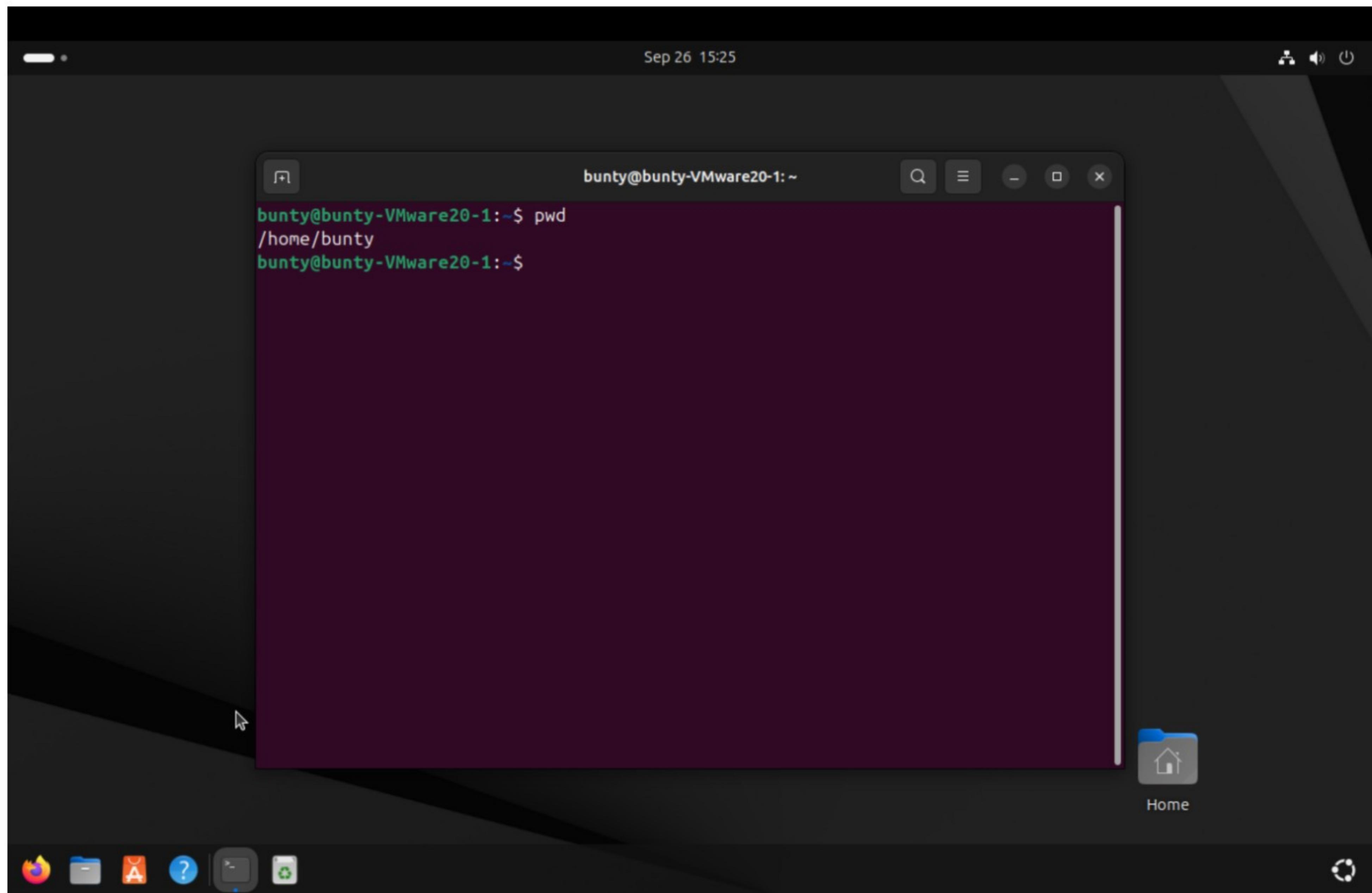
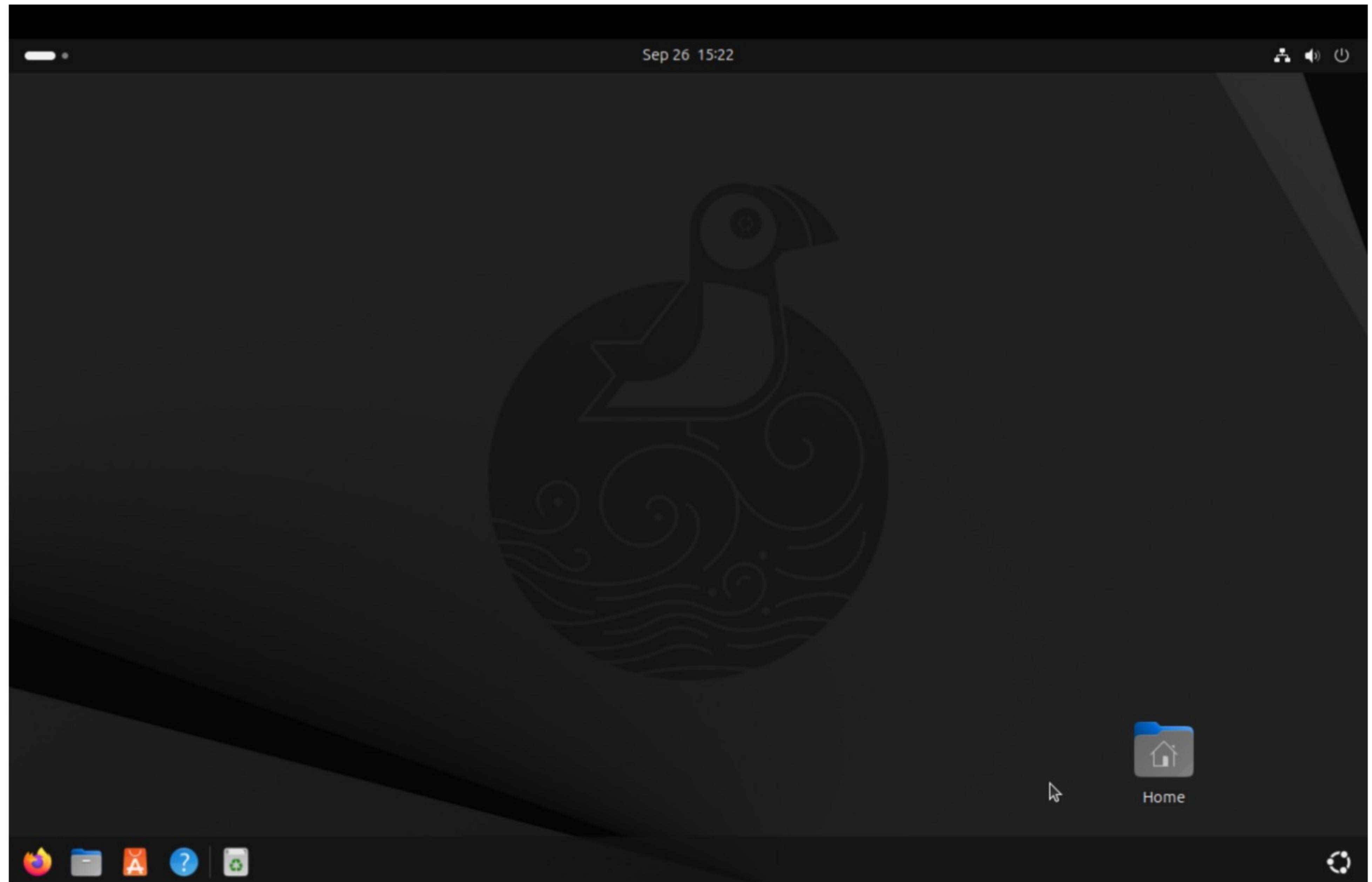


EXPERIMENT 1 - Install Virtual player and Linux.

Outputs:



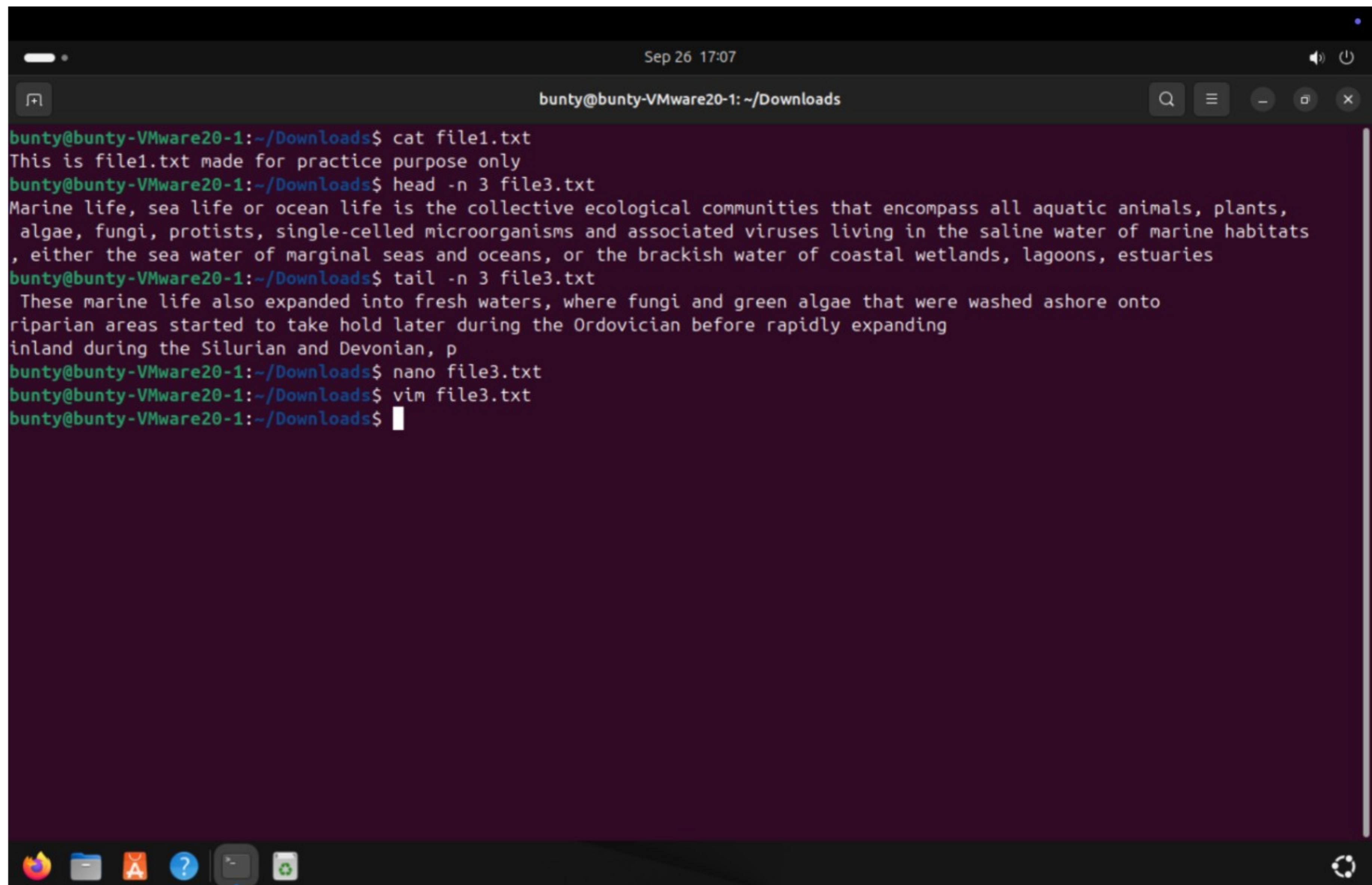


EXPERIMENT 2

Basic Commands in Linux

Basic Navigation Commands

1. **cat** - Display the contents.
2. **head and tail** - Show the beginning or end of a file .
3. **nano or vim** - Basic text editors.



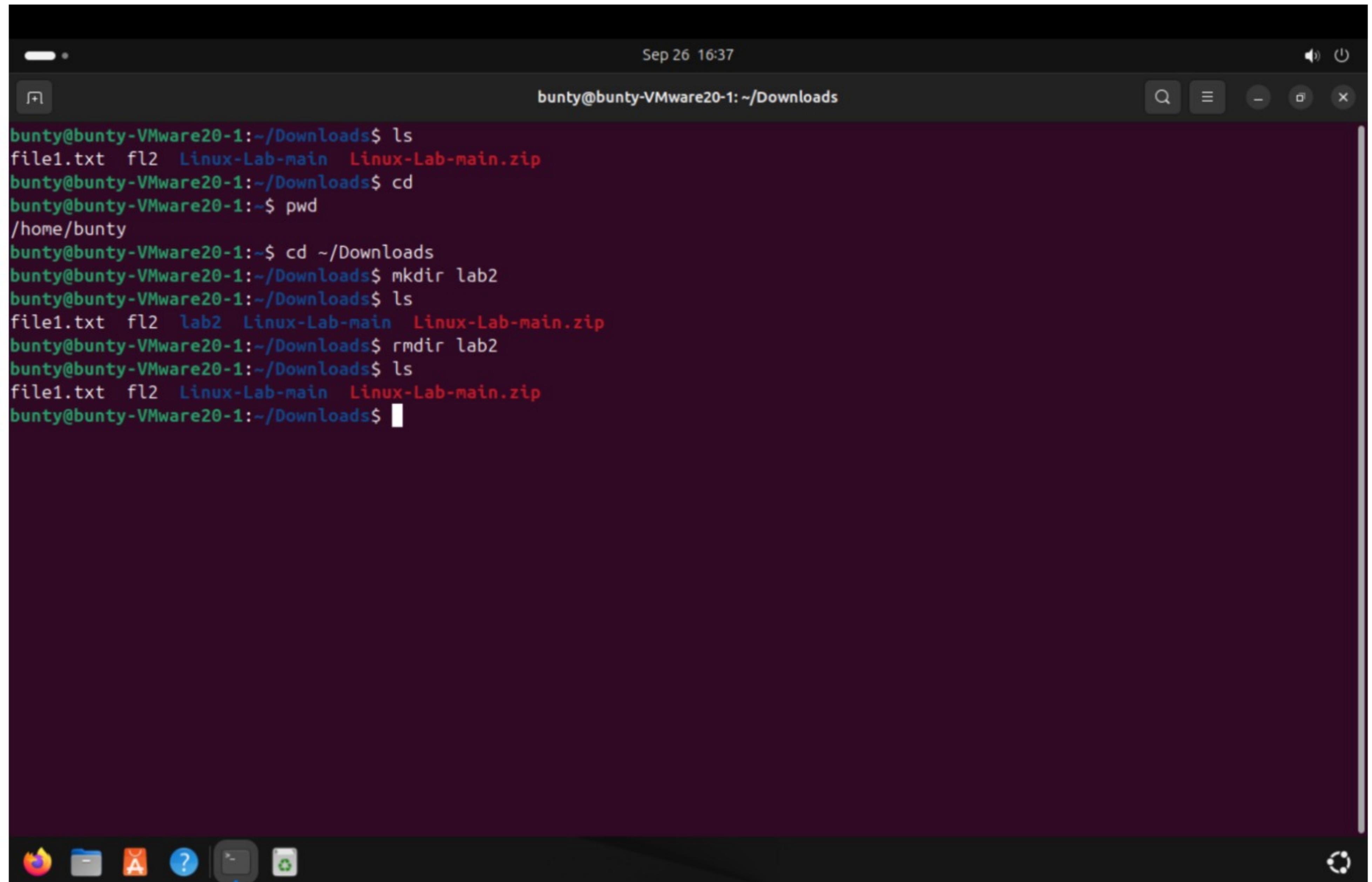
The screenshot shows a terminal window on a Linux desktop environment. The terminal title is "bunty@bunty-VMware20-1: ~/Downloads". The window contains the following command history:

```
Sep 26 17:07
bunty@bunty-VMware20-1:~/Downloads$ cat file1.txt
This is file1.txt made for practice purpose only
bunty@bunty-VMware20-1:~/Downloads$ head -n 3 file3.txt
Marine life, sea life or ocean life is the collective ecological communities that encompass all aquatic animals, plants,
algae, fungi, protists, single-celled microorganisms and associated viruses living in the saline water of marine habitats
, either the sea water of marginal seas and oceans, or the brackish water of coastal wetlands, lagoons, estuaries
bunty@bunty-VMware20-1:~/Downloads$ tail -n 3 file3.txt
These marine life also expanded into fresh waters, where fungi and green algae that were washed ashore onto
riparian areas started to take hold later during the Ordovician before rapidly expanding
inland during the Silurian and Devonian, p
bunty@bunty-VMware20-1:~/Downloads$ nano file3.txt
bunty@bunty-VMware20-1:~/Downloads$ vim file3.txt
bunty@bunty-VMware20-1:~/Downloads$
```

Baic Commands in Linux

Basic Navigation Commands

1. **ls** - Display the contents of a directory.
2. **cd** - Move between directories.
3. **pwd** - Show the current directory.
4. **mkdir** - Create new directory.
5. **rmdir** - Remove empty directory.



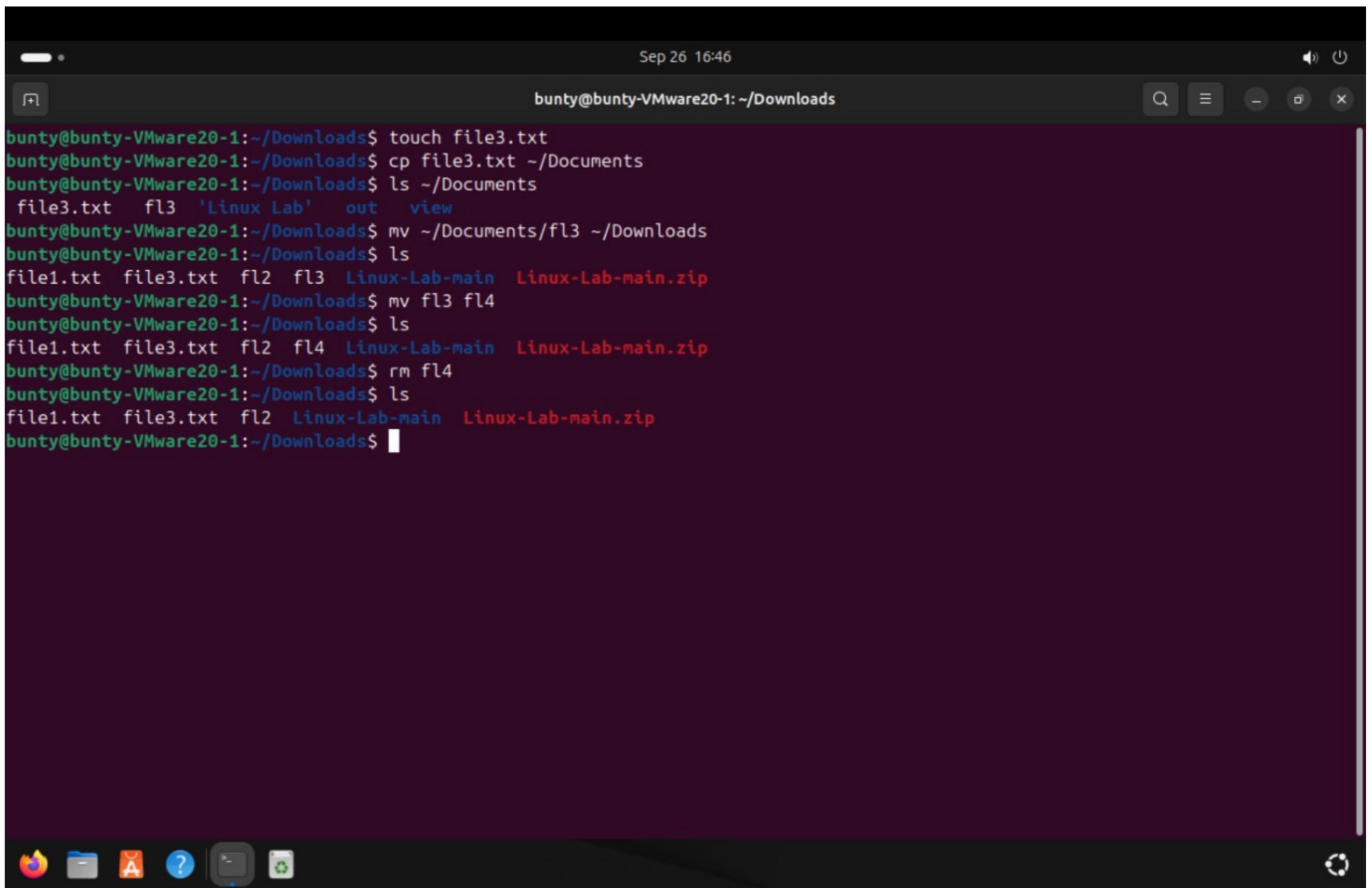
The screenshot shows a terminal window titled "bunty@bunty-VMware20-1: ~/Downloads". The terminal displays the following command-line session:

```
bunty@bunty-VMware20-1:~/Downloads$ ls
file1.txt fl2 Linux-Lab-main Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$ cd
bunty@bunty-VMware20-1:~$ pwd
/home/bunty
bunty@bunty-VMware20-1:~$ cd ~/Downloads
bunty@bunty-VMware20-1:~/Downloads$ mkdir lab2
bunty@bunty-VMware20-1:~/Downloads$ ls
file1.txt fl2 lab2 Linux-Lab-main Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$ rmdir lab2
bunty@bunty-VMware20-1:~/Downloads$ ls
file1.txt fl2 Linux-Lab-main Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$
```

File Operation

1. **touch** - create a new file.
2. **cp** - Copy files and directories.
3. **mv** - Move or rename files and directories.

4. **rm** - Delete files and directories.



The screenshot shows a terminal window titled "bunty@bunty-VMware20-1: ~/Downloads". The terminal displays the following command history:

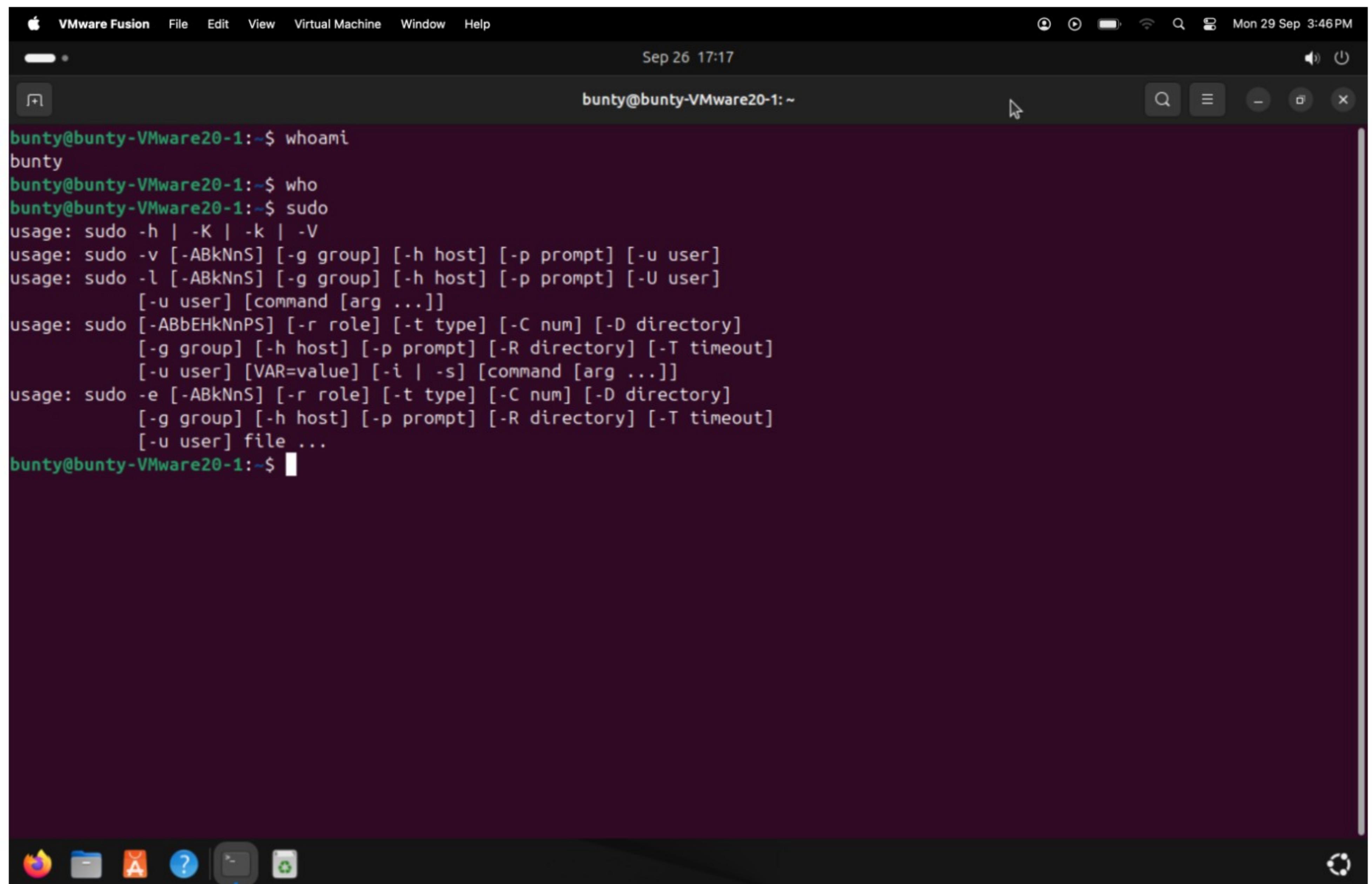
```
bunty@bunty-VMware20-1:~/Downloads$ touch file3.txt
bunty@bunty-VMware20-1:~/Downloads$ cp file3.txt ~/Documents
bunty@bunty-VMware20-1:~/Downloads$ ls ~/Documents
file3.txt fl3 'Linux Lab' out view
bunty@bunty-VMware20-1:~/Downloads$ mv ~/Documents/fl3 ~/Downloads
bunty@bunty-VMware20-1:~/Downloads$ ls
file1.txt file3.txt fl2 fl3 Linux-Lab-main Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$ mv fl3 fl4
bunty@bunty-VMware20-1:~/Downloads$ ls
file1.txt file3.txt fl2 Linux-Lab-main Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$ rm fl4
bunty@bunty-VMware20-1:~/Downloads$ ls
file1.txt file3.txt fl2 Linux-Lab-main Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$
```

The terminal window has a dark theme with light-colored text. The bottom of the window shows the desktop environment's dock with icons for a browser, file manager, terminal, and others.

File Viewing and Editing

User Management

1. **whoami** - Display the current user.
2. **passwd** - Change the password for the current user.
3. **sudo** - Execute commands with administrative privileges.
4. **who** - Show users currently.



```
bunty@bunty-VMware20-1:~$ whoami
bunty
bunty@bunty-VMware20-1:~$ who
bunty@bunty-VMware20-1:~$ sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABkNnS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABkNnS] [-g group] [-h host] [-p prompt] [-U user]
      [-u user] [command [arg ...]]
usage: sudo [-ABbEHkNnPS] [-r role] [-t type] [-C num] [-D directory]
      [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
      [-u user] [VAR=value] [-i | -s] [command [arg ...]]
usage: sudo -e [-ABkNnS] [-r role] [-t type] [-C num] [-D directory]
      [-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
      [-u user] file ...
bunty@bunty-VMware20-1:~$
```

System information

1. **uname** - Display system information.
2. **df (disk free)** - Show disk space usage.
3. **top or htop** - Monitor system process.
4. **history** - Show history.

```
Sep 26 17:24
bunty@bunty-VMware20-1:~
```

```
bunty@bunty-VMware20-1:~$ df
Filesystem 1K-blocks Used Available Use% Mounted on
tmpfs 346852 1780 345072 1% /run
/dev/nvme0n1p2 19503340 11372728 7114548 62% /
tmpfs 1734256 0 1734256 0% /dev/shm
efivarfs 256 33 224 13% /sys/firmware/efi/efivars
tmpfs 5120 8 5112 1% /run/lock
tmpfs 1024 0 1024 0% /run/credentials/systemd-journald.service
tmpfs 1024 0 1024 0% /run/credentials/systemd-resolved.service
/dev/nvme0n1p1 973952 6516 967436 1% /boot/efi
tmpfs 1734260 8 1734252 1% /tmp
tmpfs 346848 120 346728 1% /run/user/1000
bunty@bunty-VMware20-1:~$ uname
Linux
bunty@bunty-VMware20-1:~$
```



```
Sep 26 17:26
bunty@bunty-VMware20-1:~
```

```
top - 17:26:37 up 58 min, 2 users, load average: 0.00, 0.01, 0.00
Tasks: 289 total, 1 running, 288 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.8 us, 1.2 sy, 0.0 ni, 96.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3387.2 total, 812.0 free, 1082.1 used, 1688.1 buff/cache
MiB Swap: 3264.0 total, 3264.0 free, 0.0 used. 2305.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1681	bunty	20	0	3993380	388548	137428	S	7.0	11.2	1:53.53	gnome-shell
1835	bunty	20	0	388928	10644	6676	S	0.3	0.3	0:03.43	ibus-daemon
3956	root	20	0	0	0	0	I	0.3	0.0	0:00.29	kworker/u8:0-events_unbound
4115	bunty	20	0	709664	55360	44440	S	0.3	1.6	0:01.04	gnome-terminal-
4368	bunty	20	0	12876	5252	3204	R	0.3	0.2	0:00.05	top
1	root	20	0	24932	14244	9508	S	0.0	0.4	0:00.65	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	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_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-sync_wq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-kvfree_rcu_reclaim
7	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-slub_flushwq
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-netns
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-kblockd
13	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-mm_percpu_wq
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
15	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
16	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
17	root	20	0	0	0	0	S	0.0	0.0	0:00.10	ksoftirqd/0
18	root	20	0	0	0	0	I	0.0	0.0	0:00.48	rcu_preempt
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_exp_par_gp_kthread_worker/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_exp_gp_kthread_worker
21	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	migration/0

Sep 26 17:26
bunty@bunty-VMware20-1:~

```

0[|||||] 3.9%] Tasks: 113, 358 thr, 176 kthr; 1 running
1[|||] 2.6%] Load average: 0.00 0.01 0.00
Mem[|||||||||] 924M/3.31G] Uptime: 00:58:56
Swp[ 0K/3.19G]

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
1681 bunty 20 0 3883M 379M 134M S 1.3 11.2 0:35.60 /usr/bin/gnome-shell
4371 bunty 20 0 10952 4340 2932 R 1.3 0.1 0:00.13 htop
1739 bunty 20 0 3883M 379M 0 S 0.7 11.2 0:35.19 /usr/bin/gnome-shell
1740 bunty 20 0 3883M 379M 0 S 0.7 11.2 0:37.19 /usr/bin/gnome-shell
2805 bunty 39 19 406M 29644 0 S 0.7 0.9 0:00.09 /usr/libexec/localsearch-extractor-3 --socket-fd 3
4372 bunty 20 0 3883M 379M 0 S 0.7 11.2 0:00.01 /usr/bin/gnome-shell
1 root 20 0 24932 14244 9508 S 0.0 0.4 0:00.65 /sbin/init splash
384 root 19 -1 51004 18248 16712 S 0.0 0.5 0:00.24 /usr/lib/systemd/systemd-journald
401 systemd-oo 20 0 17596 6752 5984 S 0.0 0.2 0:01.48 /usr/lib/systemd/systemd-oomd
420 systemd-re 20 0 22840 13404 10844 S 0.0 0.4 0:00.11 /usr/lib/systemd/systemd-resolved
421 systemd-ti 20 0 91256 7080 6184 S 0.0 0.2 0:00.03 /usr/lib/systemd/systemd-timesyncd
429 root 20 0 38956 11720 7240 S 0.0 0.3 0:00.10 /usr/lib/systemd/systemd-udevd
452 systemd-ti 20 0 91256 7080 0 S 0.0 0.2 0:00.00 /usr/lib/systemd/systemd-timesyncd
1117 avahi 20 0 6568 3540 3284 S 0.0 0.1 0:00.14 avahi-daemon: running [bunty-VMware20-3.local]
1118 messagebus 20 0 11768 7376 3664 S 0.0 0.2 0:00.33 @dbus-daemon --system --address=systemd: --nofork --nopidfi
1122 gnome-remo 20 0 447M 15048 12744 S 0.0 0.4 0:00.00 /usr/libexec/gnome-remote-desktop-daemon --system
1125 polkitd 20 0 380M 10876 7508 S 0.0 0.3 0:00.04 /usr/lib/polkit-1/polkitd --no-debug --log-level=notice
1130 root 20 0 1804M 37496 23840 S 0.0 1.1 0:00.47 /usr/lib/snapd/snapd
1131 root 20 0 304M 6700 6060 S 0.0 0.2 0:00.01 /usr/libexec/accounts-daemon
1132 root 20 0 9560 2484 2356 S 0.0 0.1 0:00.01 /usr/sbin/cron -f -P
1133 root 20 0 303M 6488 5976 S 0.0 0.2 0:00.00 /usr/libexec/switcheroo-control

```

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

Sep 26 17:27
bunty@bunty-VMware20-1:~

```

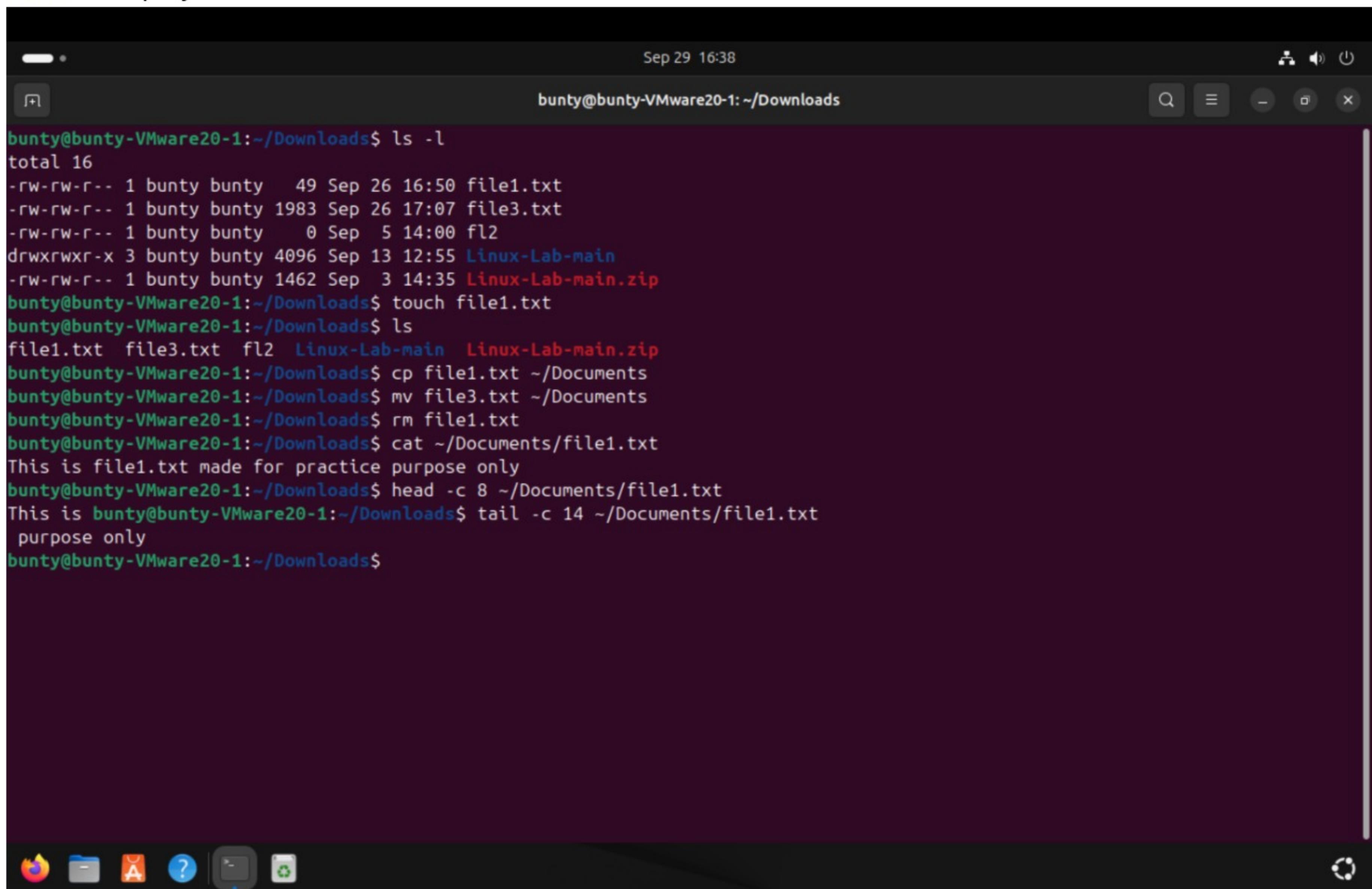
bunty@bunty-VMware20-1:~$ htop
bunty@bunty-VMware20-1:~$ history 20
261 cd
262 clear
263 whoami
264 who
265 sudo
266 passwd
267 clear
268 whoami
269 who
270 sudo
271 passwd
272 df
273 uname
274 htop
275 sudo apt install htop
276 clear
277 top
278 clear
279 htop
280 history 20
bunty@bunty-VMware20-1:~$
```

EXPERIMENT 3

File and Directories Command

Working with files

1. **touch** - Create an empty file .
2. **cp** - Copy file and directories .
3. **mv** - Move or rename files and directories .
4. **rm** - Remove files and directories .
5. **cat** - Concatenate and display the contents of a file .
6. **head** - Display the first few lines of a file .
7. **tail** - Display the last few lines of a file .



The screenshot shows a terminal window on a Linux desktop environment. The terminal window has a dark background and light-colored text. At the top, it displays the date and time as "Sep 29 16:38" and the user information as "bunty@bunty-VMware20-1: ~/Downloads". The window title bar also shows the same information. The terminal content is as follows:

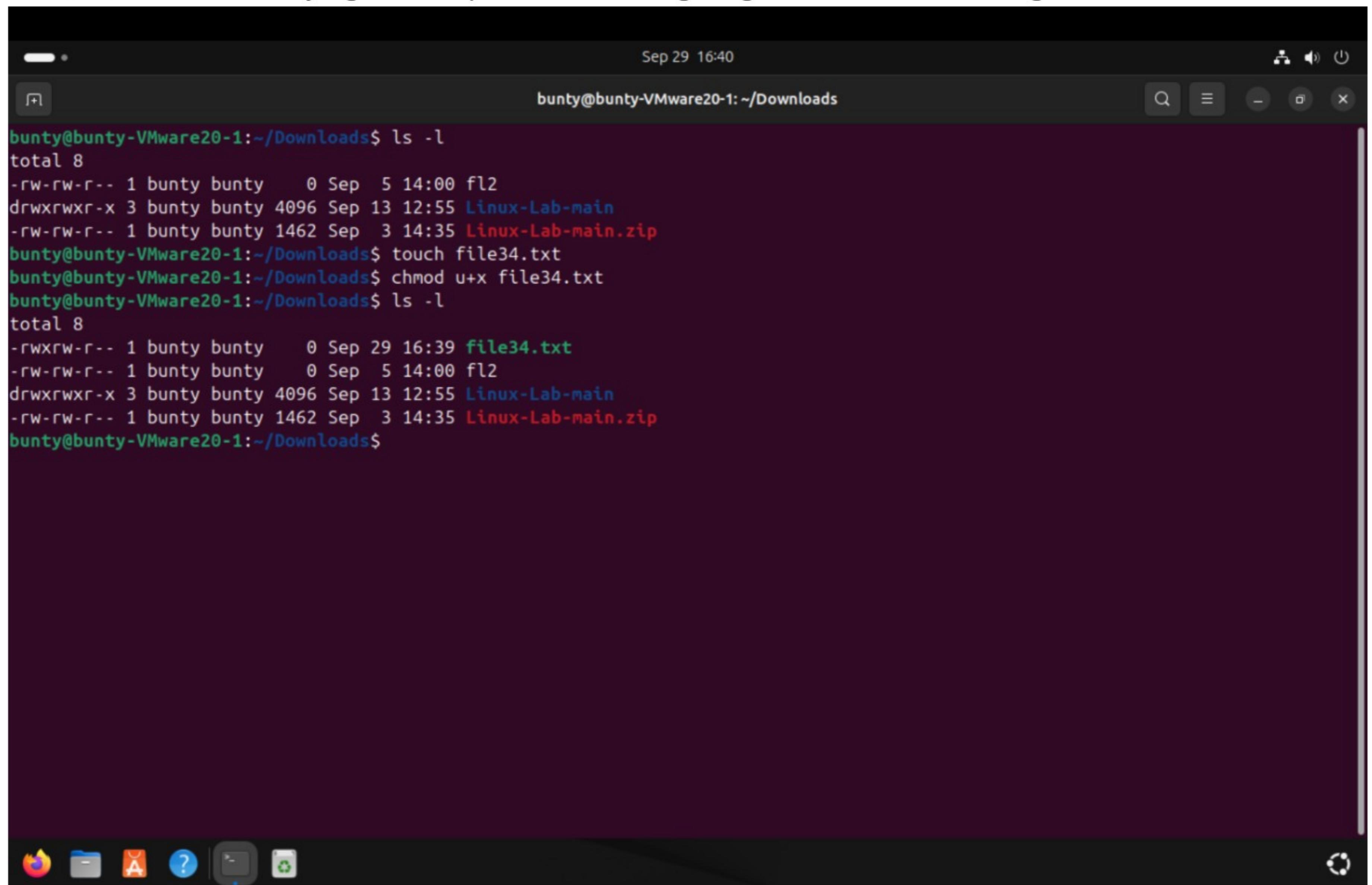
```
bunty@bunty-VMware20-1:~/Downloads$ ls -l
total 16
-rw-rw-r-- 1 bunty bunty 49 Sep 26 16:50 file1.txt
-rw-rw-r-- 1 bunty bunty 1983 Sep 26 17:07 file3.txt
-rw-rw-r-- 1 bunty bunty 0 Sep 5 14:00 fl2
drwxrwxr-x 3 bunty bunty 4096 Sep 13 12:55 Linux-Lab-main
-rw-rw-r-- 1 bunty bunty 1462 Sep 3 14:35 Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$ touch file1.txt
bunty@bunty-VMware20-1:~/Downloads$ ls
file1.txt file3.txt fl2 Linux-Lab-main Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$ cp file1.txt ~/Documents
bunty@bunty-VMware20-1:~/Downloads$ mv file3.txt ~/Documents
bunty@bunty-VMware20-1:~/Downloads$ rm file1.txt
bunty@bunty-VMware20-1:~/Downloads$ cat ~/Documents/file1.txt
This is file1.txt made for practice purpose only
bunty@bunty-VMware20-1:~/Downloads$ head -c 8 ~/Documents/file1.txt
This is bunty@bunty-VMware20-1:~/Downloads$ tail -c 14 ~/Documents/file1.txt
purpose only
bunty@bunty-VMware20-1:~/Downloads$
```

The desktop interface at the bottom includes icons for a browser, file manager, terminal, help, and system settings.

File Permission and Ownership

1. **ls -l** -

2. **chmod u+x** - for modifying the file permission and giving accesses of executing the file to the owner.



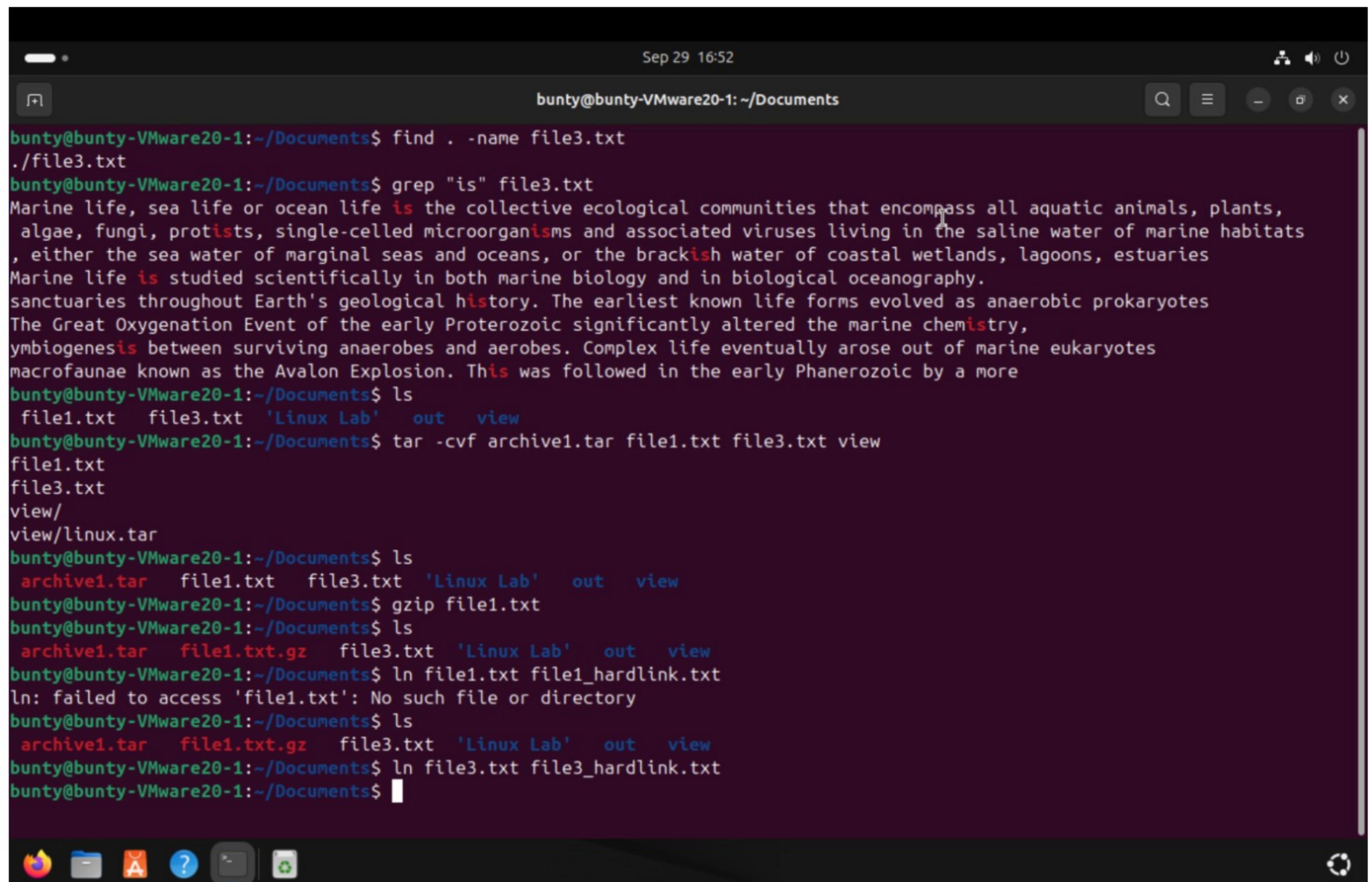
The screenshot shows a terminal window titled "bunty@bunty-VMware20-1: ~/Downloads". The terminal displays the following command sequence:

```
bunty@bunty-VMware20-1:~/Downloads$ ls -l
total 8
-rw-rw-r-- 1 bunty bunty 0 Sep  5 14:00 fl2
drwxrwxr-x 3 bunty bunty 4096 Sep 13 12:55 Linux-Lab-main
-rw-rw-r-- 1 bunty bunty 1462 Sep  3 14:35 Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$ touch file34.txt
bunty@bunty-VMware20-1:~/Downloads$ chmod u+x file34.txt
bunty@bunty-VMware20-1:~/Downloads$ ls -l
total 8
-rwxrw-r-- 1 bunty bunty 0 Sep 29 16:39 file34.txt
-rw-rw-r-- 1 bunty bunty 0 Sep  5 14:00 fl2
drwxrwxr-x 3 bunty bunty 4096 Sep 13 12:55 Linux-Lab-main
-rw-rw-r-- 1 bunty bunty 1462 Sep  3 14:35 Linux-Lab-main.zip
bunty@bunty-VMware20-1:~/Downloads$
```

The terminal window has a dark background and light-colored text. The bottom bar contains icons for a browser, file manager, terminal, and system status.

Advanced File and Directory Operations

1. **find** - Search for files and directories based on various criteria .
2. **grep** - Search for specific patterns within files .
3. **tar** - Archive file and directories into a single file .
4. **gzip/gunzip** - Compresss or decompress the file.
5. **ln** - Create hard and symbolic links .



The screenshot shows a terminal window with the following session:

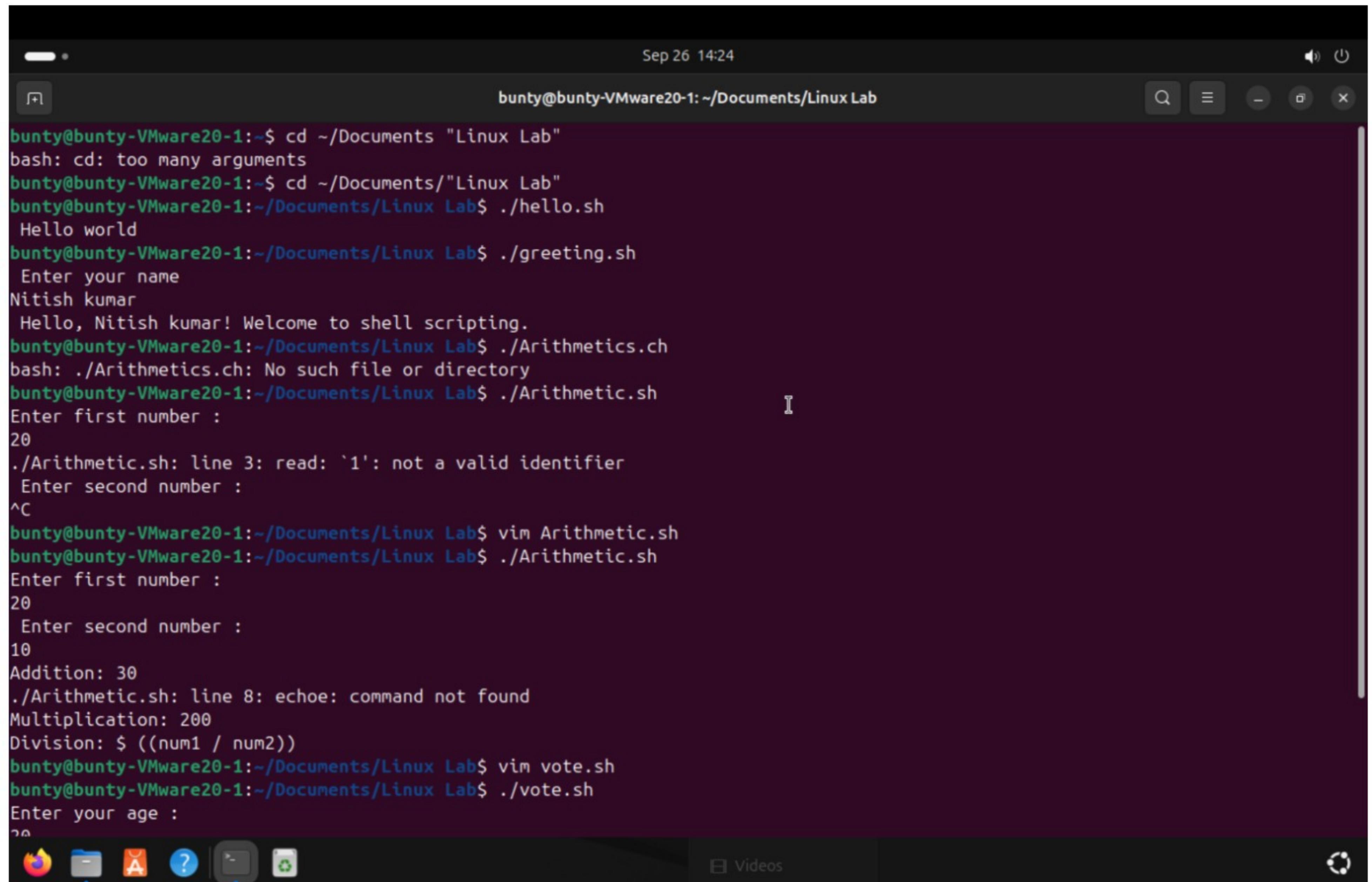
```
bunty@bunty-VMware20-1:~/Documents$ find . -name file3.txt
./file3.txt
bunty@bunty-VMware20-1:~/Documents$ grep "is" file3.txt
Marine life, sea life or ocean life is the collective ecological communities that encompass all aquatic animals, plants, algae, fungi, protists, single-celled microorganisms and associated viruses living in the saline water of marine habitats, either the sea water of marginal seas and oceans, or the brackish water of coastal wetlands, lagoons, estuaries
Marine life is studied scientifically in both marine biology and in biological oceanography.
sanctuaries throughout Earth's geological history. The earliest known life forms evolved as anaerobic prokaryotes
The Great Oxygenation Event of the early Proterozoic significantly altered the marine chemistry,
ybiogenesisis between surviving anaerobes and aerobes. Complex life eventually arose out of marine eukaryotes
macrofaunae known as the Avalon Explosion. This was followed in the early Phanerozoic by a more
bunty@bunty-VMware20-1:~/Documents$ ls
file1.txt  file3.txt  'Linux Lab'  out  view
bunty@bunty-VMware20-1:~/Documents$ tar -cvf archive1.tar file1.txt file3.txt view
file1.txt
file3.txt
view/
view/linux.tar
bunty@bunty-VMware20-1:~/Documents$ ls
archive1.tar  file1.txt  file3.txt  'Linux Lab'  out  view
bunty@bunty-VMware20-1:~/Documents$ gzip file1.txt
bunty@bunty-VMware20-1:~/Documents$ ls
archive1.tar  file1.txt.gz  file3.txt  'Linux Lab'  out  view
bunty@bunty-VMware20-1:~/Documents$ ln file1.txt file1_hardlink.txt
ln: failed to access 'file1.txt': No such file or directory
bunty@bunty-VMware20-1:~/Documents$ ls
archive1.tar  file1.txt.gz  file3.txt  'Linux Lab'  out  view
bunty@bunty-VMware20-1:~/Documents$ ln file3.txt file3_hardlink.txt
bunty@bunty-VMware20-1:~/Documents$
```

EXPERIMENT 4 SHELL PROGRAMMING

TASKS

1. Write a simple shell script that prints "Hello World" when executed.
2. Create the script that prompts the user to enter their name and then displays a personalized greeting.
3. Write a script that takes two numbers as inputs and performs various arithmetic operations like addition, subtraction, multiplication and division.
4. Create a script that asks the user to enter their age and displays the message based on whether they are eligible to vote or not.

Outputs



The screenshot shows a terminal window with a dark theme. The title bar reads "bunty@bunty-VMware20-1: ~/Documents/Linux Lab". The terminal output is as follows:

```
Sep 26 14:24
bunty@bunty-VMware20-1:~/Documents/Linux Lab$ cd ~/Documents "Linux Lab"
bash: cd: too many arguments
bunty@bunty-VMware20-1:~/Documents/"Linux Lab"
bunty@bunty-VMware20-1:~/Documents/Linux Lab$ ./hello.sh
Hello world
bunty@bunty-VMware20-1:~/Documents/Linux Lab$ ./greeting.sh
Enter your name
Nitish kumar
Hello, Nitish kumar! Welcome to shell scripting.
bunty@bunty-VMware20-1:~/Documents/Linux Lab$ ./Arithmetics.ch
bash: ./Arithmetics.ch: No such file or directory
bunty@bunty-VMware20-1:~/Documents/Linux Lab$ ./Arithmetic.sh
Enter first number :
20
./Arithmetic.sh: line 3: read: `1': not a valid identifier
Enter second number :
^C
bunty@bunty-VMware20-1:~/Documents/Linux Lab$ vim Arithmetic.sh
bunty@bunty-VMware20-1:~/Documents/Linux Lab$ ./Arithmetic.sh
Enter first number :
20
Enter second number :
10
Addition: 30
./Arithmetic.sh: line 8: echoe: command not found
Multiplication: 200
Division: $ ((num1 / num2))
bunty@bunty-VMware20-1:~/Documents/Linux Lab$ vim vote.sh
bunty@bunty-VMware20-1:~/Documents/Linux Lab$ ./vote.sh
Enter your age :
20
```

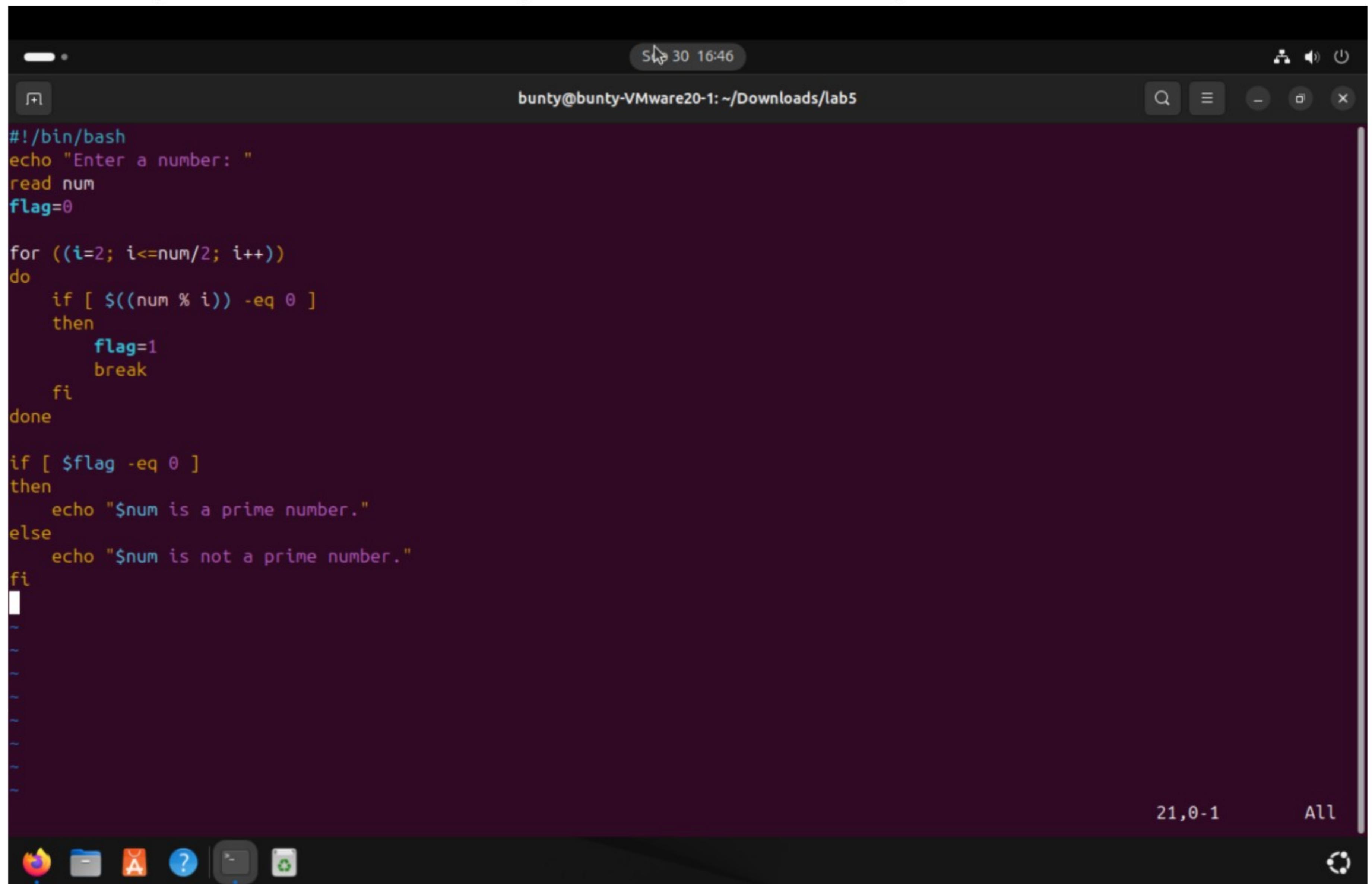
Nitish kumar
Hello, Nitish kumar! Welcome to shell scripting.
bunty@bunty-VMware20-1:~/Documents/Linux Lab\$./Arithmetics.ch
bash: ./Arithmetics.ch: No such file or directory
bunty@bunty-VMware20-1:~/Documents/Linux Lab\$./Arithmetic.sh
Enter first number :
20
. /Arithmetic.sh: line 3: read: '1': not a valid identifier
Enter second number :
^C
bunty@bunty-VMware20-1:~/Documents/Linux Lab\$ vim Arithmetic.sh
bunty@bunty-VMware20-1:~/Documents/Linux Lab\$./Arithmetic.sh
Enter first number :
20
Enter second number :
10
Addition: 30
. /Arithmetic.sh: line 8: echoe: command not found
Multiplication: 200
Division: \$ ((num1 / num2))
bunty@bunty-VMware20-1:~/Documents/Linux Lab\$ vim vote.sh
bunty@bunty-VMware20-1:~/Documents/Linux Lab\$./vote.sh
Enter your age :
20
Your are eligible to vote .
bunty@bunty-VMware20-1:~/Documents/Linux Lab\$./vote.sh
Enter your age :
10
sorry, you are not eligible to vote.
bunty@bunty-VMware20-1:~/Documents/Linux Lab\$

EXPERIMENT 5

Shell Programming

Tasks

1. Write a script that takes a number as input and checks whether it is prime number or not .



The screenshot shows a terminal window on a Linux desktop environment. The terminal title is "bunty@bunty-VMware20-1: ~/Downloads/lab5". The window contains a shell script written in bash. The script prompts the user to enter a number, reads it into the variable \$num, initializes a flag variable to 0, and then iterates from i=2 to num/2. If any divisor is found (i.e., if the remainder of num % i is 0), the flag is set to 1 and the loop is broken. After the loop, if the flag is still 0, the number is printed as prime; otherwise, it is printed as not prime.

```
#!/bin/bash
echo "Enter a number: "
read num
flag=0

for ((i=2; i<=num/2; i++))
do
    if [ $((num % i)) -eq 0 ]
    then
        flag=1
        break
    fi
done

if [ $flag -eq 0 ]
then
    echo "$num is a prime number."
else
    echo "$num is not a prime number."
fi
```

A screenshot of a Linux desktop environment showing a terminal window. The terminal title is "bunty@bunty-VMware20-1: ~/Downloads/lab5". The terminal content shows the execution of a shell script named "prime.sh". The user runs "vim prime.sh", "chmod 744 prime.sh", and then "./prime.sh". They are prompted to "Enter a number:" and enter "91". The output says "91 is not a prime number." The terminal window has a dark background with light-colored text. The desktop icons at the bottom include a browser, file manager, terminal, and others.

```
bunty@bunty-VMware20-1:~/Downloads/lab5$ vim prime.sh
bunty@bunty-VMware20-1:~/Downloads/lab5$ chmod 744 prime.sh
bunty@bunty-VMware20-1:~/Downloads/lab5$ ./prime.sh
Enter a number:
91
91 is not a prime number.
bunty@bunty-VMware20-1:~/Downloads/lab5$
```

2. Write a script that calculates the sum of the digits of a given number .

A screenshot of a Linux desktop environment showing a terminal window. The terminal title is "bunty@bunty-VMware20-1: ~/Downloads/lab5". The terminal content shows the creation of a shell script named "sum_digits.sh". The user starts by creating a new file with "vim sum_digits.sh", then adds code to read a number from the user, calculate the sum of its digits using a loop, and print the result. Finally, they save the file with ":wq". The terminal window has a dark background with light-colored text. The desktop icons at the bottom are visible.

```
#!/bin/bash
echo "Enter a number: "
read num
sum=0

while [ $num -gt 0 ]
do
    digit=$((num % 10))
    sum=$((sum + digit))
    num=$((num / 10))
done

echo "Sum of digits: $sum"

:wq
```

```
Sep 30 16:59
bunty@bunty-VMware20-1: ~/Downloads/lab5$ ./sod.sh
Enter a number:
456
Sum of digits: 15
bunty@bunty-VMware20-1: ~/Downloads/lab5$
```

3. Create a script that checks whether a given number is an Armstrong number or not .

```
#!/bin/bash
echo "Enter a number: "
read num
temp=$num
n=${#num}      # number of digits
sum=0

while [ $temp -gt 0 ]
do
    digit=$((temp % 10))
    sum=$((sum + digit**n))
    temp=$((temp / 10))
done

if [ $sum -eq $num ]
then
    echo "$num is an Armstrong number."
else
    echo "$num is not an Armstrong number."
fi
```

21,0-1

All

The screenshot shows a terminal window titled "bunty@bunty-VMware20-1: ~/Downloads/lab5". The window has a dark theme with a light-colored terminal area. The terminal displays the following command and its output:

```
bunty@bunty-VMware20-1:~/Downloads/lab5$ ./amstrong.sh
Enter a number:
153
153 is an Armstrong number.
bunty@bunty-VMware20-1:~/Downloads/lab5$
```

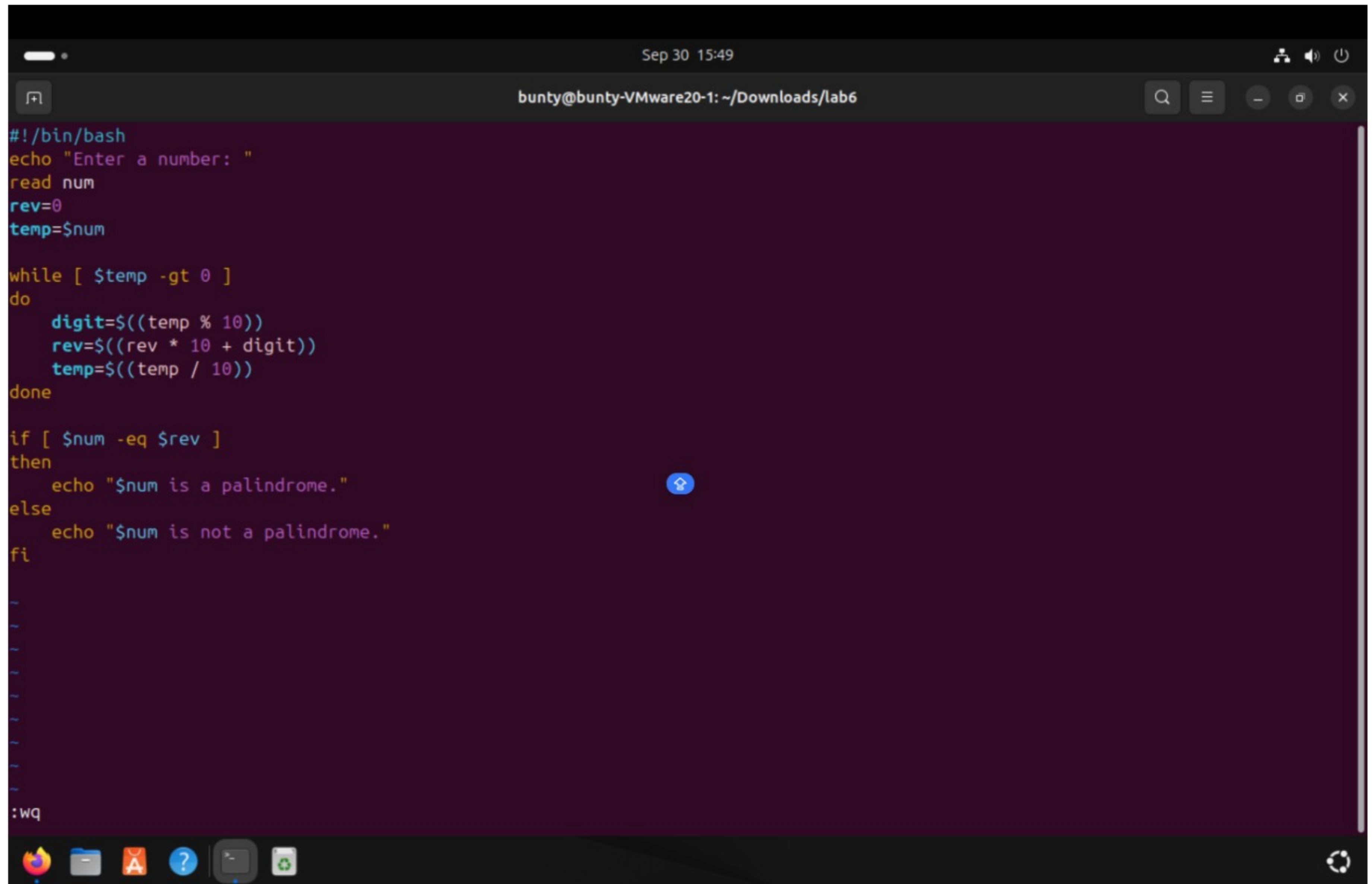
The terminal window includes standard Linux window controls at the top and a dock with icons at the bottom.

EXPERIMENT 6

Shell Programming

TASKS

1. Write a script that checks whether a given number is a palindrome or not . A palindrome number reads the same backward as forward.



The screenshot shows a terminal window on a Linux desktop. The title bar indicates it's running on a VMware VM, specifically version 20. The terminal window has a dark background and displays a shell script in white text. The script uses a while loop to reverse a number by extracting digits and appending them to a variable \$rev. It then compares the original number \$num with the reversed number \$rev to determine if it's a palindrome. The terminal window also shows the user's command history at the bottom, ending with ':wq' to save and quit.

```
#!/bin/bash
echo "Enter a number: "
read num
rev=0
temp=$num

while [ $temp -gt 0 ]
do
    digit=$((temp % 10))
    rev=$((rev * 10 + digit))
    temp=$((temp / 10))
done

if [ $num -eq $rev ]
then
    echo "$num is a palindrome."
else
    echo "$num is not a palindrome."
fi

:-
:-
:-
:-
:-
:-
:-
:wq
```

The screenshot shows a terminal window with a dark theme. The title bar indicates the date and time as "Sep 30 15:51" and the user as "bunty@bunty-VMware20-1: ~/Downloads/lab6". The terminal content shows two executions of a script named "palindrome.sh". In the first run, the user enters "454" and receives the output "454 is a palindrome.". In the second run, the user enters "456" and receives the output "456 is not a palindrome.". The terminal window has a standard Linux-style interface with icons for file operations at the bottom.

```
bunty@bunty-VMware20-1:~/Downloads/lab6$ ./palindrome.sh
Enter a number:
454
454 is a palindrome.
bunty@bunty-VMware20-1:~/Downloads/lab6$ ./palindrome.sh
Enter a number:
456
456 is not a palindrome.
bunty@bunty-VMware20-1:~/Downloads/lab6$
```

2. Write a script that calculates the greatest common divisor (GCD) and the least common multiple (LCM) of two given numbers.

The screenshot shows a terminal window with a dark theme. The title bar indicates the date and time as "Sep 30 15:52" and the user as "bunty@bunty-VMware20-1: ~/Downloads/lab6". The terminal content shows a script being run. The script uses the Euclidean algorithm to find the GCD of two input numbers, a and b. It then calculates the LCM using the formula $\text{LCM} = (\text{a} * \text{b}) / \text{GCD}$. The script outputs "GCD: \$gcd" and "LCM: \$lcm". The terminal window has a standard Linux-style interface with icons for file operations at the bottom.

```
#!/bin/bash
echo "Enter two numbers: "
read a b

x=$a
y=$b
while [ $y -ne 0 ]
do
    temp=$y
    y=$((x % y))
    x=$temp
done
gcd=$x

lcm=$(( (a * b) / gcd ))

echo "GCD: $gcd"
echo "LCM: $lcm"

:wa
```

The screenshot shows a terminal window with a dark theme. The title bar indicates the date and time as Sep 30 15:54. The command entered is `bunty@bunty-VMware20-1:~/Downloads/lab6$./GCD_LCM.sh`. The script prompts for two numbers, receives input of 45 and 72, calculates the GCD as 9, and the LCM as 360.

```
bunty@bunty-VMware20-1:~/Downloads/lab6$ ./GCD_LCM.sh
Enter two numbers:
45 72
GCD: 9
LCM: 360
bunty@bunty-VMware20-1:~/Downloads/lab6$
```

3. Create a script that takes multiple numbers as input and sorts them in ascending or descending order.

The screenshot shows a terminal window with a dark theme. The title bar indicates the date and time as Sep 30 15:56. The command entered is `bunty@bunty-VMware20-1:~/Downloads/lab6$`. Below the prompt, a script is displayed:

```
#!/bin/bash
echo "Enter numbers separated by space: "
read -a arr

echo "Ascending Order: "
printf "%s\n" "${arr[@]}" | sort -n

echo "Descending Order: "
printf "%s\n" "${arr[@]}" | sort -nr
```

The script asks for input, reads it into an array, and then prints the array elements sorted in ascending and descending order.

The screenshot shows a terminal window titled "bunty@bunty-VMware20-1: ~/Downloads/lab6". The window title bar also displays the date and time: "Sep 30 15:57". The terminal content is as follows:

```
bunty@bunty-VMware20-1:~/Downloads/lab6$ ./sort.sh
Enter numbers separated by space:
5 2 12 10
Ascending Order:
2
5
10
12
Descending Order:
12
10
5
2
bunty@bunty-VMware20-1:~/Downloads/lab6$
```

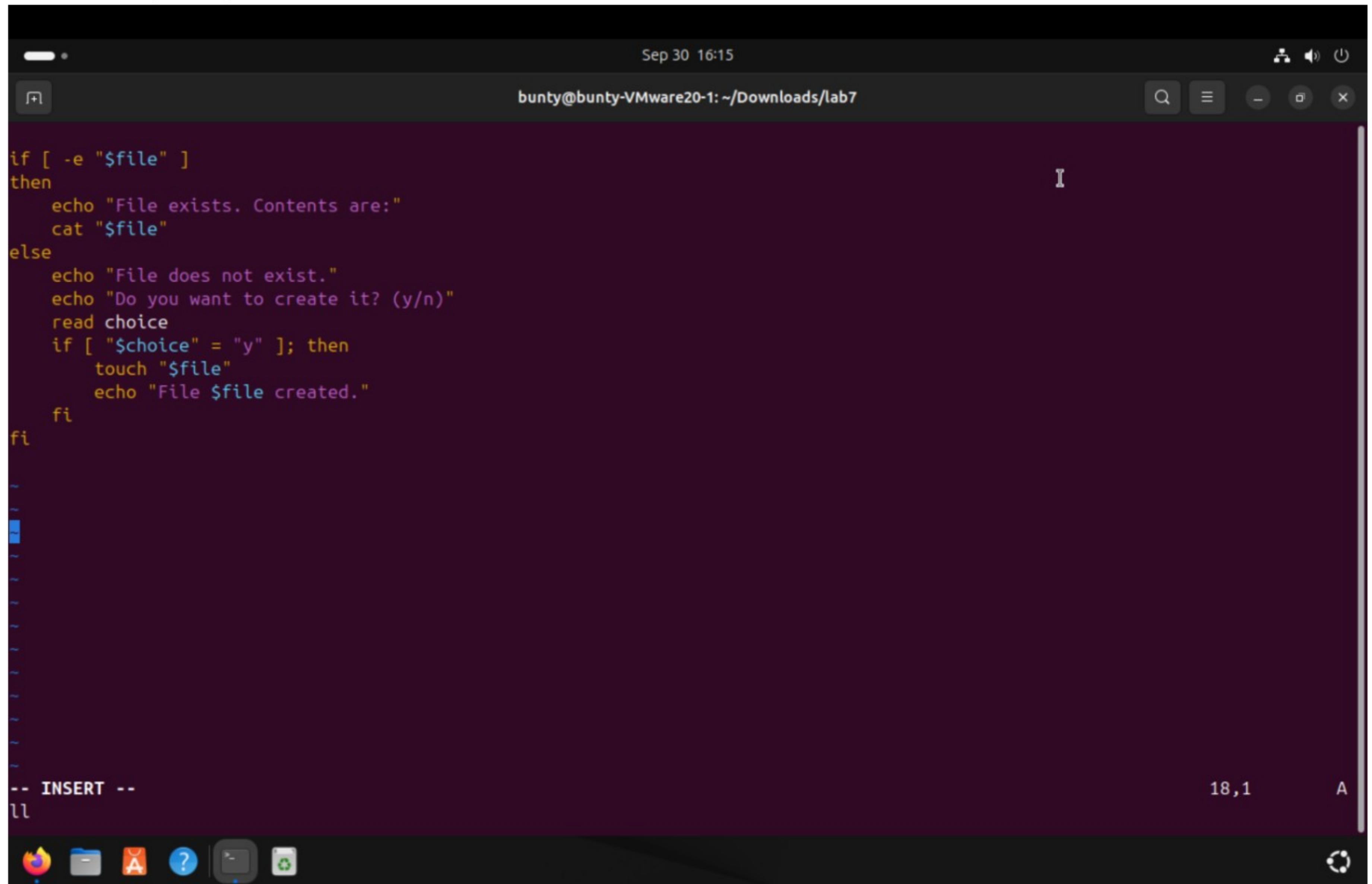
The terminal is running on a dark-themed desktop environment, likely elementary OS. The taskbar at the bottom shows icons for the Dash, Home, File Manager, Terminal, and other system applications.

EXPERIMENT 7

Shell Programming

TASKS

1. Write a script that takes a filename as input and checks if it exists. If the file exists, display its content; otherwise, prompt the user to create the file.



The screenshot shows a terminal window titled "bunty@bunty-VMware20-1: ~/Downloads/lab7". The terminal displays a shell script with syntax highlighting. The script uses an if-then-else construct to check if a file exists. If it does, it prints the file's contents. If it doesn't, it prompts the user for confirmation to create the file. The script ends with a "fi" at line 18,1. The terminal interface includes a status bar at the bottom with icons for file operations like copy, paste, and search.

```
if [ -e "$file" ]
then
    echo "File exists. Contents are:"
    cat "$file"
else
    echo "File does not exist."
    echo "Do you want to create it? (y/n)"
    read choice
    if [ "$choice" = "y" ]; then
        touch "$file"
        echo "File $file created."
    fi
fi
```

The screenshot shows a terminal window with a dark theme. The title bar indicates the session is on a VMware host, and the date and time are Sep 30 16:17. The command history shows:

```
bunty@bunty-VMware20-1:~/Downloads/lab7$ vim file_exist.sh
bunty@bunty-VMware20-1:~/Downloads/lab7$ chmod 744 file_exist.sh
bunty@bunty-VMware20-1:~/Downloads/lab7$ ./file_exist.sh
Enter filename:
file1.txt
File exists. Contents are:
This is file1.txt of lab7
bunty@bunty-VMware20-1:~/Downloads/lab7$
```

The terminal window has a standard Linux-style interface with icons at the bottom.

2. Create a script that prints the number from 1 to 10 using the loop .

The screenshot shows a terminal window with a dark theme. The title bar indicates the session is on a VMware host, and the date and time are Sep 30 16:18. The command history shows:

```
#!/bin/bash
for i in {1..10}
do
    echo $i
done
```

The terminal window has a standard Linux-style interface with icons at the bottom. The status bar at the bottom right shows "6,1 All".

```
Sep 30 16:19
bunty@bunty-VMware20-1: ~/Downloads/lab7
bunty@bunty-VMware20-1:~/Downloads/lab7$ vim prime1_10.sh
bunty@bunty-VMware20-1:~/Downloads/lab7$ chmod 744 prime1_10.sh
bunty@bunty-VMware20-1:~/Downloads/lab7$ ./prime1_10.sh
1
2
3
4
5
6
7
8
9
10
bunty@bunty-VMware20-1:~/Downloads/lab7$
```

3. Write a script that takes a filename as a command line argument and counts the number of lines , words and characters in that file.

```
Sep 30 16:20
bunty@bunty-VMware20-1: ~/Downloads/lab7

#!/bin/bash
if [ $# -eq 0 ]
then
    echo "Usage: $0 filename"
    exit 1
fi

file=$1

if [ -e "$file" ]
then
    echo "Lines: $(wc -l < $file)"
    echo "Words: $(wc -w < $file)"
    echo "Characters: $(wc -m < $file)"
else
    echo "File not found!"
fi

:wq
```

The screenshot shows a terminal window with a dark background. At the top, it displays the date and time: Sep 30 16:22. The title bar shows the user's name and the current directory: bunty@bunty-VMware20-1: ~/Downloads/lab7. The command entered was ./count.sh file1.txt. The output of the script is displayed below, showing the results for Lines, Words, and Characters.

```
bunty@bunty-VMware20-1:~/Downloads/lab7$ ./count.sh file1.txt
Lines: 1
Words: 5
Characters: 26
bunty@bunty-VMware20-1:~/Downloads/lab7$
```

4. Create a script that defines a function to calculate the factorial of a given number and call that function with different inputs.

The screenshot shows a terminal window with a dark background. At the top, it displays the date and time: Sep 30 16:24. The title bar shows the user's name and the current directory: bunty@bunty-VMware20-1: ~/Downloads/lab7. The command entered is a bash script that defines a function factorial() to calculate the factorial of a given number num. The script then calls this function for numbers 5, 7, and 10, and finally exits with :wq.

```
#!/bin/bash
factorial() {
    num=$1
    fact=1
    while [ $num -gt 1 ]
    do
        fact=$((fact * num))
        num=$((num - 1))
    done
    echo $fact
}

echo "Factorial of 5 is: $(factorial 5)"
echo "Factorial of 7 is: $(factorial 7)"
echo "Factorial of 10 is: $(factorial 10)"

:wq
```

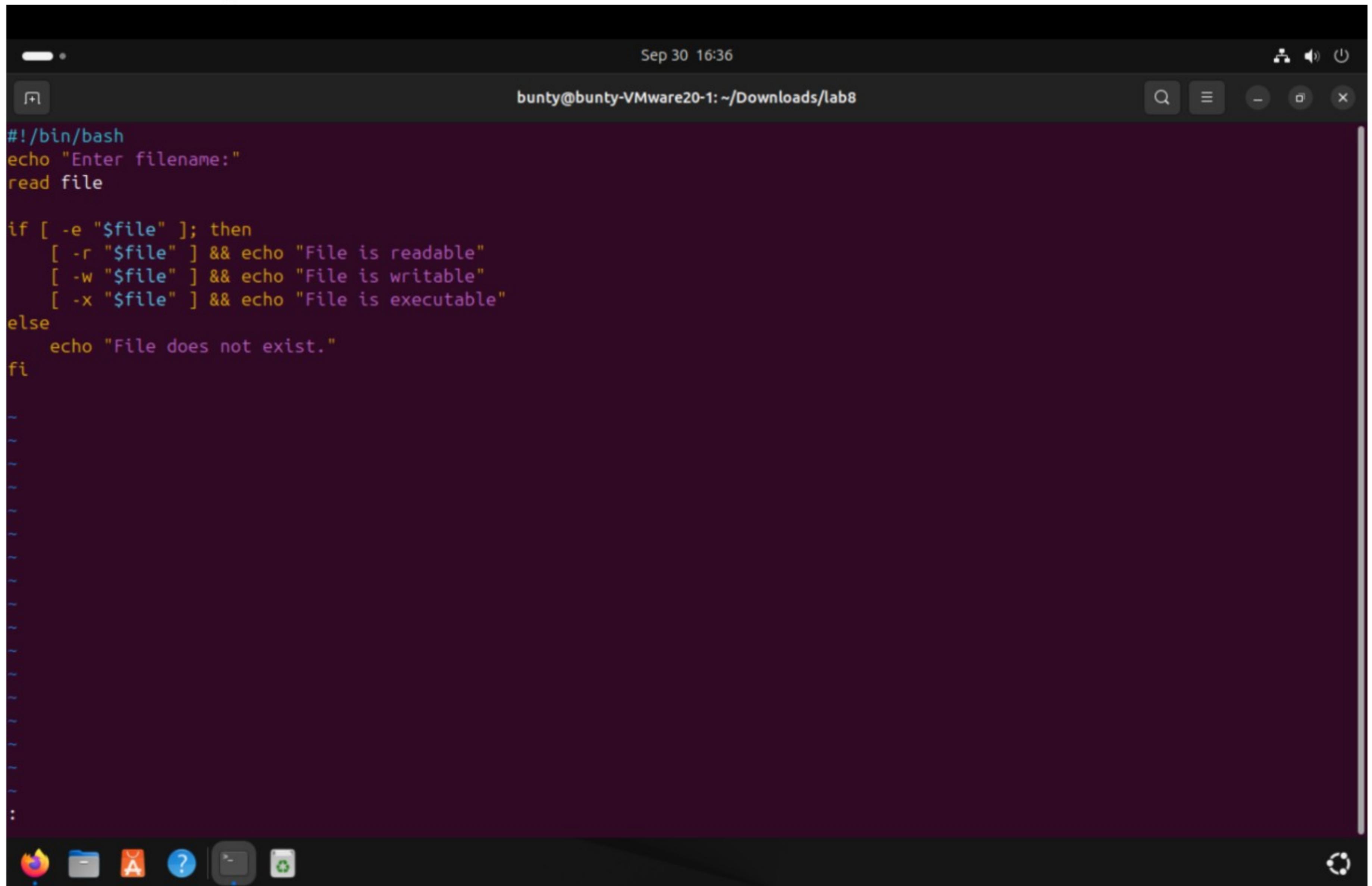
A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark purple background and a dark purple header bar. The header bar displays the date and time as "Sep 30 16:25" and the current user and location as "bunty@bunty-VMware20-1: ~/Downloads/lab7". The terminal window itself shows the command "bunty@bunty-VMware20-1:~/Downloads/lab7\$./factorial.sh" followed by three lines of output: "Factorial of 5 is: 120", "Factorial of 7 is: 5040", and "Factorial of 10 is: 3628800". Below the terminal window, the desktop's dock contains icons for a browser (Firefox), file manager, terminal, and other applications. The desktop background is a solid dark color.

EXPERIMENT 8

Shell Programming

TASKS

1. Write a script that checks the file permission of a given file and displays whether it is readable , writable , or executable by the current user.

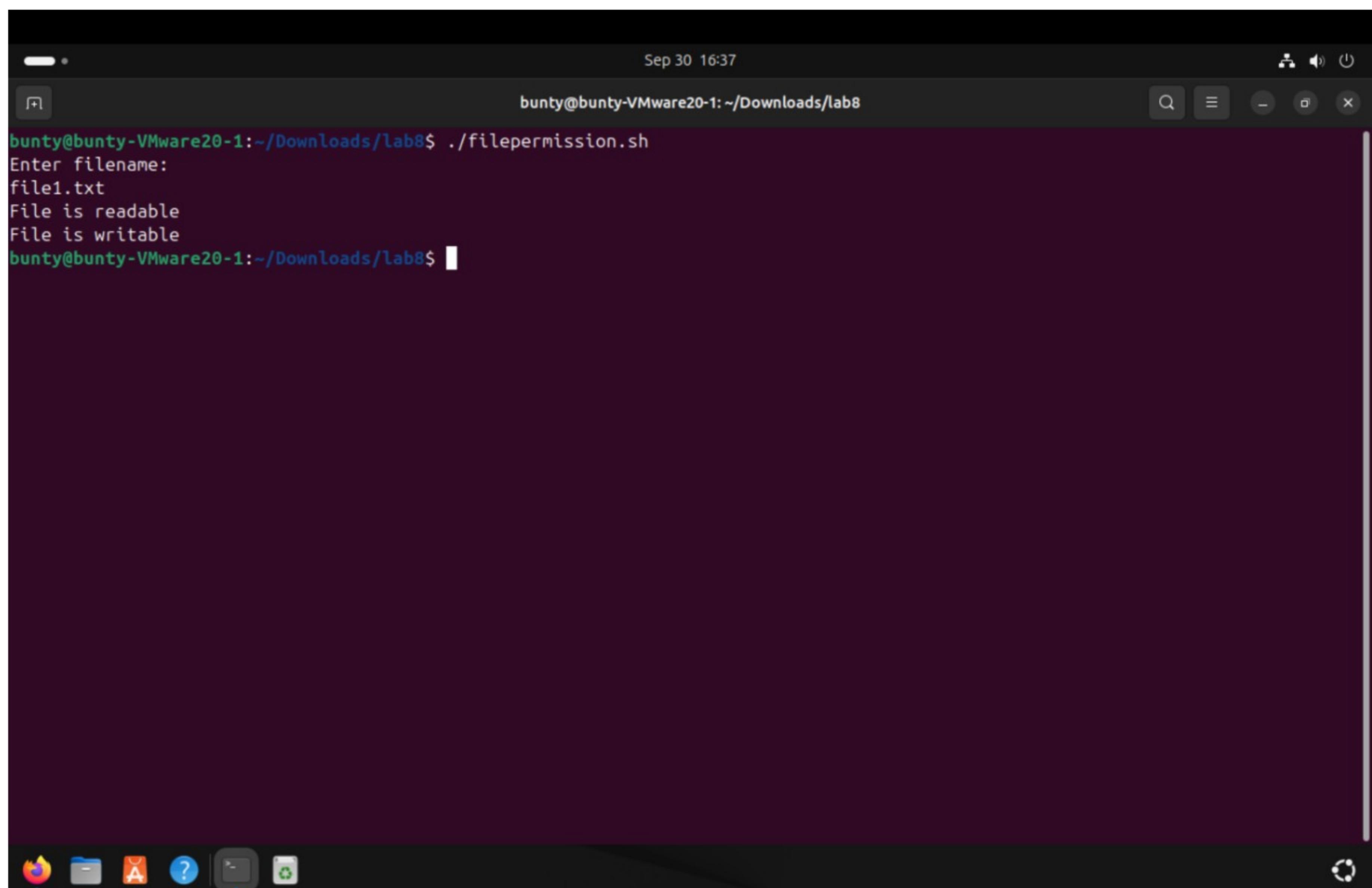


The screenshot shows a terminal window on a Linux desktop environment. The terminal title bar reads "bunty@bunty-VMware20-1: ~/Downloads/lab8". The date and time "Sep 30 16:36" are displayed at the top right. The terminal window contains a shell script written in bash:

```
#!/bin/bash
echo "Enter filename:"
read file

if [ -e "$file" ]; then
    [ -r "$file" ] && echo "File is readable"
    [ -w "$file" ] && echo "File is writable"
    [ -x "$file" ] && echo "File is executable"
else
    echo "File does not exist."
fi
```

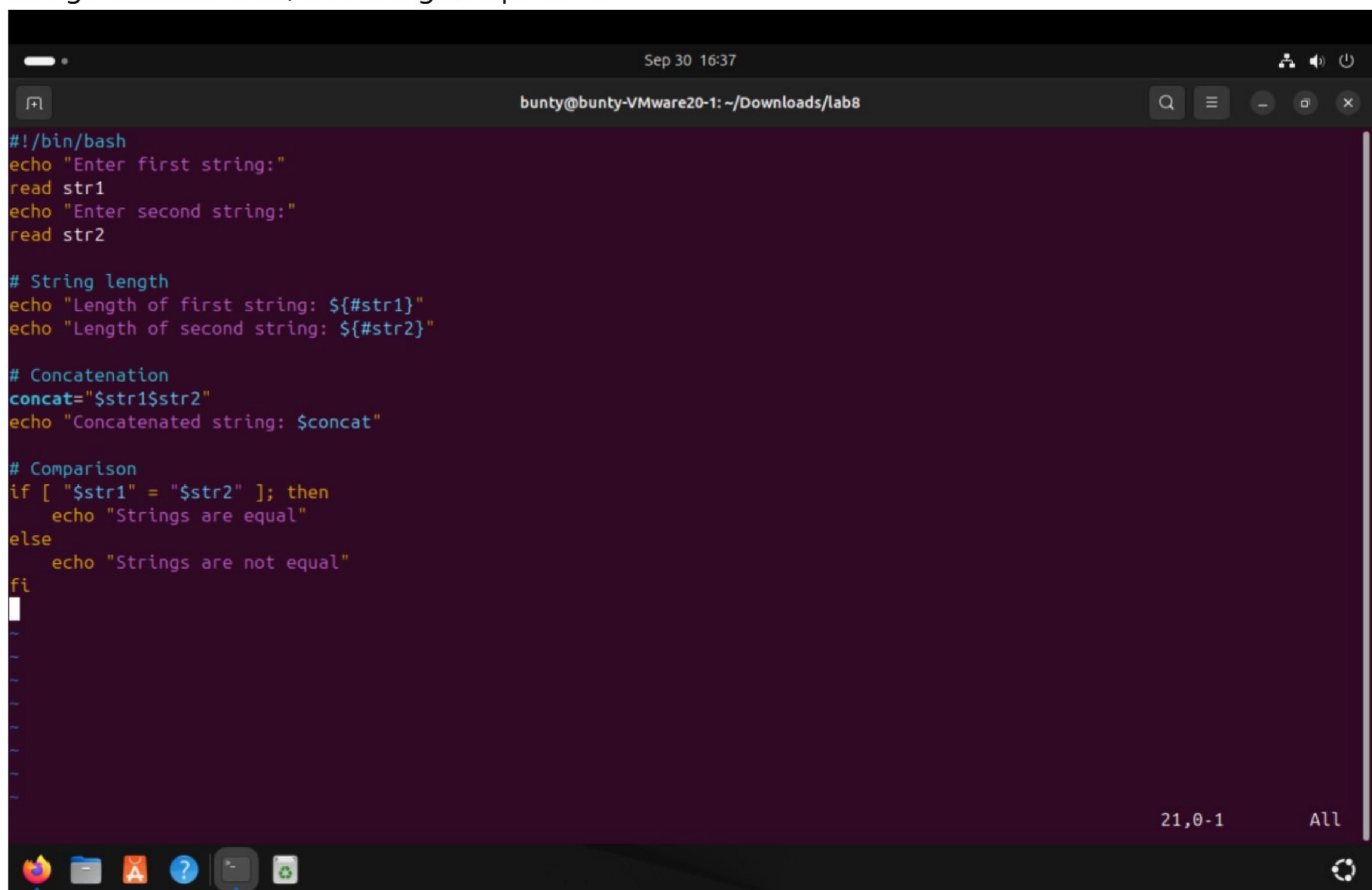
The script prompts the user to enter a filename, reads it into the variable \$file, and then checks if the file exists. If it does, it checks if it is readable, writable, and executable. If it does not exist, it prints a message indicating that. The terminal window has a dark background and light-colored text. At the bottom, there is a dock with icons for various applications, including a browser, file manager, terminal, and others.



A screenshot of a Linux desktop environment showing a terminal window. The terminal title bar says "Sep 30 16:37 bunty@bunty-VMware20-1: ~/Downloads/lab8". The terminal content shows the execution of a shell script named "filepermission.sh". The user enters "file1.txt" as the filename, and the script outputs that the file is readable and writable.

```
bunty@bunty-VMware20-1:~/Downloads/lab8$ ./filepermission.sh
Enter filename:
file1.txt
File is readable
File is writable
bunty@bunty-VMware20-1:~/Downloads/lab8$
```

2. Create a script that prompts the user to enter a string and then performs operations like string length , string concatenation, and string comparison.



A screenshot of a Linux desktop environment showing a terminal window. The terminal title bar says "Sep 30 16:37 bunty@bunty-VMware20-1: ~/Downloads/lab8". The terminal content shows the execution of a bash script that performs various string operations: it prompts for two strings, calculates their lengths, concatenates them, and compares them for equality.

```
#!/bin/bash
echo "Enter first string:"
read str1
echo "Enter second string:"
read str2

# String length
echo "Length of first string: ${#str1}"
echo "Length of second string: ${#str2}"

# Concatenation
concat="$str1$str2"
echo "Concatenated string: $concat"

# Comparison
if [ "$str1" = "$str2" ]; then
    echo "Strings are equal"
else
    echo "Strings are not equal"
fi
```

The screenshot shows a terminal window with a dark theme. The title bar indicates the date and time as "Sep 30 16:38". The command entered is "bunty@bunty-VMware20-1:~/Downloads/lab8\$./string.sh". The output of the script is displayed below:

```
bunty@bunty-VMware20-1:~/Downloads/lab8$ ./string.sh
Enter first string:
My name is Nitish Kumar
Enter second string:
and this my lab experiment 8
Length of first string: 23
Length of second string: 28
Concatenated string: My name is Nitish Kumar and this my lab experiment 8
Strings are not equal
bunty@bunty-VMware20-1:~/Downloads/lab8$
```

The terminal window has a standard Linux-style interface with icons for file operations at the bottom.

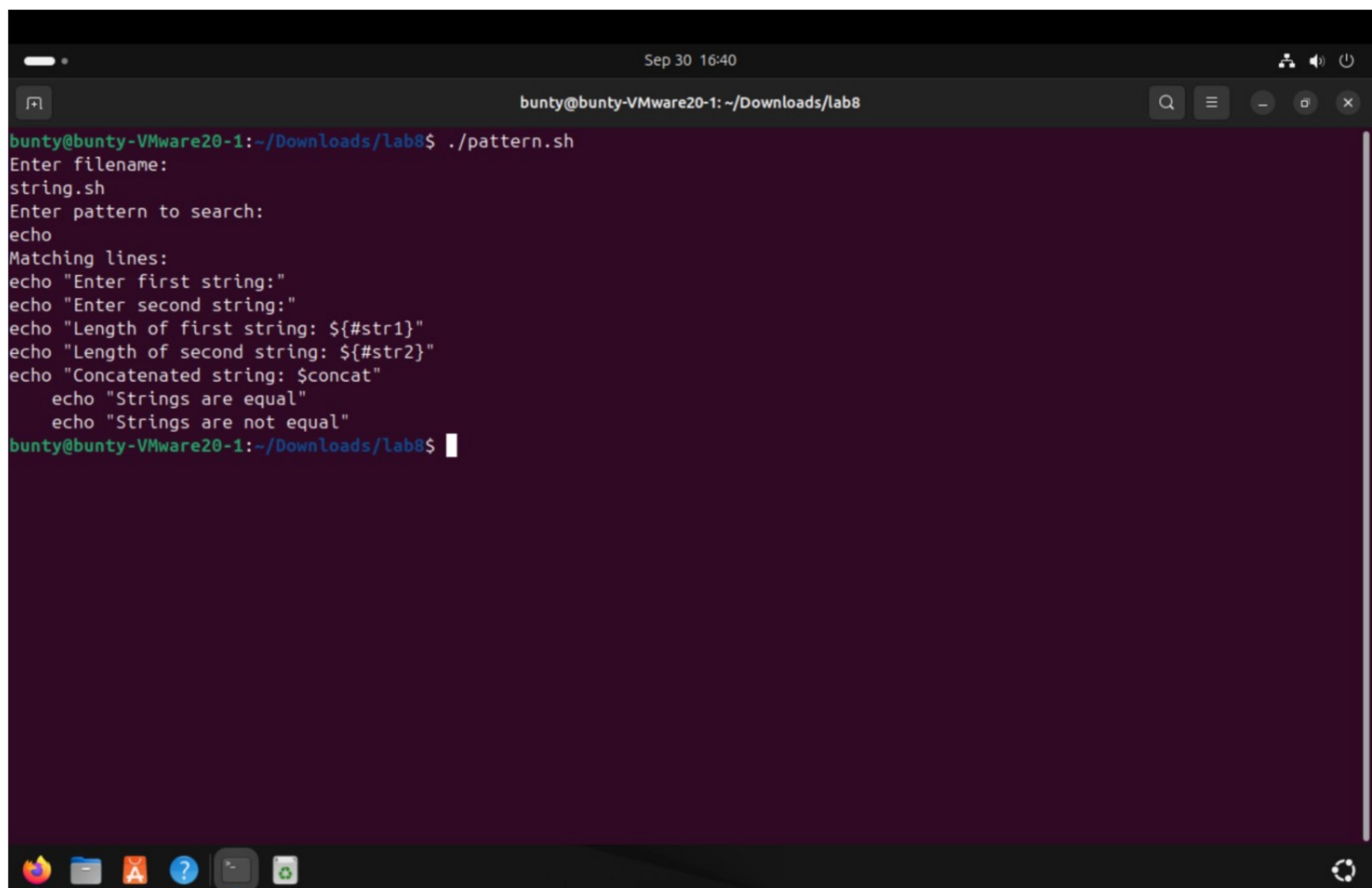
3. Write a script that searches for a specific pattern in a given file and displays the matching lines.

The screenshot shows a terminal window with a dark theme. The title bar indicates the date and time as "Sep 30 16:39". The command entered is "bunty@bunty-VMware20-1:~/Downloads/lab8\$". The script content is displayed in the terminal:

```
#!/bin/bash
echo "Enter filename:"
read file
echo "Enter pattern to search:"
read pattern

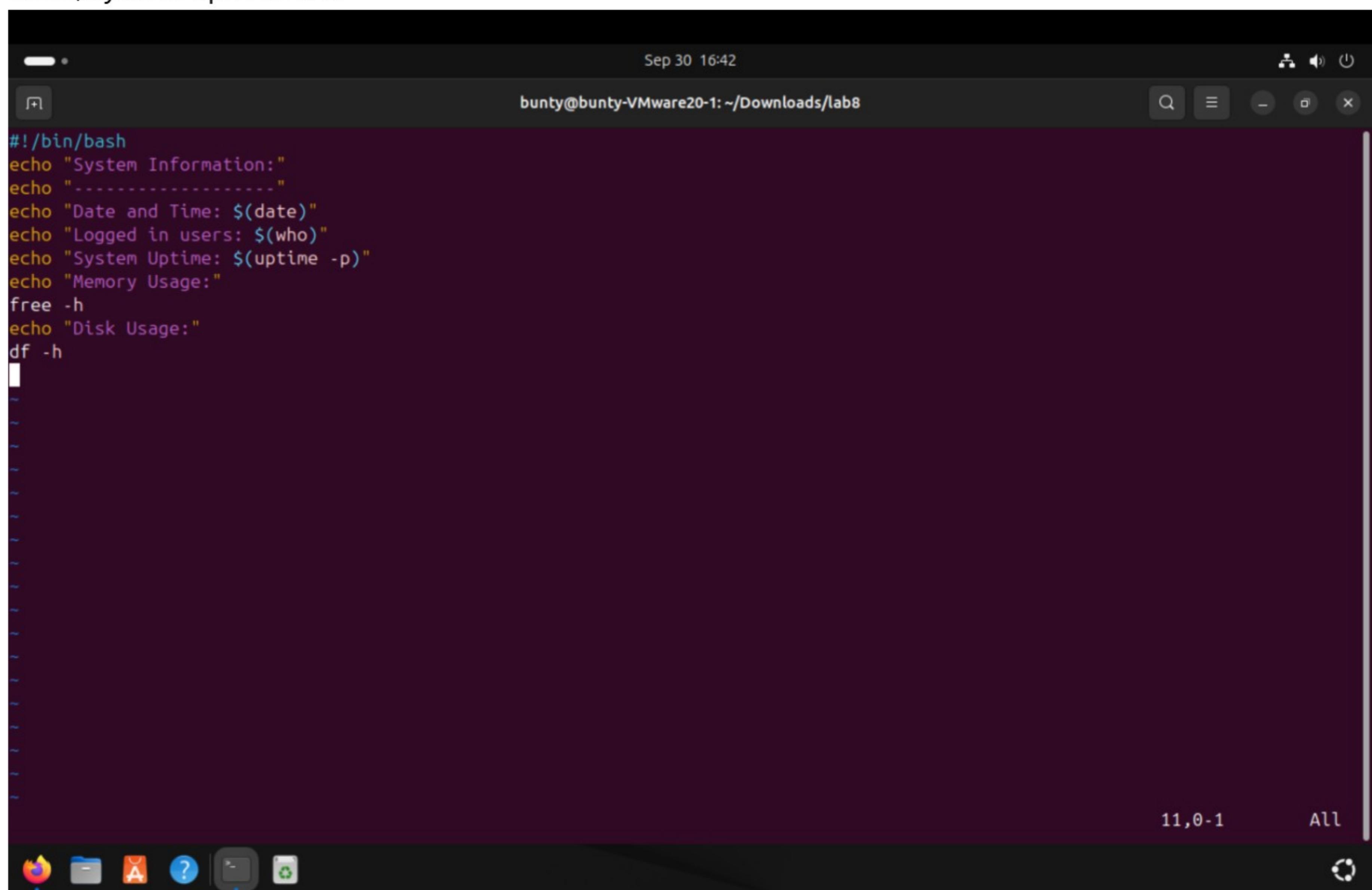
if [ -e "$file" ]; then
    echo "Matching lines:"
    grep "$pattern" "$file"
else
    echo "File not found!"
fi
```

The terminal window has a standard Linux-style interface with icons for file operations at the bottom.



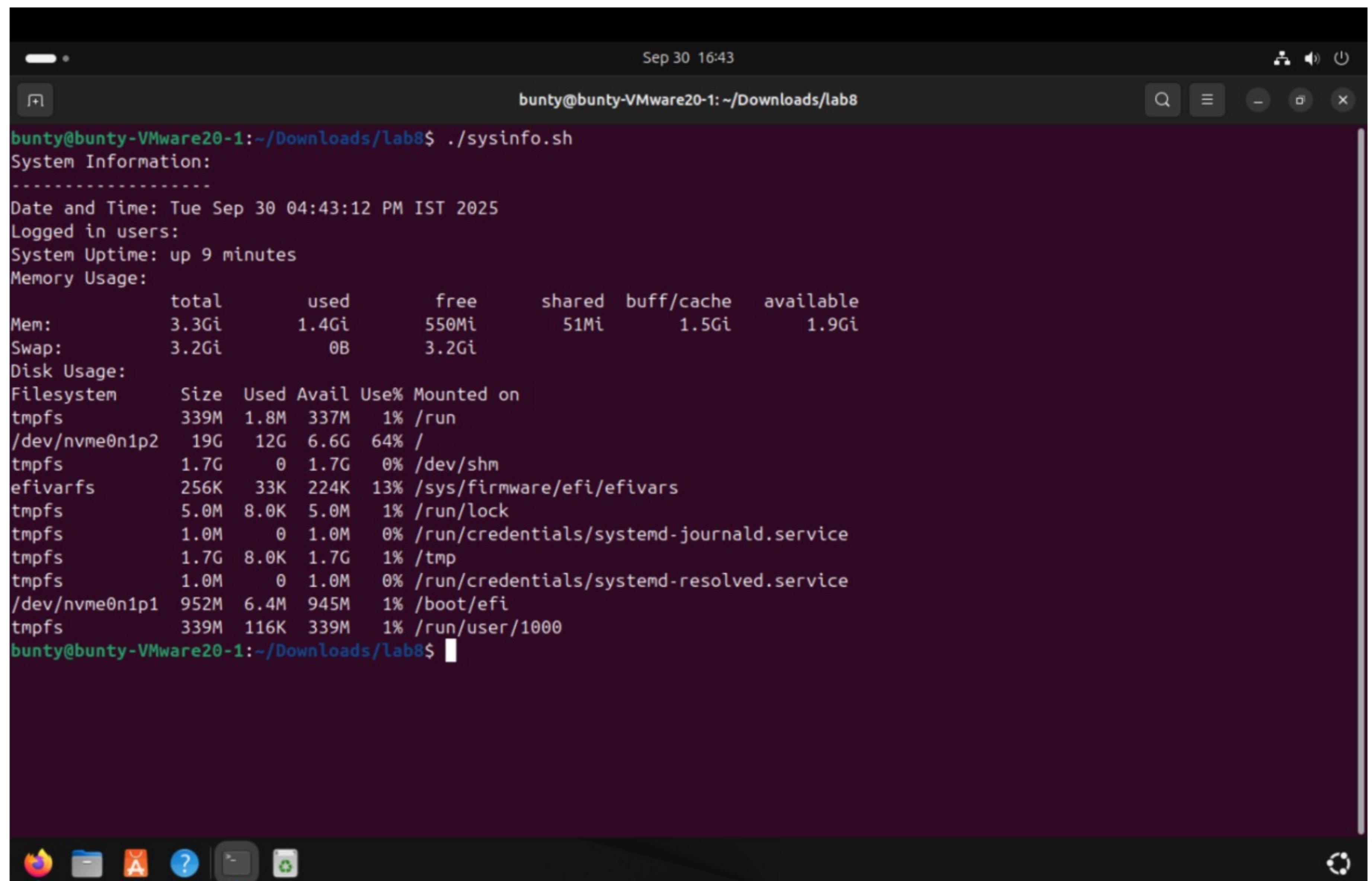
```
Sep 30 16:40
bunty@bunty-VMware20-1: ~/Downloads/lab8$ ./pattern.sh
Enter filename:
string.sh
Enter pattern to search:
echo
Matching lines:
echo "Enter first string:"
echo "Enter second string:"
echo "Length of first string: ${#str1}"
echo "Length of second string: ${#str2}"
echo "Concatenated string: $concat"
    echo "Strings are equal"
    echo "Strings are not equal"
bunty@bunty-VMware20-1: ~/Downloads/lab8$
```

4. Create a script that displays various system information like the current date and time, logged - in users, system uptime etc.



```
Sep 30 16:42
bunty@bunty-VMware20-1: ~/Downloads/lab8$ ./system_info.sh
#!/bin/bash
echo "System Information:"
echo "-----"
echo "Date and Time: $(date)"
echo "Logged in users: $(who)"
echo "System Uptime: $(uptime -p)"
echo "Memory Usage:"
free -h
echo "Disk Usage:"
df -h

```



Sep 30 16:43
bunty@bunty-VMware20-1:~/Downloads/lab8\$./sysinfo.sh
System Information:

Date and Time: Tue Sep 30 04:43:12 PM IST 2025
Logged in users:
System Uptime: up 9 minutes
Memory Usage:

	total	used	free	shared	buff/cache	available
Mem:	3.3Gi	1.4Gi	550Mi	51Mi	1.5Gi	1.9Gi
Swap:	3.2Gi	0B	3.2Gi			

Disk Usage:

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	339M	1.8M	337M	1%	/run
/dev/nvme0n1p2	19G	12G	6.6G	64%	/
tmpfs	1.7G	0	1.7G	0%	/dev/shm
efivarfs	256K	33K	224K	13%	/sys/firmware/efi/efivars
tmpfs	5.0M	8.0K	5.0M	1%	/run/lock
tmpfs	1.0M	0	1.0M	0%	/run/credentials/systemd-journald.service
tmpfs	1.7G	8.0K	1.7G	1%	/tmp
tmpfs	1.0M	0	1.0M	0%	/run/credentials/systemd-resolved.service
/dev/nvme0n1p1	952M	6.4M	945M	1%	/boot/efi
tmpfs	339M	116K	339M	1%	/run/user/1000

bunty@bunty-VMware20-1:~/Downloads/lab8\$