Lab Title: lab 1	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator	Level of Achievements				
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		

Lab Instructor:

“lab 1”

COURSE :

OPERATING SYSTEM

SUBMITTED TO :

Sir shehzad

SUBMITTED BY :

Rabia Batool (2022-BSE-064)

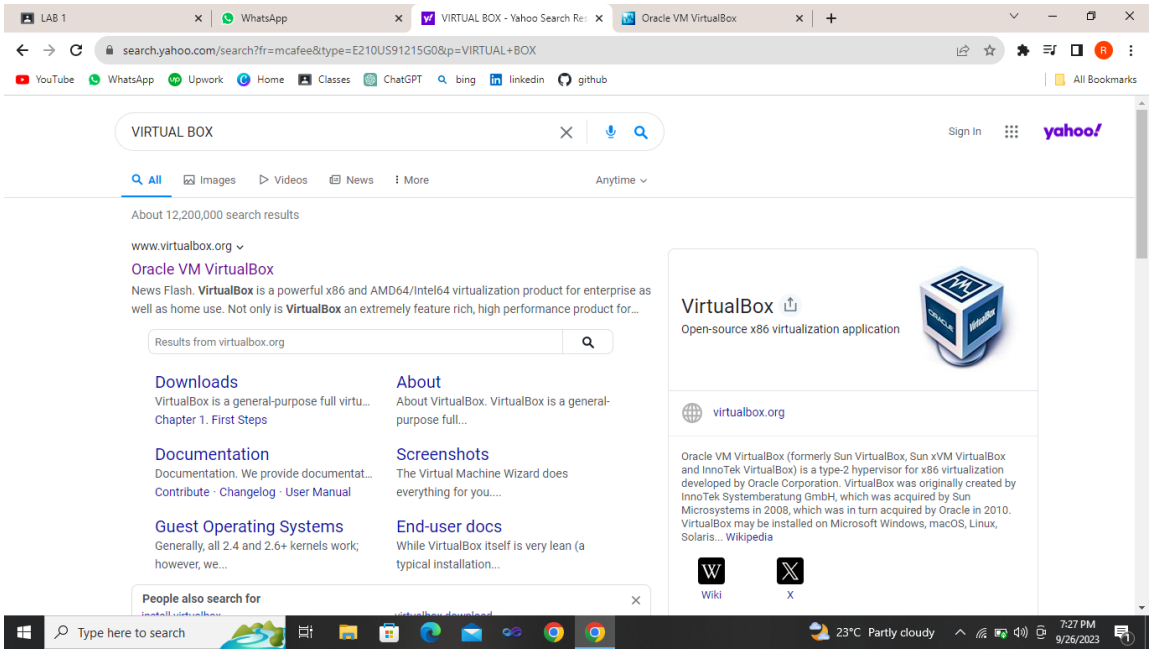
SECTION :

B

Operating system

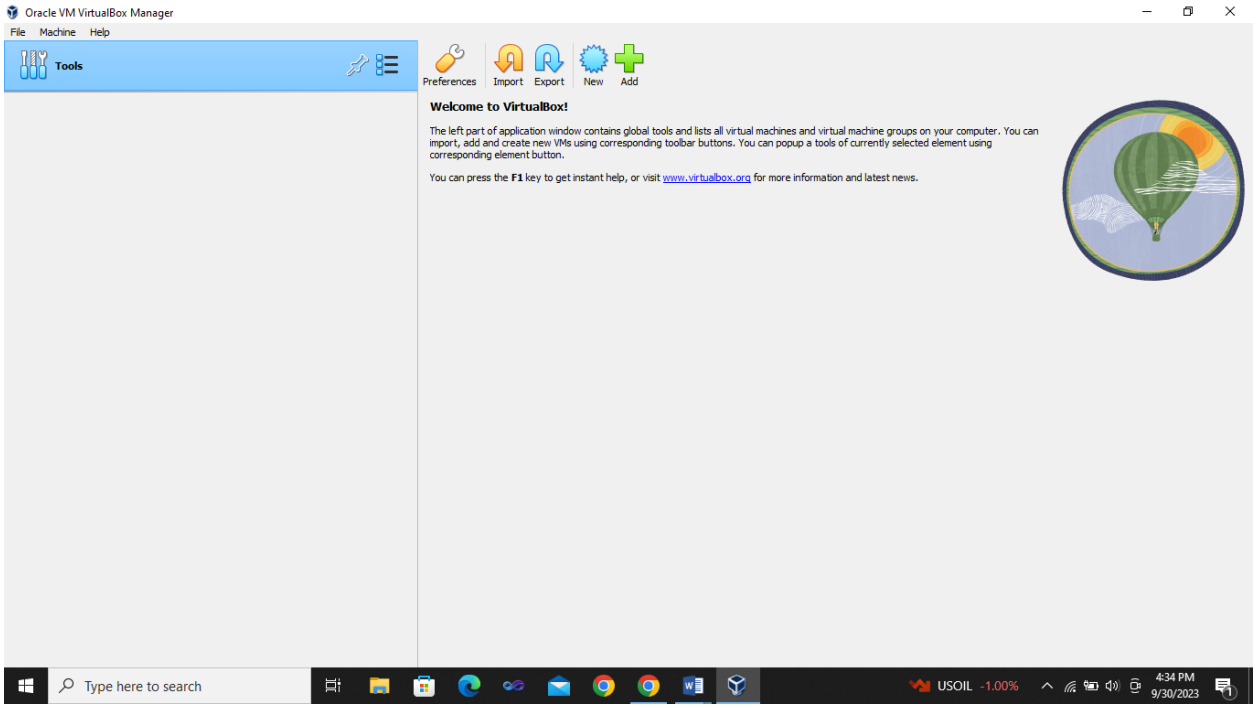
Task 1

Virtual box installation



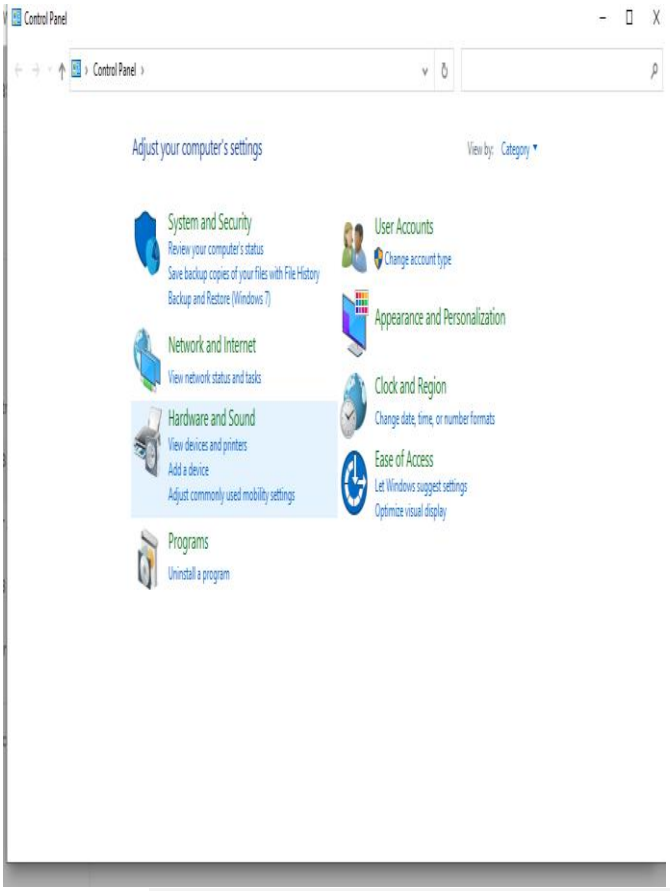

LAB 1
WhatsApp
VIRTUAL BOX - Yahoo Search Re
Downloads – Oracle VM VirtualB
virtualbox.org/wiki/Downloads
YouTube
WhatsApp
Upwork
Home
Classes
ChatGPT
bing
linkedin
github
All Bookmarks
VirtualBox
search...
Login
Preferences
Start Page
Index
History
About
Screenshots
Downloads
Documentation
End-user docs
Technical docs
Contribute
Community
Download VirtualBox
Here you will find links to VirtualBox binaries and its source code.
VirtualBox binaries
By downloading, you agree to the terms and conditions of the respective license.
If you're looking for the latest VirtualBox 6.1 packages, see VirtualBox 6.1 builds. Version 6.1 will remain supported until December 2023.
VirtualBox 7.0.10 platform packages
Windows hosts
macOS / Intel hosts
Linux distributions
Solaris hosts
Solaris 11 IPS hosts
The binaries are released under the terms of the GPL version 3.
See the changelog for what has changed.
You might want to compare the checksums to verify the integrity of downloaded packages. The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!
Type here to search
23°C Partly cloudy
7:29 PM
9/26/2023

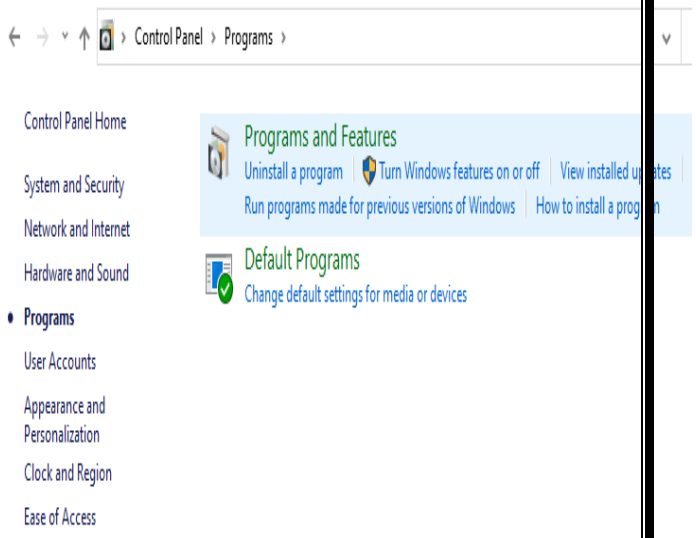

LAB 1
WhatsApp
VIRTUAL BOX - Yahoo Search Re
Downloads – Oracle VM VirtualB
virtualbox.org/wiki/Downloads
YouTube
WhatsApp
Upwork
Home
Classes
ChatGPT
bing
linkedin
github
All Bookmarks
Recent Downloads
VirtualBox-7.0.10-158379-Win.exe
5.2/106 MB • 10 minutes left
Show all downloads
VirtualBox
Start Page
Index
History
About
Screenshots
Downloads
Documentation
End-user docs
Technical docs
Contribute
Community
Download VirtualBox
Here you will find links to VirtualBox binaries and its source code.
VirtualBox binaries
By downloading, you agree to the terms and conditions of the respective license.
If you're looking for the latest VirtualBox 6.1 packages, see VirtualBox 6.1 builds. Version 6.1 will remain supported until December 2023.
VirtualBox 7.0.10 platform packages
Windows hosts
macOS / Intel hosts
Linux distributions
Solaris hosts
Solaris 11 IPS hosts
The binaries are released under the terms of the GPL version 3.
See the changelog for what has changed.
You might want to compare the checksums to verify the integrity of downloaded packages. The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!
Type here to search
23°C Partly cloudy
7:30 PM
9/26/2023

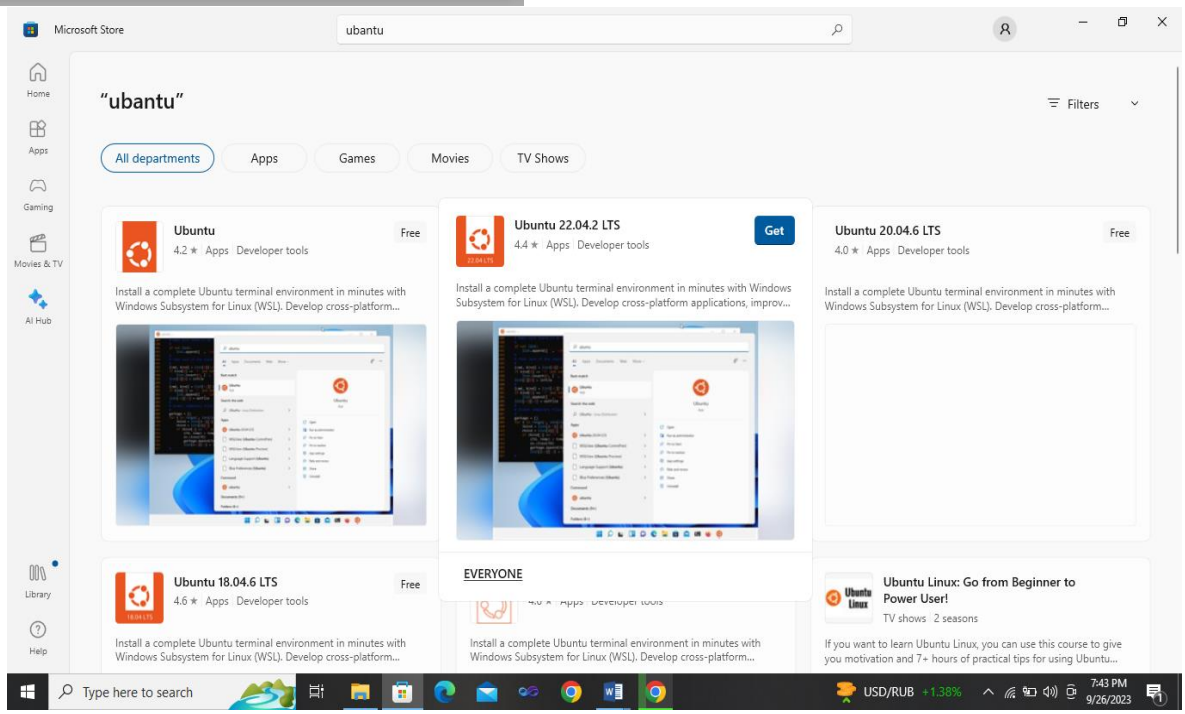


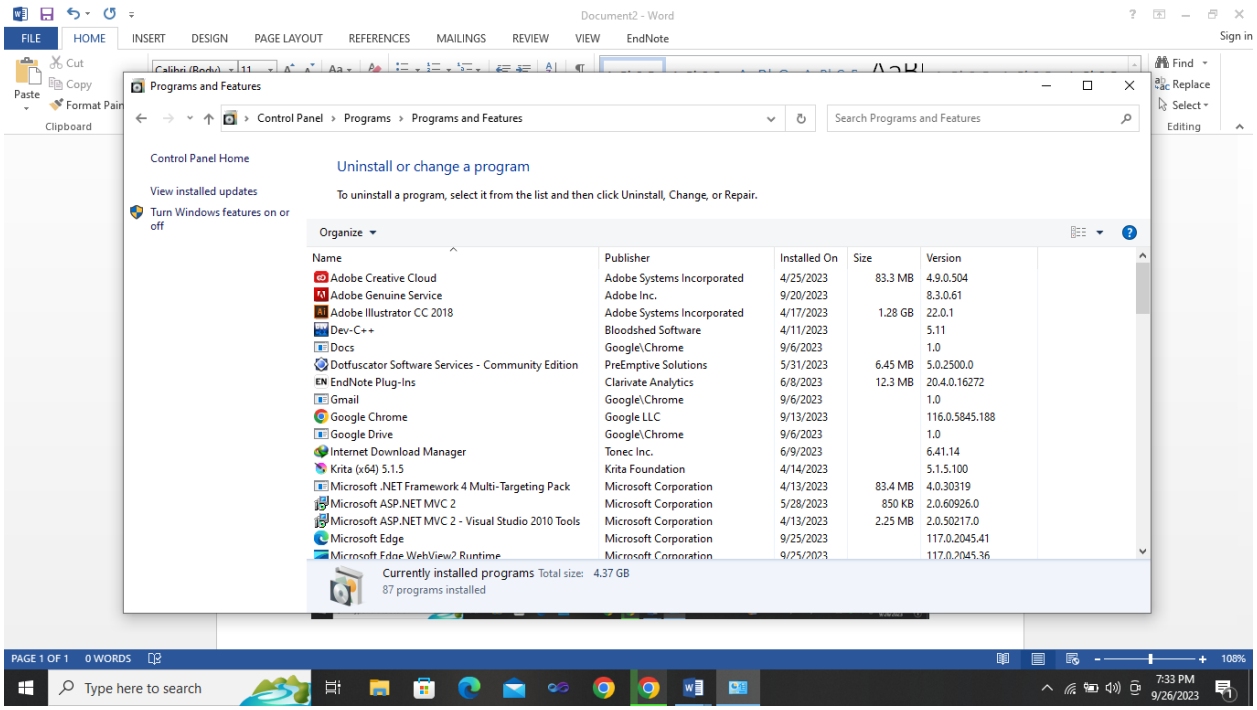
Operating system

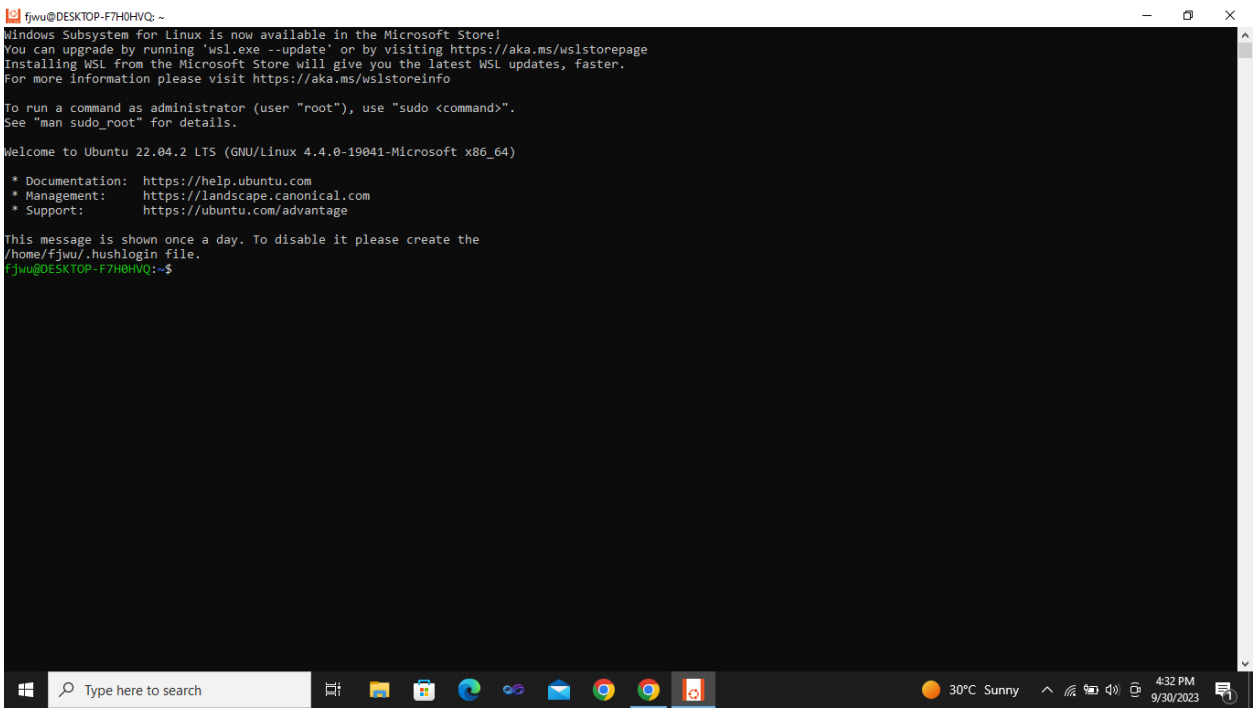
installation of ubunto

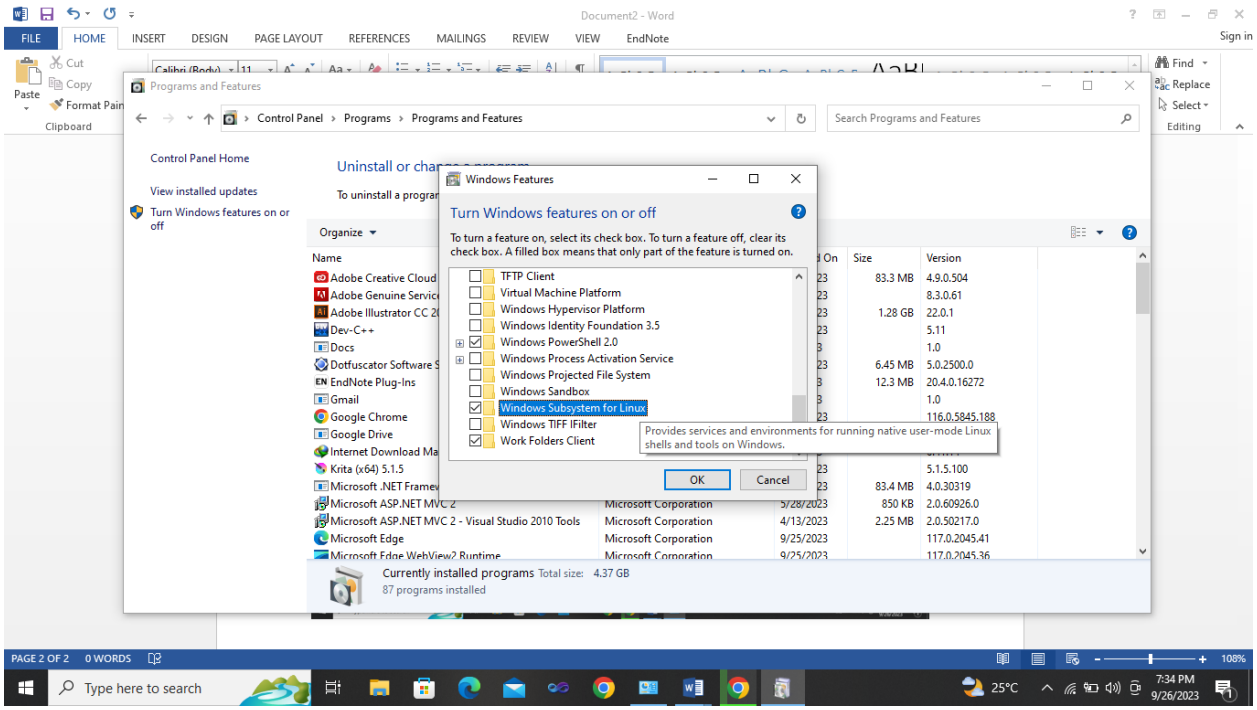

Control Panel
Control Panel
Adjust your computer's settings
View by: Category
System and Security
Review your computer's status
Save backup copies of your files with File History
Backup and Restore (Windows 7)
Network and Internet
View network status and tasks
Hardware and Sound
View devices and printers
Add a device
Adjust commonly used mobility settings
Programs
Uninstall a program
User Accounts
Change account type
Appearance and Personalization
Clock and Region
Change date, time, or number formats
Ease of Access
Let Windows suggest settings
Optimize visual display


Control Panel › Programs ›
Control Panel Home
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts
Appearance and Personalization
Clock and Region
Ease of Access
Programs and Features
Uninstall a program
Turn Windows features on or off
Run programs made for previous versions of Windows
Default Programs
Change default settings for media or devices


Microsoft Store
ubantu
"ubantu"
Filters
All departments
Apps
Games
Movies
TV Shows
Ubuntu
4.2 ★ Apps Developer tools
Free
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL). Develop cross-platform...
Ubuntu 22.04.2 LTS
4.4 ★ Apps Developer tools
Get
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL). Develop cross-platform applications, improv...
Ubuntu 20.04.6 LTS
4.0 ★ Apps Developer tools
Free
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL). Develop cross-platform...
Ubuntu 18.04.6 LTS
4.6 ★ Apps Developer tools
Free
Install a complete Ubuntu terminal environment in minutes with Windows Subsystem for Linux (WSL). Develop cross-platform...
EVERYONE
Ubuntu Linux: Go from Beginner to Power User!
TV shows 2 seasons
If you want to learn Ubuntu Linux, you can use this course to give you motivation and 7+ hours of practical tips for using Ubuntu...
Type here to search
USD/RUB +1.38%
7:43 PM
9/26/2023

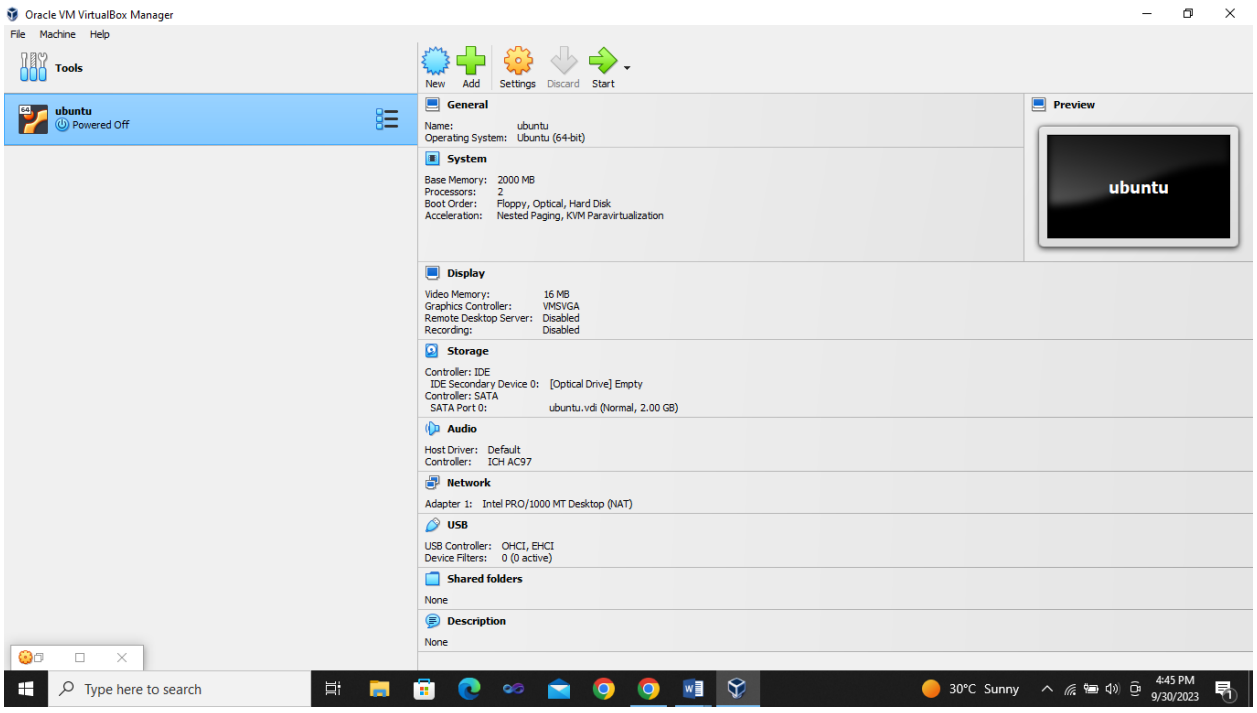

Document2 - Word
FILE
HOME
INSERT
DESIGN
PAGE LAYOUT
REFERENCES
MAILINGS
REVIEW
VIEW
EndNote
Sign in
Programs and Features
Control Panel › Programs › Programs and Features
Search Programs and Features
Control Panel Home
View installed updates
Turn Windows features on or off
Uninstall or change a program
To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.
Organize
Name
Publisher
Installed On
Size
Version
Adobe Creative Cloud
Adobe Systems Incorporated
4/25/2023
83.3 MB
4.9.0.504
Adobe Genuine Service
Adobe Inc.
9/20/2023
8.3.0.61
Adobe Illustrator CC 2018
Adobe Systems Incorporated
4/17/2023
1.28 GB
22.0.1
Dev-C++
Bloodshed Software
4/11/2023
5.11
Docs
Google\Chrome
9/6/2023
1.0
Dotfuscator Software Services - Community Edition
PreEmptive Solutions
5/31/2023
6.45 MB
5.0.2500.0
EndNote Plug-Ins
Clarivate Analytics
6/8/2023
12.3 MB
20.4.0.16272
Gmail
Google\Chrome
9/6/2023
1.0
Google Chrome
Google LLC
9/13/2023
116.0.5845.188
Google Drive
Google\Chrome
9/6/2023
1.0
Internet Download Manager
Tonec Inc.
6/9/2023
6.41.14
Krita (x64) 5.1.5
Krita Foundation
4/14/2023
5.1.5.100
Microsoft .NET Framework 4 Multi-Targeting Pack
Microsoft Corporation
4/13/2023
83.4 MB
4.0.30319
Microsoft ASP.NET MVC 2
Microsoft Corporation
5/28/2023
850 KB
2.0.60926.0
Microsoft ASP.NET MVC 2 - Visual Studio 2010 Tools
Microsoft Corporation
4/13/2023
2.25 MB
2.0.50217.0
Microsoft Edge
Microsoft Corporation
9/25/2023
117.0.2045.41
Microsoft Edge WebView2 Runtime
Microsoft Corporation
9/25/2023
117.0.2045.36
Currently installed programs Total size: 4.37 GB
87 programs installed
PAGE 1 OF 1
0 WORDS
108%
Type here to search
7:33 PM
9/26/2023

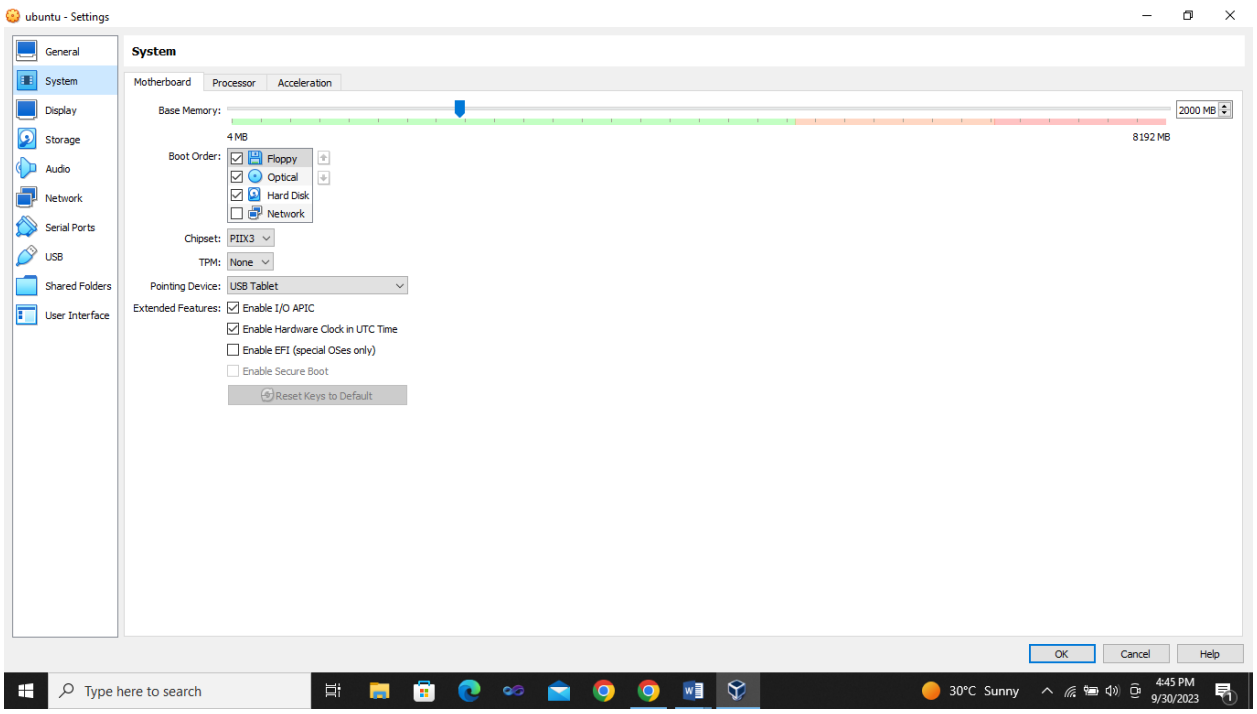

fjwu@DESKTOP-F7H0HVQ: ~
Windows Subsystem for Linux is now available in the Microsoft Store!
You can upgrade by running 'wsl.exe --update' or by visiting https://aka.ms/wslstorepage
Installing WSL from the Microsoft Store will give you the latest WSL updates, faster.
For more information please visit https://aka.ms/wslstoreinfo
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 4.4.0-19041-Microsoft x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
This message is shown once a day. To disable it please create the
/home/fjwu/.hushlogin file.
fjwu@DESKTOP-F7H0HVQ:~$
Type here to search
30°C Sunny
4:32 PM
9/30/2023

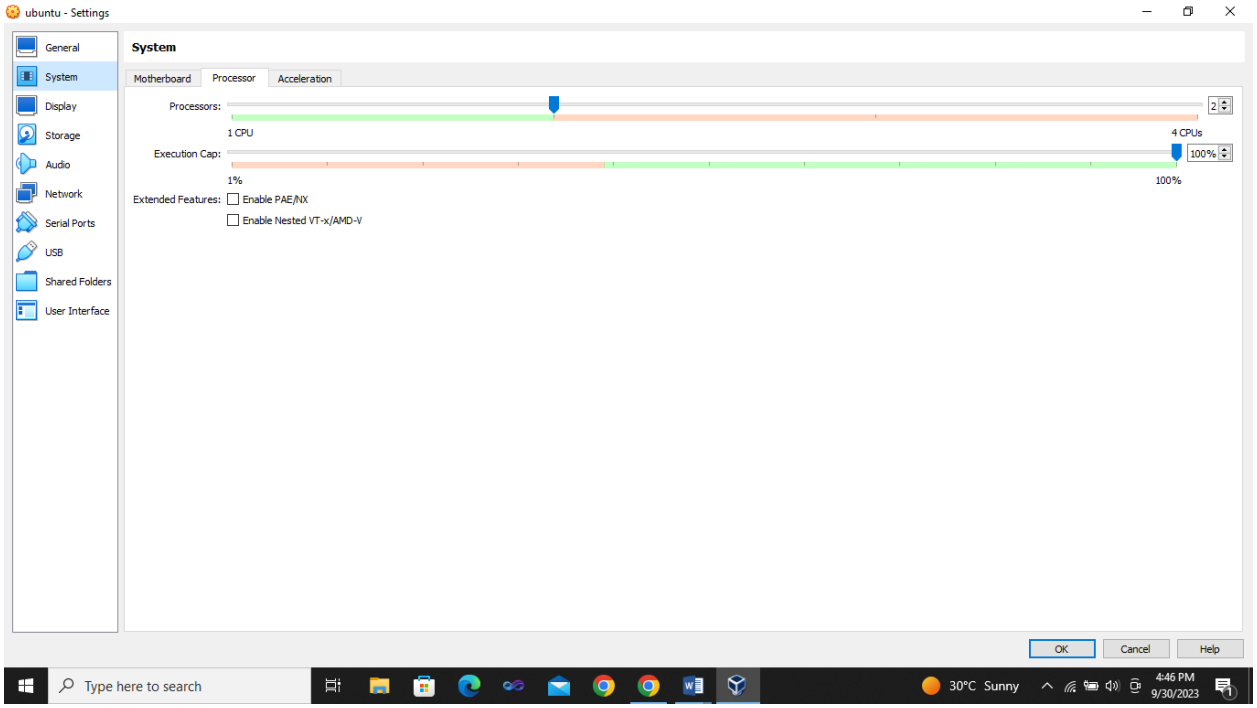

Document2 - Word
Programs and Features
Control Panel › Programs › Programs and Features
Search Programs and Features
Control Panel Home
View installed updates
Turn Windows features on or off
Windows Features
Turn Windows features on or off
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.
TFTP Client
Virtual Machine Platform
Windows Hypervisor Platform
Windows Identity Foundation 3.5
Windows PowerShell 2.0
Windows Process Activation Service
Windows Projected File System
Windows Sandbox
Windows Subsystem for Linux
Windows TIFF IFilter
Work Folders Client
Provides services and environments for running native user-mode Linux shells and tools on Windows.
OK
Cancel
Currently installed programs Total size: 4.37 GB
87 programs installed
PAGE 2 OF 2
0 WORDS
Type here to search
25°C
7:34 PM
9/26/2023

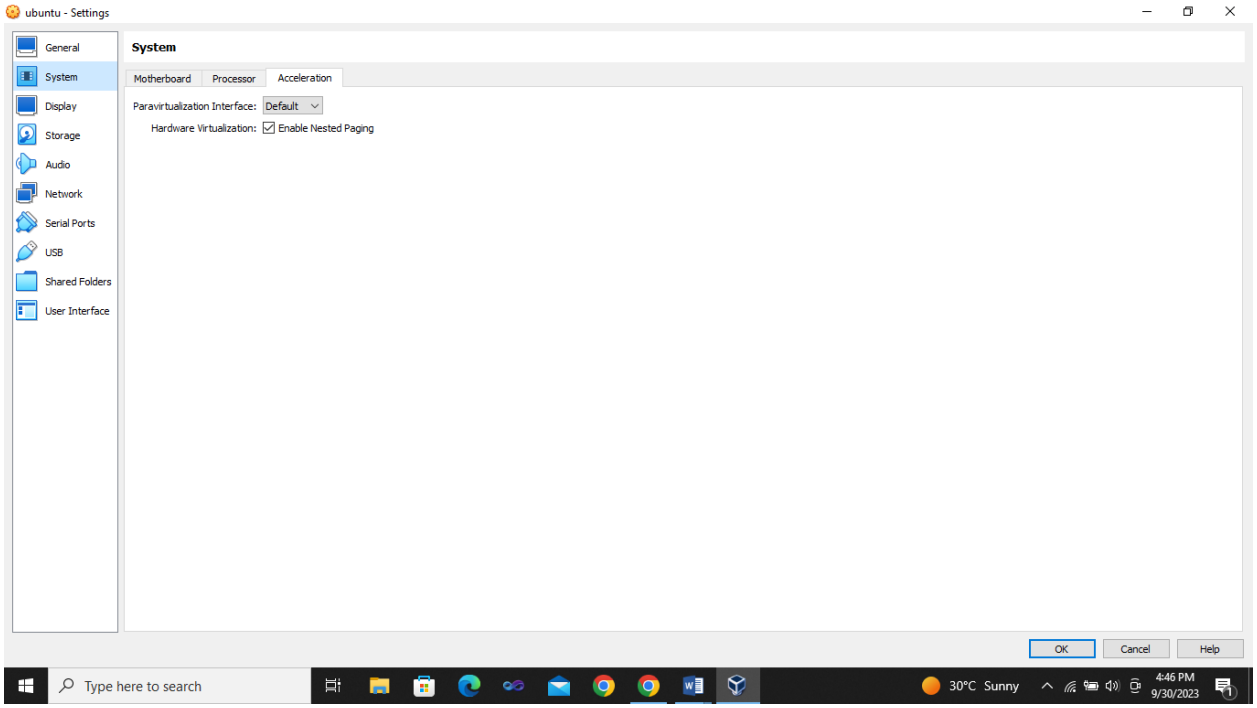
Operating system

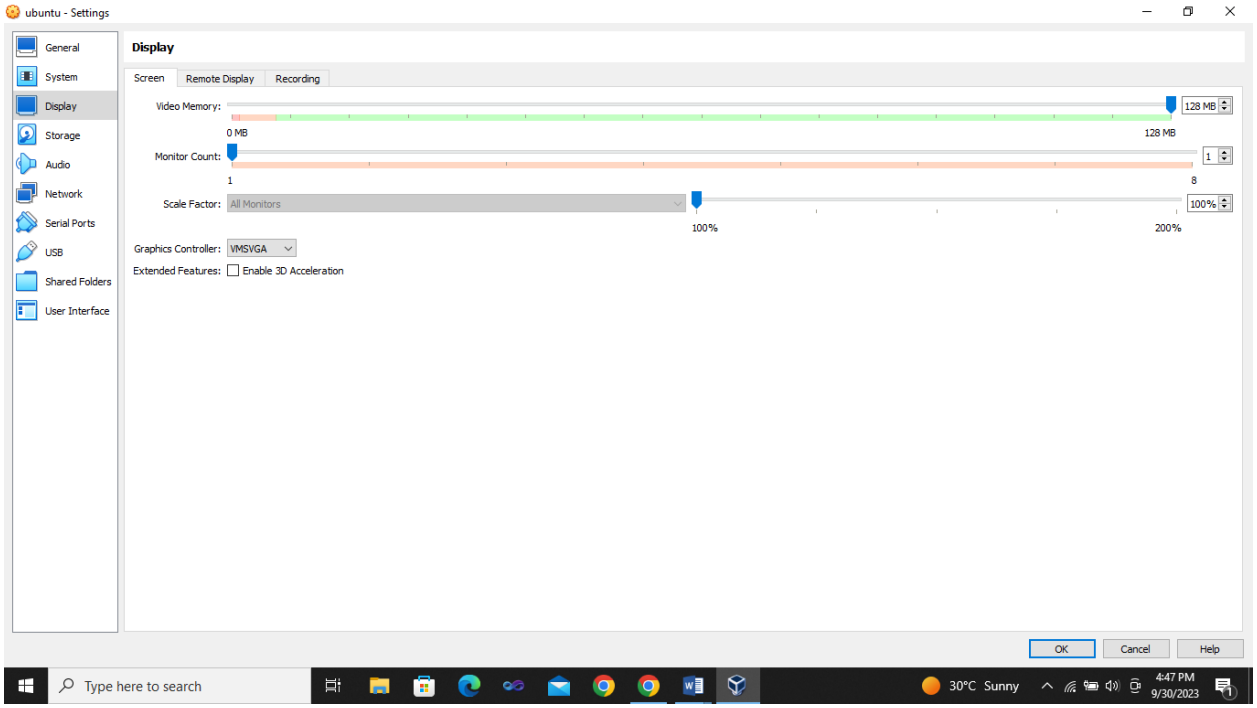
ubuntu with virtual bow

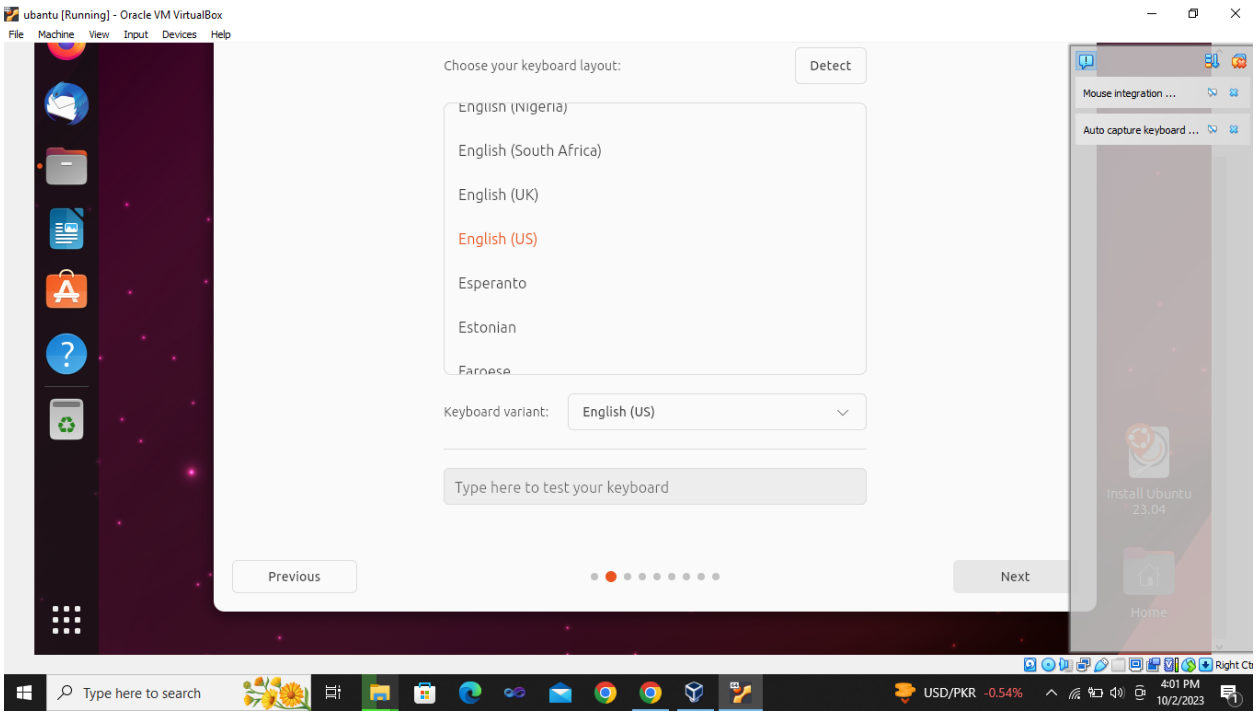


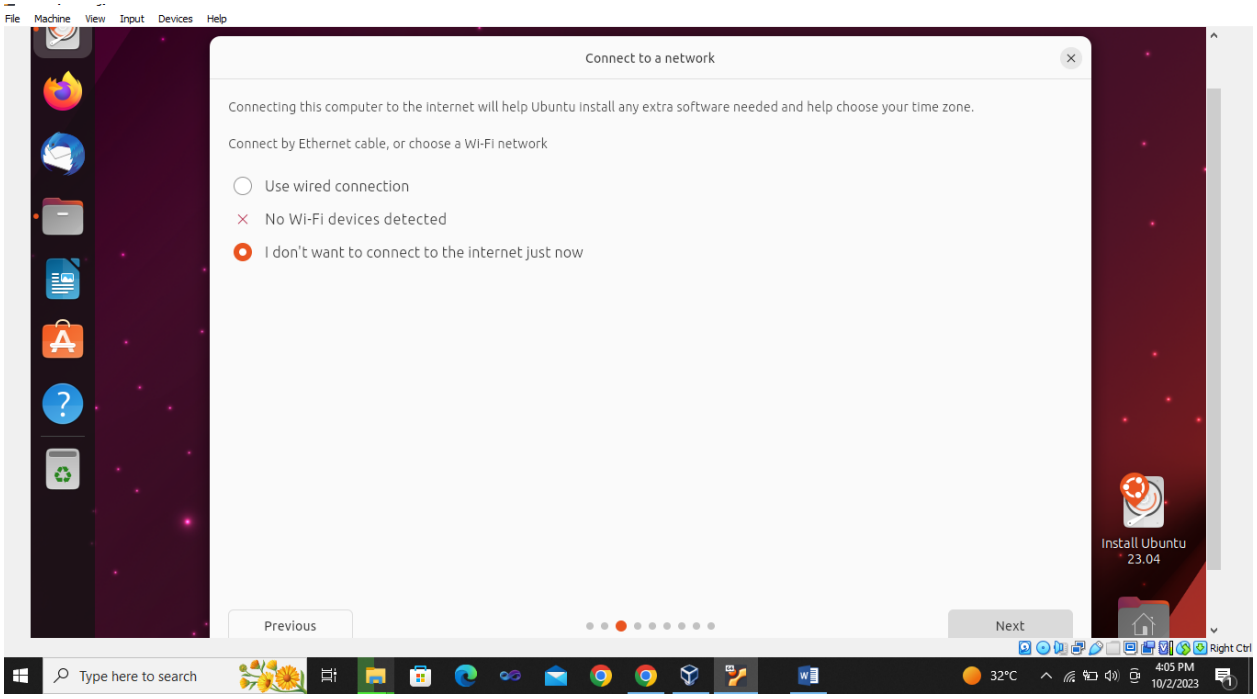


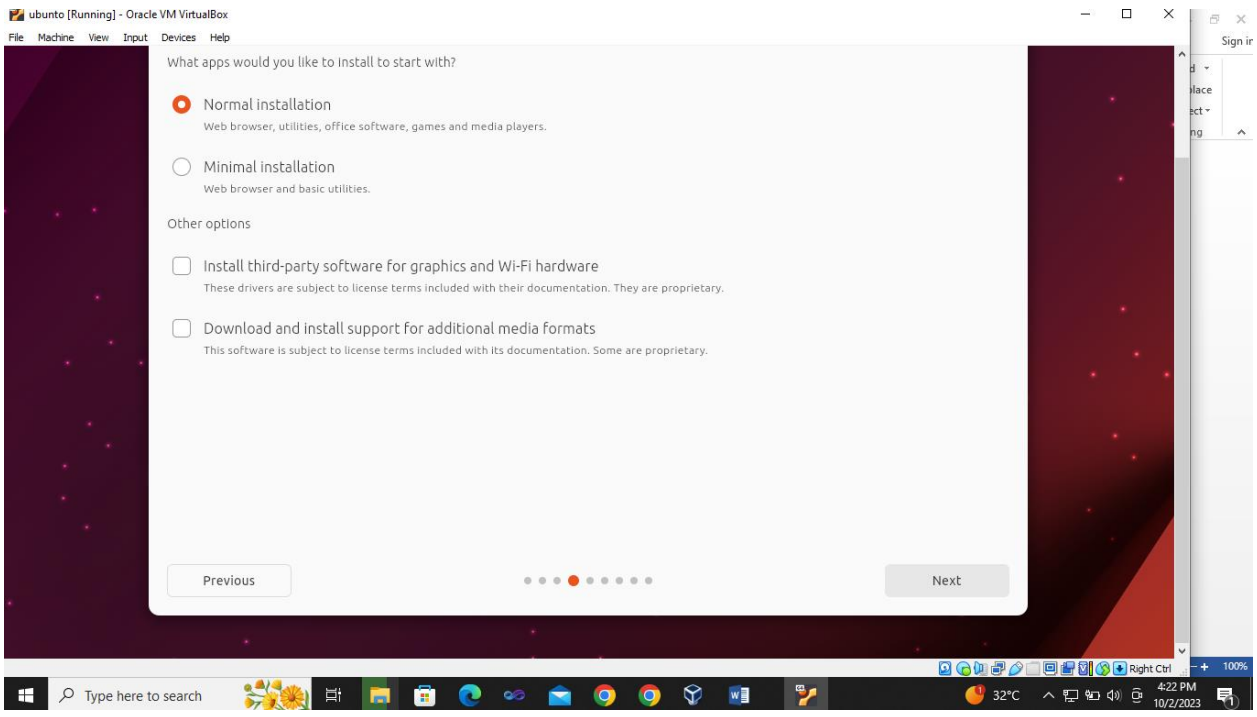

ubuntu - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
System
Motherboard
Processor
Acceleration
Processors:
2
1 CPU
4 CPUs
Execution Cap:
100%
1%
100%
Extended Features:
Enable PAE/NX
Enable Nested VT-x/AMD-V
OK
Cancel
Help
Type here to search
30°C Sunny
4:46 PM
9/30/2023

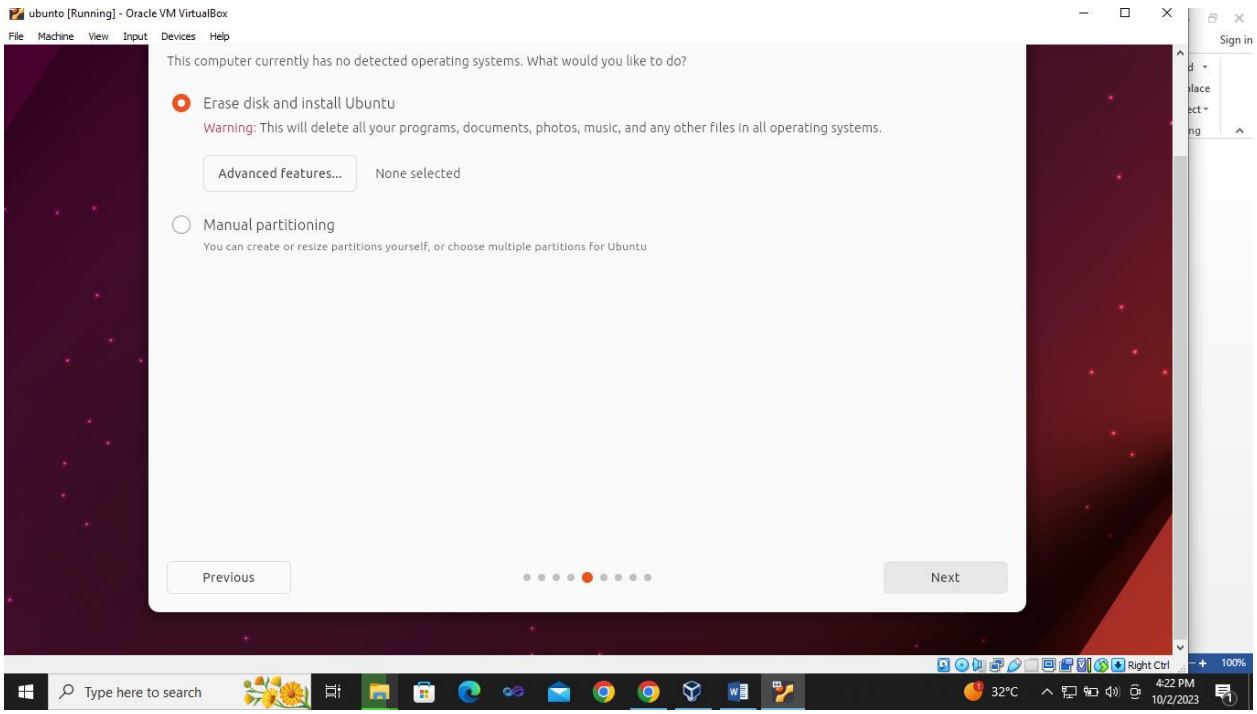

ubuntu - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
System
Motherboard
Processor
Acceleration
Paravirtualization Interface:
Default
Hardware Virtualization:
Enable Nested Paging
OK
Cancel
Help
Type here to search
30°C Sunny
4:46 PM
9/30/2023

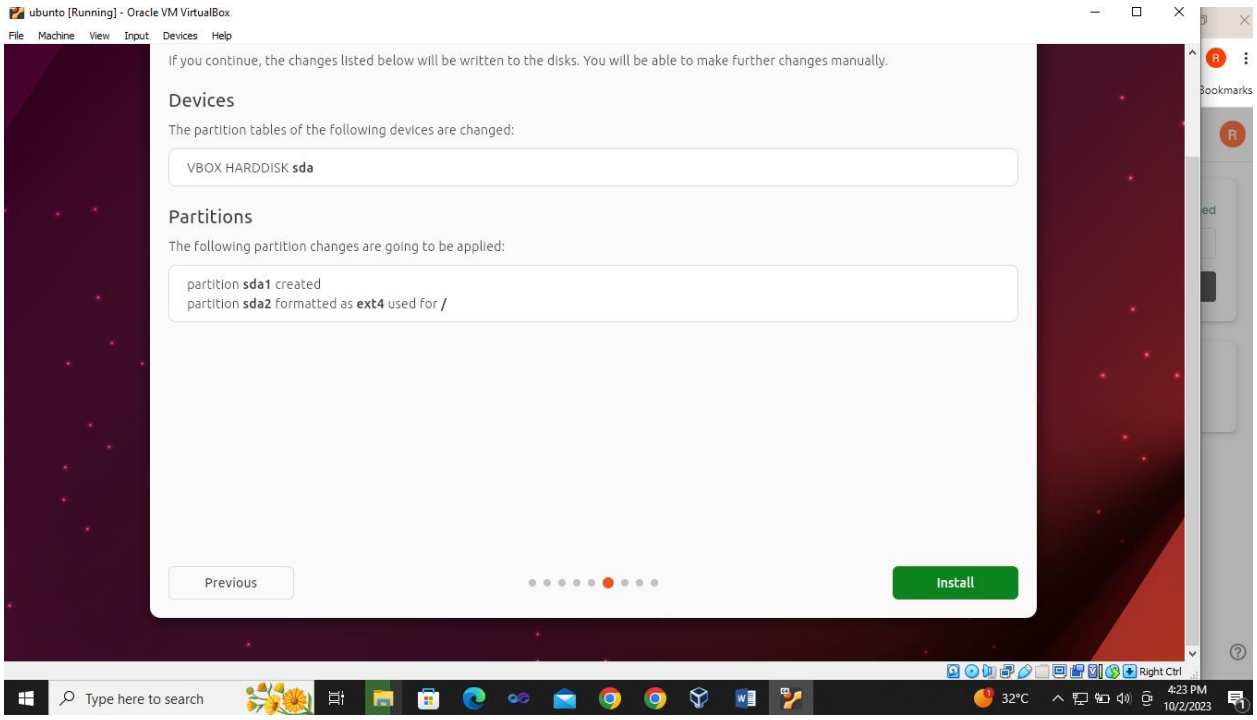

ubuntu - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Display
Screen
Remote Display
Recording
Video Memory:
128 MB
0 MB
128 MB
Monitor Count:
1
1
8
Scale Factor:
All Monitors
100%
100%
200%
Graphics Controller:
VMSVGA
Extended Features:
Enable 3D Acceleration
OK
Cancel
Help
Type here to search
30°C Sunny
4:47 PM
9/30/2023

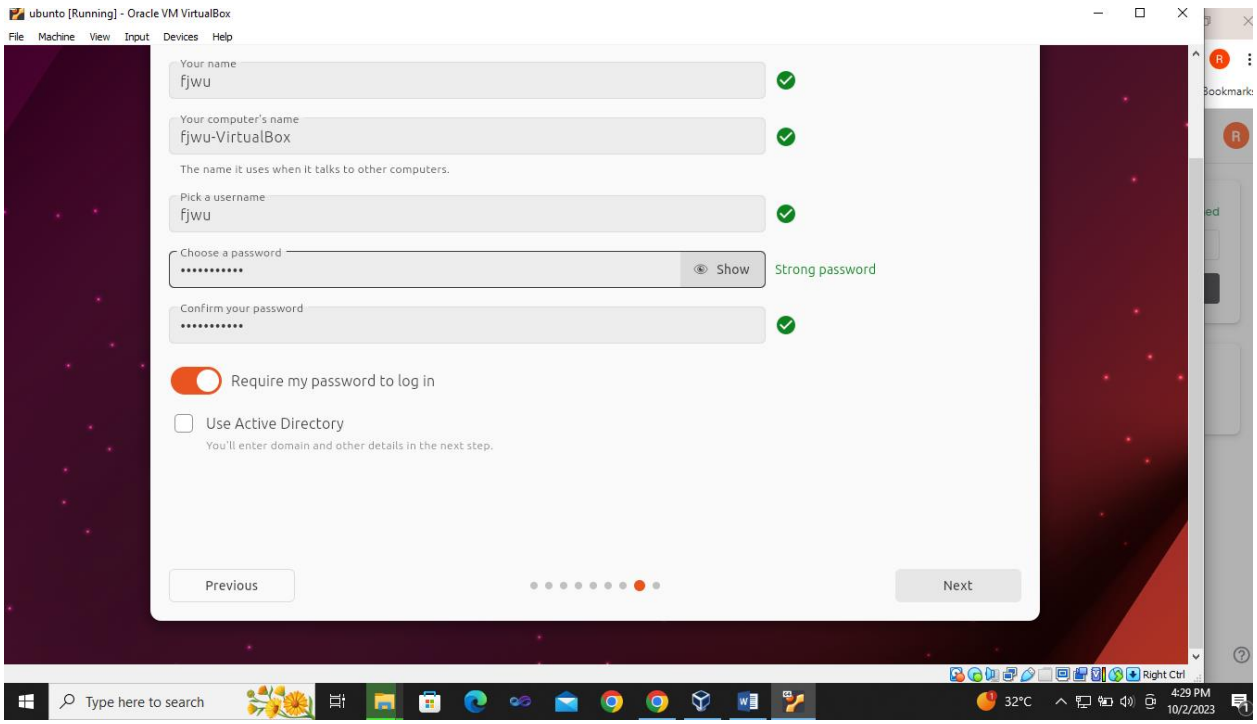

ubantu [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Choose your keyboard layout:
Detect
English (Nigeria)
English (South Africa)
English (UK)
English (US)
Esperanto
Estonian
Faroese
Keyboard variant:
English (US)
Type here to test your keyboard
Previous
Next
Mouse integration ...
Auto capture keyboard ...
Install Ubuntu 23.04
Home
Type here to search
USD/PKR -0.54%
4:01 PM
10/2/2023

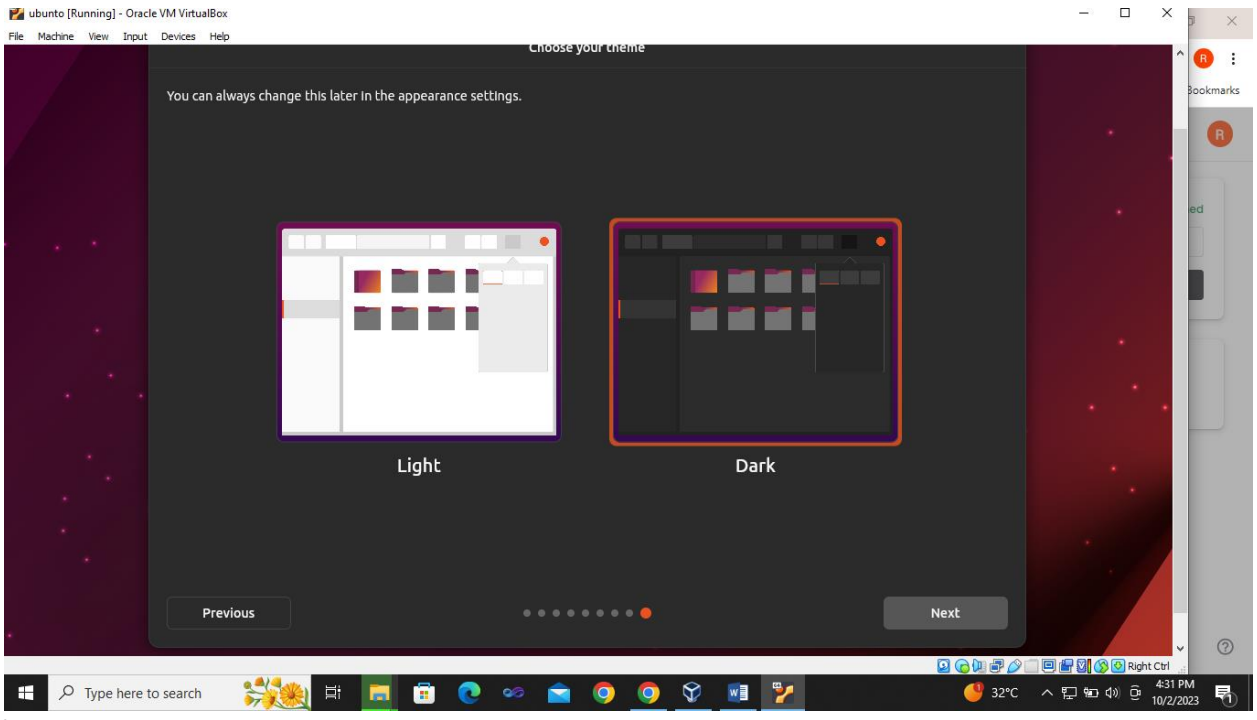

File Machine View Input Devices Help
Connect to a network
Connecting this computer to the internet will help Ubuntu install any extra software needed and help choose your time zone.
Connect by Ethernet cable, or choose a WI-FI network
Use wired connection
No Wi-Fi devices detected
I don't want to connect to the internet just now
Previous
Next
Install Ubuntu 23.04
Type here to search
32°C
4:05 PM
10/2/2023


ubunto [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
What apps would you like to install to start with?
Normal installation
Web browser, utilities, office software, games and media players.
Minimal installation
Web browser and basic utilities.
Other options
Install third-party software for graphics and Wi-Fi hardware
These drivers are subject to license terms included with their documentation. They are proprietary.
Download and install support for additional media formats
This software is subject to license terms included with its documentation. Some are proprietary.
Previous
Next
Type here to search
32°C
4:22 PM
10/2/2023

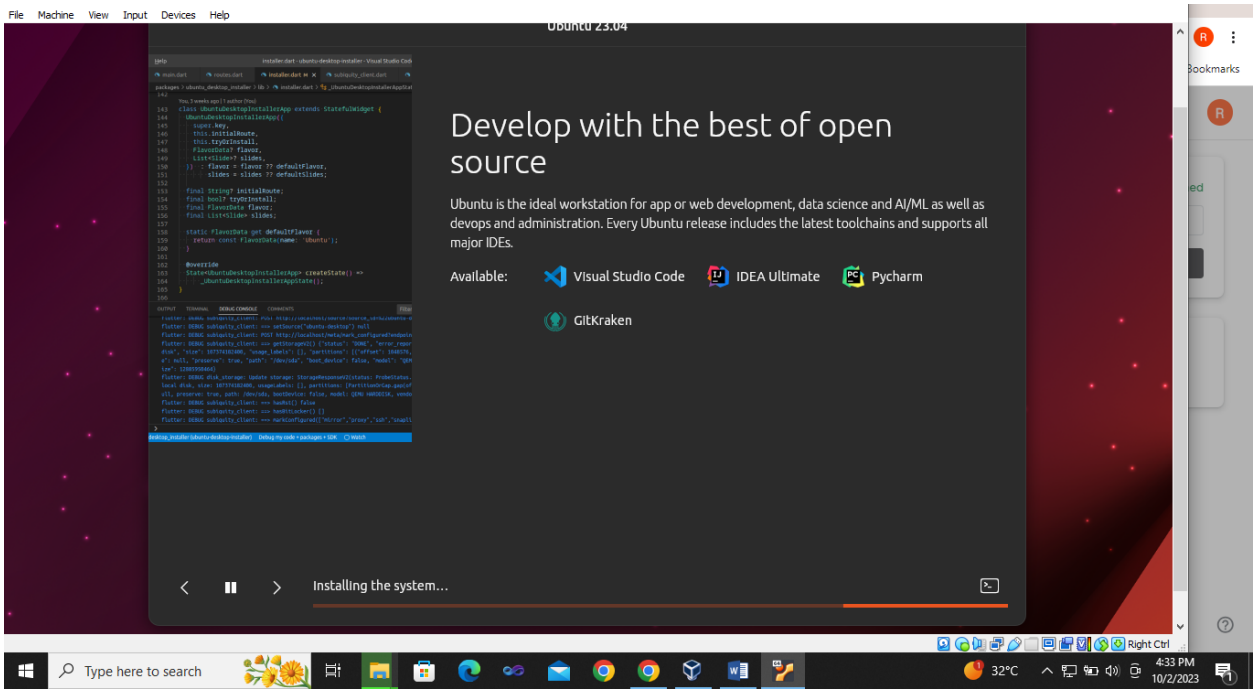

ubunto [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
This computer currently has no detected operating systems. What would you like to do?
Erase disk and install Ubuntu
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
Advanced features...
None selected
Manual partitioning
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu
Previous
Next
Type here to search
32°C
4:22 PM
10/2/2023


ubunto [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
If you continue, the changes listed below will be written to the disks. You will be able to make further changes manually.
Devices
The partition tables of the following devices are changed:
VBOX HARDDISK sda
Partitions
The following partition changes are going to be applied:
partition sda1 created
partition sda2 formatted as ext4 used for /
Previous
Install
Type here to search
32°C
4:23 PM
10/2/2023

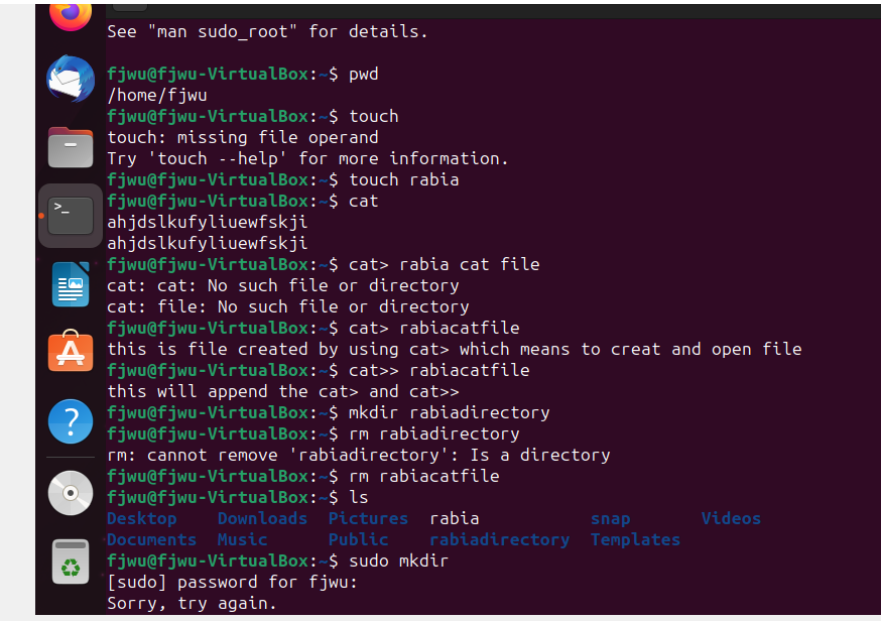

ubunto [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Your name
fjwu
Your computer's name
fjwu-VirtualBox
The name it uses when it talks to other computers.
Pick a username
fjwu
Choose a password
Show
Strong password
Confirm your password
Require my password to log in
Use Active Directory
You'll enter domain and other details in the next step.
Previous
Next
Type here to search
32°C
4:29 PM
10/2/2023


ubunto [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Choose your theme
You can always change this later in the appearance settings.
Light
Dark
Previous
Next
Type here to search
32°C
4:31 PM
10/2/2023

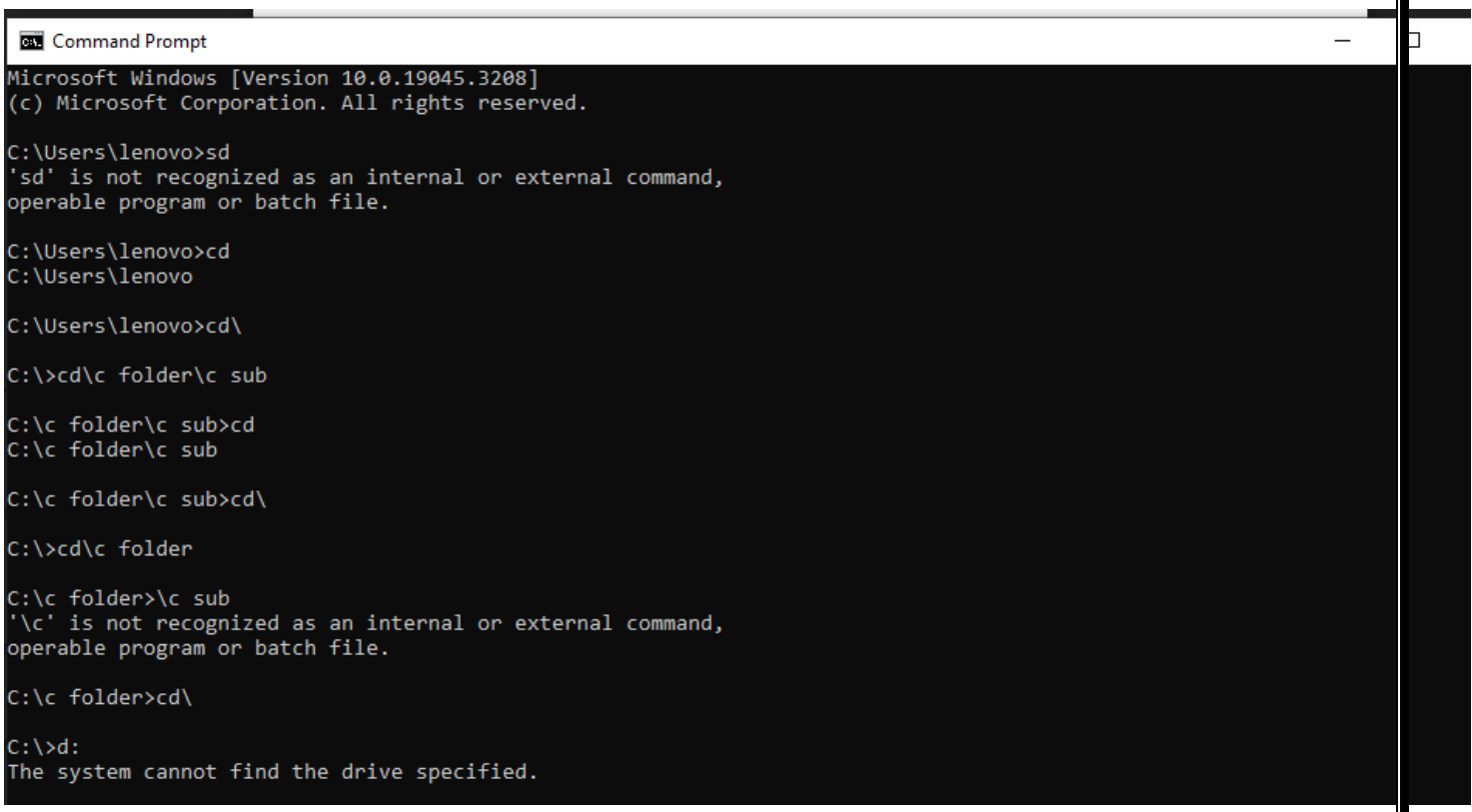
\



Commands of Ubuntu:



cmd coomands



Lab Title: lab 2	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator	Level of Achievements				
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		

Lab Instructor:

“lab 1”

COURSE :

OPERATING SYSTEM

SUBMITTED TO :

Sir shehzad

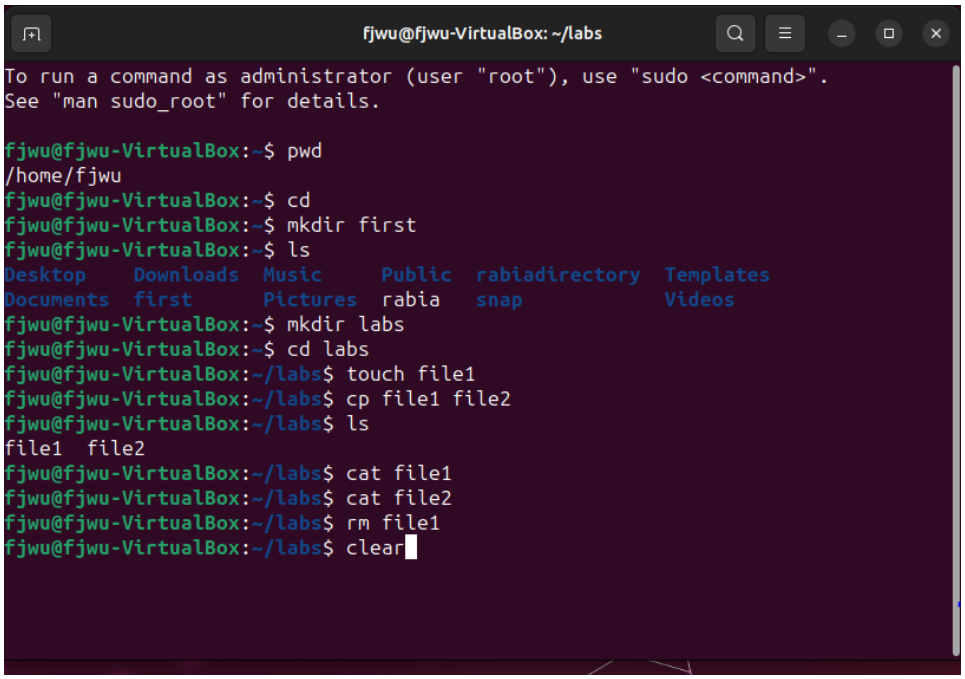
SUBMITTED BY :

Rabia Batool (2022-BSE-064)

SECTION :

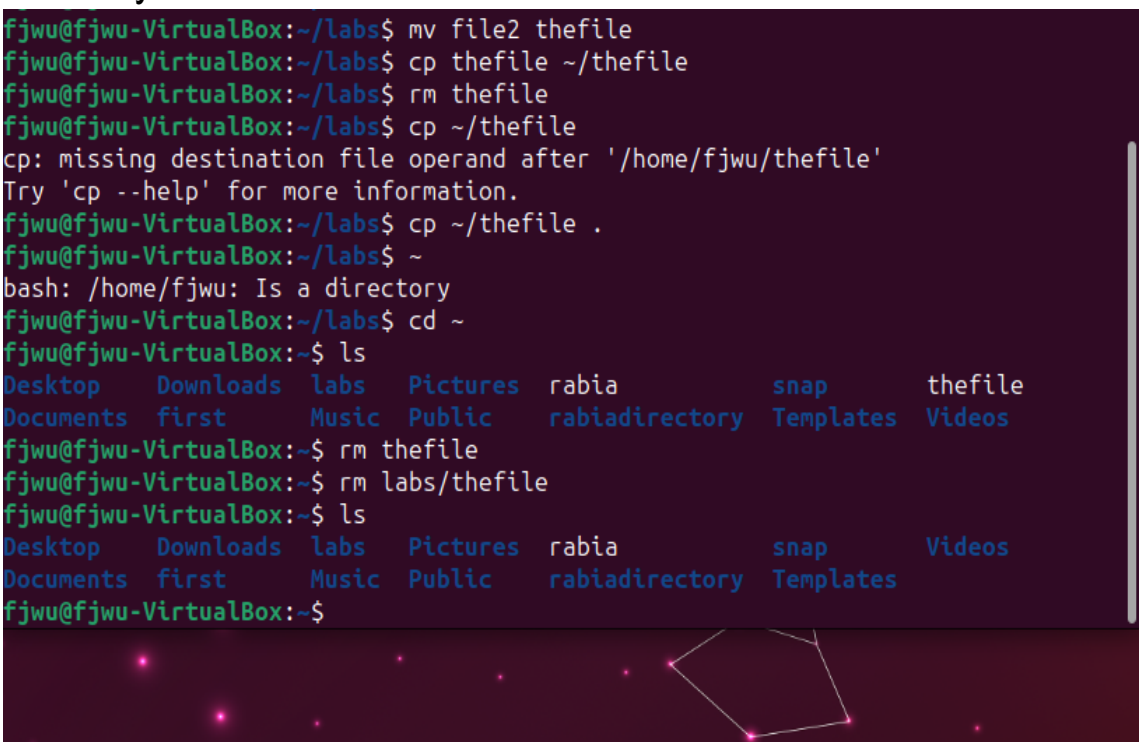
B

1. Verify that you are in your home directory.
2. Make the directory FIRST using the following command.
3. List the files in the current directory to verify that the directory FIRST has been made correctly.
4. Change directories to LABS. Create the file named file1.
5. List the contents of the file file1 to the screen.
6. Make a copy of the file file1 under the name file2.
7. Verify that the files file1 and file2 both exist.
8. List the contents of both file1 and file2 to the monitor screen.
9. Then delete the file file1 .
10. Clear the window.

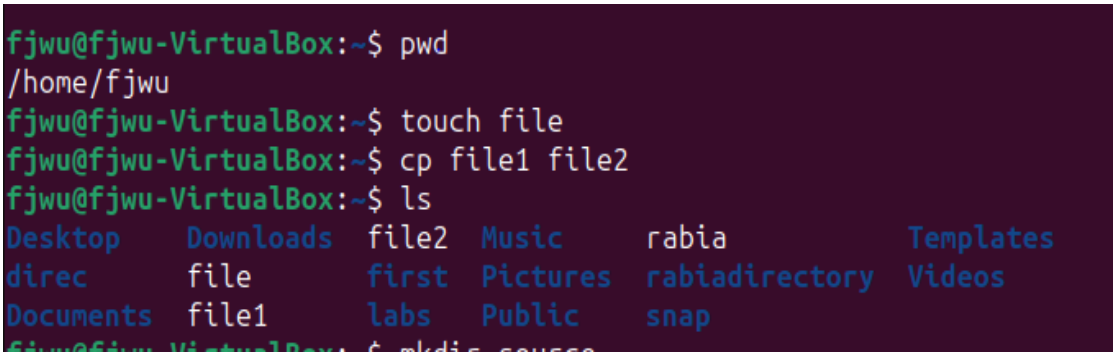


1. Copy thefile to your home directory.
2. Removethefile from the current directory.
3. Copy thefile from your home directory to the current directory.

4. Change directories to your home directory.
5. Remove thefile from your home directory and from directory LABS.
6. Verify thefile is removed from the directory LABS.
7. Remove the directory LABS from your home directory with the following command.
8. Verify that thefile and LABS are gone from your home directory.



1. cp **<file 1> <file 2>**



2. cp -r **<sourceDirectory> <destinationDirectory>**

```
fjwu@fjwu-VirtualBox:~$ mkdir source
fjwu@fjwu-VirtualBox:~$ mkdir destination
fjwu@fjwu-VirtualBox:~$ cp -r destination /home/fjwu/source/
fjwu@fjwu-VirtualBox:~$ cd source
fjwu@fjwu-VirtualBox:~/source$ ls
destination
```

3. cp *.**<extension> <destinationDirectory>**

```
fjwu@fjwu-VirtualBox:~/destination$ touch file1.txt
fjwu@fjwu-VirtualBox:~/destination$ cp *.txt /home/fjwu/source/
fjwu@fjwu-VirtualBox:~/destination$ cd source
bash: cd: source: No such file or directory
fjwu@fjwu-VirtualBox:~/destination$ ls
file1.txt
```

4. cp --backup **<filename> <destinationDirectory>**

```
fjwu@fjwu-VirtualBox:~/destination$ cp --backup file1.txt /home/fjwu/source/
fjwu@fjwu-VirtualBox:~/destination$ cd
fjwu@fjwu-VirtualBox:~$ cd source
fjwu@fjwu-VirtualBox:~/source$ ls
destination  file1.txt  file1.txt~
```

5. cp -i **<filename> <destinationDirectory>**

```
fjwu@fjwu-VirtualBox:~/destination$ cp -i file1.txt /home/fjwu/source/
cp: overwrite '/home/fjwu/source/file1.txt'? y
fjwu@fjwu-VirtualBox:~/destination$ cd
fjwu@fjwu-VirtualBox:~$ cd  source
fjwu@fjwu-VirtualBox:~/source$ ls
destination  file1.txt  file1.txt~
```

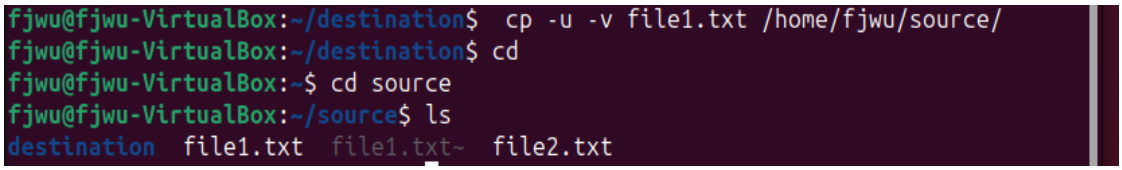
6. cp -l **<filename> <destinationDirectory>**

```
fjwu@fjwu-VirtualBox:~/destination$ touch file2.txt
fjwu@fjwu-VirtualBox:~/destination$ cp -l file2.txt /home/fjwu/source/
fjwu@fjwu-VirtualBox:~/destination$ cd
fjwu@fjwu-VirtualBox:~$ cd source
fjwu@fjwu-VirtualBox:~/source$ ls
destination  file1.txt  file1.txt~  file2.txt
```

7. cp -p **<filename> <destinationDirectory>**

```
fjwu@fjwu-VirtualBox:~/destination$ cp -p file1.txt /home/fjwu/source/
fjwu@fjwu-VirtualBox:~/destination$ cd
fjwu@fjwu-VirtualBox:~$ cd source
fjwu@fjwu-VirtualBox:~/source$ ls
destination  file1.txt  file1.txt~  file2.txt
```

8. cp -u -v **<filenames> <destinationDirectory>**
9.



Lab Title: lab 3	**Experiment # :**
Student Name : Rabia Batool	**Regd. #:**
Experiment Name:	

Performance Indicator	**Level of Achievements**

	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		**10**	**Obtained Marks**		

Lab Instructor:

Name :Rabia batool

Roll no:2022_BSE_052

LAB 3

Submitted to :

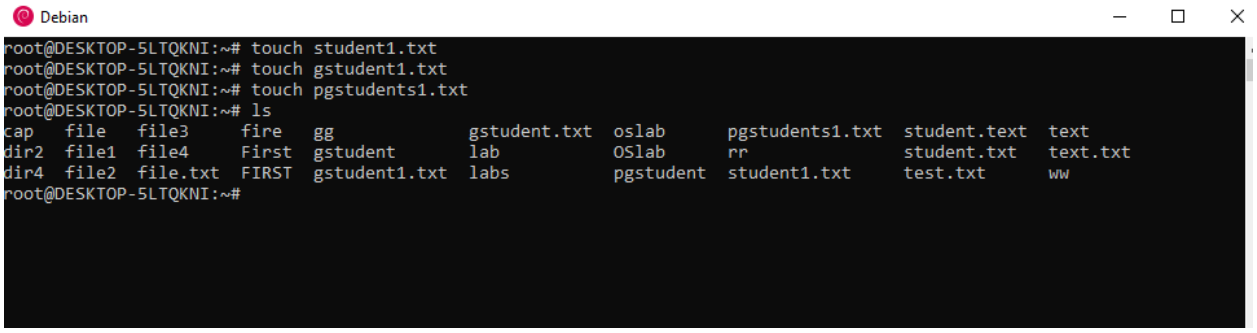
Sir shehzad

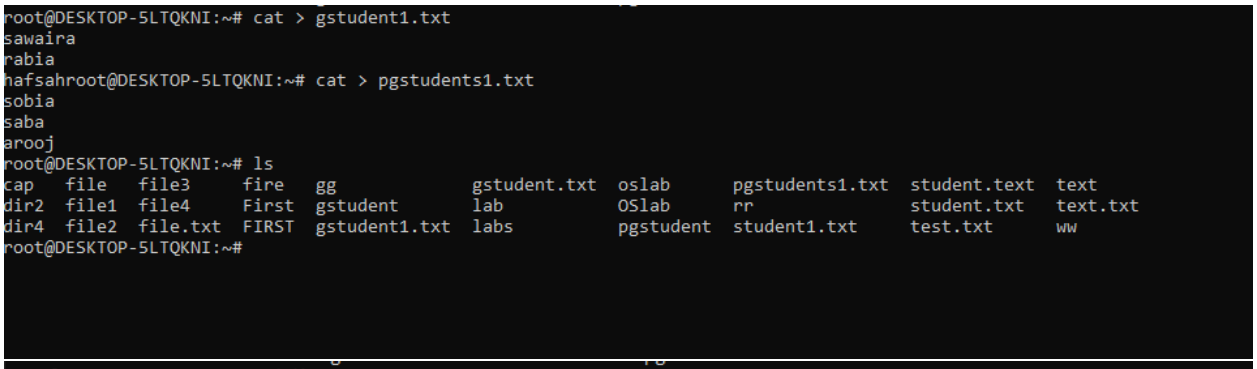
TASK 1

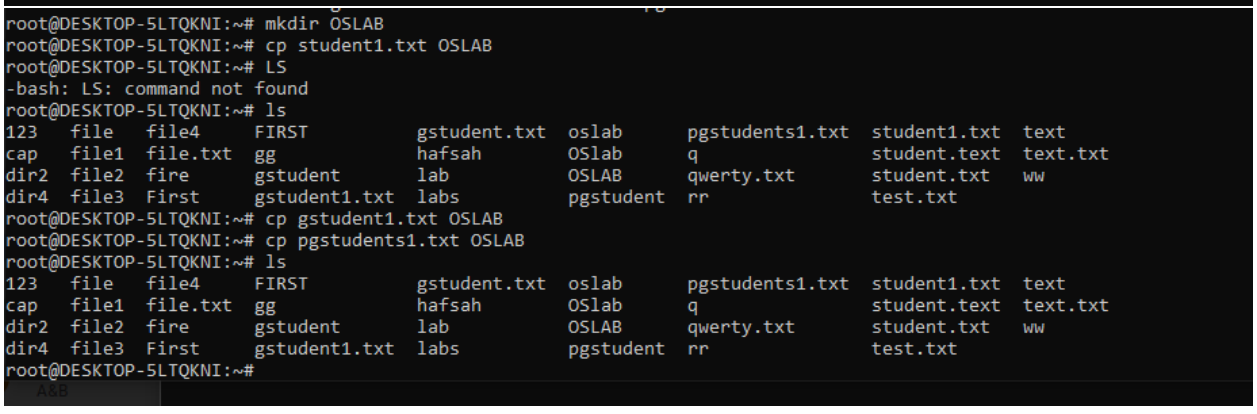
Create file name students.txt, gstudent.txt, pgstudents. Enter students' names in gstudent.txt and

pgstudent.txt. Now create directory having name OSLAB and copy all files to this directory. Now

append the file students.txt with first five sorted names from pstudent.txt and last five sorted names from gstudent.txt. Then show the contents of sorted names from file students.txt. Change the permissions of file students.txt read only and both other files read and execute only.

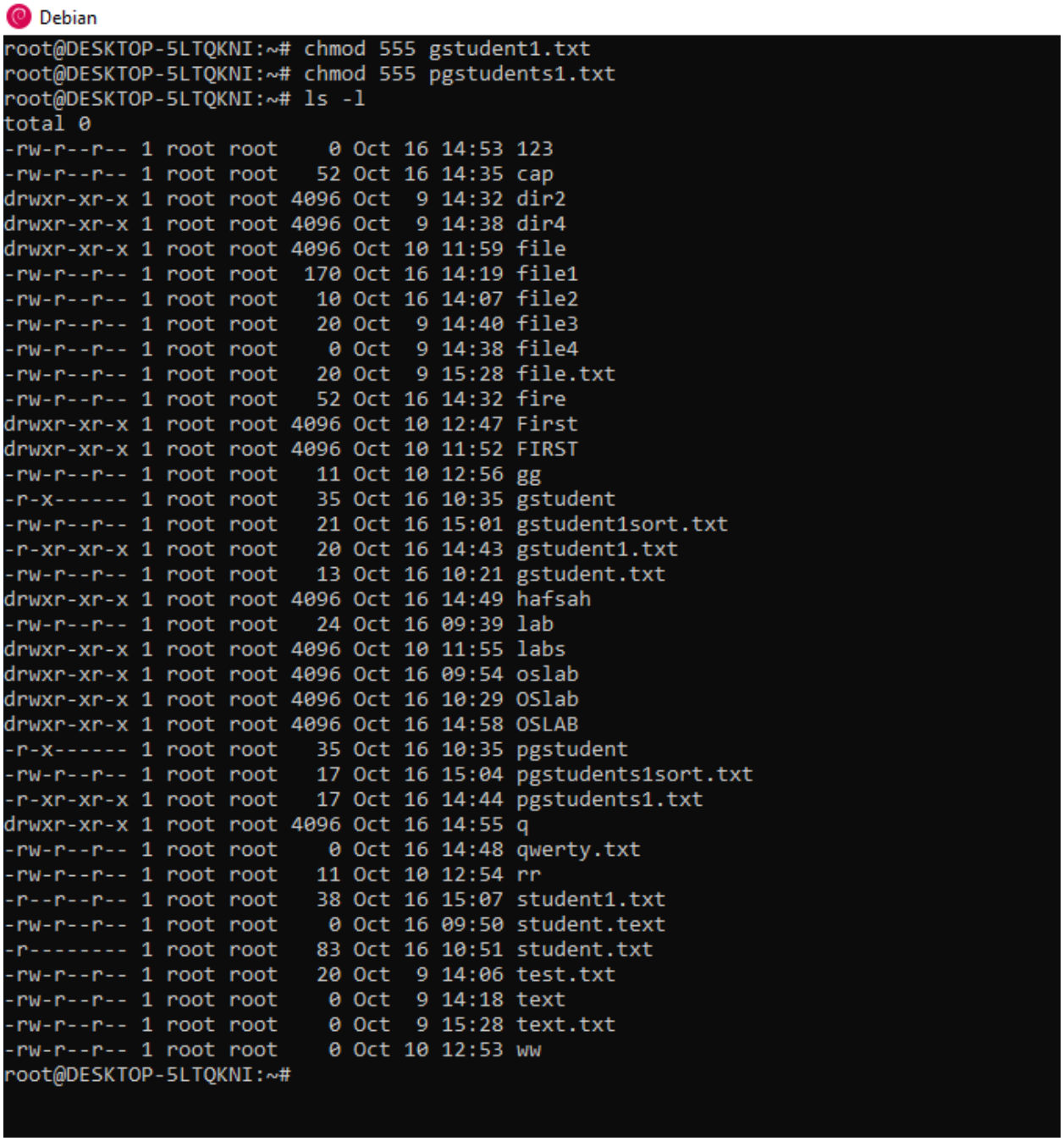






```
root@DESKTOP-5LTQKNI:~# sort <gstudent1.txt>gstudent1sort.txt
root@DESKTOP-5LTQKNI:~# cat gstudent1sort.txt
hafsah
rabia
sawaira
root@DESKTOP-5LTQKNI:~# sort <pgstudents1.txt>pgstudents1sort.txt
root@DESKTOP-5LTQKNI:~# cat pgstudents1sort.txt
arooj
saba
sobia
root@DESKTOP-5LTQKNI:~#
```

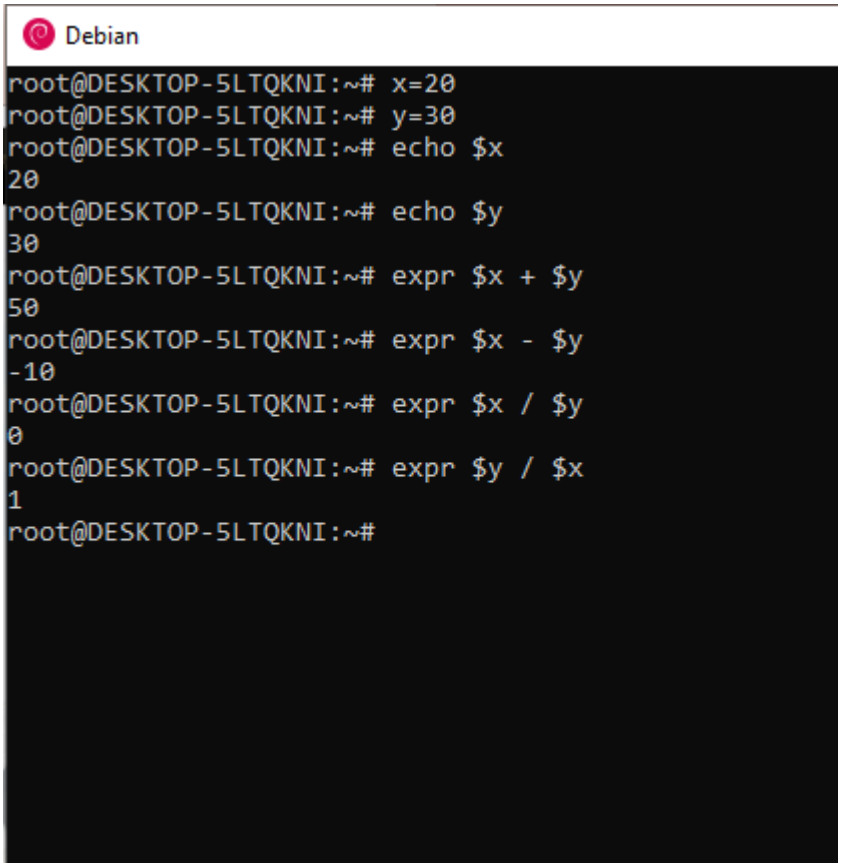
```
sobia
root@DESKTOP-5LTQKNI:~# head -n3 pgstudents1sort.txt>student1.txt
root@DESKTOP-5LTQKNI:~# tail -n3 gstudent1sort.txt>>student1.txt
root@DESKTOP-5LTQKNI:~# cat student1.txt
arooj
saba
sobia
hafsah
rabia
sawaira
root@DESKTOP-5LTQKNI:~#
```



Task #2

Define variable x and y with value 20 and 30 and print it on screen, then sum, subtract, divide

these numbers and print it on the screen.


Debian
root@DESKTOP-5LTQKNI:~# x=20
root@DESKTOP-5LTQKNI:~# y=30
root@DESKTOP-5LTQKNI:~# echo $x
20
root@DESKTOP-5LTQKNI:~# echo $y
30
root@DESKTOP-5LTQKNI:~# expr $x + $y
50
root@DESKTOP-5LTQKNI:~# expr $x - $y
-10
root@DESKTOP-5LTQKNI:~# expr $x / $y
0
root@DESKTOP-5LTQKNI:~# expr $y / $x
1
root@DESKTOP-5LTQKNI:~#

Lab Title: lab 4	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator	Level of Achievements				
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		

Lab Instructor:

Rabia Batool

2022-BSE-064

Group#B

Operating system

LAB#04

Submitted to Sir shahzad

Task 1:

Write shell script as follows:

```
cat > trmif
#
# Script to test rm command and exist status
#
if rm $1
then
echo "$1 file deleted"
fi
```

Press Ctrl + d to save

$ chmod 755 trmif

Answer the following question in reference to above script:

1. foo file exists on your disk and you give command, **$ ./trmfi foo** what will be output? 2. If bar file not present on your disk and you give command, **$ ./trmfi bar** what will be output? 3. And if you type **$ ./trmfi** What will be output?

Solution:

fjwu@DESKTOP-5LTQKNI: ~

```
Windows Subsystem for Linux is now available in the Microsoft Store!
You can upgrade by running 'wsl.exe --update' or by visiting https://aka.ms/wslstorepage
Installing WSL from the Microsoft Store will give you the latest WSL updates, faster.
For more information please visit https://aka.ms/wslstoreinfo

fjwu@DESKTOP-5LTQKNI:~$ cat > trmif
#! /bin/bash
fjwu@DESKTOP-5LTQKNI:~$ cat > trmif
if rm $1
then
echo "$1 file deleted"
fifjwu@DESKTOP-5LTQKNI:~$ chmod 755 trmif
fjwu@DESKTOP-5LTQKNI:~$ cat > foo
1020 we are in file foo
fjwu@DESKTOP-5LTQKNI:~$ ./trmif foo
foo file deleted
fjwu@DESKTOP-5LTQKNI:~$
```

```
fjwu@DESKTOP-5LTQKNI:~$ ./trmif bar
rm: cannot remove 'bar': No such file or directory
fjwu@DESKTOP-5LTQKNI:~$
```

```
fjwu@DESKTOP-5LTQKNI:~$ ./trmif
rm: missing operand
Try 'rm --help' for more information.
fjwu@DESKTOP-5LTQKNI:~$
```

Task 2:

Task 2:

Write a shell script that computes the gross salary of an employee according to the following:

1) if basic salary is <1500 then HRA=10% of the basic salary.
2) if basic salary is >1500 then HRA=20% of the basic salary.

Solution:

```
fjwu@fjwu-VirtualBox:~$ nano sal.sh
fjwu@fjwu-VirtualBox:~$ bash sal.sh
enter basic salary
1200
HRA=10% of basic salary:
```

```
fjwu@fjwu-VirtualBox:~$ nano sal.sh
fjwu@fjwu-VirtualBox:~$ bash sal.sh
enter basic salary
1900
sal.sh: line 10: echo HRA=20% of basic salar
sal.sh: line 11: echo gross salary=basic sala
fjwu@fjwu-VirtualBox:~$
```

```
  GNU nano 7.2                                  sal.sh
#!/bin/bash
echo "enter basic salary"
read a

if [ "$a" -lt  1500 ]
then
echo "HRA=10% of basic salary:"
if [ "$a" -gt 1500 ]
then
echo" HRA=20% of basic salar"
echo" gross salary=basic salary +HR"
fi
fi
```

Task 3

Task 3:

Write a shell script to ADD two numbers taken from argument?

Note: show error if no argument or more than 2 arguments are passed
$ ./ADD 5 6
Output : Sum of 5 and 6 = 5+6 = 11

```
fjwu@fjwu-VirtualBox:~$ nano add
fjwu@fjwu-VirtualBox:~$ bash add
enter two numbers
2
3
result
5
fjwu@fjwu-VirtualBox:~$
```

```
  GNU nano 7.2
#!/bin/bash
num1=0
num2=0
echo "enter two numbers"
read num1
read num2
echo "result"
expr $num1  + $num2
```

Task 4:

TASK#3:

Write a program that will displays the fibanocaii series of any given number. Take the limit from the user

```
echo "How many number of terms to be generated ?"
  read n
function fib
{
  x=0
  y=1
  i=2
  echo "Fibonacci Series up to $n terms :"
  echo "$x"
  echo "$y"
  while [ $i -lt $n ]
  do
       i=`expr $i + 1 `
       z=`expr $x + $y `
       echo "$z"
       x=$y
       y=$z
  done
}
r=`fib $n`
echo "$r"
```

```
fjwu@ubuntu:~$ sh fib.sh
How many number of terms to be generated ?
15
fib.sh: 3: fib.sh: function: not found
Fibonacci Series up to 15 terms :
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
```

Lab Title: lab 5	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator	Level of Achievements				
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		

Lab Instructor:

Rabia Batool

2022-BSE-064

Group#B

Operating system

LAB#05

Submitted to Sir shehzad

LAB Task

Q1. Write a program using fork() system call to create two child of the same process i.e., Parent P having child process P1 and P2. Q2. Write a program using fork() system call to create a hierarchy of 3 process such that P2 is the child of P1 and P1 is the child of P.

Code:

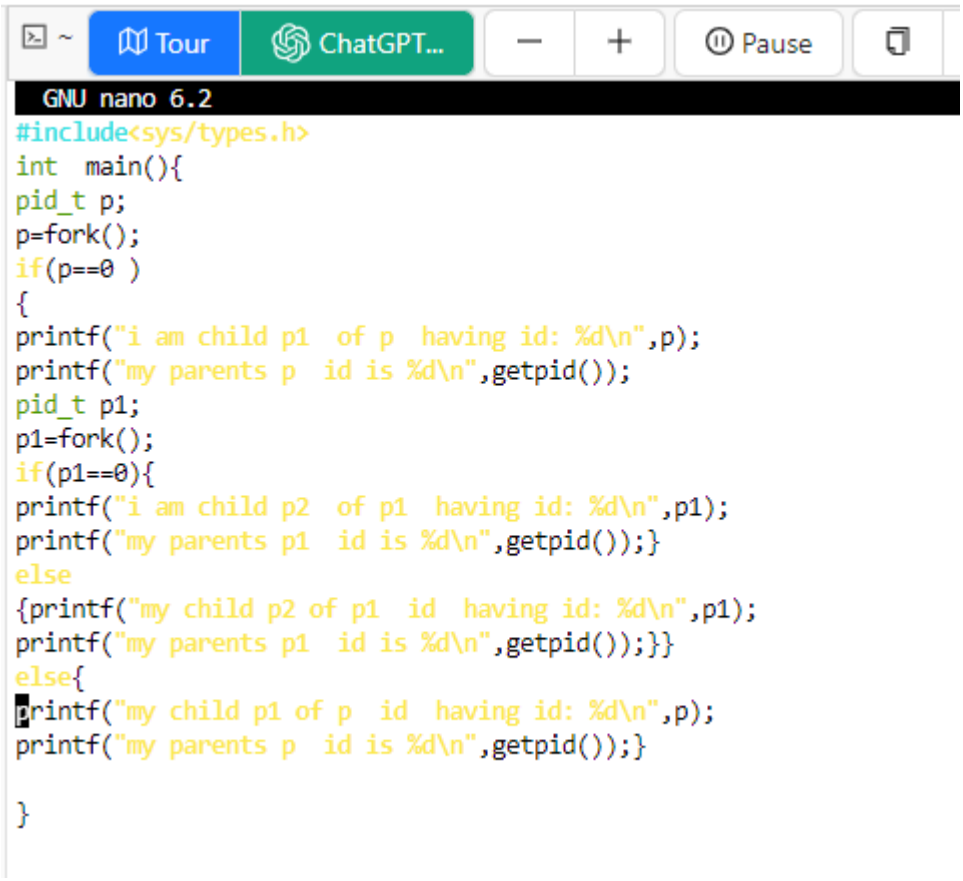
```
 GNU nano 6.2
//rabia.c
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
int  main(){
pid_t p1,p2;
p1=fork();
p2=fork();
if(p1==0 || p2==0)
{
printf("i am child 1  having id: %d\n",p1);
printf("i am child 2  having id: %d\n",p2);
printf("my parents id is %d\n",getpid());
}
else
{printf("my child 1 id  having id: %d\n",p1);
printf("my child 2  id  having id: %d\n",p2);
printf("my parents id is %d\n",getpid());}
}
```

output:

```
11300~$ nano rabia.c
~$ gcc rabia.c
~$ ./a.out
my child 1 id  having id: 1183
my child 2  id  having id: 1184
my parents id is 1182
i am child 1  having id: 0
i am child 2  having id: 1185
my parents id is 1183
i am child 1  having id: 1183
i am child 2  having id: 0
my parents id is 1184
i am child 1  having id: 0
i am child 2  having id: 0
my parents id is 1185
~$ gcc rabia.c
~$
```

LAB Task

Q1. Write a program using fork() system call to create two child of the same process i.e., Parent P having child process P1 and P2. Q2. Write a program using fork() system call to create a hierarchy of 3 process such that P2 is the child of P1 and P1 is the child of P.




```c
#include<sys/types.h>
int  main(){
pid_t p;
p=fork();
if(p==0 )
{
printf("i am child p1  of p  having id: %d\n",p);
printf("my parents p  id is %d\n",getpid());
pid_t p1;
p1=fork();
if(p1==0){
printf("i am child p2  of p1  having id: %d\n",p1);
printf("my parents p1  id is %d\n",getpid());}
else
{printf("my child p2 of p1  id  having id: %d\n",p1);
printf("my parents p1  id is %d\n",getpid());}}
else{
printf("my child p1 of p  id  having id: %d\n",p);
printf("my parents p  id is %d\n",getpid());}

}
```

Output:

```
~$ nano rabia.c
~$ gcc rabia.c
~$ ./a.out
my child p1 of p  id  having id: 1178
my parents p  id is 1177
i am child p1  of p  having id: 0
my parents p  id is 1178
my child p2 of p1  id  having id: 1179
my parents p1  id is 1178
i am child p2  of p1  having id: 0
my parents p1  id is 1179
```

Lab Title: lab 6	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator	Level of Achievements				
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		

Lab Instructor:

Rabia Batool

2022-BSE-064

Group#B

Operating system

LAB#06

Submitted to Sir shehzad

Viva questions:

Viva questions on wait() system call

Q1. Can we use wait() to make the child process wait for the parent process to finish?
Q2. What does the wait() system call return on success?

The wait() function in the context of process management is used by a parent process to wait for its child processes to complete. It is not typically used to make a child process wait for the parent process to finish. The wait() function is used to collect the exit status of a child process and perform cleanup, such as releasing resources associated with the child process.

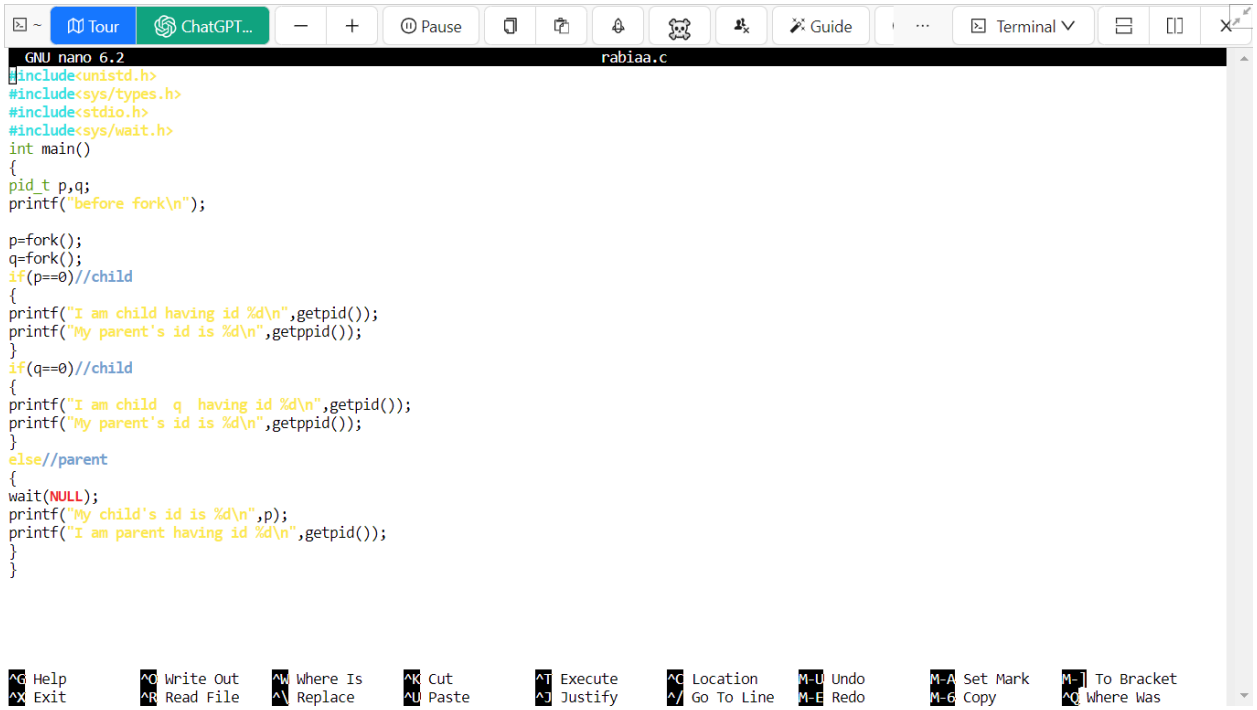
(2)

The wait() system call in Unix-like operating systems returns the process ID of the terminated child process on success. It allows the parent process to obtain information about the child process that has terminated, including its exit status.

Task#1

Q1. Write a program to create two child process. The parent process should wait for both the child to finish.

```
#include<unistd.h>
#include<sys/types.h>
#include<stdio.h>
#include<sys/wait.h>
int main()
{
pid_t p,q;
printf("before fork\n");

p=fork();
q=fork();
if(p==0)//child
{
printf("I am child having id %d\n",getpid());
printf("My parent's id is %d\n",getppid());
}
if(q==0)//child
{
printf("I am child  q  having id %d\n",getpid());
printf("My parent's id is %d\n",getppid());
}
else//parent
{
wait(NULL);
printf("My child's id is %d\n",p);
printf("I am parent having id %d\n",getpid());
}
}
```



Output:

```
~$ touch rabiaa.c
~$ nano rabiaa.c
~$ gcc rabiaa.c
~$ ./a.out
before fork
I am child  q  having id 1097
My parent's id is 1095
I am child having id 1096
My parent's id is 1095
My child's id is 1096
I am parent having id 1095
I am child having id 1098
My parent's id is 1096
I am child  q  having id 1098
My parent's id is 1096
~$ My child's id is 0
I am parent having id 1096
```

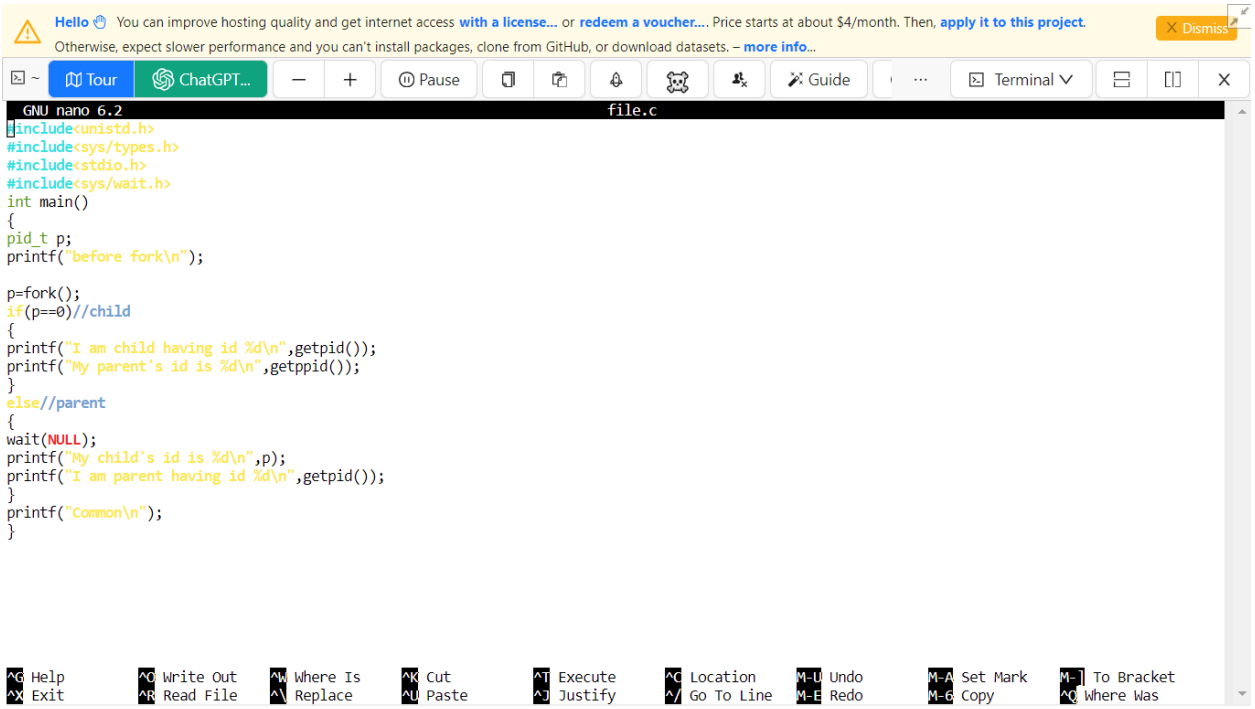
Task#2

Q2. Create a parent-child relationship between two process. The parent should print two statements:
A) its own PID
B) PID of its child
The child should print two statements:
C) its own PID
D) PID of its parent
Make use of wait() in such a manner that the order of the four statements A, B, C and D is:
A
C
D
B
You are free to use any other relevant statement/printf as you desire and their order of execution does not matter.

```
#include<unistd.h>

#include<sys/types.h>

#include<stdio.h>

#include<sys/wait.h>

int main()

{

pid_t p;

printf("before fork\n");


p=fork();

if(p==0)//child

{

printf("I am child having id %d\n",getpid());

printf("My parent's id is %d\n",getppid());

}

else//parent

{

wait(NULL);

printf("My child's id is %d\n",p);

printf("I am parent having id %d\n",getpid());

}
```

printf("Common\n");

}



Output:

```
~$ touch file.c
~$ nano file.c
~$ gcc file.c
~$ ./a.out
before fork
I am child having id 1940
My parent's id is 1939
Common
My child's id is 1940
I am parent having id 1939
Common
~$
```

Lab Title: lab 7	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator	Level of Achievements				
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		

Lab Instructor:

Operating system

Lab # 07

Submitted by:

Sawaira Saeed

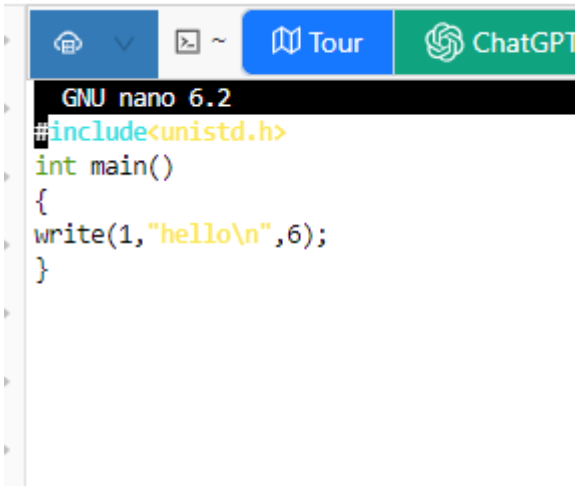
2022-BSE-067

Submitted to:

Sir Shahzad

Practice:

```
~$ touch sawaira.c
~$ nano sawaira.c
~$ touch lab7.c
~$ nano lab7.c
~$ gcc lab7.c
~$ ./a.out
hello
~$
```



```
~$ touch sawaira.c
~$ nano sawaira.c
~$ touch lab7.c
~$ nano lab7.c
~$ gcc lab7.c
~$ ./a.out
hello
~$ nano lab7.c
~$ touch prp.c
~$ nano prp.c
~$ gcc prp.c
~$ ./a.out
hello
Total bytes written: 6
~$
```

```
  GNU nano 6.2
#include<stdio.h>
#include<unistd.h>
int main()
{
int count;
count=write(1,"hello\n",6);
printf("Total bytes written: %d\n",count);
}
```

Error:

```
Total bytes written: 6
~$ nano prp.c
~$ touch err.c
~$ nano err.c
~$ gcc err.c
err.c: In function ‘main’:
err.c:5:1: error: ‘parameter’ undeclared (first use in this function)
    5 | parameter
      | ^~~~~~~~~
err.c:5:1: note: each undeclared identifier is reported only once for each function it appears in
err.c:5:10: error: expected ‘;’ before ‘}’ token
    5 | parameter
      |          ^
      |          ;
    6 | }
      | ~
~$
```

```
 GNU nano 6.2
#include<unistd.h>
int main()
{
write(1,"hello\n",60);
parameter
}
```

```
      |
      |          ;
    6 | }
      | ~
~$ nano err1.c
~$ gcc err1.c
~$ ./a.out
hel~$
```

```
 GNU nano 6.2
#include<unistd.h>
int main()
{
write(1,"hello\n",3);
}
```

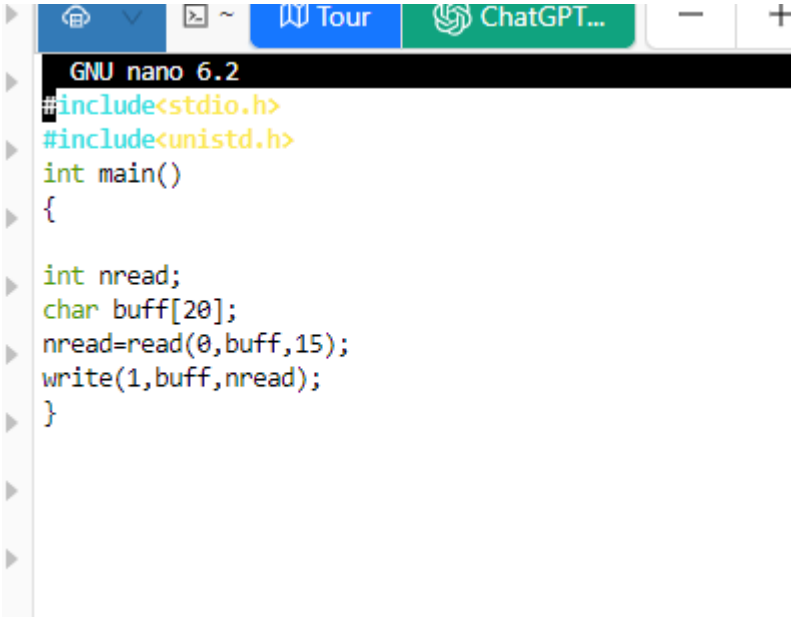
```
~$ nano err1.c
~$ gcc err1.c
~$ ./a.out
hel~$ nano err1.c
~$ touch err2.c
~$ nano err2.c
~$ gcc err2.c
~$ ./a.out
Total bytes written: -1
~$
```

```
 GNU nano 6.2
#include<unistd.h>
#include<stdio.h>

int main()
{
int count;
count=write(3,"hello\n",6);
printf("Total bytes written: %d\n",count);
}
```

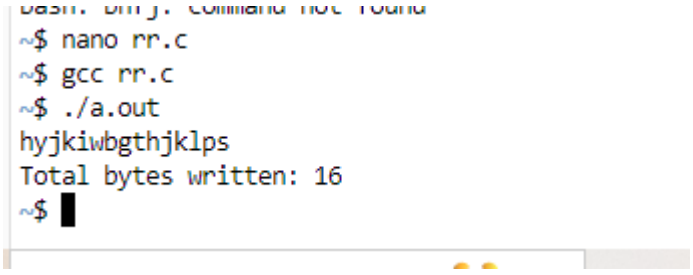
Practice Programs on write()/read() system call

Q1. Write a program to read a maximum of 15 characters from the user and print them on the screen.

```
~$ touch tt.c
~$ nano tt.c
~$ gcc tt.c
~$ ./.aout
bash: ./.aout: No such file or directory
~$ ./a.out
qwertyuiopasddfgg
qwertyuiopasddf~$ gg
bash: gg: command not found
~$
```



Q2. Write a program to print the count of characters read by the read() system call.



```
 GNU nano 6.2
#include<unistd.h>
#include<stdio.h>
int main()
{
int count;
char buff[50];
count=read(0,buff,30);
printf("Total bytes written: %d\n",count);
}
```

Practice Program on open() system call

Q1. Write a program to read the contents of file F1 into file F2. The contents of file F2 should not get deleted or overwritten.

hint: use O_APPEND flag

File1.txt:

```
 GNU nano 6.2
this is file1
```

File2.txt:

```
this is file2
```

File.c:

```
 GNU nano 6.2
#include<unistd.h>
 #include<sys/types.h>
 #include<sys/stat.h>
#include<fcntl.h>
int main()
 { int n,fd,fd1;
 char buff[50];
 fd=open("file1.txt",O_RDONLY);
fd1=open("file2.txt",O_WRONLY|O_APPEND);
 n=read(fd,buff,10);
 write(fd1,buff,n);
return 0;}
```

#include<unistd.h>

```
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
int main()
{ int n,fd,fd1;
char buff[50];
fd=open("file1.txt",O_RDONLY);
fd1=open("file2.txt",O_WRONLY|O_APPEND);
n=read(fd,buff,10); write(fd1,buff,n);}
```

output:

```
~$ cat file1.txt
this is file1
~$ cat file2.txt
this is file2

~$ gcc file2.c
~$ ./a.out
~$ cat file2.txt
this is file2

~$ nano file2.c
~$ gcc file2.c
~$ ./a.out
~$ cat file2.txt
this is file2

this is fi~$
```

Q2. Write a program using open() system call to copy the contents of one file into another file.

Read.txt:

```
  GNU nano 6.2
this is file read.txt
```

Write.txt:

```
 GNU nano 6.2
this is file write.txt
```

Read.c:

```
 GNU nano 6.2
#include<unistd.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
int main()
{
int n,fd,fd1;
char buff[50];
fd=open("read.txt",O_RDONLY);
n=read(fd,buff,10);
fd1=open("write.txt",O_WRONLY|O_CREAT,0777);
write(fd1,buff,n);
}
```

GNU nano 6.2

read.c

```
#include<unistd.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
int main()
{
int n,fd,fd1;
char buff[50];
fd=open("read.txt",O_RDONLY);
n=read(fd,buff,10);
fd1=open("write.txt",O_WRONLY|O_CREAT,0777);
write(fd1,buff,n);
}
```

Output:

```
~$ cat>>read.txt
this is file read.txt
~$ touch read.c
~$ nano read.c
~$ cat>>write.txt
this is file write.txt~$ gcc read.c
~$ ./a.out
~$ cat write.txt
this is file write.txt~$ nano read.txt
```

Practice Programs on lseek() system call:

Pre-requisite: Create a file “seeking” and write

“1234567890abcdefghijxxxxxxxxxx” into it.

Program1: Program using lseek() system call that reads 10 characters from file

“seeking” and print on screen. Again read 10 characters and write on screen.

Seek.txt :

```
 GNU nano 6.2
1
2
3
4
5
6
7
8
9
0
a
b
c
d
e
f
g
h
i
j

*
**
*
*
```

Seek.c:

```
 GNU nano 6.2
#include<unistd.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
int main()
{
int n,f;
char buff[10];
f=open("seek.txt",O_RDWR);
read(f,buff,10);
write(1,buff,10);
read(f,buff,10);
write(1,buff,10);
}
```

include<unistd.h>

#include<fcntl.h>

#include<sys/types.h>

#include<sys/stat.h>

```
int main()
{
int n,f;
char buff[10];
f=open("seek.txt",O_RDWR);
read(f,buff,10);
write(1,buff,10);
read(f,buff,10);
write(1,buff,10);
}
```

Output:

```
*~$ touch seek.c
~$ nano seek.c
~$ gcc seek.c
~$ ./a.out
1
2
3
4
5
6
7
8
9
0
```

Program2: Program using lseek() system call that reads 10 characters from file

"seeking" and print on screen. Skip next 5 characters and again read 10

characters and write on screen.

Seek.txt:

```
 GNU nano 6.2
1
2
3
4
5
6
7
8
9
0
a
b
c
d
e
f
g
h
i
j

*
**
*
*
```

Seek2.txt:

```
 GNU nano 6.2
#include<unistd.h>
#include<fcntl.h>
#include<sys/types.h>
#include<sys/stat.h>
int main()
{
int n,f;
char buff[10];
f=open("seek.txt",O_RDWR);
read(f,buff,10);
write(1,buff,10);
lseek(f,5,SEEK_CUR);//skips 5 characters from the current position
read(f,buff,10);
write(1,buff,10);
}
```

#include<unistd.h>

#include<fcntl.h>

```
#include<sys/types.h>
#include<sys/stat.h>
int main()
{
int n,f;
char buff[10];
f=open("seek.txt",O_RDWR);
read(f,buff,10);
write(1,buff,10);
lseek(f,5,SEEK_CUR);//skips 5 characters from the current position
read(f,buff,10);
write(1,buff,10);
}
```

Output:

```
~$ touch seek2.c
~$ nano seek2.c
~$ gcc seek2.c
~$ ./a.out
1
2
3
4
5

9
0
a
b
c~$ nano seek2.c
```

Lab Title: lab 8	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator	Level of Achievements				
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					

Report and presentation					
Maximum Marks		**10**	**Obtained Marks**		
Lab Instructor:					

Rabia batool

2022-BSE-064

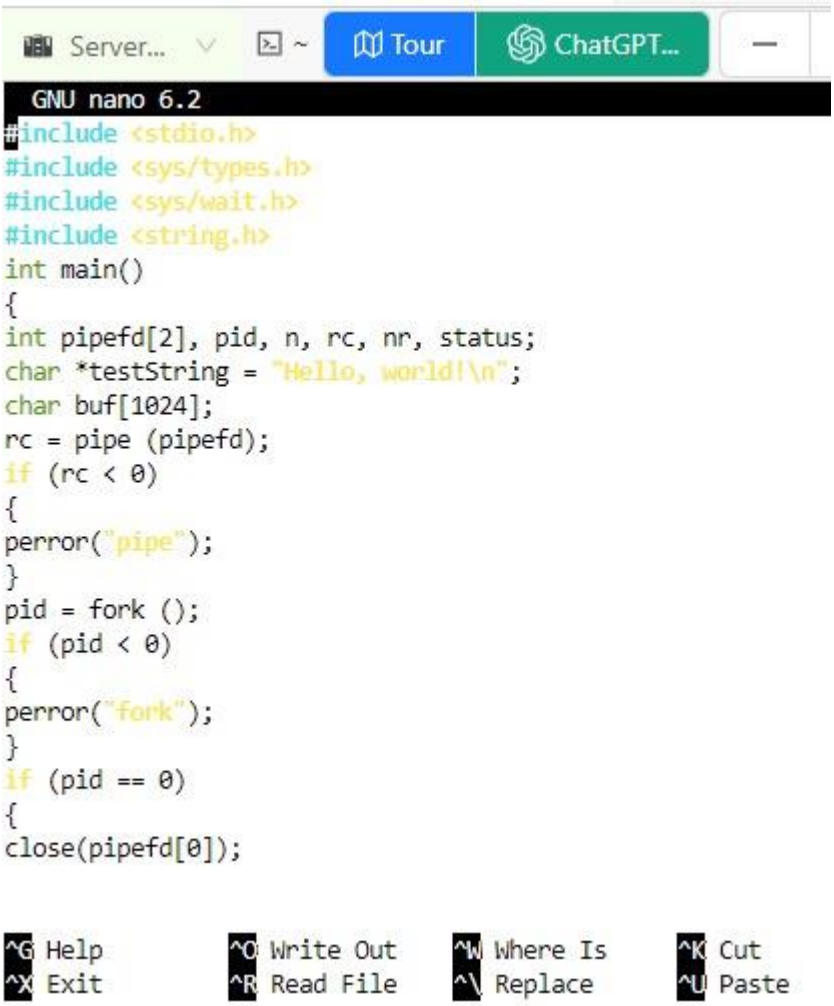
Group#B

Operating System

Lab#08

Submitted to Sir Shehzad

Practice 1


Server...
Tour
ChatGPT...
GNU nano 6.2
#include <stdio.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <string.h>
int main()
{
int pipefd[2], pid, n, rc, nr, status;
char *testString = "Hello, world!\n";
char buf[1024];
rc = pipe (pipefd);
if (rc < 0)
{
perror("pipe");
}
pid = fork ();
if (pid < 0)
{
perror("fork");
}
if (pid == 0)
{
close(pipefd[0]);
^G Help
^X Exit
^O Write Out
^R Read File
^W Where Is
^\ Replace
^K Cut
^U Paste

```
  GNU nano 6.2
{
perror("fork");
}
if (pid == 0)
{
close(pipefd[0]);

write(pipefd[1], testString, strlen(testString));
close(pipefd[1]);
}
else
{
close(pipefd[1]);
n = strlen(testString);
nr = read(pipefd[0], buf, n);
rc = write(1, buf, nr);
wait(&status);
printf("End!\n");
}
return(0);

}
```

```
      |      fread
~$ ./a.out
Hello, world!
End!
```

Practice 2

```c
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
int main()
{
int fd[2],n;
char buffer[100];
pid_t p;
pipe(fd); //creates a unidirectional pipe with two end fd[0] and fd[1]
p=fork();
if(p>0) //parent
{
printf("Parent Passing value to child\n");
write(fd[1],"hello\n",6); //fd[1] is the write end of the pipe

}
else // child
{
printf("Child printing received value\n");
n=read(fd[0],buffer,100); //fd[0] is the read end of the pipe
write(1,buffer,n);
```

```
~$ nano practice.c
~$ gcc practice.c
~$ ./a.out
Parent Passing value to child
Child printing received value
hello
~$
```

Viva Questions on Program for IPC using pipe() function

Q1. Which kind of data channel is created by pipe() system call: Unidirectional or bidirectional?

ANS:Unidirectional.

Q2. What does the pipe() system call return on success?

ANS: The pipe() system call returns 0 (zero) on success. This indicates that the pipe was created successfully and the two file descriptors are ready to be used for reading and writing.

Q3. What does the pipe() system call return on failure?

ANS: the pipe() system call returns -1 (negative one) on failure. This indicates that the pipe creation was unsuccessful, and you cannot proceed with using it for data transfer.

Q4. Why fork() is used in the above program?

ANS: To create child process.

Q5. Which process (parent or child) in the above code, is using the writing end of the pipe?

ANS: The parent process is using the writing end of the pipe.

TASK 1

o Compute the Factorial of a number using IPC (PIPE implementation).

o Parent creates pipe

o Forks a child

o Parent writes into pipe (the number whose factorial is to be calculated, take the number

from the user)

o Child reads from pipe and compute the Factorial of a number written by Parent

```
 GNU nano 6.2
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

int main()
{
    int fd[2], number, fact = 1, i;
    pid_t p;

    p = fork();

    if (p == 0)
    {
        close(fd[1]);
        read(fd[0], &number, sizeof(number));
        printf("Number received from parent: %d\n", number);
        close(fd[0]);

        for (i = 1; i <= number; i++)
        {
            fact = fact * i;

          fact = fact * i;
       }

       printf("Factorial of %d is: %d\n", number, fact);
   }
   else
   {

       printf("Enter a number: ");
       scanf("%d", &number);
       close(fd[0]);
       write(fd[1], &number, sizeof(number));
       close(fd[1]);
       wait(NULL);
   }

   return 0;
}
```

```
~$ touch file.c
~$ nano file.c
~$ gcc file.c
~$ ./a.out
Enter a number: 7
Number received from parent: 7
Factorial of 7 is: 5040
~$
```

TASK 2

Using pipes, parent read data from one file, and child write data into another file.

```
 GNU nano 6.2
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <fcntl.h>

int main()
{
    int fd[2], fd1, fd2, m;
    char buf[100];
    pid_t p;

    if (pipe(fd) == -1) {
        perror("pipe");
        return 1;
    }

    p = fork();

    if (p > 0)
    {
        close(fd[0]);
        fd1 = open("file1.txt", O_RDONLY);
        if (fd1 == -1) {
            perror("open");
            return 1;
        }

        m = read(fd1, buf, sizeof(buf));
        printf("this is read from parent process:\n%s\n", buf);
        wait(NULL);
    }
    else
    {
        close(fd[1]);
        fd2 = open("file2.txt", O_WRONLY | O_CREAT | O_APPEND, 0642);
        if (fd2 == -1) {
            perror("open");
            return 1;
        }

        write(fd2, buf, m);
        close(fd2);
    }
```

Output:

```
~$ ls
a.out  file1.txt  file2.txt  l.c  laab1.c  task.c  test
~$ cat task.c
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <fcntl.h>

int main()
{
    int fd[2], fd1, fd2, m;
    char buf[100];
    pid_t p;

    if (pipe(fd) == -1) {
        perror("pipe");
        return 1;
    }

    p = fork();

    if (p > 0)
    {
        close(fd[0]);
        fd1 = open("file1.txt", O_RDONLY);
        if (fd1 == -1) {
```

```
    if (p > 0)
    {
        close(fd[0]);
        fd1 = open("file1.txt", O_RDONLY);
        if (fd1 == -1) {
            perror("open");
            return 1;
        }

        m = read(fd1, buf, sizeof(buf));
        printf("this is read from parent process:\n%s\n", buf);
        wait(NULL);
    }
    else
    {
        close(fd[1]);
        fd2 = open("file2.txt", O_WRONLY | O_CREAT | O_APPEND, 0642);
        if (fd2 == -1) {
            perror("open");
            return 1;
        }

        write(fd2, buf, m);
        close(fd2);
    }
```

```
            return 1;
        }

        m = read(fd1, buf, sizeof(buf));
        printf("this is read from parent process:\n%s\n", buf);
        wait(NULL);
    }
    else
    {
        close(fd[1]);
        fd2 = open("file2.txt", O_WRONLY | O_CREAT | O_APPEND, 0642);
        if (fd2 == -1) {
            perror("open");
            return 1;
        }

        write(fd2, buf, m);
        close(fd2);
    }

    return 0;
}
~$ gcc task.c
~$ ./a.out
this is read from parent process:

~$
```

Lab Title: lab 9	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator	Level of Achievements				
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		

Lab Instructor:

Rabia batool

2022-BSE-064

Group#B

Operating System

Lab#09

Submitted to Sir Shehzad

Practice program:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/shm.h>
#include<string.h>
int main()
{
int i;
void *shared_memory;
char buff[100];
int shmid;

shmid=shmget((key_t)2345, 1024, 0666|IPC_CREAT);
printf("Key of shared memory is %d\n",shmid);
shared_memory=shmat(shmid,NULL,0);
printf("Process attached at %p\n",shared_memory);
printf("Enter some data to write to shared memory\n");
read(0,buff,100);
strcpy(shared_memory,buff);
printf("You wrote : %s\n",(char *)shared_memory);
return  0;
```

```
 GNU nano 6.2
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/shm.h>
#include<string.h>
int main()
{
int i;
void *shared_memory;
char buff[100];
int shmid;

shmid=shmget((key_t)2345, 1024, 0666|IPC_CREAT);
printf("Key of shared memory is %d\n",shmid);
shared_memory=shmat(shmid,NULL,0);
printf("Process attached at %p\n",shared_memory);
printf("Enter some data to write to shared memory\n");
read(0,buff,100);
strcpy(shared_memory,buff);
printf("You wrote : %s\n",(char *)shared_memory);
return  0;
```

```
~$ gcc sm.c
~$ ./a.out
Key of shared memory is 0
Process attached at 0x7f15ebb41000
Enter some data to write to shared memory
hafsa asad
You wrote : hafsa asad
```

Practice program 2:

GNU nano 6.2 smh.c

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

```
#include<sys/shm.h>
#include<string.h>
int main()
{
int i;
void *shared_memory;
char buff[100];
int shmid;
shmid=shmget((key_t)2345, 1024, 0666);
printf("Key of shared memory is %d\n",shmid);
shared_memory=shmat(shmid,NULL,0);
printf("Process attached at %p\n",shared_memory);
printf("Data read from shared memory is : %s\n",(char
*)shared_memory);
}
```

```
 GNU nano 6.2
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

#include<sys/shm.h>
#include<string.h>
int main()
{
int i;
void *shared_memory;
char buff[100];
int shmid;
shmid=shmget((key_t)2345, 1024, 0666);
printf("Key of shared memory is %d\n",shmid);
shared_memory=shmat(shmid,NULL,0);
printf("Process attached at %p\n",shared_memory);
printf("Data read from shared memory is : %s\n",(char
*)shared_memory);
}
```

```
~$ touch smh.c
~$ nano smh.c
~$ gcc smh.c
~$ ./a.out
Key of shared memory is 0
Process attached at 0x7f25e9cbb000
Data read from shared memory is : hafsa asad

~$
```

VIVA QS:

Q1. What does shmget() function return on success?

On success, the `shmget()` function returns a non-negative integer, which is a **shared memory identifier (shmid)**. This identifier is unique within the system and can be used by other processes to access the shared memory segment.

Q2. What does shmat() function return on success?

On success, the `shmat()` function returns the **address of the attached shared memory segment** in the calling process's address space. This is a pointer that can be used to access the shared memory directly.

Q3. The value of segment size in shmget() is a round-up value to a multiple of ____.

The value of segment size in `shmget()` is a round-up value to a multiple of the **system page size**.2^n

Q4. Can a process create two shared segment? Will there returned identifiers be same or

different?

Yes, a process can create two shared segments. In fact, a process can create any nUmber of shared segments, as long as it has sufficient resources and permissions.The identifiers returned by `shmget()` for each shared segment will be different. This is because each shared segment is unique and has its own control structure in the kernel. The identifier acts as a reference to the shared segment and is used by other processes to access it

Q5. Which function can be used to detach the shared segment from the address space of the process?

The function used to detach a shared segment from the address space of the process is **shmdt()**.

Task 1

```
#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/shm.h>

#include<string.h>

int main()

{pid_t p;

int i;

void *shared_memory;

char buff[100];

int shmid;

p=fork();

if(p==0)

{
```

```
int i;
void *shared_memory;
char buff[100];
int shmid;
printf("this is child \n");
shmid=shmget((key_t)2345, 1024, 0666);
printf("Key of shared memory is %d\n",shmid);
shared_memory=shmat(shmid,NULL,0);
printf("Process attached at %p\n",shared_memory);
printf("Data read from shared memory is : %s\n",(char
*)shared_memory); }
else{
printf("this is parent  \n");
shmid=shmget((key_t)2048, 1024, 0666|IPC_CREAT);
printf("Key of shared memory is %d\n",shmid);
shared_memory=shmat(shmid,NULL,0);
printf("Process attached at %p\n",shared_memory);
printf("Enter some data to write to shared memory\n");
read(0,buff,100);
strcpy(shared_memory,buff);
printf("You wrote : %s\n",(char *)shared_memory);}
return  0;
}
```

```
 GNU nano 6.2
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/shm.h>
#include<string.h>
int main()
{pid_t p;
int i;
void *shared_memory;
char buff[100];
int shmid;
p=fork();
if(p==0)
{
int i;
void *shared_memory;
char buff[100];
int shmid;
printf("this is child \n");
shmid=shmget((key_t)2345, 1024, 0666);
printf("Key of shared memory is %d\n",shmid);
shared_memory=shmat(shmid,NULL,0);
printf("Process attached at %p\n",shared_memory);
printf("Data read from shared memory is : %s\n",(char
*)shared_memory); }

else{
printf("this is parent  \n");
shmid=shmget((key_t)2048, 1024, 0666|IPC_CREAT);
printf("Key of shared memory is %d\n",shmid);
shared_memory=shmat(shmid,NULL,0);
printf("Process attached at %p\n",shared_memory);
printf("Enter some data to write to shared memory\n");
read(0,buff,100);
strcpy(shared_memory,buff);
printf("You wrote : %s\n",(char *)shared_memory);}
return  0;
}
```

Output :

```
~$ nano lab.c
~$ gcc lab.c
~$ ./a.out
this is parent
Key of shared memory is 0
this is child
Process attached at 0x7f59499bb000
Enter some data to write to shared memory
Key of shared memory is -1
Process attached at 0xffffffffffffffff
this is lab 9
You wrote : this is lab 9
```

Lab Title: lab 10	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator					
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		
Lab Instructor:					

Operating system

Lab # 10

Submitted to:

Sir Shahzad

Submitted by:

Rabia batool

2022-BSE-064

III 'B'

Practice program:

1.

```
~$ gcc p1.c
~$ ./a.out
Inside Thread
0
1
2
3
4
Inside Main Program
20
21
22
23
24
~$
```

```
 GNU nano 6.2
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
void *thread_function(void *arg);
 int i,j;
 int main() {
 pthread_t a_thread;
 pthread_create(&a_thread, NULL, thread_function, NULL);
 pthread_join(a_thread, NULL);
 printf("Inside Main Program\n");
 for(j=20;j<25;j++)
 {
 printf("%d\n",j);
 sleep(1);
 }
 }
void *thread_function(void *arg) {
 printf("Inside Thread\n");
 for(i=0;i<5;i++)
 {
 for(j=20;j<25;j++)
 {
 printf("%d\n",j);
 sleep(1);
 }
 }
void *thread_function(void *arg) {
 printf("Inside Thread\n");
 for(i=0;i<5;i++)
 {
 printf("%d\n",i);
 sleep(1);
 }
 }
```

When no join:

```
~$ nano p1.c
~$ gcc p1.c
~$ ./a.out
Inside Main Program
Inside Thread
0
20
1
21
2
22
3
23
4
24
~$
```

```
  GNU nano 6.2
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
void *thread_function(void *arg);
 int i,j;
 int main() {
 pthread_t a_thread;
 pthread_create(&a_thread, NULL, thread_function, NULL);
 printf("Inside Main Program\n");
 for(j=20;j<25;j++)
 {
 printf("%d\n",j);
 sleep(1);
 }
 }
void *thread_function(void *arg) {
 printf("Inside Thread\n");
 for(i=0;i<5;i++)
 {
 printf("%d\n",i);
```

```
  GNU nano 6.2
void *thread_function(void *arg);
 int i,j;
 int main() {
 pthread_t a_thread;
 pthread_create(&a_thread, NULL, thread_function, NULL);
 printf("Inside Main Program\n");
 for(j=20;j<25;j++)
 {
 printf("%d\n",j);
 sleep(1);
 }
 }
void *thread_function(void *arg) {
 printf("Inside Thread\n");
 for(i=0;i<5;i++)
 {
 printf("%d\n",i);
 sleep(1);
 }
 }
```

2.

```
  GNU nano 6.2
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
#include<string.h>
void *thread_function(void *arg);
 int i,n,j;
 int main() {
 char *m="5";
 pthread_t a_thread;
 void *result;
 pthread_create(&a_thread, NULL, thread_function, m);
 pthread_join(a_thread, &result);
 printf("Thread joined\n");
for(i=0;i<n;i++)
 for(j=20;j<25;j++)
 {
 printf("%d\n",j);
 sleep(1);
 }
 printf("thread returned %s\n",(char *)result);
```

```
 GNU nano 6.2
 pthread_create(&a_thread, NULL, thread_function, m);
 pthread_join(a_thread, &result);
 printf("Thread joined\n");
for(i=0;i<n;i++)
 for(j=20;j<25;j++)
 {
 printf("%d\n",j);
 sleep(1);
 }
 printf("thread returned %s\n",(char *)result);
 }
 void *thread_function(void *arg) {
 int sum=0;
 n=atoi(arg);
 {
 printf("%d\n",i);
 sleep(1);
 }
 pthread_exit("Done");
 }
```

```
24
~$ nano p1.c
~$ gcc p.c
~$ ./a.out
0
Thread joined
20
21
22
23
24
20
21
22
23
24
20
21
22
23
24
20
21
22
23
```

```
21
22
23
24
20
21
22
23
24
20
21
22
23
24
20
21
22
23
24
20
21
22
23
24
thread returned Done
```

Program 03

```
Writing "I am tread T2 " in the file test.txt...
Finished
~$ touch h.c
~$ nano h.c
~$ gcc h.c
~$ ./a.out
5
7
~$
```

```
 GNU nano 6.2
#include<stdio.h>
#include<pthread.h>
struct arg_struct {
     int arg1;
     int arg2;
 };
void *arguments(void *arguments)
{
    struct arg_struct *args=arguments;
    printf("%d\n", args -> arg1);
    printf("%d\n", args -> arg2);
    pthread_exit(NULL);
}
int main()
 {
     pthread_t t;
     struct arg_struct args;
     args.arg1 = 5;
     args.arg2 = 7;
     pthread_create(&t, NULL, arguments, &args);
     pthread_join(t, NULL);
}
```

Program 04

```
Creating file test.txt...
Writing "I am tread T2 " in the file test.txt...
Finished
~$ touch h.c
~$ nano h.c
~$ gcc h.c
~$ ./a.out
5
7
~$ nano h.c
~$ touch hh.c
~$ nano hh.c
~$ gcc hh.c
~$ ./a.out
Inside Thread
Inside Main process
Sum is 8
~$
```

```
 GNU nano 6.2
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<pthread.h>
void *thread_function(void *arg);
int num[2]={3,5};
int sum;
void *result;
int main() {
pthread_t a_thread;
pthread_create(&a_thread, NULL, thread_function, (void *)num);
pthread_join(a_thread,&result);
printf("Inside Main process\n");
printf("Sum is %d\n",*((int *)result));
}
void *thread_function(void *arg) {
printf("Inside Thread\n");
int *x=arg;
int *sum=(int *)malloc(sizeof(int));
*sum=x[0]+x[1];
pthread_exit((void *)sum);
}
```

Task # 01:

Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <unistd.h>


void *t1_func(void *arg) {

    printf("Creating file test.txt...\n");

    FILE *f = fopen("test.txt","w");

    fclose(f);

    return NULL;

}


void *t2_func(void *arg) {

    sleep(1);

    printf("Writing \"I am tread T2 \" in the file test.txt...\n");

    FILE *f = fopen("test.txt", "a");

    fprintf(f, "I am tread T2 \n");

    fclose(f);

    return NULL;

}


int main() {

    pthread_t t1, t2;
```

```
    pthread_create(&t1, NULL, t1_func, NULL);
    pthread_create(&t2, NULL, t2_func, NULL);

    pthread_join(t1, NULL);
    pthread_join(t2, NULL);

    printf("Finished\n");

    return 0;
}
```

Output :

```
T.c:8:33: note: each undeclared identifier is repor
~$ nano T.c
~$ gcc T.c
~$ ./a.out
Creating file test.txt...
Writing "I am tread T2 " in the file test.txt...
Finished
~$
```

Lab Title: lab 11	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator					
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		
Lab Instructor:					

“lab 11”

COURSE :

OPERATING SYSTEM

SUBMITTED TO :

Sir shehzad

SUBMITTED BY :

Rabia Batool (2022-BSE-064)

SECTION :

B

Code:

```
#include <stdio.h>

struct process
{
   char name[10];
   int burst_time;
   int arrival_time;
   int waiting_time;
   int turnaround_time;
};

int main()
{
   struct process p[3];

   printf("Enter name, burst time, and arrival time of the processes\n");
   for (int i = 0; i < 3; i++)
   {
      printf("Name of process p%d: ", i);
      scanf("%s", p[i].name);
      printf("Burst time of p%d: ", i);
      scanf("%d", &p[i].burst_time);
      printf("Arrival time of p%d: ", i);
      scanf("%d", &p[i].arrival_time);
   }

   printf("Processes after sorting according to shortest job first\n");
   for (int i = 0; i < 3; i++)
   {
      for (int j = 0; j < 2; j++)
      {
         if (p[j].burst_time > p[j + 1].burst_time)
         {
            // Swap
            struct process temp = p[j];
            p[j] = p[j + 1];
            p[j + 1] = temp;
         }
      }
   }
```

```
    printf("PROCESS NAME   BURST TIME   ARRIVAL TIME\n");
    for (int i = 0; i < 3; i++)
    {
        printf("%s          %d          %d\n", p[i].name, p[i].burst_time, p[i].arrival_time);
    }

    p[0].waiting_time = 0;
    p[1].waiting_time = p[0].burst_time;
    p[2].waiting_time = p[0].burst_time + p[1].burst_time;
    p[0].turnaround_time = p[0].burst_time;
    p[1].turnaround_time = p[0].burst_time + p[1].burst_time;
    p[2].turnaround_time = p[0].burst_time + p[1].burst_time + p[2].burst_time;

    printf("\nImplementation of Shortest Job First Algorithm\n");
    printf("PROCESS NAME   BURST TIME   ARRIVAL TIME   WAITING TIME   TURNAROUND TIME\n");
    for (int i = 0; i < 3; i++)
    {
        printf("%s          %d          %d          %d          %d\n", p[i].name, p[i].burst_time, p[i].arrival_time,
p[i].waiting_time, p[i].turnaround_time);
    }

    int temp = 0, temp1 = 0;
    for (int i = 0; i < 3; i++)
    {
        temp1 += p[i].turnaround_time;
        temp += p[i].waiting_time;
    }

    printf("\nAverage Waiting Time of the Processes = %d\n", temp / 3);
    printf("Average Turnaround Time of the Processes = %d\n", temp1 / 3);

    return 0;
}
```

Output:

```
Average Turnaround Time of the Processes = 0
~$ gcc lab.c
~$ ./a.out
Enter name, burst time, and arrival time of the processes
Name of process p0: p1
Burst time of p0: 24
Arrival time of p0: 0
Name of process p1: p2
Burst time of p1: 3
Arrival time of p1: 0
Name of process p2: p3
Burst time of p2: 2
Arrival time of p2: 0
Processes after sorting according to shortest job first
PROCESS NAME   BURST TIME   ARRIVAL TIME
p3             2            0
p2             3            0
p1             24            0

Implementation of Shortest Job First Algorithm
PROCESS NAME   BURST TIME   ARRIVAL TIME   WAITING TIME   TURNAROUND TIME
p3             2            0            0             2
p2             3            0            2             5
p1             24            0            5             29

Average Waiting Time of the Processes = 2
Average Turnaround Time of the Processes = 12
```

Lab Title: lab 12 & 13	Experiment # :
Student Name : Rabia Batool	Regd. #:
Experiment Name:	

Performance Indicator					
	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)
Understanding & implementation Problem					
Report and presentation					
Maximum Marks		10	Obtained Marks		
Lab Instructor:					

“lab 12 &13”

COURSE :

OPERATING SYSTEM

SUBMITTED TO :

Sir shehzad

SUBMITTED BY :

Rabia Batool (2022-BSE-064)

SECTION :

B

LAB TASK

- Implement Banker's Algorithm.

Code:

```
#include <stdio.h>
#include <stdbool.h>

struct process_info
{
   int max[10];
   int allocated[10];
   int need[10];
};

int no_of_process,no_of_resources;

void input(struct process_info process[no_of_process],int available[no_of_resources])
{
    for(int i=0;i<no_of_process;i++)
   {
     printf("Enter process[%d] info\n",i);
     printf("Enter Maximum Need: ");
```

```
    for(int j=0;j<no_of_resources;j++)
      scanf("%d",&process[i].max[j]);
    printf("Enter No. of Allocated Resources for this process: ");
    for(int j=0;j<no_of_resources;j++)
    {
      scanf("%d",&process[i].allocated[j]);
            process[i].need[j]= process[i].max[j] - process[i].allocated[j];
    }
   }
   printf("Enter Available Resources: ");
   for(int i=0;i<no_of_resources;i++)
   {
    scanf("%d",&available[i]);
   }
}
void showTheInfo(struct process_info process[no_of_process])
{
   printf("\nPID\tMaximum\t\tAllocated\tNeed\n");
   for(int i=0;i<no_of_process;i++)
   {
      printf("P[%d]\t",i);
      for(int j=0;j<no_of_resources;j++)
        printf("%d ",process[i].max[j]);
      printf("\t\t");
      for(int j=0;j<no_of_resources;j++)
        printf("%d ",process[i].allocated[j]);
      printf("\t\t");
      for(int j=0;j<no_of_resources;j++)
        printf("%d ",process[i].need[j]);
```

```
        printf("\n");


    }
}
bool applySafetyAlgo(struct process_info process[no_of_process],int available[no_of_resources],int
safeSequence[no_of_process])
{
   bool finish[no_of_process];
   int work[no_of_resources];
   for(int i=0;i<no_of_resources;i++)
   {
      work[i]=available[i];
   }
   for(int i=0;i<no_of_process;i++)
    finish[i]=false;
   bool proceed=true;
   int k=0;
   while(proceed)
   {
    proceed=false;
    for(int i=0;i<no_of_process;i++)
    {
        bool flag=true;



        if(finish[i]==false)
        {


          for(int j=0;j<no_of_resources;j++)
```

```
    {
      if(process[i].need[j] <= work[j])
      {
        continue;
      }
      else
      {
        flag=false;
        break;
      }
    }
    if(flag==false)
      continue;
    for(int j=0;j<no_of_resources;j++)
      work[j]=work[j]+ process[i].allocated[j];
    finish[i]=true;
    safeSequence[k++]=i;
    proceed=true;
   }
  }

}

int i;
for( i=0;i<no_of_process&&finish[i]==true;i++)
{
  continue;
}
if(i==no_of_process)
```

```
     return true;
    else
     return false;
}

bool isSafeState(struct process_info process[no_of_process],int available[no_of_resources],int
safeSequence[no_of_process])
{

   if(applySafetyAlgo(process,available,safeSequence)==true)
     return true;
   return false;

}



int main()
{
  printf("Enter No of Process\n");
   scanf("%d",&no_of_process);
  printf("Enter No of Resource Instances in system\n");
   scanf("%d",&no_of_resources);
   int available[no_of_resources];
   int safeSequence[no_of_process];
   struct process_info process[no_of_process];

  printf("***********Enter details of processes*************\n");
   input(process,available);
```

```
  showTheInfo(process);
   if(isSafeState(process,available,safeSequence))
   {

   printf("\nSystem is in SAFE State\n");
   printf("Safe Sequence is: ");
   for(int i=0;i<no_of_process;i++)
      printf("P[%d] ",safeSequence[i]);
 printf("1");
   }
   else
     printf("0");
   return 0;
}
```

Output :

```
Enter No of Process
5
Enter No of Resource Instances in system
3
**********Enter details of processes**********
Enter process[0] info
Enter Maximum Need: 7
5
3
Enter No. of Allocated Resources for this process: 0
1
0
Enter process[1] info
Enter Maximum Need: 3
2
2
Enter No. of Allocated Resources for this process: 2
0
0
Enter process[2] info
Enter Maximum Need: 9
0
2
Enter No. of Allocated Resources for this process: 3
0
```

```
Enter Maximum Need: 9
0
2
Enter No. of Allocated Resources for this process: 3
0
2
Enter process[3] info
Enter Maximum Need: 2
2
2
Enter No. of Allocated Resources for this process: 2
1
1
Enter process[4] info
Enter Maximum Need: 4
3
3
Enter No. of Allocated Resources for this process: 3
3
2
Enter Available Resources: 3
3
2
```

```
PID     Maximum         Allocated       Need
P[0]    7 5 3           0 1 0           7 4 3
P[1]    3 2 2           2 0 0           1 2 2
P[2]    9 0 2           3 0 2           6 0 0
P[3]    2 2 2           2 1 1           0 1 1
P[4]    4 3 3           3 3 2           1 0 1

System is in SAFE State
Safe Sequence is: P[1] P[3] P[4] P[0] P[2] 1~$
```