:5   0  100.9G  0 part
└─nvme0n1p6 259:6   0  156.5G  0 part
└─nvme0n1p7 259:7   0    1.5G  0 part /boot/efi
└─nvme0n1p8 259:8   0   68.5G  0 part /
```

# Mount command

- Second **mount the device** :

```
ooper-os:~$ cd /mnt/usb  
ooper-os:/mnt/usb$ sudo mount -o ro /dev/nvme0n1p5 /mnt/usb
```

**-o ro** is a mount option that tells Linux to mount the  
Mounts the partition **/dev/sdb1** to **/mnt/usb** in **read-only**  
mode you can access files without modifying them,  
which is safe for NTFS partitions left in an unsafe state .

# Mount command

## Important mounting Notes:

- The mount point must **already exist**
- Mounting usually requires **sudo**
- Always **unmount the device safely before removing it using unmount command :**

```
@pop-os:/mnt/usb$ sudo umount /mnt/usb
```

# HANDS ON (3)

- 1)Create a directory to be used as a mount point.
- 2)Mount a file system to that directory.
- 3)Verify that the file system is mounted.
- 4)Unmount the file system safely.

06



# Open source programs



## Alternative Open source programs in Linux

Linux usually offers open-source alternatives such as :

**Microsoft Word**      -> **LibreOffice Writer**

**Microsoft Excel**      -> **LibreOffice Calc**

**VS Code**                  -> **VSCodium**

**Media Player**              -> **VLC**

07



## How to Search about your problem



# How to Search About Your Problem in Linux

When facing a problem in Linux, searching correctly saves time and effort.

1. **Search by Distribution:**

Linux distributions use different package managers, so always include your distro name when searching.

2. **Search in Terms of Packages:**

search for the package name instead of app name.

Also search in online resources such as [Ask Ubuntu](#).



Thanks!

