Usman Institute of Technology

Department of Computer Science Fall 2022

Name: Muhammad Waleed

Roll no: 20B-115-SE

Course: Operating Systems (CS312)

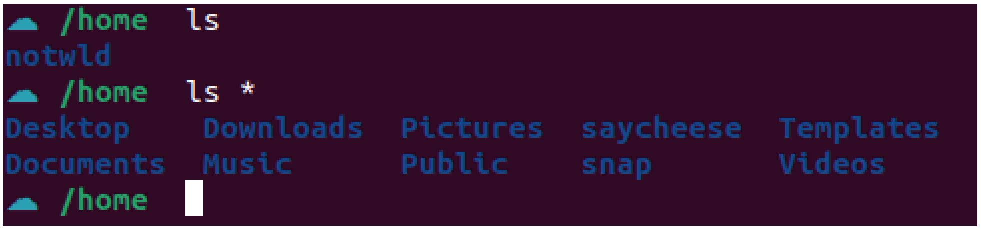
Course Instructor: Ma’am Shabina Mushtaq

Date: 13-Oct-2022

Muhammad Waleed
20B-115-SE
Operation Systems
Lab#01

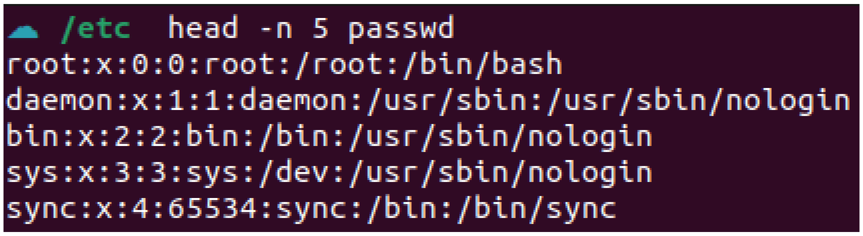
Lab Tasks:

1. Write Linux command to List all files (and subdirectories) in the home directory.



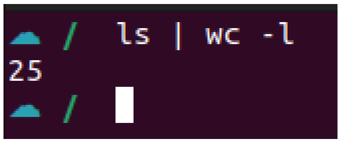
```
☁ /home  ls
notwld
☁ /home  ls *
Desktop    Downloads  Pictures  saycheese  Templates
Documents  Music      Public    snap       Videos
☁ /home  
```

2. Write Linux command to display the content of /etc/passwd file with as many lines at a time as the last digit of your roll number.



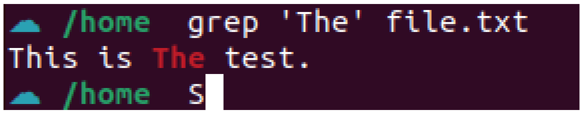
```
☁ /etc  head -n 5 passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

3. Write Linux command to count all files in the current directory.



```
☁ /  ls | wc -l
25
☁ /  
```

4. Use grep to search for the pattern. “The” in the text file in the home directory.

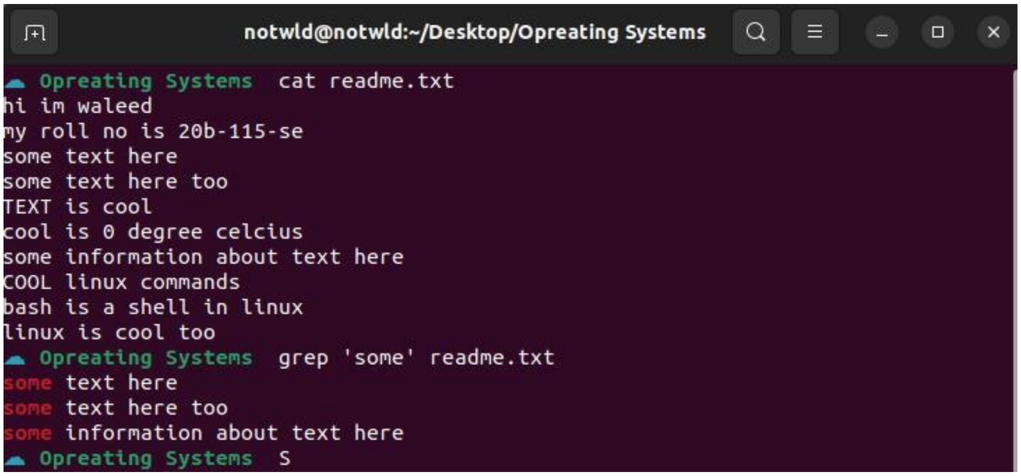


```
☁ /home  grep 'The' file.txt
This is The test.
☁ /home  S
```

Muhammad Waleed
20B-115-SE
Operation Systems
Lab#01

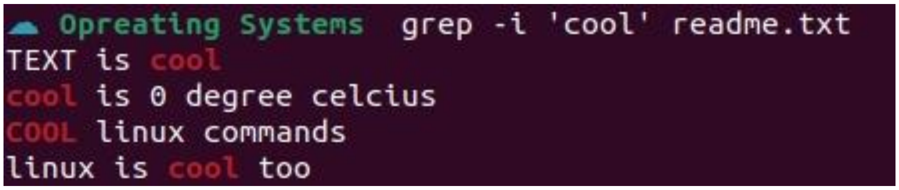
Grep Commands:

1. Search for the given string in a single file.



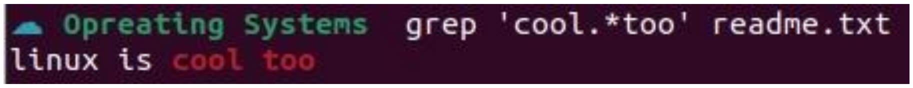
```
☁ Opreating Systems  cat readme.txt
hi im waleed
my roll no is 20b-115-se
some text here
some text here too
TEXT is cool
cool is 0 degree celcius
some information about text here
COOL linux commands
bash is a shell in linux
linux is cool too
☁ Opreating Systems  grep 'some' readme.txt
some text here
some text here too
some information about text here
☁ Opreating Systems  S
```

2. Case insensitive search using grep -I.



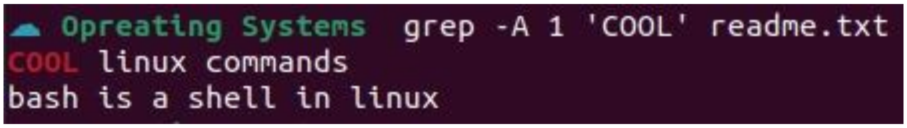
```
☁ Opreating Systems  grep -i 'cool' readme.txt
TEXT is cool
cool is 0 degree celcius
COOL linux commands
linux is cool too
```

3. Match regular expression in files.



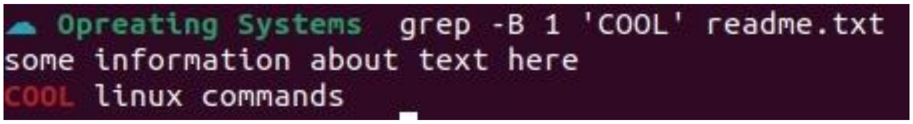
```
☁ Opreating Systems  grep 'cool.*too' readme.txt
linux is cool too
```

4. Display N lines after match.



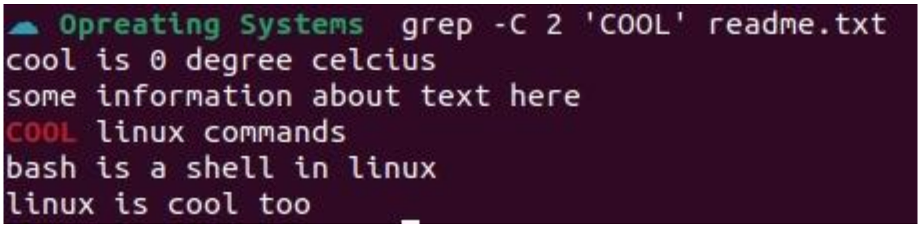
```
☁ Opreating Systems  grep -A 1 'COOL' readme.txt
COOL linux commands
bash is a shell in linux
```

5. Display N lines before match.



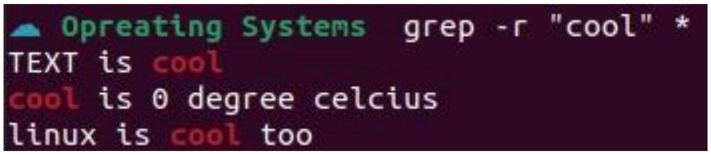
```
☁ Opreating Systems  grep -B 1 'COOL' readme.txt
some information about text here
COOL linux commands
```

6. Display N lines around match.



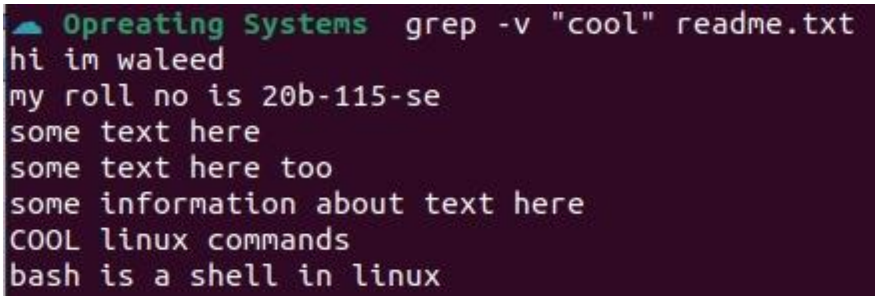
```
☁ Opreating Systems  grep -C 2 'COOL' readme.txt
cool is 0 degree celcius
some information about text here
COOL linux commands
bash is a shell in linux
linux is cool too
```

7. Searching in all files recursively using grep -r.



```
☁ Opreating Systems  grep -r "cool" *
TEXT is cool
cool is 0 degree celcius
linux is cool too
```

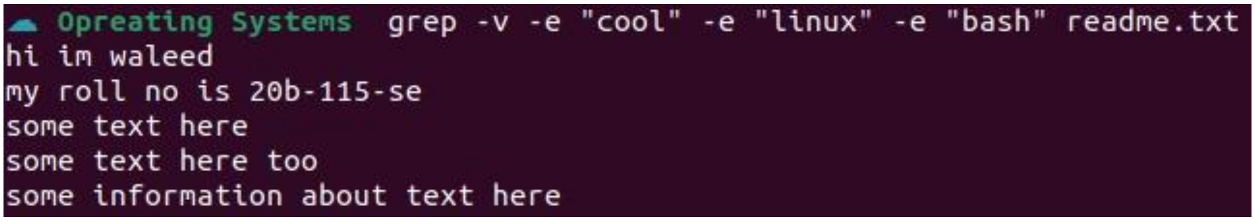
8. Invert match using grep -v.



```
☁ Opreating Systems  grep -v "cool" readme.txt
hi im waleed
my roll no is 20b-115-se
some text here
some text here too
some information about text here
COOL linux commands
bash is a shell in linux
```

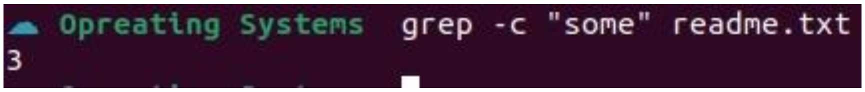
Muhammad Waleed
20B-115-SE
Operation Systems
Lab#01

9. Display the lines which does not matches all the given pattern.



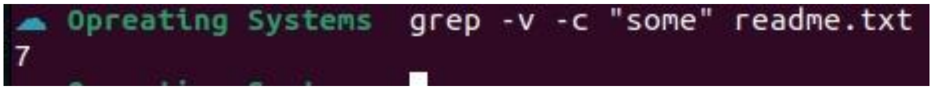
```
☁ Opreating Systems  grep -v -e "cool" -e "linux" -e "bash" readme.txt
hi im waleed
my roll no is 20b-115-se
some text here
some text here too
some information about text here
```

10. Counting the number of matches using grep -c.



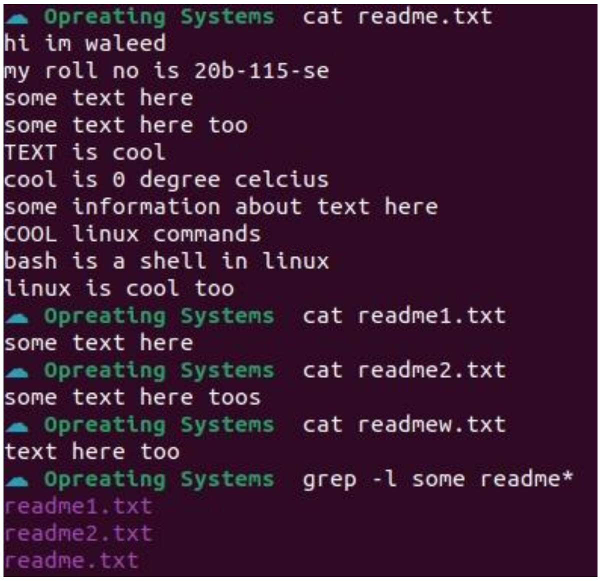
```
☁ Opreating Systems  grep -c "some" readme.txt
3
```

11. When you want do find out how many lines that does not match the pattern.



```
☁ Opreating Systems  grep -v -c "some" readme.txt
7
```

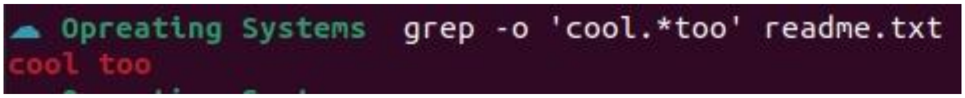
12. Display only the file names which matches the given pattern using grep -l.



```
☁ Opreating Systems  cat readme.txt
hi im waleed
my roll no is 20b-115-se
some text here
some text here too
TEXT is cool
cool is 0 degree celcius
some information about text here
COOL linux commands
bash is a shell in linux
linux is cool too
☁ Opreating Systems  cat readme1.txt
some text here
☁ Opreating Systems  cat readme2.txt
some text here toos
☁ Opreating Systems  cat readmew.txt
text here too
☁ Opreating Systems  grep -l some readme*
readme1.txt
readme2.txt
readme.txt
```

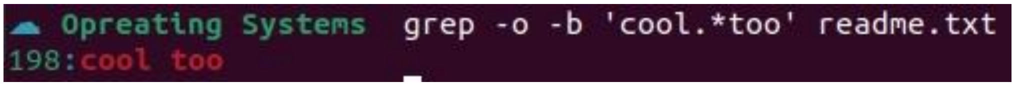
Muhammad Waleed
20B-115-SE
Operation Systems
Lab#01

13. Show only the matched string.



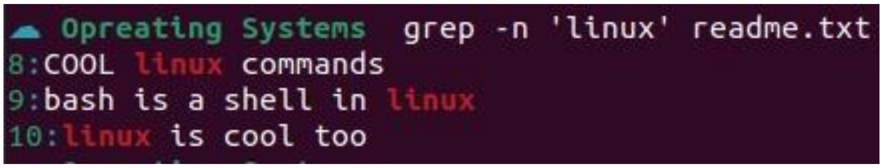
```
☁ Opreating Systems  grep -o 'cool.*too' readme.txt
cool too
```

14. Show the position of match in the line.



```
☁ Opreating Systems  grep -o -b 'cool.*too' readme.txt
198:cool too
```

15. Show line number while displaying the output using grep -n.



```
☁ Opreating Systems  grep -n 'linux' readme.txt
8:COOL linux commands
9:bash is a shell in linux
10:linux is cool too
```

Usman Institute of Technology

Department of Computer Science Fall 2022

Name: Muhammad Waleed

Roll no: 20B-115-SE

Course: Operating Systems (CS312)

Course Instructor: Ma’am Shabina Mushtaq

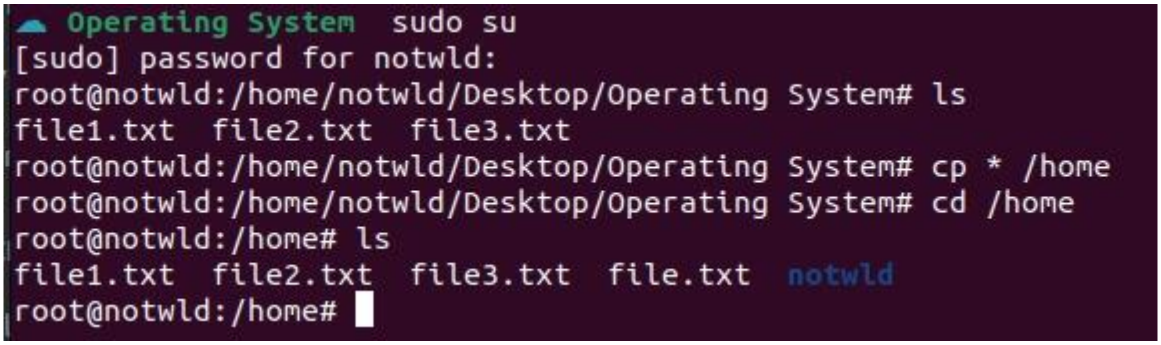
Date: 20-Oct-2022

Muhammad Waleed
20B-115-SE
Operating Systems
Lab#02

Lab Exercises:

1. Write a command to copy all files of current directory to /home?

   Command: cp * /home



```
☁ Operating System  sudo su
[sudo] password for notwld:
root@notwld:/home/notwld/Desktop/Operating System# ls
file1.txt  file2.txt  file3.txt
root@notwld:/home/notwld/Desktop/Operating System# cp * /home
root@notwld:/home/notwld/Desktop/Operating System# cd /home
root@notwld:/home# ls
file1.txt  file2.txt  file3.txt  file.txt  notwld
root@notwld:/home# 
```

2. What is the difference between the permissions 777 and 775 of the chmod command?

   Chmod is a command to grant permission for either write, edit, or execute a file or all to a user, group, and other user.

   Syntax:

   chmod 777 [file_name]

   OR

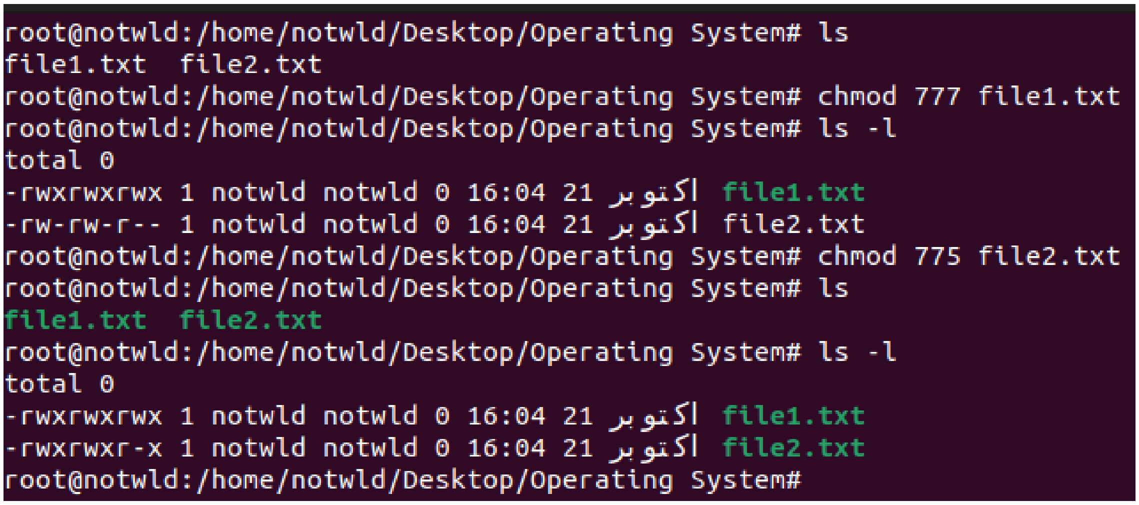
   chmod ugo+rwx [file_name]

   For different permissions can be represented in numeric form

   Read = 4, **Write** = 2, **Execute** = 1

To give all permissions to all users (root, group and other) we write 777 (read + write + execute) with chmod which means that numeric values for read, write, and execute are added to represent all permissions and it is written 3 time for different users i.e root, group and other.

To give specific permissions to different users like root can write, read and execute the file and same for group user but other user can only read and execute the file then we write 775 with chmod it means that numeric values are added for root user that is 7 to give all permissions and same with group user and for other user only read's and execute's numeric values are added to only give them r+w permission.

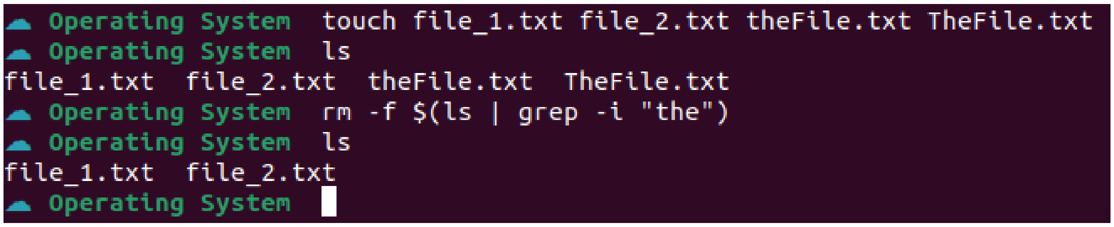


```
root@notwld:/home/notwld/Desktop/Operating System# ls
file1.txt  file2.txt
root@notwld:/home/notwld/Desktop/Operating System# chmod 777 file1.txt
root@notwld:/home/notwld/Desktop/Operating System# ls -l
total 0
-rwxrwxrwx 1 notwld notwld 0 16:04 21 اکتوبر  file1.txt
-rw-rw-r-- 1 notwld notwld 0 16:04 21 اکتوبر  file2.txt
root@notwld:/home/notwld/Desktop/Operating System# chmod 775 file2.txt
root@notwld:/home/notwld/Desktop/Operating System# ls
file1.txt  file2.txt
root@notwld:/home/notwld/Desktop/Operating System# ls -l
total 0
-rwxrwxrwx 1 notwld notwld 0 16:04 21 اکتوبر  file1.txt
-rwxrwxr-x 1 notwld notwld 0 16:04 21 اکتوبر  file2.txt
root@notwld:/home/notwld/Desktop/Operating System#
```

Muhammad Waleed
20B-115-SE
Operating Systems
Lab#02

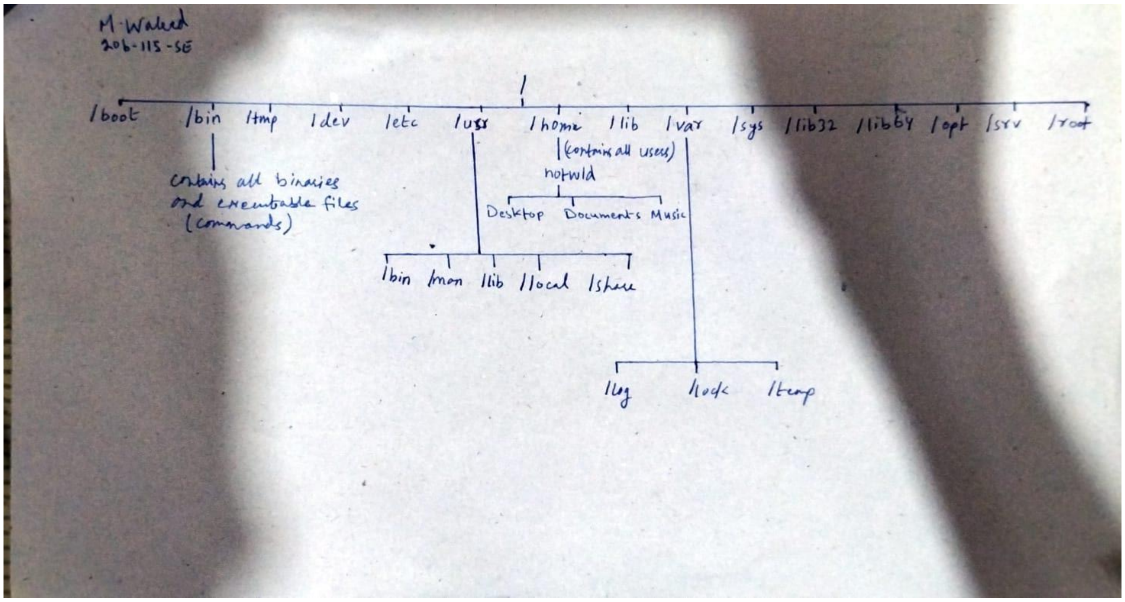
3. Write a command to remove all files with name containing text ‘the’?

Command: rm -f $(ls | grep -I “the”)



```
☁ Operating System  touch file_1.txt file_2.txt theFile.txt TheFile.txt
☁ Operating System  ls
file_1.txt  file_2.txt  theFile.txt  TheFile.txt
☁ Operating System  rm -f $(ls | grep -i "the")
☁ Operating System  ls
file_1.txt  file_2.txt
☁ Operating System  
```

4. Draw Linux Directory Structure (Tree Like structure).



Usman Institute of Technology

Department of Computer Science Fall 2022

Name: Muhammad Waleed

Roll no: 20B-115-SE

Course: Operating Systems (CS312)

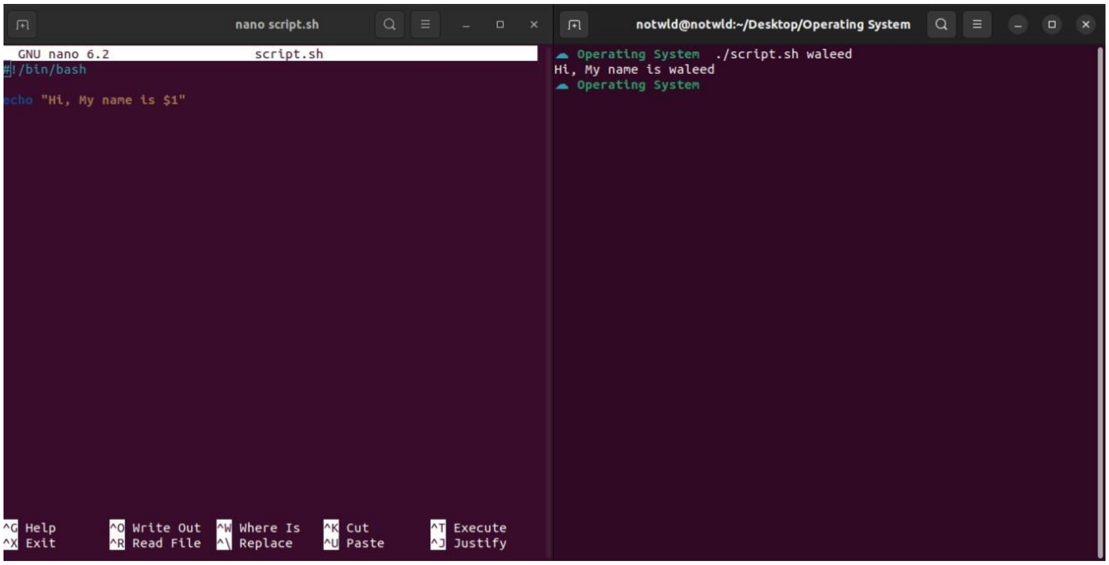
Course Instructor: Ma'am Shabina Mushtaq

Date: 27-Oct-2022

Muhammad Waleed
20B-115-SE
Operating Systems
Lab#03

Lab Tasks:

1. Write a shell program that takes one parameter (your name) and displays it on the screen.

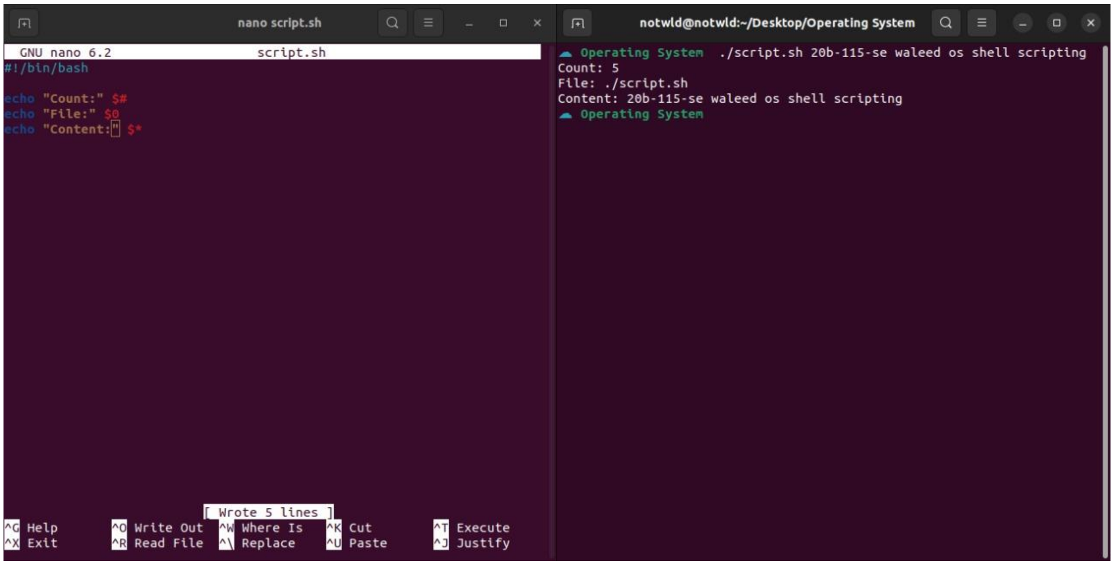


```
  GNU nano 6.2                    script.sh
#!/bin/bash

echo "Hi, My name is $1"
```

```
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

```
☁ Operating System  ./script.sh waleed
Hi, My name is waleed
☁ Operating System
```

2. Write a shell program that takes a number parameters equal to the last digit of your roll number and displays the values of the built-in variables such as $#, $0, and $* on the screen.



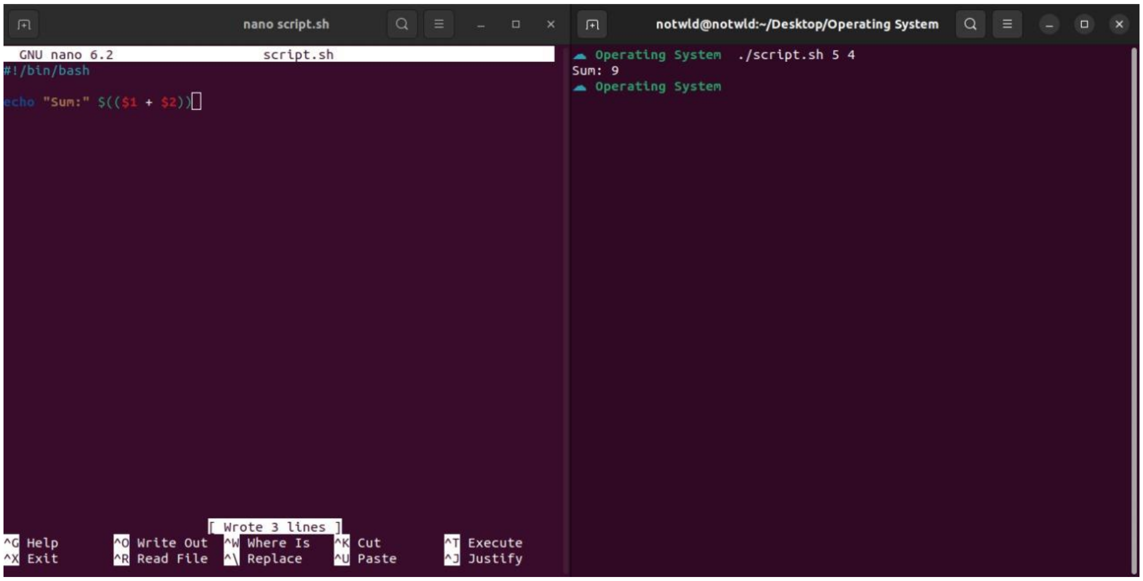
```
  GNU nano 6.2                    script.sh
#!/bin/bash

echo "Count:" $#
echo "File:" $0
echo "Content:" $*
```

```
                          [ Wrote 5 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

```
☁ Operating System  ./script.sh 20b-115-se waleed os shell scripting
Count: 5
File: ./script.sh
Content: 20b-115-se waleed os shell scripting
☁ Operating System
```

Muhammad Waleed
20B-115-SE
Operating Systems
Lab#03

3. Write a Shell script to perform addition on numbers provided by command line parameters.



```
  GNU nano 6.2                     script.sh
#!/bin/bash

echo "Sum:" $(($1 + $2))

                         [ Wrote 3 lines ]
```

```
 Operating System  ./script.sh 5 4
Sum: 9
 Operating System
```

Usman Institute of Technology

Department of Computer Science Fall 2022

Name: Muhammad Waleed

Roll no: 20B-115-SE

Course: Operating Systems (CS312)

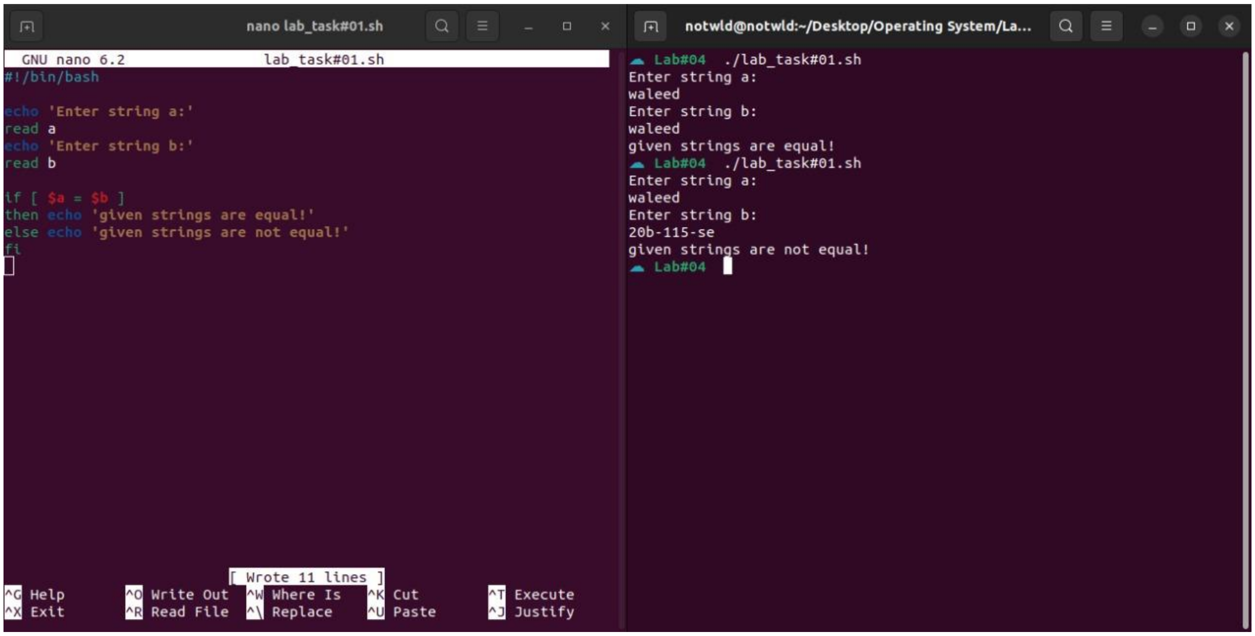
Course Instructor: Ma’am Shabina Mushtaq

Date: 3-Nov-2022

Muhammad Waleed
20b-115-se
Operating Systems
Lab#04

Lab Tasks:

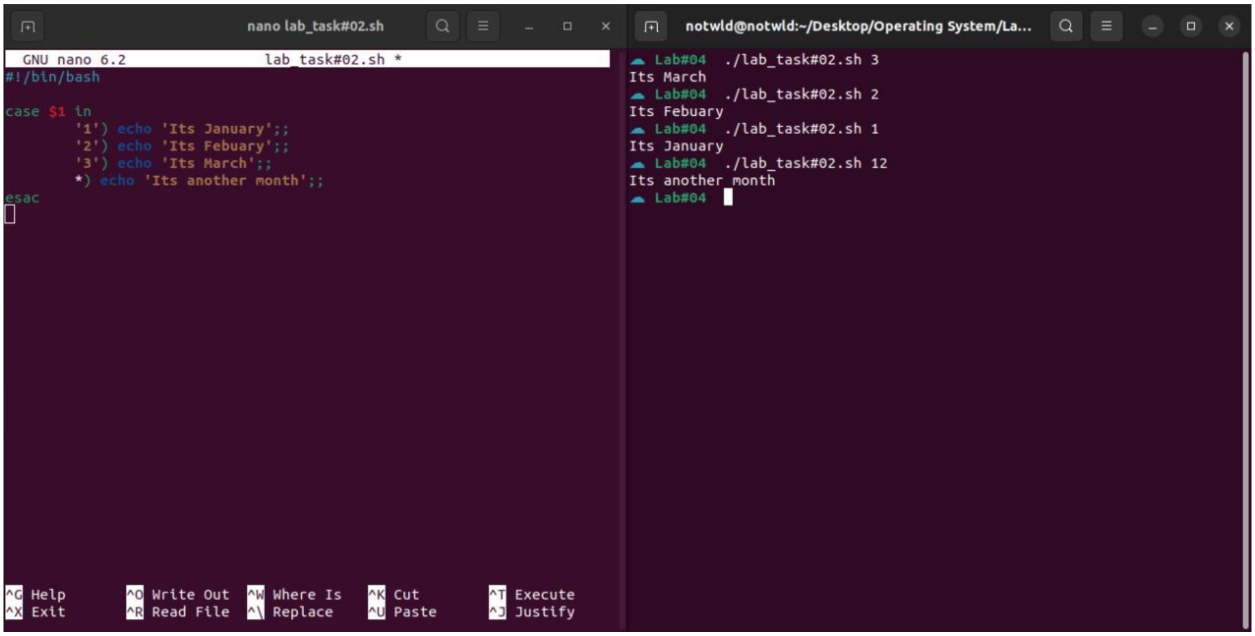
1. Write a script that takes two strings as input compares them and depending upon the results of the comparison prints the results.
   The user may provide input to the Bash script using read var



```
#!/bin/bash

echo 'Enter string a:'
read a
echo 'Enter string b:'
read b

if [ $a = $b ]
then echo 'given strings are equal!'
else echo 'given strings are not equal!'
fi
```

```
Lab#04  ./lab_task#01.sh
Enter string a:
waleed
Enter string b:
waleed
given strings are equal!
Lab#04  ./lab_task#01.sh
Enter string a:
waleed
Enter string b:
20b-115-se
given strings are not equal!
Lab#04
```

2. Write a script that takes a number (parameter) from 1-3 as input and uses case to display the name of corresponding month.

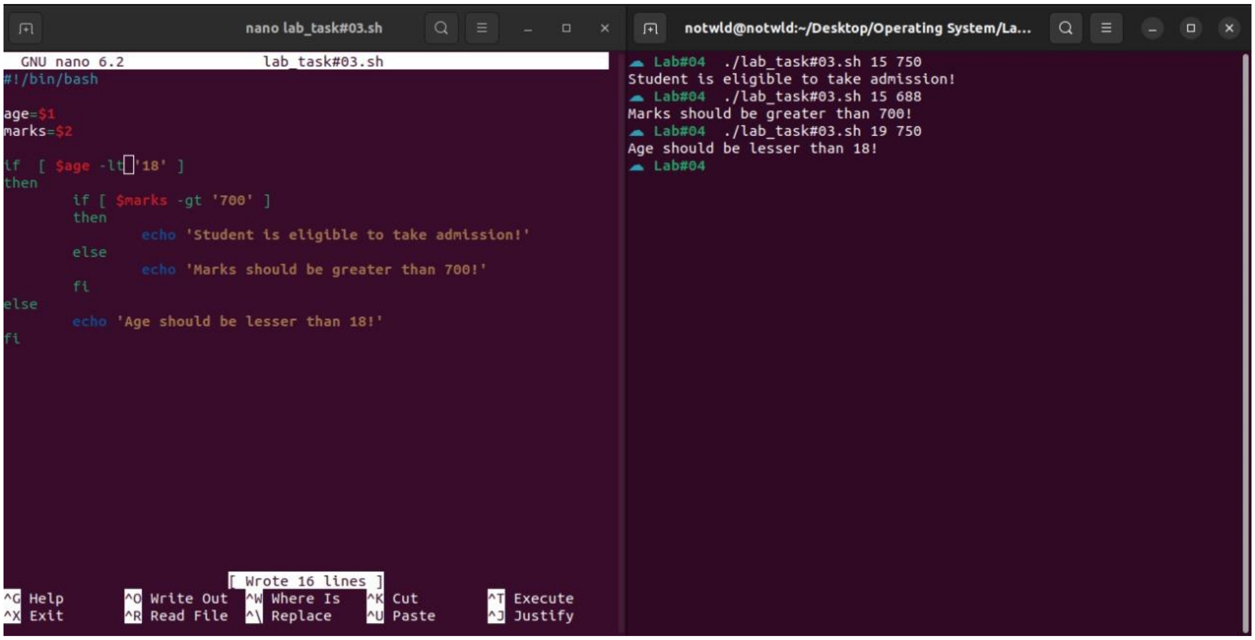


```
#!/bin/bash

case $1 in
        '1') echo 'Its January';;
        '2') echo 'Its Febuary';;
        '3') echo 'Its March';;
        *) echo 'Its another month';;
esac
```

```
Lab#04  ./lab_task#02.sh 3
Its March
Lab#04  ./lab_task#02.sh 2
Its Febuary
Lab#04  ./lab_task#02.sh 1
Its January
Lab#04  ./lab_task#02.sh 12
Its another month
Lab#04
```

Muhammad Waleed
20b-115-se
Operating Systems
Lab#04

3. Write a script that takes command-line argument for age and marks and decide whether student is eligible for admission or not. Eligibility Criteria: Age should be lesser than 18 and marks should be greater than 700



```
#!/bin/bash

age=$1
marks=$2

if  [ $age -lt '18' ]
then
        if [ $marks -gt '700' ]
        then
                echo 'Student is eligible to take admission!'
        else
                echo 'Marks should be greater than 700!'
        fi
else
        echo 'Age should be lesser than 18!'
fi
```

```
☁ Lab#04  ./lab_task#03.sh 15 750
Student is eligible to take admission!
☁ Lab#04  ./lab_task#03.sh 15 688
Marks should be greater than 700!
☁ Lab#04  ./lab_task#03.sh 19 750
Age should be lesser than 18!
☁ Lab#04
```

Usman Institute of Technology

Department of Computer Science Fall 2022

Name: Muhammad Waleed

Roll no: 20B-115-SE

Course: Operating Systems (CS312)

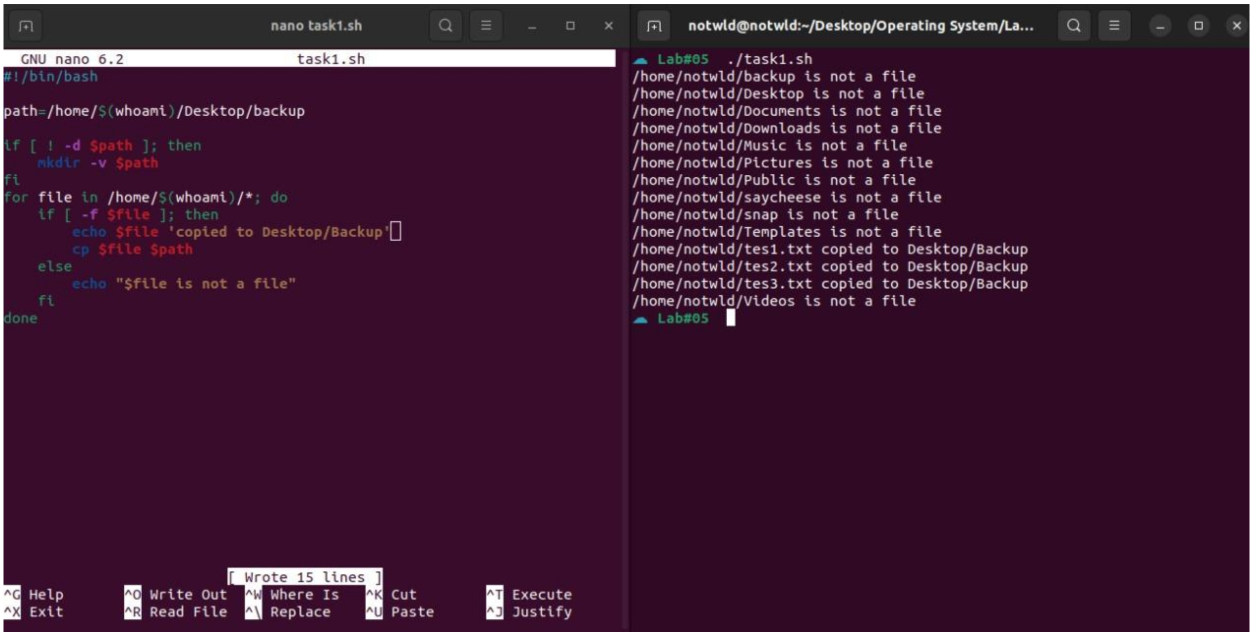
Course Instructor: Ma’am Shabina Mushtaq

Date: 10-Nov-2022

Muhammad Waleed
20b-115-se
Operating Systems
Lab#05

Lab Tasks:

1. Write a script that creates a backup version of each file in your home directory to a subdirectory called backup using for statement. If the operation fails an error message is to be displayed.

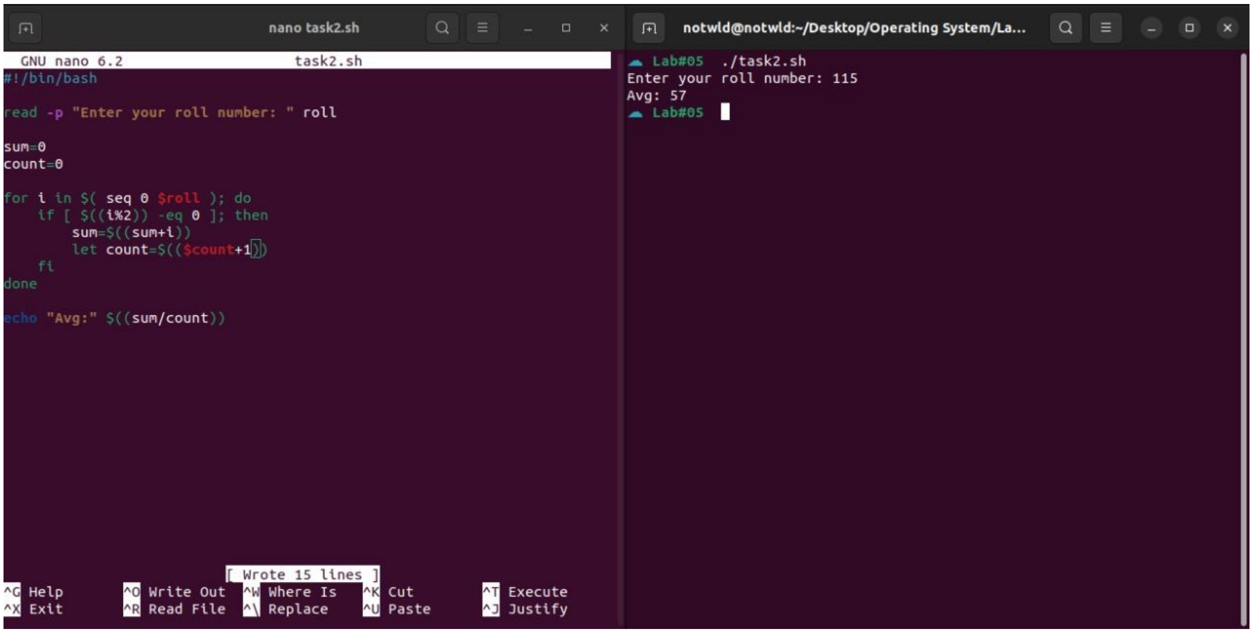


```
#!/bin/bash

path=/home/$(whoami)/Desktop/backup

if [ ! -d $path ]; then
    mkdir -v $path
fi
for file in /home/$(whoami)/*; do
    if [ -f $file ]; then
        echo $file 'copied to Desktop/Backup'
        cp $file $path
    else
        echo "$file is not a file"
    fi
done
```

```
[ Wrote 15 lines ]
```

```
Lab#05  ./task1.sh
/home/notwld/backup is not a file
/home/notwld/Desktop is not a file
/home/notwld/Documents is not a file
/home/notwld/Downloads is not a file
/home/notwld/Music is not a file
/home/notwld/Pictures is not a file
/home/notwld/Public is not a file
/home/notwld/saycheese is not a file
/home/notwld/snap is not a file
/home/notwld/Templates is not a file
/home/notwld/tes1.txt copied to Desktop/Backup
/home/notwld/tes2.txt copied to Desktop/Backup
/home/notwld/tes3.txt copied to Desktop/Backup
/home/notwld/Videos is not a file
Lab#05
```

2. Write a script that calculates the average of all even numbers less than or equal to your roll number and prints the result.



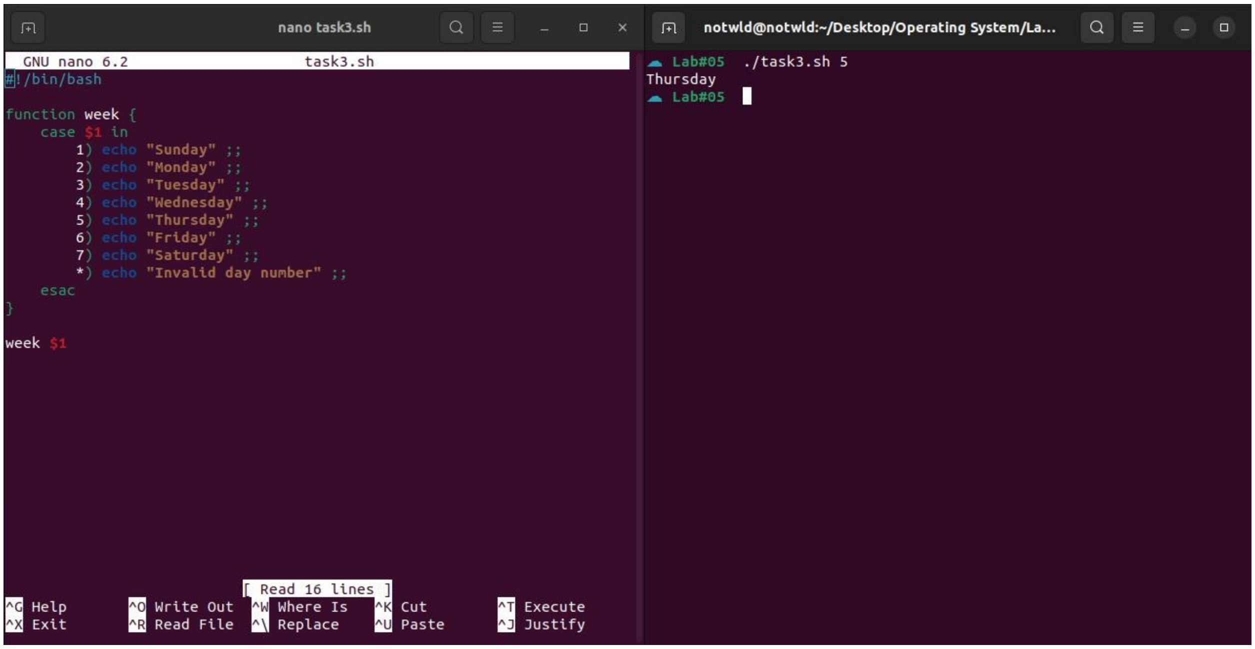
```
#!/bin/bash

read -p "Enter your roll number: " roll

sum=0
count=0

for i in $( seq 0 $roll ); do
    if [ $((i%2)) -eq 0 ]; then
        sum=$((sum+i))
        let count=$(($count+1))
    fi
done

echo "Avg:" $((sum/count))
```

```
[ Wrote 15 lines ]
```

```
Lab#05  ./task2.sh
Enter your roll number: 115
Avg: 57
Lab#05
```

Muhammad Waleed
20b-115-se
Operating Systems
Lab#05

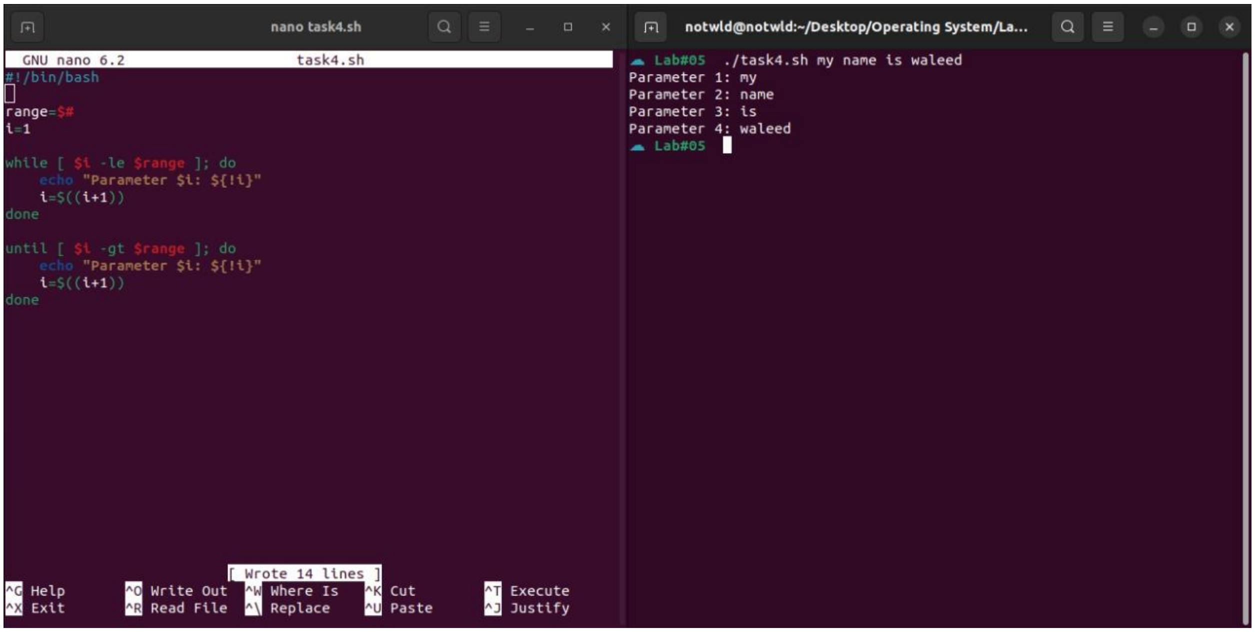
3. Write a function that displays the name of the week days starting from Sunday if the user passes a day number. If a number provided is not between 1 and 7 an error message is displayed.



```
  GNU nano 6.2                     task3.sh
#!/bin/bash

function week {
    case $1 in
        1) echo "Sunday" ;;
        2) echo "Monday" ;;
        3) echo "Tuesday" ;;
        4) echo "Wednesday" ;;
        5) echo "Thursday" ;;
        6) echo "Friday" ;;
        7) echo "Saturday" ;;
        *) echo "Invalid day number" ;;
    esac
}

week $1

                           [ Read 16 lines ]
```

```
☁ Lab#05  ./task3.sh 5
Thursday
☁ Lab#05
```

4. Write scripts that displays the parameters passed along with the parameter number using while and until statements.



```
  GNU nano 6.2                     task4.sh
#!/bin/bash

range=$#
i=1

while [ $i -le $range ]; do
    echo "Parameter $i: ${!i}"
    i=$((i+1))
done

until [ $i -gt $range ]; do
    echo "Parameter $i: ${!i}"
    i=$((i+1))
done

                          [ Wrote 14 lines ]
```

```
☁ Lab#05  ./task4.sh my name is waleed
Parameter 1: my
Parameter 2: name
Parameter 3: is
Parameter 4: waleed
☁ Lab#05
```