CYBER SECURITY UPSKILLING PROGRAM

قدم خلال مبادرة زنك/2 في جامعة البلقاء التطبيقية
بالتعاون مع أكاديمية سايبر شيلد

SEP 2024

Linux Part

Version 1

INST.:ENG.ALI BANI BAKAR-0778642376(CYBER SHIELD ACADEMY)

DONE BY: ENG. Dana Al-Mahrouk-0798697842-BAU.UNIV.

Outline

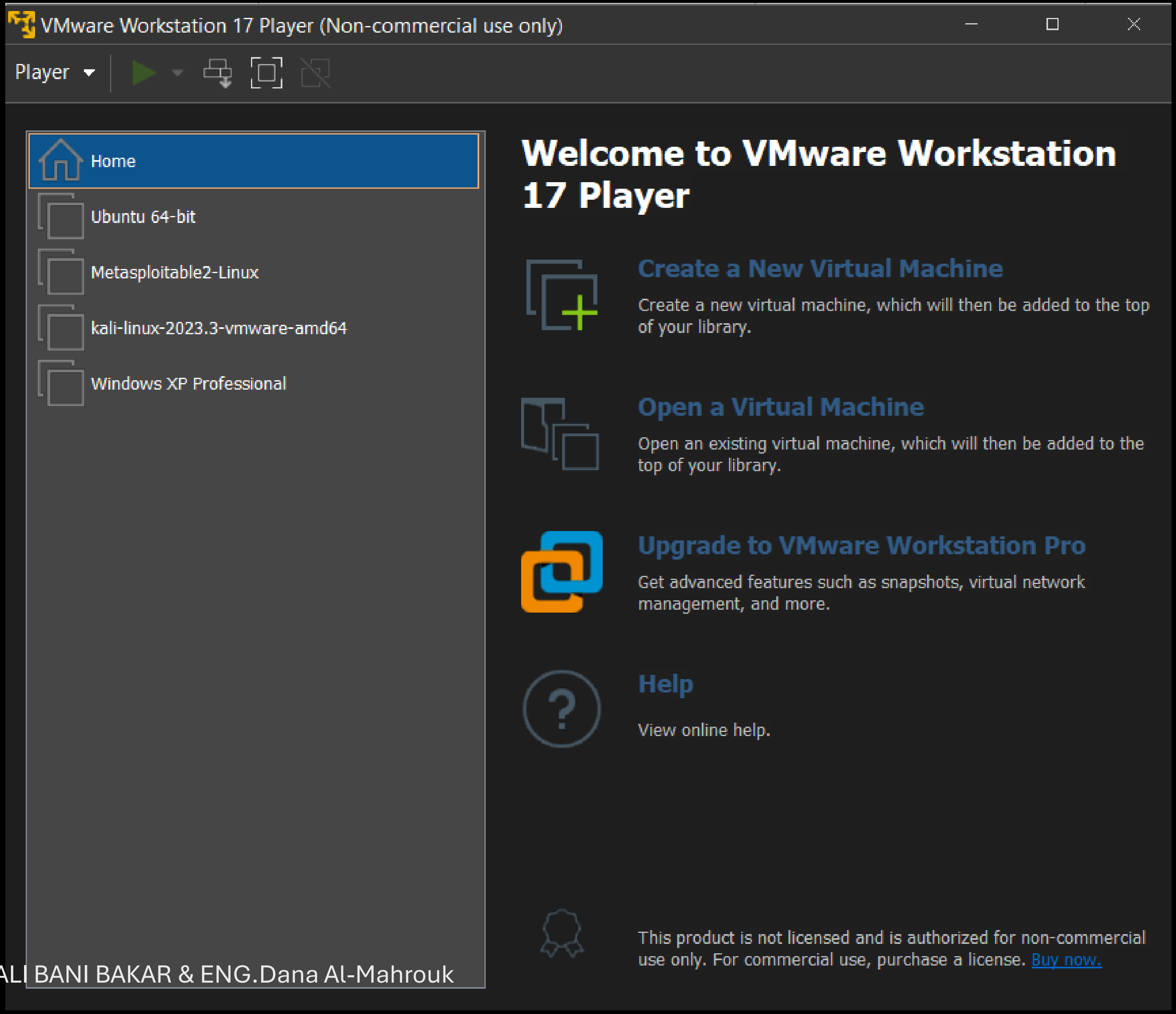
1. Networks
2. Linux Essentials
3. Cybersecurity Foundation
4. Ethical Hacking
5. Digital Forensic Investigation

Day 7

- Outline:
  1. VMware & Ubuntu
  2. Pwd
  3. Ls
  4. Touch
  5. Mkdir
  6. Cd
  7. Path (relative & absolute)
  8. Sudo
  9. Adduser
  10. Su
  11. Telder (~)
  12. $PATH

VMware

- VMware specializes in providing software solutions that enable the creation and management of virtualized IT environments.



VMware Workstation 17 Player (Non-commercial use only)

Welcome to VMware Workstation 17 Player

Create a New Virtual Machine

Create a new virtual machine, which will then be added to the top of your library.

Open a Virtual Machine

Open an existing virtual machine, which will then be added to the top of your library.

Upgrade to VMware Workstation Pro

Get advanced features such as snapshots, virtual network management, and more.

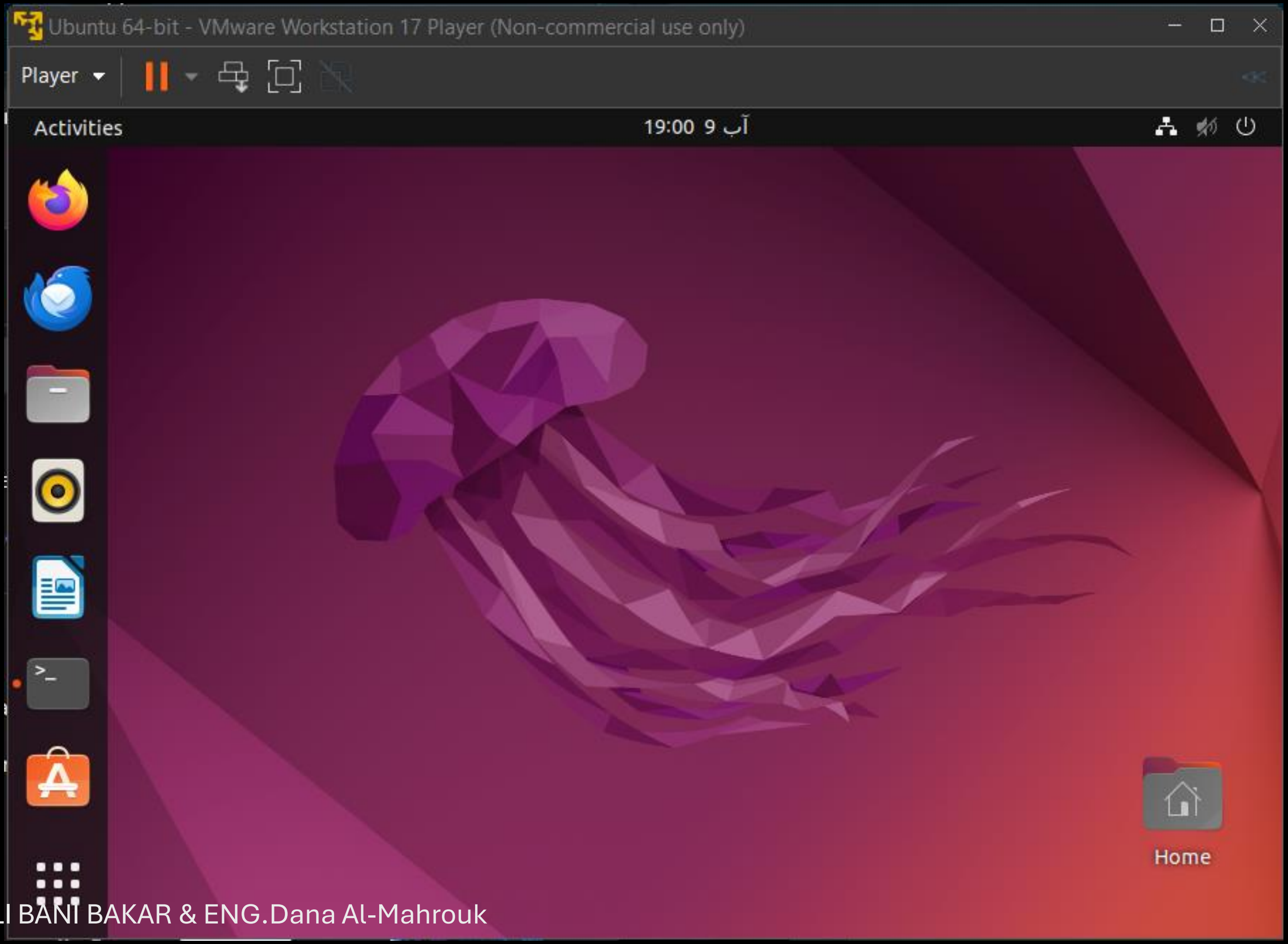
Help

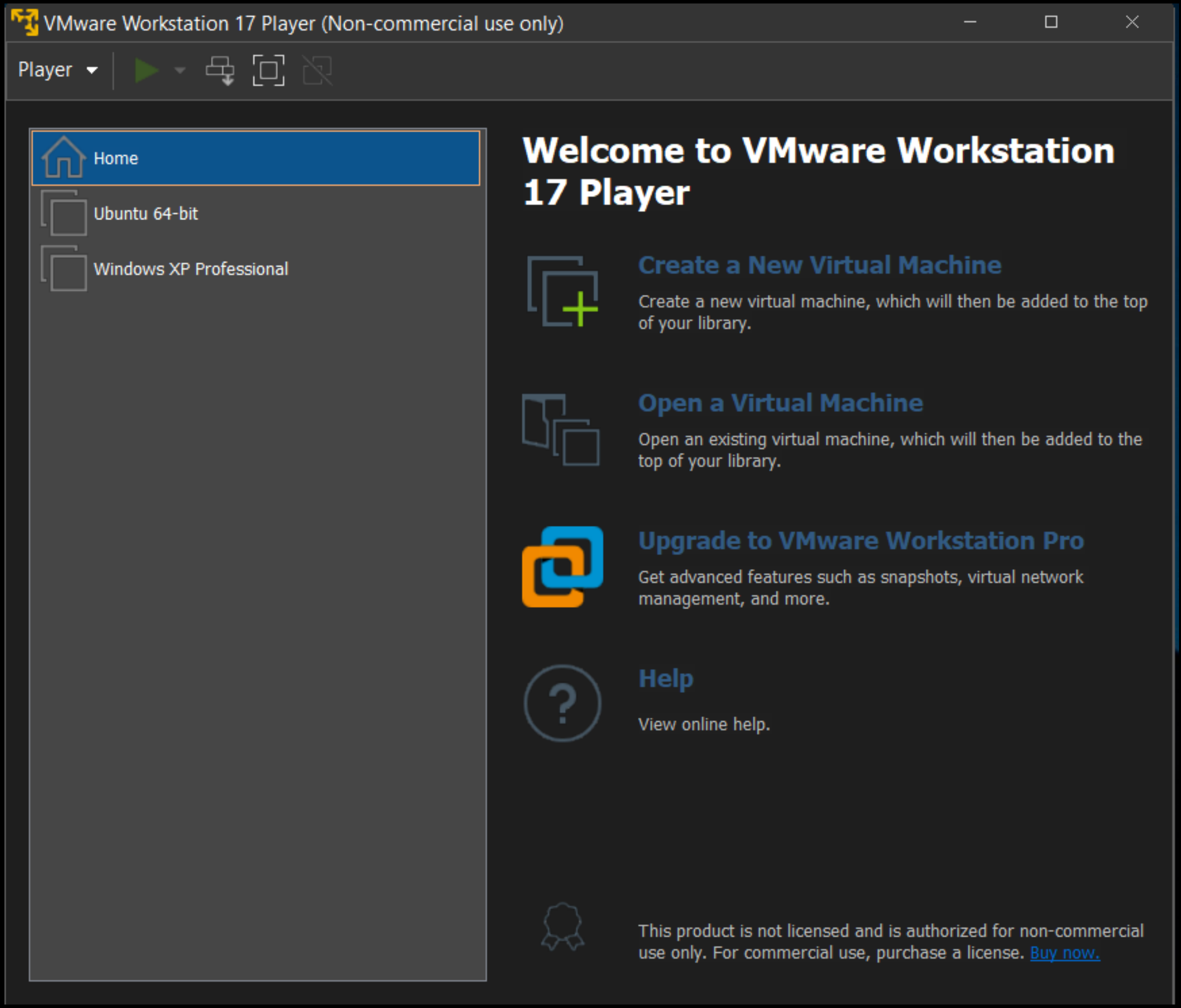
View online help.

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Ubuntu

- Ubuntu is a popular open-source Linux distribution based on Debian and has gained widespread use due to its user-friendly interface, robust security features, and strong community support.


Ubuntu 64-bit - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
19:00 9 آب
Home



Welcome to VMware Workstation 17 Player

Create a New Virtual Machine

Create a new virtual machine, which will then be added to the top of your library.

Open a Virtual Machine

Open an existing virtual machine, which will then be added to the top of your library.

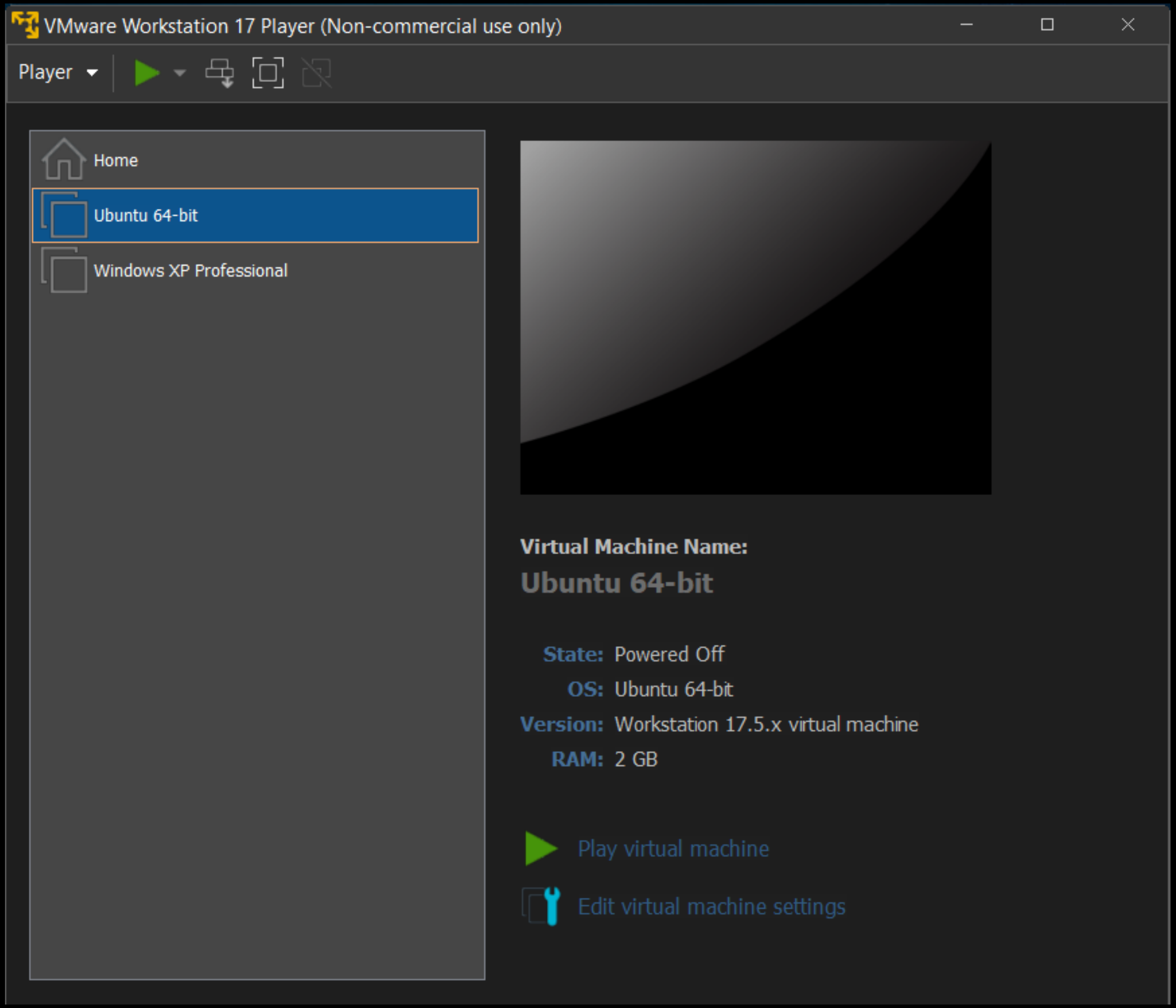
Upgrade to VMware Workstation Pro

Get advanced features such as snapshots, virtual network management, and more.

Help

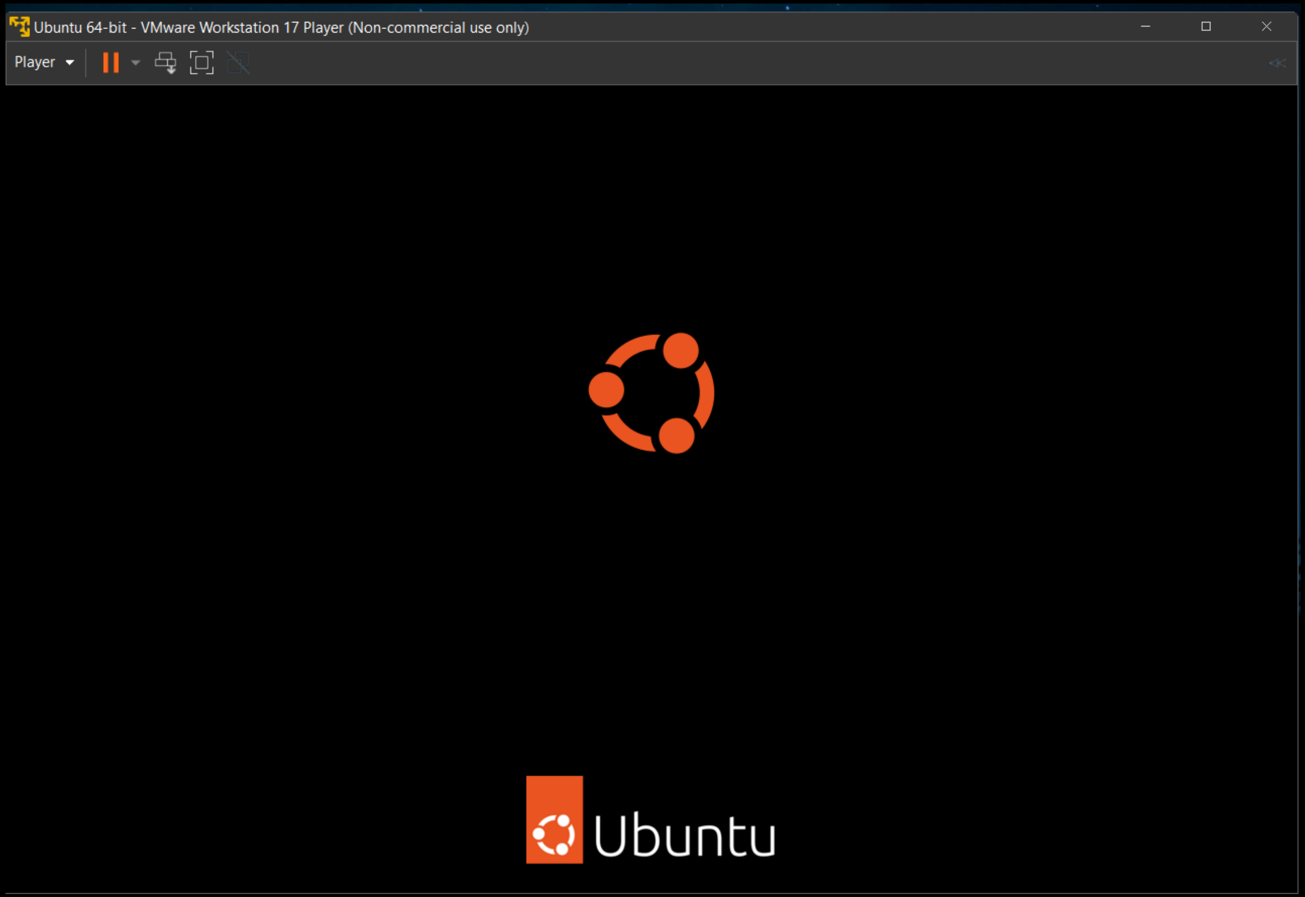
View online help.

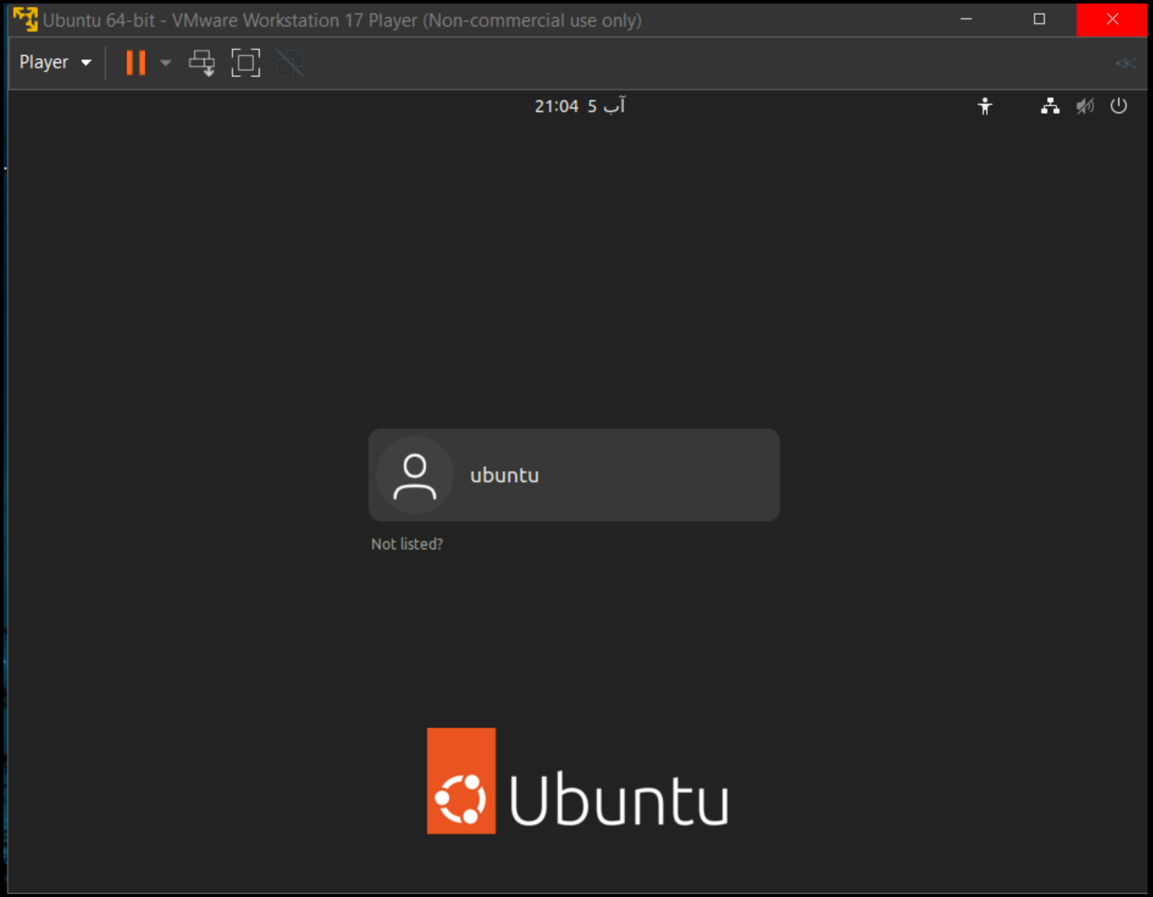
This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

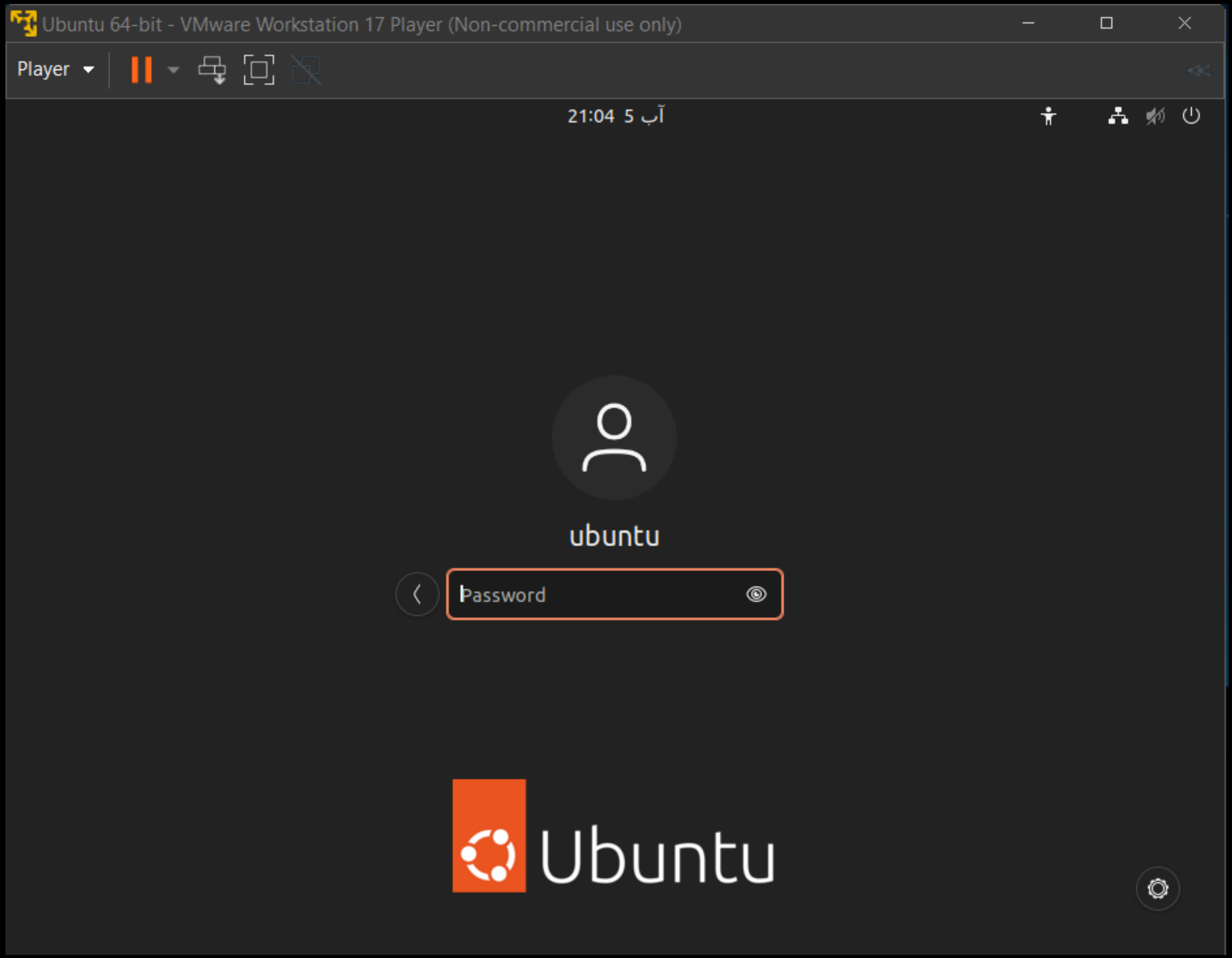

VMware Workstation 17 Player (Non-commercial use only)
Player
Home
Ubuntu 64-bit
Windows XP Professional
Play virtual machine
Edit virtual machine settings

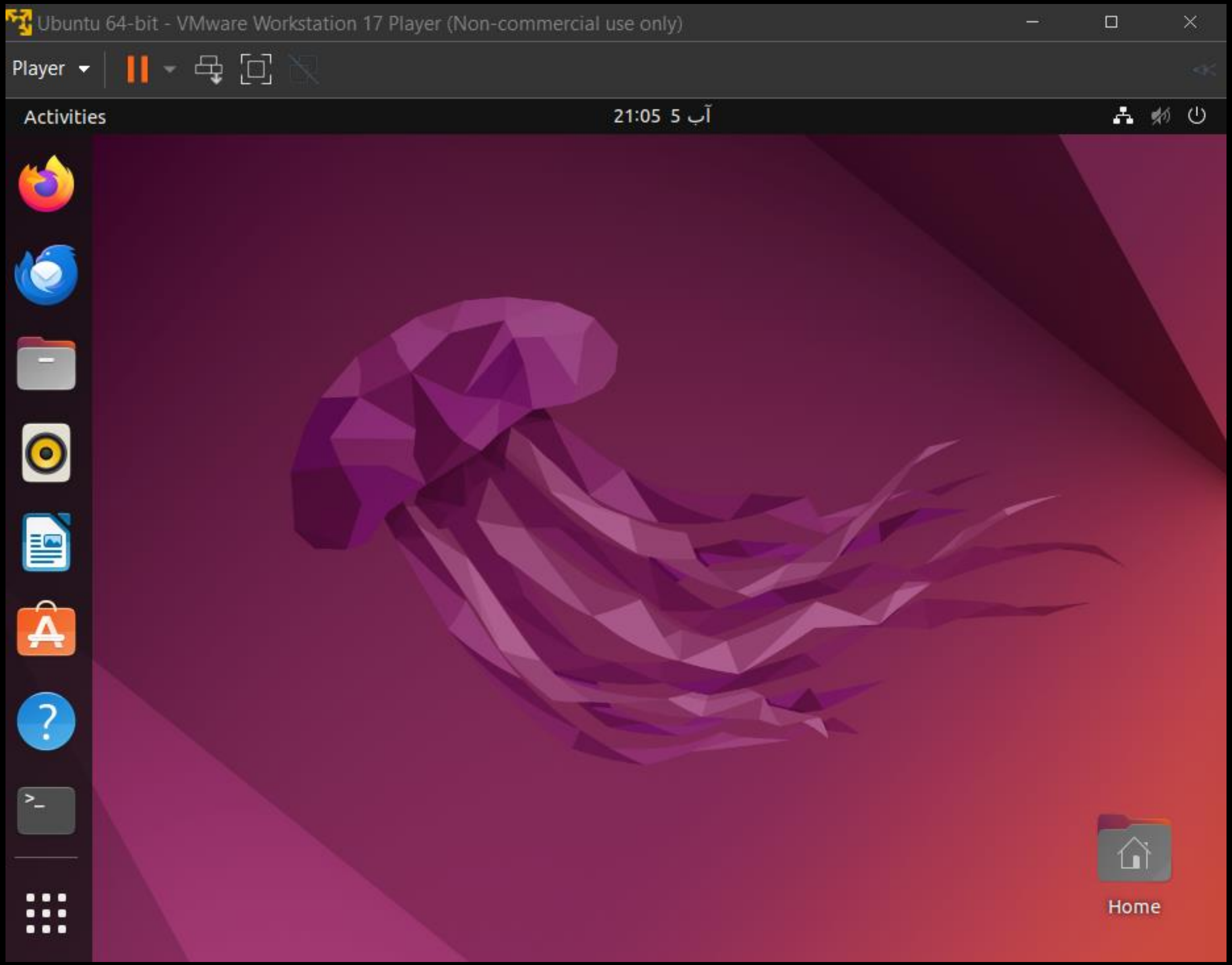
Virtual Machine Name:
Ubuntu 64-bit

State: Powered Off
OS: Ubuntu 64-bit
Version: Workstation 17.5.x virtual machine
RAM: 2 GB


Ubuntu 64-bit - VMware Workstation 17 Player (Non-commercial use only)
Player
Ubuntu

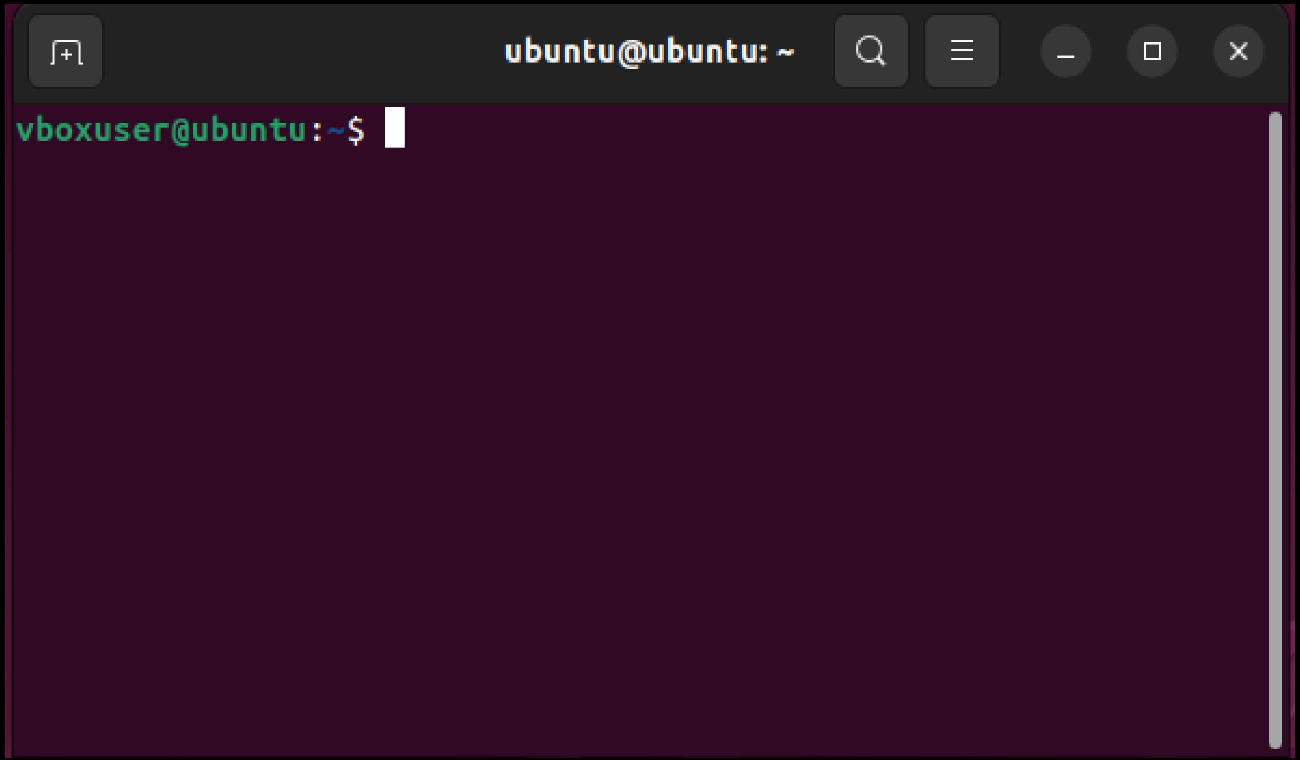

Ubuntu 64-bit - VMware Workstation 17 Player (Non-commercial use only)
Player
21:04 5 آب
ubuntu
Not listed?
Ubuntu


Ubuntu 64-bit - VMware Workstation 17 Player (Non-commercial use only)
Player
21:04 5 آب
ubuntu
Password
Ubuntu


Ubuntu 64-bit - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
21:05 5 آب
Home

Terminal

- is a text-based interface used to interact with your computer's operating system. Instead of using a graphical user interface (GUI) with windows, icons, and menus, you type commands into the terminal to perform tasks.



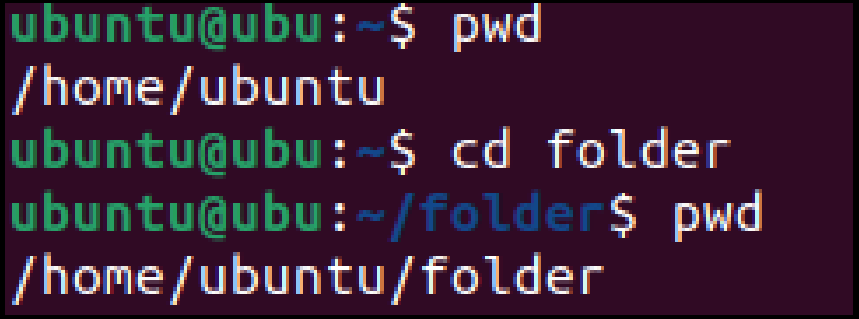
```
vboxuser@ubuntu:~$ 
```

- Type commands **directly** into the terminal to perform various tasks.
- The terminal runs a **command-line interpreter**, or "**shell**," which processes the commands you enter.

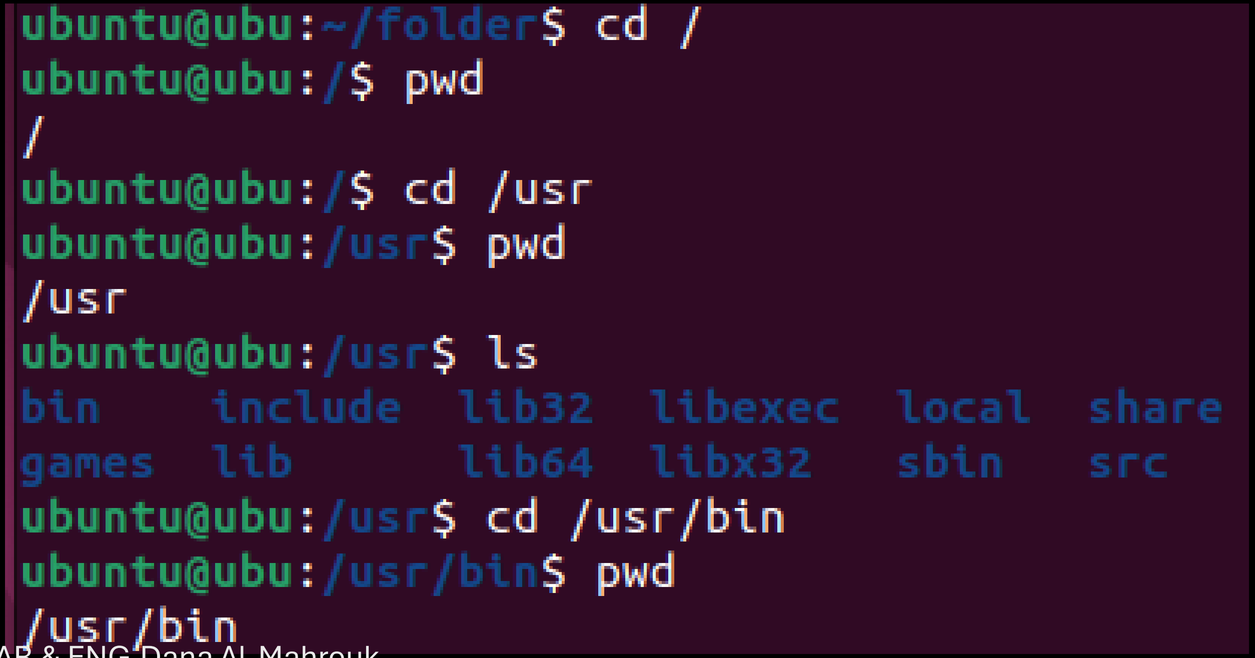
- **Faster for performing** repetitive tasks, as it allows for scripting and automation.
- Gives you **more control** over the system

pwd (Print Working Directory)

- It is used to display the path or location where you are now.



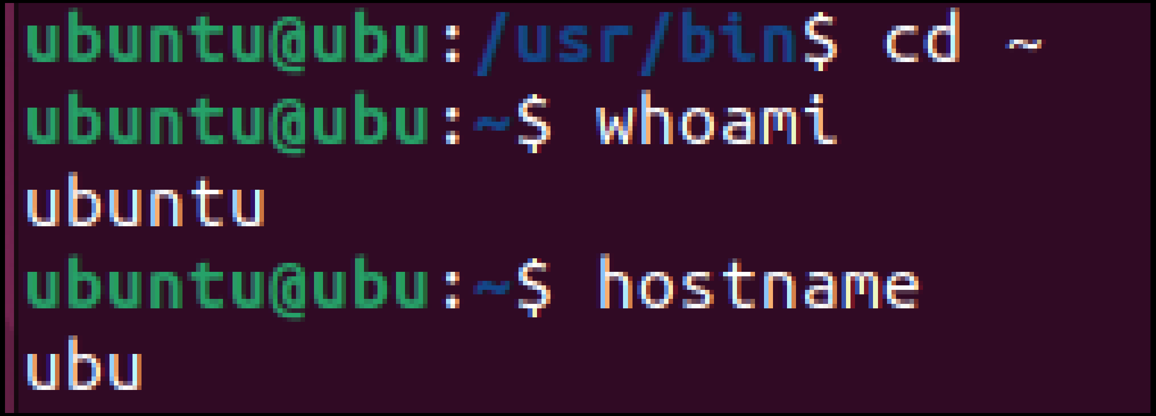
```
ubuntu@ubu:~$ pwd
/home/ubuntu
ubuntu@ubu:~$ cd folder
ubuntu@ubu:~/folder$ pwd
/home/ubuntu/folder
```



```
ubuntu@ubu:~/folder$ cd /
ubuntu@ubu:/$ pwd
/
ubuntu@ubu:/$ cd /usr
ubuntu@ubu:/usr$ pwd
/usr
ubuntu@ubu:/usr$ ls
bin     include  lib32  libexec  local  share
games   lib      lib64  libx32   sbin   src
ubuntu@ubu:/usr$ cd /usr/bin
ubuntu@ubu:/usr/bin$ pwd
/usr/bin
```

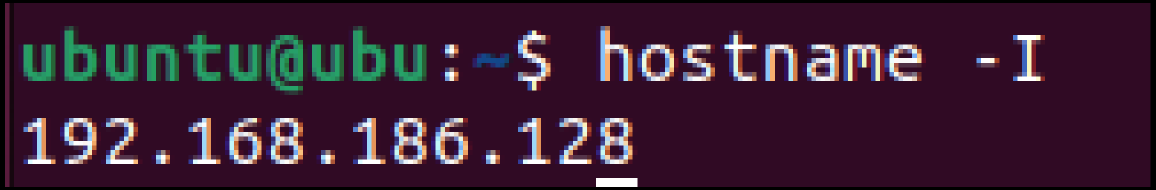
whoami

- This command displays the **username of the current user** who is logged into the system.



```
ubuntu@ubu:/usr/bin$ cd ~
ubuntu@ubu:~$ whoami
ubuntu
ubuntu@ubu:~$ hostname
ubu
```

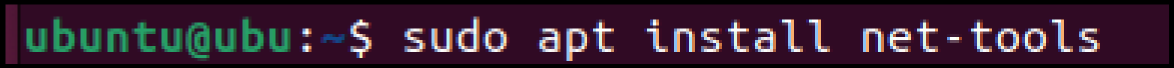
hostname



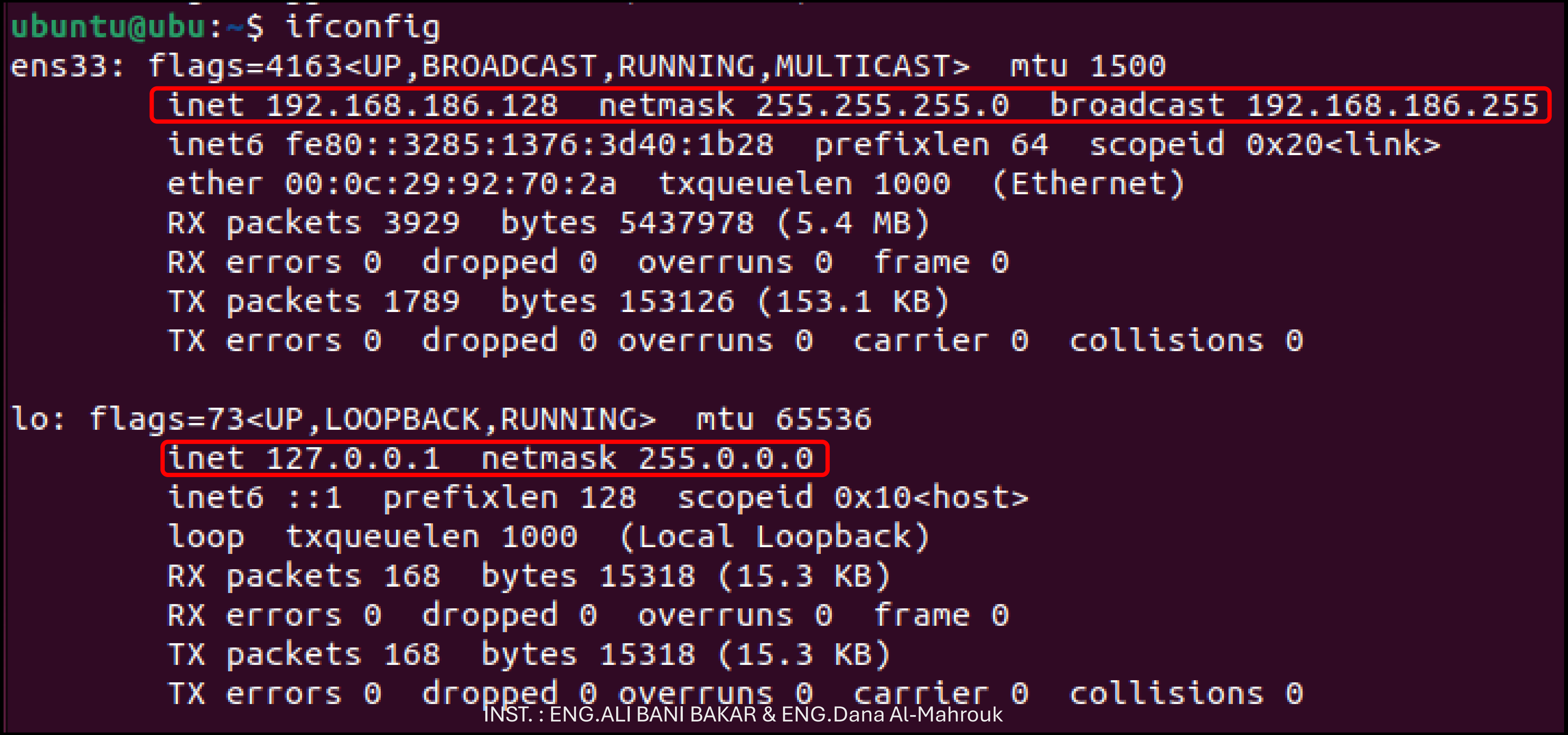
```
ubuntu@ubu:~$ hostname -I
192.168.186.128
```

- - This command displays the name of the **current machine or system** you're working on.
- - The hostname is typically **set during the installation** of the operating system and is used to identify the machine on a network.
- - You can also set a new hostname using this command

Ifconfig (old version)



```
ubuntu@ubu:~$ sudo apt install net-tools
```

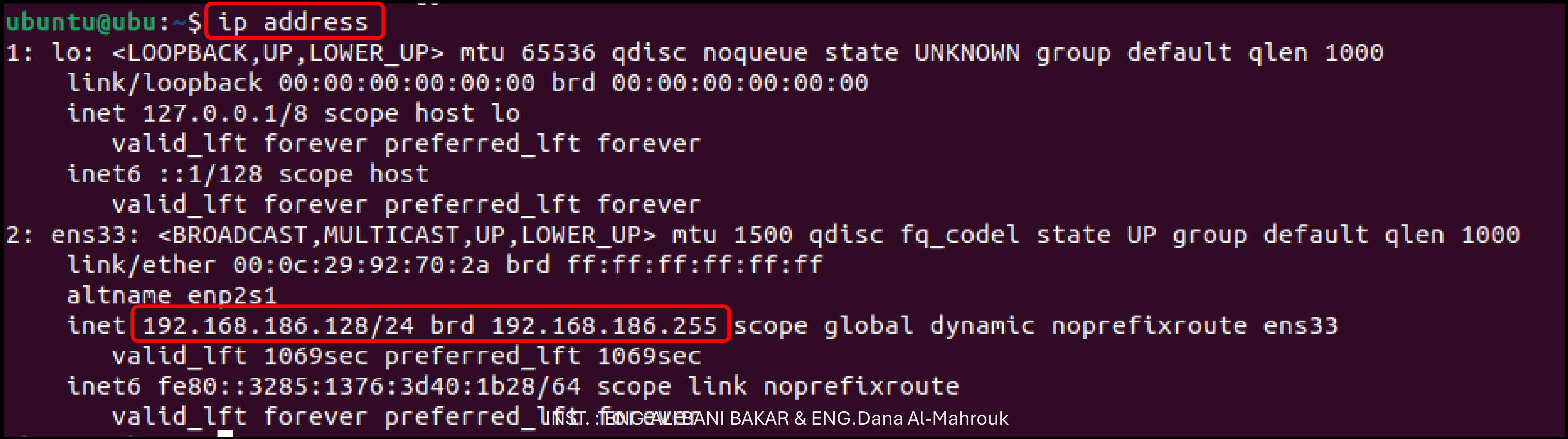


```
ubuntu@ubu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.186.128  netmask 255.255.255.0  broadcast 192.168.186.255
        inet6 fe80::3285:1376:3d40:1b28  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:92:70:2a  txqueuelen 1000  (Ethernet)
        RX packets 3929  bytes 5437978 (5.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1789  bytes 153126 (153.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 168  bytes 15318 (15.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 168  bytes 15318 (15.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

ip address (newer version)

iproute2 package

- This command will list all network interfaces along with their IP addresses.
- determine the IP address or addresses of your Linux system.



```
ubuntu@ubu:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:92:70:2a brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.186.128/24 brd 192.168.186.255 scope global dynamic noprefixroute ens33
       valid_lft 1069sec preferred_lft 1069sec
    inet6 fe80::3285:1376:3d40:1b28/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

ls (List)

- Lists the files and directories in the current directory.

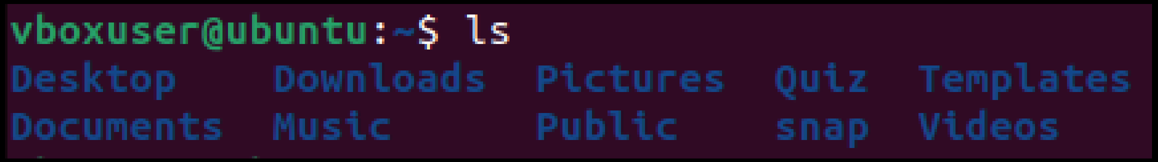
-l ➔ detailed information

-h ➔ Human-Readable Sizes, KB, MB, GB, etc…

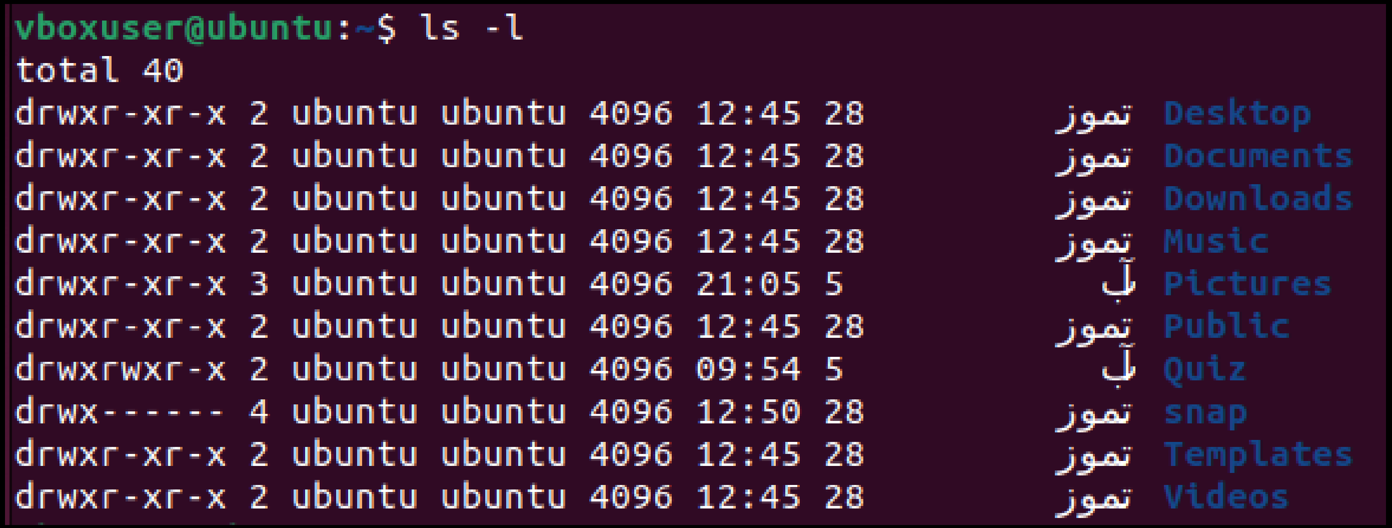
-a ➔ all files, including hidden files

-i ➔ inode number

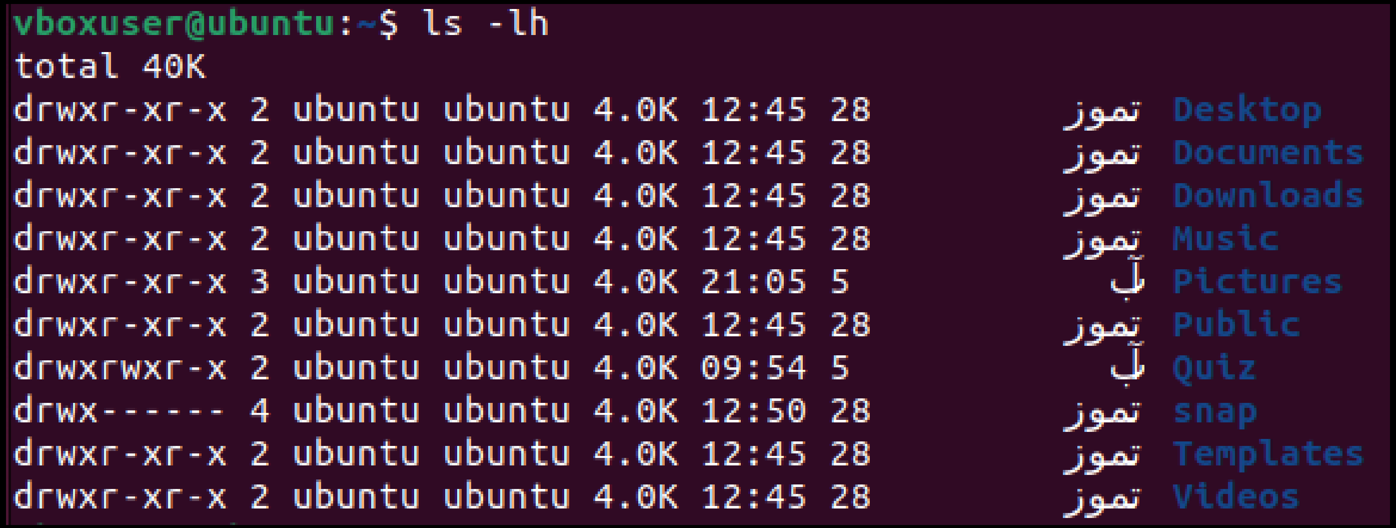
-R ➔ list the current directory and all of its subdirectories.



```
vboxuser@ubuntu:~$ ls
Desktop    Downloads  Pictures  Quiz  Templates
Documents  Music      Public    snap  Videos
```

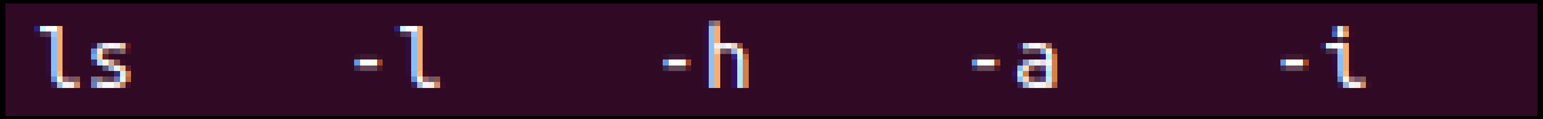


```
vboxuser@ubuntu:~$ ls -l
total 40
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28        تموز Desktop
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28        تموز Documents
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28        تموز Downloads
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28        تموز Music
drwxr-xr-x 3 ubuntu ubuntu 4096 21:05 5          آب Pictures
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28        تموز Public
drwxrwxr-x 2 ubuntu ubuntu 4096 09:54 5          آب Quiz
drwx------ 4 ubuntu ubuntu 4096 12:50 28        تموز snap
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28        تموز Templates
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28        تموز Videos
```



```
vboxuser@ubuntu:~$ ls -lh
total 40K
drwxr-xr-x 2 ubuntu ubuntu 4.0K 12:45 28        تموز Desktop
drwxr-xr-x 2 ubuntu ubuntu 4.0K 12:45 28        تموز Documents
drwxr-xr-x 2 ubuntu ubuntu 4.0K 12:45 28        تموز Downloads
drwxr-xr-x 2 ubuntu ubuntu 4.0K 12:45 28        تموز Music
drwxr-xr-x 3 ubuntu ubuntu 4.0K 21:05 5          آب Pictures
drwxr-xr-x 2 ubuntu ubuntu 4.0K 12:45 28        تموز Public
drwxrwxr-x 2 ubuntu ubuntu 4.0K 09:54 5          آب Quiz
drwx------ 4 ubuntu ubuntu 4.0K 12:50 28        تموز snap
drwxr-xr-x 2 ubuntu ubuntu 4.0K 12:45 28        تموز Templates
drwxr-xr-x 2 ubuntu ubuntu 4.0K 12:45 28        تموز Videos
```

Program

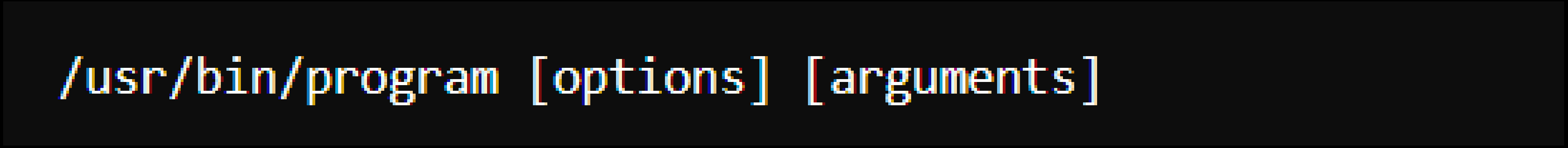


```
ls      -l      -h      -a      -i
```

Program Name	Option	Option	Option	Option
Argument[0]	Arg [1]	Arg [2]	Arg [3]	Arg[4]

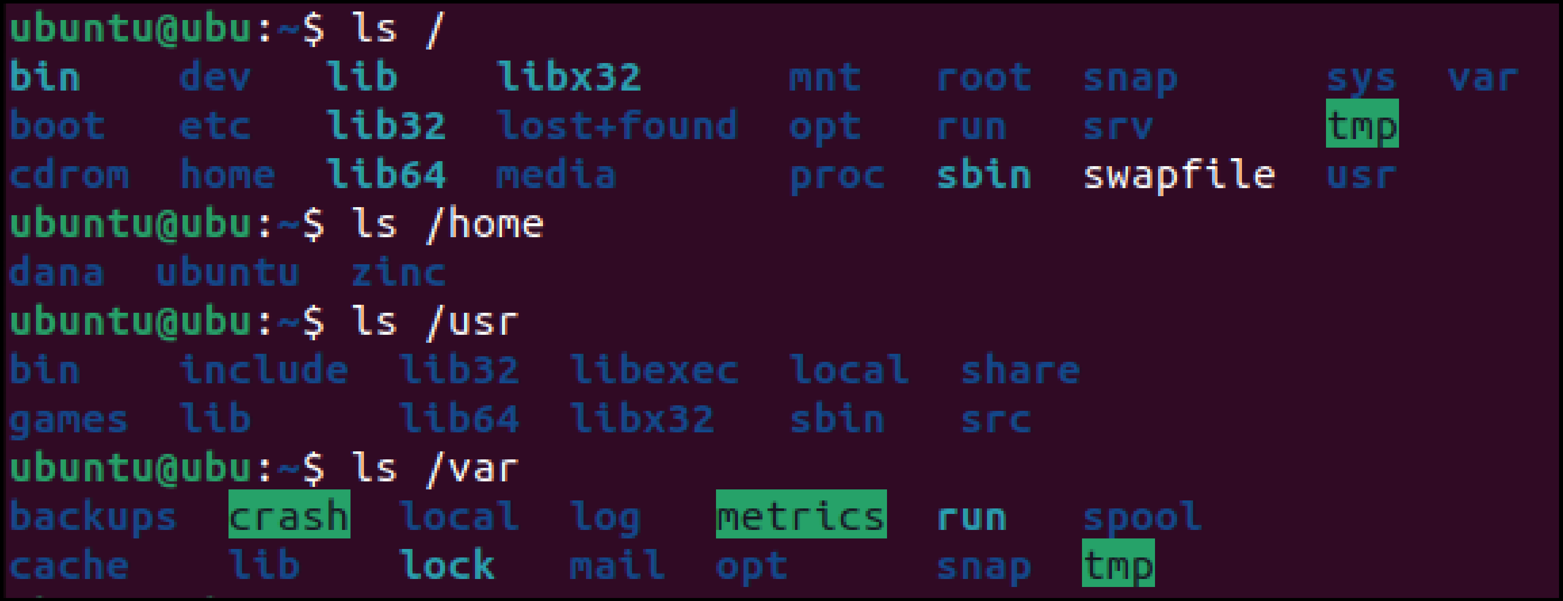
/usr/bin directory

- **/usr/bin**: is a directory that contains **executable programs**, or binaries, which can be **run from the command line**.
- When you run a program from ` /usr/bin ` , you can often specify **options and arguments** to control its behavior.

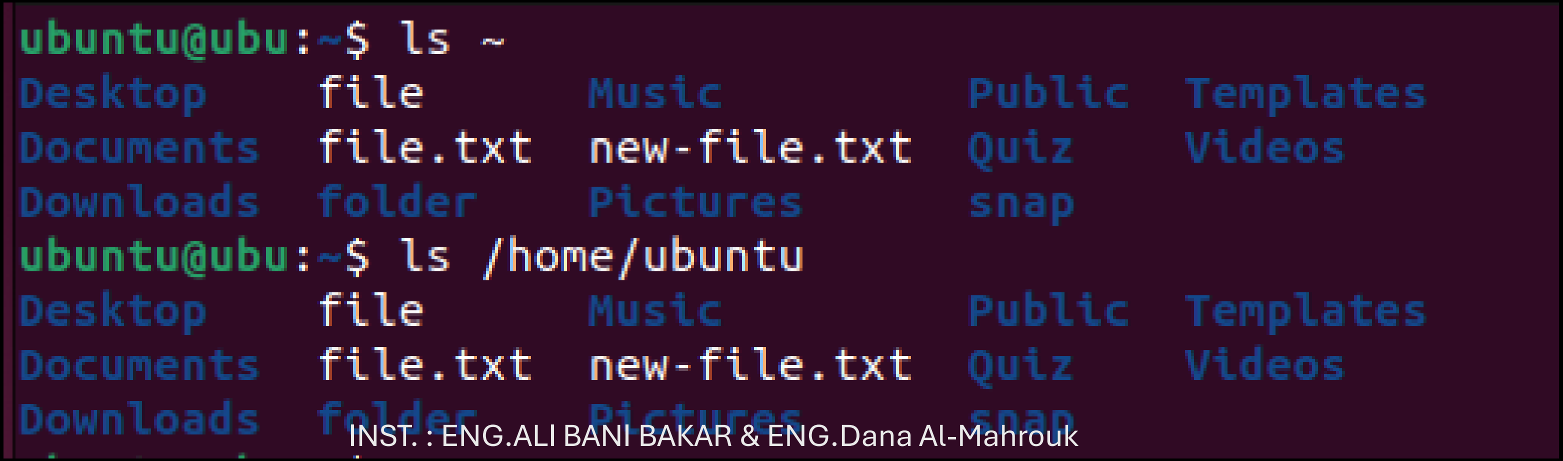


```
/usr/bin/program [options] [arguments]
```

ls <path> ➔ Absolute

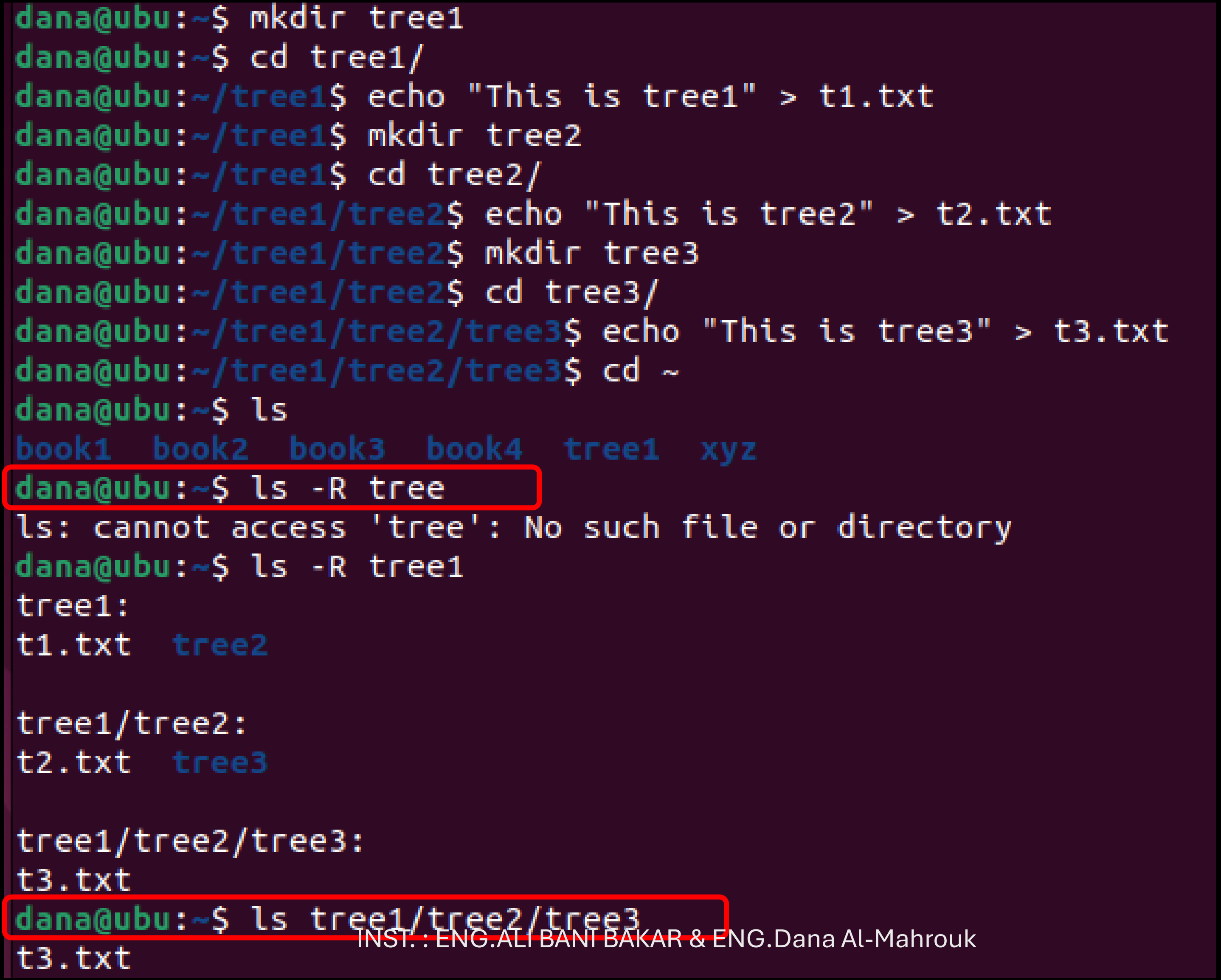


```
ubuntu@ubu:~$ ls /
bin     dev    lib     libx32      mnt    root   snap      sys   var
boot    etc    lib32   lost+found  opt    run    srv       tmp
cdrom   home   lib64   media       proc   sbin   swapfile  usr
ubuntu@ubu:~$ ls /home
dana  ubuntu  zinc
ubuntu@ubu:~$ ls /usr
bin    include  lib32  libexec  local  share
games  lib      lib64  libx32   sbin   src
ubuntu@ubu:~$ ls /var
backups  crash  local  log   metrics  run   spool
cache    lib    lock   mail  opt      snap  tmp
```



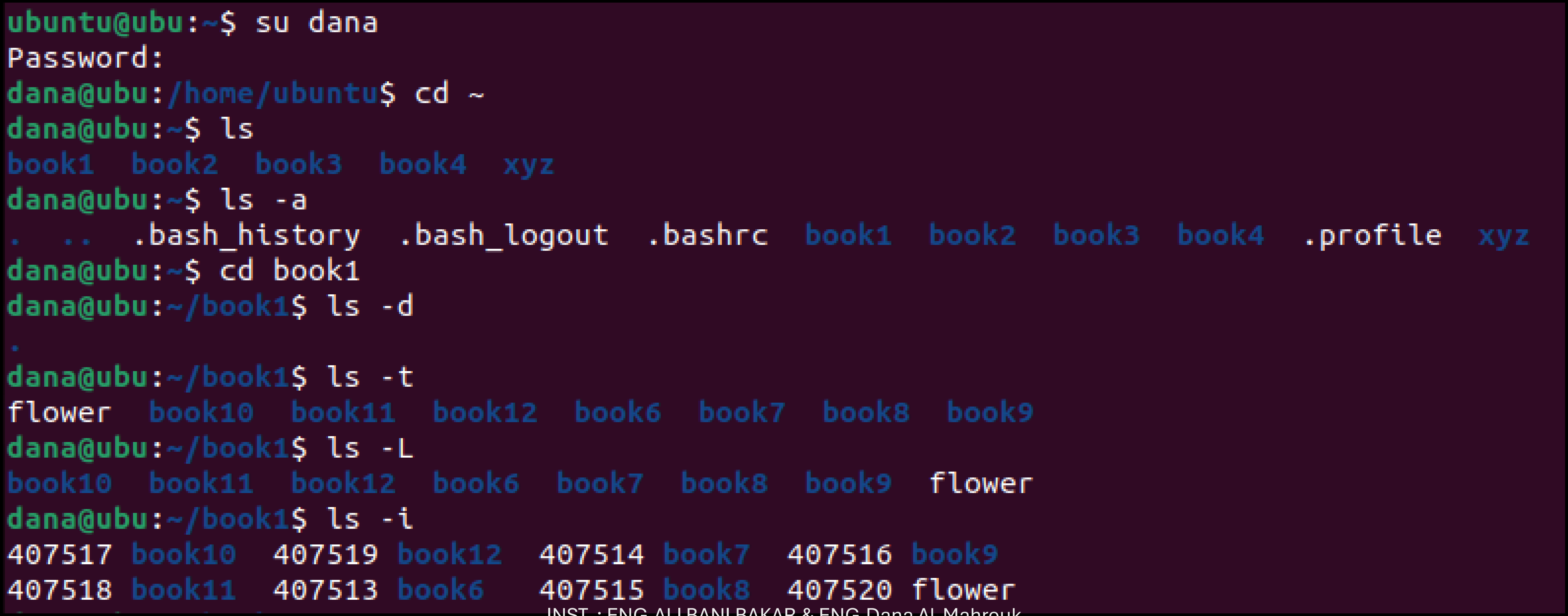
```
ubuntu@ubu:~$ ls ~
Desktop    file      Music         Public  Templates
Documents  file.txt  new-file.txt  Quiz    Videos
Downloads  folder    Pictures      snap
ubuntu@ubu:~$ ls /home/ubuntu
Desktop    file      Music         Public  Templates
Documents  file.txt  new-file.txt  Quiz    Videos
Downloads  folder    Pictures      snap
```

Ls <path> ➔ Relative



```
dana@ubu:~$ mkdir tree1
dana@ubu:~$ cd tree1/
dana@ubu:~/tree1$ echo "This is tree1" > t1.txt
dana@ubu:~/tree1$ mkdir tree2
dana@ubu:~/tree1$ cd tree2/
dana@ubu:~/tree1/tree2$ echo "This is tree2" > t2.txt
dana@ubu:~/tree1/tree2$ mkdir tree3
dana@ubu:~/tree1/tree2$ cd tree3/
dana@ubu:~/tree1/tree2/tree3$ echo "This is tree3" > t3.txt
dana@ubu:~/tree1/tree2/tree3$ cd ~
dana@ubu:~$ ls
book1  book2  book3  book4  tree1  xyz
dana@ubu:~$ ls -R tree
ls: cannot access 'tree': No such file or directory
dana@ubu:~$ ls -R tree1
tree1:
t1.txt  tree2

tree1/tree2:
t2.txt  tree3

tree1/tree2/tree3:
t3.txt
dana@ubu:~$ ls tree1/tree2/tree3
t3.txt
```

ls more options…



```
ubuntu@ubu:~$ su dana
Password:
dana@ubu:/home/ubuntu$ cd ~
dana@ubu:~$ ls
book1  book2  book3  book4  xyz
dana@ubu:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  book1  book2  book3  book4  .profile  xyz
dana@ubu:~$ cd book1
dana@ubu:~/book1$ ls -d
.
dana@ubu:~/book1$ ls -t
flower  book10  book11  book12  book6  book7  book8  book9
dana@ubu:~/book1$ ls -L
book10  book11  book12  book6  book7  book8  book9  flower
dana@ubu:~/book1$ ls -i
407517 book10  407519 book12  407514 book7  407516 book9
407518 book11  407513 book6   407515 book8  407520 flower
```

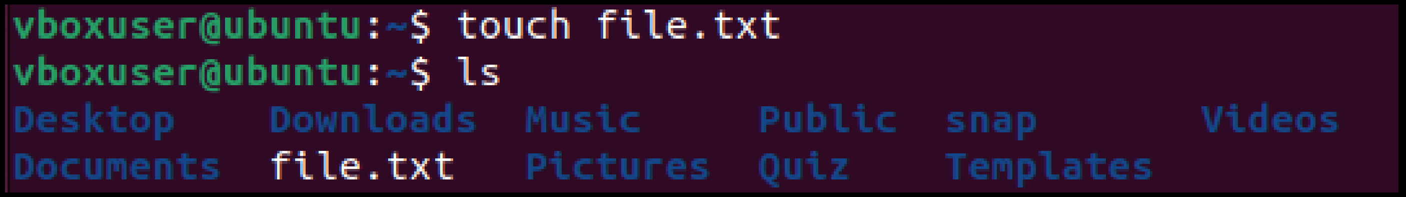
touch

- **Creates a new empty file** or updates the timestamp of an existing file.
- If `file.txt` already exists, `touch` will update its access and modification timestamps to the current time.
- file extensions like `.txt` are primarily used as a convention to indicate the file type or content.
- **The extension is more for user convenience and does not dictate the file's behavior in Linux.**

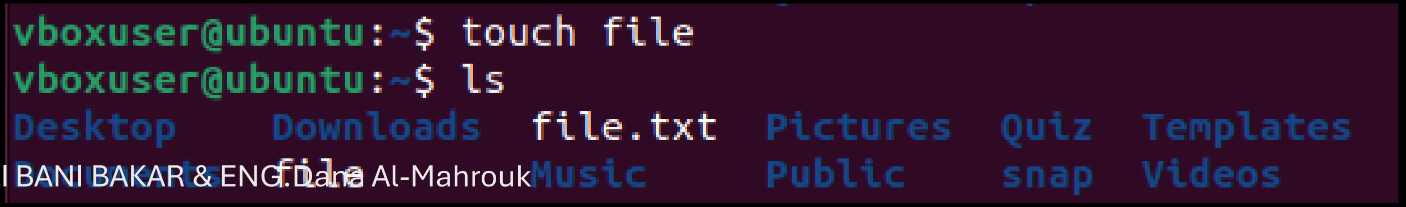
In **Windows**, file extensions are crucial as they are used by the operating system to **determine which program should open the file.**

For example, `.txt` files open with

Notepad by default.



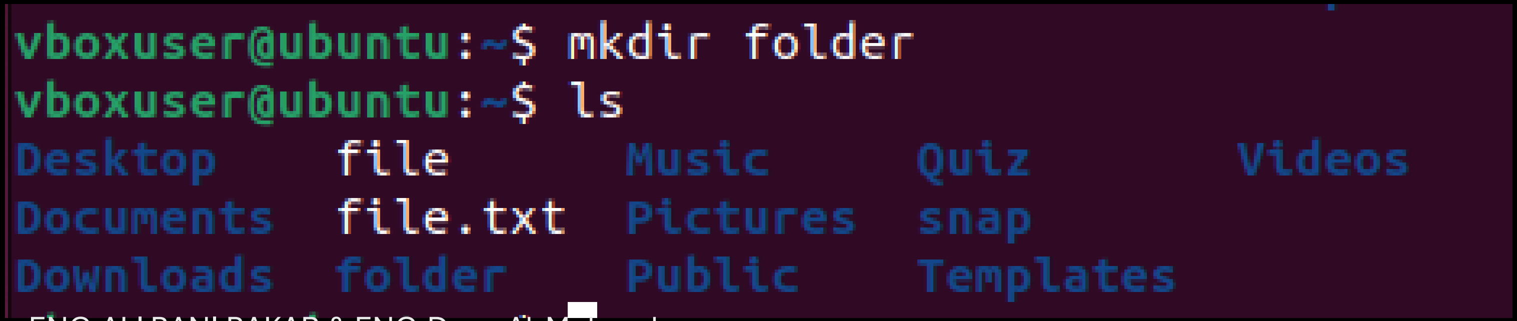
```
vboxuser@ubuntu:~$ touch file.txt
vboxuser@ubuntu:~$ ls
Desktop     Downloads  Music      Public  snap       Videos
Documents   file.txt   Pictures   Quiz    Templates
```



```
vboxuser@ubuntu:~$ touch file
vboxuser@ubuntu:~$ ls
Desktop     Downloads  file.txt  Pictures  Quiz  Templates
Documents   file       Music     Public    snap  Videos
```

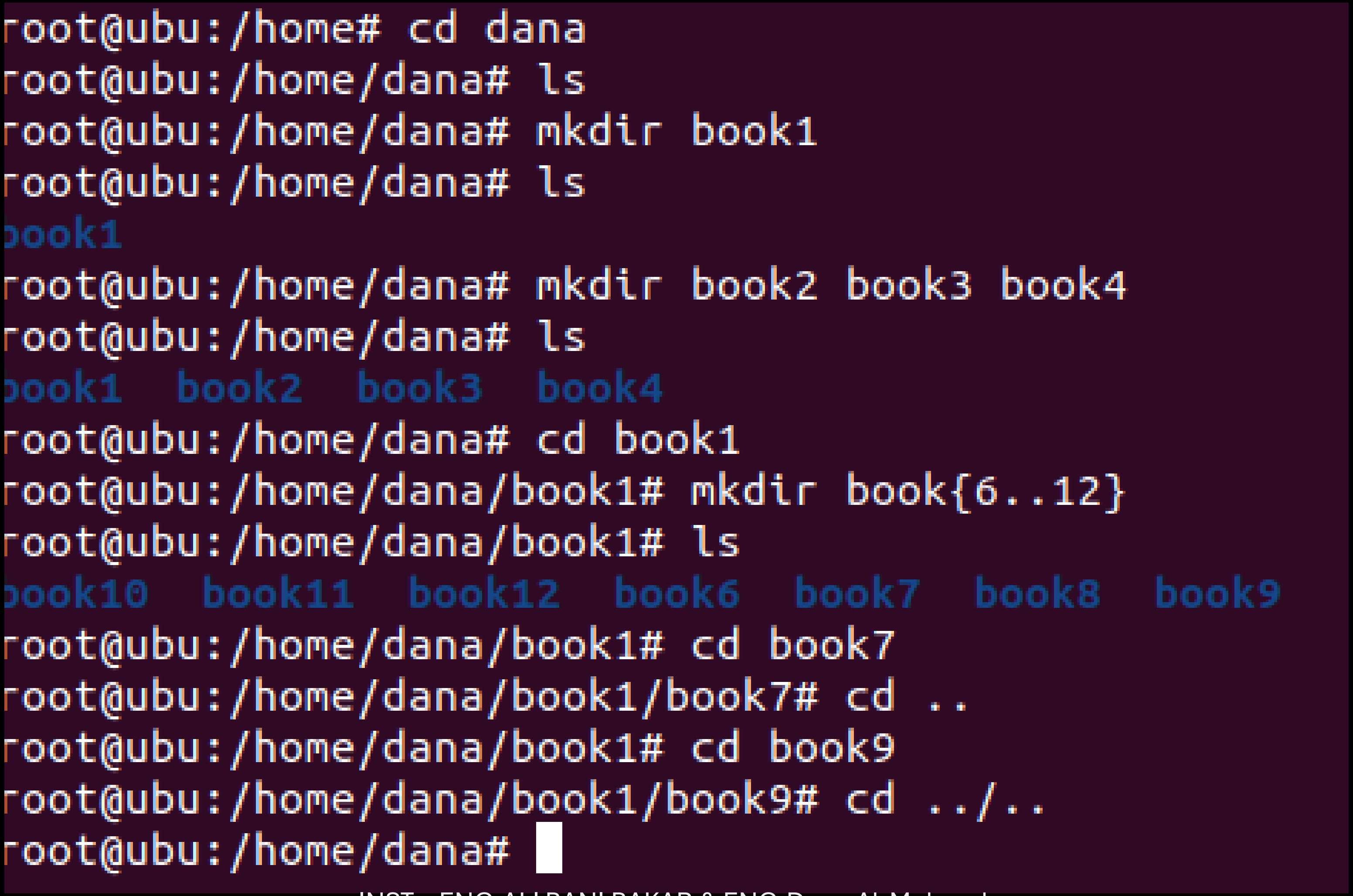
mkdir (Make Directory)

- Creates a new directory.



```
vboxuser@ubuntu:~$ mkdir folder
vboxuser@ubuntu:~$ ls
Desktop    file      Music     Quiz       Videos
Documents  file.txt  Pictures  snap
Downloads  folder    Public    Templates
```

mkdir

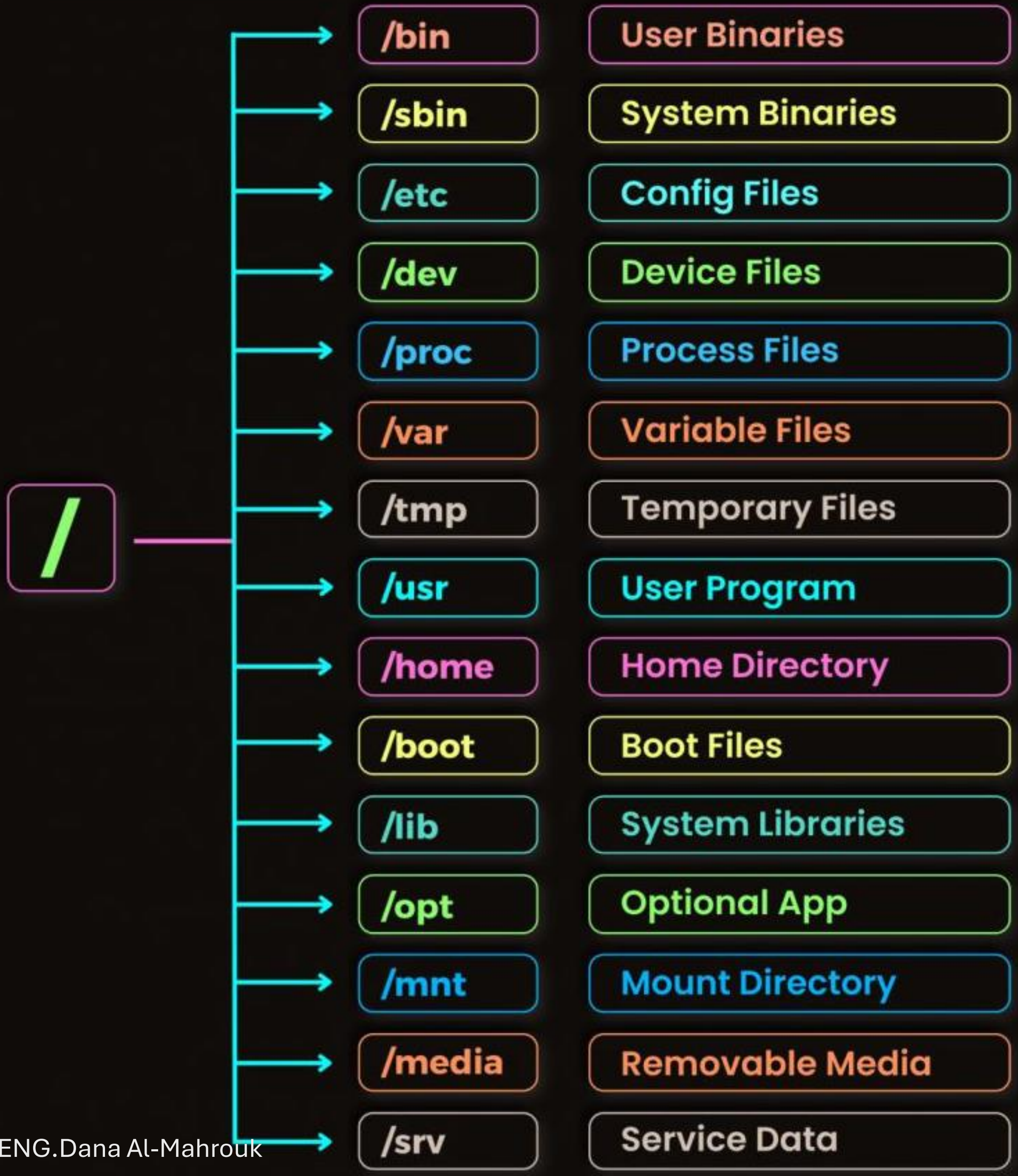


```
root@ubu:/home# cd dana
root@ubu:/home/dana# ls
root@ubu:/home/dana# mkdir book1
root@ubu:/home/dana# ls
book1
root@ubu:/home/dana# mkdir book2 book3 book4
root@ubu:/home/dana# ls
book1  book2  book3  book4
root@ubu:/home/dana# cd book1
root@ubu:/home/dana/book1# mkdir book{6..12}
root@ubu:/home/dana/book1# ls
book10  book11  book12  book6  book7  book8  book9
root@ubu:/home/dana/book1# cd book7
root@ubu:/home/dana/book1/book7# cd ..
root@ubu:/home/dana/book1# cd book9
root@ubu:/home/dana/book1/book9# cd ../..
root@ubu:/home/dana# 
```

Path Relative & Absolute

- Absolute Path: is a complete path that specifies the location of a file or directory from the root of the file system.
- Ex: /home/user/documents/report.txt
- `/` is the root directory.

- Relative Path: specifies the location of a file or directory in relation to the current working directory. It does not start from the root but rather from the directory you are currently in.
- Ex: documents/report.txt



/

/bin	User Binaries
/sbin	System Binaries
/etc	Config Files
/dev	Device Files
/proc	Process Files
/var	Variable Files
/tmp	Temporary Files
/usr	User Program
/home	Home Directory
/boot	Boot Files
/lib	System Libraries
/opt	Optional App
/mnt	Mount Directory
/media	Removable Media
/srv	Service Data

cd (Change Directory)

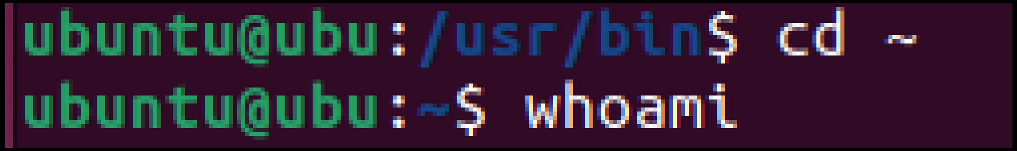
- Changes the current directory to another directory.

- **The tilde (~):**

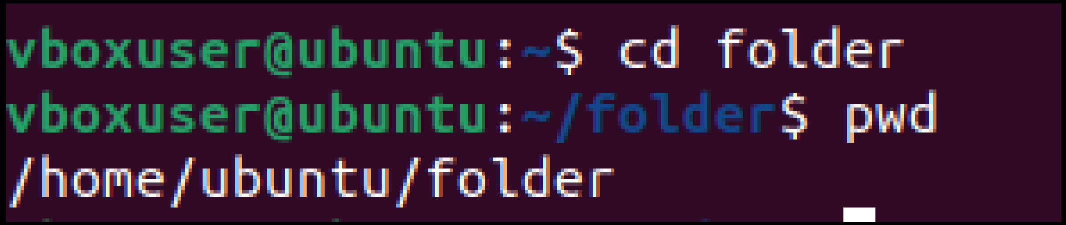
is a Linux "shortcut" to denote a user's **home directory.**

For example, for user01, file /home/user01/test.

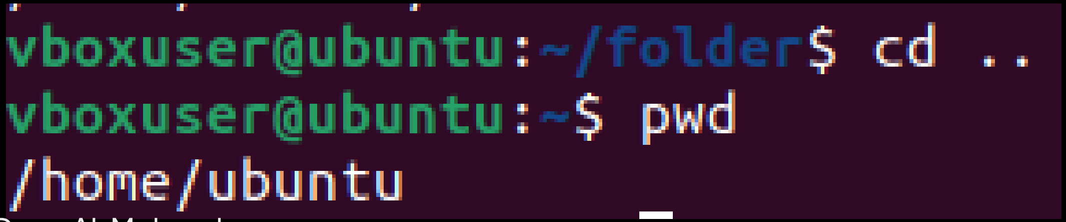
file can also be denoted by ~/test.



```
ubuntu@ubu:/usr/bin$ cd ~
ubuntu@ubu:~$ whoami
```



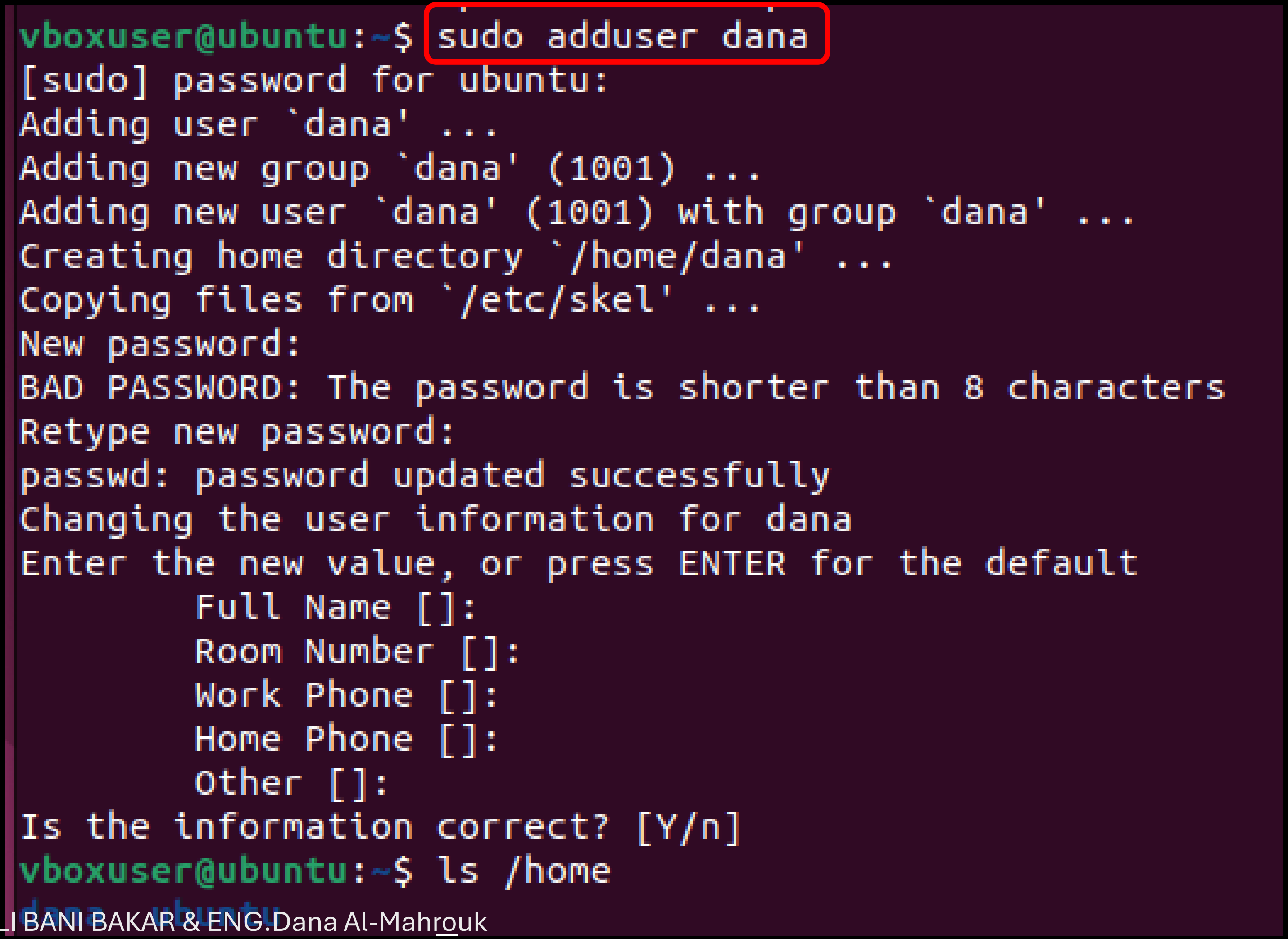
```
vboxuser@ubuntu:~$ cd folder
vboxuser@ubuntu:~/folder$ pwd
/home/ubuntu/folder
```



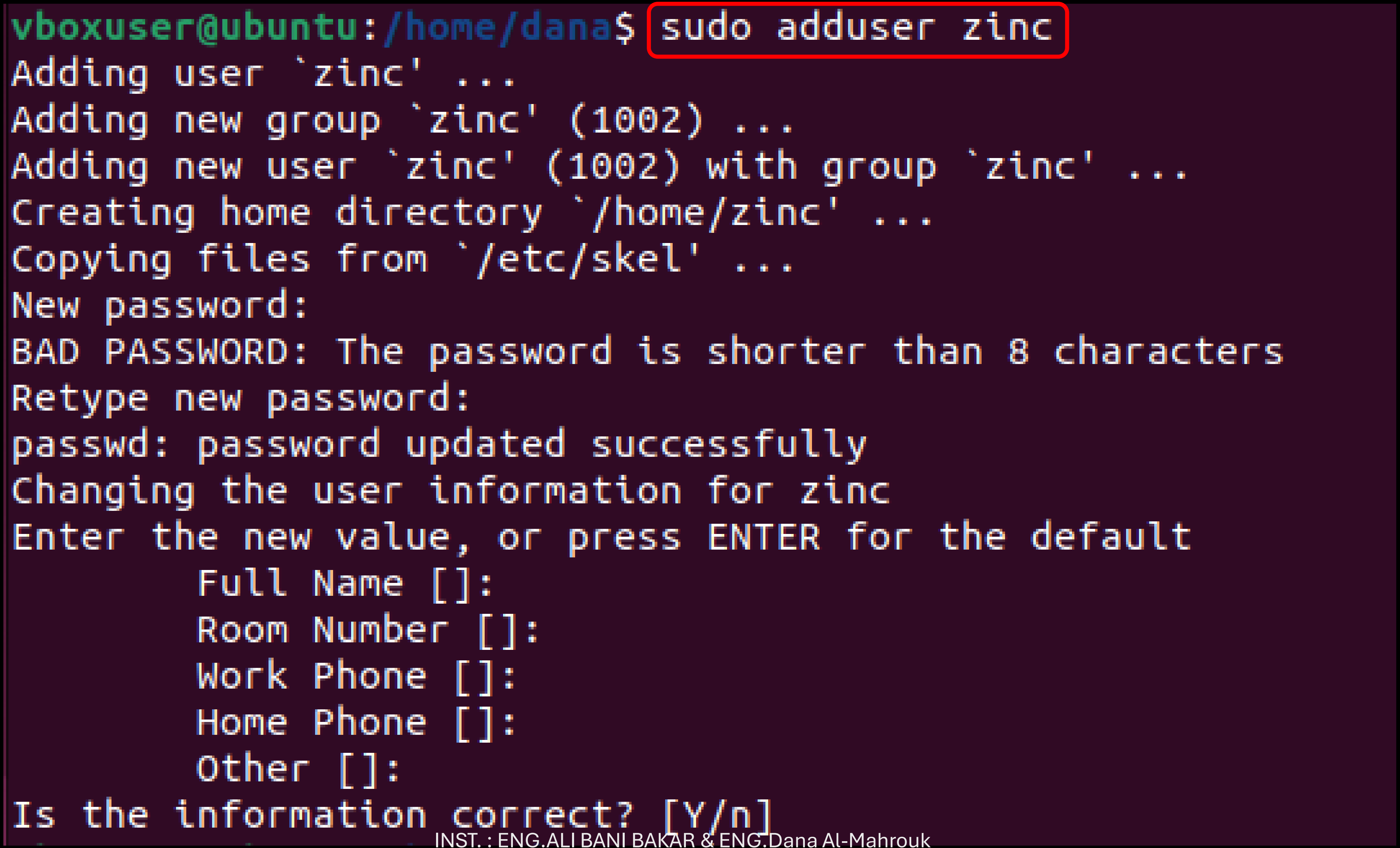
```
vboxuser@ubuntu:~/folder$ cd ..
vboxuser@ubuntu:~$ pwd
/home/ubuntu
```

adduser

- A command used to add a new user to the system.



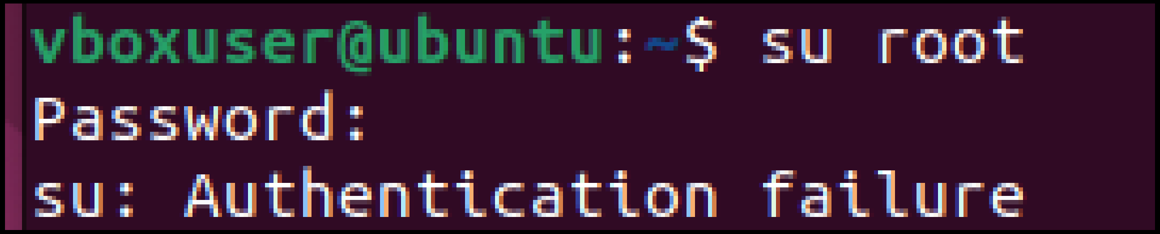
```
vboxuser@ubuntu:~$ sudo adduser dana
[sudo] password for ubuntu:
Adding user `dana' ...
Adding new group `dana' (1001) ...
Adding new user `dana' (1001) with group `dana' ...
Creating home directory `/home/dana' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for dana
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
vboxuser@ubuntu:~$ ls /home
dana  ubuntu
```



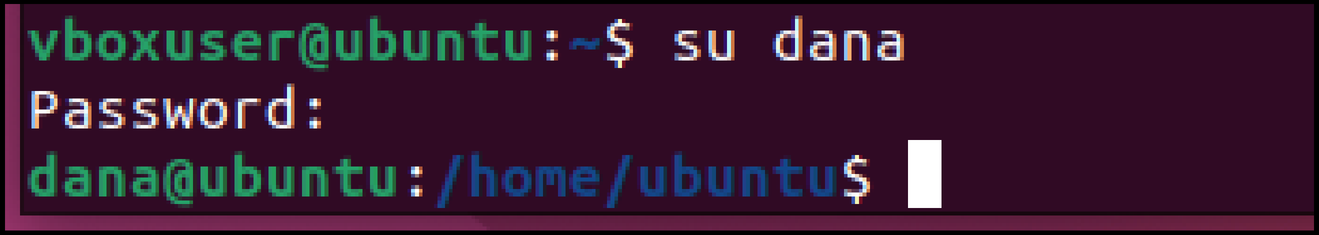
```
vboxuser@ubuntu:/home/dana$ sudo adduser zinc
Adding user `zinc' ...
Adding new group `zinc' (1002) ...
Adding new user `zinc' (1002) with group `zinc' ...
Creating home directory `/home/zinc' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for zinc
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
```

su (Substitute User)

- Switches to another user account. Often used to switch to the root (administrator) account.



```
vboxuser@ubuntu:~$ su root
Password:
su: Authentication failure
```



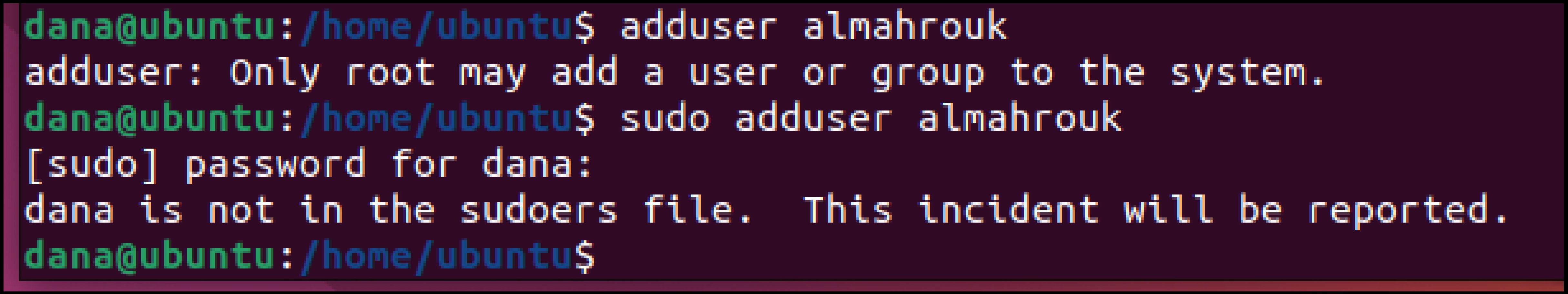
```
vboxuser@ubuntu:~$ su dana
Password:
dana@ubuntu:/home/ubuntu$
```

Day 8

- Outline
  - Permission Deny
  - File Sudoers
  - /etc directory
    - /sudoers
    - /passwd
    - /shadow
  - $PATH
  - /usr directory
    - /bin
  - Type
  - Sudo
  - Cp
  - Echo [-e \n > >> 1> 2>]

Permission

- adduser: used to **create new user accounts**. Running ` adduser` requires **root privileges**, meaning it should be executed with ` **sudo**` if you're not logged in as the **root user**.
- Regular users cannot create other users unless they have ` sudo` privileges or root access.



```
dana@ubuntu:/home/ubuntu$ adduser almahrouk
adduser: Only root may add a user or group to the system.
dana@ubuntu:/home/ubuntu$ sudo adduser almahrouk
[sudo] password for dana:
dana is not in the sudoers file.  This incident will be reported.
dana@ubuntu:/home/ubuntu$
```

Permission Denied

- The "Permission Denied" error occurs when a user tries to perform an action that they don't have the **necessary permissions** for. This could **involve reading, writing, or executing a file or directory, or running a command that requires elevated privileges.**

- Common Scenarios:
  - Trying to access a file that is owned by another user and not world-readable.
  - Attempting to run a command that requires root privileges without using ` sudo ` .
  - Modifying system files without appropriate permissions.

Sudoers File

- located at `/etc/sudoers`, is used to control which users or groups could run commands as the root or another user via `sudo`.
- It also allows fine-grained control over which commands can be run.

- user ALL=(ALL:ALL) ALL

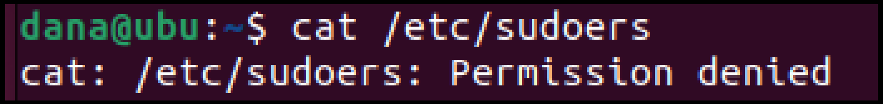
This allows `user` to run any command as any user on the system using `sudo`.

- user ALL=(ALL) /usr/bin/apt-get

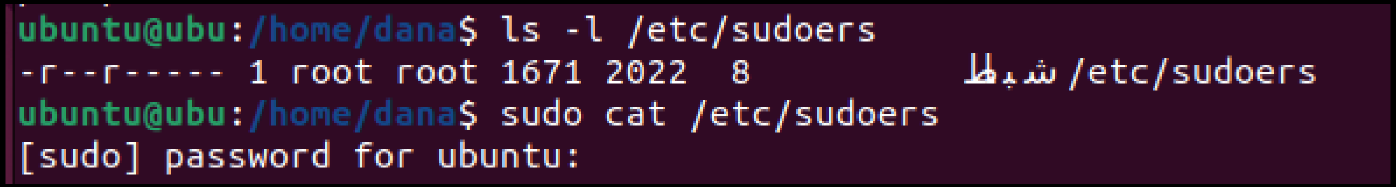
This allows `user` to run `apt-get` commands with `sudo`, but nothing else.

/etc/sudoers

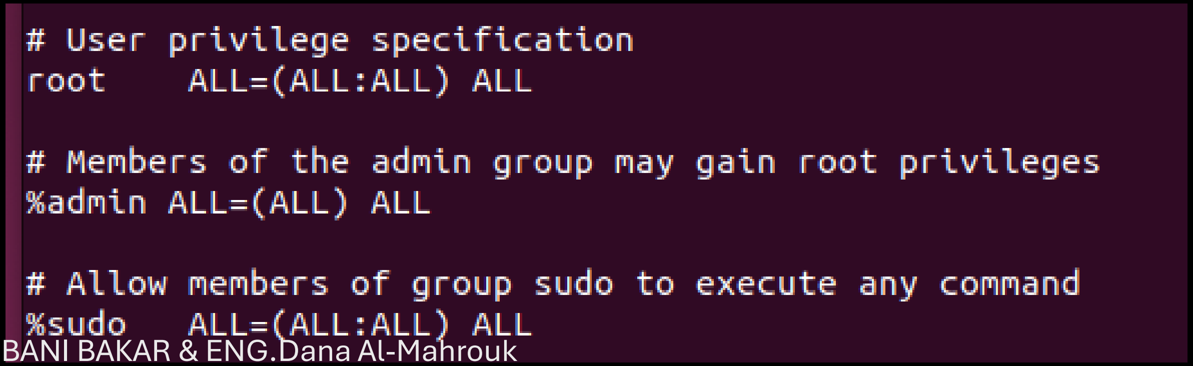
- is a configuration file that defines which users and groups have access to execute commands as the root user or another user. It also specifies the level of permissions granted and any specific conditions or restrictions on those permissions.



```
dana@ubu:~$ cat /etc/sudoers
cat: /etc/sudoers: Permission denied
```



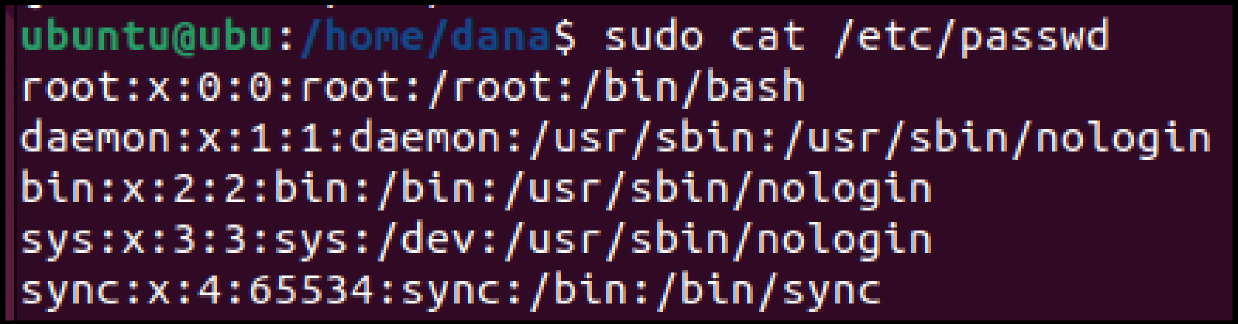
```
ubuntu@ubu:/home/dana$ ls -l /etc/sudoers
-r--r----- 1 root root 1671 2022  8        شباط /etc/sudoers
ubuntu@ubu:/home/dana$ sudo cat /etc/sudoers
[sudo] password for ubuntu:
```



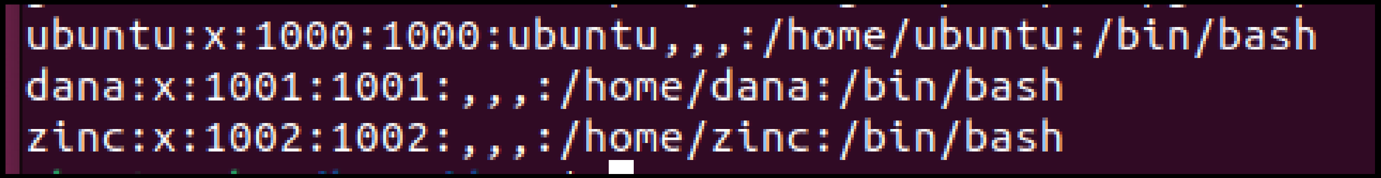
```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

/etc/passwd

- This file contains information about all the system's users.
- Each line represents one user and contains seven fields:
  - username
  - password
    - (x indicates it is stored in `/etc/shadow`)
  - User ID
  - Group ID
  - user information (GECOS)
  - home directory ~
  - shell



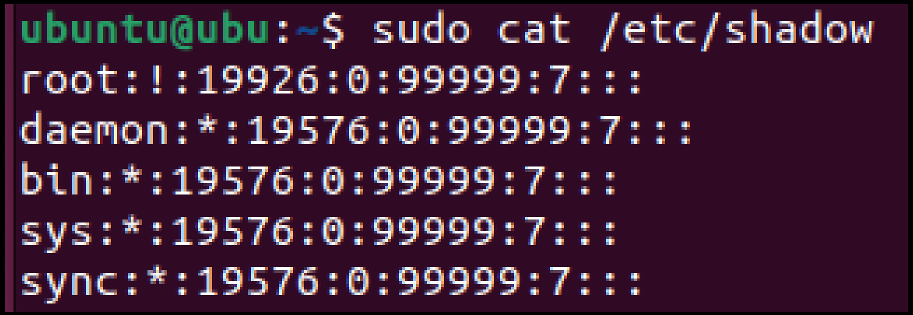
```
ubuntu@ubu:/home/dana$ sudo cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```



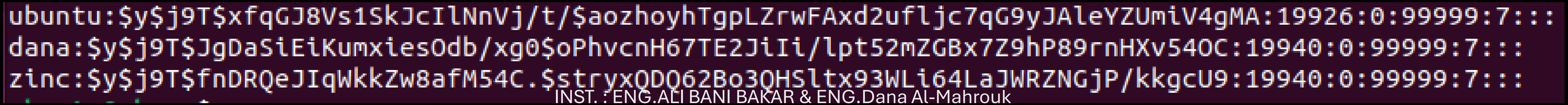
```
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
dana:x:1001:1001:,,,:/home/dana:/bin/bash
zinc:x:1002:1002:,,,:/home/zinc:/bin/bash
```

/etc/shadow

- It contains the **hashed passwords** for all user accounts on the system, along with additional information related to password aging and account policies.



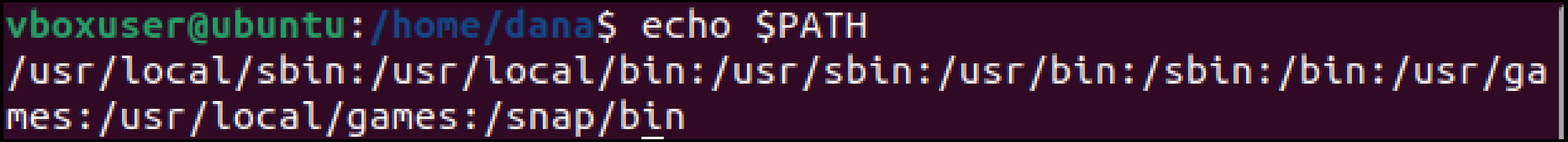
```
ubuntu@ubu:~$ sudo cat /etc/shadow
root:!:19926:0:99999:7:::
daemon:*:19576:0:99999:7:::
bin:*:19576:0:99999:7:::
sys:*:19576:0:99999:7:::
sync:*:19576:0:99999:7:::
```



```
ubuntu:$y$j9T$xfqGJ8Vs1SkJcIlNnVj/t/$aozhoyhTgpLZrwFAxd2ufljc7qG9yJAleYZUmiV4gMA:19926:0:99999:7:::
dana:$y$j9T$JgDaSiEiKumxiesOdb/xg0$oPhvcnH67TE2JiIi/lpt52mZGBx7Z9hP89rnHXv54OC:19940:0:99999:7:::
zinc:$y$j9T$fnDRQeJIqWkkZw8afM54C.$stryxQDQ62Bo3QHSltx93WLi64LaJWRZNGjP/kkgcU9:19940:0:99999:7:::
```

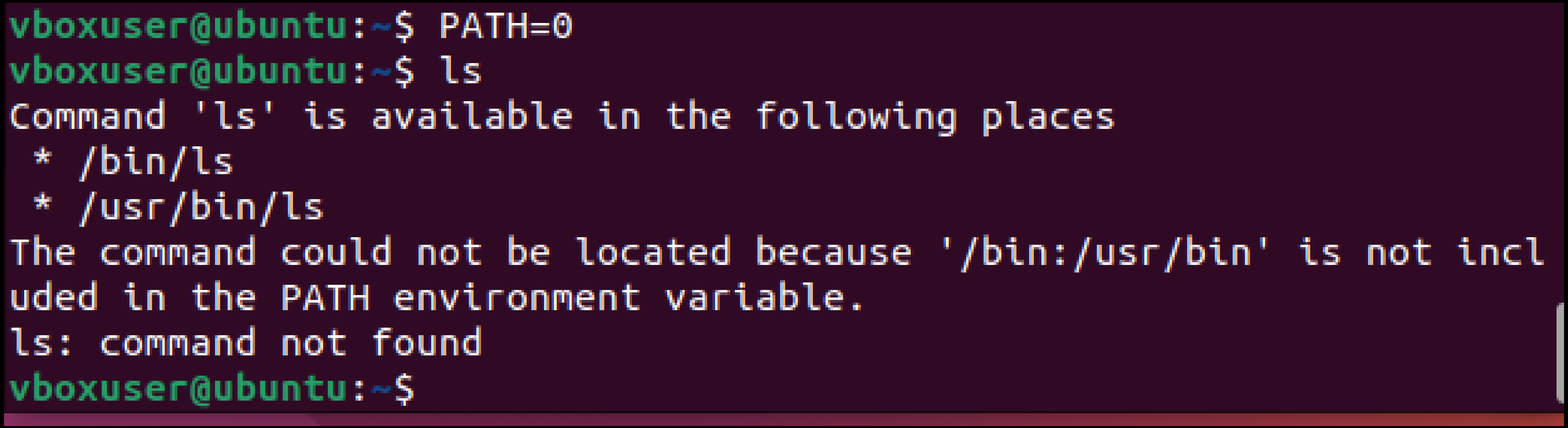
$PATH

- An environment variable that specifies the directories in which the system looks for executable files.
- shell searches through to find executable files when you type a command.
- Setting ` $PATH=0` is not a valid or meaningful operation and should be avoided. The ` PATH` variable is essential for the functioning of your shell

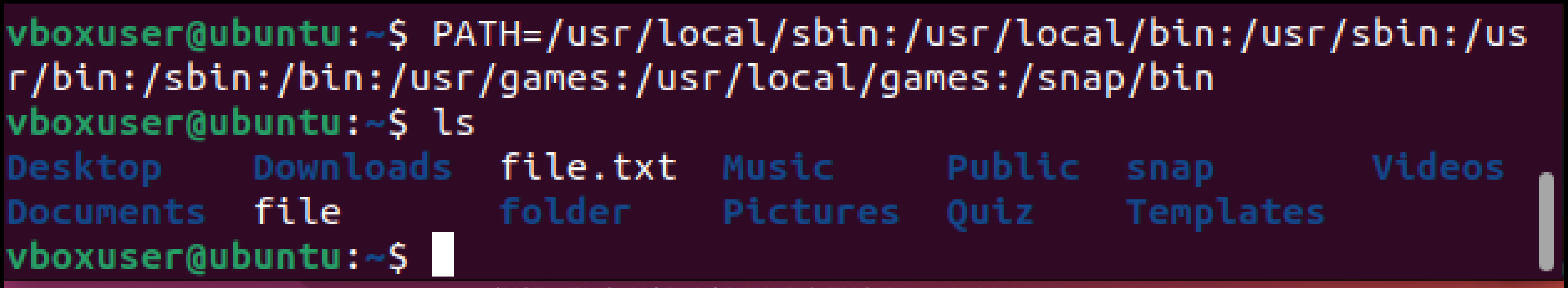


```
vboxuser@ubuntu:/home/dana$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/ga
mes:/usr/local/games:/snap/bin
```

$PATH



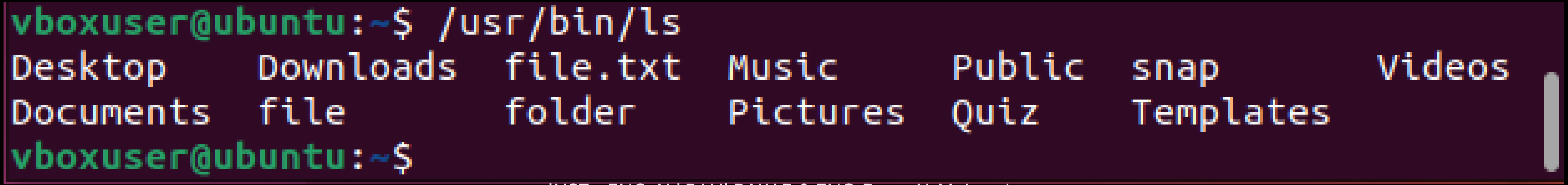
```
vboxuser@ubuntu:~$ PATH=0
vboxuser@ubuntu:~$ ls
Command 'ls' is available in the following places
 * /bin/ls
 * /usr/bin/ls
The command could not be located because '/bin:/usr/bin' is not incl
uded in the PATH environment variable.
ls: command not found
vboxuser@ubuntu:~$
```



```
vboxuser@ubuntu:~$ PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/us
r/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
vboxuser@ubuntu:~$ ls
Desktop    Downloads  file.txt  Music     Public  snap       Videos
Documents  file       folder    Pictures  Quiz    Templates
vboxuser@ubuntu:~$ 
```

/usr/bin

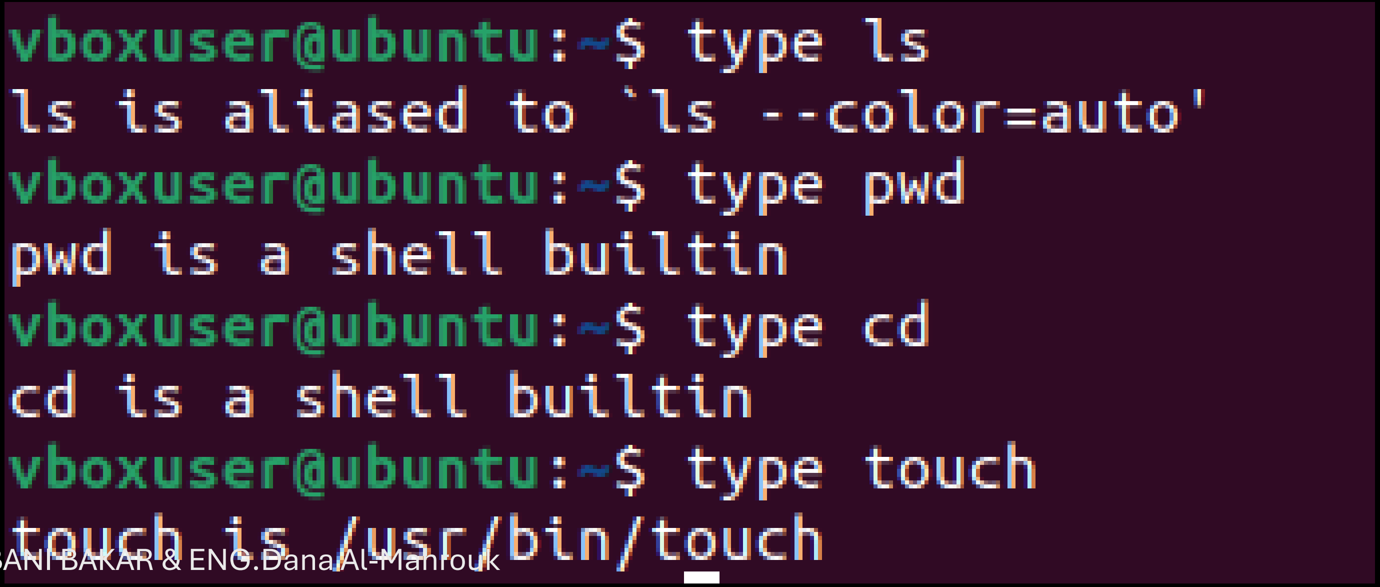
- The **` /usr/bin` directory** is where many **user-level executables are stored** on Unix and Linux systems.
- Executables in ` /usr/bin` are typically available to all users.
- Running ` /usr/bin/ls` specifically calls the ` ls` command **from this location**, ensuring that you're using the **standard version** of ` ls` that comes with your system, rather than a different version that might be located elsewhere in the ` PATH` .



```
vboxuser@ubuntu:~$ /usr/bin/ls
Desktop    Downloads  file.txt  Music     Public  snap       Videos
Documents  file       folder    Pictures  Quiz    Templates
vboxuser@ubuntu:~$
```

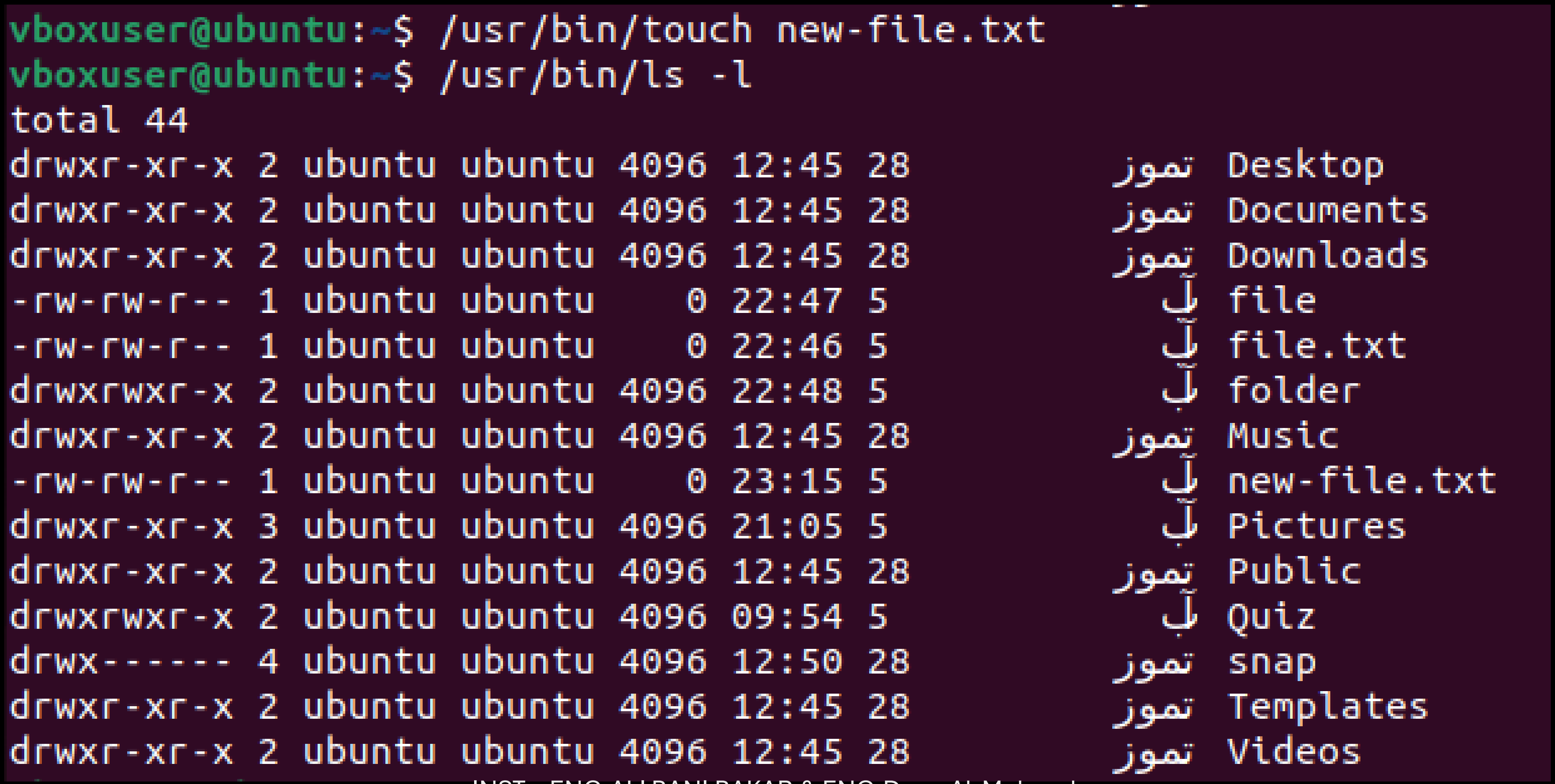
type

- determine how a command will be interpreted by the shell. It tells you whether the command is a built-in shell command, an alias, a function, or an external executable.
- \`type ls\` ➔ shows that \`ls\` is an **external** command located at \`/usr/bin/ls\`.
- \`type pwd\` ➔ shows that \`pwd\` is a **built-in** command within the **shell**.



```
vboxuser@ubuntu:~$ type ls
ls is aliased to `ls --color=auto'
vboxuser@ubuntu:~$ type pwd
pwd is a shell builtin
vboxuser@ubuntu:~$ type cd
cd is a shell builtin
vboxuser@ubuntu:~$ type touch
touch is /usr/bin/touch
```

Ex2



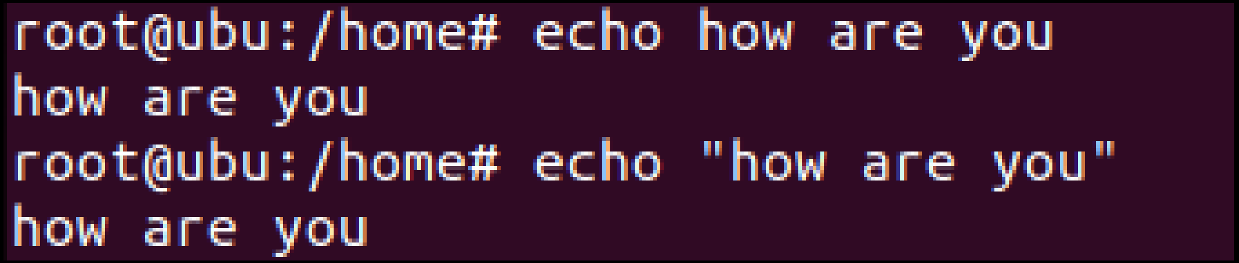
```
vboxuser@ubuntu:~$ /usr/bin/touch new-file.txt
vboxuser@ubuntu:~$ /usr/bin/ls -l
total 44
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28      تموز  Desktop
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28      تموز  Documents
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28      تموز  Downloads
-rw-rw-r-- 1 ubuntu ubuntu    0 22:47 5         آب  file
-rw-rw-r-- 1 ubuntu ubuntu    0 22:46 5         آب  file.txt
drwxrwxr-x 2 ubuntu ubuntu 4096 22:48 5         آب  folder
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28      تموز  Music
-rw-rw-r-- 1 ubuntu ubuntu    0 23:15 5         آب  new-file.txt
drwxr-xr-x 3 ubuntu ubuntu 4096 21:05 5         آب  Pictures
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28      تموز  Public
drwxrwxr-x 2 ubuntu ubuntu 4096 09:54 5         آب  Quiz
drwx------ 4 ubuntu ubuntu 4096 12:50 28      تموز  snap
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28      تموز  Templates
drwxr-xr-x 2 ubuntu ubuntu 4096 12:45 28      تموز  Videos
```

sudo (Superuser Do)

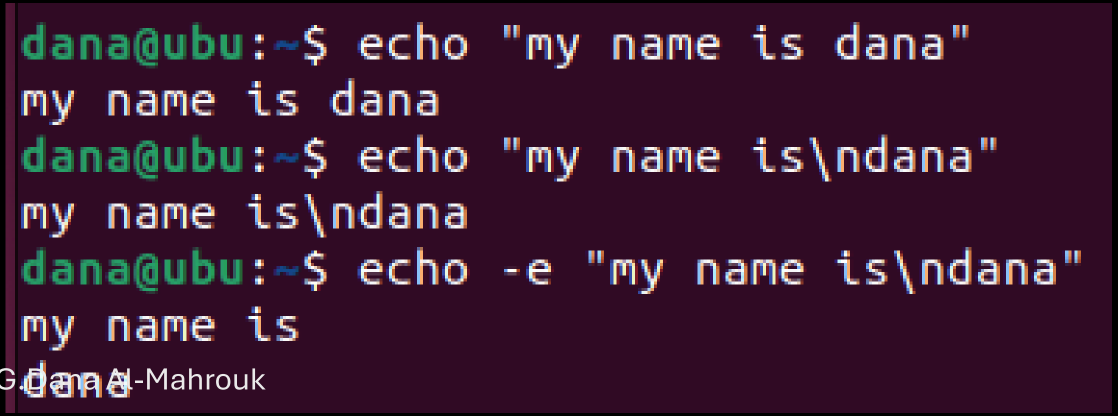
- Allows a permitted user to execute a command as the superuser or another user.
- sudo apt update
- sudo nano /etc/hosts
- sudo yum install httpd

echo

- display a line of text or a string to the terminal.
- -e → interpretation of backslash escapes
- \n → new line
- -n → prevents `echo` from adding a newline at the end of the output.



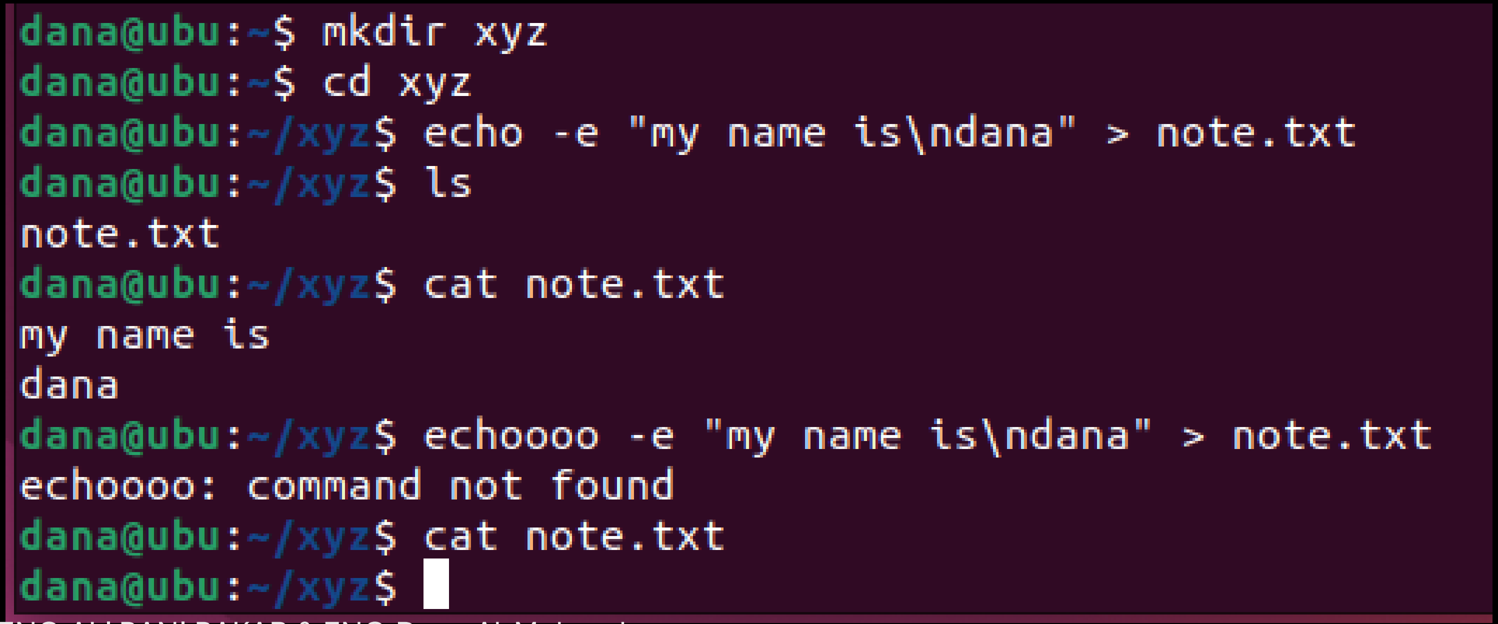
```
root@ubu:/home# echo how are you
how are you
root@ubu:/home# echo "how are you"
how are you
```



```
dana@ubu:~$ echo "my name is dana"
my name is dana
dana@ubu:~$ echo "my name is\ndana"
my name is\ndana
dana@ubu:~$ echo -e "my name is\ndana"
my name is
dana
```

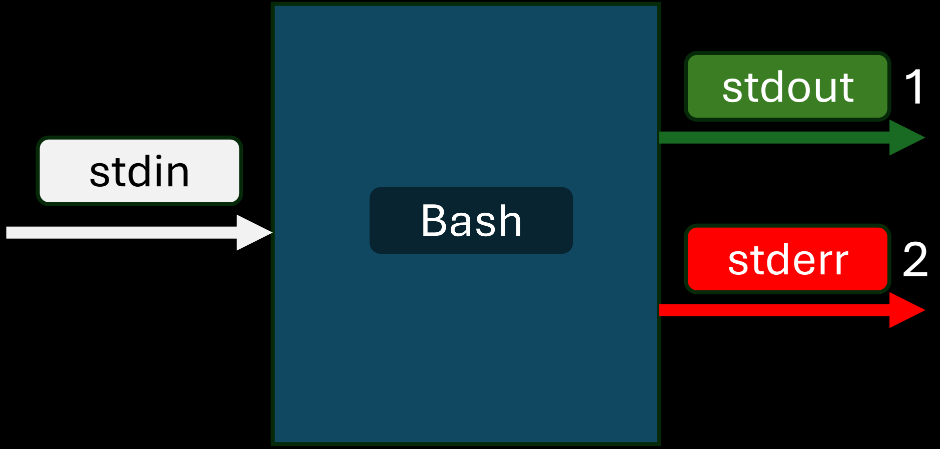
echo “…” > text.txt → Overwrite

- Writes the string “my name is\ndana" to ` note.txt` , creating the file if it doesn't exist or overwriting it if it does.



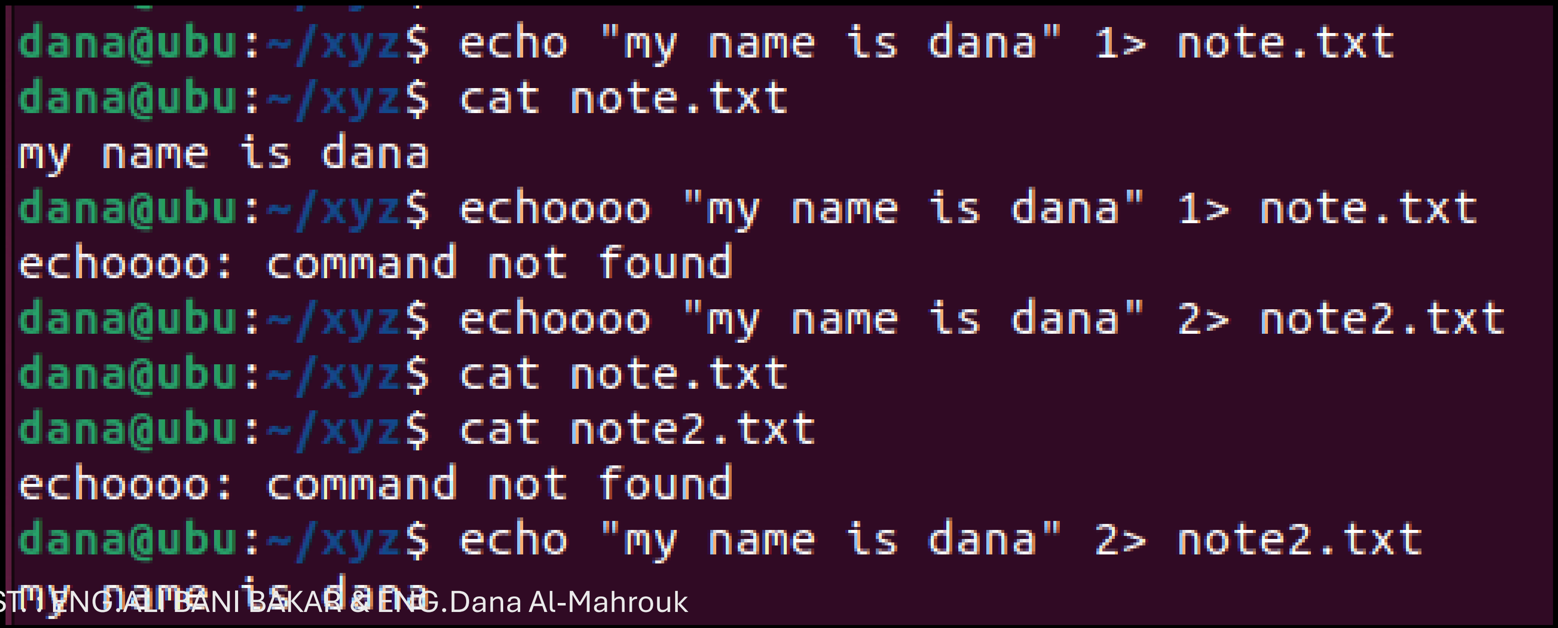
```
dana@ubu:~$ mkdir xyz
dana@ubu:~$ cd xyz
dana@ubu:~/xyz$ echo -e "my name is\ndana" > note.txt
dana@ubu:~/xyz$ ls
note.txt
dana@ubu:~/xyz$ cat note.txt
my name is
dana
dana@ubu:~/xyz$ echoooo -e "my name is\ndana" > note.txt
echoooo: command not found
dana@ubu:~/xyz$ cat note.txt
dana@ubu:~/xyz$ 
```

File 1 & 2



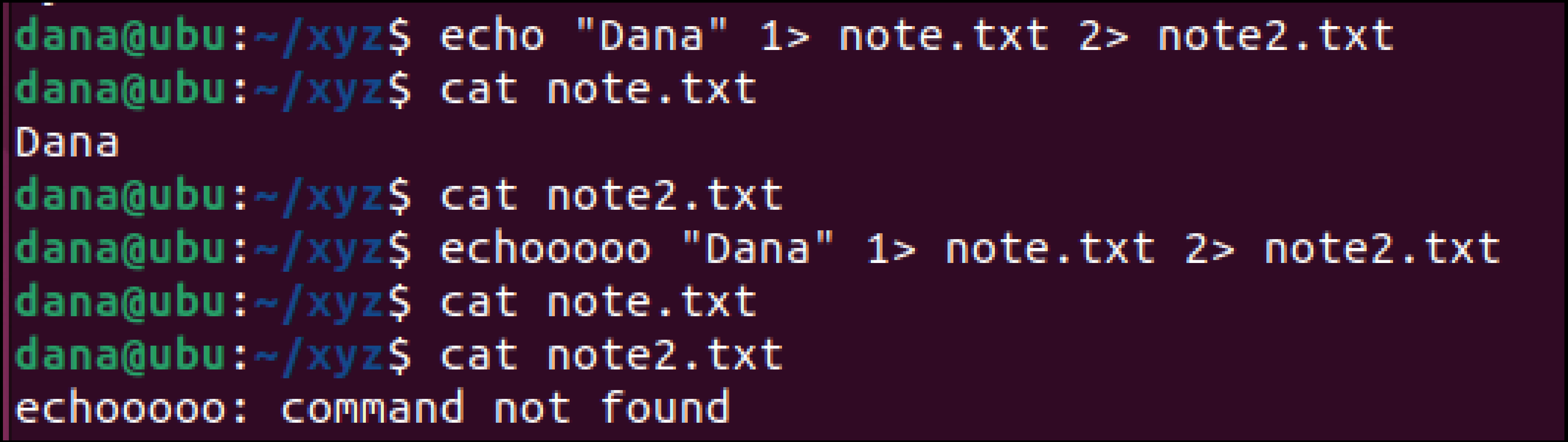
- **Standard Output (stdout):** File descriptor `1`. This is the default output stream where commands send their **regular output**.
- **Standard Error (stderr):** File descriptor `2`. This is the output stream where commands send their **error messages**.

- **1>**: Redirects **stdout** to a file.
- **2>**: Redirects **stderr** to a file.



```
dana@ubu:~/xyz$ echo "my name is dana" 1> note.txt
dana@ubu:~/xyz$ cat note.txt
my name is dana
dana@ubu:~/xyz$ echoooo "my name is dana" 1> note.txt
echoooo: command not found
dana@ubu:~/xyz$ echoooo "my name is dana" 2> note2.txt
dana@ubu:~/xyz$ cat note.txt
dana@ubu:~/xyz$ cat note2.txt
echoooo: command not found
dana@ubu:~/xyz$ echo "my name is dana" 2> note2.txt
my name is dana
```

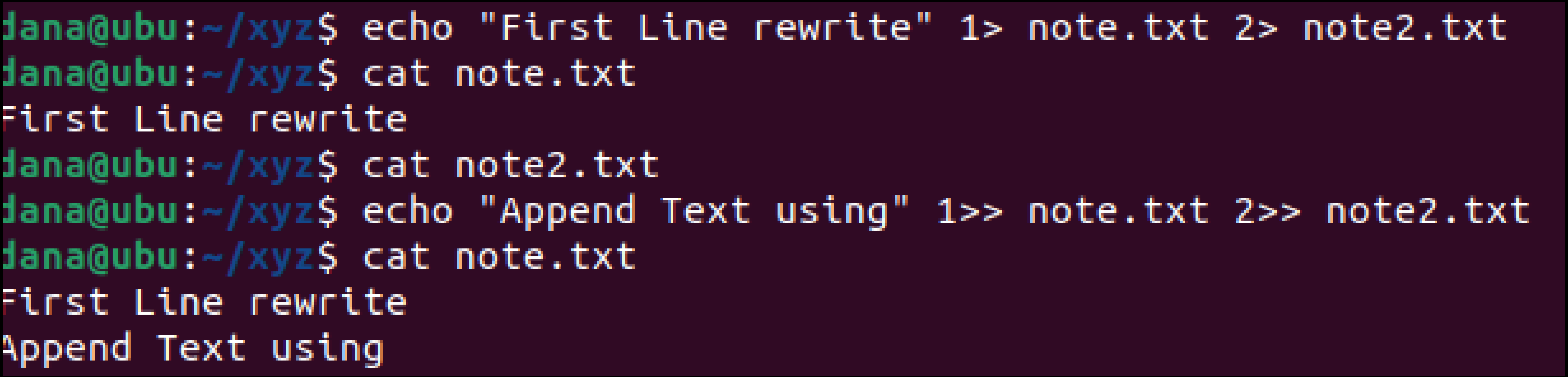
Solution



```
dana@ubu:~/xyz$ echo "Dana" 1> note.txt 2> note2.txt
dana@ubu:~/xyz$ cat note.txt
Dana
dana@ubu:~/xyz$ cat note2.txt
dana@ubu:~/xyz$ echoooooo "Dana" 1> note.txt 2> note2.txt
dana@ubu:~/xyz$ cat note.txt
dana@ubu:~/xyz$ cat note2.txt
echoooooo: command not found
```

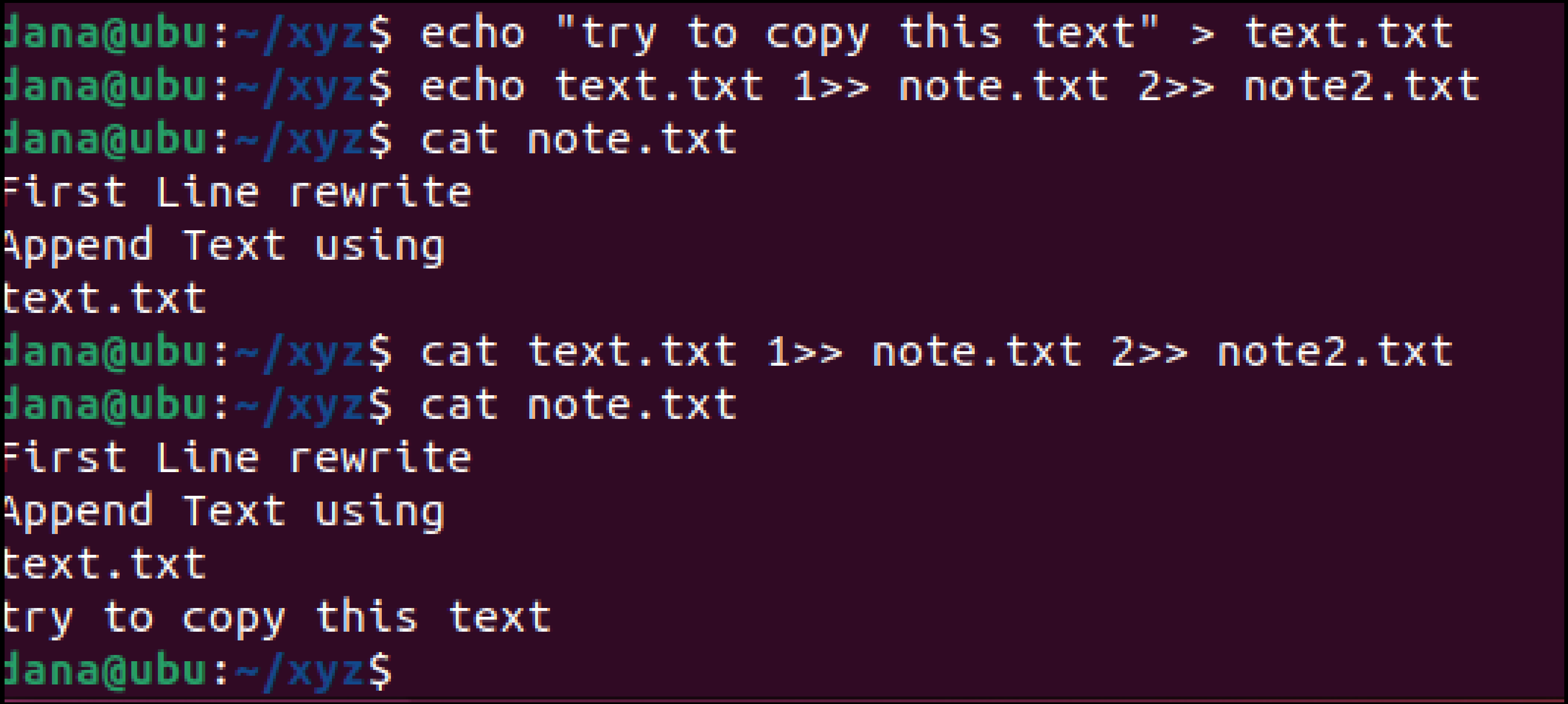
echo “…” >> text.txt ➔ Append

- Appends "This is another line." to ` file.txt ` without overwriting the existing content.



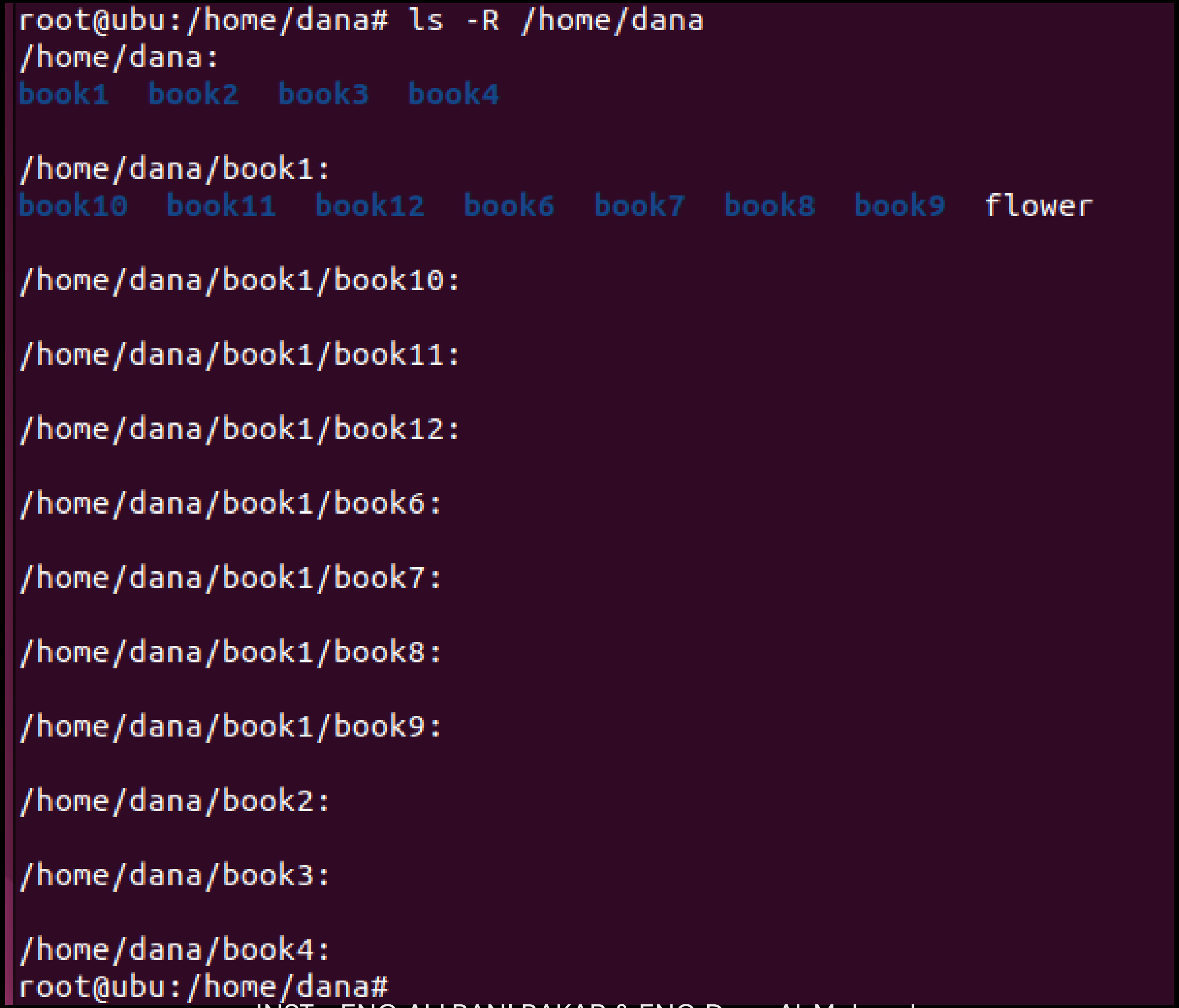
```
dana@ubu:~/xyz$ echo "First Line rewrite" 1> note.txt 2> note2.txt
dana@ubu:~/xyz$ cat note.txt
First Line rewrite
dana@ubu:~/xyz$ cat note2.txt
dana@ubu:~/xyz$ echo "Append Text using" 1>> note.txt 2>> note2.txt
dana@ubu:~/xyz$ cat note.txt
First Line rewrite
Append Text using
```

Append File in File



```
dana@ubu:~/xyz$ echo "try to copy this text" > text.txt
dana@ubu:~/xyz$ echo text.txt 1>> note.txt 2>> note2.txt
dana@ubu:~/xyz$ cat note.txt
First Line rewrite
Append Text using
text.txt
dana@ubu:~/xyz$ cat text.txt 1>> note.txt 2>> note2.txt
dana@ubu:~/xyz$ cat note.txt
First Line rewrite
Append Text using
text.txt
try to copy this text
dana@ubu:~/xyz$
```

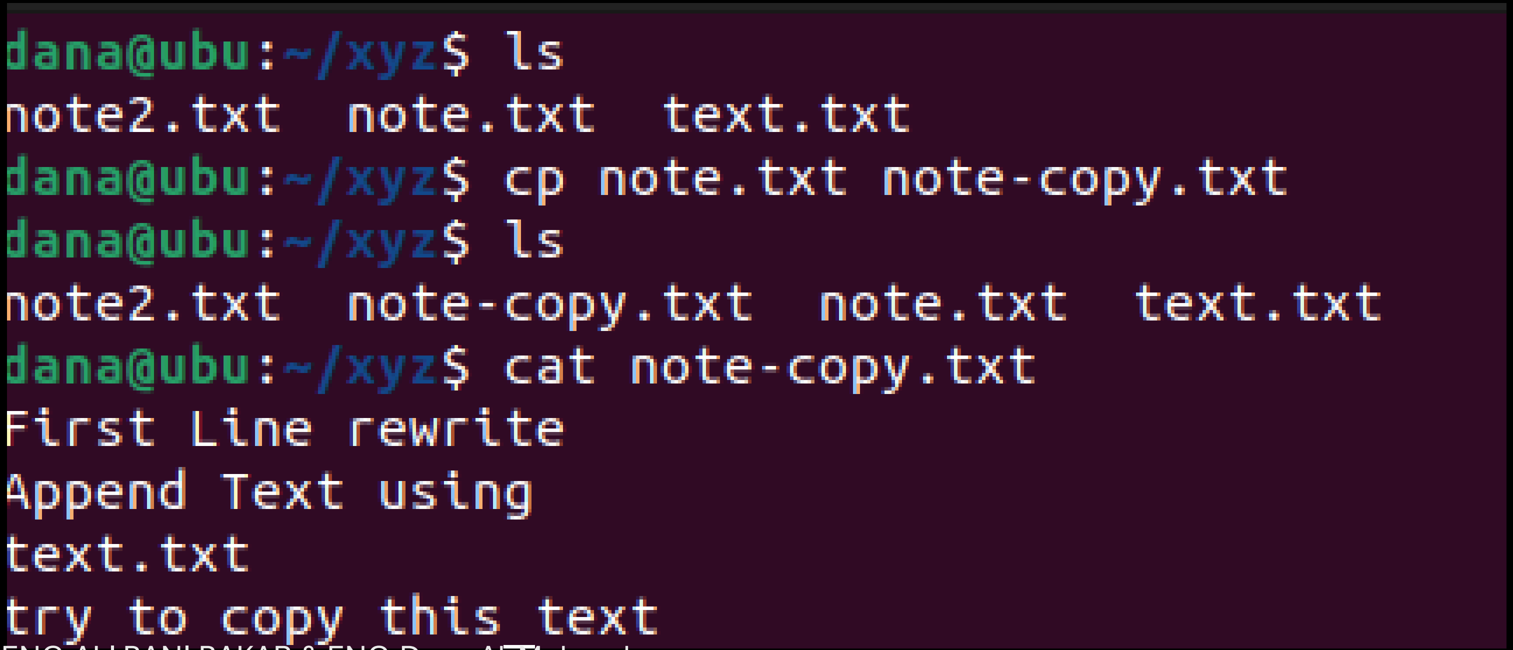
Ls -R



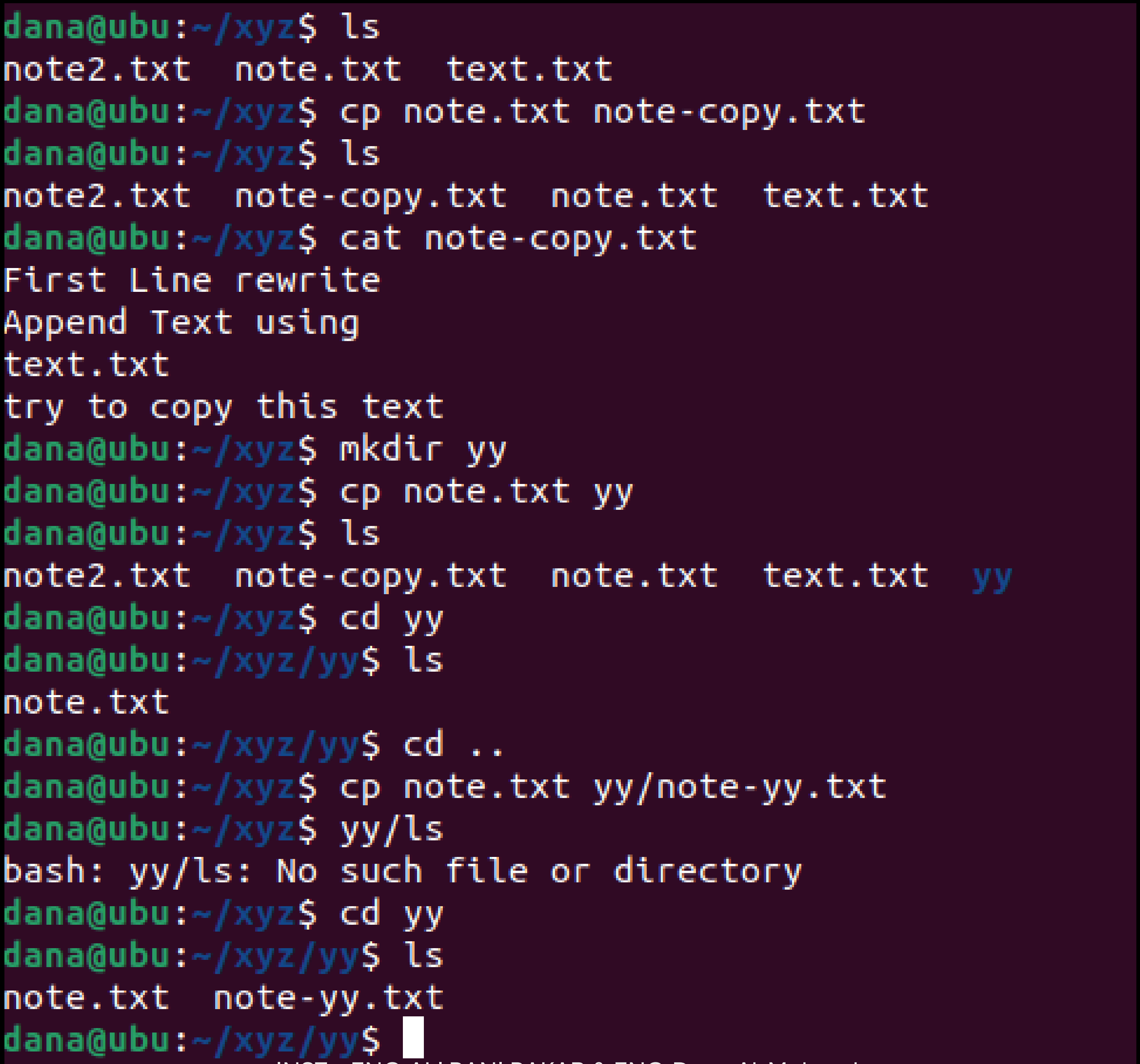
```
root@ubu:/home/dana# ls -R /home/dana
/home/dana:
book1  book2  book3  book4

/home/dana/book1:
book10  book11  book12  book6  book7  book8  book9  flower

/home/dana/book1/book10:

/home/dana/book1/book11:

/home/dana/book1/book12:

/home/dana/book1/book6:

/home/dana/book1/book7:

/home/dana/book1/book8:

/home/dana/book1/book9:

/home/dana/book2:

/home/dana/book3:

/home/dana/book4:
root@ubu:/home/dana#
```

cp (Copy)

- Copies files or directories from one location to another.



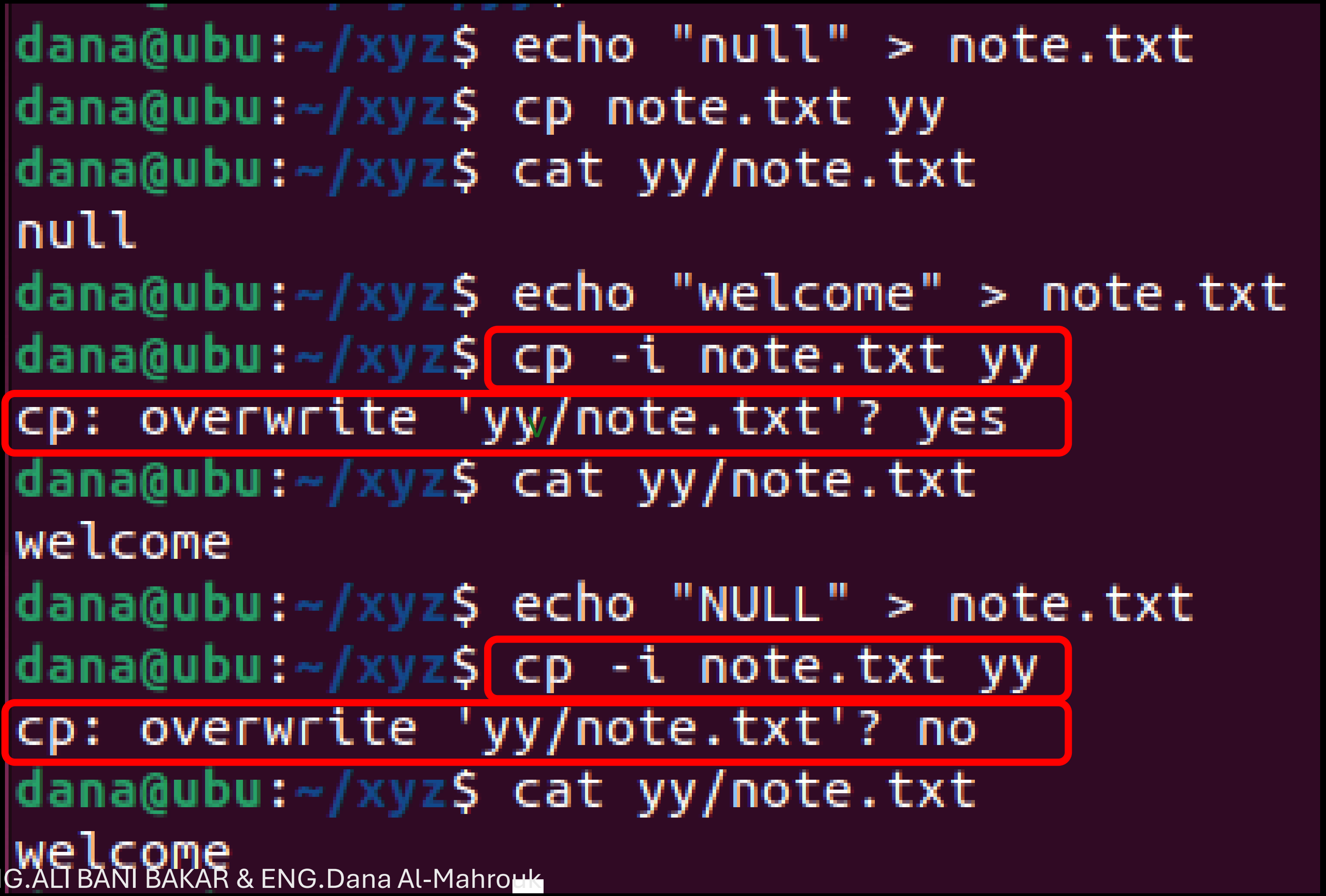
```
dana@ubu:~/xyz$ ls
note2.txt  note.txt  text.txt
dana@ubu:~/xyz$ cp note.txt note-copy.txt
dana@ubu:~/xyz$ ls
note2.txt  note-copy.txt  note.txt  text.txt
dana@ubu:~/xyz$ cat note-copy.txt
First Line rewrite
Append Text using
text.txt
try to copy this text
```



```
dana@ubu:~/xyz$ ls
note2.txt  note.txt  text.txt
dana@ubu:~/xyz$ cp note.txt note-copy.txt
dana@ubu:~/xyz$ ls
note2.txt  note-copy.txt  note.txt  text.txt
dana@ubu:~/xyz$ cat note-copy.txt
First Line rewrite
Append Text using
text.txt
try to copy this text
dana@ubu:~/xyz$ mkdir yy
dana@ubu:~/xyz$ cp note.txt yy
dana@ubu:~/xyz$ ls
note2.txt  note-copy.txt  note.txt  text.txt  yy
dana@ubu:~/xyz$ cd yy
dana@ubu:~/xyz/yy$ ls
note.txt
dana@ubu:~/xyz/yy$ cd ..
dana@ubu:~/xyz$ cp note.txt yy/note-yy.txt
dana@ubu:~/xyz$ yy/ls
bash: yy/ls: No such file or directory
dana@ubu:~/xyz$ cd yy
dana@ubu:~/xyz/yy$ ls
note.txt  note-yy.txt
dana@ubu:~/xyz/yy$ 
```

cp -i

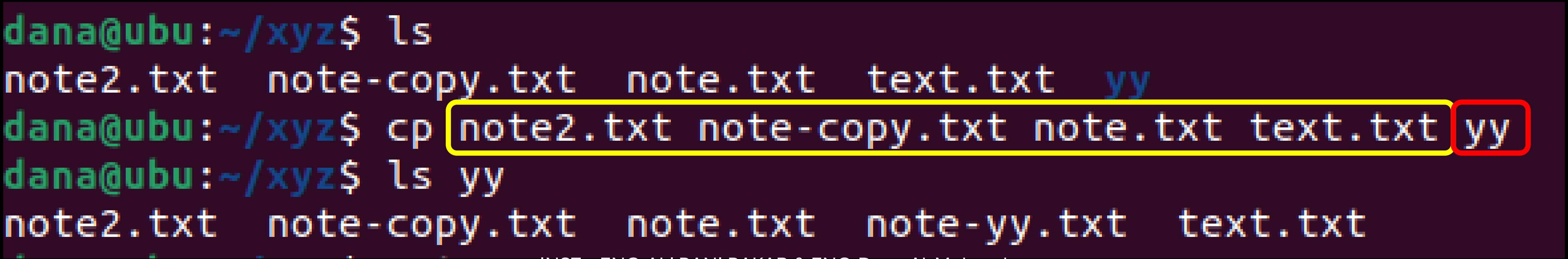
- `-i` (interactive): Prompts the user for confirmation before overwriting a file.



```
dana@ubu:~/xyz$ echo "null" > note.txt
dana@ubu:~/xyz$ cp note.txt yy
dana@ubu:~/xyz$ cat yy/note.txt
null
dana@ubu:~/xyz$ echo "welcome" > note.txt
dana@ubu:~/xyz$ cp -i note.txt yy
cp: overwrite 'yy/note.txt'? yes
dana@ubu:~/xyz$ cat yy/note.txt
welcome
dana@ubu:~/xyz$ echo "NULL" > note.txt
dana@ubu:~/xyz$ cp -i note.txt yy
cp: overwrite 'yy/note.txt'? no
dana@ubu:~/xyz$ cat yy/note.txt
welcome
```

Copy multiple input to one directory

- cp <...files|directory> <dest directory>



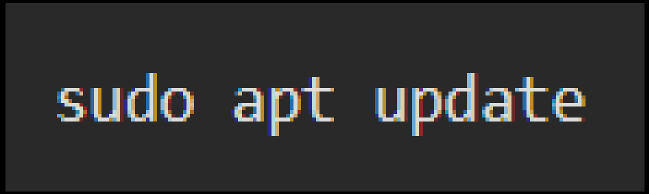
```
dana@ubu:~/xyz$ ls
note2.txt  note-copy.txt  note.txt  text.txt  yy
dana@ubu:~/xyz$ cp note2.txt note-copy.txt note.txt text.txt yy
dana@ubu:~/xyz$ ls yy
note2.txt  note-copy.txt  note.txt  note-yy.txt  text.txt
```

Package management

- **Update:**

is used to refresh the package index. This means it updates the local list of available packages and their versions from the repositories configured on your system.

This command does not install or upgrade any packages. It simply ensures that your package manager has the most current information.

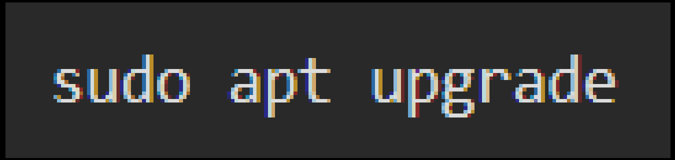


```
sudo apt update
```

- **Upgrade:**

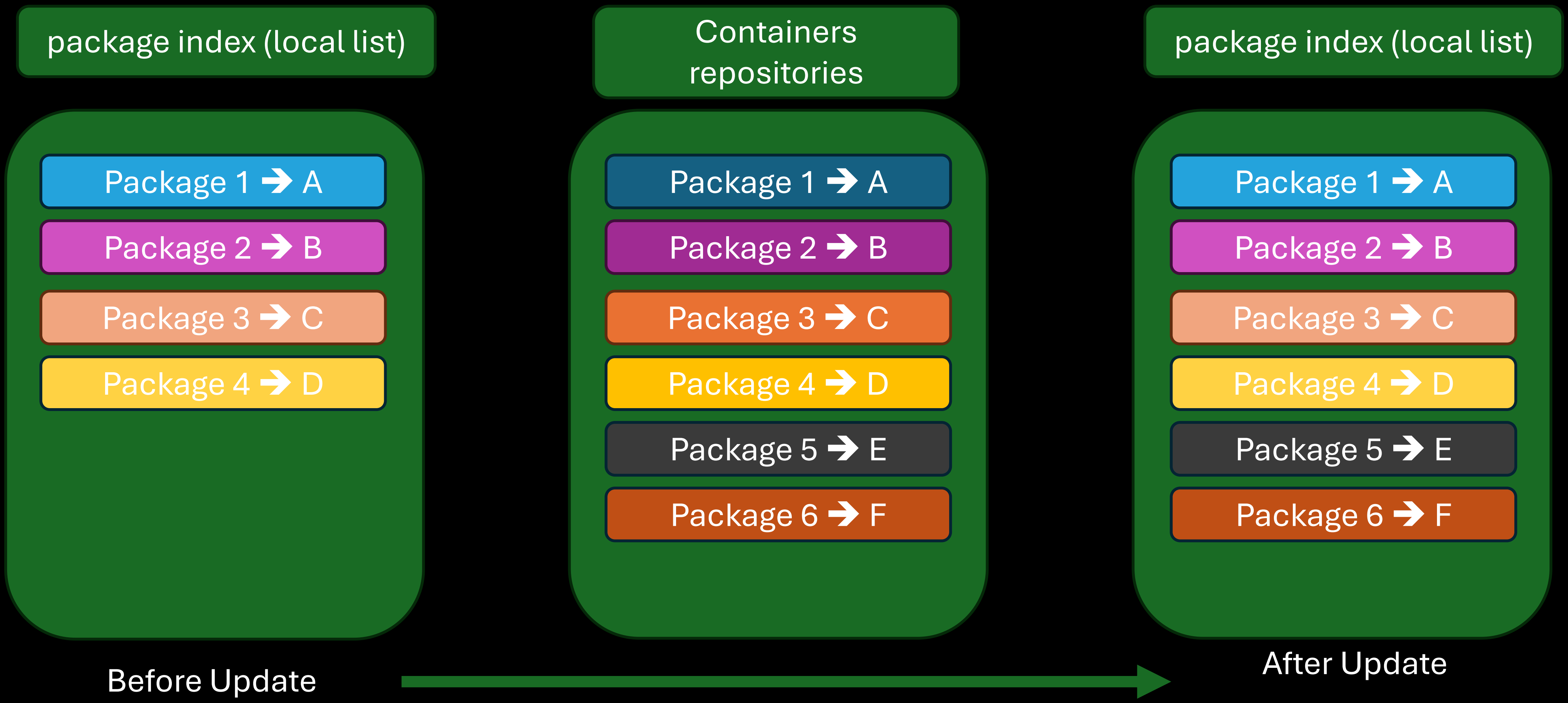
is used to install the newest versions of all packages currently installed on the system, based on the updated package index.

This command will upgrade all packages that can be upgraded without removing any installed packages or installing new ones. If a package upgrade requires additional dependencies or package removal, it will not be done with this command.



```
sudo apt upgrade
```

Update



package index (local list)

- Package 1 → A
- Package 2 → B
- Package 3 → C
- Package 4 → D

Before Update

Containers repositories

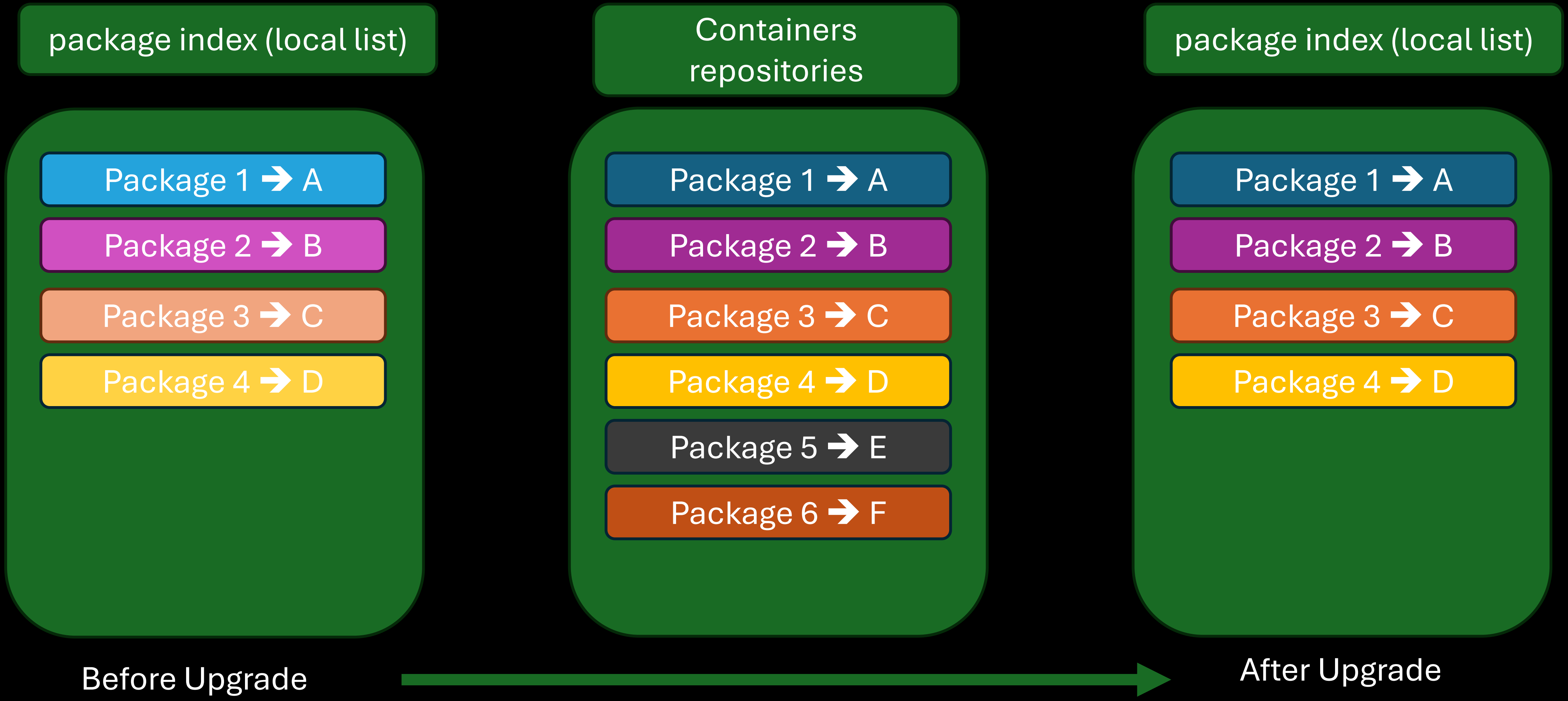
- Package 1 → A
- Package 2 → B
- Package 3 → C
- Package 4 → D
- Package 5 → E
- Package 6 → F

package index (local list)

- Package 1 → A
- Package 2 → B
- Package 3 → C
- Package 4 → D
- Package 5 → E
- Package 6 → F

After Update

Upgrade


package index (local list)
Package 1 → A
Package 2 → B
Package 3 → C
Package 4 → D
Containers repositories
Package 1 → A
Package 2 → B
Package 3 → C
Package 4 → D
Package 5 → E
Package 6 → F
package index (local list)
Package 1 → A
Package 2 → B
Package 3 → C
Package 4 → D
Before Upgrade
After Upgrade

Account

- **Service-Level:**

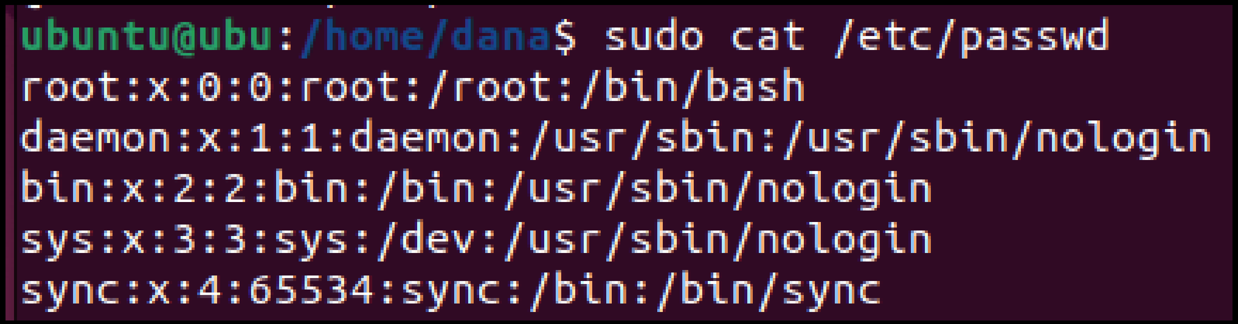
- services run under specific user accounts with particular permissions. These permissions define what the service can and cannot do on the system.
- **Least Privilege:** services run with the least privilege necessary to perform their function.
- **Exploiting a Service:** If a hacker **exploits a vulnerability in a service**, they typically **inherit** the permissions of the user under which that service is running.

- **User-Level:**

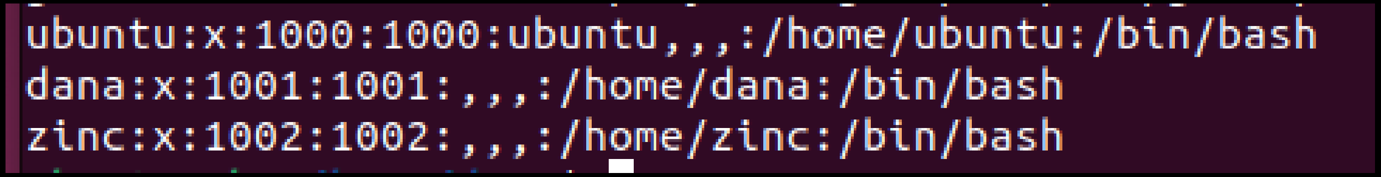
- **Logged-In User:** If a service is compromised, the hacker does not automatically gain the permissions of any other users who are logged in. They are restricted to the permissions of the service itself.
- **Privilege Escalation:** After gaining initial access, hackers may attempt to **escalate their privileges** to gain broader access. This could involve exploiting other vulnerabilities to move from the permissions of a low-privileged service to a higher-privileged user, like ` root ` .

/etc/passwd

- This file contains information about all the system's users.
- Each line represents one user and contains seven fields:
  - username
  - password
    - (x indicates it is stored in `/etc/shadow`)
  - User ID
  - Group ID
  - user information (GECOS)
  - home directory ~
  - shell



```
ubuntu@ubu:/home/dana$ sudo cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```



```
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
dana:x:1001:1001:,,,:/home/dana:/bin/bash
zinc:x:1002:1002:,,,:/home/zinc:/bin/bash
```

- When a new user is created, both a User ID (UID) and a Group ID (GID) are generated.

- The user is assigned a primary group, usually with the same name and ID as the user,

File permission

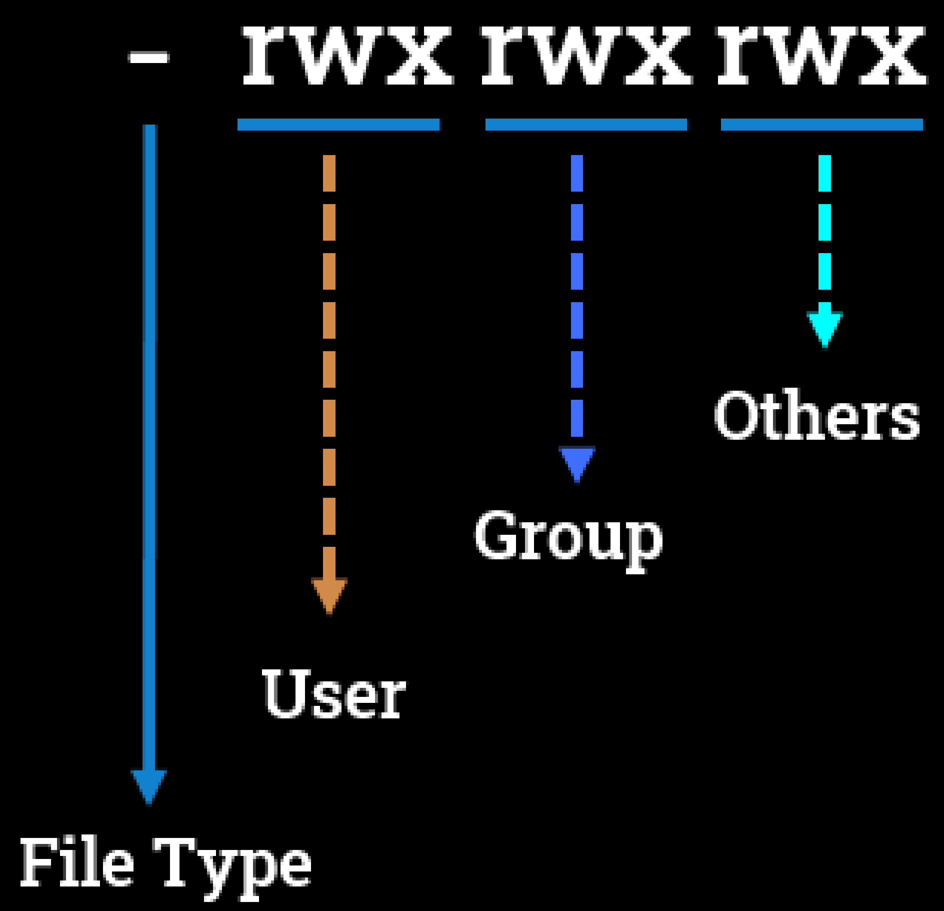
- File permissions in Linux control who can read, write, or execute a file. Understanding these permissions is key to managing access to files and directories.
- Everything is a file
- Permission type:
  - Read
  - Write
  - Execute

  - Allow
  - Deny

	File	Directory
Read	Read Cat | nano | vim	List contents
Write	Modify Cut | past | edit	altering the contents Create | Remove
Execute	running as a program Script	enter it Cd

Permission Groups

1. Owner (User): The user who owns the file.
2. Group: Other users who are in the same group as the file.
3. Others: All other users.



- rwx rwx rwx

Others

Group

User

File Type

r = Readable

w = Writeable

x = Executable

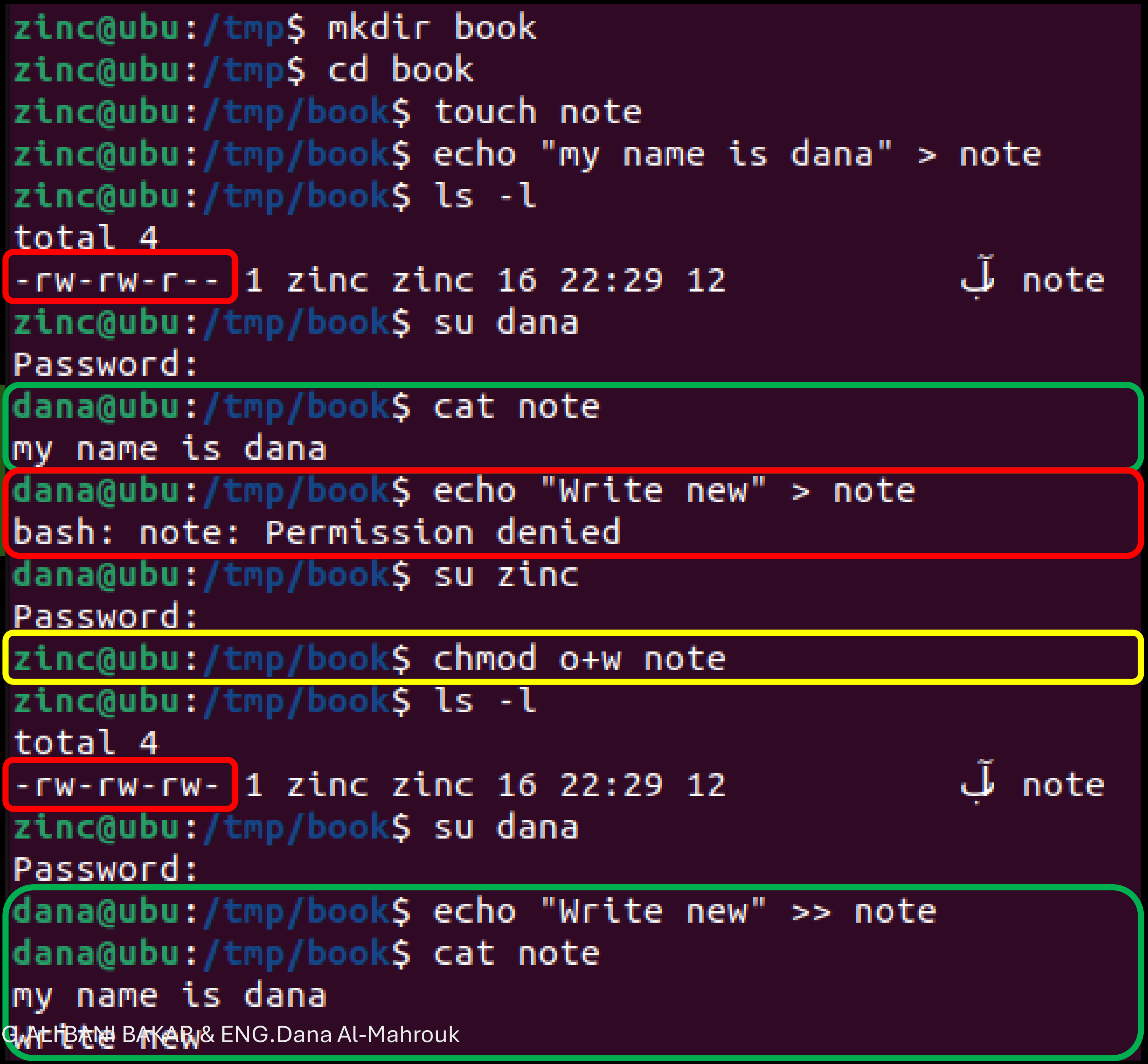
- = Denied

d = Directory

- = Regular File

l = Symbolic link

chmod



```
zinc@ubu:/tmp$ mkdir book
zinc@ubu:/tmp$ cd book
zinc@ubu:/tmp/book$ touch note
zinc@ubu:/tmp/book$ echo "my name is dana" > note
zinc@ubu:/tmp/book$ ls -l
total 4
-rw-rw-r-- 1 zinc zinc 16 22:29 12          آب  note
zinc@ubu:/tmp/book$ su dana
Password:
dana@ubu:/tmp/book$ cat note
my name is dana
dana@ubu:/tmp/book$ echo "Write new" > note
bash: note: Permission denied
dana@ubu:/tmp/book$ su zinc
Password:
zinc@ubu:/tmp/book$ chmod o+w note
zinc@ubu:/tmp/book$ ls -l
total 4
-rw-rw-rw- 1 zinc zinc 16 22:29 12          آب  note
zinc@ubu:/tmp/book$ su dana
Password:
dana@ubu:/tmp/book$ echo "Write new" >> note
dana@ubu:/tmp/book$ cat note
my name is dana
Write new
```

Other Permission

Can Read

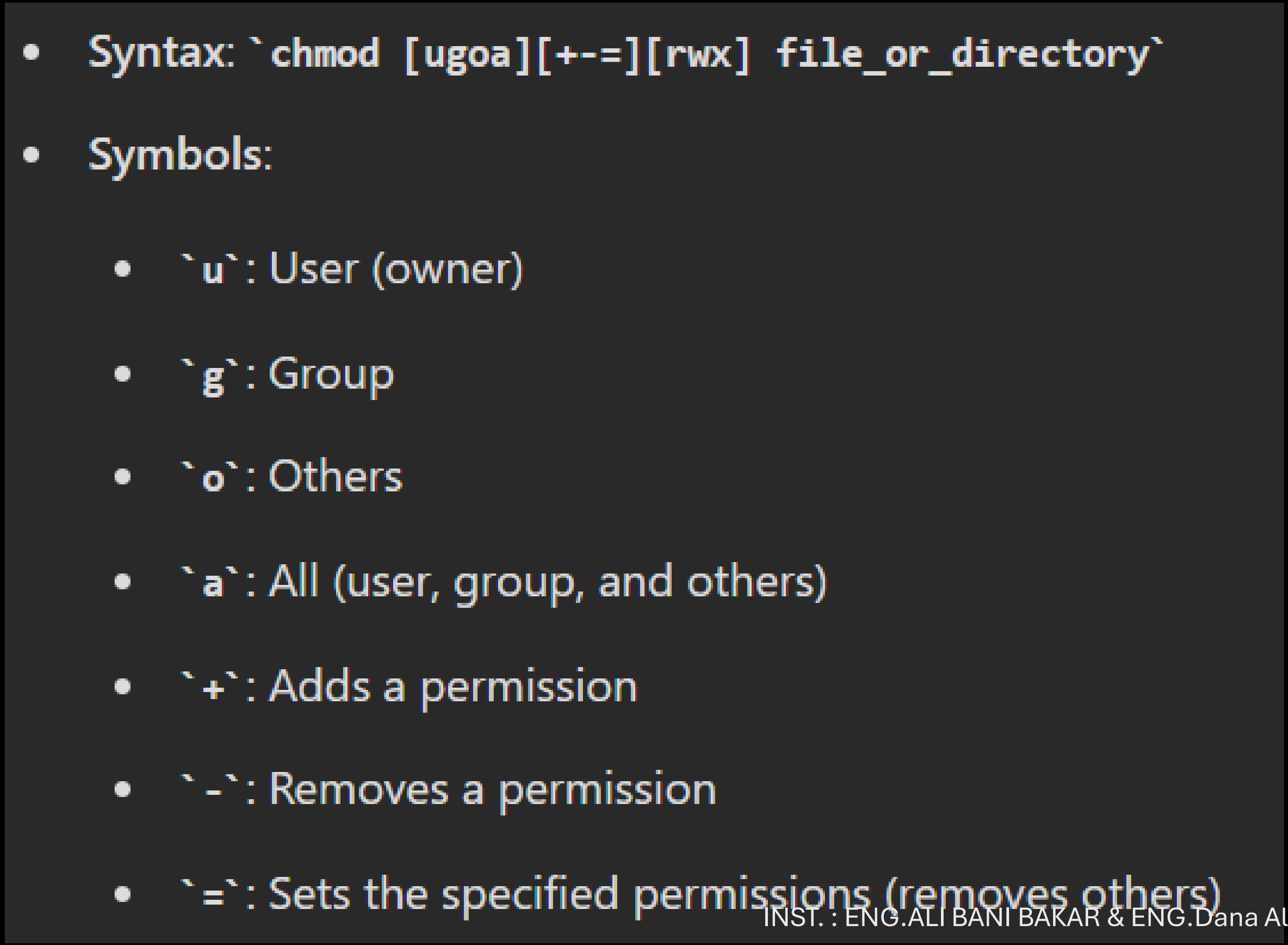
Can not Write

Other Permission

Can Write

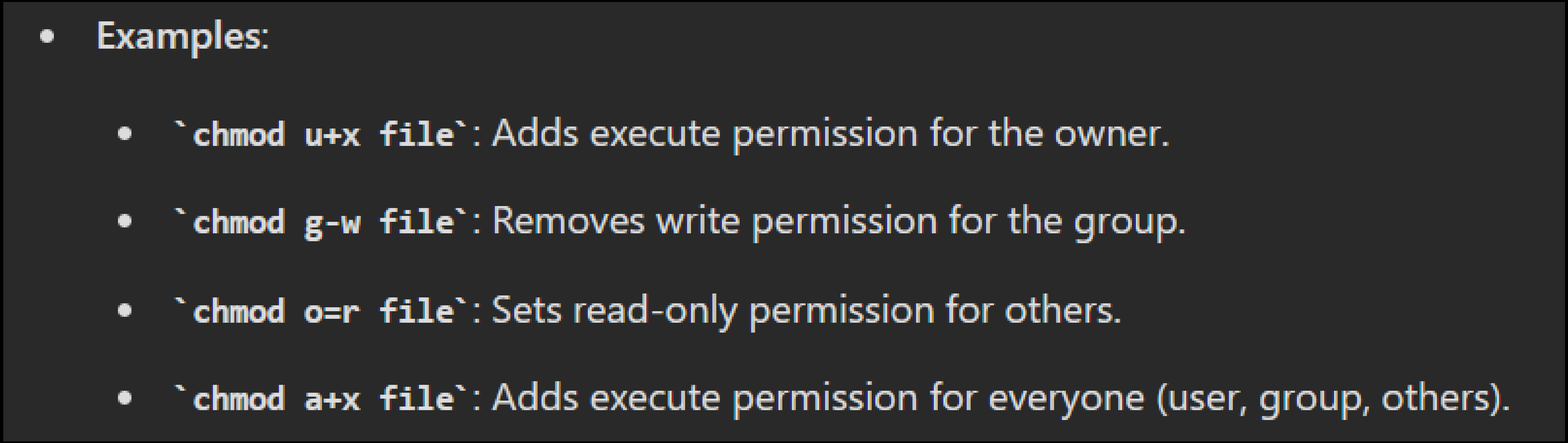
Can Read

chmod ➔ change mode

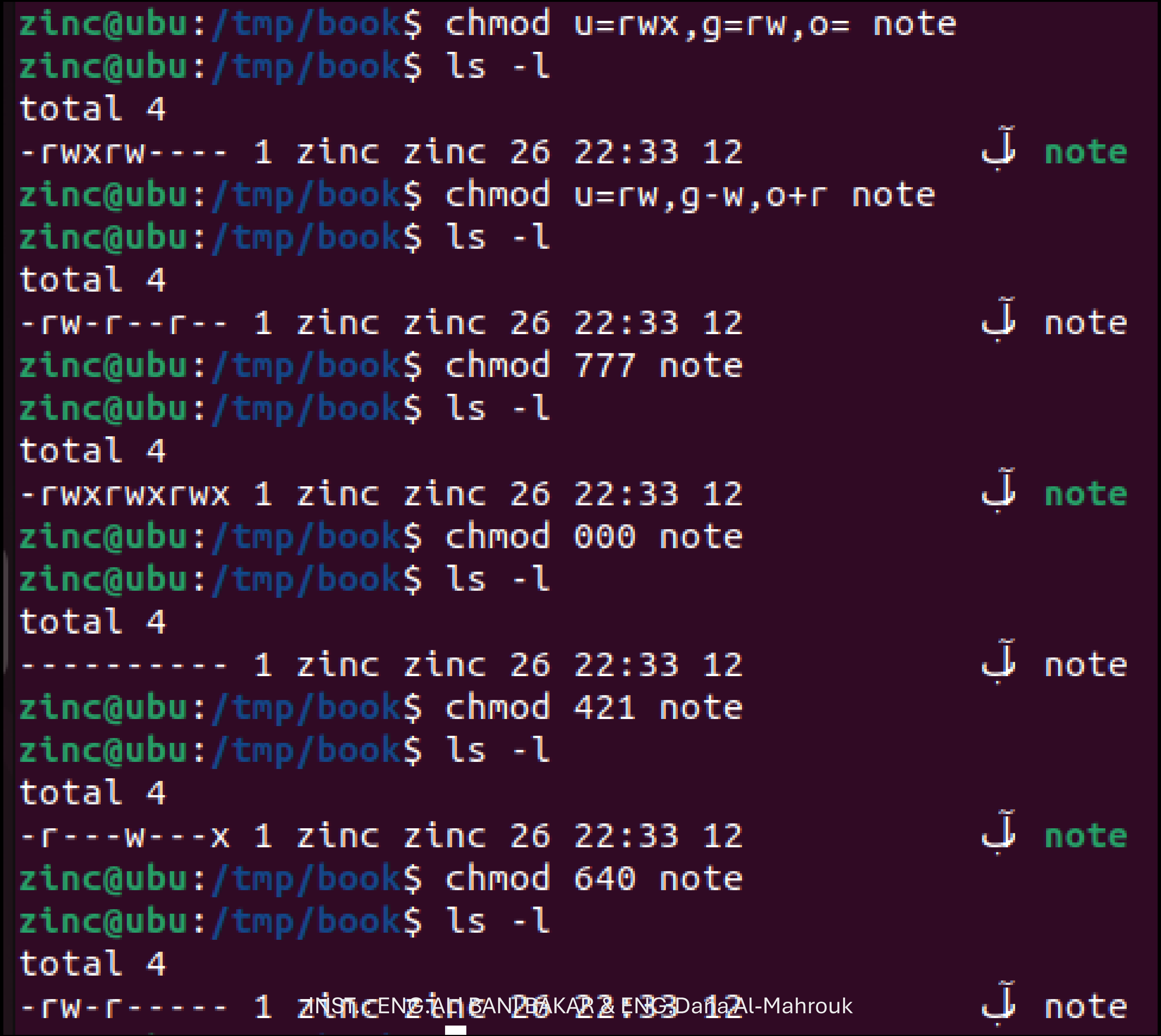


- Syntax: `chmod [ugoa][+-=][rwx] file_or_directory`
- Symbols:
  - `u`: User (owner)
  - `g`: Group
  - `o`: Others
  - `a`: All (user, group, and others)
  - `+`: Adds a permission
  - `-`: Removes a permission
  - `=`: Sets the specified permissions (removes others)

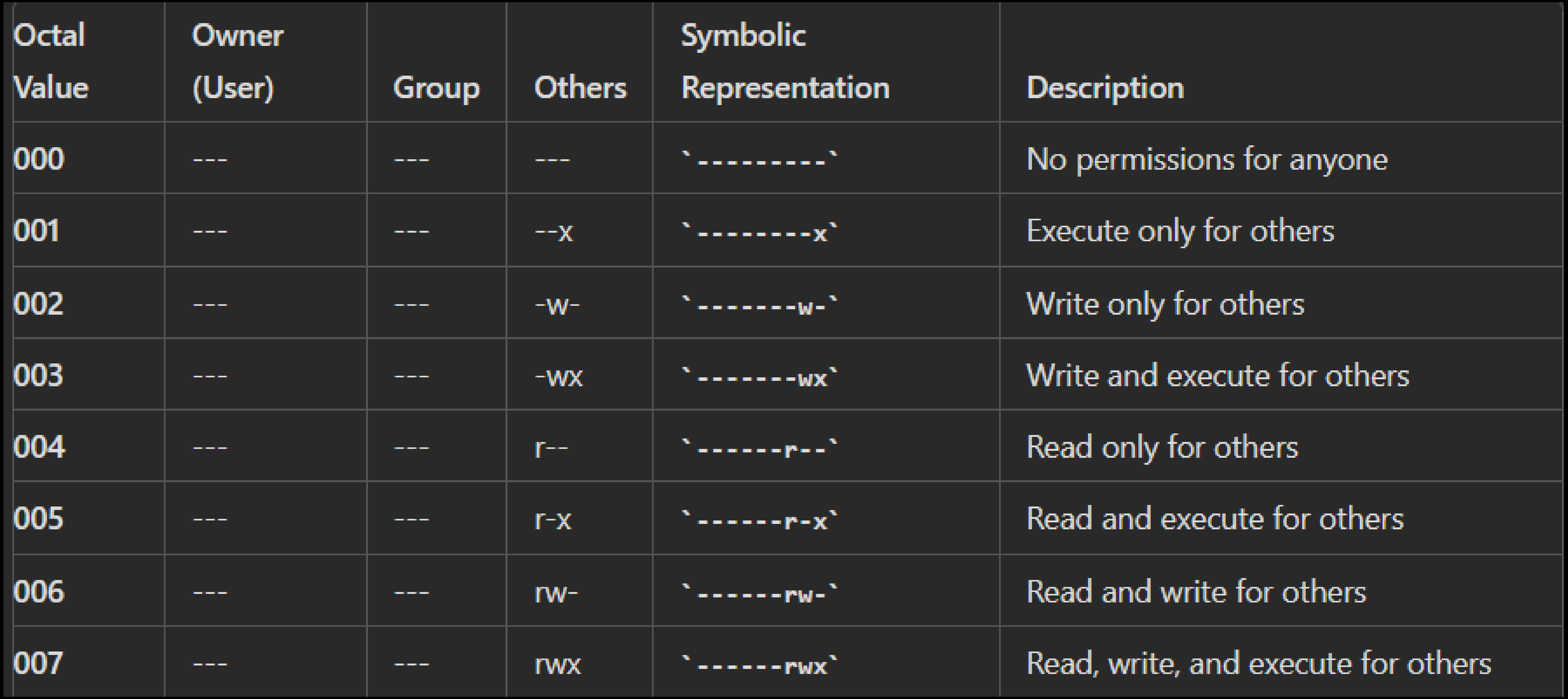
Example



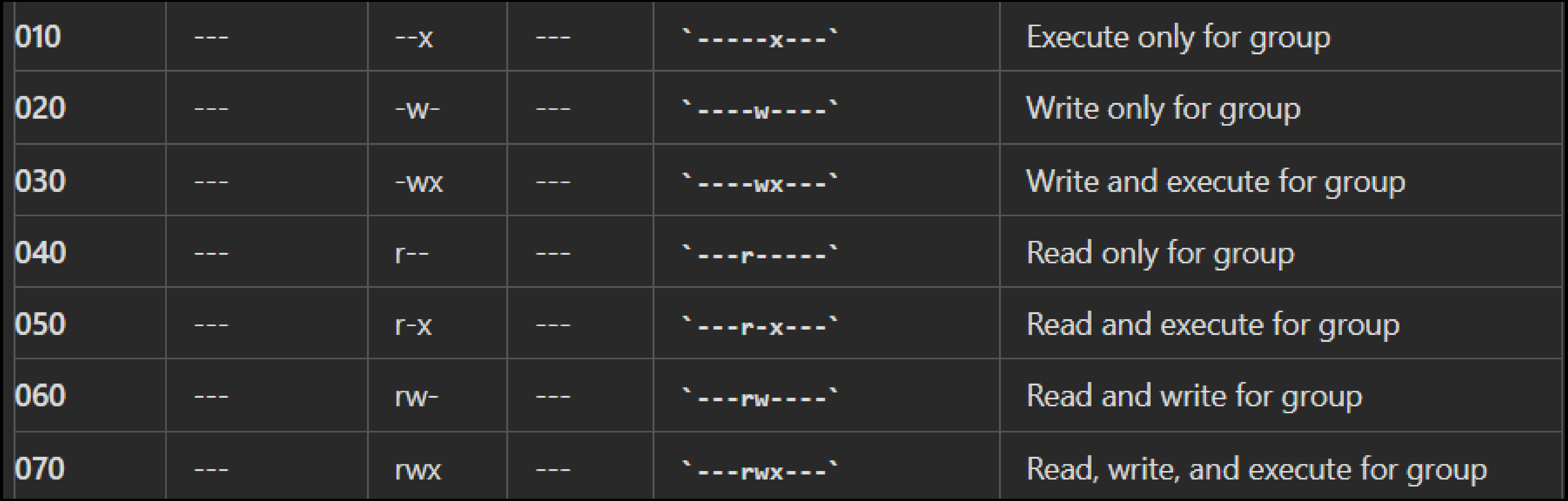
- Examples:
  - `chmod u+x file`: Adds execute permission for the owner.
  - `chmod g-w file`: Removes write permission for the group.
  - `chmod o=r file`: Sets read-only permission for others.
  - `chmod a+x file`: Adds execute permission for everyone (user, group, others).



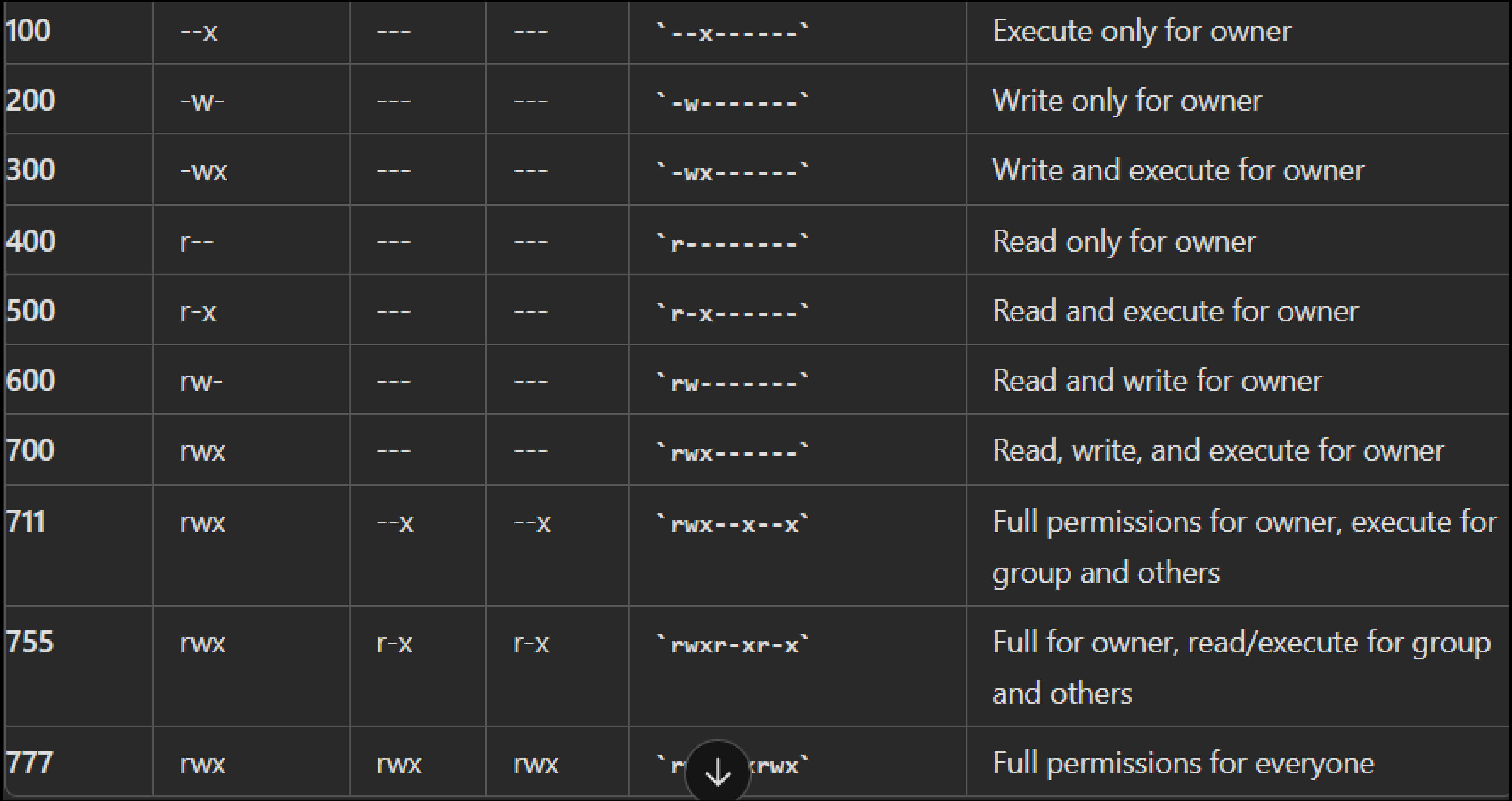
```
zinc@ubu:/tmp/book$ chmod u=rwx,g=rw,o= note
zinc@ubu:/tmp/book$ ls -l
total 4
-rwxrw---- 1 zinc zinc 26 22:33 12        آب  note
zinc@ubu:/tmp/book$ chmod u=rw,g-w,o+r note
zinc@ubu:/tmp/book$ ls -l
total 4
-rw-r--r-- 1 zinc zinc 26 22:33 12        آب  note
zinc@ubu:/tmp/book$ chmod 777 note
zinc@ubu:/tmp/book$ ls -l
total 4
-rwxrwxrwx 1 zinc zinc 26 22:33 12        آب  note
zinc@ubu:/tmp/book$ chmod 000 note
zinc@ubu:/tmp/book$ ls -l
total 4
---------- 1 zinc zinc 26 22:33 12        آب  note
zinc@ubu:/tmp/book$ chmod 421 note
zinc@ubu:/tmp/book$ ls -l
total 4
-r---w---x 1 zinc zinc 26 22:33 12        آب  note
zinc@ubu:/tmp/book$ chmod 640 note
zinc@ubu:/tmp/book$ ls -l
total 4
-rw-r----- 1 zinc zinc 26 22:33 12        آب  note
```



Octal Value	Owner (User)	Group	Others	Symbolic Representation	Description
000	---	---	---	`---------`	No permissions for anyone
001	---	---	--x	`--------x`	Execute only for others
002	---	---	-w-	`-------w-`	Write only for others
003	---	---	-wx	`-------wx`	Write and execute for others
004	---	---	r--	`------r--`	Read only for others
005	---	---	r-x	`------r-x`	Read and execute for others
006	---	---	rw-	`------rw-`	Read and write for others
007	---	---	rwx	`------rwx`	Read, write, and execute for others



010	---	--x	---	`-----x---`	Execute only for group
020	---	-w-	---	`----w----`	Write only for group
030	---	-wx	---	`----wx---`	Write and execute for group
040	---	r--	---	`---r-----`	Read only for group
050	---	r-x	---	`---r-x---`	Read and execute for group
060	---	rw-	---	`---rw----`	Read and write for group
070	---	rwx	---	`---rwx---`	Read, write, and execute for group



100	--x	---	---	`--x------`	Execute only for owner
200	-w-	---	---	`-w-------`	Write only for owner
300	-wx	---	---	`-wx------`	Write and execute for owner
400	r--	---	---	`r--------`	Read only for owner
500	r-x	---	---	`r-x------`	Read and execute for owner
600	rw-	---	---	`rw-------`	Read and write for owner
700	rwx	---	---	`rwx------`	Read, write, and execute for owner
711	rwx	--x	--x	`rwx--x--x`	Full permissions for owner, execute for group and others
755	rwx	r-x	r-x	`rwxr-xr-x`	Full for owner, read/execute for group and others
777	rwx	rwx	rwx	`r [illegible] rwx`	Full permissions for everyone

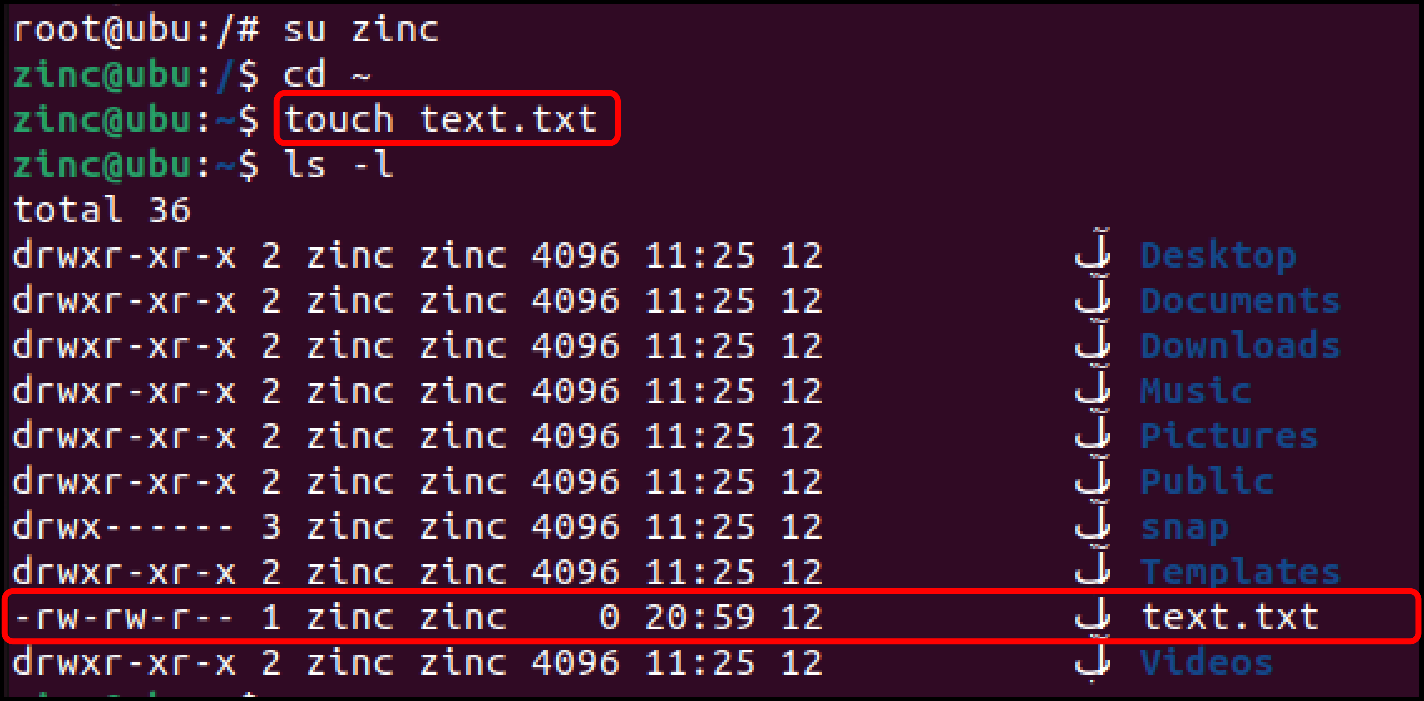
- By default, when creating a file it is:
- User & Group ➔ read & write

Q: Why execute is not?

Most files created by users are text files, scripts, or data files.

If every file were executable by default, it could **lead to accidental or malicious execution** of files that shouldn't be run as programs.

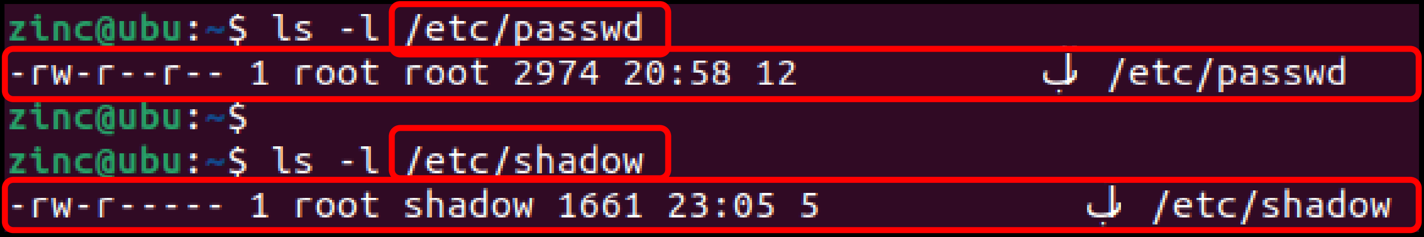
- Other ➔ read only



```
root@ubu:/# su zinc
zinc@ubu:/$ cd ~
zinc@ubu:~$ touch text.txt
zinc@ubu:~$ ls -l
total 36
drwxr-xr-x 2 zinc zinc 4096 11:25 12          لَ  Desktop
drwxr-xr-x 2 zinc zinc 4096 11:25 12          لَ  Documents
drwxr-xr-x 2 zinc zinc 4096 11:25 12          لَ  Downloads
drwxr-xr-x 2 zinc zinc 4096 11:25 12          لَ  Music
drwxr-xr-x 2 zinc zinc 4096 11:25 12          لَ  Pictures
drwxr-xr-x 2 zinc zinc 4096 11:25 12          لَ  Public
drwx------ 3 zinc zinc 4096 11:25 12          لَ  snap
drwxr-xr-x 2 zinc zinc 4096 11:25 12          لَ  Templates
-rw-rw-r-- 1 zinc zinc    0 20:59 12          لَ  text.txt
drwxr-xr-x 2 zinc zinc 4096 11:25 12          لَ  Videos
```

- /etc/passwd
- /etc/shadow

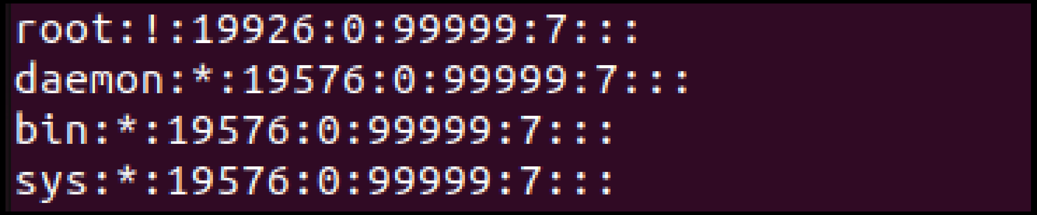
The `/etc/shadow` file is highly sensitive because it stores hashed passwords. Therefore, only the `root` user is allowed to read this file



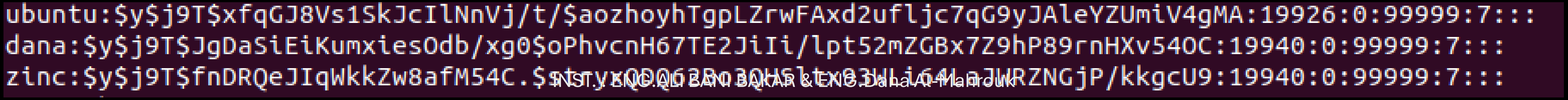
```
zinc@ubu:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 2974 20:58 12          ل  /etc/passwd
zinc@ubu:~$
zinc@ubu:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1661 23:05 5          ل  /etc/shadow
```

/etc/shadow

- Even though the passwords are hashed, it's crucial to protect this file from unauthorized access to prevent offline attacks, such as brute force or dictionary attacks on the hashed passwords.
- Notice that the beginning of the hash value is the same for all three users because it is the same password, so why don't they have the same full hash value? The reason is that a random value is added to the password to make it more difficult to discover



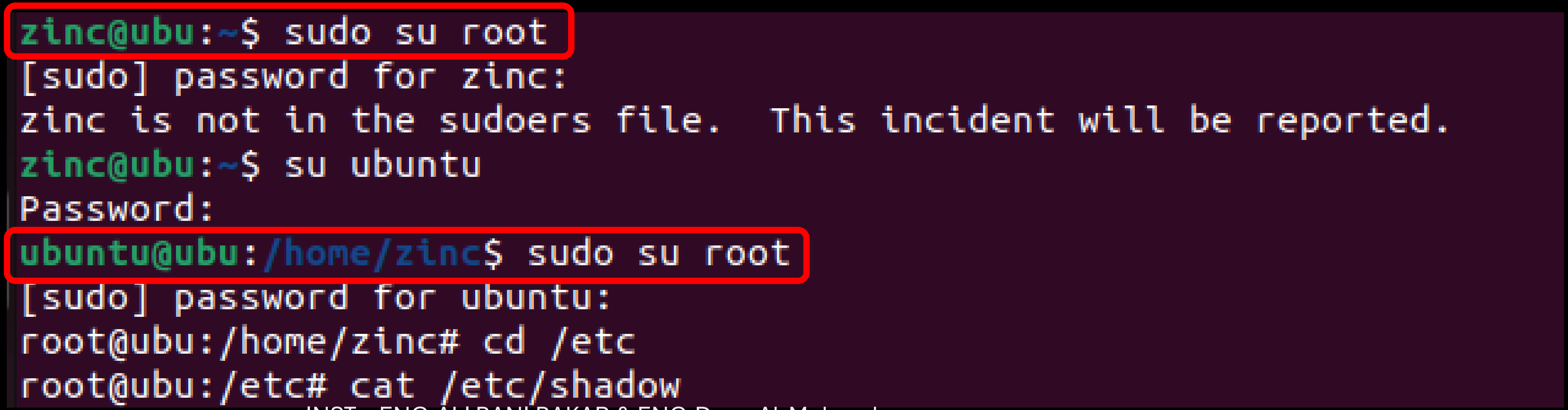
```
root:!:19926:0:99999:7:::
daemon:*:19576:0:99999:7:::
bin:*:19576:0:99999:7:::
sys:*:19576:0:99999:7:::
```



```
ubuntu:$y$j9T$xfqGJ8Vs1SkJcIlNnVj/t/$aozhoyhTgpLZrwFAxd2ufljc7qG9yJAleYZUmiV4gMA:19926:0:99999:7:::
dana:$y$j9T$JgDaSiEiKumxiesOdb/xg0$oPhvcnH67TE2JiIi/lpt52mZGBx7Z9hP89rnHXv54OC:19940:0:99999:7:::
zinc:$y$j9T$fnDRQeJIqWkkZw8afM54C.$sttyxQDQ63Bo3QHSltx93WLi64LaJWRZNGjP/kkgcU9:19940:0:99999:7:::
```

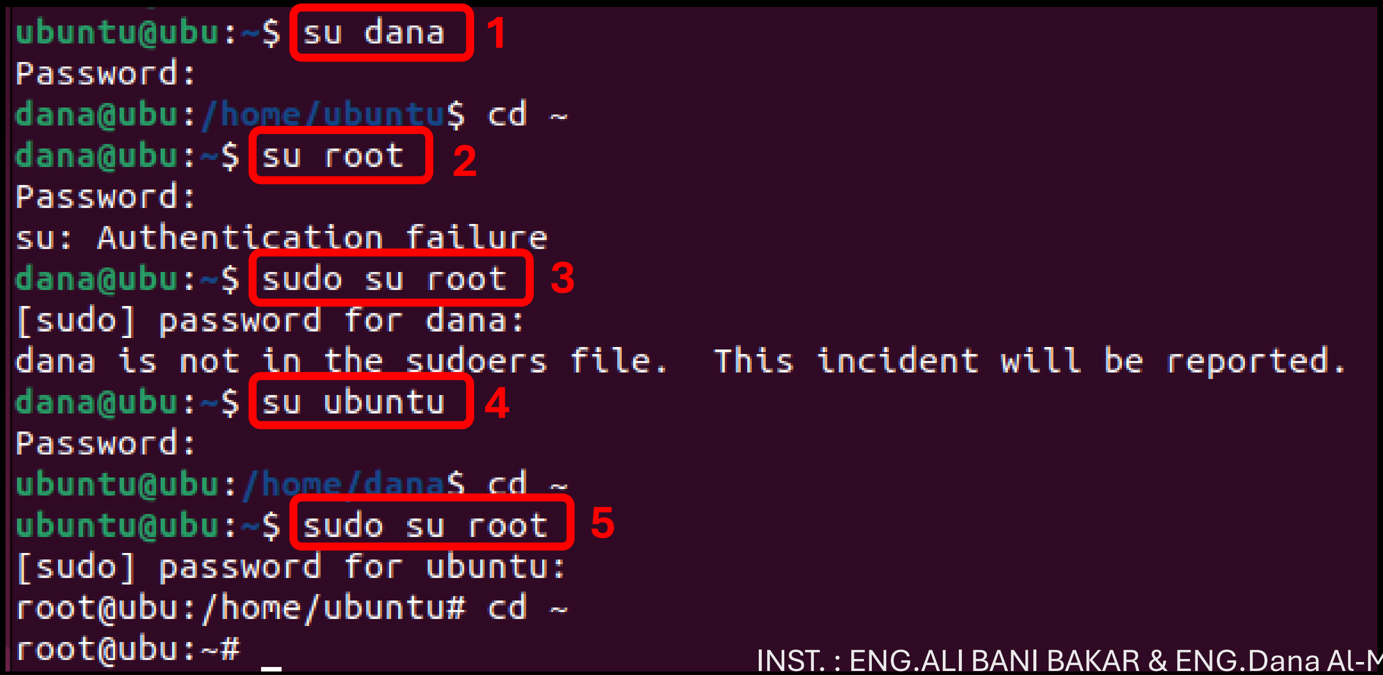
Login to Root User

- normal users do not have the ability to execute administrator commands or access the root account directly. Instead, when they attempt to perform actions that require elevated privileges, the system prompts for a password or logs the attempt if it fails.
- if a user tries to use `sudo` without permission, they will see a message like `user is not in the sudoers file. This incident will be reported.`



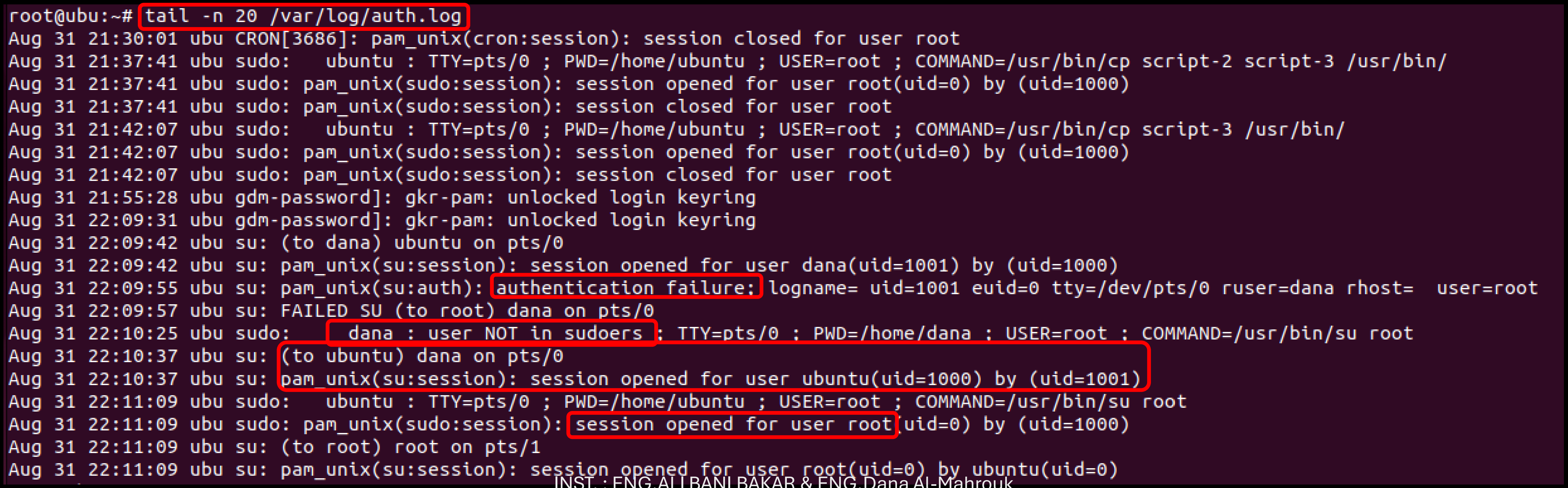
```
  zinc@ubu:~$ sudo su root
1 [sudo] password for zinc:
2 zinc is not in the sudoers file.  This incident will be reported.
  zinc@ubu:~$ su ubuntu
  Password:
  ubuntu@ubu:/home/zinc$ sudo su root
  [sudo] password for ubuntu:
  root@ubu:/home/zinc# cd /etc
  root@ubu:/etc# cat /etc/shadow
```

- `sudo`: the system checks whether the user is allowed to perform this action by consulting the `/etc/sudoers` file. If allowed, the system will prompt the user for their own password (not the root password) to verify their identity.
- `su`: the system will ask for the root password. If the user doesn’t know the root password, the attempt will fail.



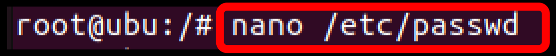
```
ubuntu@ubu:~$ su dana
Password:
dana@ubu:/home/ubuntu$ cd ~
dana@ubu:~$ su root
Password:
su: Authentication failure
dana@ubu:~$ sudo su root
[sudo] password for dana:
dana is not in the sudoers file.  This incident will be reported.
dana@ubu:~$ su ubuntu
Password:
ubuntu@ubu:/home/dana$ cd ~
ubuntu@ubu:~$ sudo su root
[sudo] password for ubuntu:
root@ubu:/home/ubuntu# cd ~
root@ubu:~# _
```

- Logging of Failed Login Attempts: ➔ `/var/log` directory
  - **`/var/log/auth.log`:** On Debian-based systems (like Ubuntu), authentication logs, including `sudo` attempts and `su` attempts, are recorded here.
  - **`/var/log/secure`:** On Red Hat-based systems (like CentOS and Fedora), the corresponding log file is `/var/log/secure`.
- Contents: These log files record details such as the date and time of the attempt, the user who attempted the action, the command they tried to execute, and whether the attempt was successful.

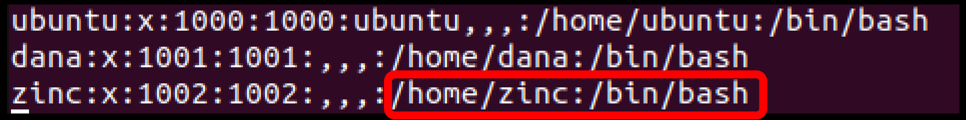


```
root@ubu:~# tail -n 20 /var/log/auth.log
Aug 31 21:30:01 ubu CRON[3686]: pam_unix(cron:session): session closed for user root
Aug 31 21:37:41 ubu sudo:   ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/bin/cp script-2 script-3 /usr/bin/
Aug 31 21:37:41 ubu sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Aug 31 21:37:41 ubu sudo: pam_unix(sudo:session): session closed for user root
Aug 31 21:42:07 ubu sudo:   ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/bin/cp script-3 /usr/bin/
Aug 31 21:42:07 ubu sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Aug 31 21:42:07 ubu sudo: pam_unix(sudo:session): session closed for user root
Aug 31 21:55:28 ubu gdm-password]: gkr-pam: unlocked login keyring
Aug 31 22:09:31 ubu gdm-password]: gkr-pam: unlocked login keyring
Aug 31 22:09:42 ubu su: (to dana) ubuntu on pts/0
Aug 31 22:09:42 ubu su: pam_unix(su:session): session opened for user dana(uid=1001) by (uid=1000)
Aug 31 22:09:55 ubu su: pam_unix(su:auth): authentication failure; logname= uid=1001 euid=0 tty=/dev/pts/0 ruser=dana rhost=  user=root
Aug 31 22:09:57 ubu su: FAILED SU (to root) dana on pts/0
Aug 31 22:10:25 ubu sudo:     dana : user NOT in sudoers ; TTY=pts/0 ; PWD=/home/dana ; USER=root ; COMMAND=/usr/bin/su root
Aug 31 22:10:37 ubu su: (to ubuntu) dana on pts/0
Aug 31 22:10:37 ubu su: pam_unix(su:session): session opened for user ubuntu(uid=1000) by (uid=1001)
Aug 31 22:11:09 ubu sudo:   ubuntu : TTY=pts/0 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/bin/su root
Aug 31 22:11:09 ubu sudo: pam_unix(sudo:session): session opened for user root(uid=0) by (uid=1000)
Aug 31 22:11:09 ubu su: (to root) root on pts/1
Aug 31 22:11:09 ubu su: pam_unix(su:session): session opened for user root(uid=0) by ubuntu(uid=0)
```

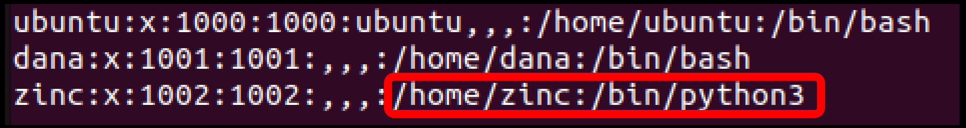
/etc/passwd → /bin/bash-python3



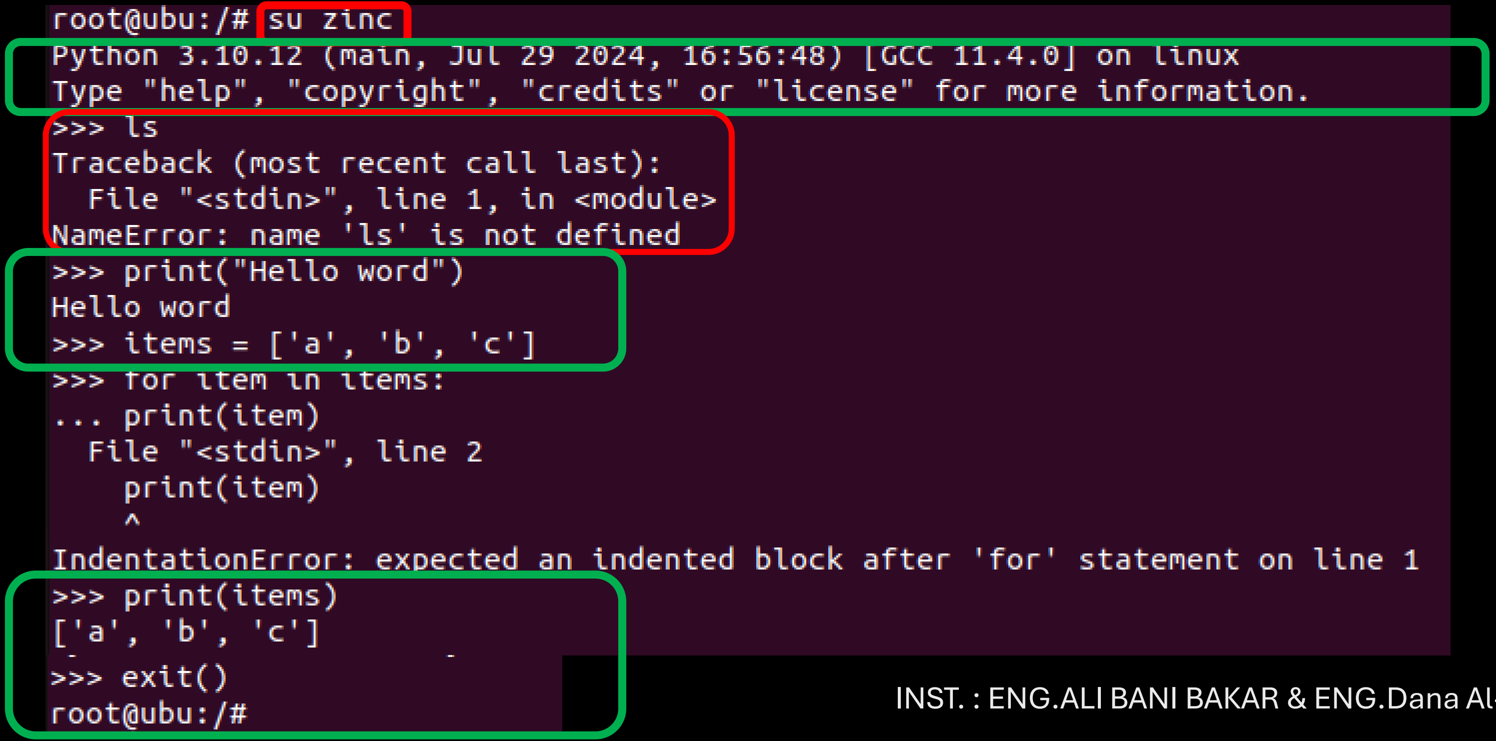
```
root@ubu:/# nano /etc/passwd
```



```
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
dana:x:1001:1001:,,,:/home/dana:/bin/bash
zinc:x:1002:1002:,,,:/home/zinc:/bin/bash
```

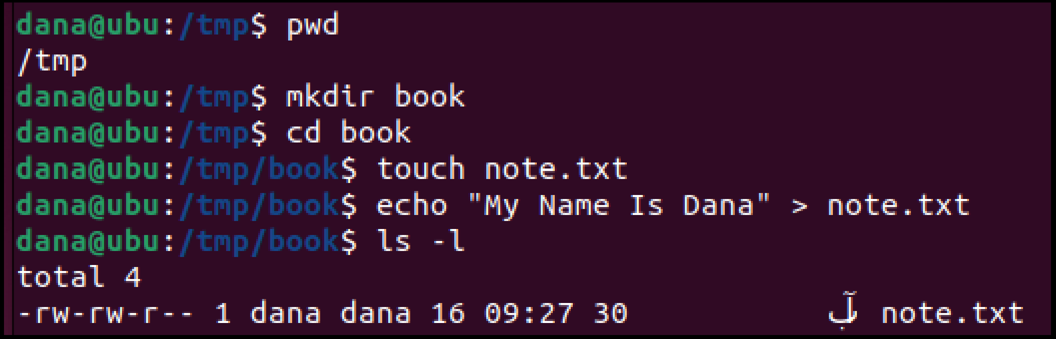


```
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
dana:x:1001:1001:,,,:/home/dana:/bin/bash
zinc:x:1002:1002:,,,:/home/zinc:/bin/python3
```

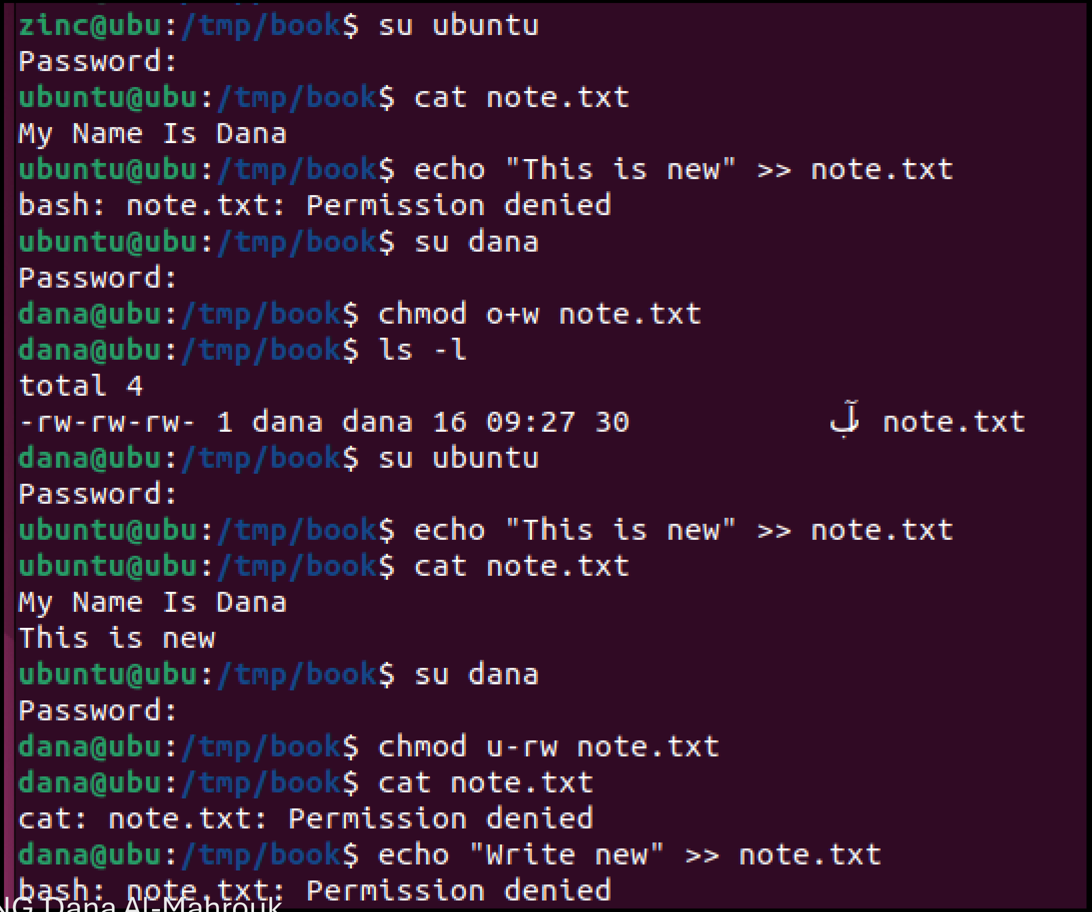


```
root@ubu:/# su zinc
Python 3.10.12 (main, Jul 29 2024, 16:56:48) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> ls
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'ls' is not defined
>>> print("Hello word")
Hello word
>>> items = ['a', 'b', 'c']
>>> for item in items:
... print(item)
  File "<stdin>", line 2
    print(item)
    ^
IndentationError: expected an indented block after 'for' statement on line 1
>>> print(items)
['a', 'b', 'c']
>>> exit()
root@ubu:/#
```

Permission of Files



```
dana@ubu:/tmp$ pwd
/tmp
dana@ubu:/tmp$ mkdir book
dana@ubu:/tmp$ cd book
dana@ubu:/tmp/book$ touch note.txt
dana@ubu:/tmp/book$ echo "My Name Is Dana" > note.txt
dana@ubu:/tmp/book$ ls -l
total 4
-rw-rw-r-- 1 dana dana 16 09:27 30          آب note.txt
```



```
zinc@ubu:/tmp/book$ su ubuntu
Password:
ubuntu@ubu:/tmp/book$ cat note.txt
My Name Is Dana
ubuntu@ubu:/tmp/book$ echo "This is new" >> note.txt
bash: note.txt: Permission denied
ubuntu@ubu:/tmp/book$ su dana
Password:
dana@ubu:/tmp/book$ chmod o+w note.txt
dana@ubu:/tmp/book$ ls -l
total 4
-rw-rw-rw- 1 dana dana 16 09:27 30          آب note.txt
dana@ubu:/tmp/book$ su ubuntu
Password:
ubuntu@ubu:/tmp/book$ echo "This is new" >> note.txt
ubuntu@ubu:/tmp/book$ cat note.txt
My Name Is Dana
This is new
ubuntu@ubu:/tmp/book$ su dana
Password:
dana@ubu:/tmp/book$ chmod u-rw note.txt
dana@ubu:/tmp/book$ cat note.txt
cat: note.txt: Permission denied
dana@ubu:/tmp/book$ echo "Write new" >> note.txt
bash: note.txt: Permission denied
```

Day 9

- Outline
  - Review chmod
  - Process
    - Intro
    - PID, PPID, State
    - Fork → bash
    - Kill → exit
    - Echo $$
    - Firefox, sleep
    - Ctrl + C
    - Ctrl + Z
    - Jobs
    - Fg %<…>
    - &

Day 9

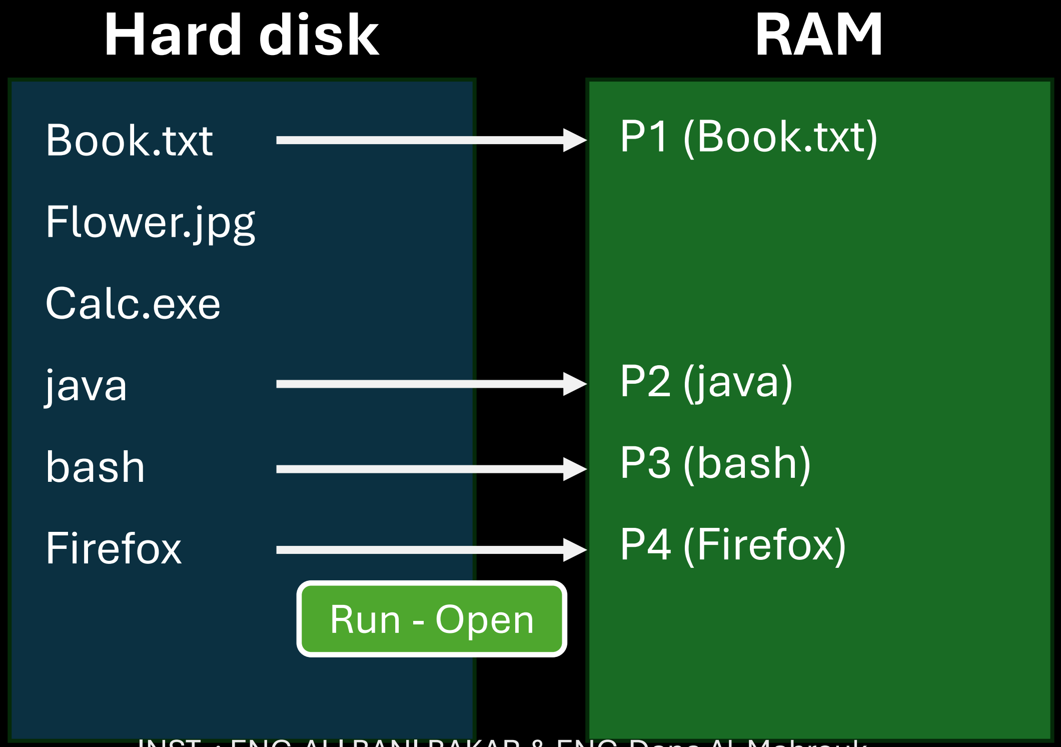
- Outline
  - Process Cont…
    - ps
    - Ps –a
    - Ps –aux
    - Ps –aux | more
    - Pstree
  - Gnom
  - Ctrl + Shift + f4

Day 9

- Outline
  - Cp
  - rm
  - Ls –la
  - Cat
  - More
  - Less
  - Grep
    - Grep –i
    - Grep \|
    - Grep –r

Process

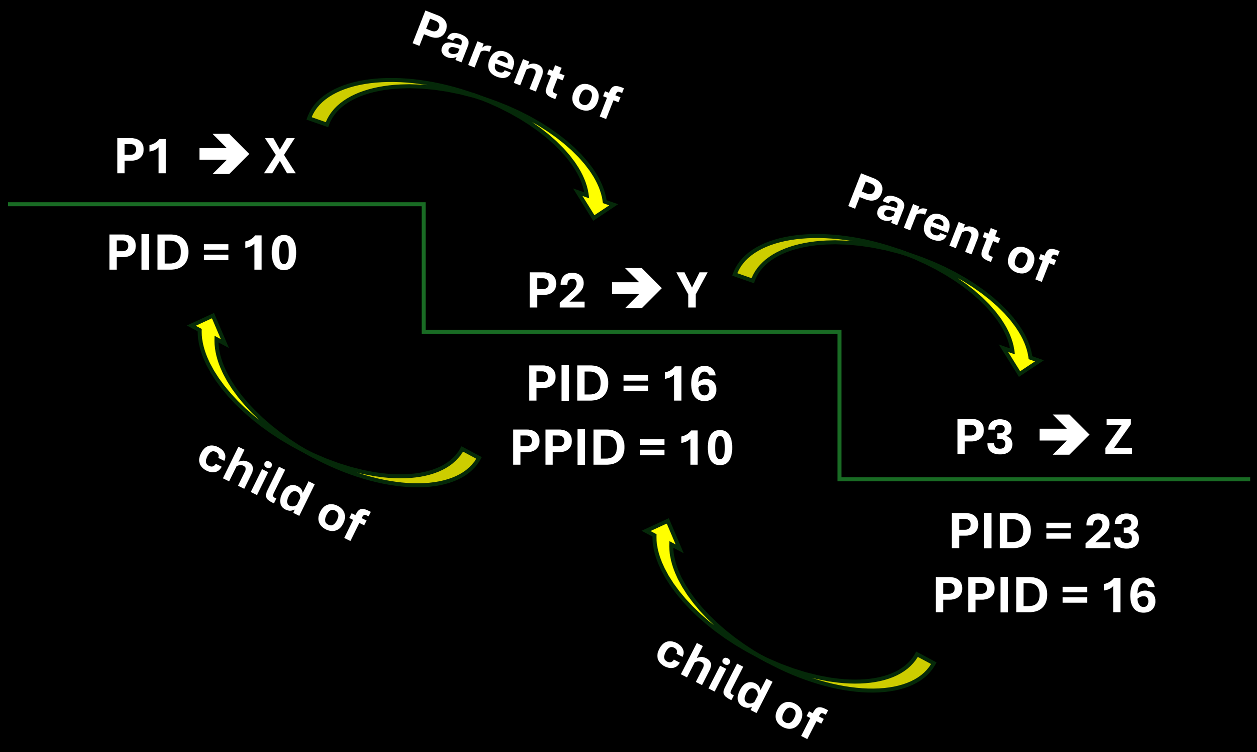
- A process is an instance of a running program. When you execute a command or run an application, the operating system creates a process to handle it. Each process has a unique Process ID (PID) and is managed by the Linux kernel.



Child Process

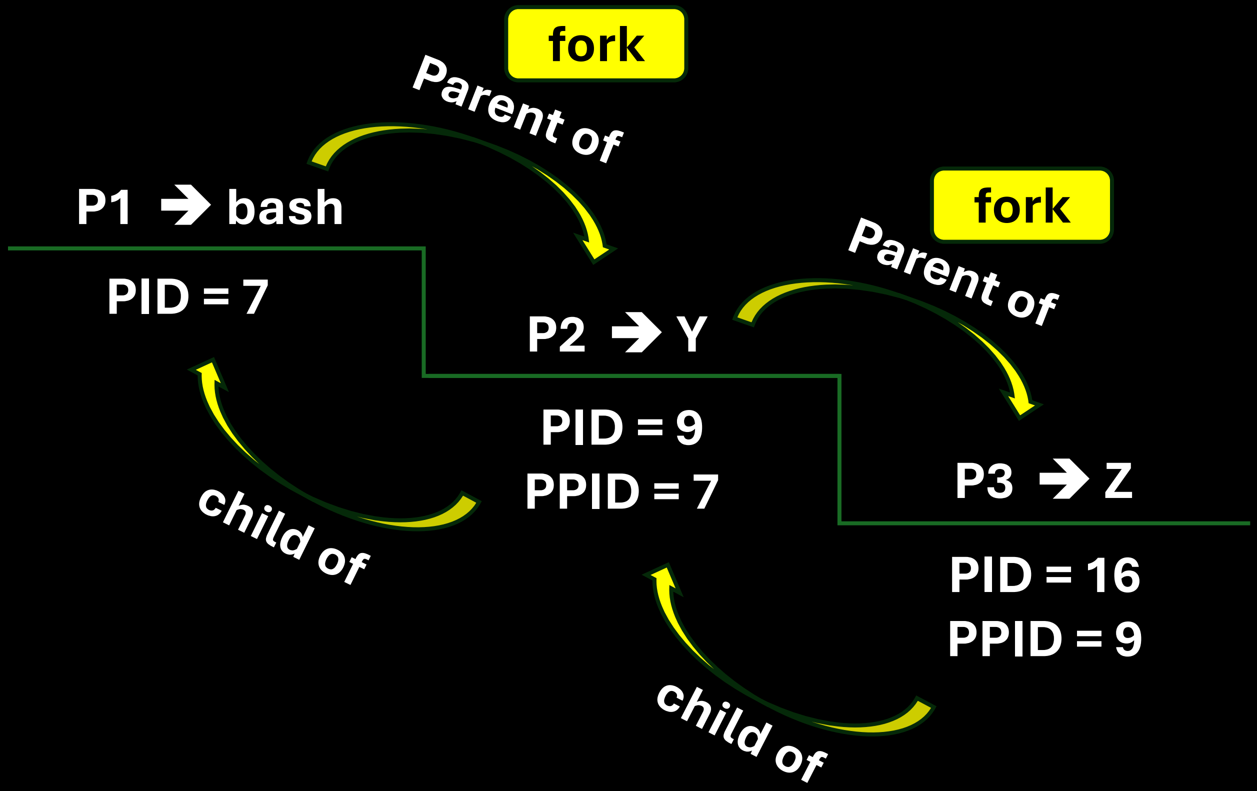
- A child process in Linux is a process created by another process, referred to as the parent process.

- The child process inherits the parent's **memory space**, including **code, data, and file descriptors**. However, the child process has its **own PID and separate memory space**, allowing it to **execute independently**.



Fork

- The `fork()` system call is used to create a new process by duplicating the calling process. This is fundamental in Linux for process management.

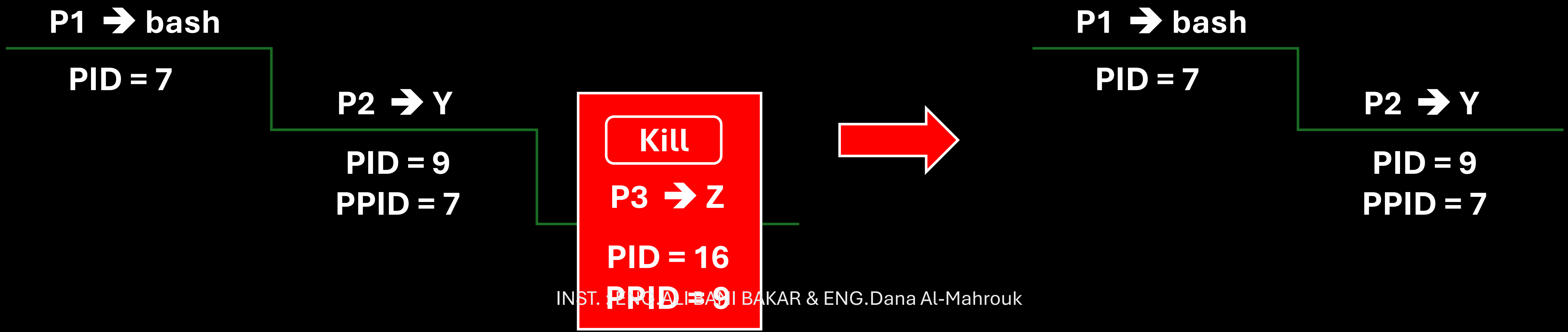


Kill

- send signals to processes, often to terminate them.
- `kill -9` : This sends the `SIGKILL` signal, which forcefully and immediately kills the process without giving it a chance to clean up.
- `kill -15` : This sends the `SIGTERM` signal, which politely asks the process to terminate. The process can catch this signal and clean up resources before exiting.
- `kill -2` : This sends the `SIGINT` signal, which is like pressing `Ctrl+C` in the terminal.

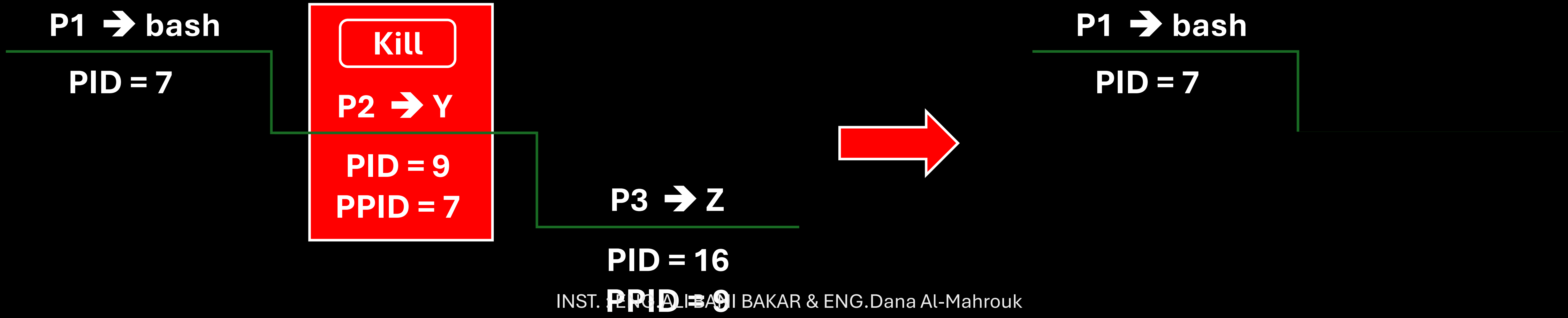
Ctrl + C on Child process

- The `Ctrl + C` sends a `SIGINT` (Signal Interrupt) signal to the foreground process group in a terminal.
- Sends `SIGINT` to the child if it's in the foreground. The child usually terminates, with control returning to the parent.



Ctrl + C on Parent process

- Sends `SIGINT` to the parent. If the parent is in the foreground, it typically terminates, potentially also affecting child processes based on how the parent is coded.

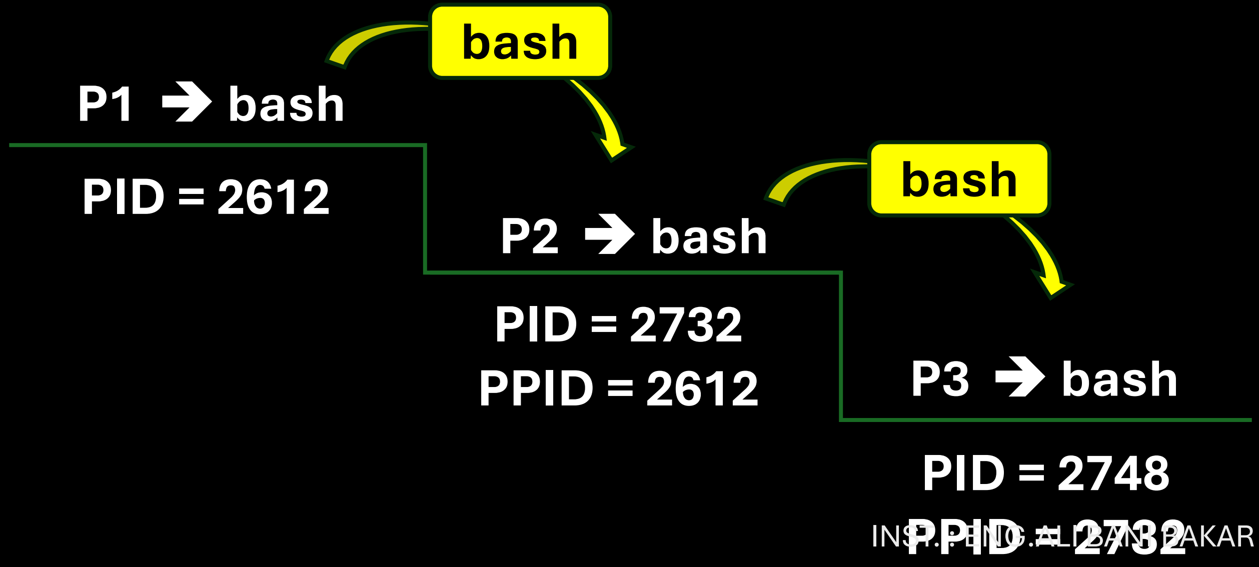


Ex

echo $$ → prints the Process ID (PID) of the current shell session to the terminal.

bash → in a terminal starts a new Bash shell session, nested within the current one.

exit or Ctrl + D → exit the new Bash shell by typing, which returns you to the parent shell.



P1 → bash

PID = 2612

bash

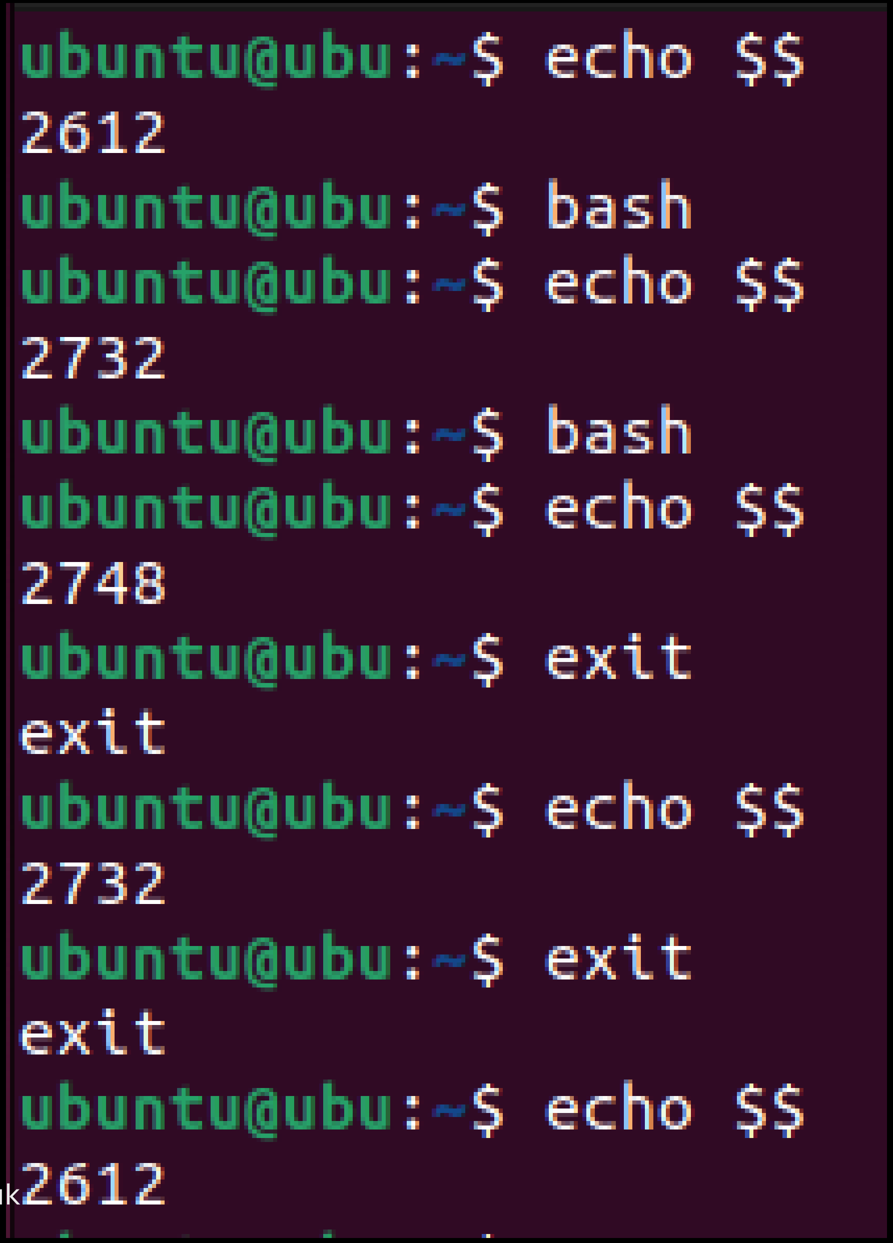
P2 → bash

PID = 2732
PPID = 2612

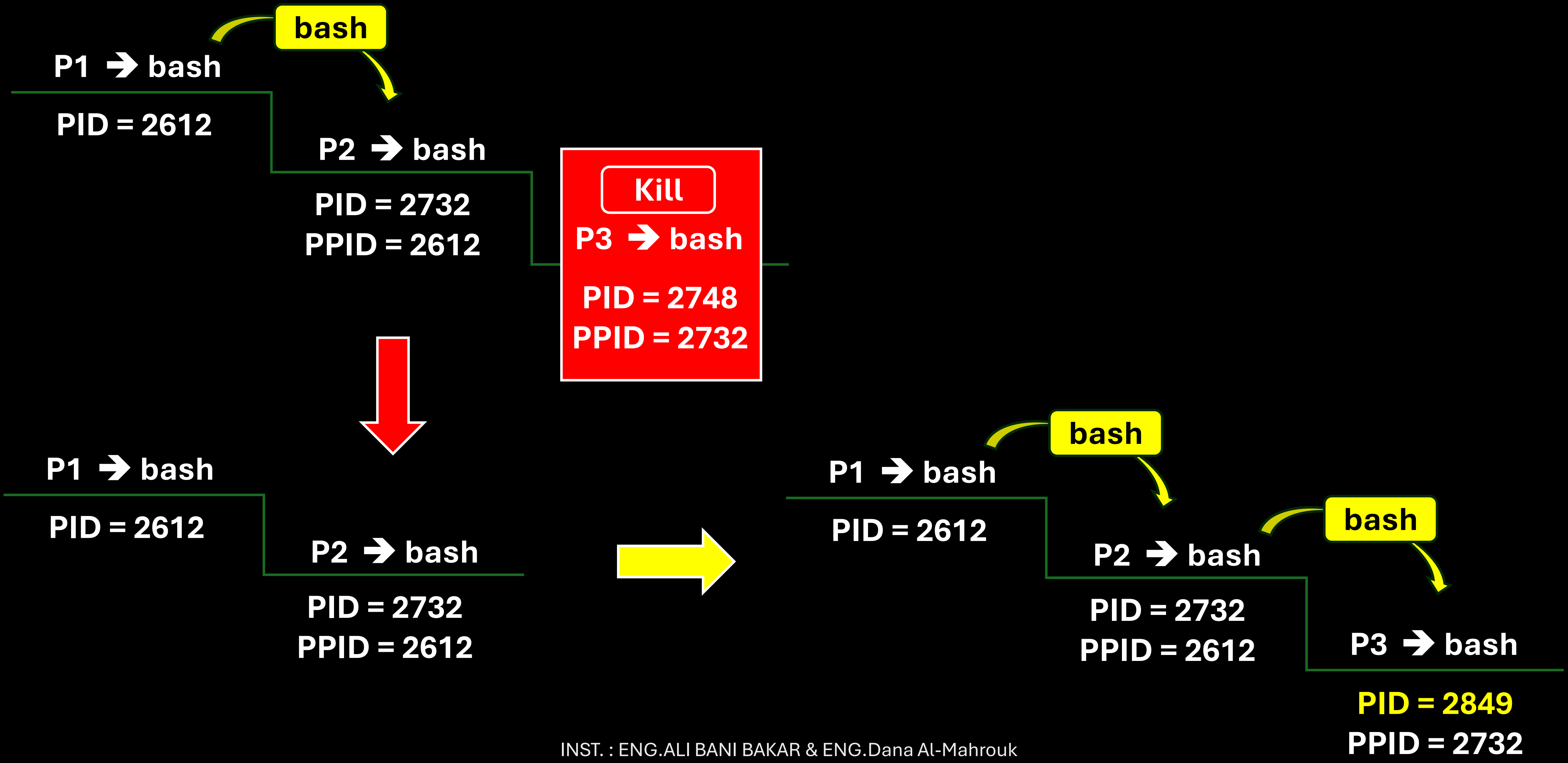
bash

P3 → bash

PID = 2748
PPID = 2732



```
ubuntu@ubu:~$ echo $$
2612
ubuntu@ubu:~$ bash
ubuntu@ubu:~$ echo $$
2732
ubuntu@ubu:~$ bash
ubuntu@ubu:~$ echo $$
2748
ubuntu@ubu:~$ exit
exit
ubuntu@ubu:~$ echo $$
2732
ubuntu@ubu:~$ exit
exit
ubuntu@ubu:~$ echo $$
2612
```



bash

P1 → bash

PID = 2612

P2 → bash

PID = 2732
PPID = 2612

Kill

P3 → bash

PID = 2748
PPID = 2732

P1 → bash

PID = 2612

P2 → bash

PID = 2732
PPID = 2612

bash

P1 → bash

PID = 2612

bash

P2 → bash

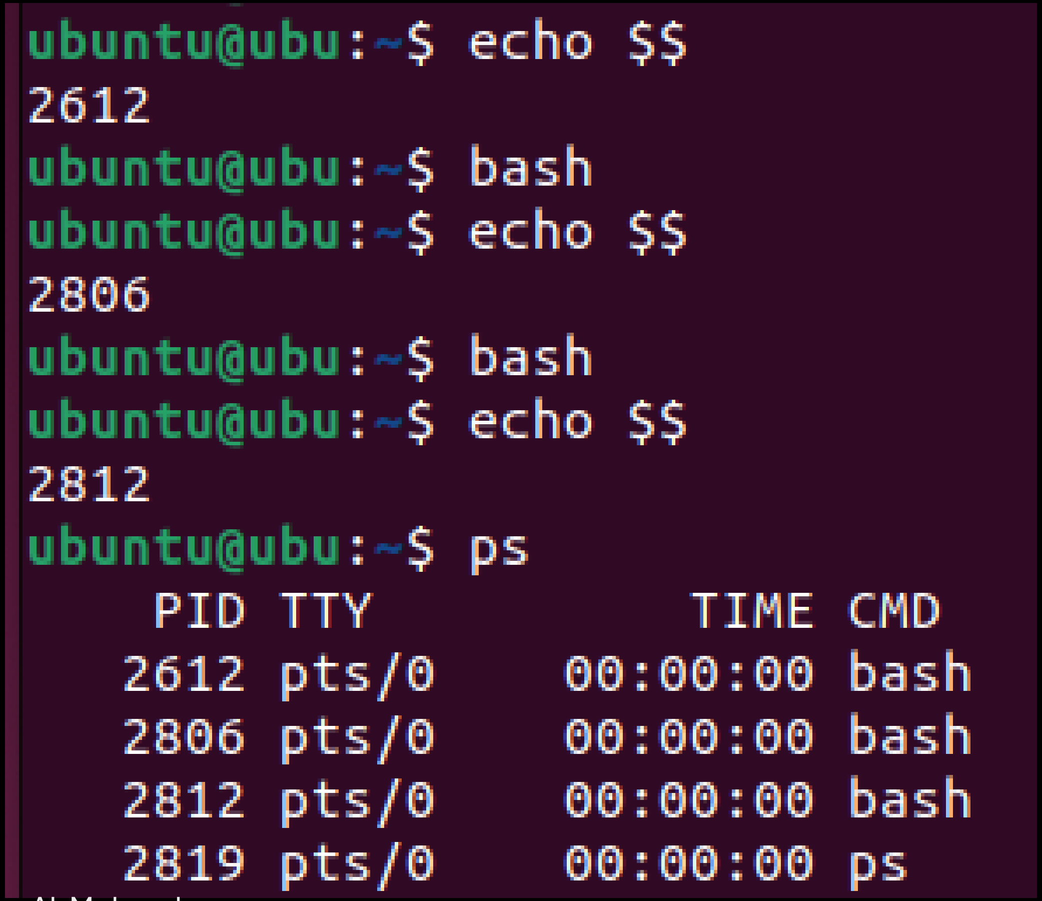
PID = 2732
PPID = 2612

P3 → bash

PID = 2849
PPID = 2732

ps

- The ` ps ` command in Linux is used to display information about the currently running processes on the system.
- It provides a snapshot of the processes running at the time the command is executed.
- including details like process IDs, user IDs, CPU usage, memory usage, and more.



```
ubuntu@ubu:~$ echo $$
2612
ubuntu@ubu:~$ bash
ubuntu@ubu:~$ echo $$
2806
ubuntu@ubu:~$ bash
ubuntu@ubu:~$ echo $$
2812
ubuntu@ubu:~$ ps
    PID TTY          TIME CMD
   2612 pts/0    00:00:00 bash
   2806 pts/0    00:00:00 bash
   2812 pts/0    00:00:00 bash
   2819 pts/0    00:00:00 ps
```

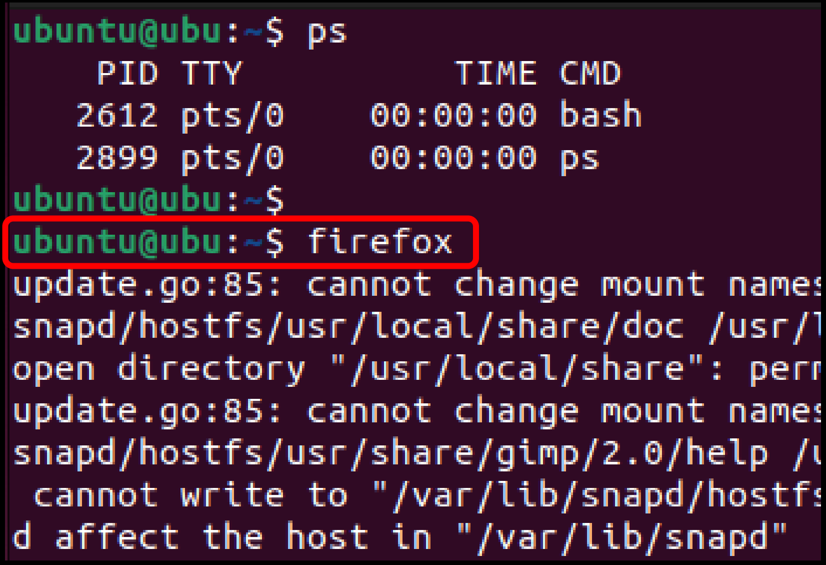
ps

- ` ps -e` or ` ps -A` : Lists all processes running on the system.
- ` ps -f` : Provides a full-format listing, showing more details about each process.
- ` ps -ef` : Another option for a full-format listing, showing processes in a standard Unix format.
- ` ps -l` : Provides a long format listing, including additional information like priority and nice value.
- ` ps -ejH` : Displays the process hierarchy in a tree format, showing parent-child relationships.
- ` ps -axjf` : Another way to view the process tree, including more details.
- ` ps -aux` : Displays all processes with detailed information, including
  - processes from all users (` a`)
  - processes not attached to a terminal (` x`)
  - user-oriented format (` u`)

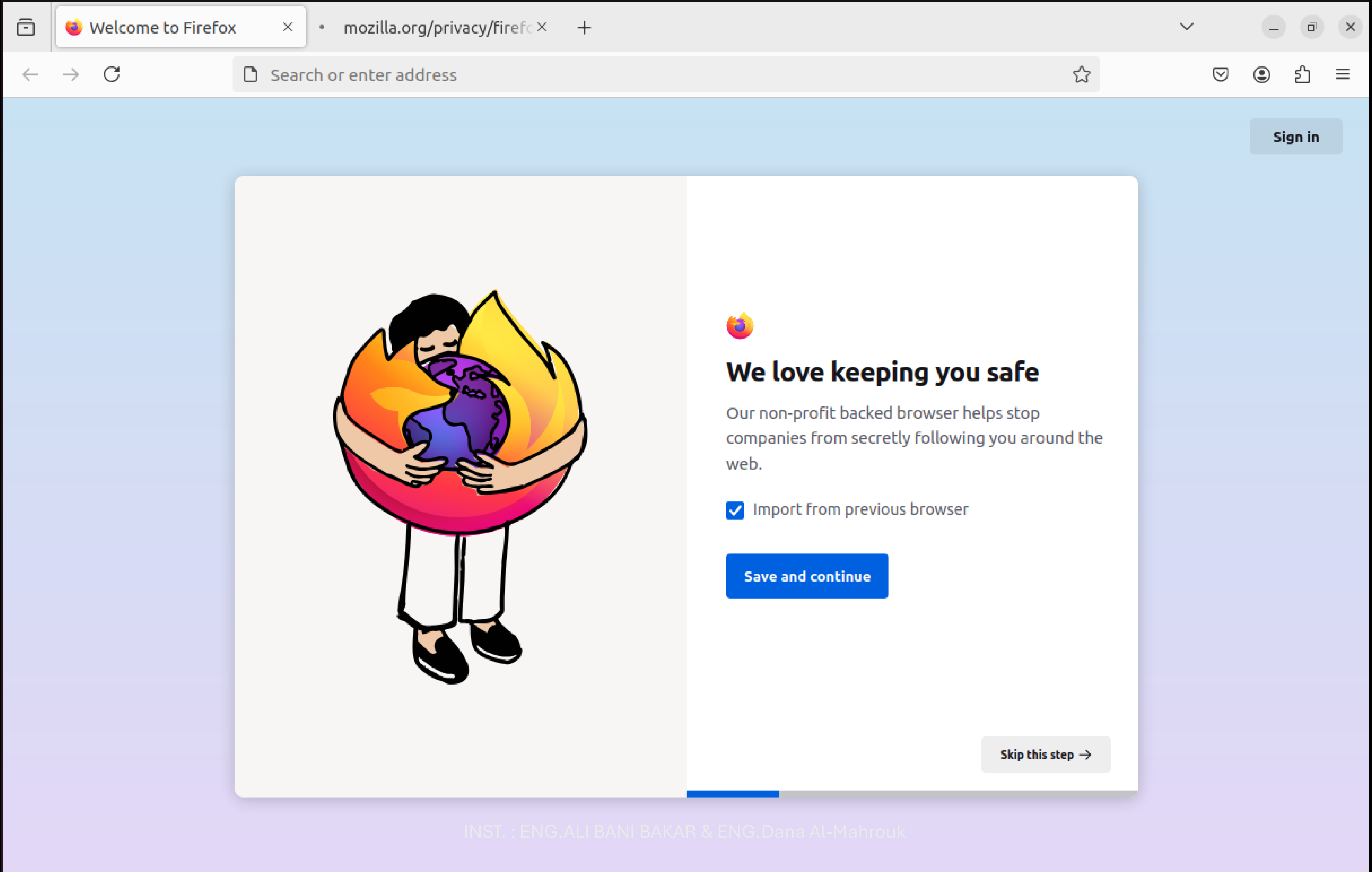
- PID: Process ID.
- TTY: Terminal associated with the process.
- TIME: CPU time consumed by the process.
- CMD: Command that started the process.
- C: CPU utilization of the process.
- STIME: Start time of the process.

firefox

- launch the Firefox web browser from the terminal.
- Open a Specific URL: firefox https://www.facebook.com
- Run Firefox in the Background: firefox &
- Private Browsing Mode: firefox --private-window
- Open Firefox with a Specific Profile: firefox -P "ProfileName"
- Safe Mode: firefox --safe-mode



```
ubuntu@ubu:~$ ps
    PID TTY          TIME CMD
   2612 pts/0    00:00:00 bash
   2899 pts/0    00:00:00 ps
ubuntu@ubu:~$
ubuntu@ubu:~$ firefox
update.go:85: cannot change mount names
snapd/hostfs/usr/local/share/doc /usr/l
open directory "/usr/local/share": perm
update.go:85: cannot change mount names
snapd/hostfs/usr/share/gimp/2.0/help /u
 cannot write to "/var/lib/snapd/hostfs
d affect the host in "/var/lib/snapd"
```



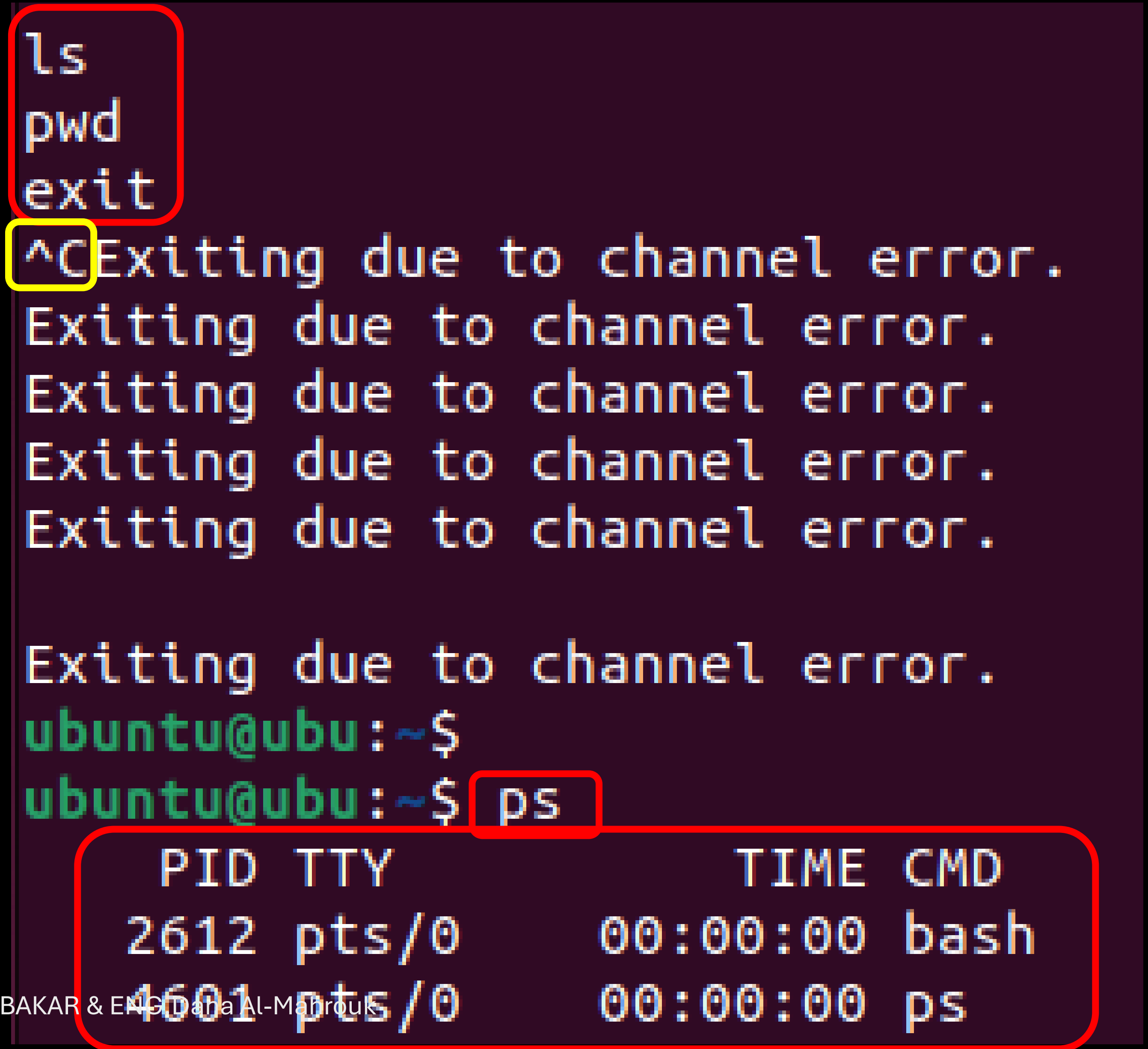
We love keeping you safe

Our non-profit backed browser helps stop companies from secretly following you around the web.

☑ Import from previous browser

Ctrl + C

- The ` Ctrl + C ` sends a ` SIGINT ` (Signal Interrupt) signal to the foreground process group in a terminal.



```
ls
pwd
exit
^CExiting due to channel error.
Exiting due to channel error.
Exiting due to channel error.
Exiting due to channel error.
Exiting due to channel error.

Exiting due to channel error.
ubuntu@ubu:~$
ubuntu@ubu:~$ ps
    PID TTY          TIME CMD
   2612 pts/0    00:00:00 bash
   4601 pts/0    00:00:00 ps
```

Foreground & Background

- **Foreground Processes**
- runs directly in the terminal and takes control of the terminal until it completes. the terminal is tied up with that process, and you cannot use it for other commands until the process finishes or is interrupted
- - You can interact with the process.

- **Background Processes**
- runs independently of the terminal. the terminal remains free for you to execute other commands. useful for tasks that do not require user interaction and may take a long time to complete.
- You can view and manage background processes using commands like `jobs`, `fg`, `bg`, and `kill`.

Managing Background Processes

- \` **jobs** \` : Lists all **background** jobs associated with the current terminal.

- \` **fg** \` : Brings a background process to the **foreground**. If you have multiple background jobs, you can specify which one by its job number.

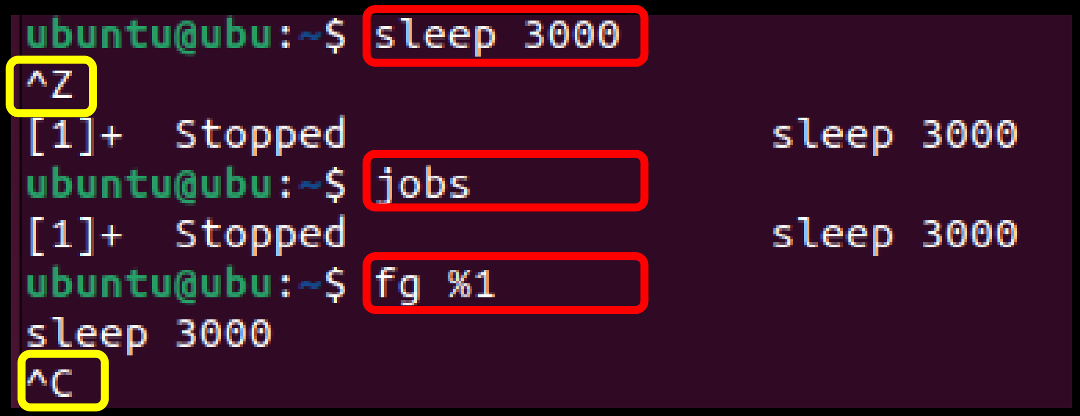
- \` **bg** \` : Resumes a paused job in the **background**. If a job is stopped (e.g., after pressing \` Ctrl + Z \`), you can use \` bg \` to continue it in the background.

- \` **kill** \` : Sends a signal to **terminate a background process** (or any process) by its PID or job number.

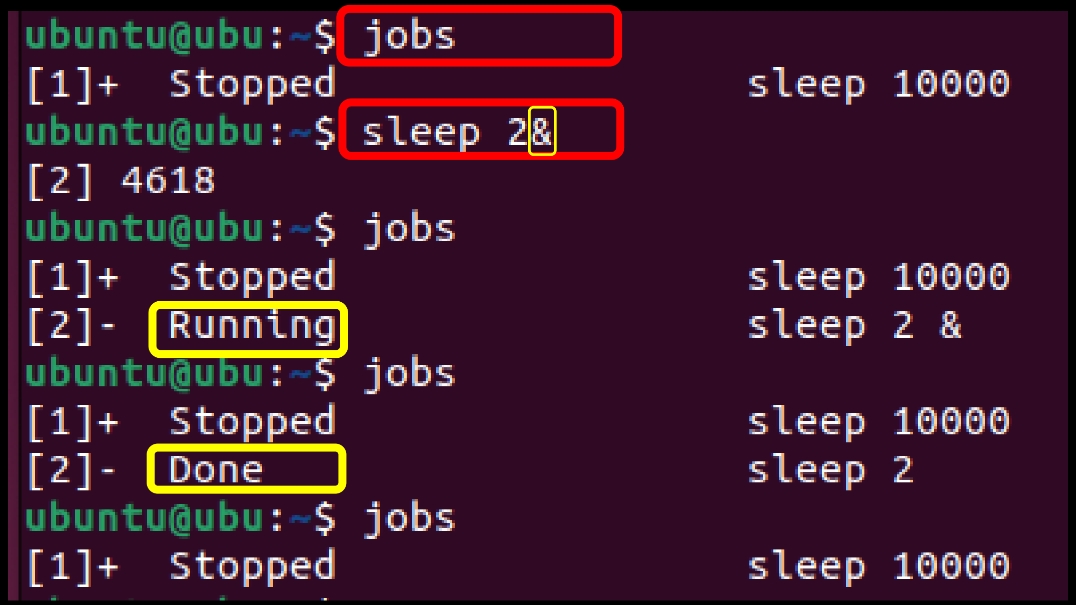
Process state

- **Running (R)**: The process is either executing instructions on the CPU or waiting in the run queue to be scheduled.
- Sleeping: The process is waiting for an event (like I/O) to complete
  - **Interruptible Sleep (S)**
  - **Uninterruptible Sleep (D)**
- **Stopped (T):** using `SIGSTOP` , `SIGTSTP` , `SIGTTIN` , or `SIGTTOU` signal, A stopped process can be resumed by sending it a `SIGCONT` signal or `Ctrl + Z` .
- **Zombie (Z)**: The process has completed execution, but still has an entry in the process table.
- **Idle (I)**: The process is waiting for an event to occur and is not actively using CPU resources.
- **Dead (X)**: The process has finished execution and has been removed from the process table.

Ctrl + Z → Stopped process



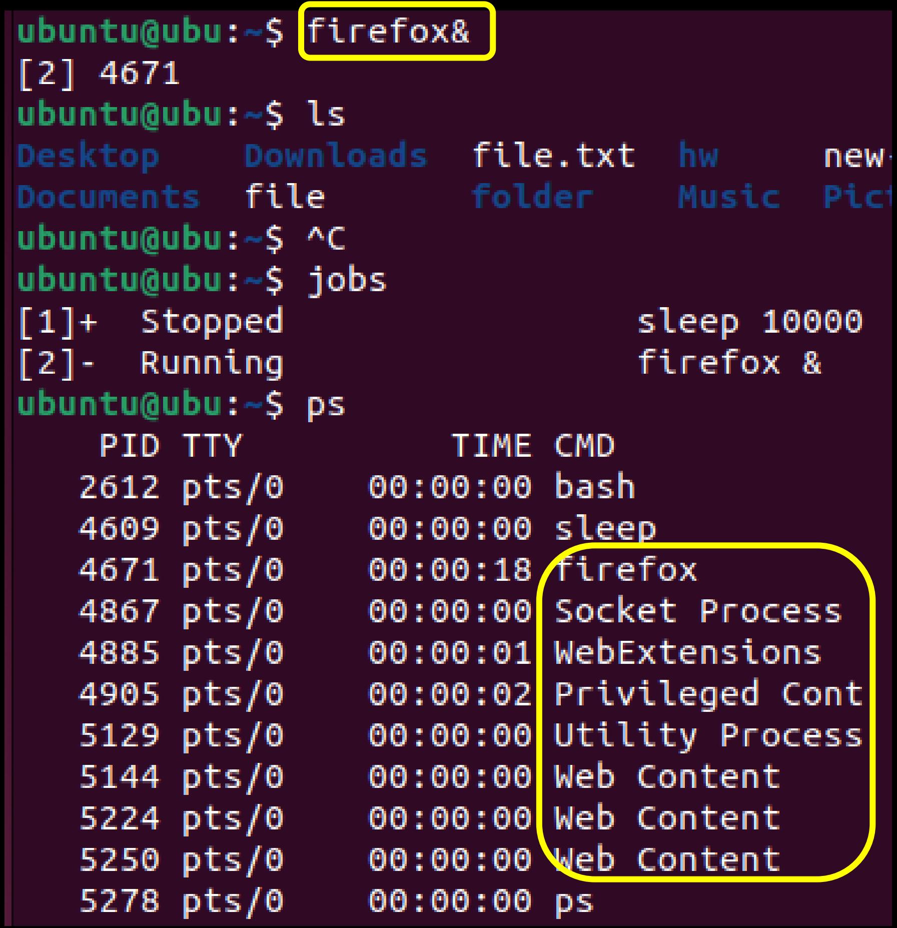
```
ubuntu@ubu:~$ sleep 3000
^Z
[1]+  Stopped                 sleep 3000
ubuntu@ubu:~$ jobs
[1]+  Stopped                 sleep 3000
ubuntu@ubu:~$ fg %1
sleep 3000
^C
```



```
ubuntu@ubu:~$ jobs
[1]+  Stopped                 sleep 10000
ubuntu@ubu:~$ sleep 2&
[2] 4618
ubuntu@ubu:~$ jobs
[1]+  Stopped                 sleep 10000
[2]-  Running                 sleep 2 &
ubuntu@ubu:~$ jobs
[1]+  Stopped                 sleep 10000
[2]-  Done                    sleep 2
ubuntu@ubu:~$ jobs
[1]+  Stopped                 sleep 10000
```

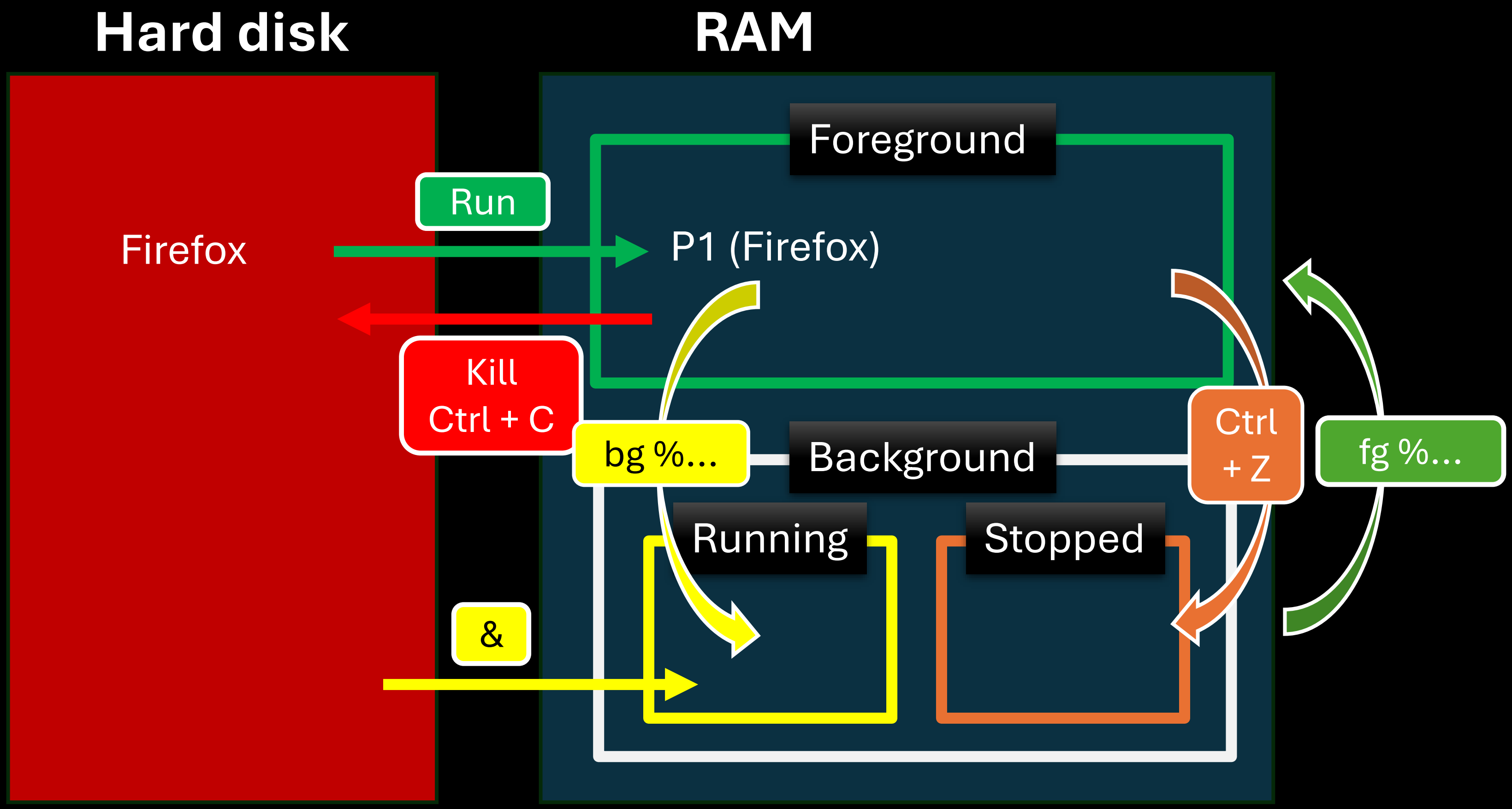
Firefox Child Processes

- **Socket Process:** This process handles network communication and is responsible for managing sockets. It helps in managing the data transfer between the browser and web servers.
- **WebExtensions**: These are extensions or add-ons that enhance the functionality of the browser. do not interfere with the main browser operations.
- **Privileged Content:** This process deals with content that requires special permissions or access
- **Utility Process:** for tasks that don't fit into the other categories, such as handling certain types of data or performing specific functions needed by the browser.
- **Web Content:** These processes are responsible for rendering web pages and handling the content displayed in the browser.



```
ubuntu@ubu:~$ firefox&
[2] 4671
ubuntu@ubu:~$ ls
Desktop     Downloads  file.txt  hw      new-
Documents   file       folder    Music   Pict
ubuntu@ubu:~$ ^C
ubuntu@ubu:~$ jobs
[1]+  Stopped                 sleep 10000
[2]-  Running                 firefox &
ubuntu@ubu:~$ ps
    PID TTY          TIME CMD
   2612 pts/0    00:00:00 bash
   4609 pts/0    00:00:00 sleep
   4671 pts/0    00:00:18 firefox
   4867 pts/0    00:00:00 Socket Process
   4885 pts/0    00:00:01 WebExtensions
   4905 pts/0    00:00:02 Privileged Cont
   5129 pts/0    00:00:00 Utility Process
   5144 pts/0    00:00:00 Web Content
   5224 pts/0    00:00:00 Web Content
   5250 pts/0    00:00:00 Web Content
   5278 pts/0    00:00:00 ps
```

Process State

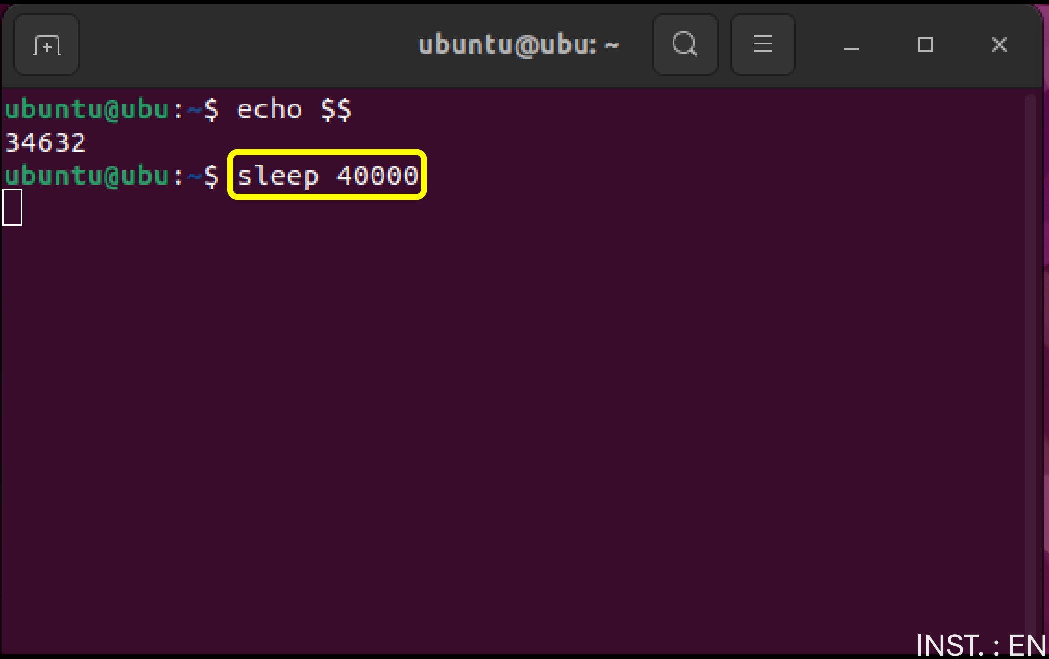


Socket Process

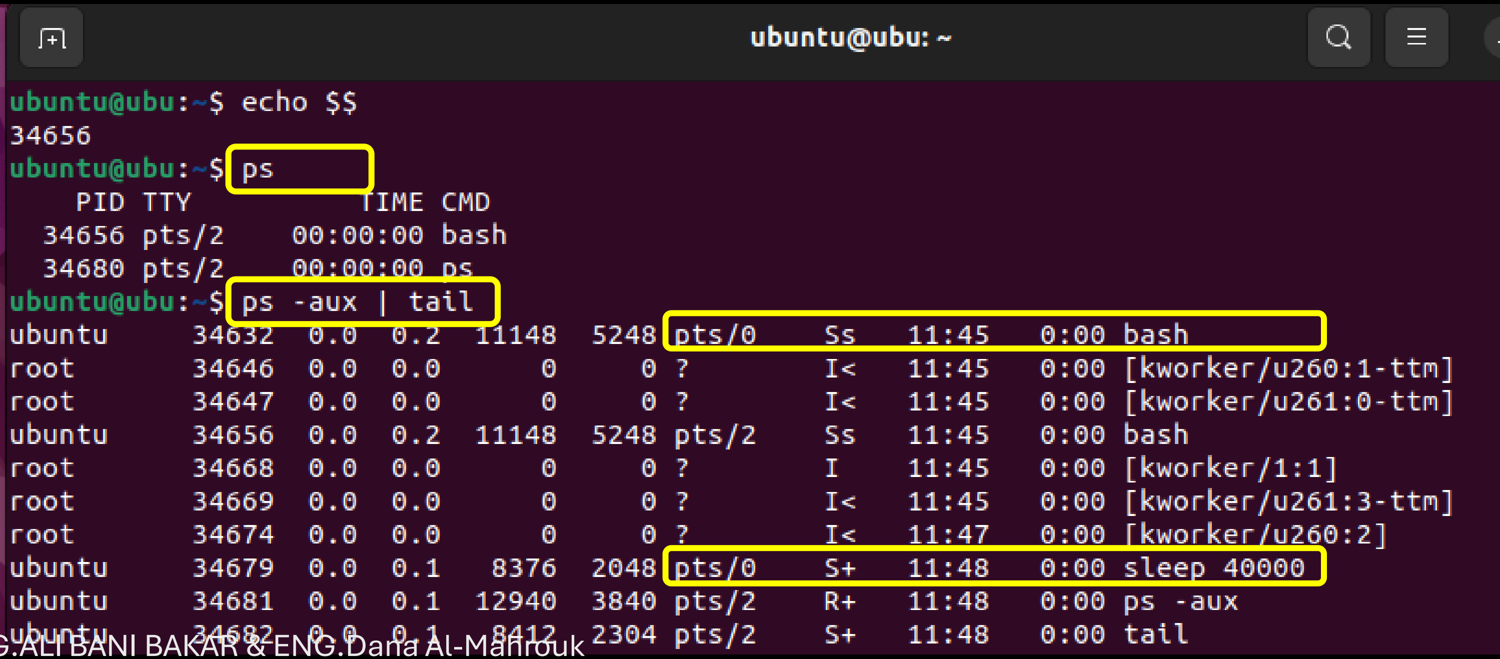
- manages the creation, maintenance, and closure **of network connections**.
- This **isolation** helps in ensuring that network-related issues or vulnerabilities don’t affect other parts of the browser.
- If there’s an issue with the network communication or an attempted attack, it is less likely to impact the entire browser or user data.

Ps -aux

- to display a snapshot of the current processes running on the system.
- -a: about all users' processes except session leaders.
- -u: detailed information about each process, including the user who owns the process.
- -x: Shows processes that are not running on a terminal (not connected to a TTY).

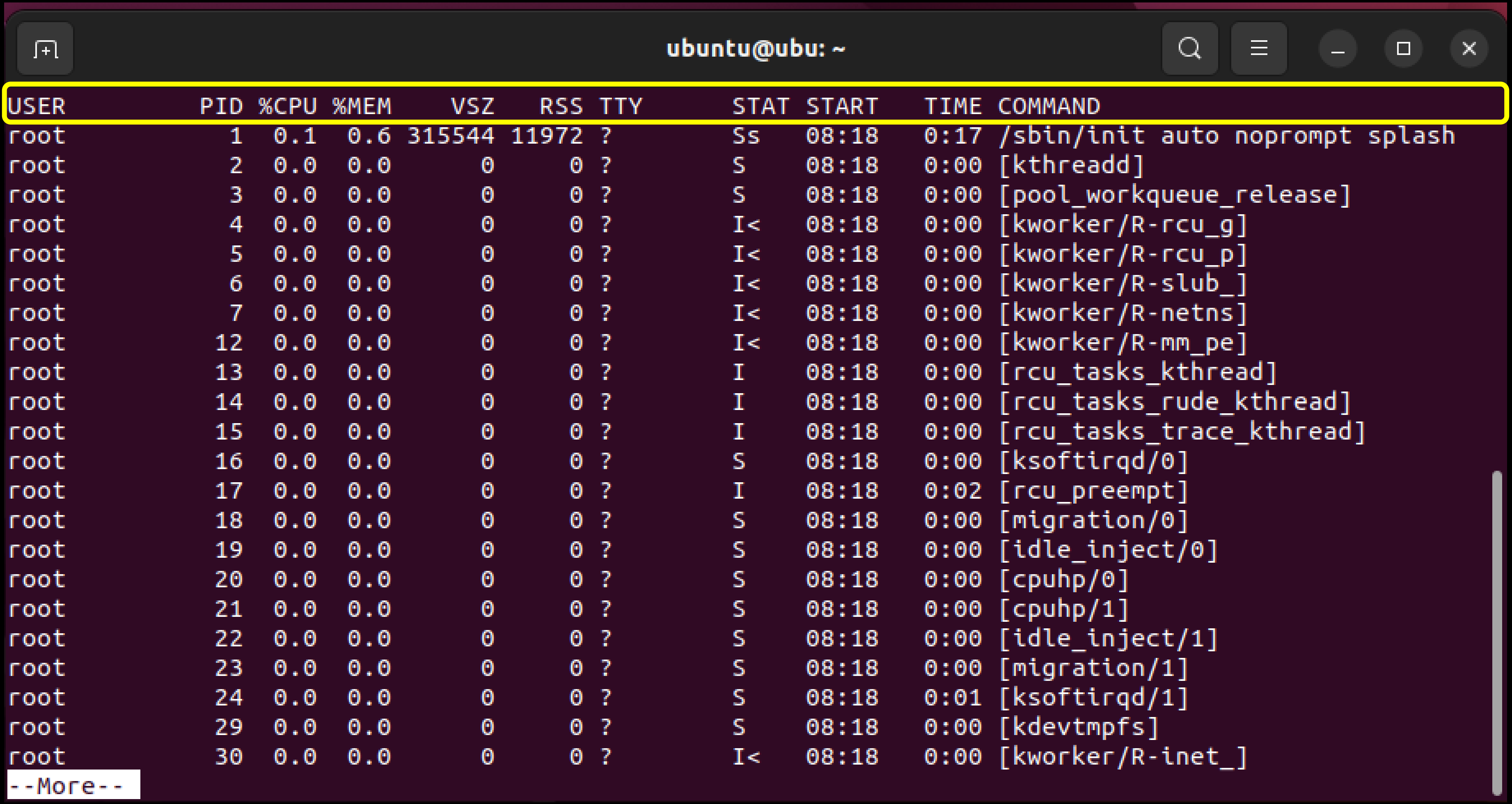


```
ubuntu@ubu:~$ echo $$
34632
ubuntu@ubu:~$ sleep 40000
```

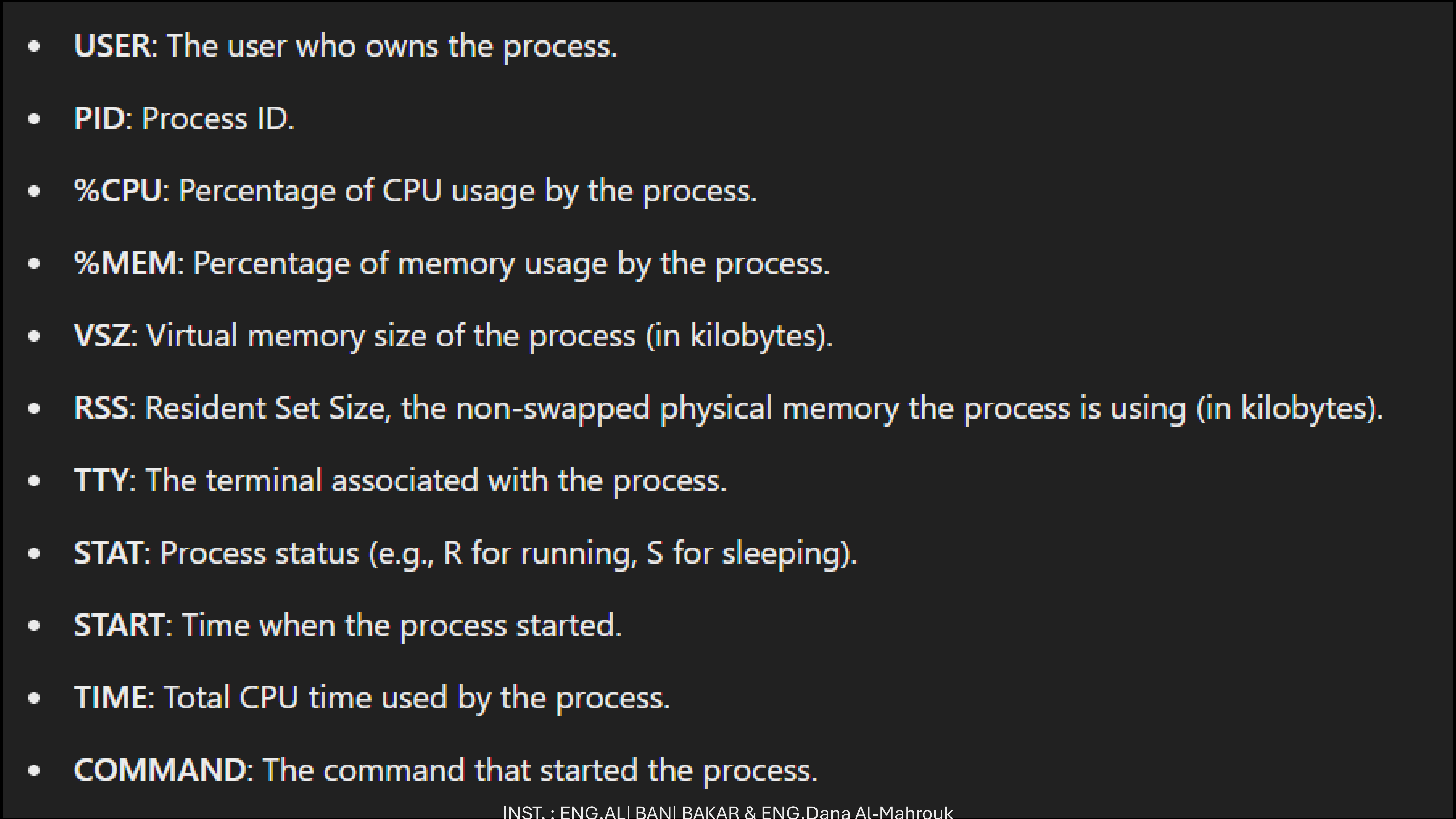


```
ubuntu@ubu:~$ echo $$
34656
ubuntu@ubu:~$ ps
    PID TTY          TIME CMD
  34656 pts/2    00:00:00 bash
  34680 pts/2    00:00:00 ps
ubuntu@ubu:~$ ps -aux | tail
ubuntu     34632  0.0  0.2  11148   5248 pts/0    Ss   11:45   0:00 bash
root       34646  0.0  0.0      0      0 ?        I<   11:45   0:00 [kworker/u260:1-ttm]
root       34647  0.0  0.0      0      0 ?        I<   11:45   0:00 [kworker/u261:0-ttm]
ubuntu     34656  0.0  0.2  11148   5248 pts/2    Ss   11:45   0:00 bash
root       34668  0.0  0.0      0      0 ?        I    11:45   0:00 [kworker/1:1]
root       34669  0.0  0.0      0      0 ?        I<   11:45   0:00 [kworker/u261:3-ttm]
root       34674  0.0  0.0      0      0 ?        I<   11:47   0:00 [kworker/u260:2]
ubuntu     34679  0.0  0.1   8376   2048 pts/0    S+   11:48   0:00 sleep 40000
ubuntu     34681  0.0  0.1  12940   3840 pts/2    R+   11:48   0:00 ps -aux
ubuntu     34682  0.0  0.1   8412   2304 pts/2    S+   11:48   0:00 tail
```

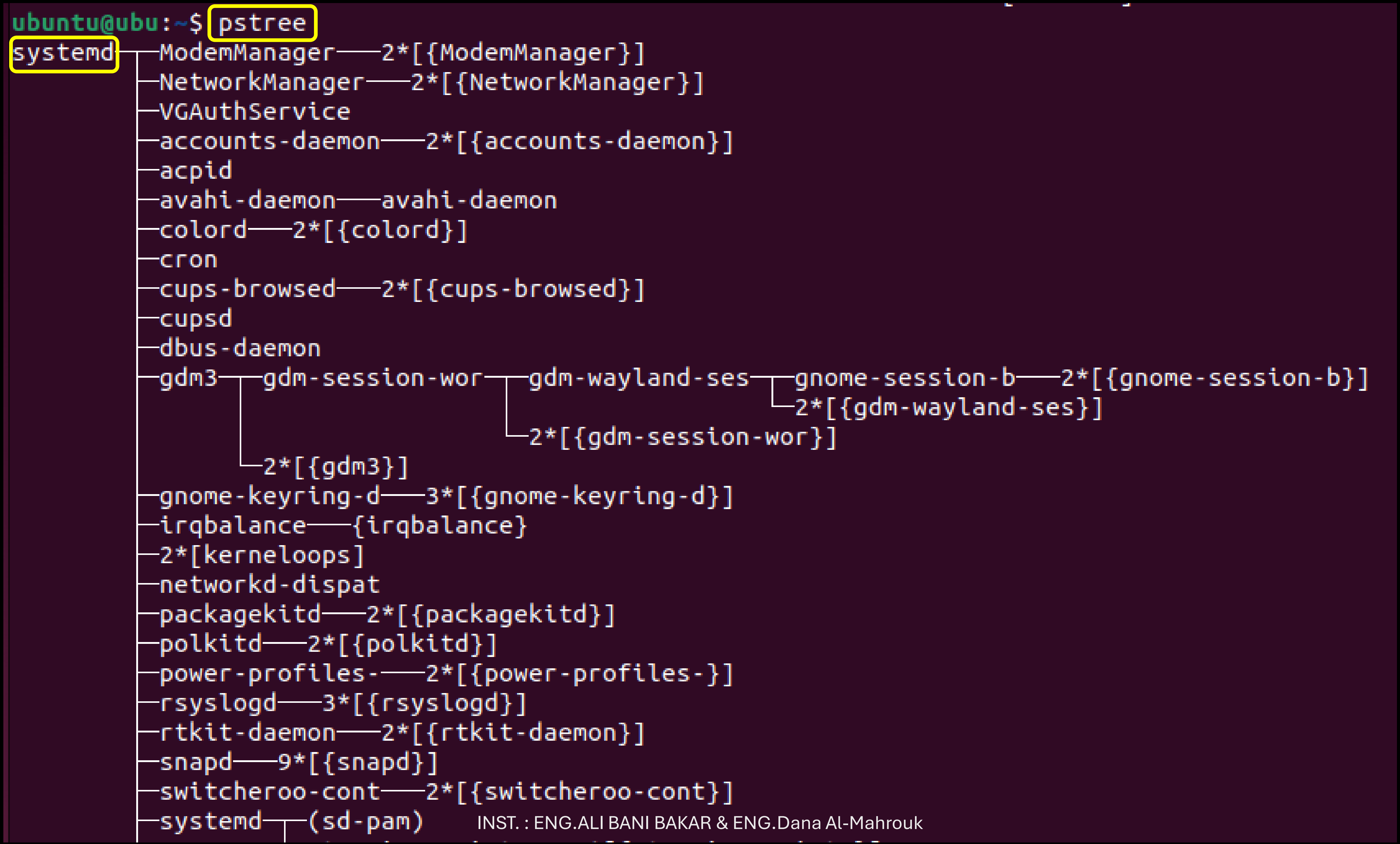
Ps –aux | more



```
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.6 315544 11972 ?        Ss   08:18   0:17 /sbin/init auto noprompt splash
root           2  0.0  0.0      0     0 ?        S    08:18   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    08:18   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   08:18   0:00 [kworker/R-rcu_g]
root           5  0.0  0.0      0     0 ?        I<   08:18   0:00 [kworker/R-rcu_p]
root           6  0.0  0.0      0     0 ?        I<   08:18   0:00 [kworker/R-slub_]
root           7  0.0  0.0      0     0 ?        I<   08:18   0:00 [kworker/R-netns]
root          12  0.0  0.0      0     0 ?        I<   08:18   0:00 [kworker/R-mm_pe]
root          13  0.0  0.0      0     0 ?        I    08:18   0:00 [rcu_tasks_kthread]
root          14  0.0  0.0      0     0 ?        I    08:18   0:00 [rcu_tasks_rude_kthread]
root          15  0.0  0.0      0     0 ?        I    08:18   0:00 [rcu_tasks_trace_kthread]
root          16  0.0  0.0      0     0 ?        S    08:18   0:00 [ksoftirqd/0]
root          17  0.0  0.0      0     0 ?        I    08:18   0:02 [rcu_preempt]
root          18  0.0  0.0      0     0 ?        S    08:18   0:00 [migration/0]
root          19  0.0  0.0      0     0 ?        S    08:18   0:00 [idle_inject/0]
root          20  0.0  0.0      0     0 ?        S    08:18   0:00 [cpuhp/0]
root          21  0.0  0.0      0     0 ?        S    08:18   0:00 [cpuhp/1]
root          22  0.0  0.0      0     0 ?        S    08:18   0:00 [idle_inject/1]
root          23  0.0  0.0      0     0 ?        S    08:18   0:00 [migration/1]
root          24  0.0  0.0      0     0 ?        S    08:18   0:01 [ksoftirqd/1]
root          29  0.0  0.0      0     0 ?        S    08:18   0:00 [kdevtmpfs]
root          30  0.0  0.0      0     0 ?        I<   08:18   0:00 [kworker/R-inet_]
--More--
```



- **USER**: The user who owns the process.
- **PID**: Process ID.
- **%CPU**: Percentage of CPU usage by the process.
- **%MEM**: Percentage of memory usage by the process.
- **VSZ**: Virtual memory size of the process (in kilobytes).
- **RSS**: Resident Set Size, the non-swapped physical memory the process is using (in kilobytes).
- **TTY**: The terminal associated with the process.
- **STAT**: Process status (e.g., R for running, S for sleeping).
- **START**: Time when the process started.
- **TIME**: Total CPU time used by the process.
- **COMMAND**: The command that started the process.



```
ubuntu@ubu:~$ pstree
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─NetworkManager───2*[{NetworkManager}]
        ├─VGAuthService
        ├─accounts-daemon───2*[{accounts-daemon}]
        ├─acpid
        ├─avahi-daemon───avahi-daemon
        ├─colord───2*[{colord}]
        ├─cron
        ├─cups-browsed───2*[{cups-browsed}]
        ├─cupsd
        ├─dbus-daemon
        ├─gdm3─┬─gdm-session-wor─┬─gdm-wayland-ses─┬─gnome-session-b───2*[{gnome-session-b}]
        │      │                 │                 └─2*[{gdm-wayland-ses}]
        │      │                 └─2*[{gdm-session-wor}]
        │      └─2*[{gdm3}]
        ├─gnome-keyring-d───3*[{gnome-keyring-d}]
        ├─irqbalance───{irqbalance}
        ├─2*[kerneloops]
        ├─networkd-dispat
        ├─packagekitd───2*[{packagekitd}]
        ├─polkitd───2*[{polkitd}]
        ├─power-profiles-───2*[{power-profiles-}]
        ├─rsyslogd───3*[{rsyslogd}]
        ├─rtkit-daemon───2*[{rtkit-daemon}]
        ├─snapd───9*[{snapd}]
        ├─switcheroo-cont───2*[{switcheroo-cont}]
        ├─systemd─┬─(sd-pam)
```

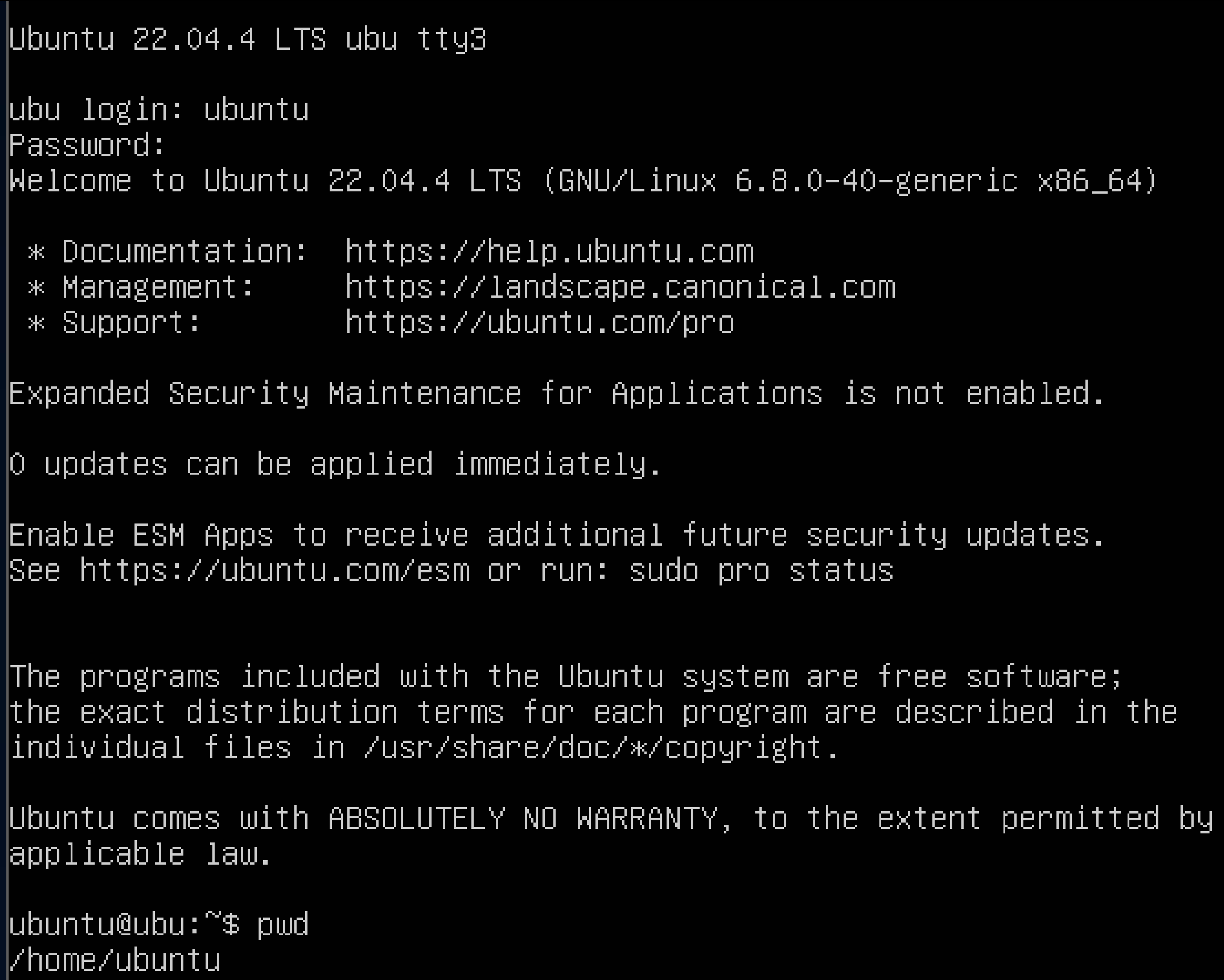
Performance Linux GUI VS TTY

- TTYs are text-based and don't have the graphical overhead of a GUI, which means they use fewer system resources.
- If the GUI becomes unresponsive or crashes, TTYs can be used to troubleshoot and recover the system.
- TTYs provide a direct command-line interface without any graphical layer, which might be more straightforward for certain administrative tasks.

virtual terminal TTY

- Ctrl + alt + f<3-6>
- Ctrl + alt + f2: return to your gui terminal
- TTY1
- TTY2
- TTY3-6

- \`logout\` or \`Ctrl + D\` to end your session in the TTY and return to the login prompt.



```
Ubuntu 22.04.4 LTS ubu tty3

ubu login: ubuntu
Password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.8.0-40-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ubu:~$ pwd
/home/ubuntu
```

Systemd

- Systemd: is an init system and system manager used in Linux OS. It serves as the first process that starts during boot and is responsible for managing all other processes and services running on the system.
- **Units:**
  - (.service): Define services that can be started, stopped, or restarted.
  - (.socket): Manage socket activation for services.
  - (.target): Group together units for a particular state or goal.
  - (.mount): Handle file system mount points.
  - (.timer): Trigger services based on time or intervals.
- **journald**: logging service, it collects and stores log data, which can be accessed using the `journalctl` command.

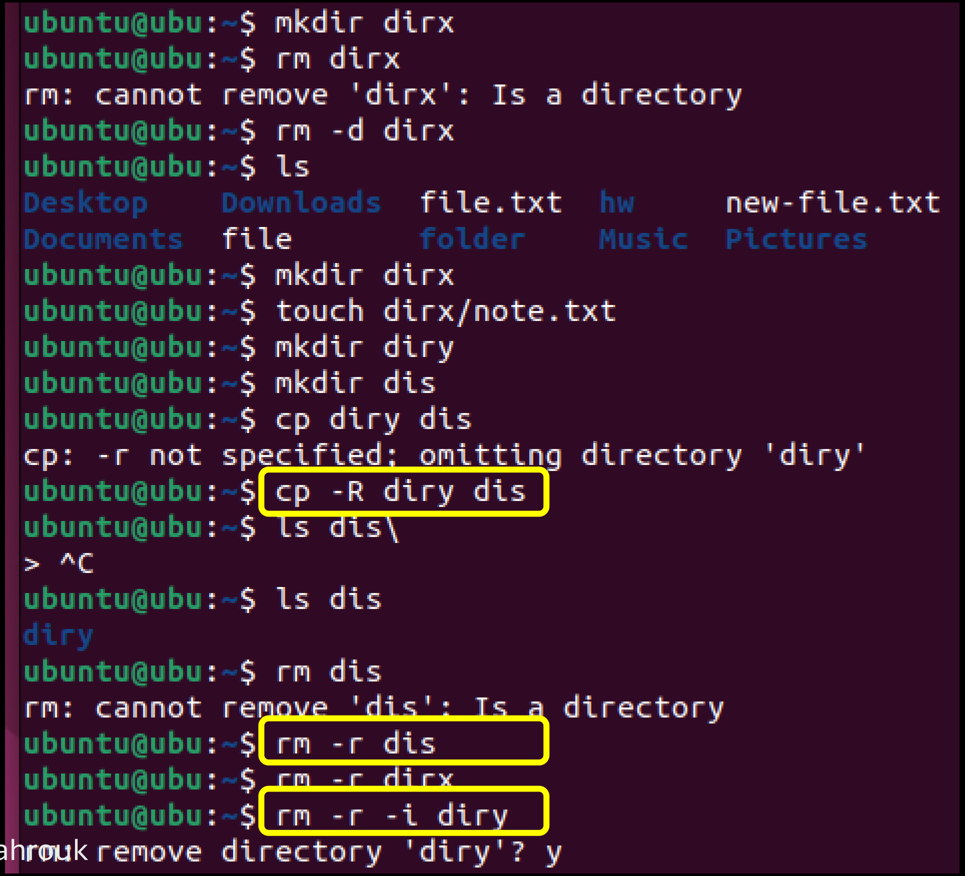
Service Management

- `systemctl` is the primary command for interacting with systemd units.
- Examples:
  - `systemctl start [service]`: Start a service.
  - `systemctl stop [service]`: Stop a service.
  - `systemctl restart [service]`: Restart a service.
  - `systemctl status [service]`: Check the status of a service.
  - `systemctl enable [service]`: Enable a service to start on boot.
  - `systemctl disable [service]`: Disable a service from starting on boot.
- **Dependency Management**: Systemd manages dependencies between units. It ensures that required units are started in the correct order based on dependencies.
- **Cgroups**: Systemd uses control groups (cgroups) to manage resources (CPU, memory, I/O, etc.) allocated to services. This helps in tracking and limiting resource usage.
- **Timers**: Systemd can replace traditional cron jobs with timer units.

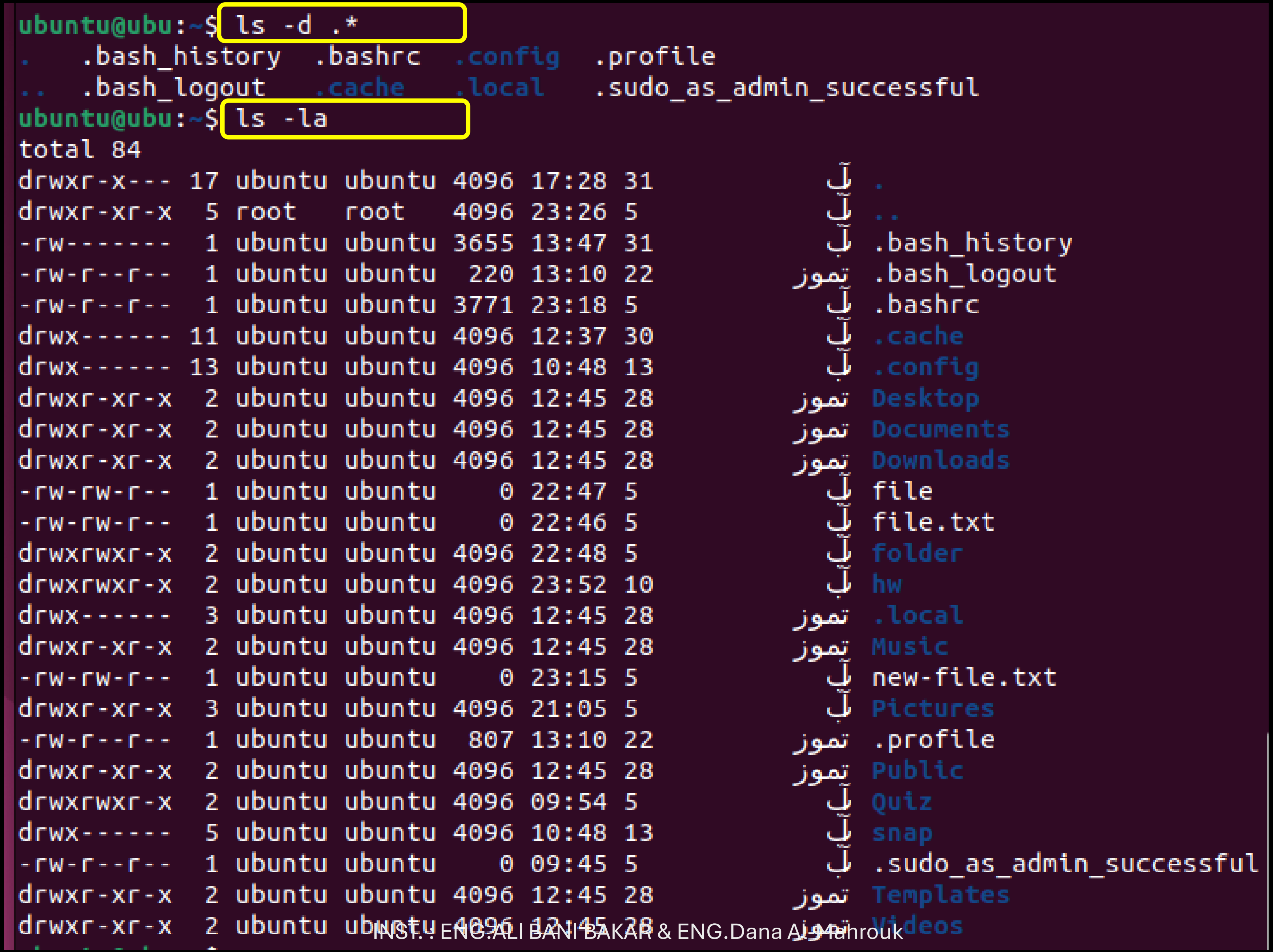
cp (Copy)

- cp [options] source destination
- copy files and directories from one location to another.
- \`-r\` : copy directories and their contents.

- rm (Remove)
- remove (delete) files and directories.
- \`-r\` : delete directories and their contents.
- \`-i\` : Prompt for confirmation before each file is deleted.



```
ubuntu@ubu:~$ mkdir dirx
ubuntu@ubu:~$ rm dirx
rm: cannot remove 'dirx': Is a directory
ubuntu@ubu:~$ rm -d dirx
ubuntu@ubu:~$ ls
Desktop     Downloads  file.txt  hw     new-file.txt
Documents   file       folder    Music  Pictures
ubuntu@ubu:~$ mkdir dirx
ubuntu@ubu:~$ touch dirx/note.txt
ubuntu@ubu:~$ mkdir diry
ubuntu@ubu:~$ mkdir dis
ubuntu@ubu:~$ cp diry dis
cp: -r not specified; omitting directory 'diry'
ubuntu@ubu:~$ cp -R diry dis
ubuntu@ubu:~$ ls dis\
> ^C
ubuntu@ubu:~$ ls dis
diry
ubuntu@ubu:~$ rm dis
rm: cannot remove 'dis': Is a directory
ubuntu@ubu:~$ rm -r dis
ubuntu@ubu:~$ rm -r dirx
ubuntu@ubu:~$ rm -r -i diry
rm: remove directory 'diry'? y
```

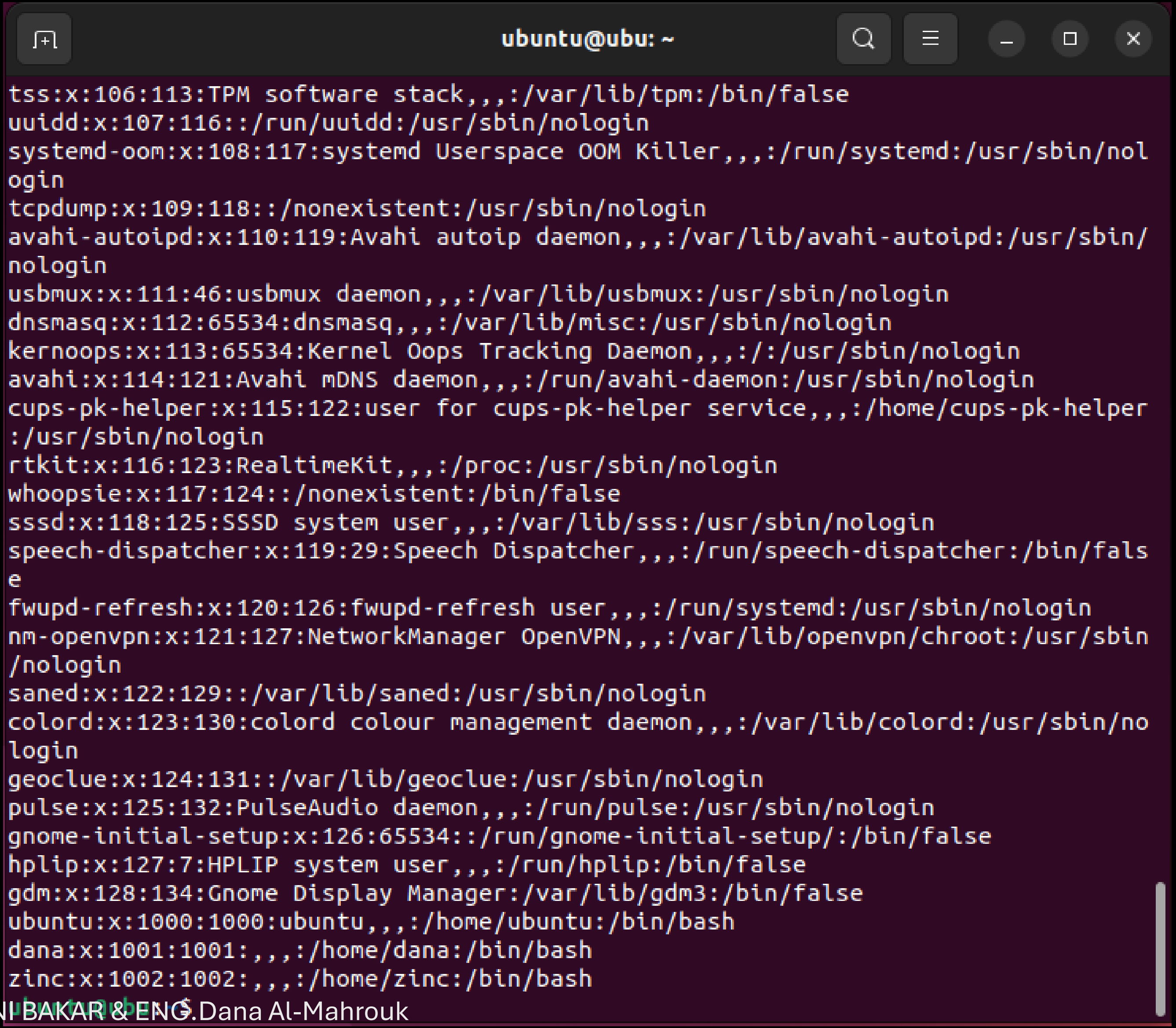


```
ubuntu@ubu:~$ ls -d .*
.    .bash_history  .bashrc  .config  .profile
..   .bash_logout   .cache   .local   .sudo_as_admin_successful
ubuntu@ubu:~$ ls -la
total 84
drwxr-x--- 17 ubuntu ubuntu 4096 17:28 31          آب  .
drwxr-xr-x  5 root   root   4096 23:26 5           آب  ..
-rw-------  1 ubuntu ubuntu 3655 13:47 31          آب  .bash_history
-rw-r--r--  1 ubuntu ubuntu  220 13:10 22        تموز  .bash_logout
-rw-r--r--  1 ubuntu ubuntu 3771 23:18 5           آب  .bashrc
drwx------ 11 ubuntu ubuntu 4096 12:37 30          آب  .cache
drwx------ 13 ubuntu ubuntu 4096 10:48 13          آب  .config
drwxr-xr-x  2 ubuntu ubuntu 4096 12:45 28        تموز  Desktop
drwxr-xr-x  2 ubuntu ubuntu 4096 12:45 28        تموز  Documents
drwxr-xr-x  2 ubuntu ubuntu 4096 12:45 28        تموز  Downloads
-rw-rw-r--  1 ubuntu ubuntu    0 22:47 5           آب  file
-rw-rw-r--  1 ubuntu ubuntu    0 22:46 5           آب  file.txt
drwxrwxr-x  2 ubuntu ubuntu 4096 22:48 5           آب  folder
drwxrwxr-x  2 ubuntu ubuntu 4096 23:52 10          آب  hw
drwx------  3 ubuntu ubuntu 4096 12:45 28        تموز  .local
drwxr-xr-x  2 ubuntu ubuntu 4096 12:45 28        تموز  Music
-rw-rw-r--  1 ubuntu ubuntu    0 23:15 5           آب  new-file.txt
drwxr-xr-x  3 ubuntu ubuntu 4096 21:05 5           آب  Pictures
-rw-r--r--  1 ubuntu ubuntu  807 13:10 22        تموز  .profile
drwxr-xr-x  2 ubuntu ubuntu 4096 12:45 28        تموز  Public
drwxrwxr-x  2 ubuntu ubuntu 4096 09:54 5           آب  Quiz
drwx------  5 ubuntu ubuntu 4096 10:48 13          آب  snap
-rw-r--r--  1 ubuntu ubuntu    0 09:45 5           آب  .sudo_as_admin_successful
drwxr-xr-x  2 ubuntu ubuntu 4096 12:45 28        تموز  Templates
drwxr-xr-x  2 ubuntu ubuntu 4096 12:45 28        تموز  Videos
```

cat VS more VS less

- **Cat**: is used to display the entire content of a file(s) to the standard output (usually the terminal).

- Displays the entire content of a file at once.
- Useful for viewing small files.
- Can concatenate multiple files and display them together.

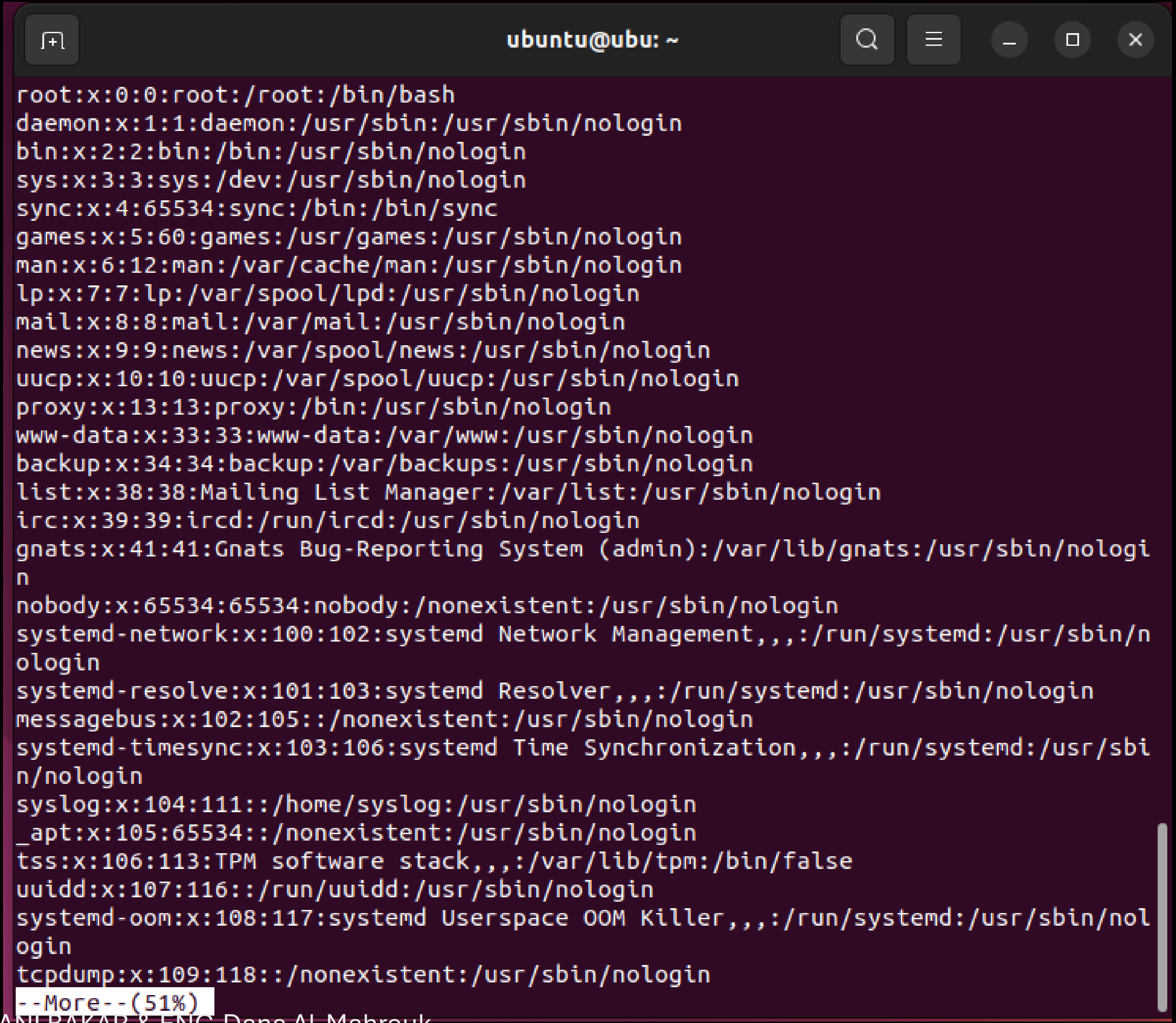


```
tss:x:106:113:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:116::/run/uuidd:/usr/sbin/nologin
systemd-oom:x:108:117:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nol
ogin
tcpdump:x:109:118::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/
nologin
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
avahi:x:114:121:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:115:122:user for cups-pk-helper service,,,:/home/cups-pk-helper
:/usr/sbin/nologin
rtkit:x:116:123:RealtimeKit,,,:/proc:/usr/sbin/nologin
whoopsie:x:117:124::/nonexistent:/bin/false
sssd:x:118:125:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
speech-dispatcher:x:119:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/fals
e
fwupd-refresh:x:120:126:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
nm-openvpn:x:121:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin
/nologin
saned:x:122:129::/var/lib/saned:/usr/sbin/nologin
colord:x:123:130:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/no
login
geoclue:x:124:131::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:125:132:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:126:65534::/run/gnome-initial-setup/:/bin/false
hplip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
dana:x:1001:1001:,,,:/home/dana:/bin/bash
zinc:x:1002:1002:,,,:/home/zinc:/bin/bash
ubuntu@ubu:~$
```

more

- is a simple pager that allows you to view the content of a file one screen at a time.

- Displays content page by page.
- Allows forward navigation by pressing the space bar (one screen) or Enter (one line).
- Cannot scroll backward.

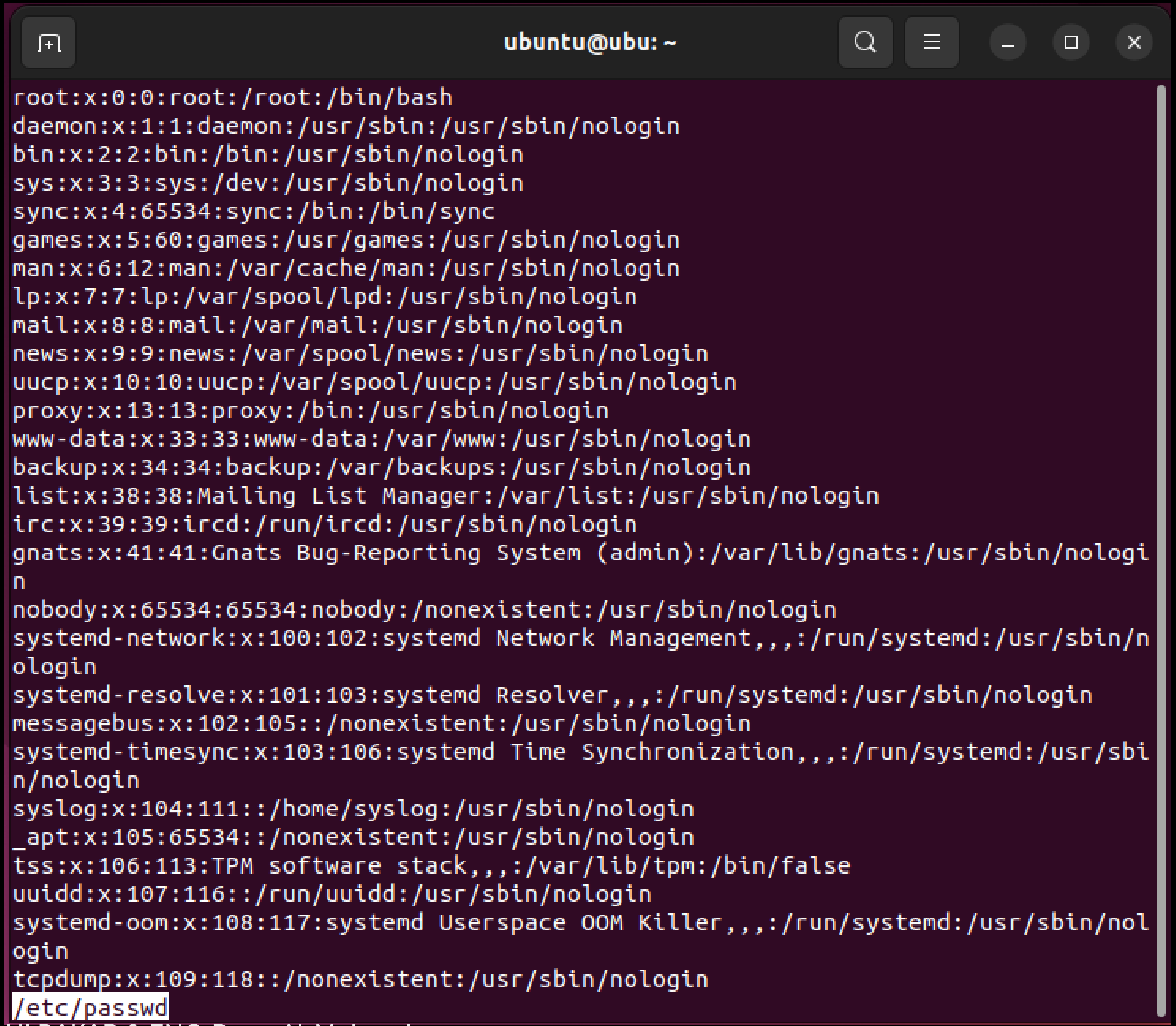


```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologi
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/n
ologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbi
n/nologin
syslog:x:104:111::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:113:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:116::/run/uuidd:/usr/sbin/nologin
systemd-oom:x:108:117:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nol
ogin
tcpdump:x:109:118::/nonexistent:/usr/sbin/nologin
--More--(51%)
```

less

- is a more advanced pager that allows you to view the content of a file with more control.

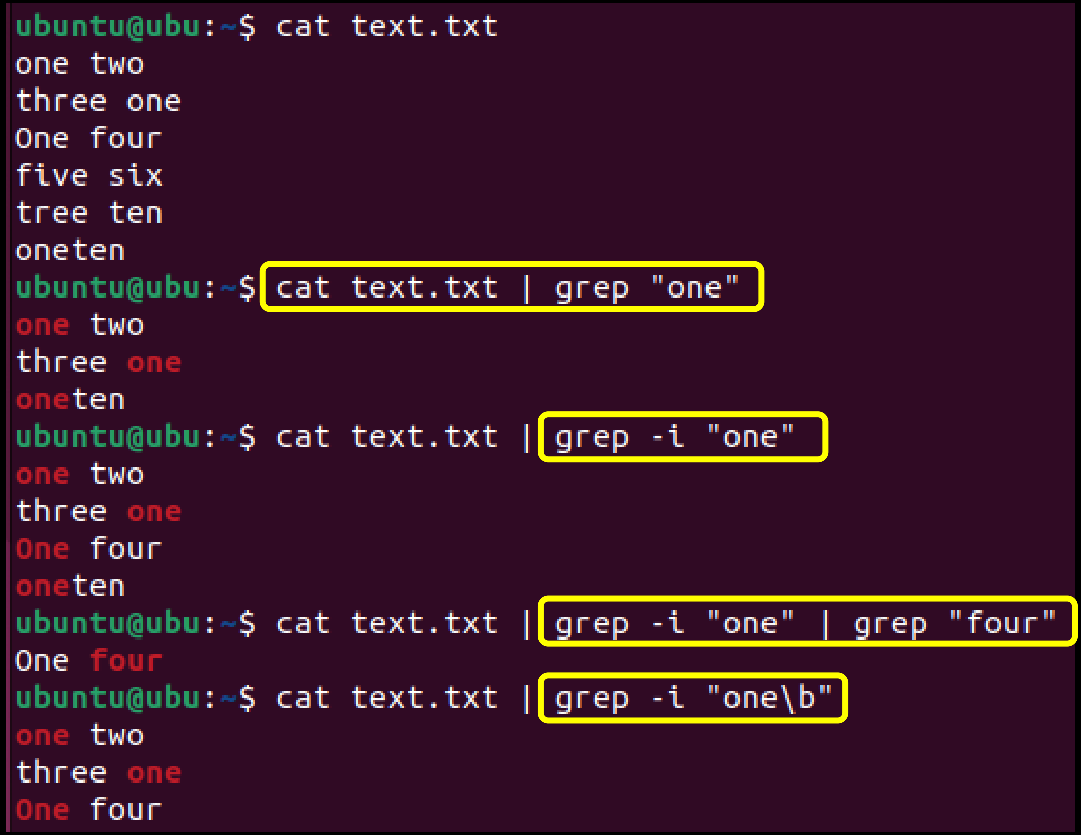
- Displays content one page at a time.
- Allows both forward and backward navigation (using arrow keys).
- Search within the file (using ` / ` followed by the search term).
- Can scroll through content with page up/page down or arrow keys.
- Efficient for viewing large files.
- Doesn't load the entire file into memory, so it is faster for large files.



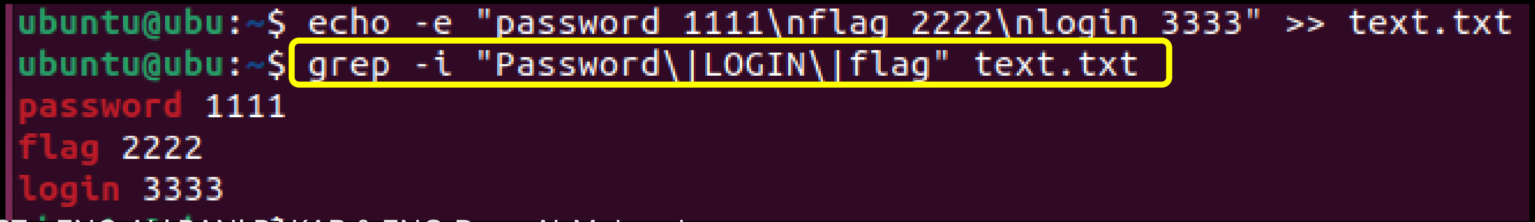
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologi
n
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/n
ologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbi
n/nologin
syslog:x:104:111::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:113:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:116::/run/uuidd:/usr/sbin/nologin
systemd-oom:x:108:117:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nol
ogin
tcpdump:x:109:118::/nonexistent:/usr/sbin/nologin
/etc/passwd
```

Grep

- search for patterns within files or input.
- `-i`: Ignore case distinctions (case-insensitive search)
- `-r`, `-R`: Recursively search directories for the pattern.
- `\b`: specify a word boundary
- `\|`: alternation, multiple patterns to match. "OR" operator.

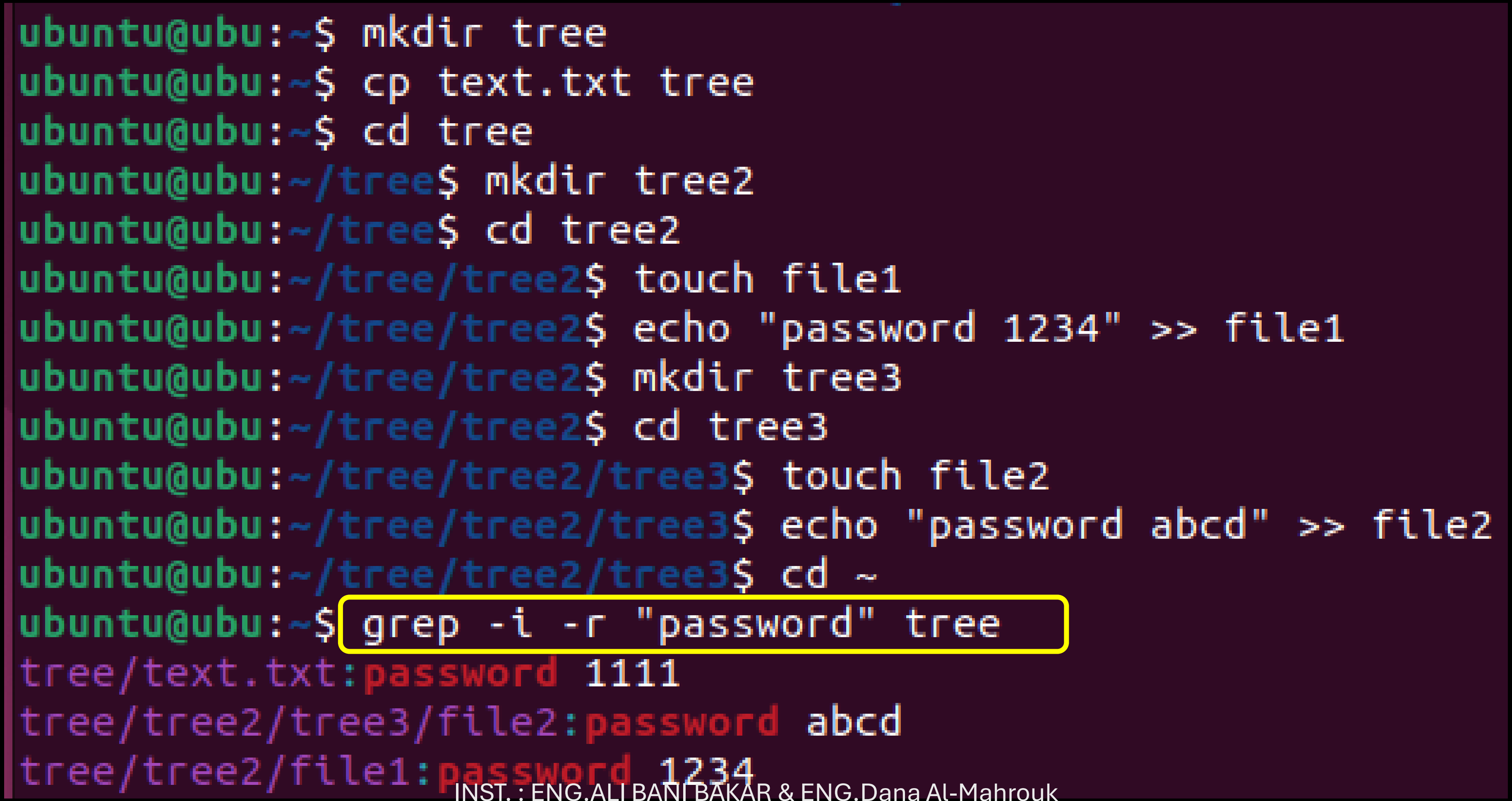


```
ubuntu@ubu:~$ cat text.txt
one two
three one
One four
five six
tree ten
oneten
ubuntu@ubu:~$ cat text.txt | grep "one"
one two
three one
oneten
ubuntu@ubu:~$ cat text.txt |grep -i "one"
one two
three one
One four
oneten
ubuntu@ubu:~$ cat text.txt |grep -i "one" | grep "four"
One four
ubuntu@ubu:~$ cat text.txt |grep -i "one\b"
one two
three one
One four
```



```
ubuntu@ubu:~$ echo -e "password 1111\nflag 2222\nlogin 3333" >> text.txt
ubuntu@ubu:~$ grep -i "Password\|LOGIN\|flag" text.txt
password 1111
flag 2222
login 3333
```

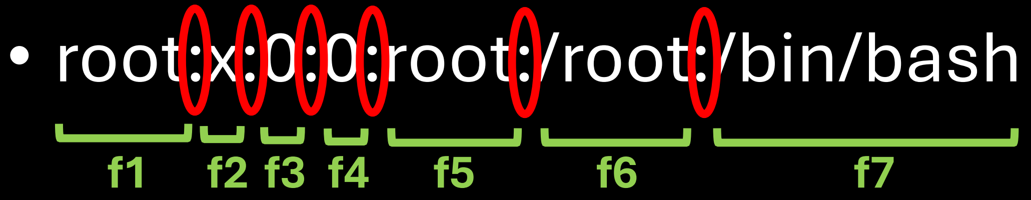
Grep -i -r



```
ubuntu@ubu:~$ mkdir tree
ubuntu@ubu:~$ cp text.txt tree
ubuntu@ubu:~$ cd tree
ubuntu@ubu:~/tree$ mkdir tree2
ubuntu@ubu:~/tree$ cd tree2
ubuntu@ubu:~/tree/tree2$ touch file1
ubuntu@ubu:~/tree/tree2$ echo "password 1234" >> file1
ubuntu@ubu:~/tree/tree2$ mkdir tree3
ubuntu@ubu:~/tree/tree2$ cd tree3
ubuntu@ubu:~/tree/tree2/tree3$ touch file2
ubuntu@ubu:~/tree/tree2/tree3$ echo "password abcd" >> file2
ubuntu@ubu:~/tree/tree2/tree3$ cd ~
ubuntu@ubu:~$ grep -i -r "password" tree
tree/text.txt:password 1111
tree/tree2/tree3/file2:password abcd
tree/tree2/file1:password 1234
```

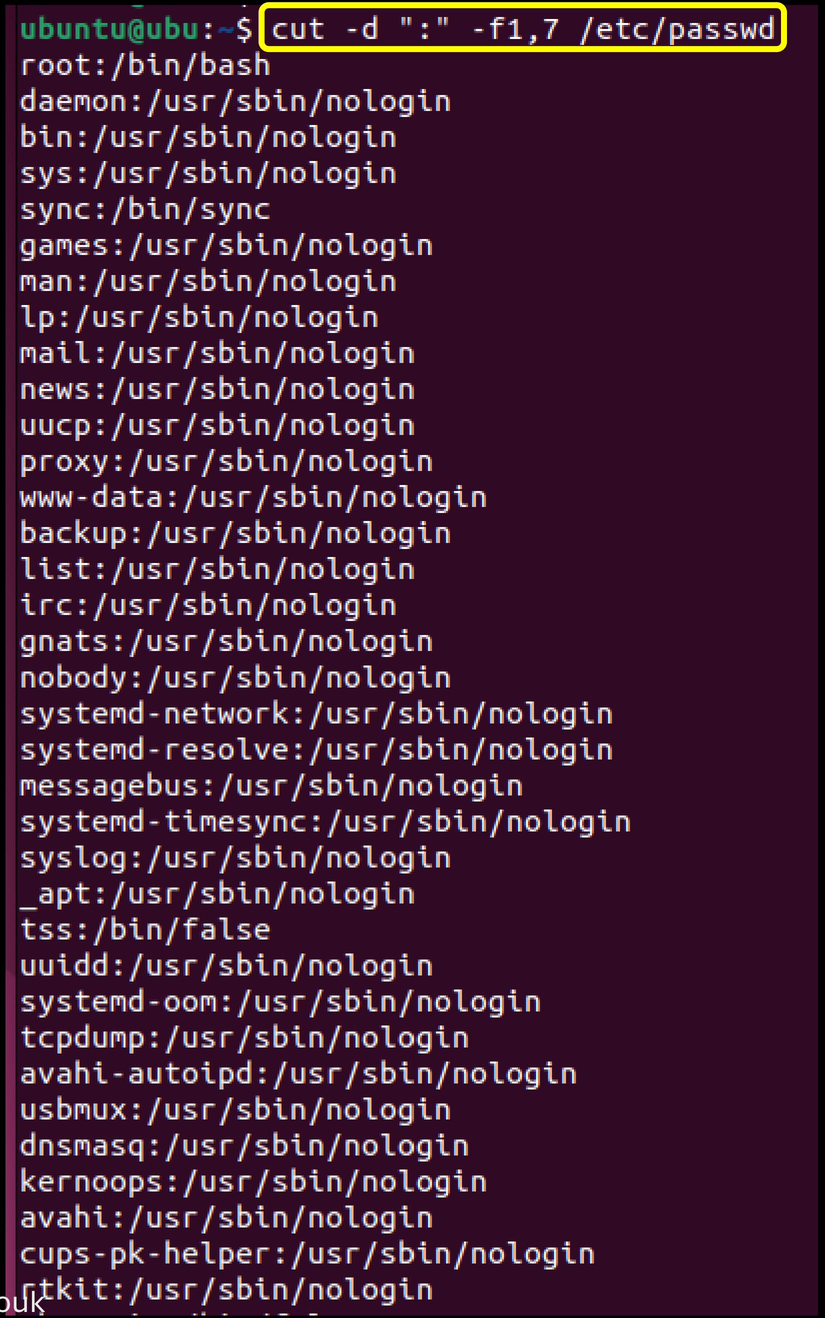
Cut

- to extract specific sections from each line of a file or input.



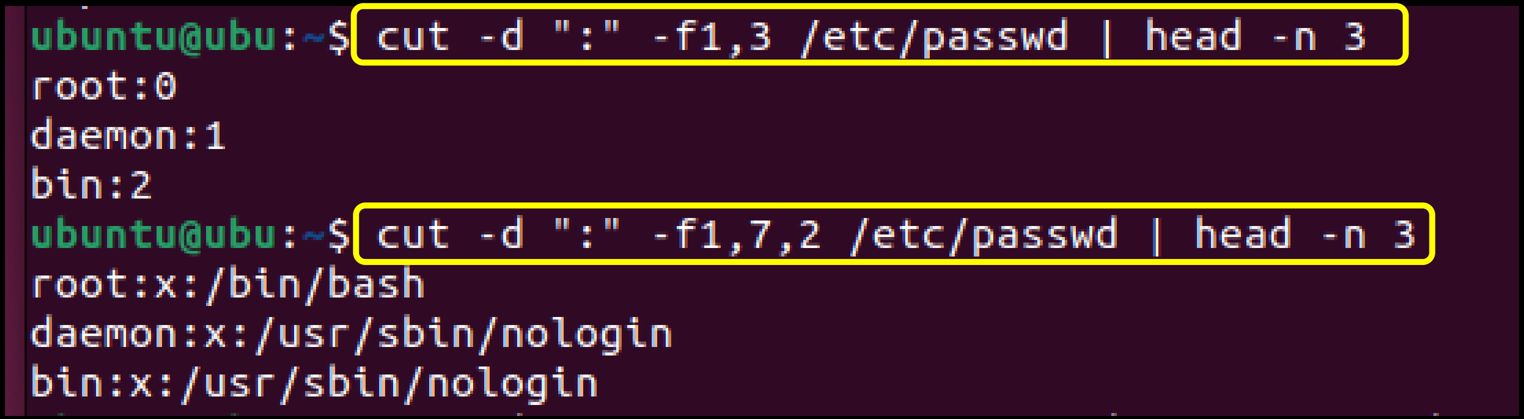
- root:x:0:0:root:/root:/bin/bash

- `-f` : Specifies the fields to extract. Fields are defined by a delimiter
- `-d` : Specifies the delimiter that separates fields.

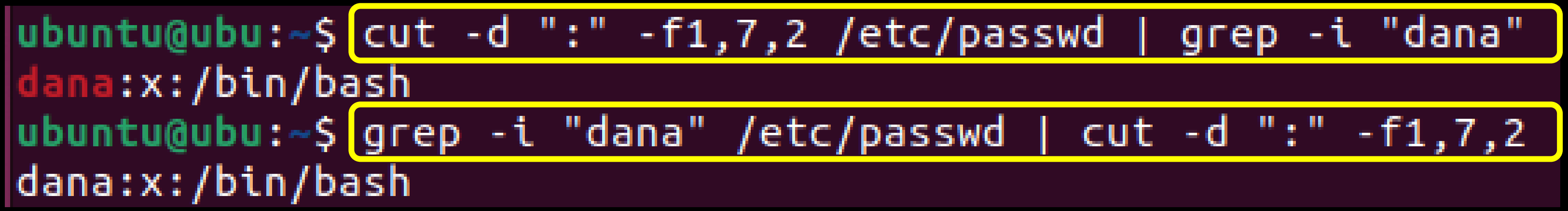


```
ubuntu@ubu:~$ cut -d ":" -f1,7 /etc/passwd
root:/bin/bash
daemon:/usr/sbin/nologin
bin:/usr/sbin/nologin
sys:/usr/sbin/nologin
sync:/bin/sync
games:/usr/sbin/nologin
man:/usr/sbin/nologin
lp:/usr/sbin/nologin
mail:/usr/sbin/nologin
news:/usr/sbin/nologin
uucp:/usr/sbin/nologin
proxy:/usr/sbin/nologin
www-data:/usr/sbin/nologin
backup:/usr/sbin/nologin
list:/usr/sbin/nologin
irc:/usr/sbin/nologin
gnats:/usr/sbin/nologin
nobody:/usr/sbin/nologin
systemd-network:/usr/sbin/nologin
systemd-resolve:/usr/sbin/nologin
messagebus:/usr/sbin/nologin
systemd-timesync:/usr/sbin/nologin
syslog:/usr/sbin/nologin
_apt:/usr/sbin/nologin
tss:/bin/false
uuidd:/usr/sbin/nologin
systemd-oom:/usr/sbin/nologin
tcpdump:/usr/sbin/nologin
avahi-autoipd:/usr/sbin/nologin
usbmux:/usr/sbin/nologin
dnsmasq:/usr/sbin/nologin
kernoops:/usr/sbin/nologin
avahi:/usr/sbin/nologin
cups-pk-helper:/usr/sbin/nologin
rtkit:/usr/sbin/nologin
```

Ex:



```
ubuntu@ubu:~$ cut -d ":" -f1,3 /etc/passwd | head -n 3
root:0
daemon:1
bin:2
ubuntu@ubu:~$ cut -d ":" -f1,7,2 /etc/passwd | head -n 3
root:x:/bin/bash
daemon:x:/usr/sbin/nologin
bin:x:/usr/sbin/nologin
```

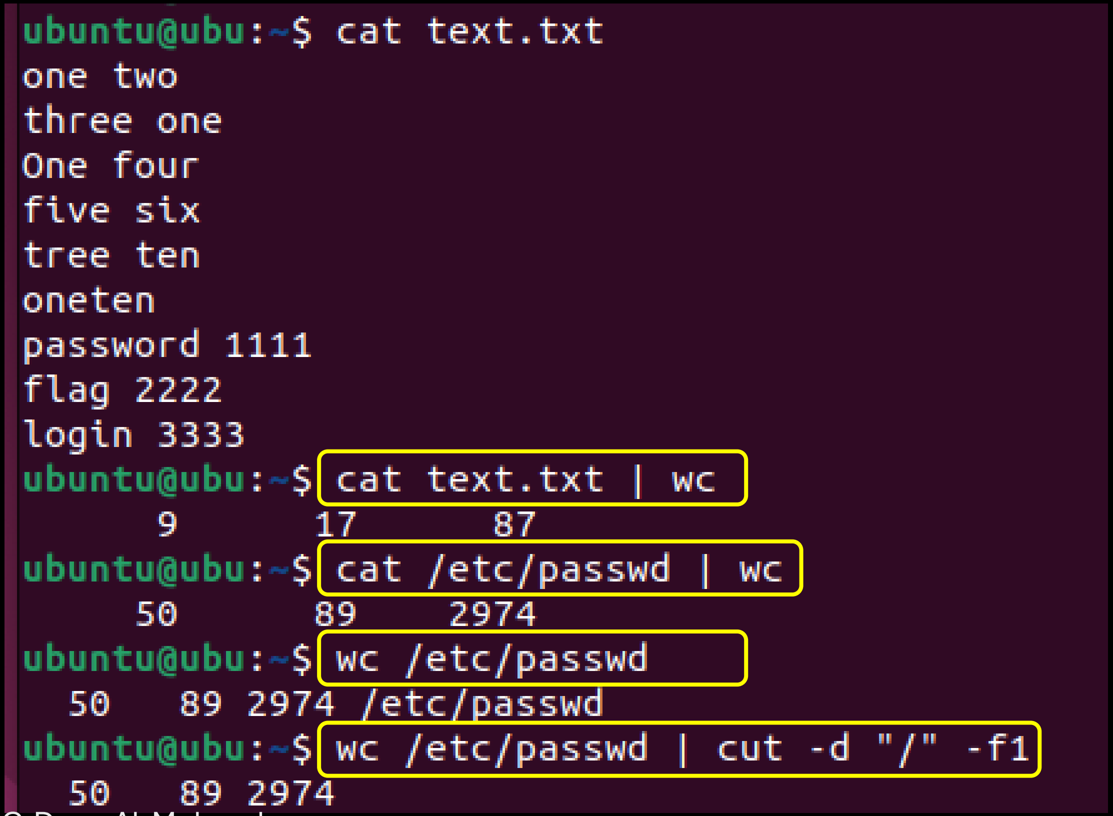


```
ubuntu@ubu:~$ cut -d ":" -f1,7,2 /etc/passwd | grep -i "dana"
dana:x:/bin/bash
ubuntu@ubu:~$ grep -i "dana" /etc/passwd | cut -d ":" -f1,7,2
dana:x:/bin/bash
```

wc (word count)

- (word count) command used to count lines, words, and characters in files.

- `-l` : Count the number of lines.
- `-w` : Count the number of words.
- `-c` : Count the number of bytes (characters).
- `-m` : Count the number of characters .
- `-L` : Display the length of the longest line.

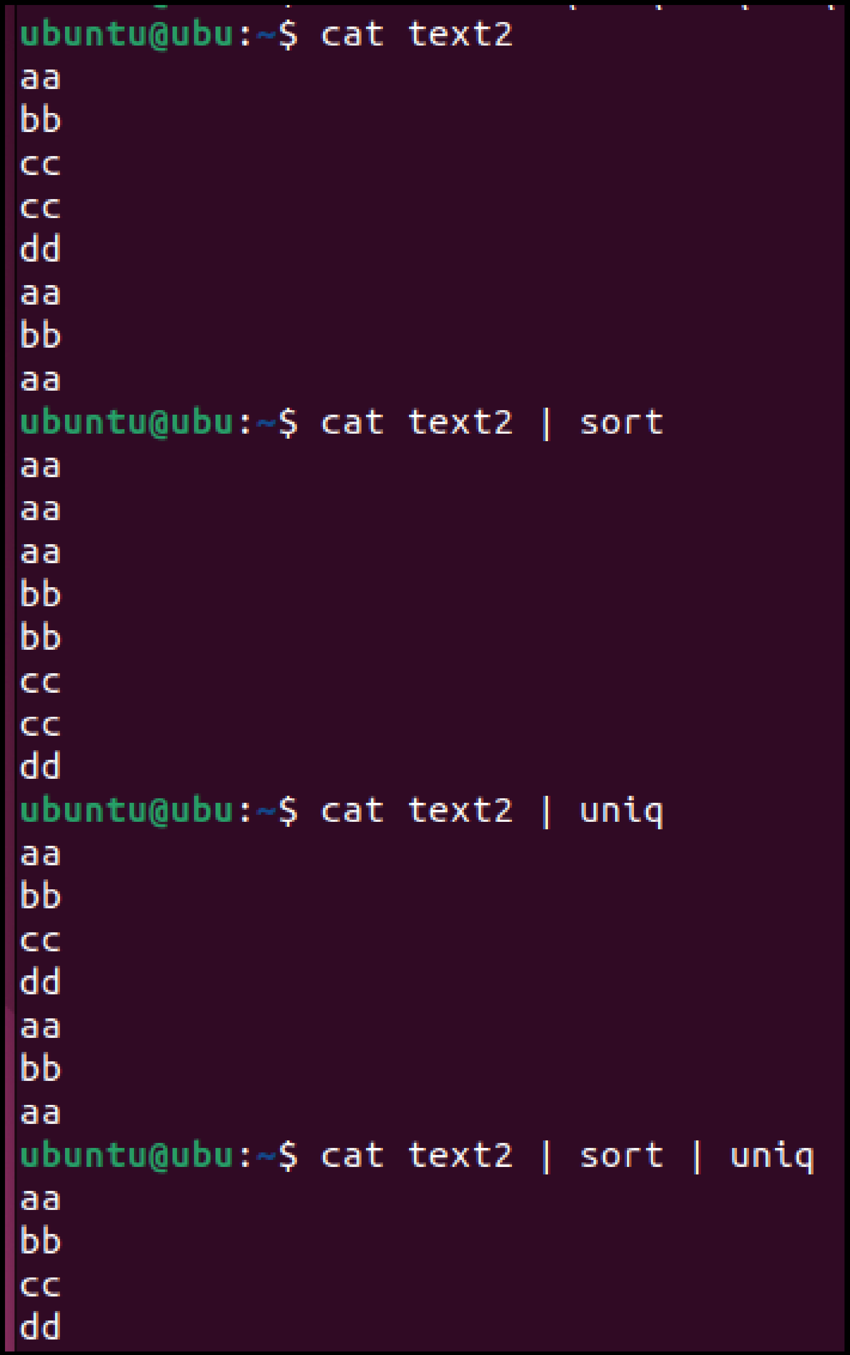


```
ubuntu@ubu:~$ cat text.txt
one two
three one
One four
five six
tree ten
oneten
password 1111
flag 2222
login 3333
ubuntu@ubu:~$ cat text.txt | wc
      9      17      87
ubuntu@ubu:~$ cat /etc/passwd | wc
     50      89    2974
ubuntu@ubu:~$ wc /etc/passwd
  50   89 2974 /etc/passwd
ubuntu@ubu:~$ wc /etc/passwd | cut -d "/" -f1
  50   89 2974
```

Sort & Uniq

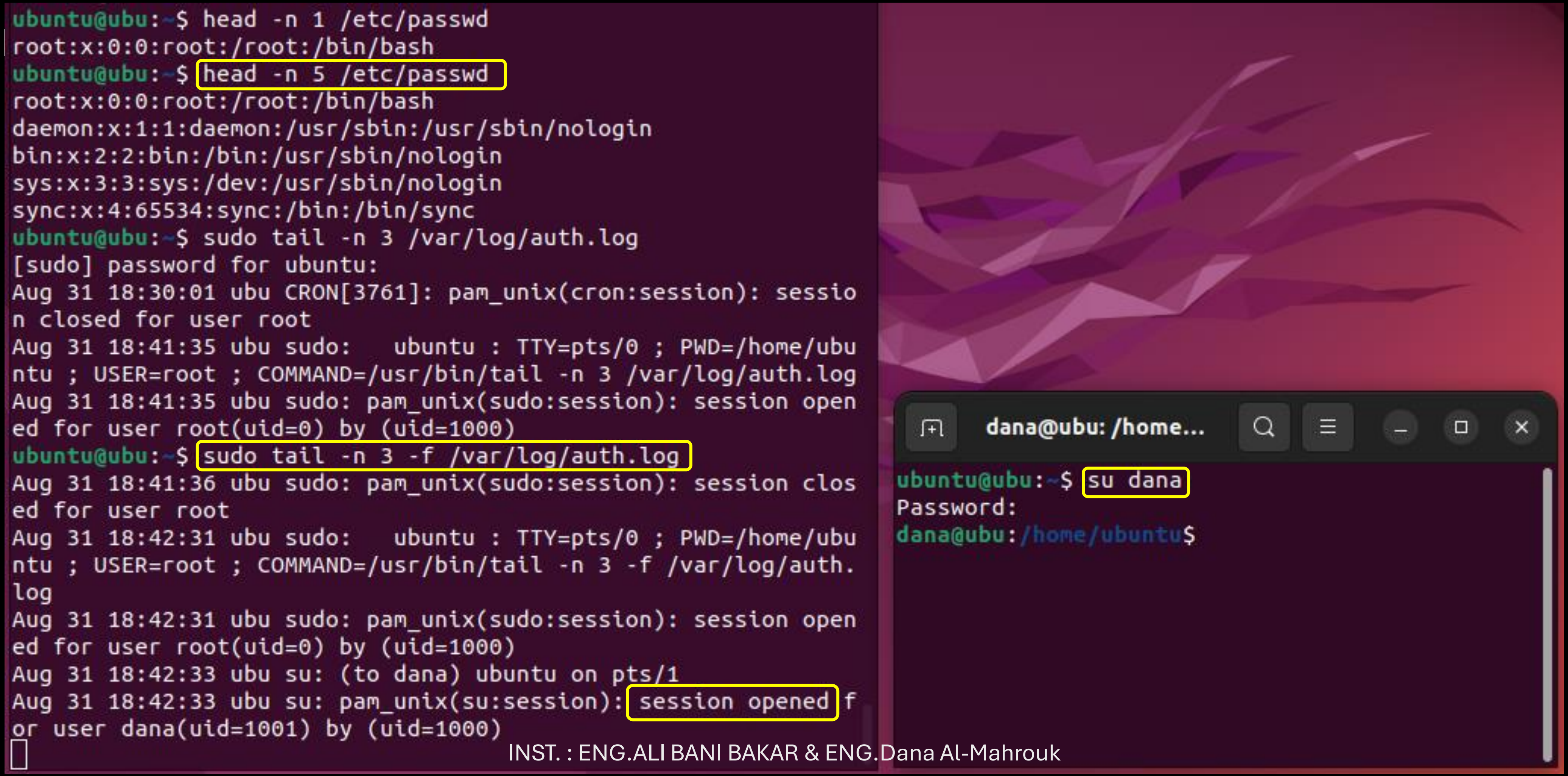
- Sort: arrange lines of text files in a specified order. By default ascending order.

- Uniq: filter out or report unique lines from sorted text files. It is typically used after sort to remove duplicate lines.



```
ubuntu@ubu:~$ cat text2
aa
bb
cc
cc
dd
aa
bb
aa
ubuntu@ubu:~$ cat text2 | sort
aa
aa
aa
bb
bb
cc
cc
dd
ubuntu@ubu:~$ cat text2 | uniq
aa
bb
cc
dd
aa
bb
aa
ubuntu@ubu:~$ cat text2 | sort | uniq
aa
bb
cc
dd
```

/var/log/auth.log

- 

```
ubuntu@ubu:~$ head -n 1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
ubuntu@ubu:~$ head -n 5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
ubuntu@ubu:~$ sudo tail -n 3 /var/log/auth.log
[sudo] password for ubuntu:
Aug 31 18:30:01 ubu CRON[3761]: pam_unix(cron:session): sessio
n closed for user root
Aug 31 18:41:35 ubu sudo:   ubuntu : TTY=pts/0 ; PWD=/home/ubu
ntu ; USER=root ; COMMAND=/usr/bin/tail -n 3 /var/log/auth.log
Aug 31 18:41:35 ubu sudo: pam_unix(sudo:session): session open
ed for user root(uid=0) by (uid=1000)
ubuntu@ubu:~$ sudo tail -n 3 -f /var/log/auth.log
Aug 31 18:41:36 ubu sudo: pam_unix(sudo:session): session clos
ed for user root
Aug 31 18:42:31 ubu sudo:   ubuntu : TTY=pts/0 ; PWD=/home/ubu
ntu ; USER=root ; COMMAND=/usr/bin/tail -n 3 -f /var/log/auth.
log
Aug 31 18:42:31 ubu sudo: pam_unix(sudo:session): session open
ed for user root(uid=0) by (uid=1000)
Aug 31 18:42:33 ubu su: (to dana) ubuntu on pts/1
Aug 31 18:42:33 ubu su: pam_unix(su:session): session opened f
or user dana(uid=1001) by (uid=1000)
```

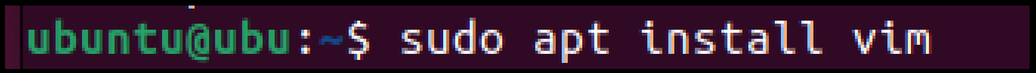
```
ubuntu@ubu:~$ su dana
Password:
dana@ubu:/home/ubuntu$
```

text editor

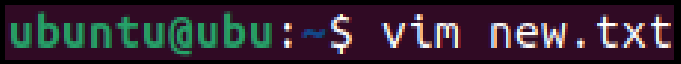
- **Vim**: Preferred by many programmers for its efficiency, customizability, and powerful features, but has a steep learning curve.
- **Nano**: A good choice for beginners or users who need a simple, straightforward text editor without a learning curve.
- **Gedit**: Suitable for those who prefer a graphical interface and a user-friendly environment, with basic functionality and plugin support.

vim

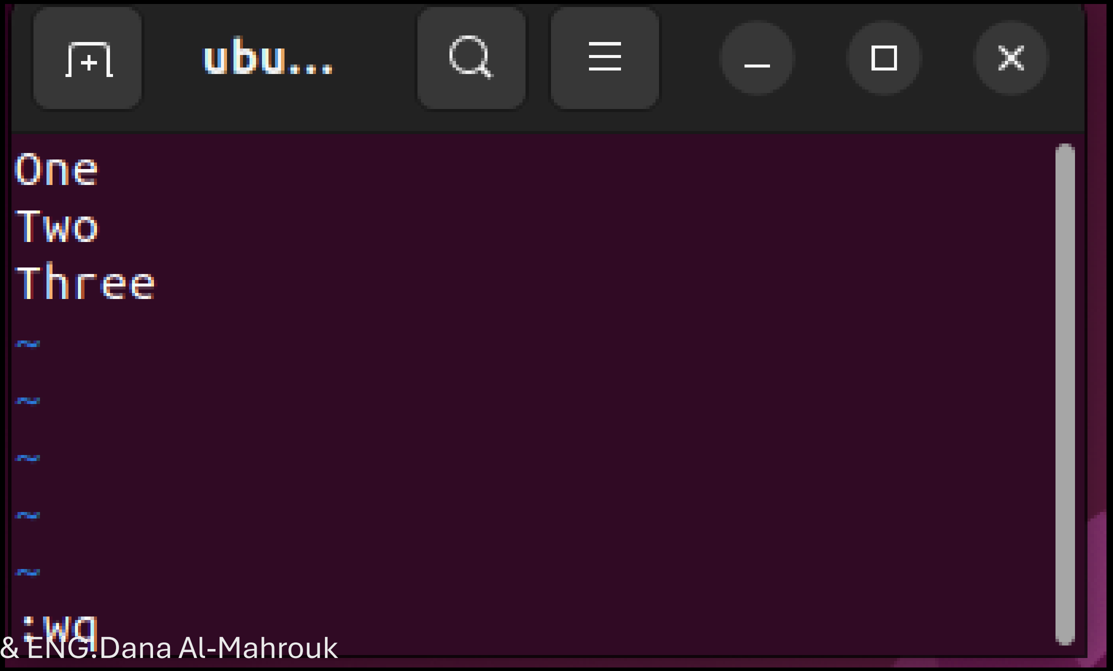
- Mode;
  - Insert mode: press `i`
  - User mode: press `ecs`
    - `:wq` write & quit



```
ubuntu@ubu:~$ sudo apt install vim
```



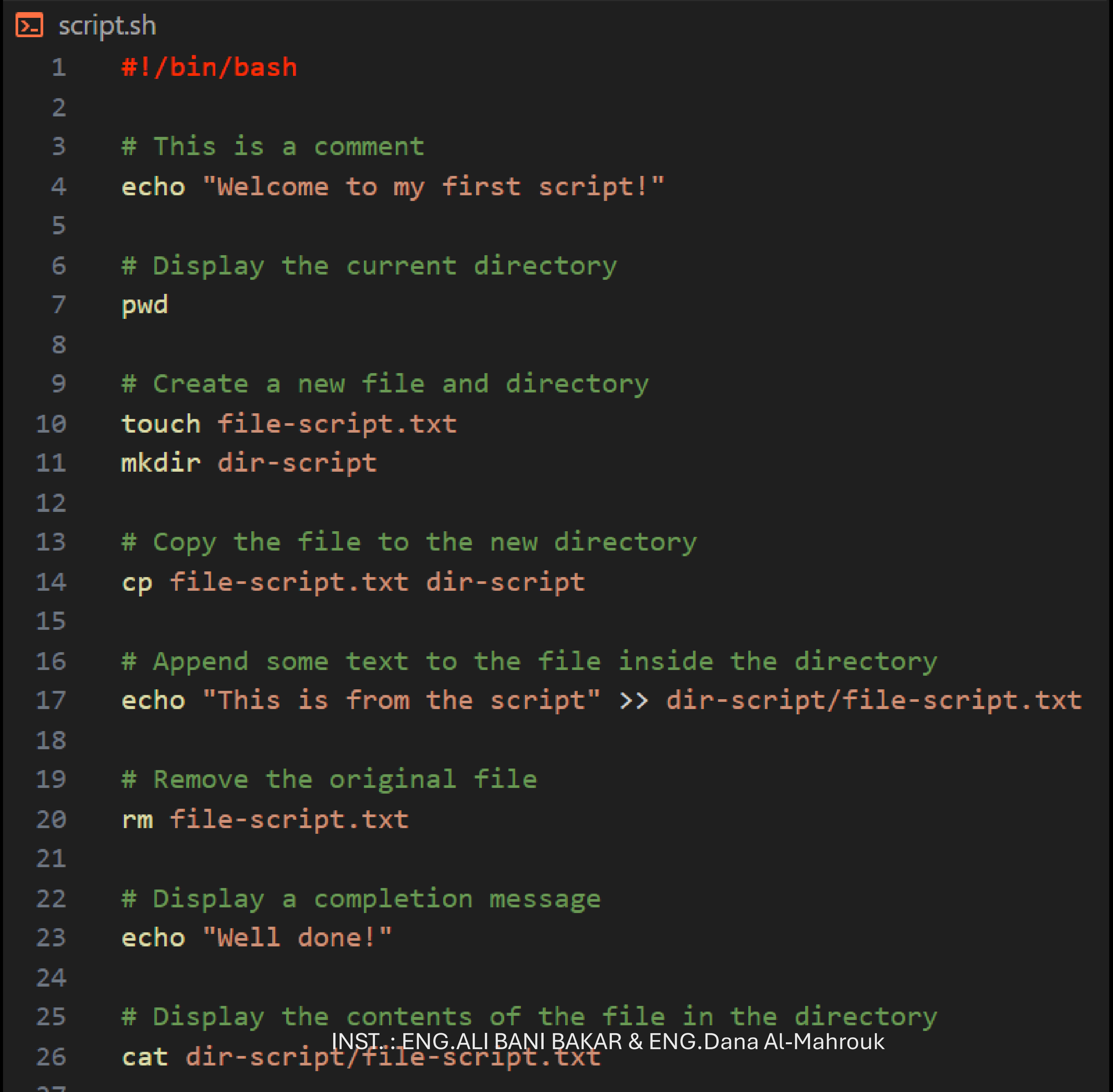
```
ubuntu@ubu:~$ vim new.txt
```



```
One
Two
Three
~
~
~
~
~
:wq
```

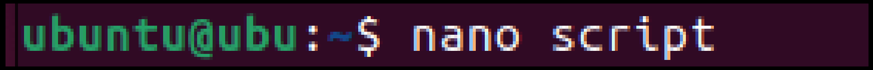
Script

- Creating and running scripts in Linux is a fundamental skill for automating tasks, managing system operations, or even just simplifying repetitive command sequences.
- Nano
    - CTRL + O: save the file
    - CTRL + X: exit

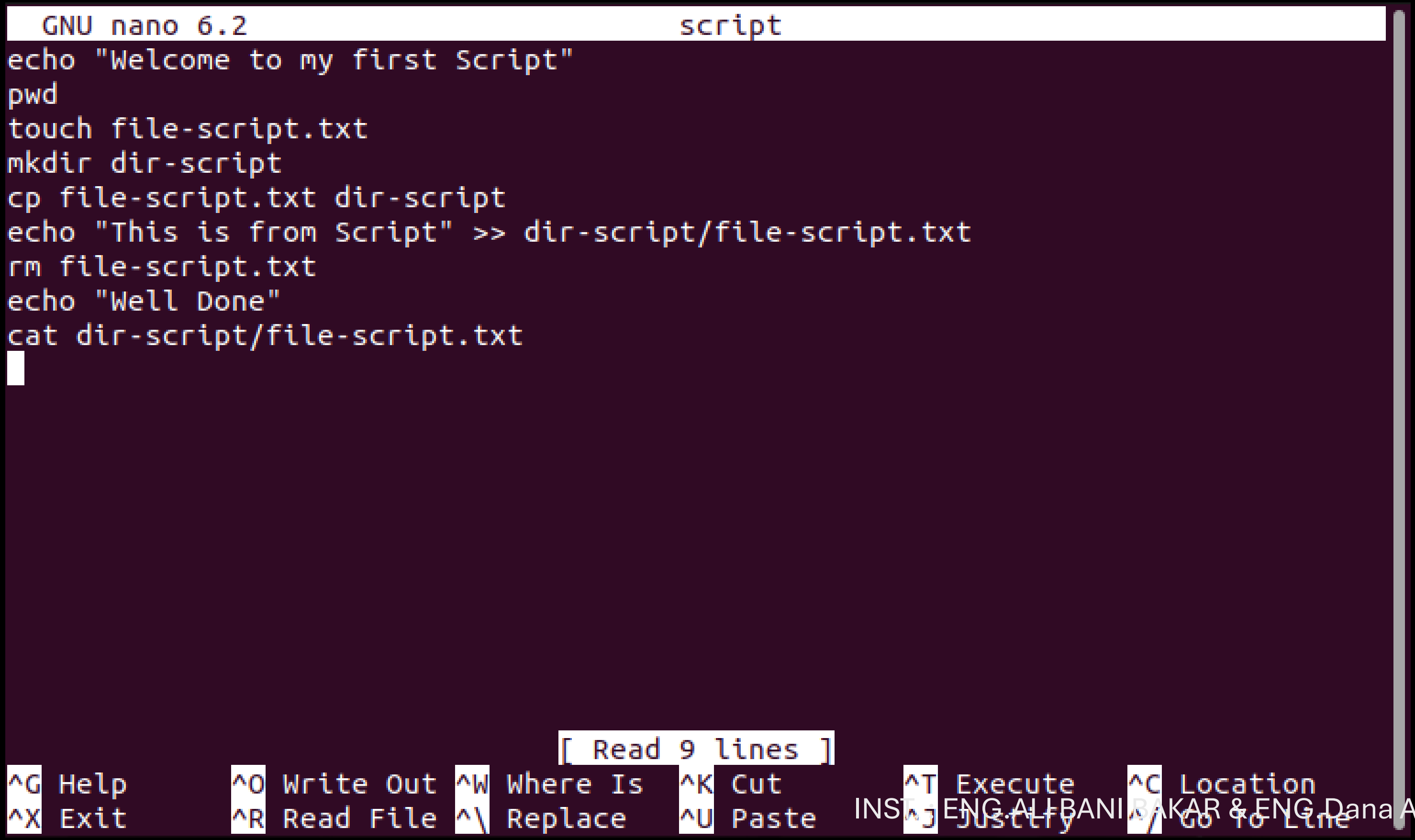


```bash
#!/bin/bash

# This is a comment
echo "Welcome to my first script!"

# Display the current directory
pwd

# Create a new file and directory
touch file-script.txt
mkdir dir-script

# Copy the file to the new directory
cp file-script.txt dir-script

# Append some text to the file inside the directory
echo "This is from the script" >> dir-script/file-script.txt

# Remove the original file
rm file-script.txt

# Display a completion message
echo "Well done!"

# Display the contents of the file in the directory
cat dir-script/file-script.txt
```

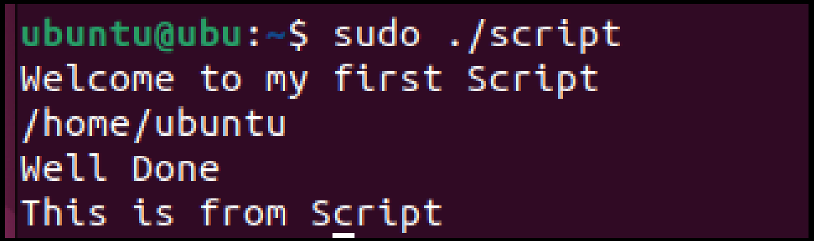
Ex1:



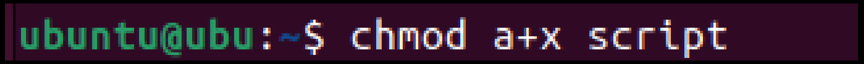
```
ubuntu@ubu:~$ nano script
```



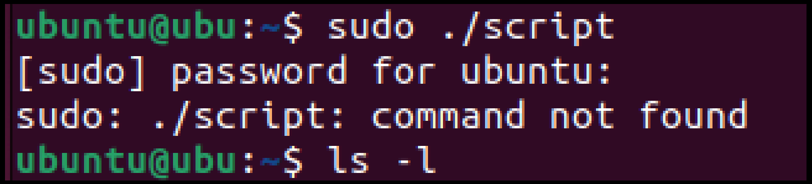
```
  GNU nano 6.2                          script
echo "Welcome to my first Script"
pwd
touch file-script.txt
mkdir dir-script
cp file-script.txt dir-script
echo "This is from Script" >> dir-script/file-script.txt
rm file-script.txt
echo "Well Done"
cat dir-script/file-script.txt

                                [ Read 9 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



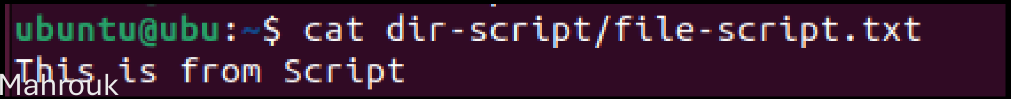
```
ubuntu@ubu:~$ sudo ./script
Welcome to my first Script
/home/ubuntu
Well Done
This is from Script
```



```
ubuntu@ubu:~$ chmod a+x script
```

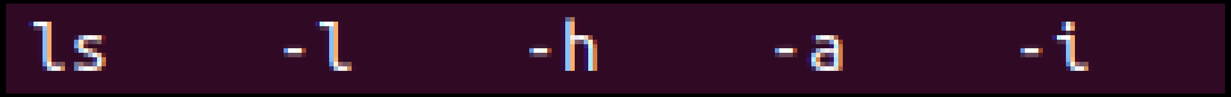


```
ubuntu@ubu:~$ sudo ./script
[sudo] password for ubuntu:
sudo: ./script: command not found
ubuntu@ubu:~$ ls -l
```



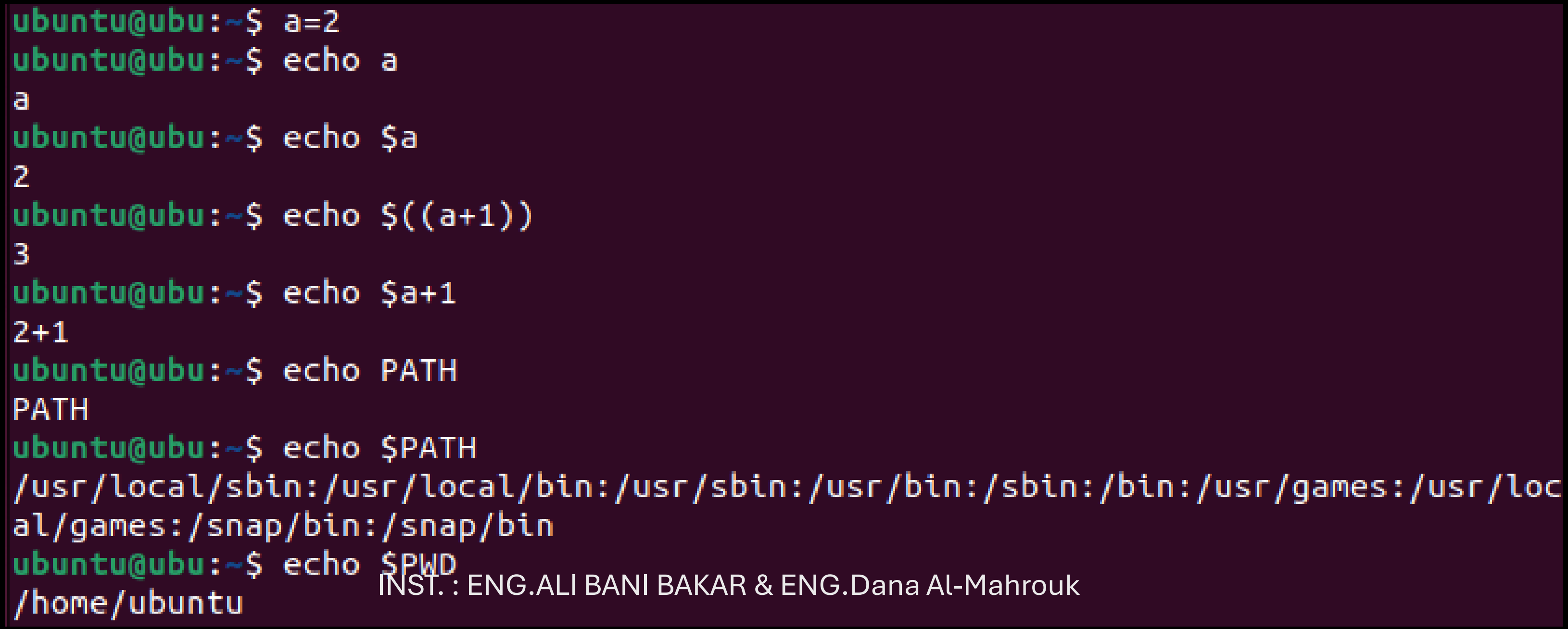
```
ubuntu@ubu:~$ cat dir-script/file-script.txt
This is from Script
```

User input



```
ls   -l   -h   -a   -i
```

Program Name	Option	Option	Option	Option
Argument[0]	Arg [1]	Arg [2]	Arg [3]	Arg[4]

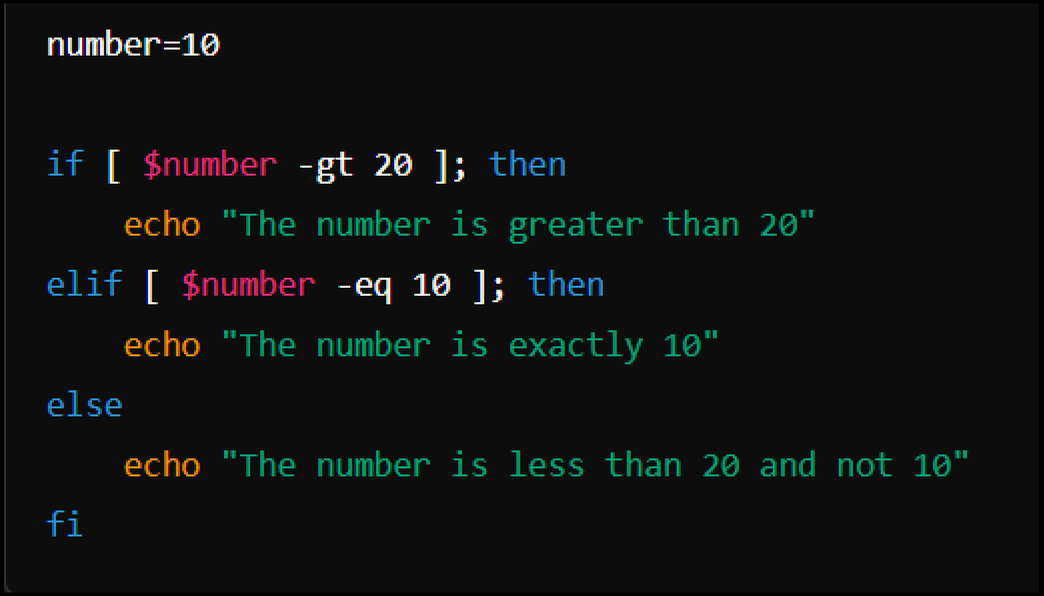


```
ubuntu@ubu:~$ a=2
ubuntu@ubu:~$ echo a
a
ubuntu@ubu:~$ echo $a
2
ubuntu@ubu:~$ echo $((a+1))
3
ubuntu@ubu:~$ echo $a+1
2+1
ubuntu@ubu:~$ echo PATH
PATH
ubuntu@ubu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/loc
al/games:/snap/bin:/snap/bin
ubuntu@ubu:~$ echo $PWD
/home/ubuntu
```

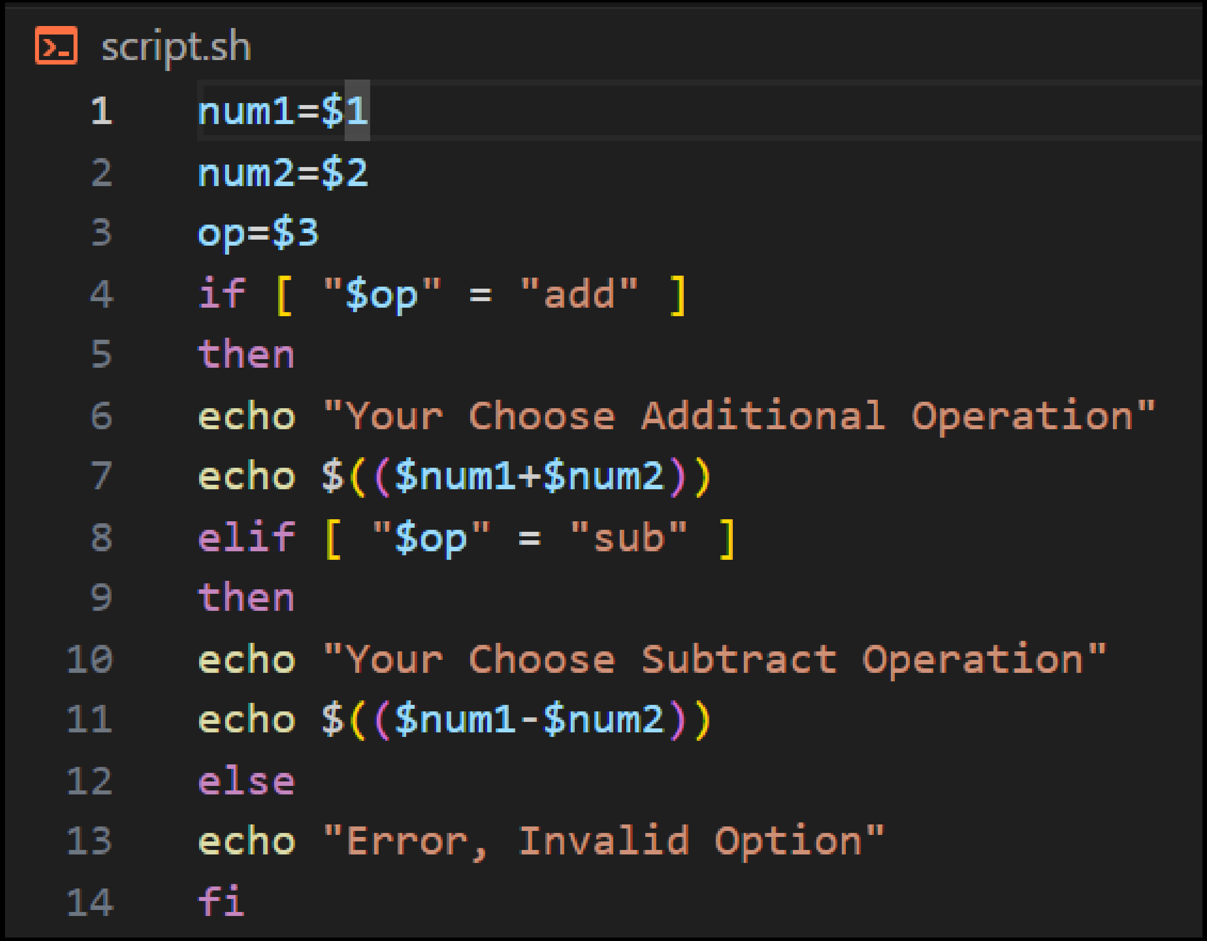
Day 11

- Outline
    - Script
        - if statement
            - if then
            - elif then
            - else
            - fi

if, elif, else, then, fi



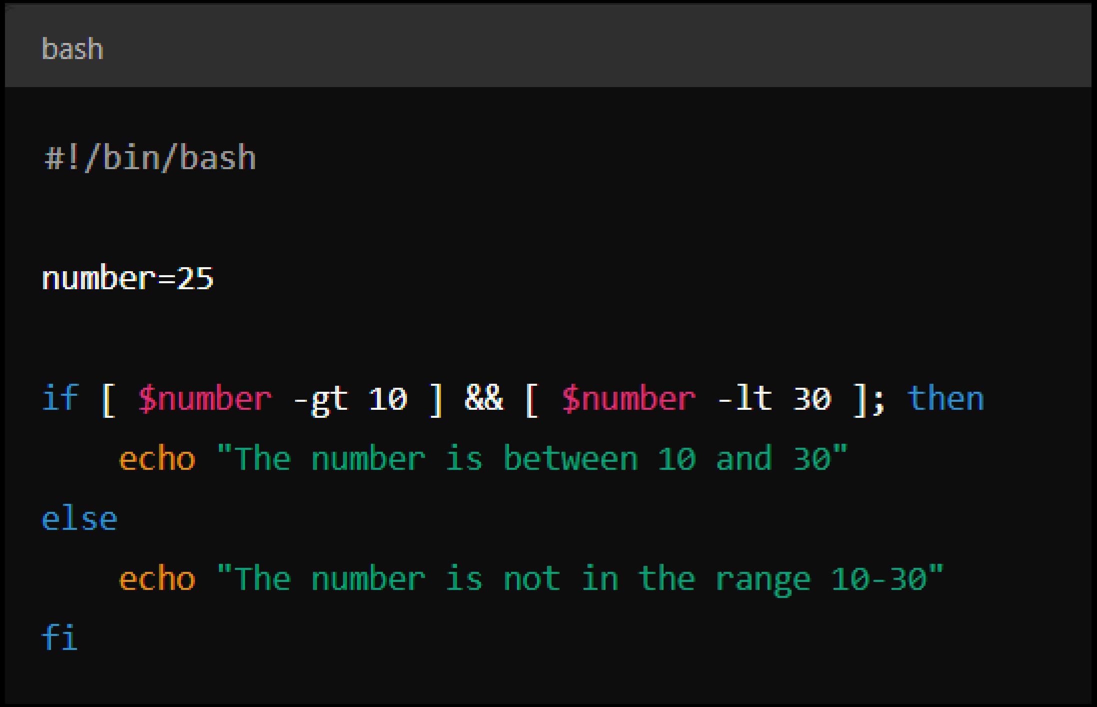
```
number=10

if [ $number -gt 20 ]; then
    echo "The number is greater than 20"
elif [ $number -eq 10 ]; then
    echo "The number is exactly 10"
else
    echo "The number is less than 20 and not 10"
fi
```



script.sh

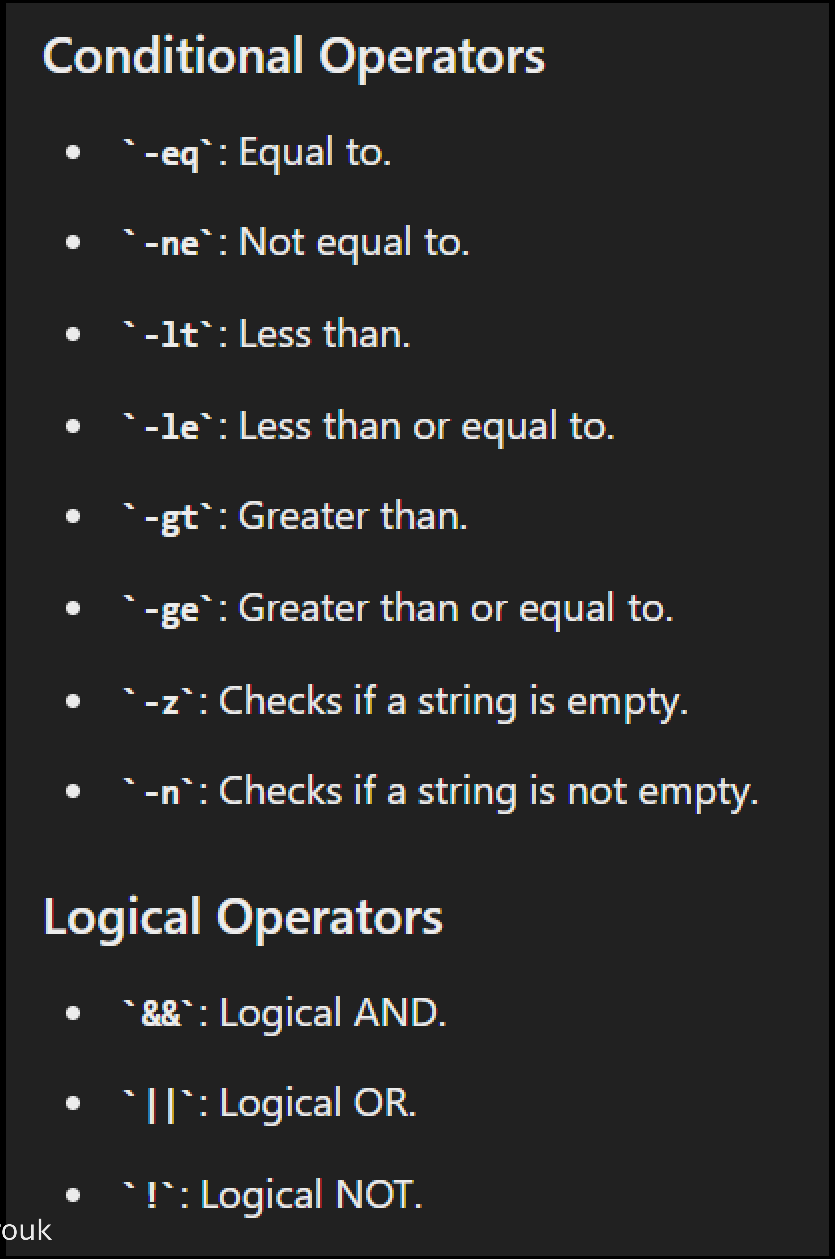
```
num1=$1
num2=$2
op=$3
if [ "$op" = "add" ]
then
echo "Your Choose Additional Operation"
echo $(($num1+$num2))
elif [ "$op" = "sub" ]
then
echo "Your Choose Subtract Operation"
echo $(($num1-$num2))
else
echo "Error, Invalid Option"
fi
```

Operations



bash

```bash
#!/bin/bash

number=25

if [ $number -gt 10 ] && [ $number -lt 30 ]; then
    echo "The number is between 10 and 30"
else
    echo "The number is not in the range 10-30"
fi
```



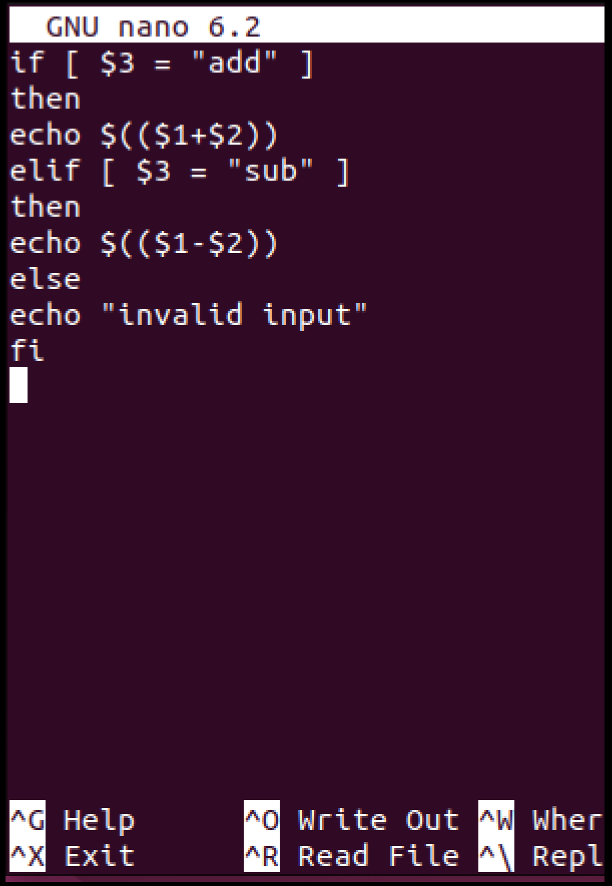
Conditional Operators

- `-eq`: Equal to.
- `-ne`: Not equal to.
- `-lt`: Less than.
- `-le`: Less than or equal to.
- `-gt`: Greater than.
- `-ge`: Greater than or equal to.
- `-z`: Checks if a string is empty.
- `-n`: Checks if a string is not empty.

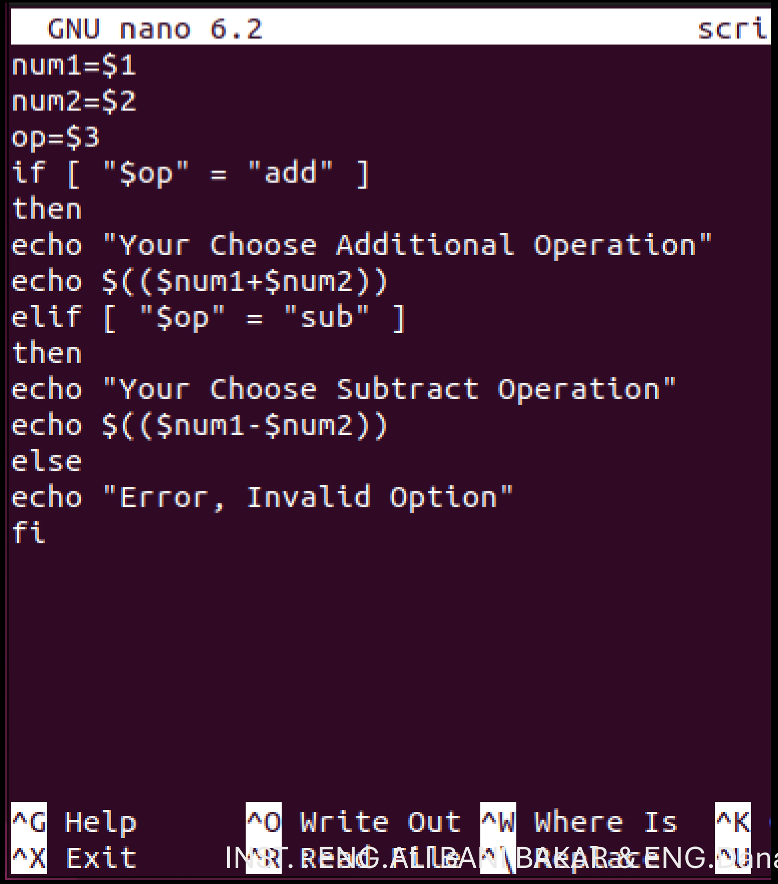
Logical Operators

- `&&`: Logical AND.
- `||`: Logical OR.
- `!`: Logical NOT.

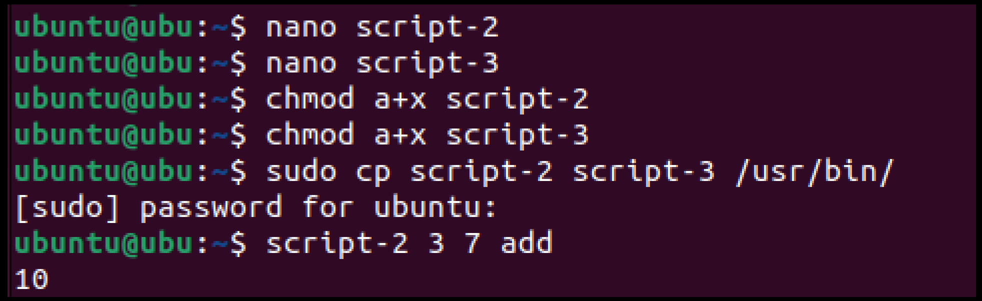
Ex2: Script-2 & Script-3



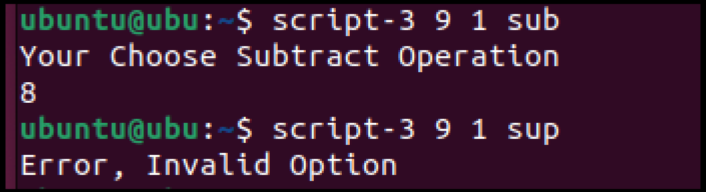
```
if [ $3 = "add" ]
then
echo $(($1+$2))
elif [ $3 = "sub" ]
then
echo $(($1-$2))
else
echo "invalid input"
fi
```



```
num1=$1
num2=$2
op=$3
if [ "$op" = "add" ]
then
echo "Your Choose Additional Operation"
echo $(($num1+$num2))
elif [ "$op" = "sub" ]
then
echo "Your Choose Subtract Operation"
echo $(($num1-$num2))
else
echo "Error, Invalid Option"
fi
```



```
ubuntu@ubu:~$ nano script-2
ubuntu@ubu:~$ nano script-3
ubuntu@ubu:~$ chmod a+x script-2
ubuntu@ubu:~$ chmod a+x script-3
ubuntu@ubu:~$ sudo cp script-2 script-3 /usr/bin/
[sudo] password for ubuntu:
ubuntu@ubu:~$ script-2 3 7 add
10
```



```
ubuntu@ubu:~$ script-3 9 1 sub
Your Choose Subtract Operation
8
ubuntu@ubu:~$ script-3 9 1 sup
Error, Invalid Option
```