**Introduction to basic Command of Linux**

**Lab no# 01**

****

**Fall 2022**

**CSE-302 System Programming Lab**

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr: Ma’am Madeha sheer**

**November** 9, 2021

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

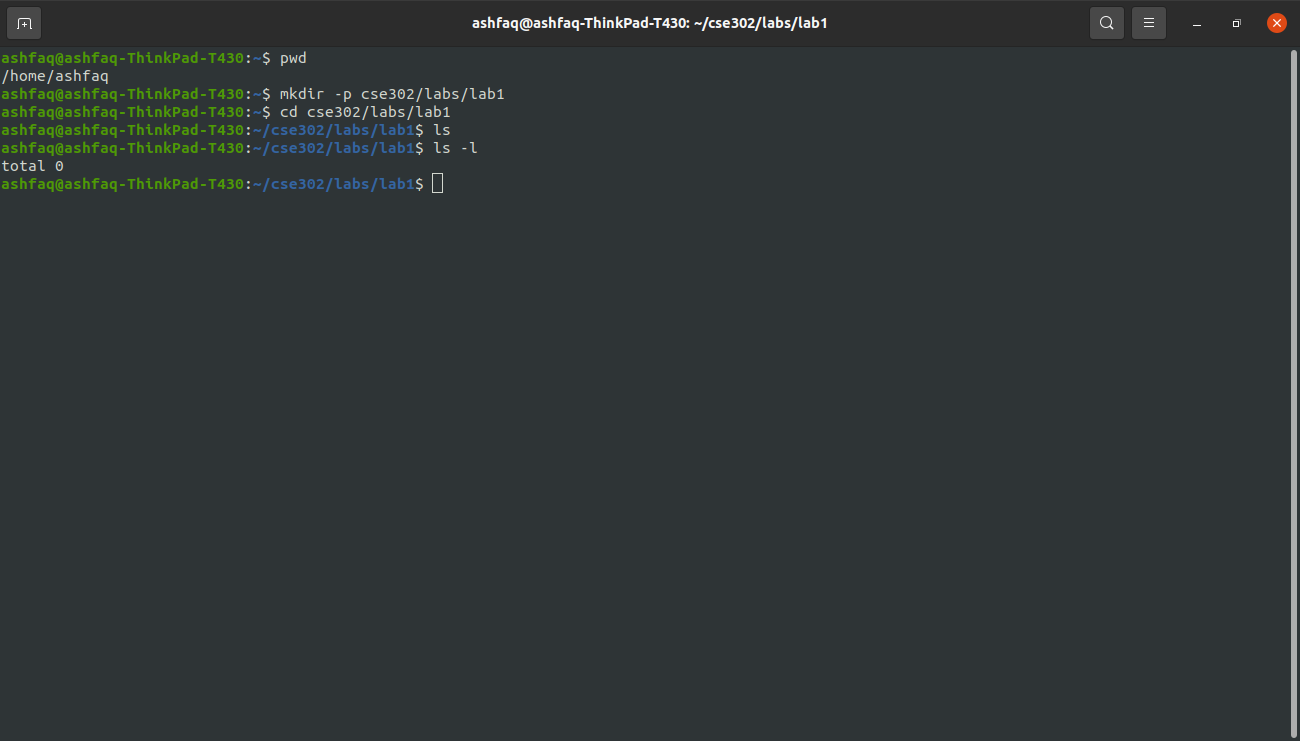
**Objective:**

* In this lab we will learn about basic command of Linux system.
* To know about hard link and soft link.
* To know about the permissions of file and directory and its types.
* To know about types of users.
* To know how we create a transcript.

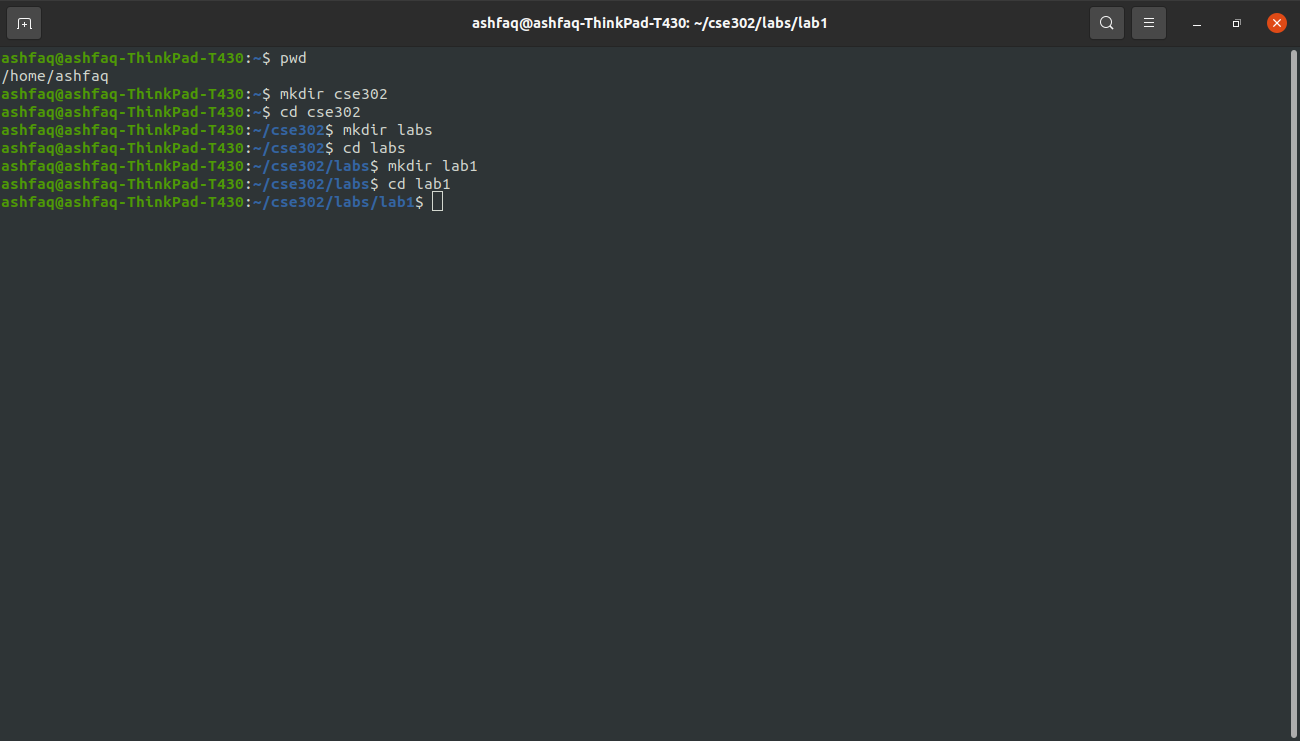
**Task 1:**

1. In your home directory create the subdirectory ~/cse302/labs/lab1.

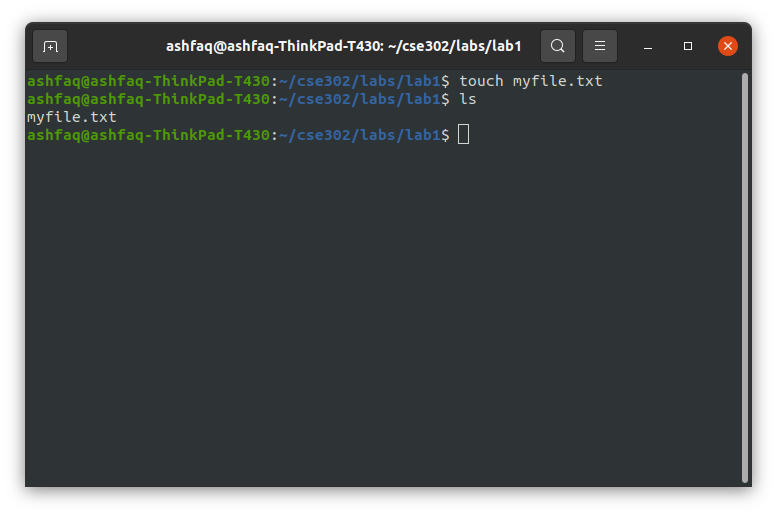
Cd ~/cse302/labs/lab1.



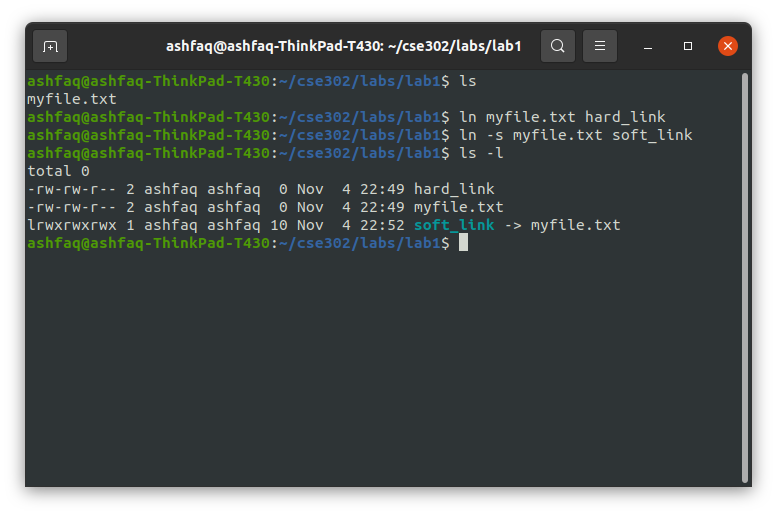
**2nd method:**



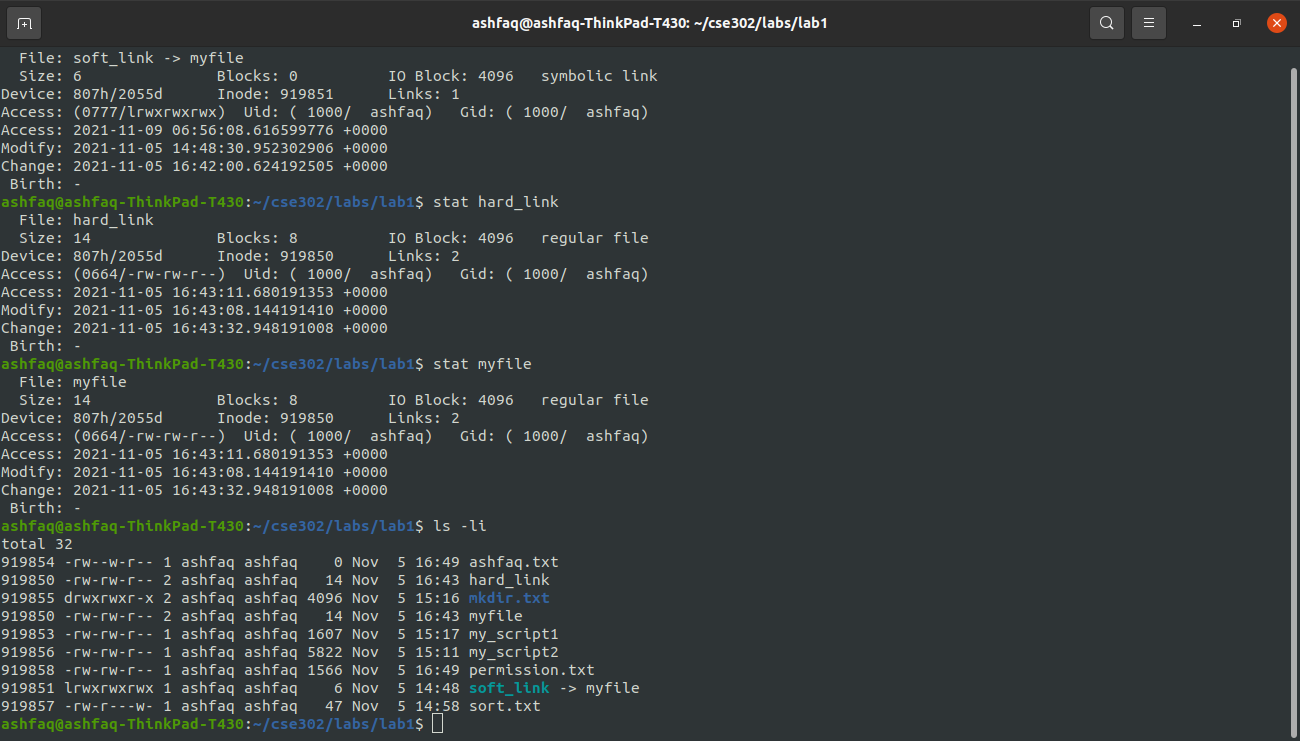
1. Create a file named **myfile** into ~/cse302/labs/lab1.



1. Create a soft link **soft link** and a hard link **hard link** to that file.



1. Based on the output returned by stat and ls commands (using all relevant options), explain in detail (but briefly) the differences between the three files.



**Discussion and Conclusion:**

A soft link is an actual link to the original file, whereas a hard link is a mirror copy of the original file. If you delete the original file, the soft link has no value, because it points to a nonexistent file. But in the case of hard link, it is entirely opposite. Even if you delete the original file, the hard link will still has the data of the original file. Because hard link acts as a mirror copy of the original file. As shown in the above screenshot, original file and hard link has the same inode number and permission while the inode number and permission for soft link is different. If we edit original file such edition also appear in hard link file. Such phenomena also occur in case of soft link. If we change the name of original file the hard link will work but soft link will have no value.

**Task 2:**  Read the man pages for the following commands:

* 1. script
  2. finger, who, w
  3. touch
  4. top
  5. mkdir
  6. umask: umask [value] (shell built in command)
  7. text utilities: sort, uniq, tr, expand, unexpand, cut, grep .
  8. vi, gcc
  9. history

1. **Script command:**

* store all terminal activities
* **command** “script file name”
* Terminated by exit.

1. **Who:** show who is login.
2. **W command:**

* W displays information about the users currently on the machine.
* It shows what the user doing.

1. **Touch:**

* Update the access and modification times of each FILE to the current time.
* Also create new file if such file does not exist before.
* Touch -a file name: change only access time.
* Touch -m file name: change only modification time.

1. **Top:**

* it display all Linux process currently running on the system.
* We can kill any process by pressing k key then enter pid of process then press enter key two time.
* We can also change their updating time by pressing d key then enter time then press enter key.

1. **Finger command:** give detail information about all users log in.
2. **Mkdir:** used for the creation of new directory.
3. **Umask:**

* Umask give default permission to each newly created directory or file.
* Umask has value. We can change default permission by changing the value of umask.
* Max value: 777
* min value: 000

**How we find default permission of any file or directory?**

We will subtract umask value from 777, and then we will convert it into binary. It will give default permission.

0=no permission

1=permission

**Note:** we can also change default permission manually.

Linux systems do not give permission of execution for file even if we set complete permission using umask. For directory Linux system give complete permission.

**Text utilities:**

**1) Sort:**

* **command:** sort file name
* Sort is used for arranging list or table in ascending or descending order.

**2) Uniq:**

* command: cat file name |sort| uniq
* it is used to report or omit the repeated lines.
* Without sort command it does not work.

**3) Tr command:**

* command : tr a-z A-Z <filename
* It is used for converting text to upper case or lower case.
* tr -d ‘alphabet’ is used for deleting selected alphabets.
* tr -s ‘alphabet’ is used for squeezing selected repeated alphabet to single one.

**4) Cut:**

* remove sections from each line of files
* command: cut -b -n

It will display first n number bits of the list only and remove other bits.

* command: cut -b n-

It will remove first n no of bits from the list and display the remaining bits.

* Command : cut -c n-n

it will display character from n to n no of each line of the list.

**5) Grep:**

* it is used for searching any string in a file.
* Command: grep “string” <file name or cat file name | grep “string”.

**6) Expand:**  used to convert tabs in to spaces.

**7) Unexpand:** used to convert spaces into tabs.

**8) vi:** Improved, a programmer's text editor.

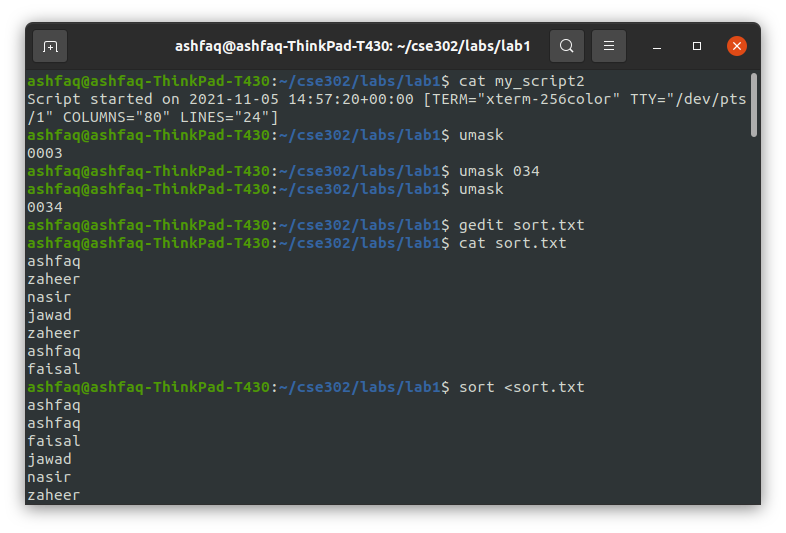
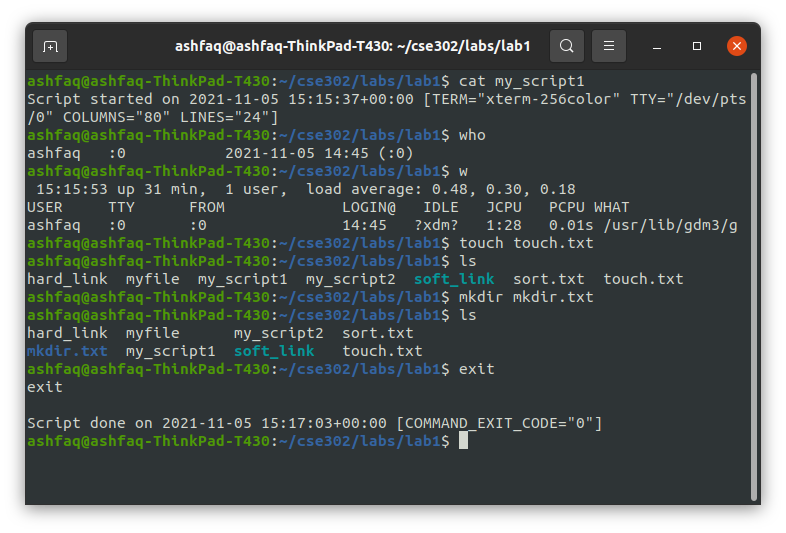
**9) Gcc command**:

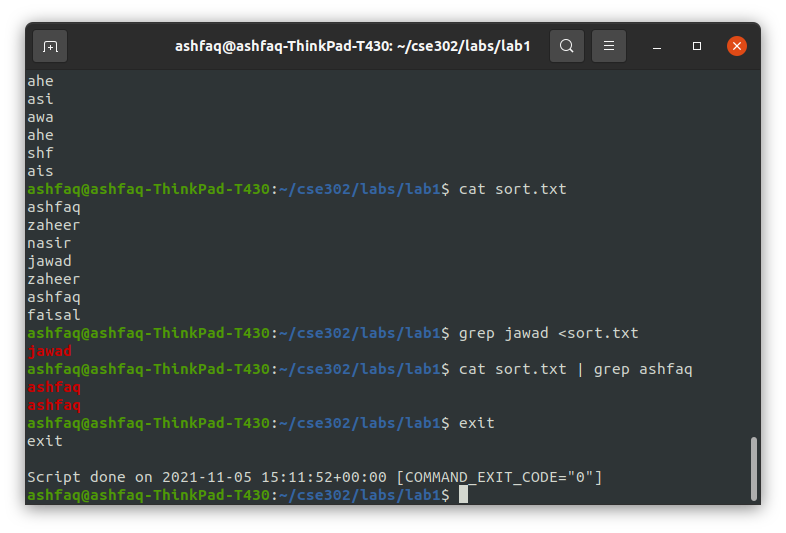
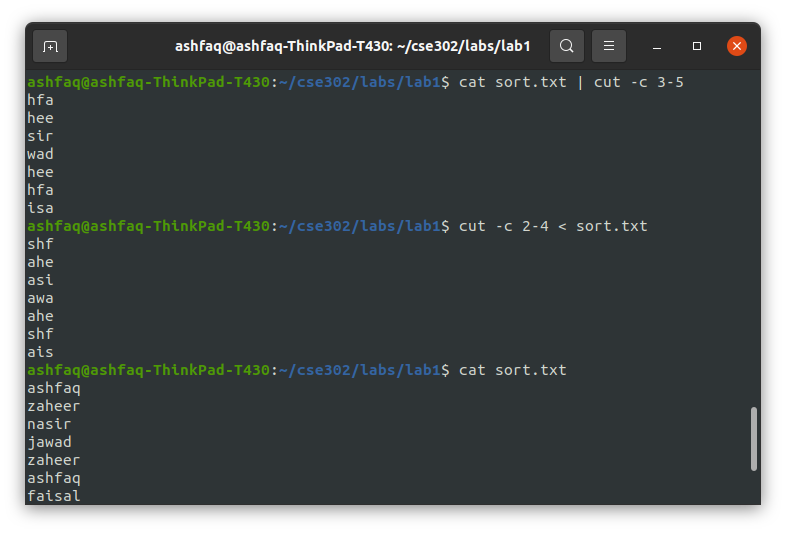
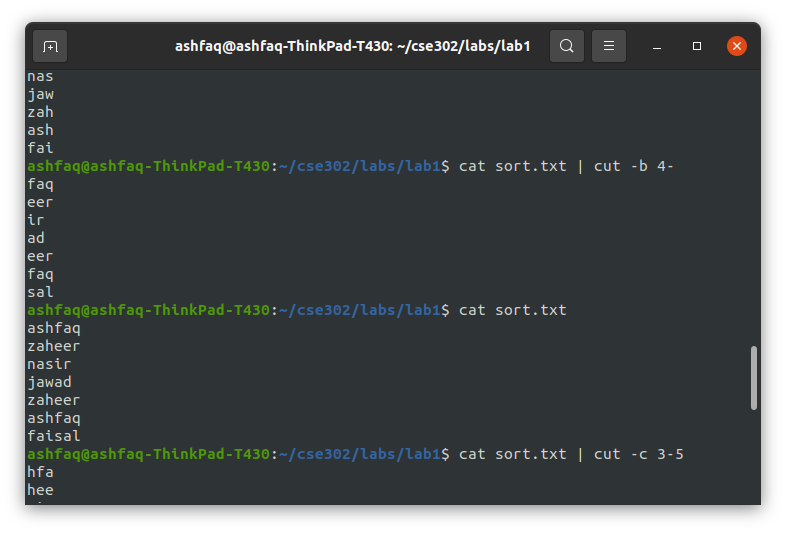
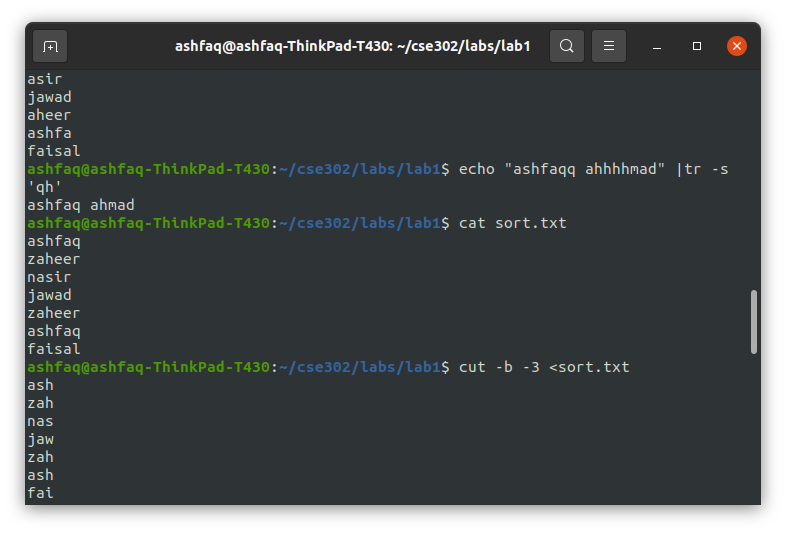
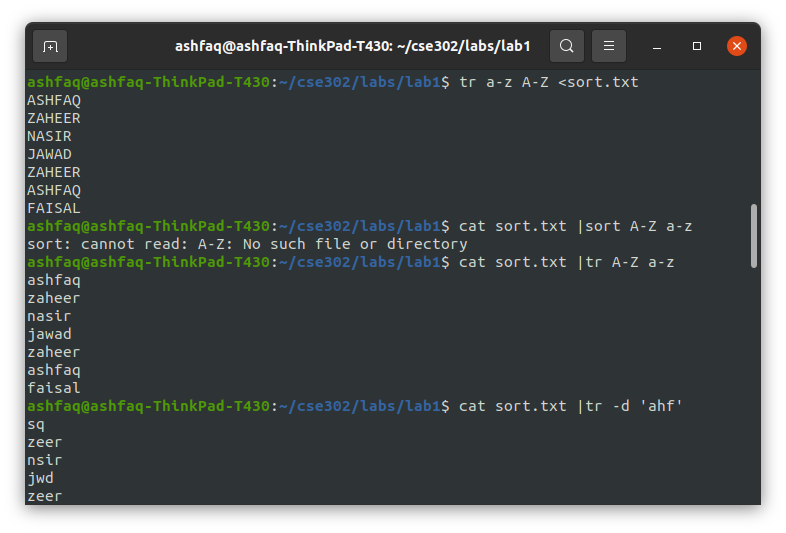
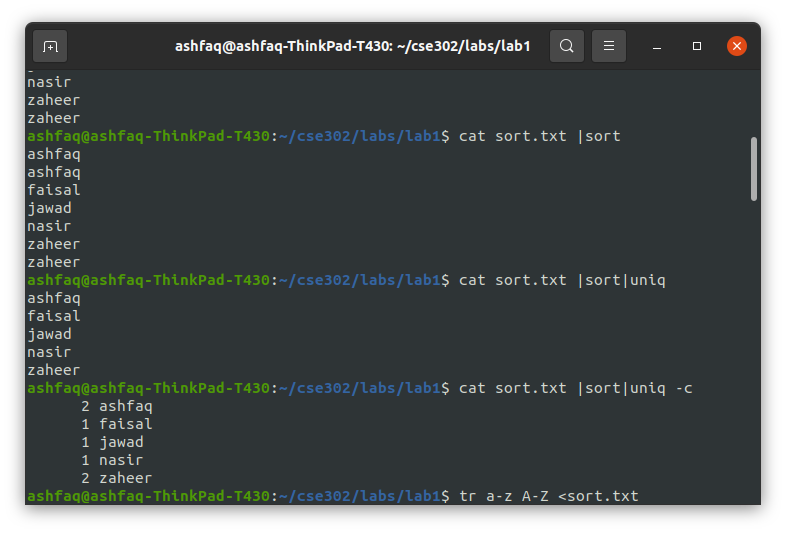
* It is used for compiling c and c++ program.
* Command: gcc file\_name.c -o file name.

**10) History:**

* It is used to view all previously used command in terminal.

**Transcript of the above command:**



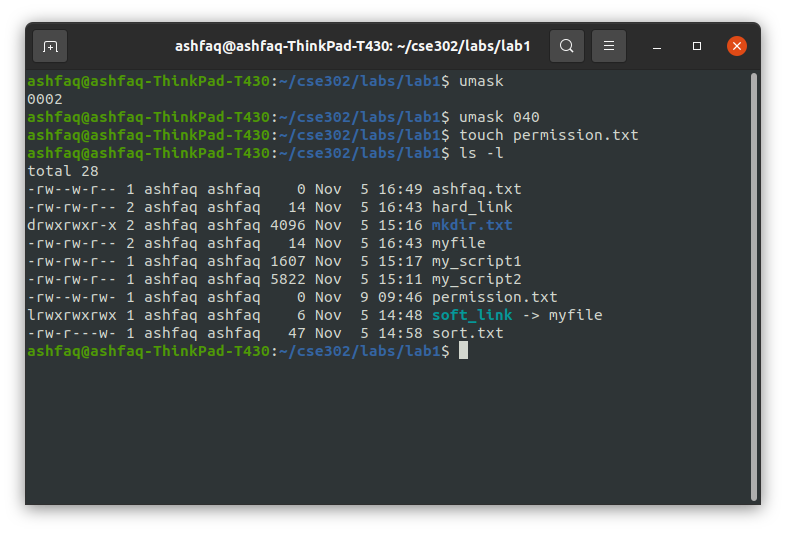


**Task 3:**

Change your file permission mask such that by default your colleagues do not have read permissions for your newly created files. Please show in the transcript files the following:

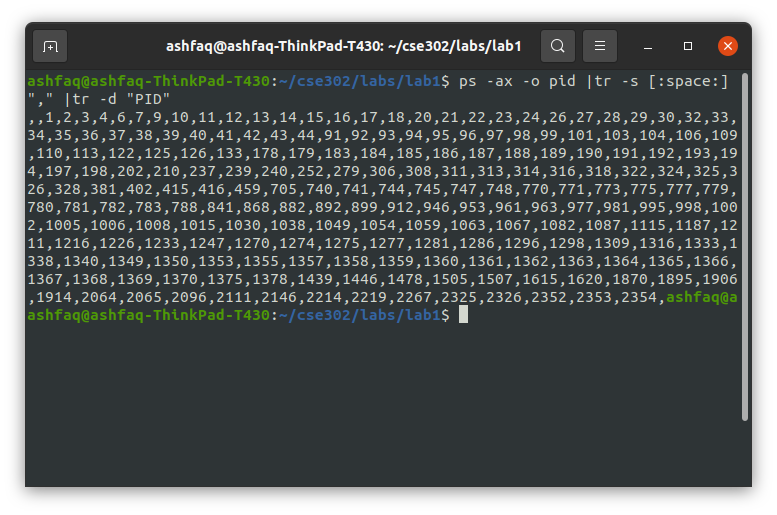
* the initial mask
* how you changed it
* Show that people in your user group don't have read permissions for a new file you're creating.

Change the umask permanently by placing the umask ... command in your .bash profile file.



**Task 4:**

List the PIDs of all processes running as root on your computer on a line, separated by commas. E.g.,: 1,2,3,4,5,657,658, ... Use pipes to create a one-line command that accomplishes this. You'll need some of the text processing tools presented in class. Hint: man ps (-a and -x flags), man tr.



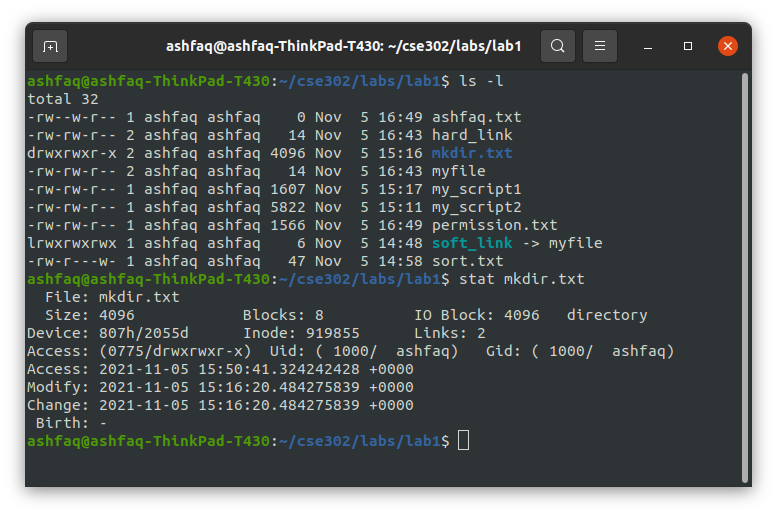
**Task 5:**

List the usernames and names of the people logged on 192.168.2.139. (NOTE:  you may find the command option of ssh helpful.) The list returned should be sorted, should not contain duplicates (e.g., same user listed multiple times) and should have the following format:

1 <username1> <name1>  
2 <username2> <name2>  
3 <username3> <name3>  
...

Hints: use the "nl" command to number lines.

Explain what information you can get about ~Desktop/cse302/labs/lab1 using all (and only) relevant options of ls and stat commands: (is it a file? a directory? how large? permissions? access info? etc.) Indicate the fields that reveal these pieces of information.



**Discussion:**

We can see from the above screenshot that mkdir.txt is a directory, “soft link” is a link and all the others are normal files. Permissions for directory and a normal file are different. A directory has 2 links by default while file has always one link. The size of directory is also bigger than normal files. We can also see that soft link has an arrow next to it showing that it points to myfile.

The End