



# Operating Systems

Lab Report 2  
(4,5,6)

**SUBMITTED BY:**  
**SHAHZANEER AHMED**

**REGISTRATION**  
**NUMBER:**  
**SP21-BCS-087**

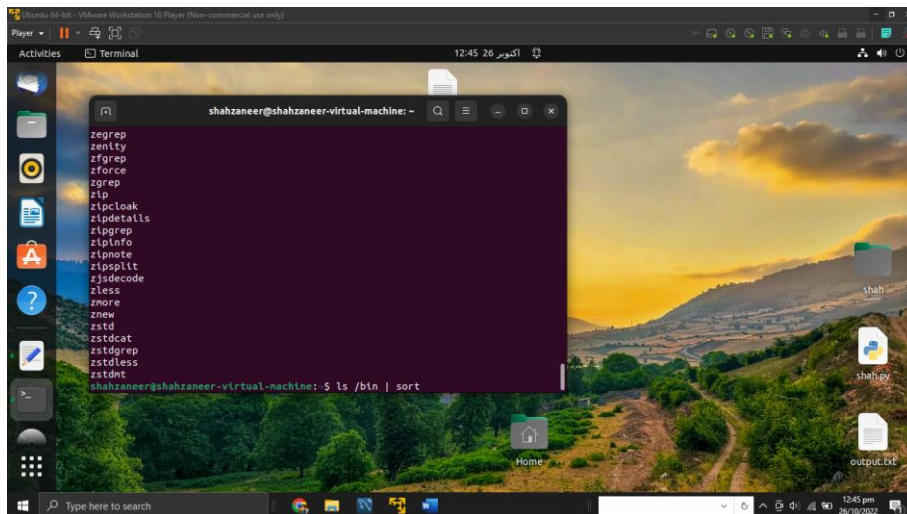
**SUBMITTED TO:**  
**Dr. Rubina Adnan**

**DATE OF**  
**SUBMISSION:**  
**October 26, 2022**

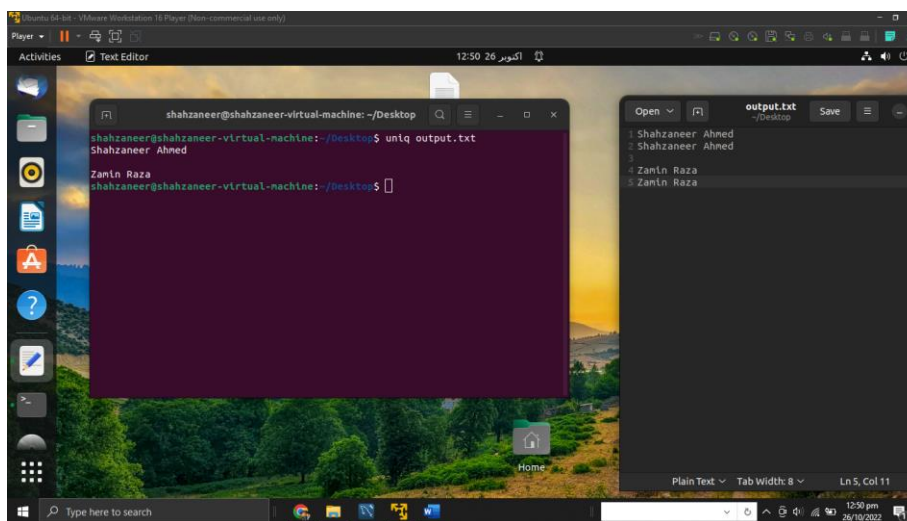
# Lab 4

## // Task 1

Use sort and unique command to sort a file and print unique values.



```
shahzaneer@shahzaneer-virtual-machine: ~  
zegrep  
zenity  
zfgrep  
zforce  
zgrep  
zip  
zipcloak  
zipdetails  
zipgrep  
zipinfo  
zipnote  
zipsplit  
zjsdecode  
zless  
znore  
znw  
zstd  
zstdcat  
zstdgrep  
zstdless  
zstdmt  
shahzaneer@shahzaneer-virtual-machine: $ ls /bin | sort
```



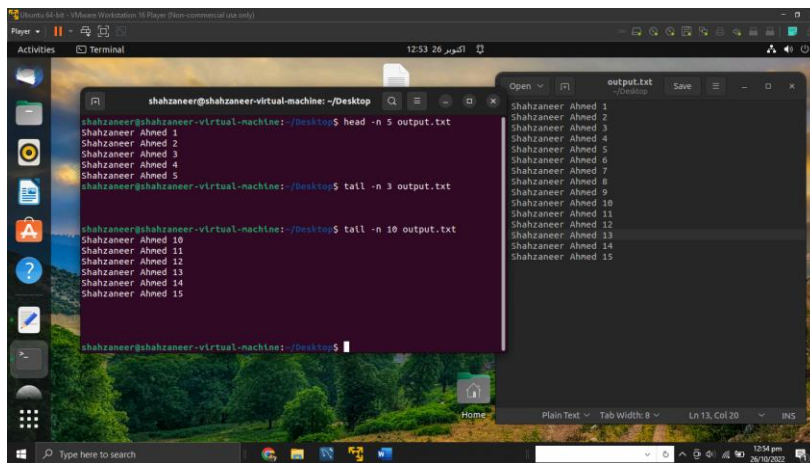
```
shahzaneer@shahzaneer-virtual-machine: ~/Desktop  
shahzaneer@shahzaneer-virtual-machine: ~/Desktop$ uniq output.txt  
Shahzaneer Ahmed  
Zamin Raza  
shahzaneer@shahzaneer-virtual-machine: ~/Desktop$
```

output.txt

```
Shahzaneer Ahmed  
Shahzaneer Ahmed  
Zamin Raza  
Zamin Raza
```

## // Task 2

Use head and tail to print lines in a particular range in a file.



A screenshot of a Linux terminal window titled "shahzaneer@shahzaneer-virtual-machine: ~/Desktop". The terminal shows the following commands and output:

```
shahzaneer@shahzaneer-virtual-machine:~/Desktop$ head -n 5 output.txt
Shahzaneer Ahmed 1
Shahzaneer Ahmed 2
Shahzaneer Ahmed 3
Shahzaneer Ahmed 4
Shahzaneer Ahmed 5

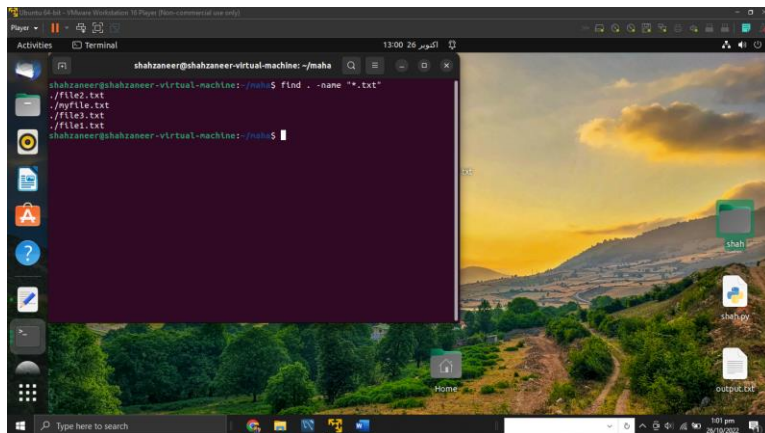
shahzaneer@shahzaneer-virtual-machine:~/Desktop$ tail -n 3 output.txt
Shahzaneer Ahmed 12
Shahzaneer Ahmed 13
Shahzaneer Ahmed 14

shahzaneer@shahzaneer-virtual-machine:~/Desktop$ tail -n 10 output.txt
Shahzaneer Ahmed 10
Shahzaneer Ahmed 11
Shahzaneer Ahmed 12
Shahzaneer Ahmed 13
Shahzaneer Ahmed 14
Shahzaneer Ahmed 15
```

An external file named "output.txt" is open in a text editor, displaying the same list of names and numbers as the terminal output.

### // Task 3

Use ls and find to list and print all lines matching a particular pattern in matching files.



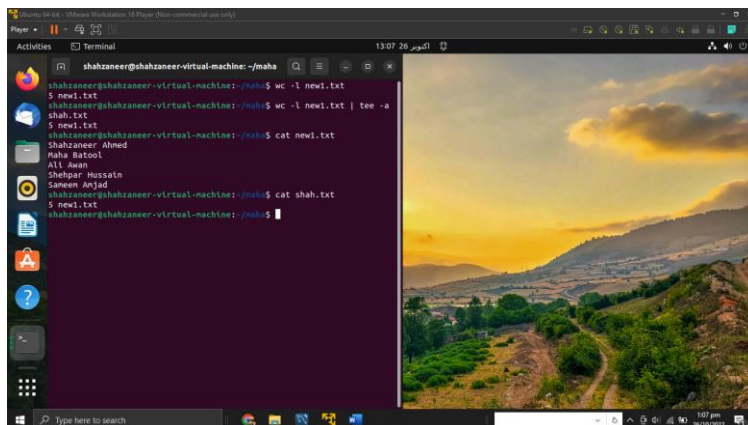
A screenshot of a Linux terminal window titled "shahzaneer@shahzaneer-virtual-machine: ~/maha". The terminal shows the following command and output:

```
shahzaneer@shahzaneer-virtual-machine:~/maha$ find . -name "*.txt"
./file2.txt
./myfile.txt
./file3.txt
./file1.txt
```

The background of the terminal window shows a desktop environment with a landscape wallpaper and icons for "shah", "maha", and "output.txt".

### // Task 4

Use cat, grep, tee and wc command to read the particular entry from user and store in a file and print line count.



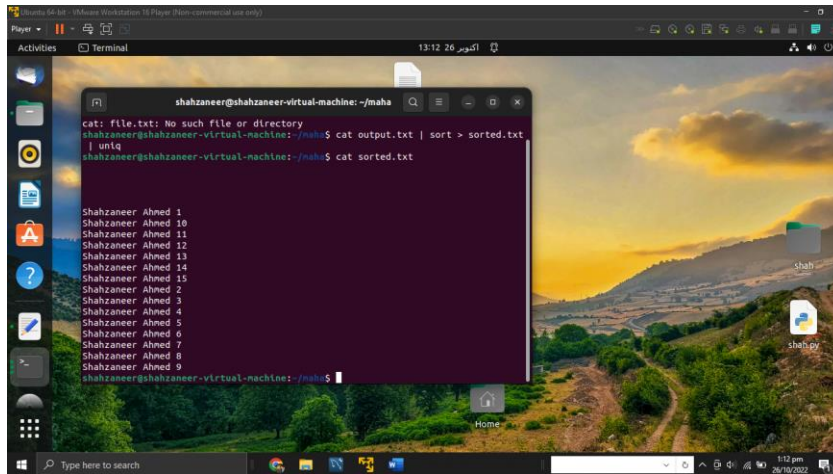
A screenshot of a Linux terminal window titled "shahzaneer@shahzaneer-virtual-machine: ~/maha". The terminal shows the following commands and output:

```
shahzaneer@shahzaneer-virtual-machine:~/maha$ wc -l new1.txt
5 new1.txt
shahzaneer@shahzaneer-virtual-machine:~/maha$ wc -l new1.txt | tee -a shah.txt
5 new1.txt
shahzaneer@shahzaneer-virtual-machine:~/maha$ cat new1.txt
Shahzaneer Ahmed
Maha Batool
Ali Awan
Shehpar Hussain
Saneem Anjad
shahzaneer@shahzaneer-virtual-machine:~/maha$ cat shah.txt
5 new1.txt
```

The background of the terminal window shows a desktop environment with a landscape wallpaper and icons for "shah", "maha", and "output.txt".

## // Task 5

Pipes the output from the cat (concatenate) process into the sort process to produce sorted output, and then pipes the sorted output into the uniq process to eliminate duplicate records.



## Lab 5

### // Task 1

Change the program given in Activity 3 such that it accepts the input at command-line.

```
#include<iostream>
using namespace std;
int main ()
{
    int arr[10], n, i, max, min;
    cout << "Enter the size of the array : ";
    cin >> n;
    cout << "Enter the elements of the array : ";
    for (i = 0; i < n; i++){
        cin >> arr[i];
    }
    max = arr[0];
    for (i = 0; i < n; i++){
```

```

    if (max < arr[i])
        max = arr[i];
}

min = arr[0];
for (i = 0; i < n; i++){
    if (min > arr[i])
        min = arr[i];
}

cout << "Largest element : " << max;
cout<< "Smallest element : " << min;

return 0;
}

```

The screenshot shows a terminal window titled "max\_min.cpp - Cpp - Visual Studio Code" running on a virtual machine named "ahmad@ahmad-virtual-machine". The terminal output is as follows:

```

ahmad@ahmad-virtual-machine: ~/Desktop/Cpp
ahmad@ahmad-virtual-machine:~/Desktop/Cpp$ g++ max_min.cpp -o max_min
ahmad@ahmad-virtual-machine:~/Desktop/Cpp$ ./max_min
Enter the size of the array : 5
Enter the elements of the array : 1 2 3 4 6
Largest element : 6Smallest element : 1ahmad@ahmad-virtual-machine:~/Desktop/Cpp$

```

The background shows the Visual Studio Code interface with the file explorer displaying "max\_min.cpp" and the code editor showing the C++ code for finding the largest and smallest elements in an array.

// Task 2

Write a C++ program that accepts a number as input and find whether it is a palindrome or not

```
#include<iostream>

#include<string>

using namespace std;

int main(){

int number;

cout<<"Enter a number to check is a number is palindrome or not";

cin>>number;

int num,count;

count=0;

num=number;

string charnum;

stringstream ss;

ss<<number;

ss>>charnum;

while(num!=0){

num=num/10;

count++;

}

bool gg=true;

if(count%2==1){

int i = 0;

count--;

int half = count/2;

while(i!=half){

if(charnum.at(i)!=charnum.at(count-i)){
```

```

gg=false;
}
i++;
}
}
else{
int i =0;
count--;
int half = count/2;
while(i!=half+1){
if(charnum.at(i)!=charnum.at(count-i)){
gg=false;
}
i++;
}
}
if(gg){
cout<<"Yes the number is palindrome";
}
else{
cout<<"NO the number is not a palindrome";
}}

```

## Lab 6

// Task 1

Write a C++ program that creates an array of size 1000 and populates it with random integers between 1 and 100. Now, it creates two child processes. The first child process finds how many prime numbers are there among first 500 number while the second child process finds the number of prime numbers among the remaining 500 numbers

```
#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/wait.h>

#include<cstdlib>

using namespace std;

int main(){

int arr[1000];

for(int i = 0 ; i<1000 ; i++ ){

arr[i]=(rand()%100);

}

int ch1 = fork();

if(ch1!=0){

cout<<"I am the parent process of first child with pid: "<<getpid()<<endl;

}

else{

int count=0;

for(int i = 0 ; i<499 ; i++){

int j =9;
```



```

while(j!=1){
    if(arr[i]%j==0){
        count++;
        break;
    }
    j--;
}

cout<<"I am the first child process with pid"<<getpid()<<endl;

cout<<"Count of prime numbers in first half is "<<(500-count)<<endl;

exit(0);
}

int ch2 = fork();

if(ch2!=0){

    cout<<"I am the parent of the second child process with pid

    :"<<getpid()<<endl;

}

else{

    int count = 0;

    for(int i = 499 ; i<1000 ; i++){

        int j =9;

        while(j!=1){

```

```
if(arr[i]%j==0){  
    count++;  
    break;  
}  
j--;  
}  
}  
  
cout<<"I am the second child process with pid"<<getpid()<<endl;  
cout<<"Count of prime numbers in second half is "<<(500-count)<<endl;  
exit(0);  
}  
}
```