RIPHAH
INTERNATIONAL
UNIVERSITY

Operating Systems Lab 07

Submitted By:

Soha Ali

51169

Submitted To:

Ma’am Rehana Sanam

Question 01

What would be the result of the following commands. Understand that is

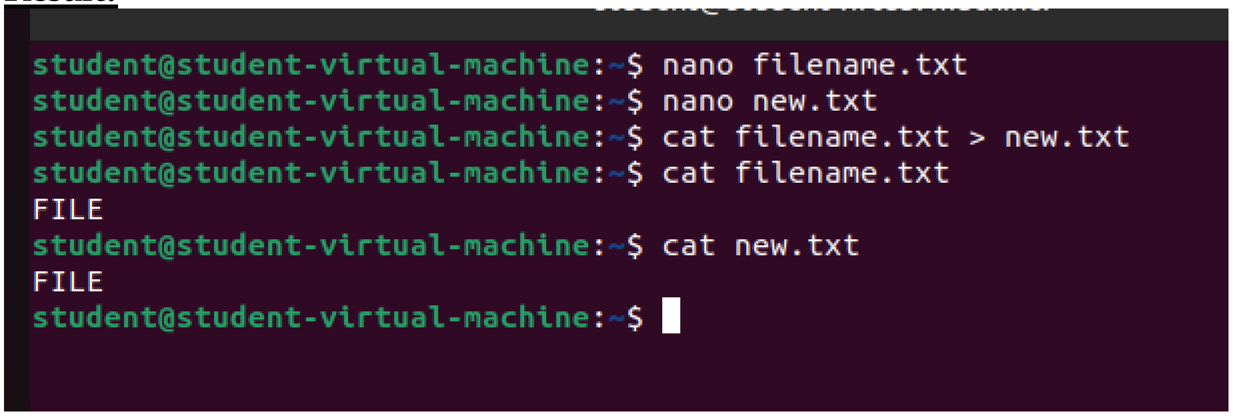
- cat filename > new   // filename is name of file that exist at current path
- who > new
- ls | sort –r
- ls | sort -r >> new

Also define the understanding of these commands in your own words.

Solution:

1. **cat filename > new**

   Result:



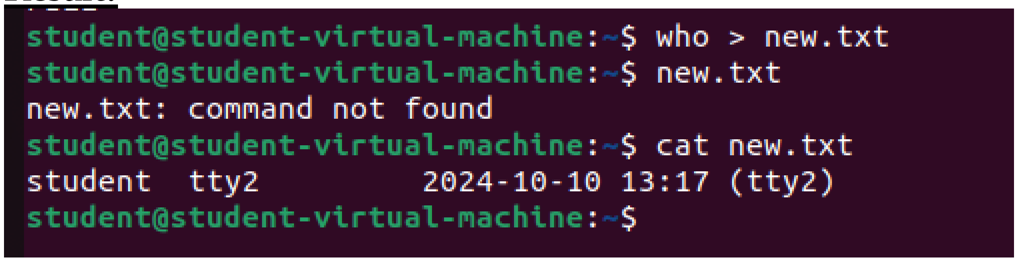
```
student@student-virtual-machine:~$ nano filename.txt
student@student-virtual-machine:~$ nano new.txt
student@student-virtual-machine:~$ cat filename.txt > new.txt
student@student-virtual-machine:~$ cat filename.txt
FILE
student@student-virtual-machine:~$ cat new.txt
FILE
student@student-virtual-machine:~$
```

Explanation:

This > operator gets the output from the first file and copy's its contents into the second file. For example, if I write "FILE" in the first file named filename.txt and write "NEW" in the second file named new.txt, by using the > operator and typing in the command cat filename.txt > new.txt it will copy the word "FILE" from filename.txt into new.txt. So, now new.txt will also contain the word "FILE" instead of "NEW". It will overwrite the word "NEW". Basically, this command is used to pass the output of one file and overwrite the output of the other file.

2. **who > new**

   Result:



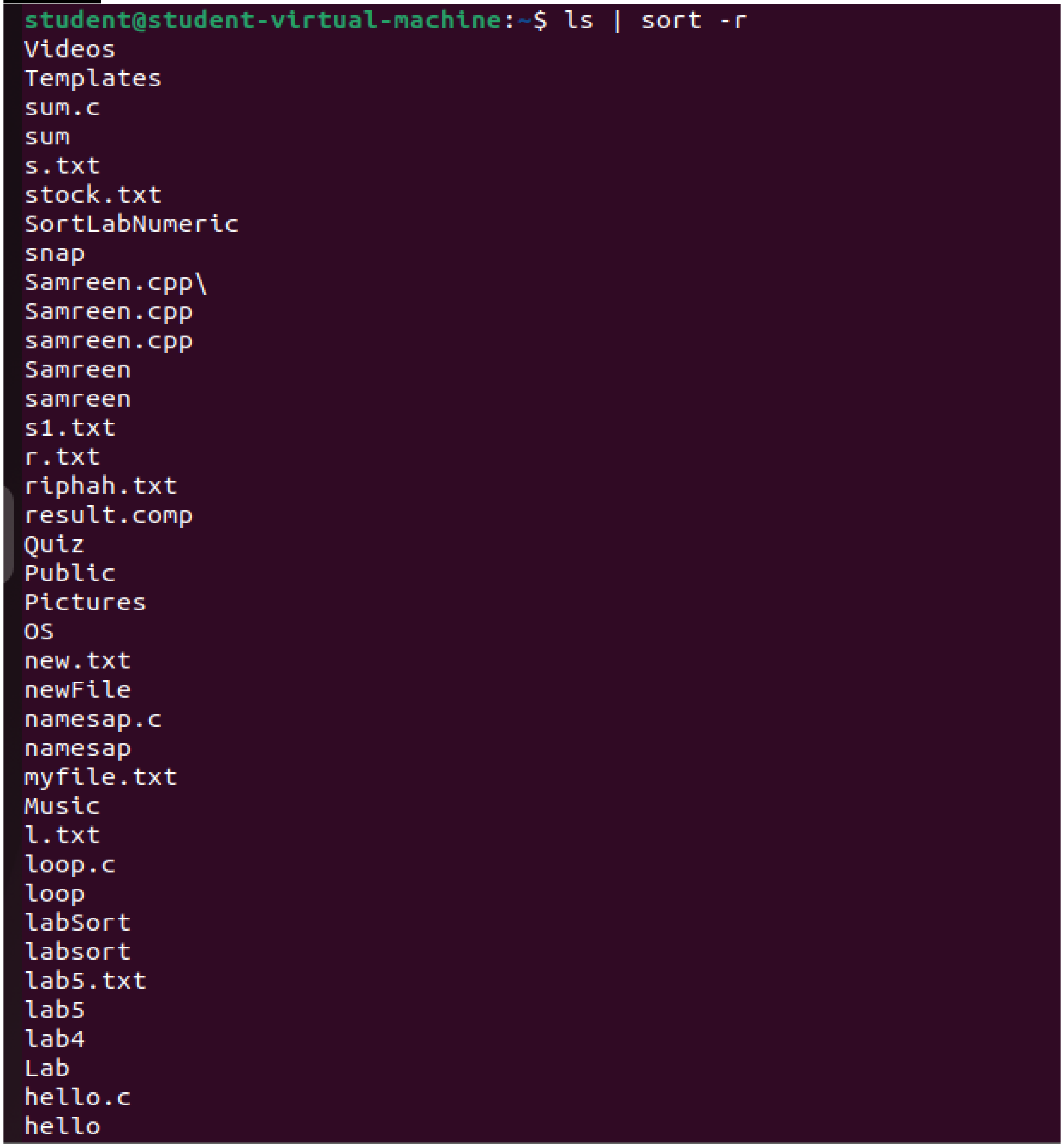
```
student@student-virtual-machine:~$ who > new.txt
student@student-virtual-machine:~$ new.txt
new.txt: command not found
student@student-virtual-machine:~$ cat new.txt
student  tty2         2024-10-10 13:17 (tty2)
student@student-virtual-machine:~$
```

Explanation:

This command will overwrite the content of file named new.txt and display us the information on who is the owner/user of the current Ubuntu system along with when they first logged in with date and time.

3. **ls | sort –r**

Result:



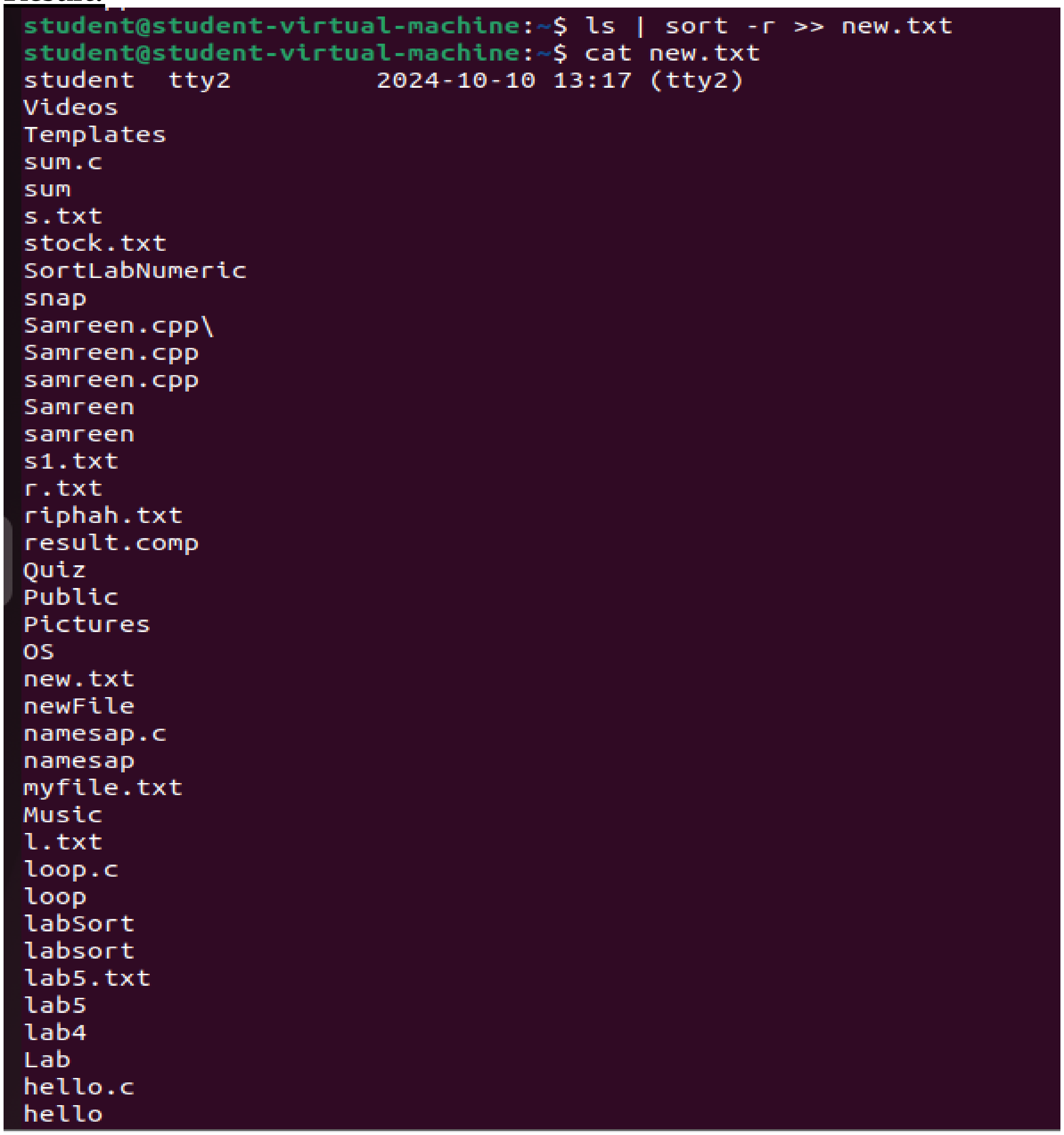
```
student@student-virtual-machine:~$ ls | sort -r
Videos
Templates
sum.c
sum
s.txt
stock.txt
SortLabNumeric
snap
Samreen.cpp\
Samreen.cpp
samreen.cpp
Samreen
samreen
s1.txt
r.txt
riphah.txt
result.comp
Quiz
Public
Pictures
OS
new.txt
newFile
namesap.c
namesap
myfile.txt
Music
l.txt
loop.c
loop
labSort
labsort
lab5.txt
lab5
lab4
Lab
hello.c
hello
```

Explanation:

This command shows us the list of directories and files and also displays it in reverse alphabetical order. We us "*ls*" command to show the list of directories and files and we use "*sort –r*" command to sort and directories and files into reverse alphabetical order. We use the "*|*" operator (which is called the pipe operator) to do both displaying directories and files and displaying them in reverse at the same time in parallel. In this way we won't have to type in commands separately, we can just use the pipe operator to have both things done at once.

4. ls | sort -r >> new

Result:



```
student@student-virtual-machine:~$ ls | sort -r >> new.txt
student@student-virtual-machine:~$ cat new.txt
student  tty2         2024-10-10 13:17 (tty2)
Videos
Templates
sum.c
sum
s.txt
stock.txt
SortLabNumeric
snap
Samreen.cpp\
Samreen.cpp
samreen.cpp
Samreen
samreen
s1.txt
r.txt
riphah.txt
result.comp
Quiz
Public
Pictures
OS
new.txt
newFile
namesap.c
namesap
myfile.txt
Music
l.txt
loop.c
loop
labSort
labsort
lab5.txt
lab5
lab4
Lab
hello.c
hello
```

Explanation:

The *“ls | sort –r”* command is used to display the directories and files in reverse alphabetical order. Now, when we use the operator >> in between *“ls | sort –r“* and *“new.txt”* we are appending the output of *“ls | sort –r”* into *“new.txt*. Now what this will do is that it will not override the content that is written in “new.txt”(in this case the content inside new.txt which is owner/user information and login date and time). It will just copy the output we got from *“ls | sort –r”* into *“new.txt”* without overriding the content already stored in *“new.txt”*.

Question 02

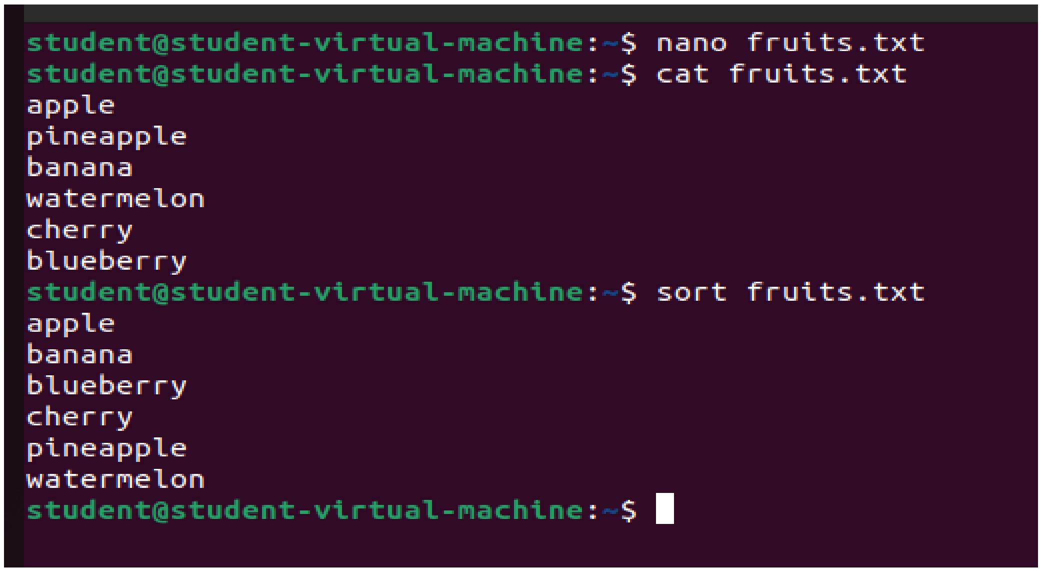
Write a command that does the following:

1. Sorts the contents of **fruits.txt** in alphabetical order.
2. Filters the sorted output to only display fruits that contain the substring "ap".

Define streams, redirection and pipes with one example of each in your own words.

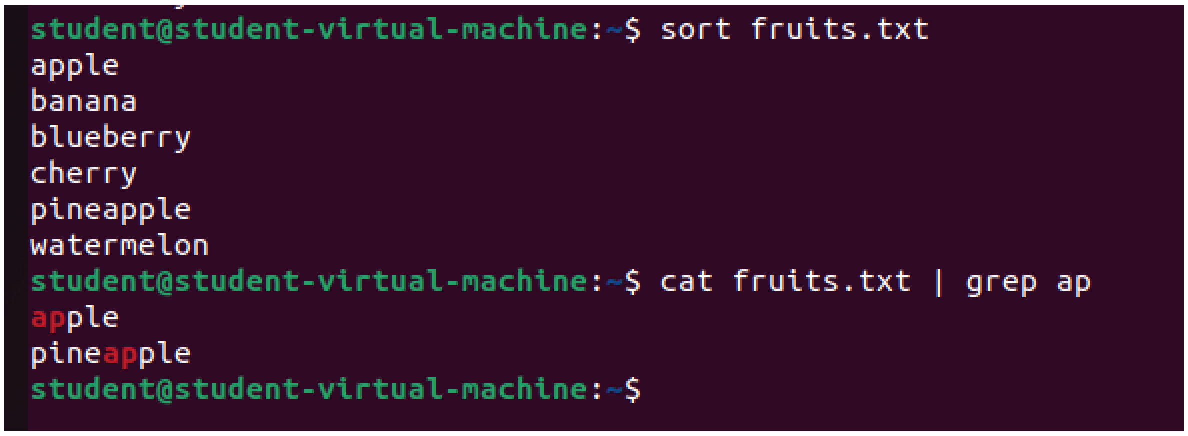
Solution:

1. **Sorts the contents of fruits.txt in alphabetical order.**



```
student@student-virtual-machine:~$ nano fruits.txt
student@student-virtual-machine:~$ cat fruits.txt
apple
pineapple
banana
watermelon
cherry
blueberry
student@student-virtual-machine:~$ sort fruits.txt
apple
banana
blueberry
cherry
pineapple
watermelon
student@student-virtual-machine:~$
```

2. **Filters the sorted output to only display fruits that contain the substring "ap".**



```
student@student-virtual-machine:~$ sort fruits.txt
apple
banana
blueberry
cherry
pineapple
watermelon
student@student-virtual-machine:~$ cat fruits.txt | grep ap
apple
pineapple
student@student-virtual-machine:~$
```

Streams:

Streams basically refers to the stream of input or output data we get from the keyboard to the main memory

Example:

For example, in C programs we use the header file or #include<stdio> in order to allow the stream of input and output data from the keyboard to the main memory. Another example could be when we use “stdout” to display the output in C++ programs, It displays the stream of output data in C++ programs.

Redirection:

Redirection basically refers to the direction of the input output data coming from the files instead of the keyboard. For redirection we use less than for input (<), greater than for output (>), append (>>) operators.

Example:

For example, we write the command cat < new.txt to read the input of the data.

Pipes:

Pipes basically help us to run two commands at once. It combines two commands and runs them in parallel.

Example:

For example, in the command ls | sort –r , the displaying of directories and files and displaying in reverse alphabetical order is shown at once.