

UNIVERSITETI POLITEKNIK I TIRANËS  
FAKULTETI I TEKNOLOGJISË DHE INFORMACIONIT  
DEPARTAMENTI I INXHINIERISË INFORMATIKE

## **Punë Laboratori nr. 1**

**Lënda:** Sisteme Operative

**Grupi:** III-B

**Tema:** Instalimi i sistemit operativ *Linux Ubuntu* (në VirtualBox).  
Ambjentimi me komandat kryesore të *Linux-it* në terminal.

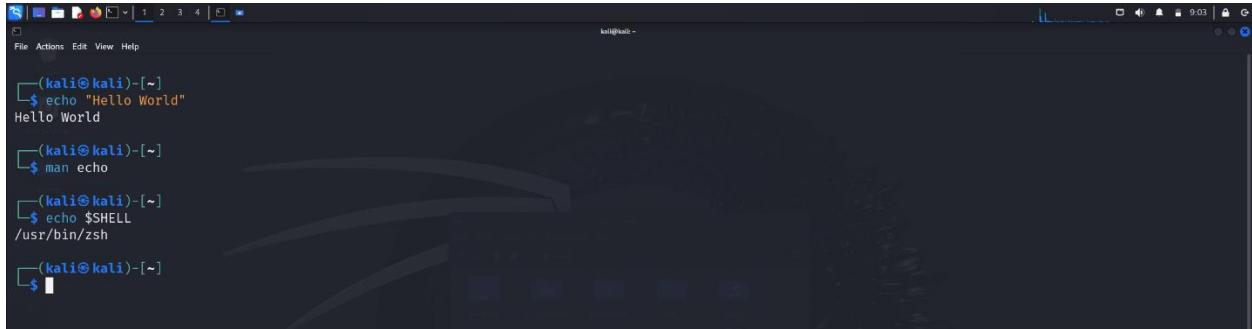
**Punoi:**  
Piro Gjikdhima

**Pranoi:**  
MSc.Megi Tartari

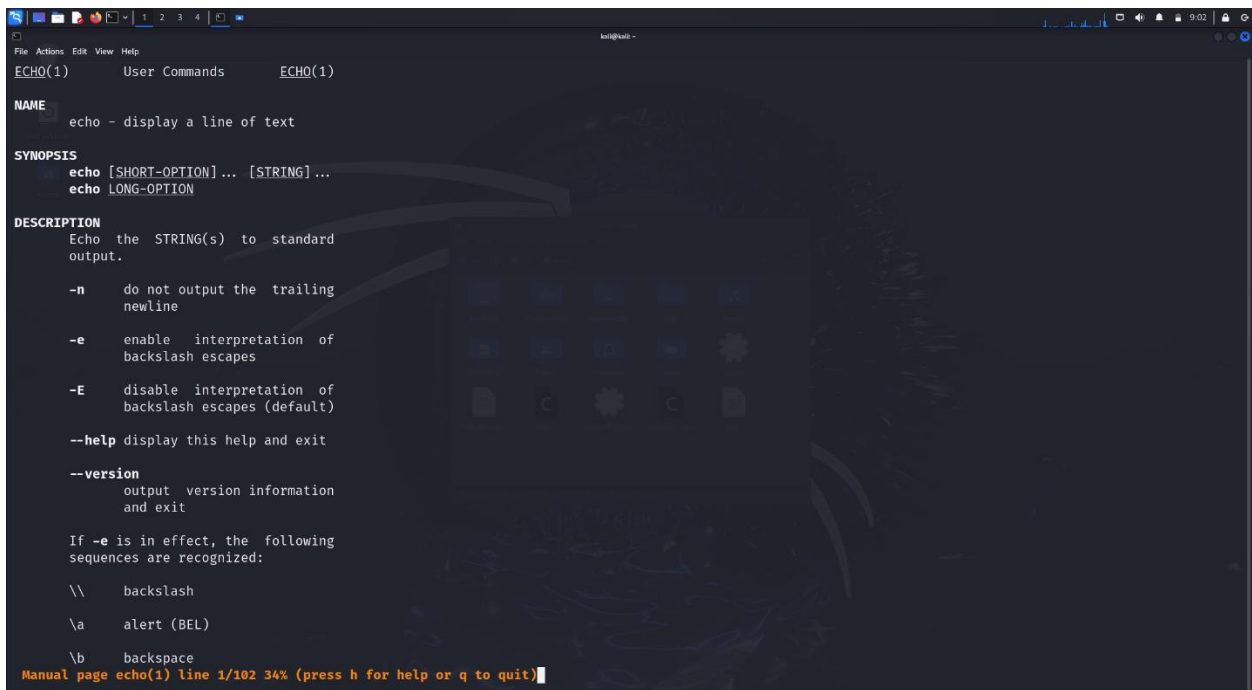
## Teori:

**Pika 1:** Hap terminal në *Linux* dhe shtyp komandat më poshtë:

- ***\$ echo Hello World***
- ***\$ man echo*** (*man* online manual për komandat)
- ***\$ echo \$SHELL*** (printon shell-in që po përdor)



```
(kali@kali)-[~]  
$ echo "Hello World"  
Hello World  
  
(kali@kali)-[~]  
$ man echo  
  
(kali@kali)-[~]  
$ echo $SHELL  
/usr/bin/zsh  
  
(kali@kali)-[~]  
$
```



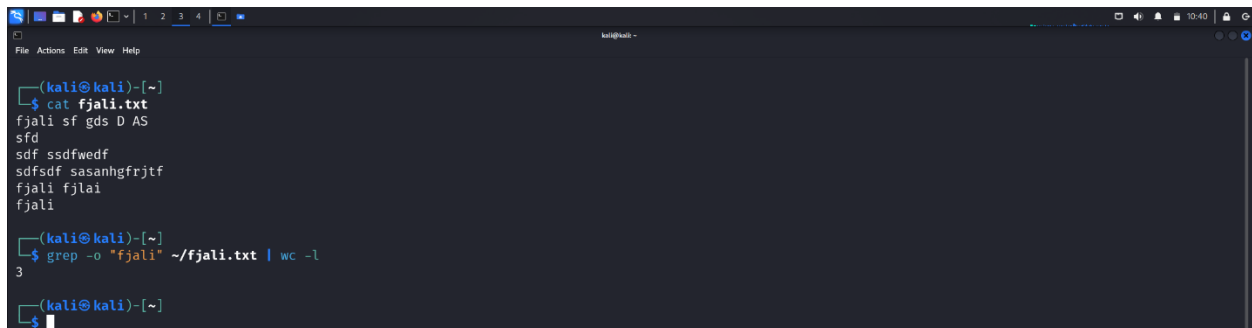
```
ECHO(1)      User Commands      ECHO(1)  
  
NAME  
    echo - display a line of text  
  
SYNOPSIS  
    echo [SHORT-OPTION]... [STRING]...  
    echo LONG-OPTION  
  
DESCRIPTION  
    Echo the STRING(s) to standard  
    output.  
  
    -n    do not output the trailing  
          newline  
  
    -e    enable interpretation of  
          backslash escapes  
  
    -E    disable interpretation of  
          backslash escapes (default)  
  
    --help display this help and exit  
  
    --version  
           output version information  
           and exit  
  
    If -e is in effect, the following  
    sequences are recognized:  
  
    \\    backslash  
  
    \a    alert (BEL)  
  
    \b    backspace  
  
Manual page echo(1) line 1/102 34% (press h for help or q to quit)
```

Studio komandat në vijim referuar dokumentit *unix\_tutorial.pdf* si dhe anë të komandës *man*: *man*, *who*, *cat*, *cd*, *cp*, *ps*, *ls*, *mv*, *rm*, *mkdir*, *rmdir*, *echo*, *more*, *date*, *time*, *kill*, *history*, *chmod*, *chown*, *finger*, *pwd*, *cal*, *logout*, *shutdown*.

- ***man***- an interface of the system reference manuals
- ***who***- show who is logged on
- ***cat***- concatenate files and print on the standard output
- ***cd***- offers several ways to navigate and change the working directory using the terminal window

- **cp**- copy files and directories
- **ps**- raport a snapshop of the current process
- **ls**- list directory content
- **mv**- move (rename) files
- **rm**- remove files or directories
- **mkdir**- make directories
- **rmdir**- remove directories
- **echo**- display a line of text
- **more**- file perusal filter for crt viewing
- **date**- print or set the system date and time
- **time**- run programs and summarize system resource usage
- **kill**- send a signal to a process
- **history**- GNU history library
- **chmod**- change file mode bits
- **chown**- change file owner and group
- **finger**- gives details of all the users who are logged in
- **pwd**- print name of current/working directory
- **cal**- displays a calendar and the date of Easter
- **logout**- write utmp and wtmp entries
- **shutdown**- Halt, poweroff or reboot the machine
- **grep**- print line that matches patterns

Studio komandën **grep** dhe përdore për të kërkuar sa rreshta të një file mbajnë një fjalë kyçe. Lexoni ose vetem shfaqni permbajtjen e file:cat



```

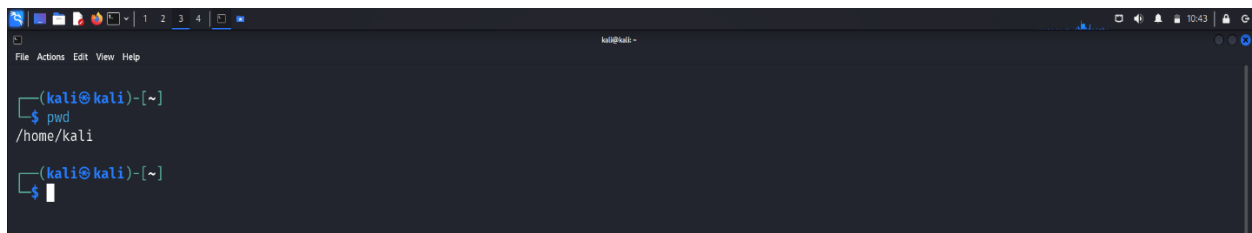
(kali@kali)-[~]
$ cat fjali.txt
fjali sf gds D AS
sfd
sdf ssdfwedf
sdfsdf sasanhgfrjtf
fjali fjlai
fjali

(kali@kali)-[~]
$ grep -o "fjali" ~/fjali.txt | wc -l
3

(kali@kali)-[~]
$

```

Verifikoni ne cilen direkto jeni aktualisht:pwd



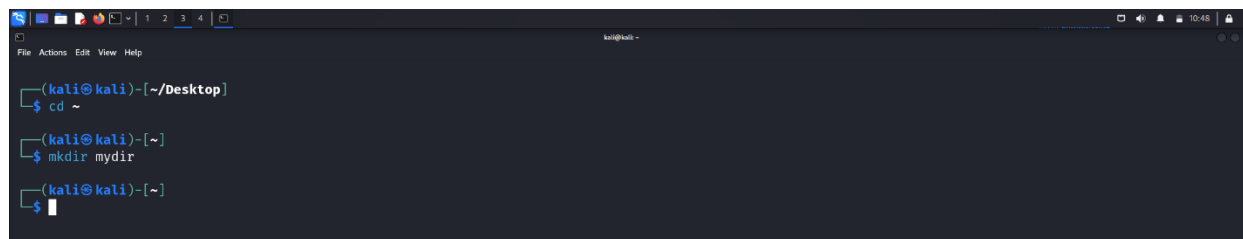
```

(kali@kali)-[~]
$ pwd
/home/kali

(kali@kali)-[~]
$

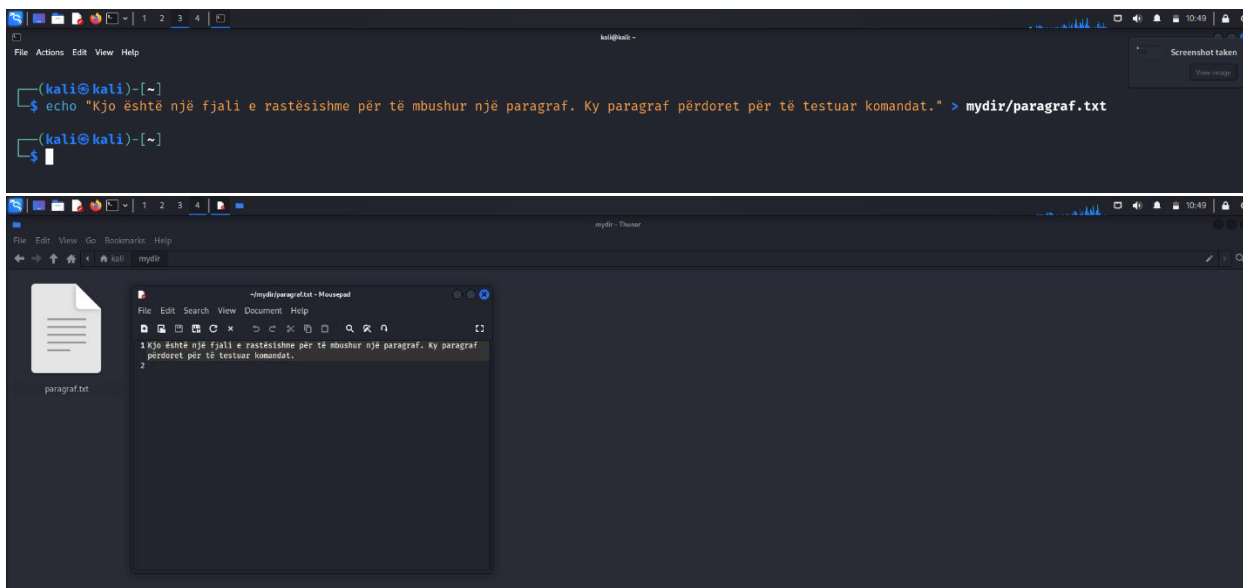
```

Shkoni tek home directory: `cd ~` dhe krijoni nje nendirektori.

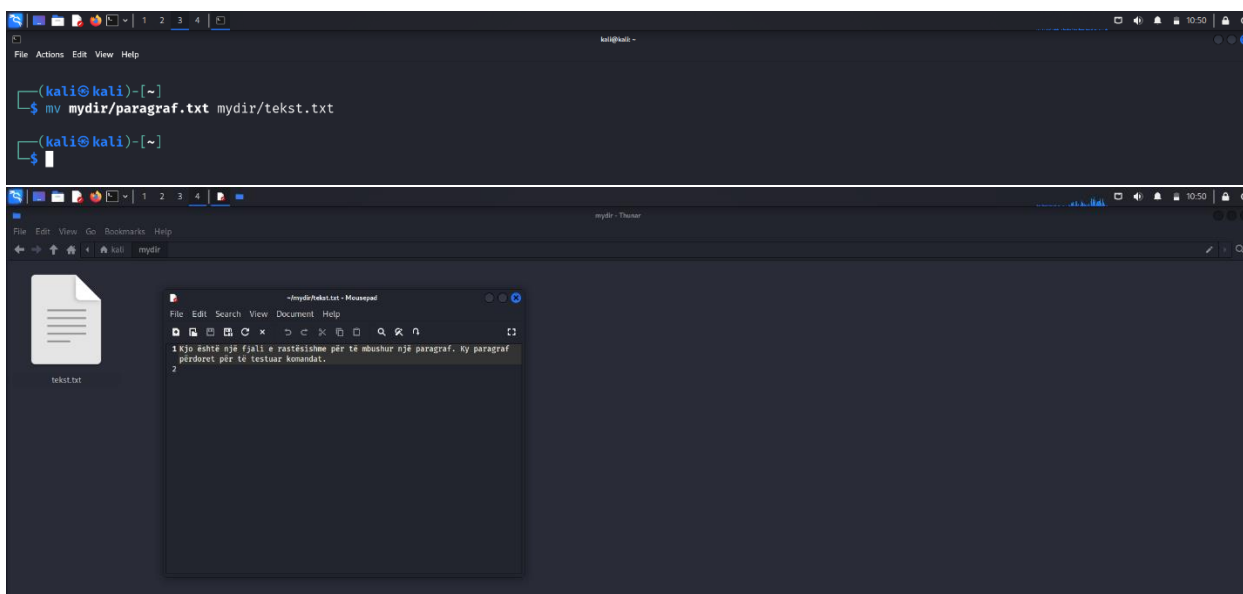
A terminal window titled 'kali@kali' showing the following commands and output:

```
(kali@kali)-[~/Desktop]
$ cd ~
(kali@kali)-[~]
$ mkdir mydir
(kali@kali)-[~]
$
```

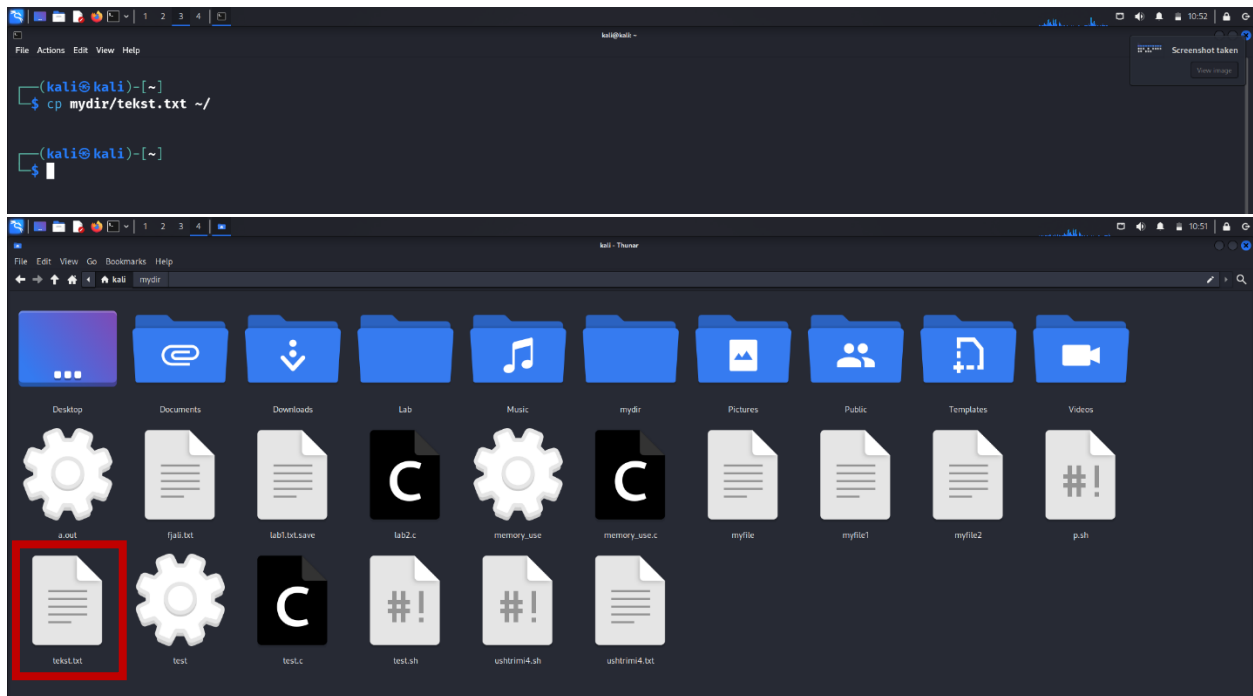
Krijoni nje text file me nje paragraph cfaredo.



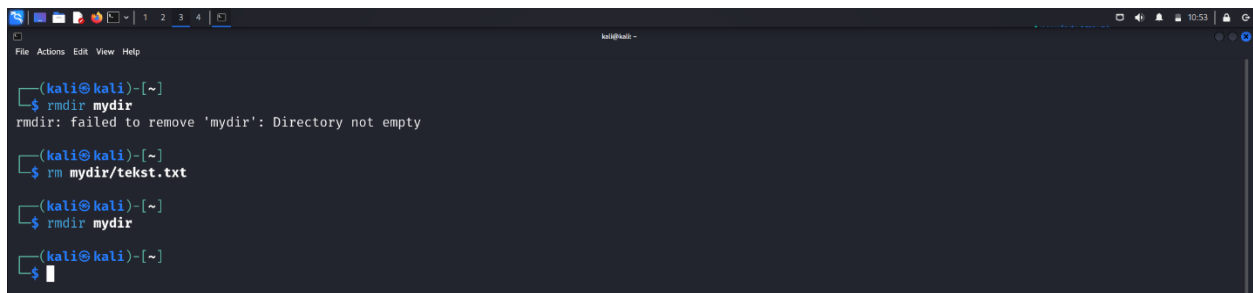
Beni rename te tij



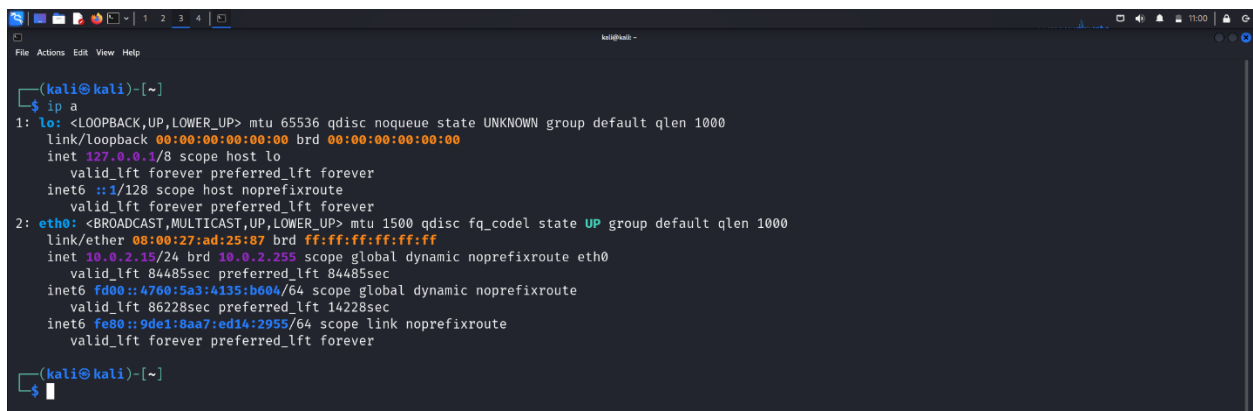
Kopjojeni kete file dhe kalojeni ne nje direktori me siper:cp



Fshijeni nendirektore:rmmdir



Cekoni konfigurimin e kartave te rrjetit: ip a ose tek /etc/netplan



Bejeni file readonly.

```
kali@kali ~  
$ chmod 444 tekst.txt  
$
```

Cilet user jane te loguar aktualisht:who

```
kali@kali ~  
$ who  
kali    tty7      2024-11-24 10:28 (:0)  
$
```

Cilat jane proceset qe po ekzekutohen dhe sa ram cpu po harxhojne?

```
kali@kali ~  
top - 11:05:26 up 37 min, 2 users, load average: 0.17, 0.14, 0.15  
Tasks: 177 total, 1 running, 176 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 5.3 us, 3.6 sy, 0.0 ni, 90.9 id, 0.2 wa, 0.0 hi, 0.0 si, 0.0 st  
MiB Mem : 3922.8 total, 2502.6 free, 1149.3 used, 548.6 buff/cache  
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used, 2773.5 avail Mem  


| PID   | USER | PR  | NI  | VIRT   | RES    | SHR    | S | %CPU | %MEM | TIME+   | COMMAND                     |
|-------|------|-----|-----|--------|--------|--------|---|------|------|---------|-----------------------------|
| 671   | root | 20  | 0   | 696096 | 258224 | 117564 | S | 22.3 | 6.4  | 1:56.45 | Xorg                        |
| 10091 | kali | 20  | 0   | 462048 | 105372 | 91700  | S | 1.3  | 2.6  | 0:01.65 | qterminal                   |
| 852   | kali | 20  | 0   | 337728 | 25564  | 17280  | S | 0.3  | 0.6  | 0:00.58 | xfce4-session               |
| 921   | kali | 20  | 0   | 215444 | 3212   | 2816   | S | 0.3  | 0.1  | 0:01.10 | VBoxClient                  |
| 930   | kali | 20  | 0   | 215960 | 3212   | 2816   | S | 0.3  | 0.1  | 0:06.42 | VBoxClient                  |
| 969   | kali | 20  | 0   | 233844 | 7424   | 6656   | S | 0.3  | 0.2  | 0:00.31 | at-spi2-registr             |
| 984   | kali | 20  | 0   | 969084 | 119432 | 79152  | S | 0.3  | 3.0  | 0:10.07 | xfwm4                       |
| 1054  | kali | 20  | 0   | 285908 | 45752  | 19328  | S | 0.3  | 1.1  | 0:04.75 | panel-13-cpugra             |
| 2351  | root | 20  | 0   | 537904 | 47992  | 34496  | S | 0.3  | 1.2  | 0:03.58 | thunar                      |
| 1     | root | 20  | 0   | 22124  | 13944  | 10196  | S | 0.0  | 0.3  | 0:01.64 | systemd                     |
| 2     | root | 20  | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:00.04 | kthreadd                    |
| 3     | root | 20  | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:00.00 | pool_workqueue_release      |
| 4     | root | 0   | -20 | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.00 | kworker/R-rcu_g             |
| 5     | root | 0   | -20 | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.00 | kworker/R-rcu_p             |
| 6     | root | 0   | -20 | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.00 | kworker/R-slub_             |
| 7     | root | 0   | -20 | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.00 | kworker/R-netns             |
| 12    | root | 0   | -20 | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.00 | kworker/R-mm_pe             |
| 13    | root | 20  | 0   | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.00 | rcu_tasks_kthread           |
| 14    | root | 20  | 0   | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.00 | rcu_tasks_rude_kthread      |
| 15    | root | 20  | 0   | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.00 | rcu_tasks_trace_kthread     |
| 16    | root | 20  | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:00.31 | ksoftirqd/0                 |
| 17    | root | 20  | 0   | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.57 | rcu_preempt                 |
| 18    | root | rt  | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:00.02 | migration/0                 |
| 19    | root | -51 | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:00.00 | idle_inject/0               |
| 20    | root | 20  | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:00.00 | cpuhp/0                     |
| 21    | root | 20  | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:00.00 | cpuhp/1                     |
| 22    | root | -51 | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:00.00 | idle_inject/1               |
| 23    | root | rt  | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:01.11 | migration/1                 |
| 24    | root | 20  | 0   | 0      | 0      | 0      | S | 0.0  | 0.0  | 0:00.19 | ksoftirqd/1                 |
| 26    | root | 0   | -20 | 0      | 0      | 0      | I | 0.0  | 0.0  | 0:00.00 | kworker/1:0H-events_highpri |


```

Pika 2: Shkruaj një shell script që printon Hello World në ekran.

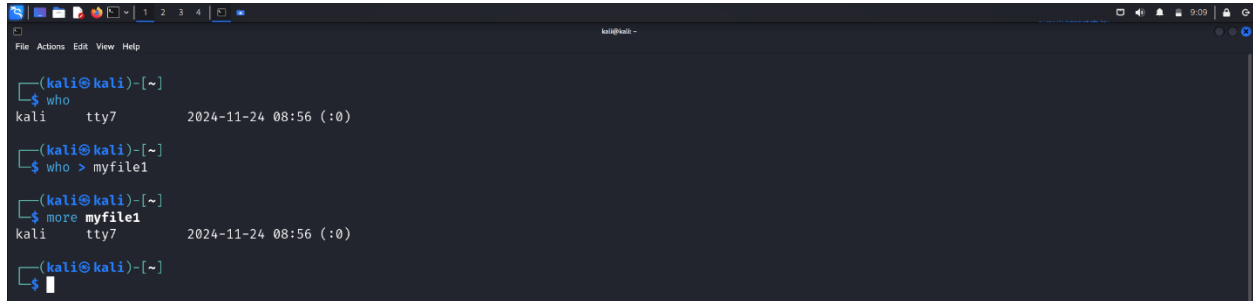
```
GNU nano 8.1 test.sh  
echo "Hello World"  
$ sh test.sh  
Hello World  
$
```

## Ushtrimi 1:

Përdor komandën **who** dhe ridrejto outputin në file-in *myfile1* dhe më pas përdor komandën **more** për të parë përmbajtjen e file-it.

Shkruajmë ne terminal komandën e dhënë më poshtë për të ridrejtuar outputin në *myfile1*:

Tani përdorim komandën **more** dhe shohim outputin e file-it që është user-i që po përdorim dhe të dhënat e datës dhe orës që është përdorur komanda **who**.



```
(kali@kali)-[~]
$ who
kali    tty7      2024-11-24 08:56 (:0)

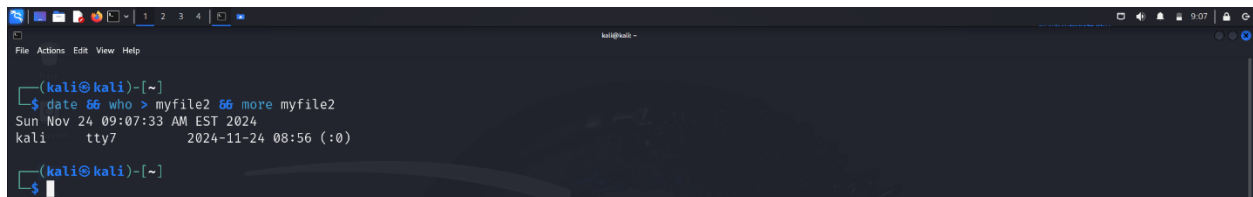
(kali@kali)-[~]
$ who > myfile1

(kali@kali)-[~]
$ more myfile1
kali    tty7      2024-11-24 08:56 (:0)

(kali@kali)-[~]
$
```

## Ushtrimi 2:

Përdor komandën **date** dhe **who** në sekuençë në një rresht në mënyrë që output-i i **date** të shfaqet në ekran kurse ai i **who** të ridrejtohet tek file-i *myfile2*. Përdor komandën **more** për të parë përmbajtjen e file-it *myfile2*.



```
(kali@kali)-[~]
$ date && who > myfile2 && more myfile2
Sun Nov 24 09:07:33 AM EST 2024
kali    tty7      2024-11-24 08:56 (:0)

(kali@kali)-[~]
$
```

Në këtë rast ne përdorim **AND (&&)** që nënkupton detyrimin e të tre komandave të kryhen.

### Ushtrimi 3:

Shkruaj një program në C që printon “Hello World” dhe krahaso kohët e ekzekutimit me shell scriptin e krijuar me komandën *time*.

*Programi në C i krijuar:*

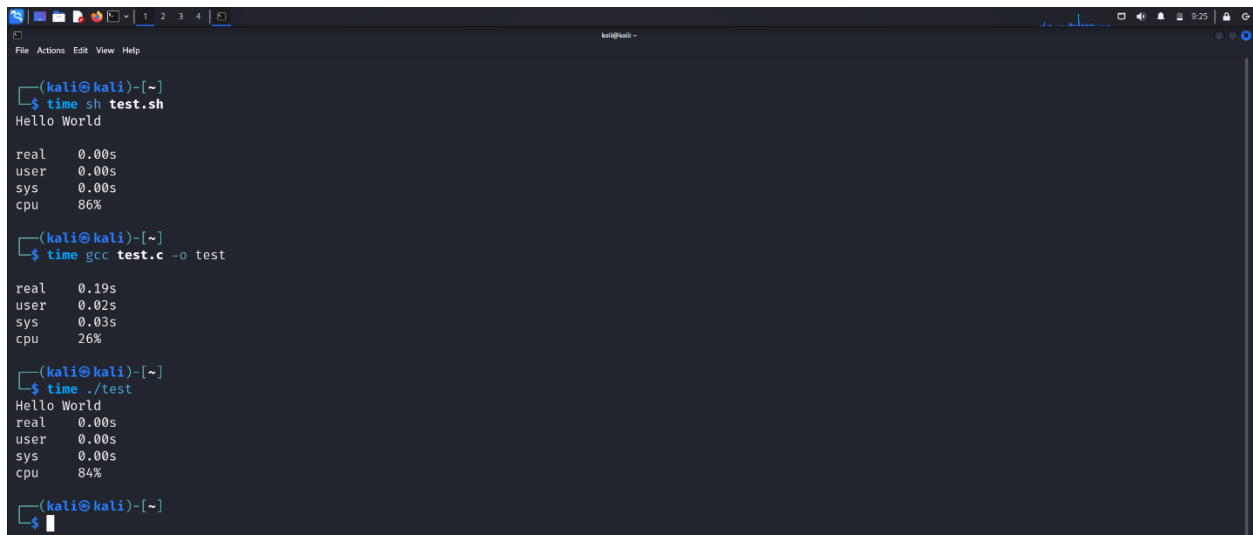
A screenshot of a terminal window with the nano text editor open. The editor is editing a file named 'test.c'. The code visible is: #include <stdio.h> int main(){ printf("Hello World"); return 0; }

```
GNU nano 8.1 test.c

#include <stdio.h>

int main(){
    printf("Hello World");
    return 0;
}
```

Për të ekzekutuar shell scriptin përdorim komandën *\$sh*. Përpara te dy komandave do vendosim komandën *time* të masim kohën.

A screenshot of a terminal window showing the execution of a shell script and a C program. The user runs 'time sh test.sh' which outputs 'Hello World' and timing statistics. Then the user runs 'time gcc test.c -o test' which outputs compilation timing statistics. Finally, the user runs 'time ./test' which outputs 'Hello World' and timing statistics.

```
(kali@kali)-[~]
$ time sh test.sh
Hello World

real    0.00s
user    0.00s
sys     0.00s
cpu     86%

(kali@kali)-[~]
$ time gcc test.c -o test

real    0.19s
user    0.02s
sys     0.03s
cpu     26%

(kali@kali)-[~]
$ time ./test
Hello World

real    0.00s
user    0.00s
sys     0.00s
cpu     84%

(kali@kali)-[~]
$
```

Siç shohim dhe më sipër në lidhje me ekzekutimin e programeve, shell scripti do më shumë kohë për t’u ekzekutuar sesa programi në C. Por shell scripti ka një build-in kompilator dhe në kohë reale ekzekutohet më shpejt dhe me një rrjesht komande, kurse në rastin e programit C për sa kohë nuk kemi krijuar file-in që më pas mund ta ekzekutojmë do një kohë pak më të gjatë. Për të krijuar këtë file, përdorim komandën *gcc*.

### Ushtrimi 4:

Shkruaj një shell script që merr si argumenta një emër file-i dhe dy numra që përfaqësojnë rreshtin e fillimit dhe të mbarimit dhe shfaq të gjithë rreshtat e file-it ndërmjet rreshtit të fillimit dhe mbarimit.

*File-i që do shqyrtojmë:*



```
(kali@kali)-[~]
$ cat ushtrimi4.txt
Technology has revolutionized the way we live, work, and communicate. From smartphones to artificial intelligence, the advancements in technology have created endless opportunities for innovation. These tools not only simplify our daily lives but also open doors to solutions for complex global challenges. As we continue to explore the potential of these advancements, the ethical implications and societal impacts remain crucial considerations. Nature offers us a breathtaking array of wonders, from towering mountain ranges to serene ocean waves. Every element of the natural world tells a story of resilience and interconnectedness. The rustling of leaves, the chirping of birds, and the vibrant colors of a sunset remind us of the beauty that surrounds us. Protecting this fragile ecosystem is a responsibility we all share.

(kali@kali)-[~]
$
```

Shkruajmë në nano kodin e dëshiruar të shell script-it:

```
GNU nano 8.1 ushtrimi4.sh
if [ "$#" -ne 3 ]; then
    echo "Usage: $0 <file> <start_line> <end_line>"
    exit 1
fi

file=$1
start=$2
end=$3

if ! [[ -f "$file" ]]; then
    echo "Error: File '$file' does not exist."
    exit 1
fi

if ! [[ "$start" =~ ^[0-9]+$ ]] || ! [[ "$end" =~ ^[0-9]+$ ]]; then
    echo "Error: Start and end lines must be positive integers."
    exit 1
fi

if (( start > end )); then
    echo "Error: Start line cannot be greater than end line."
    exit 1
fi

sed -n "${start},${end}p" "$file"
```

Siç vëmë re në kodin e mësipërm si fillim kontrollojmë argumentat që dëshirojmë me një kusht *if*. Më pas i'u japim vlerë tre variablave me argumentat e marra, dhe çekojmë nëse file-i i marrë ekziston në librari. Gjithashtu vendosim dhe një kusht për rrjeshtat. Me komandën `sed -n "$p"` kryejmë funksionin që dëshirojmë.

Tani le të ekzekutojmë kodin me file-in *ushtrimi4lab1.txt* dhe si rrjesht fillimi 2 e mbarimi 5.

```
(kali@kali)-[~]
$ sh ushtrimi4.sh ushtrimi4.txt 2 4
created endless opportunities for innovation. These tools not only simplify our daily lives but also open doors to solutions for complex global challenges. As we continue to explore the potential of these advancements, the ethical implications and societal impacts remain crucial considerations. Nature offers us a breathtaking array of wonders, from towering mountain ranges to serene ocean waves. Every element of the natural world tells a story of

(kali@kali)-[~]
$
```

Në rastin që nuk vendosëm argumentat na i kërkoi programi siç tregohet dhe në figurë. Në rastin e dytë kodi u ekzekutua pa asnjë error.