**Operating System Lab**

**Lab Exercise #5**



**Session: 2021**

**Submitted by:**

**Wali Muhammad 2021-SE-39**

**Submitted to:**

**Mr. Waqas Ali**

Department of Computer Science, New Campus

**University of Engineering and Technology Lahore, Pakistan**

**Task 1**

**Description**

Step1: create a file with name “pipe1.c”. Type in the following code.

**Solution**

**Code**

#include<unistd.h>

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

int main(int argc , char \*\* argv)

{

int data\_processed=0;

int file\_descripter[2];

const char some\_data [] = "123";

char buffer[BUFSIZ + 1];

memset (buffer, '\0', sizeof(buffer));

if(pipe (file\_descripter) == 0)

{

data\_processed = write (file\_descripter[1],buffer,sizeof(buffer));

printf("Wrote %d bytes :", data\_processed);

data\_processed = read(file\_descripter[0],buffer,sizeof(buffer));

printf("Read %d, byte %s ", data\_processed, buffer);

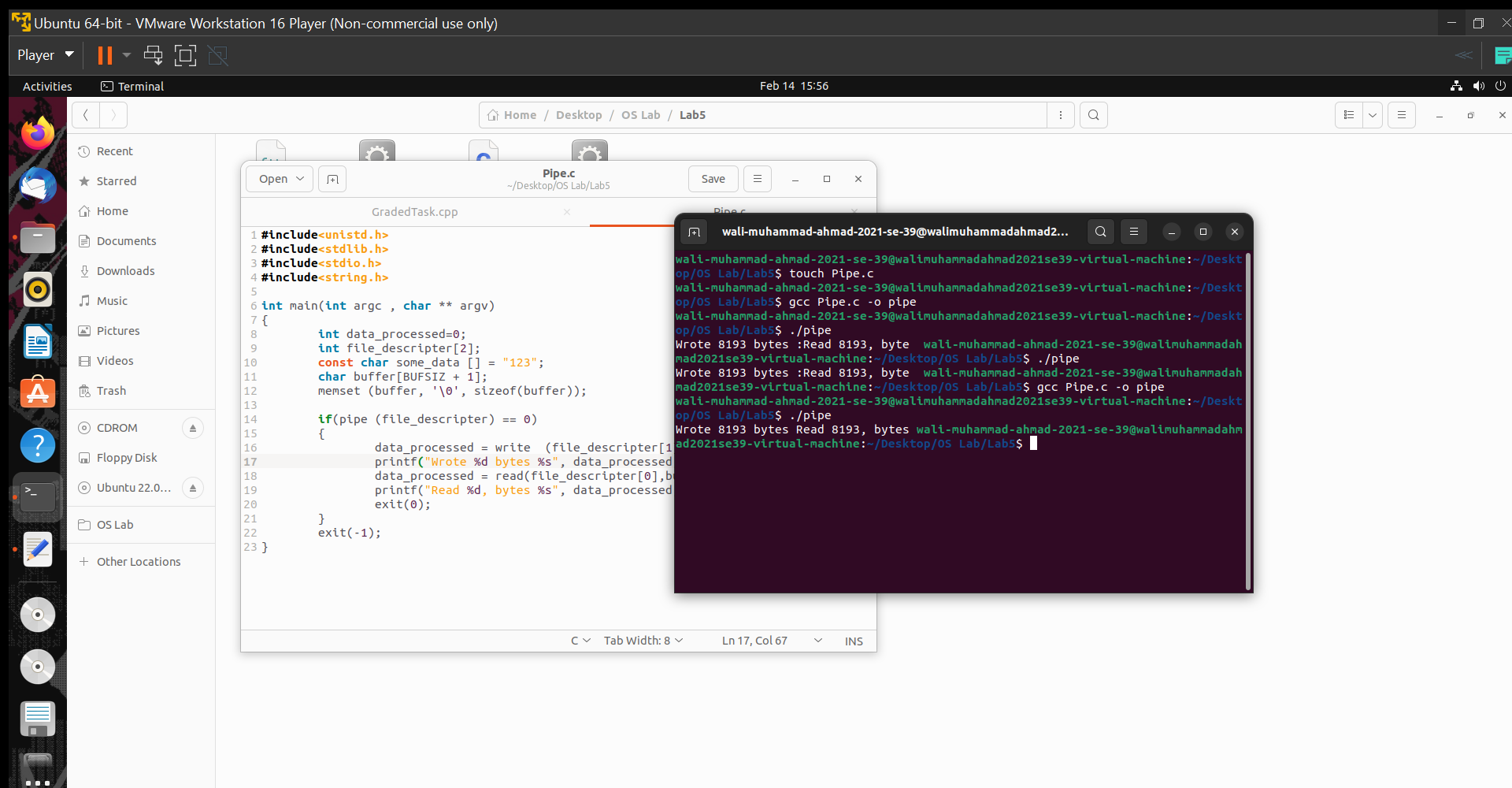
exit(0);

}

exit(-1);

}

**Code Execution Screenshot**

****

**Graded Task**

**Description**

Multiple pipes across a fork/exec

In this task, you will write a program that has two-way communication between parent and

child. Write a c program in which parent receive array of integer from command line

argument and pass it to child using pipe, child need to sort that array (using bubble sort)

and return that array to parent using pipe and parent need to display that sorted array.

**Solution**

**Code**

#include<iostream>

#include<unistd.h>

#include<sys/wait.h>

using namespace std;

int main(int argc,char \*args[])

{

int file\_descriptor1[2];

int file\_descriptor2[2];

int size=argc-1;

bool swap = false;

int data[size];

for(int i=0;i<size;i++)

{

data[i]=atoi(args[i+1]);

}

if(pipe(file\_descriptor1)==-1)

{

cout<<"Failed to Create Pipe 1\n";

}

if(pipe(file\_descriptor2)==-1)

{

cout<<"Failed to Create Pipe 2\n";

}

if(fork()>0)

{

close(file\_descriptor1[0]);

write(file\_descriptor1[1],data,sizeof(data));

close(file\_descriptor1[1]);

wait(EXIT\_SUCCESS);

close(file\_descriptor2[1]);

read(file\_descriptor2[0],data,sizeof(data));

close(file\_descriptor2[0]);

//Loop to display the values in Array

cout<<"Sorted Attay"<<endl;

for(int i=0;i<size;i++)

{

cout<<data[i]<<" ";

}

cout<<endl;

}

else

{

close(file\_descriptor1[1]);

read(file\_descriptor1[0],data,sizeof(data));

close(file\_descriptor1[0]);

//Nested Loop for Bubble sort

do

{

swap = false;

for(int i=0; i < size; i++)

{

if(argv[i] > argv[i+1])

{

temp = argv[i];

argv[i] = argv[i+1];

argv[i+1] = temp;

swap = true;

}

}

}while(swap);

close(file\_descriptor2[0]);

write(file\_descriptor2[1],data,sizeof(data));

close(file\_descriptor2[1]);

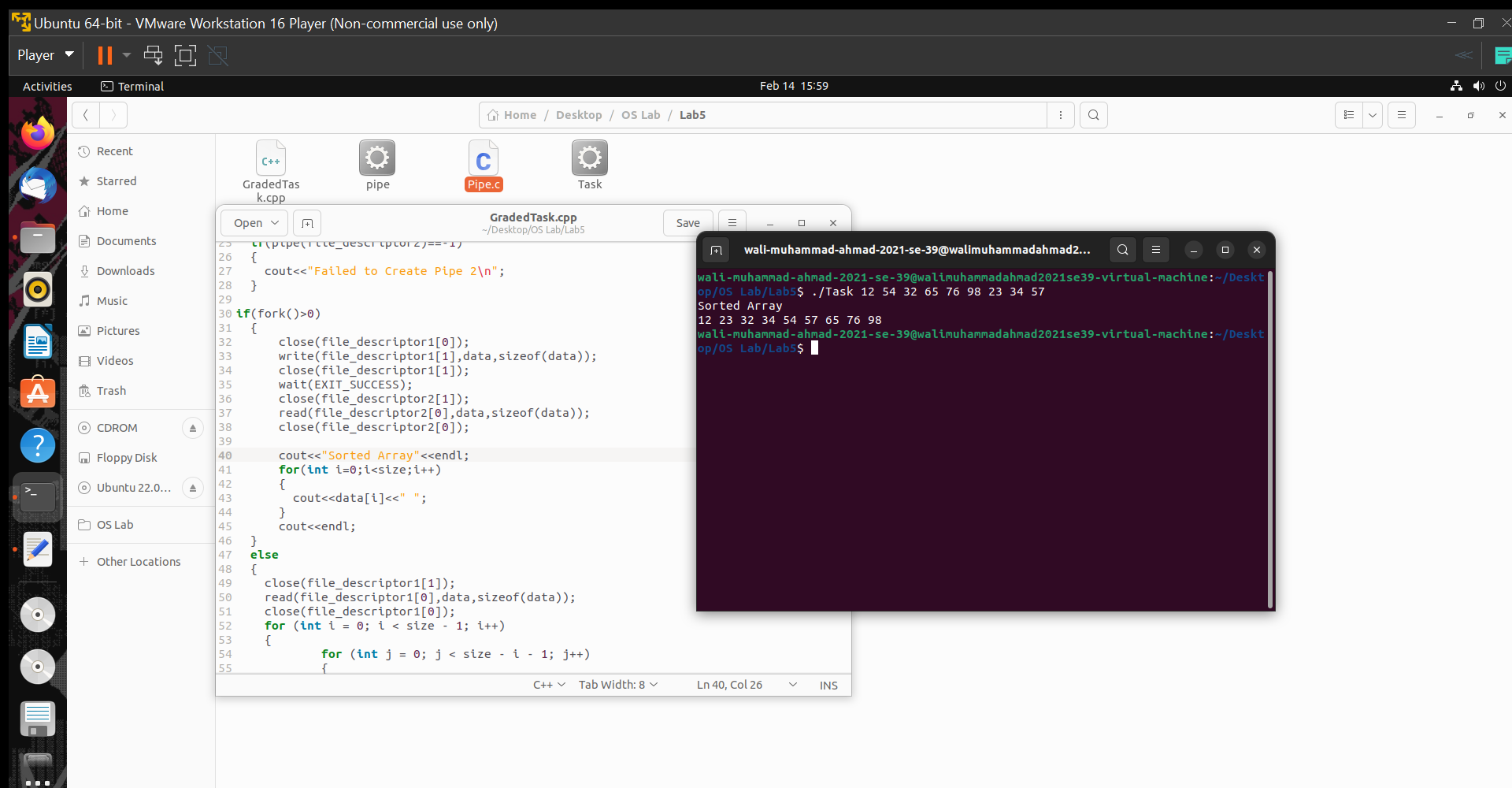
exit(EXIT\_SUCCESS);

}

return 0;

}

**Code Execution Screenshot**

****

**THE END**