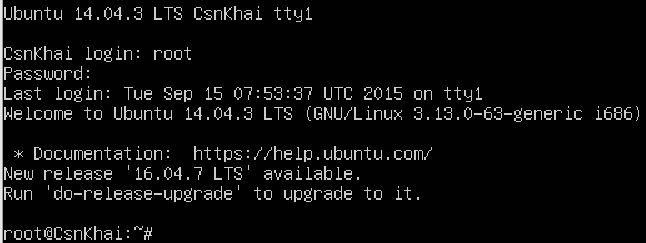
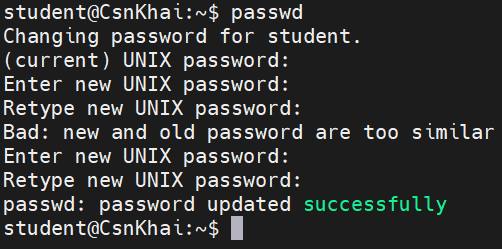
# Task1.Part1

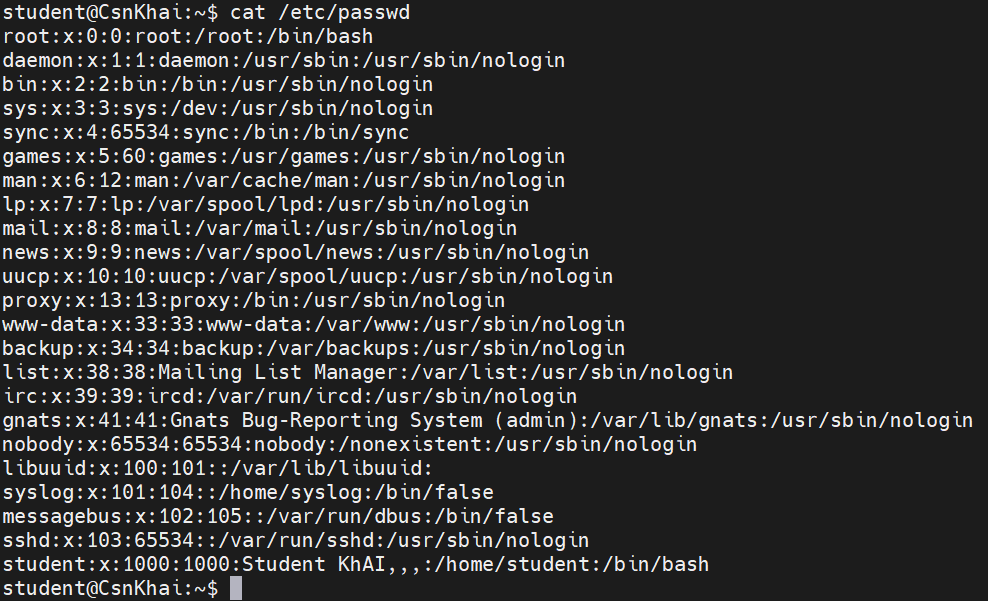
1. Log in to the system as root.
2. Use the passwd command to change the password. Examine the basic parameters of the command. What system file does it change \*?



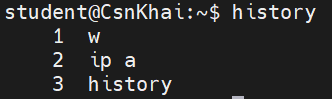
\*This command will change /etc/shadow file.

1. Determine the users registered in the system, as well as what commands they execute. What additional information can be gleaned from the command execution?

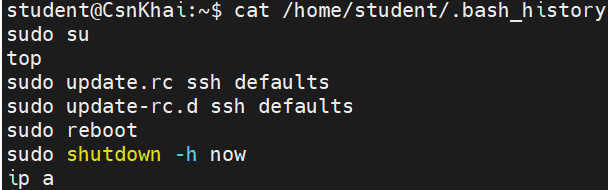
Registered users:



You can check what commands executed by user with a command “history”:

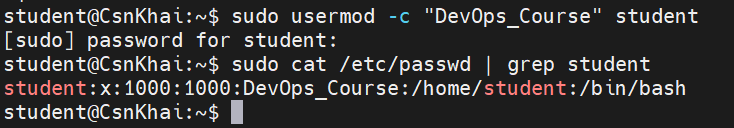


or with displaying of .bash\_history file:



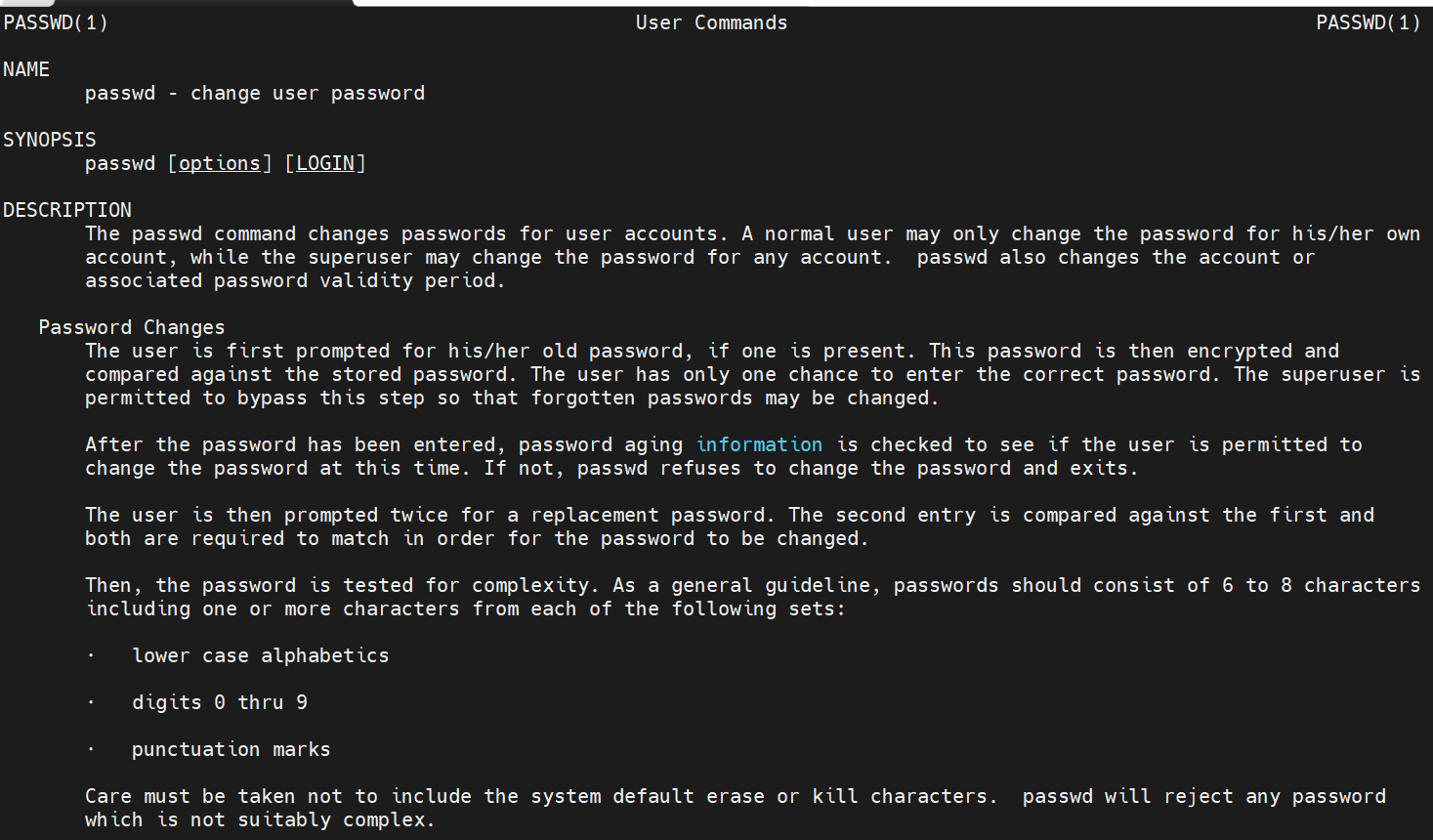
From this methods you can get command execution order for each user.

1. Change personal information about yourself.

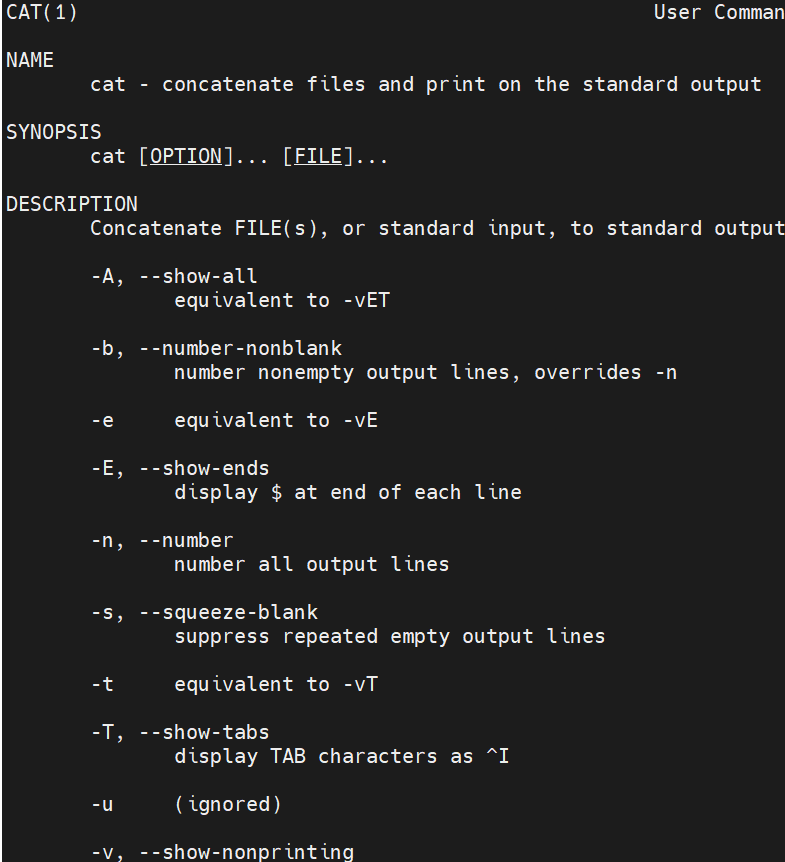
Adding comment for myself:

1. Become familiar with the Linux help system and the man and info commands. Get help on the previously discussed commands, define and describe any two keys for these commands. Give examples.

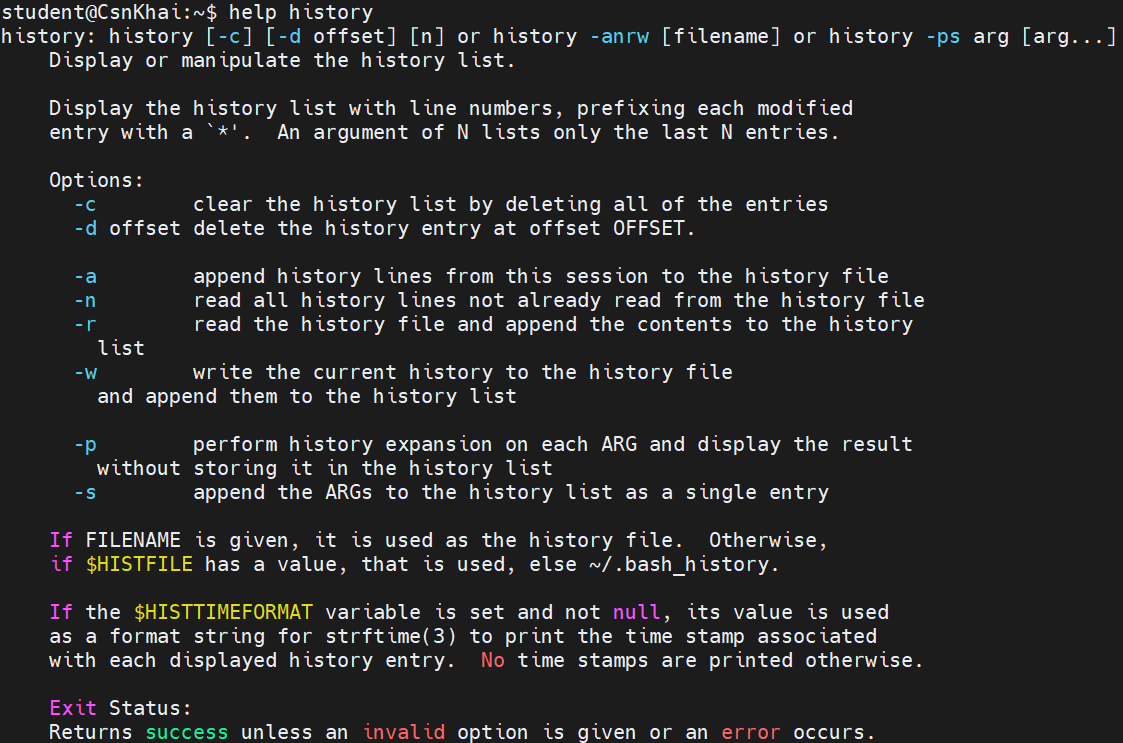
man passwd:



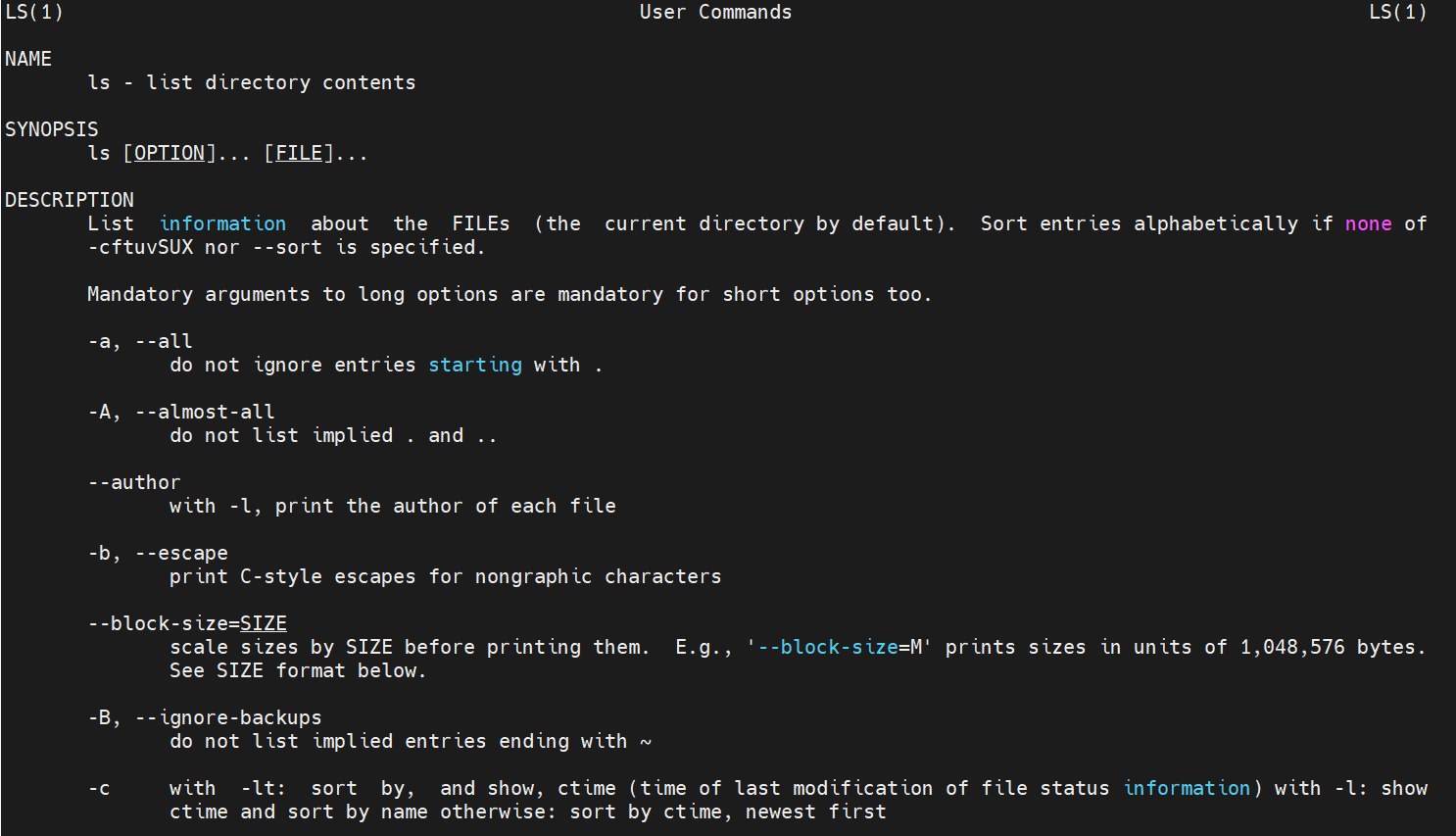
man cat:



help history:



man ls:

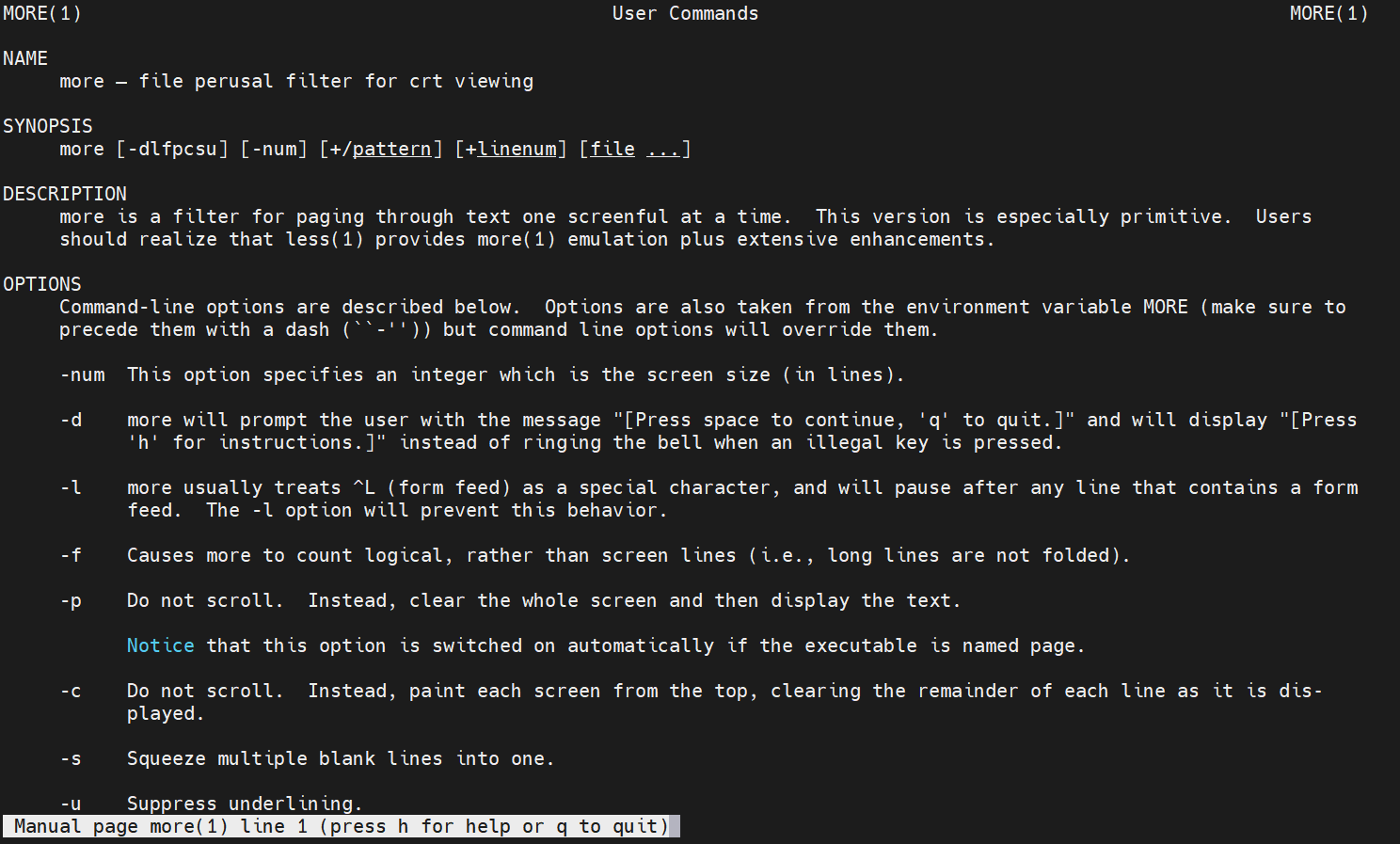


Command “ls” with “-la” will display a detailed listing of all files and directories in the current directory, including hidden files.

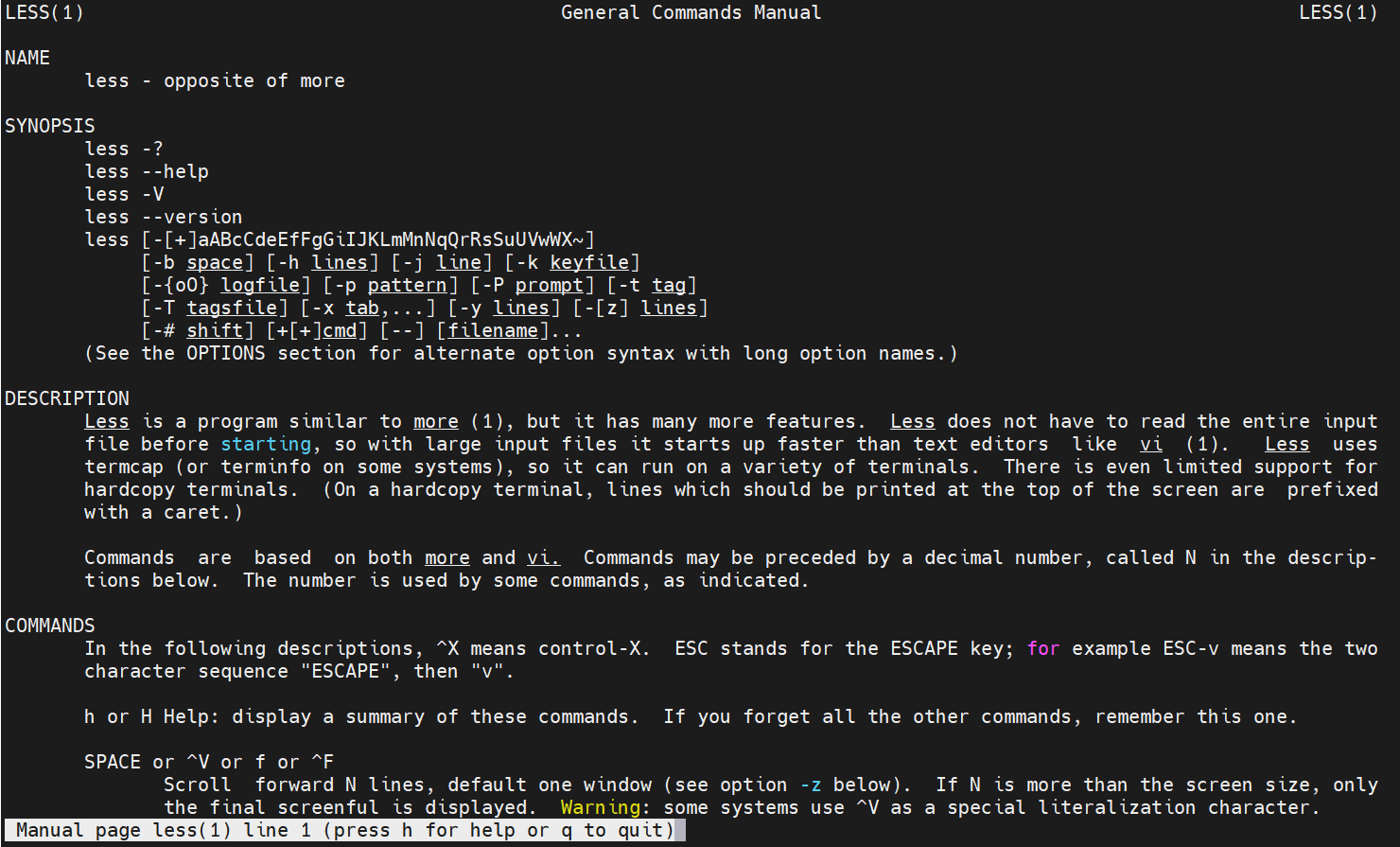
Command “history” with “-c” will clear history.

1. Explore the more and less commands using the help system. View the contents of files .bash\* using commands.

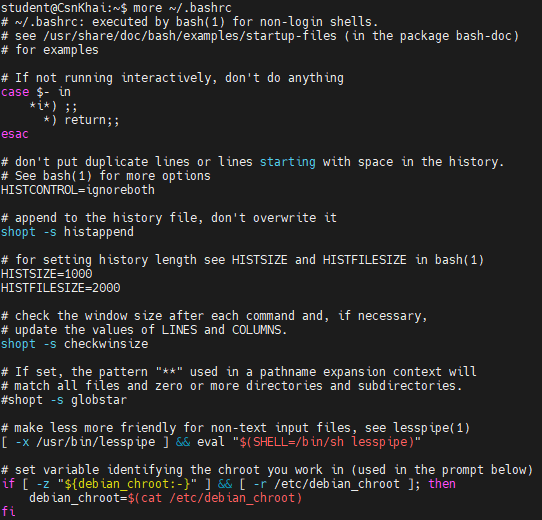
man more:



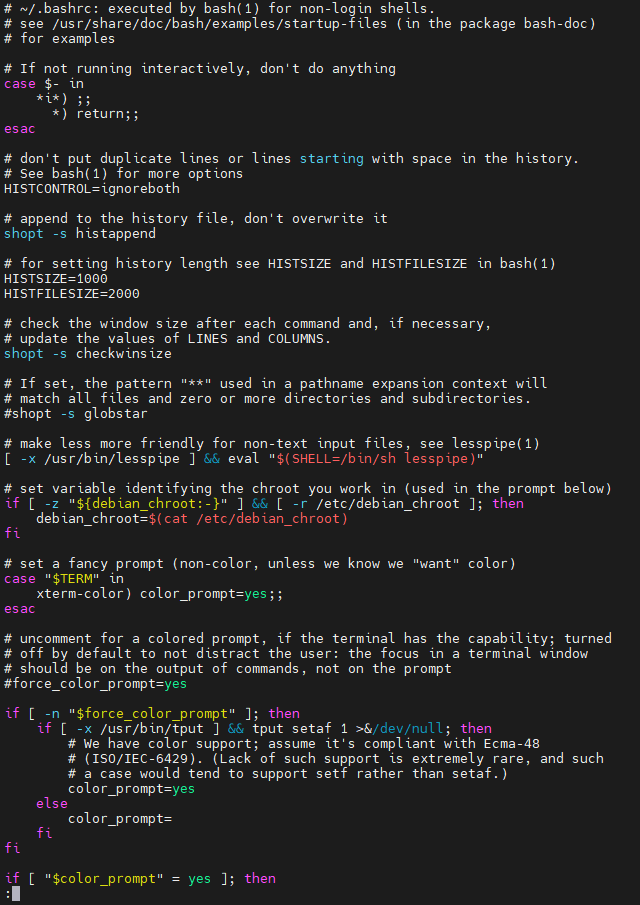
man less:

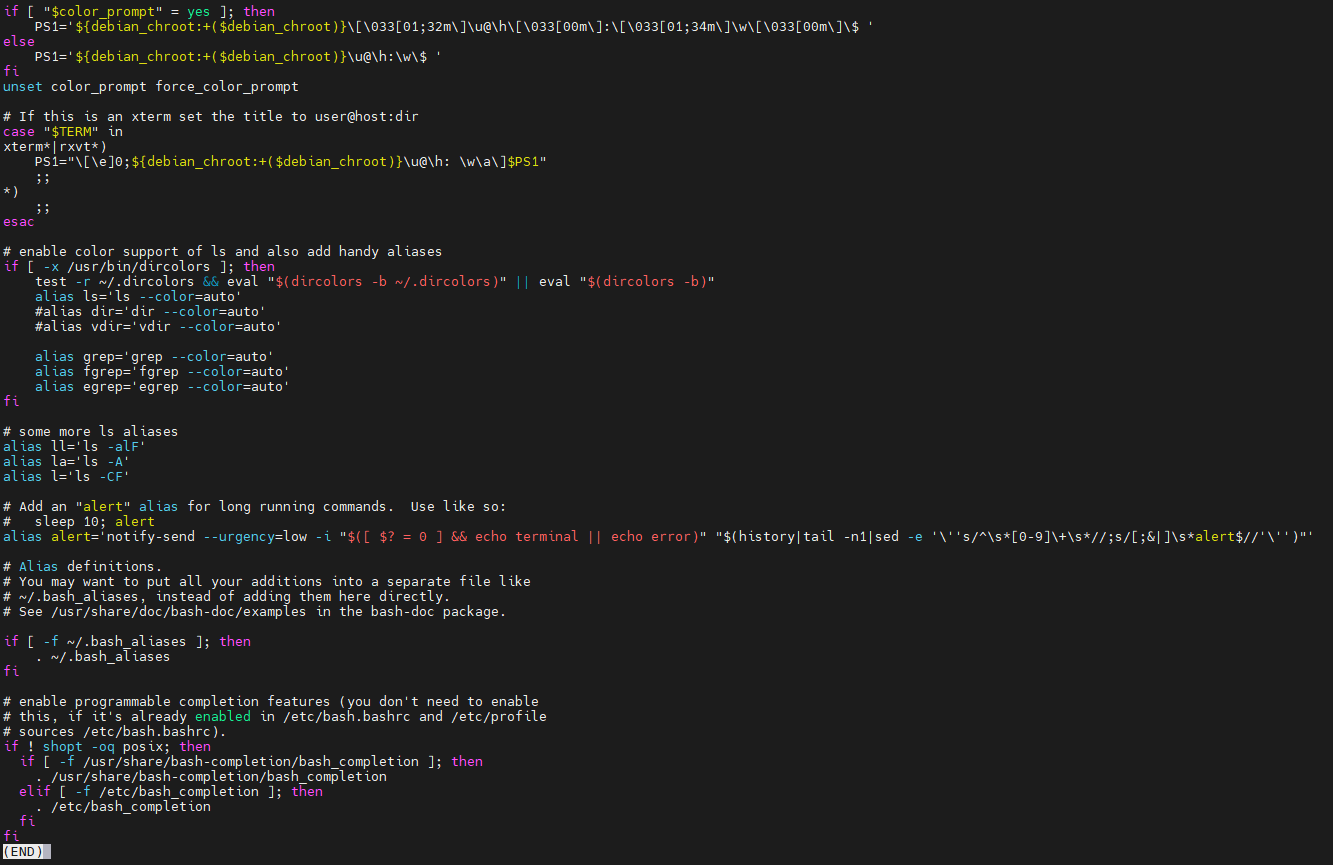


more ~/.bashrc:



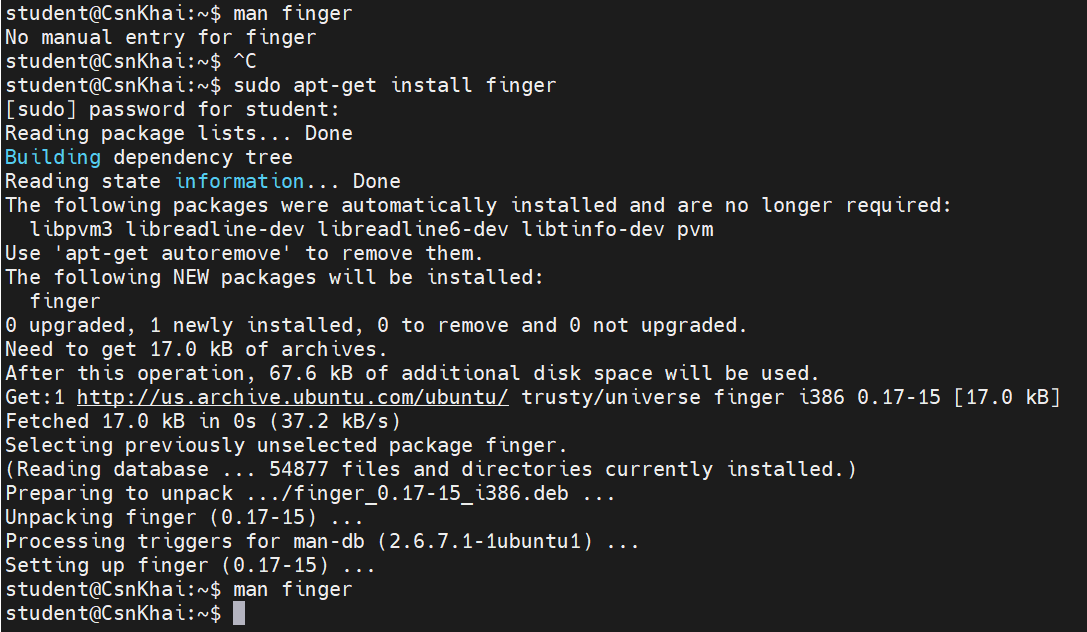
less ~/.bashrc:



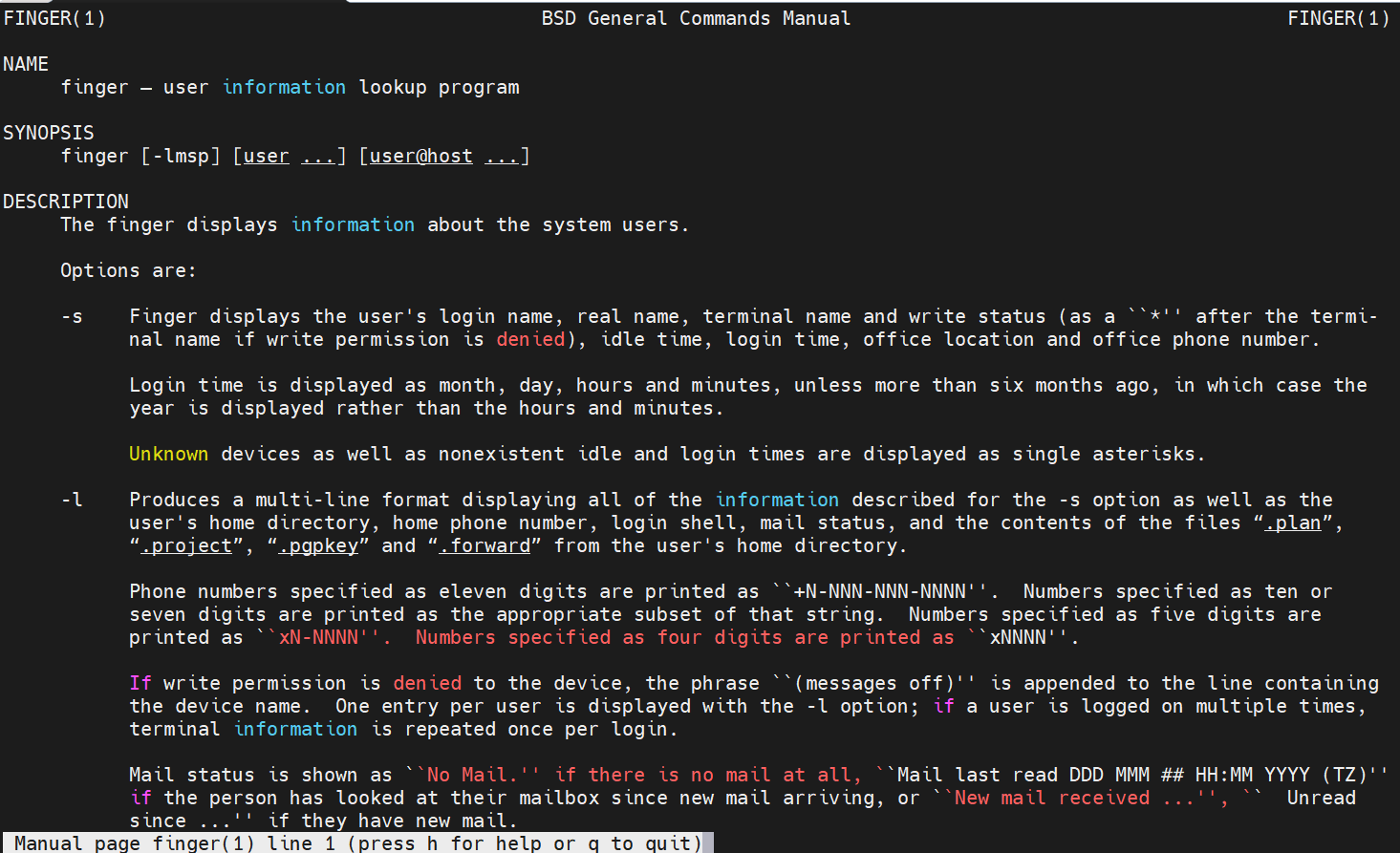


1. \* Describe in plans that you are working on laboratory work 1. Tip: You should read the documentation for the finger command.

Installing finger:

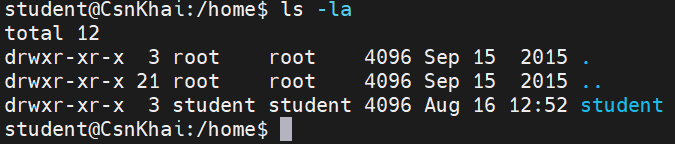


man finger:

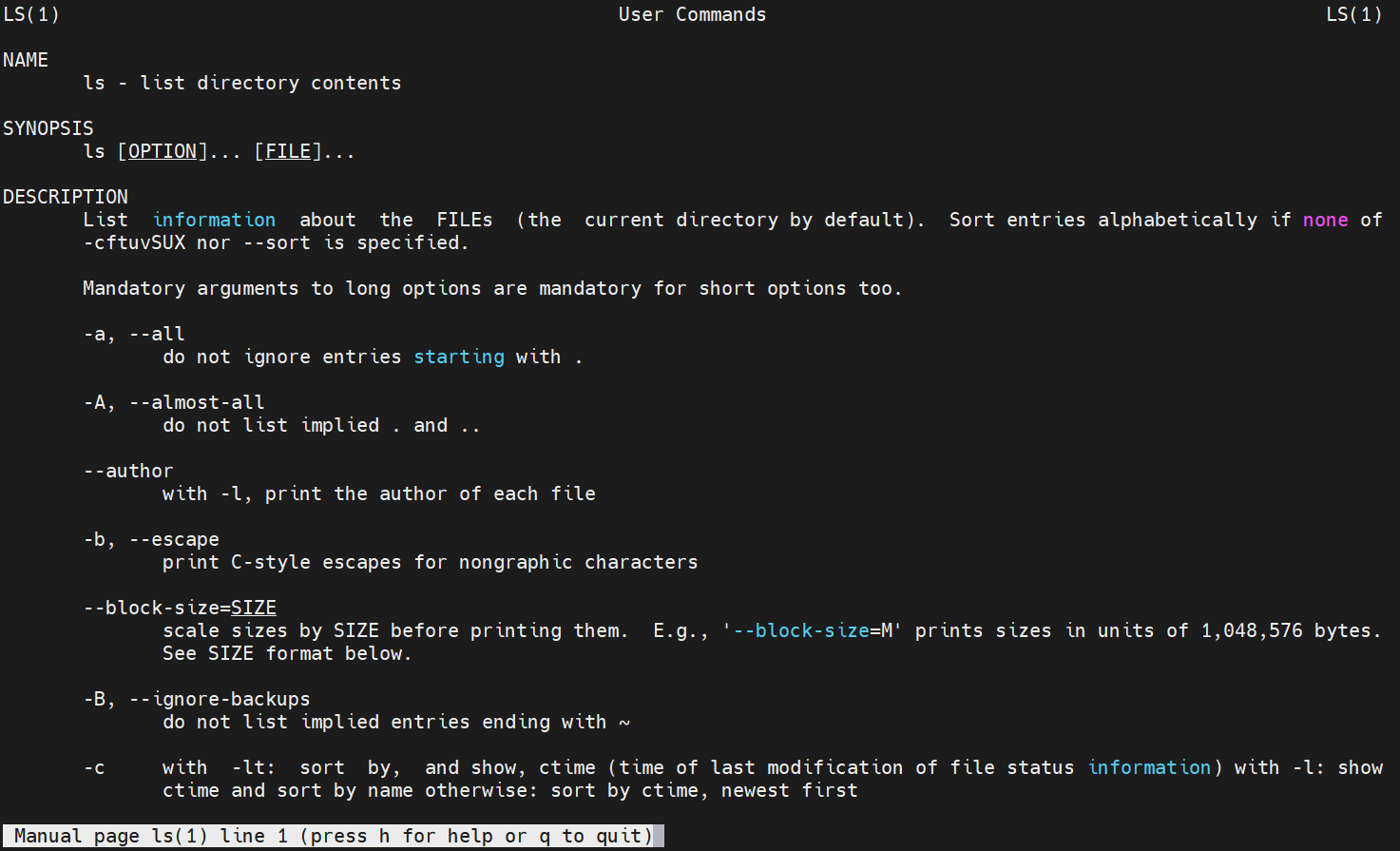


1. \* List the contents of the home directory using the ls command, define its files and directories. Hint: Use the help system to familiarize yourself with the ls command.

Contents of home directory using “ls –la”:



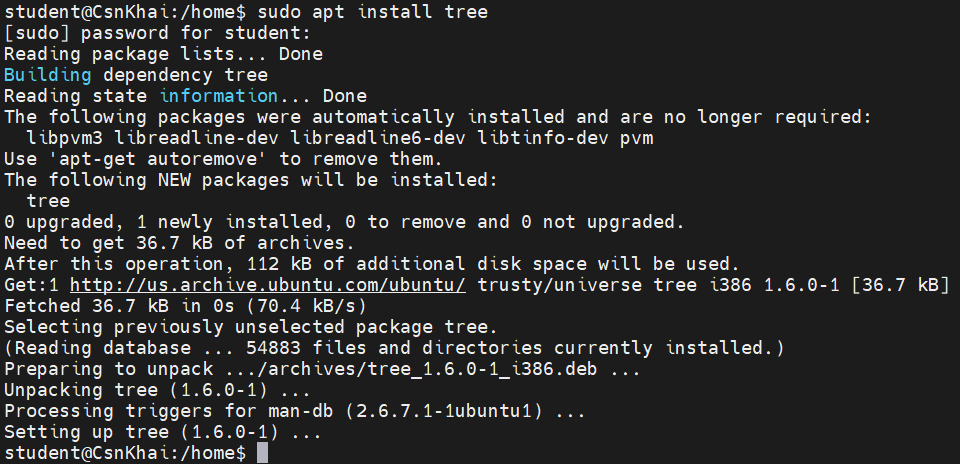
man ls:



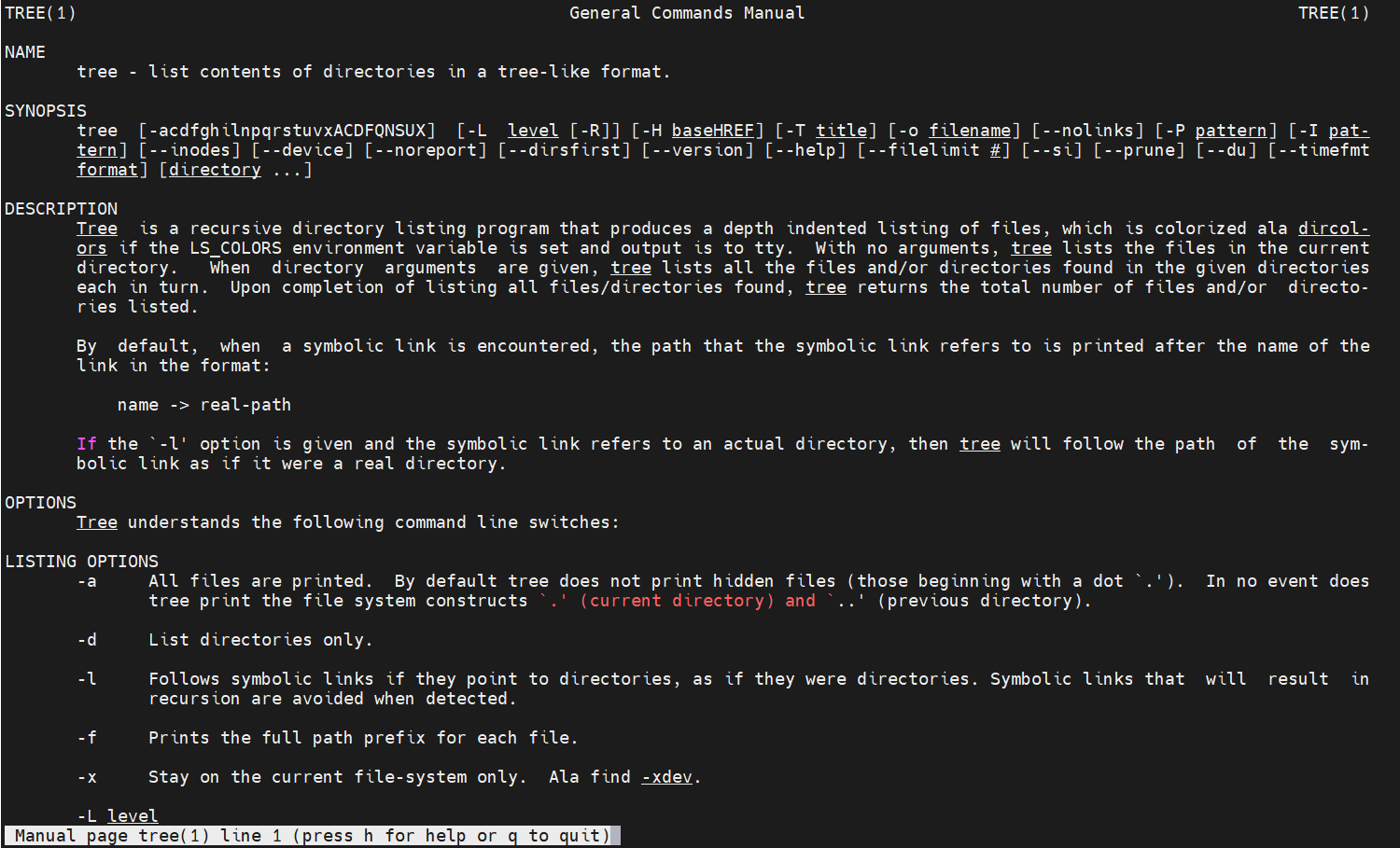
# Task1.Part2

1. Examine the **tree** command. Master the technique of applying a template, for example, display all files that contain a character **c**, or files that contain a specific sequence of characters. List subdirectories of the root directory up to and including the second nesting level.

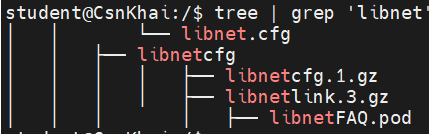
Install tree:



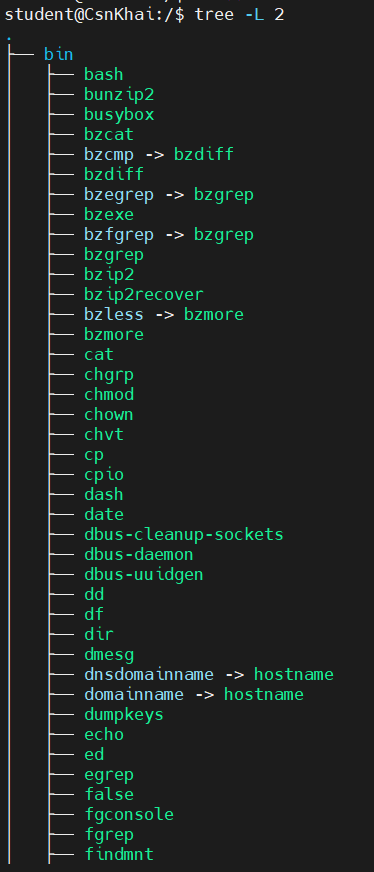
man tree:



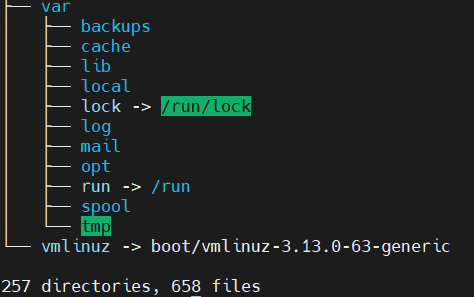
tree | grep ‘libnet’:



tree -L 2:



…



1. What command can be used to determine the type of file (for example, text or binary)? Give an example.

“file” can be used to determine the type of file:

Example:



or



1. Master the skills of navigating the file system using relative and absolute paths. How can you go back to your home directory from anywhere in the filesystem?

Navigation using relative path:



Navigation using absolute path:



To go back to your home directory you can use:



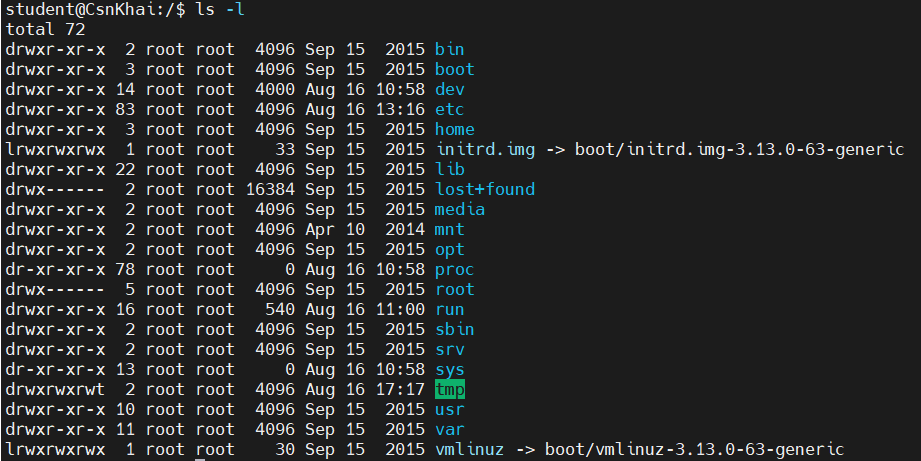
or



1. Become familiar with the various options for the **ls** command. Give examples of listing directories using different keys. Explain the information displayed on the terminal using the **-l** and **-a** switches.

The “-l” option displays the content of directories in a detailed long format. It provides information about file permissions, number of links, owner, group, file size, modification date, and filename.

Example:

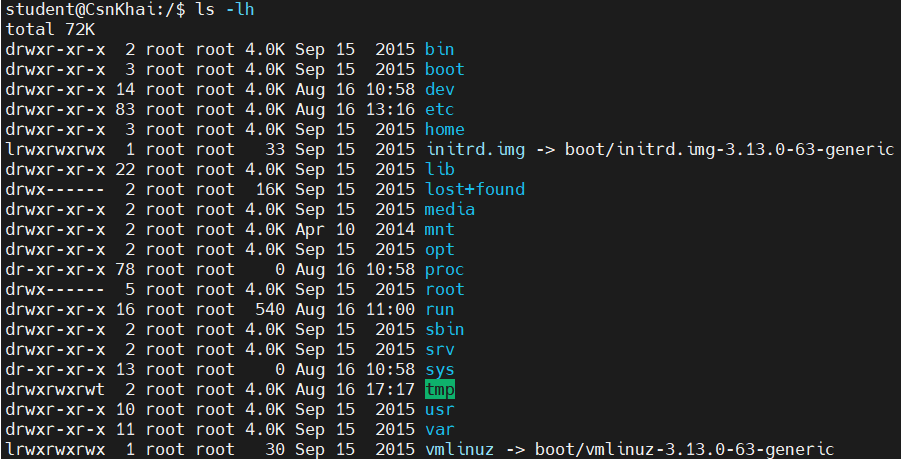


The “-a” option includes hidden files and directories (those starting with a dot .) in the listing. By default, these hidden files are not shown.

Example:

The “-h” option makes the file sizes more human-readable by using units like KB, MB, GB.

Example:



1. Perform the following sequence of operations:
   * create a subdirectory in the home directory:
   * in this subdirectory create a file containing information about directories located in the root directory (using I/O redirection operations):



* + view the created file:
  + copy the created file to your home directory using relative and absolute addressing.

Relative:



Absolute:



* + delete the previously created subdirectory with the file requesting removal:
  + delete the file copied to the home directory:

1. Perform the following sequence of operations:
   * create a subdirectory **test** in the home directory:
   * copy the **.bash\_history** file to this directory while changing its name to

# labwork2:

* + create a hard and soft link to the **labwork2** file in the test subdirectory:

Hard link:

Soft link:

* + how to define soft and hard link, what do these concepts:

A hard link is a reference to an inode (data structure on disk) of a file. When you create a hard link to a file, both the original file and the hard link share the same inode, which means they point to the same data blocks on the disk.

A symbolic link (also known as a symlink or soft link) is a separate file that contains a reference to another file's pathname. It acts as a pointer to the target file or directory, rather than sharing data blocks.

* + change the data by opening a symbolic link. What changes will happen and why:

opening link:





After opening I have changed original file.

* + rename the hard link file to **hard\_lnk\_labwork2**:

Name of hard link file has changed.

* + rename the soft link file to **symb\_lnk\_labwork2 file**:



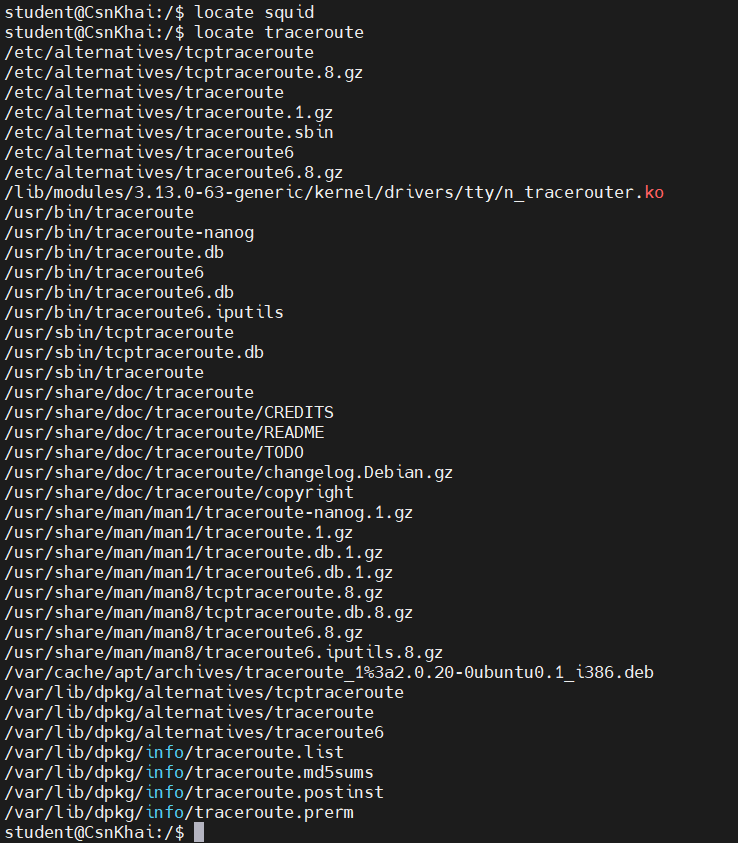
Only the link itself is renamed.

* + then delete the **labwork2**. What changes have occurred and why?

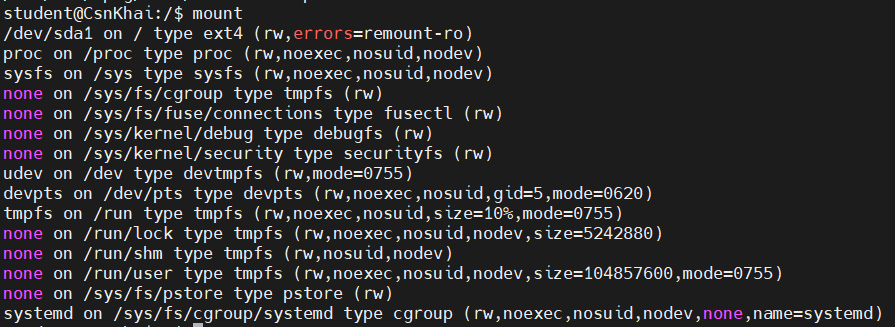
Deleting the original file will not immediately affect hard links, as they still reference the same data blocks.

Deleting the original file will cause symbolic links pointing to it to become invalid and unusable (dangling links).

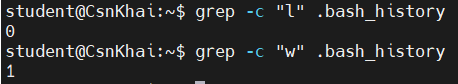
1. Using the locate utility, find all files that contain the squid and traceroute sequence:



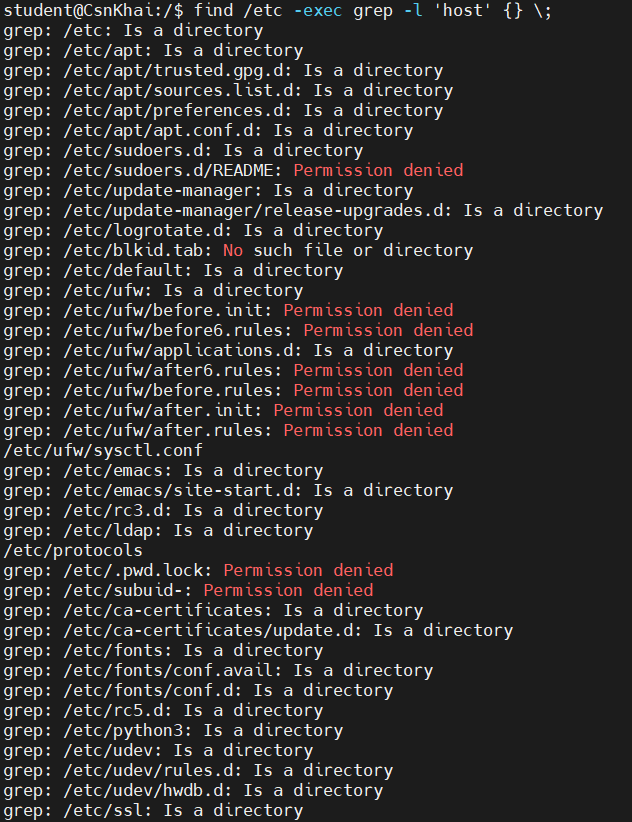
1. Determine which partitions are mounted in the system, as well as the types of these partitions:

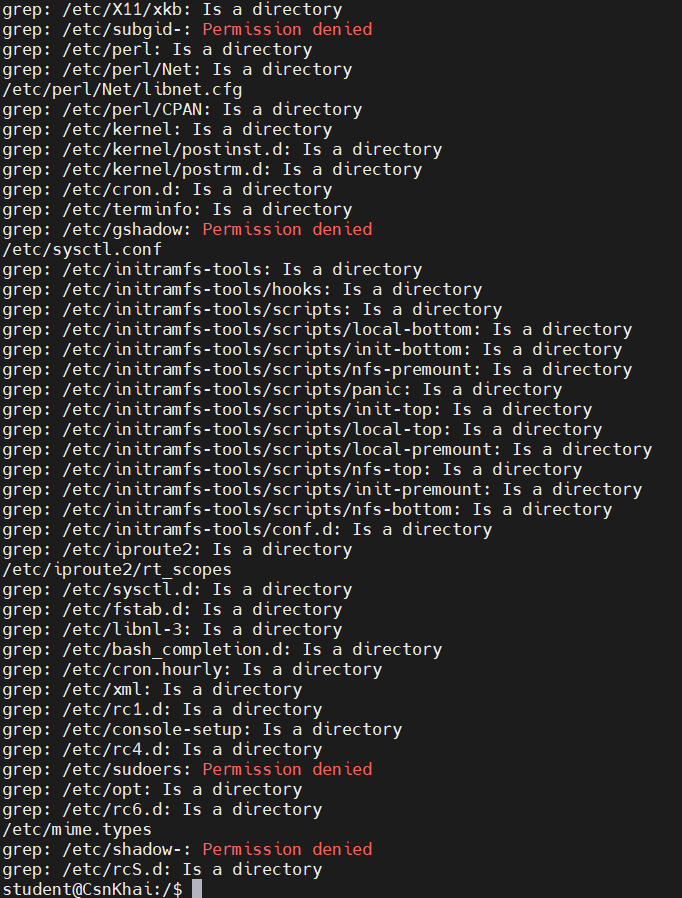


1. Count the number of lines containing a given sequence of characters in a given file:

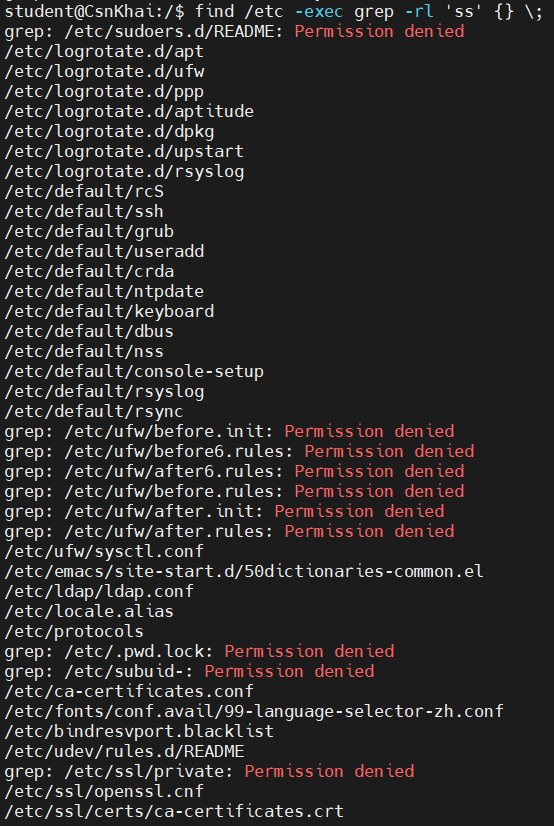


1. Using the **find** command, find all files in the /etc directory containing the

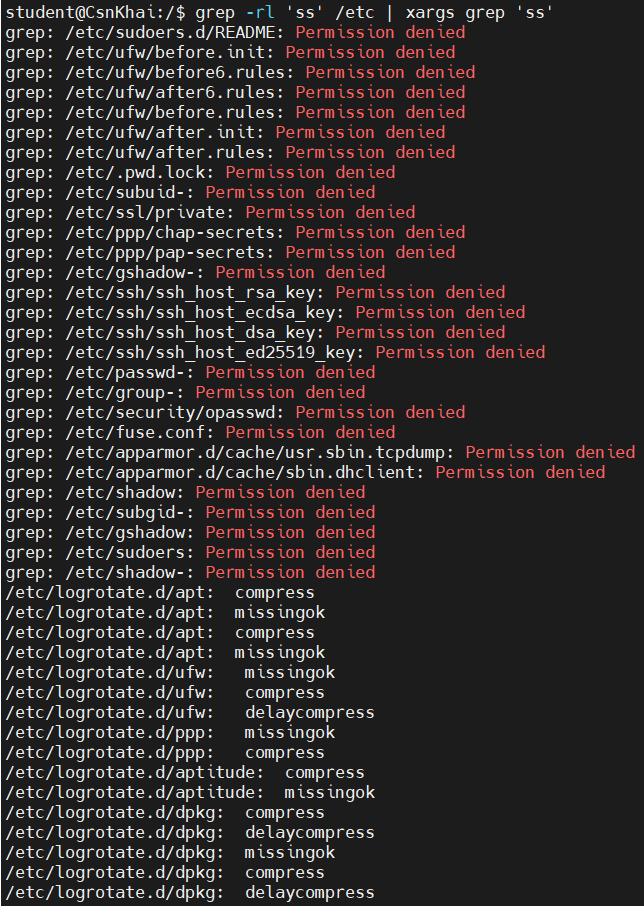
**host** character sequence:

…

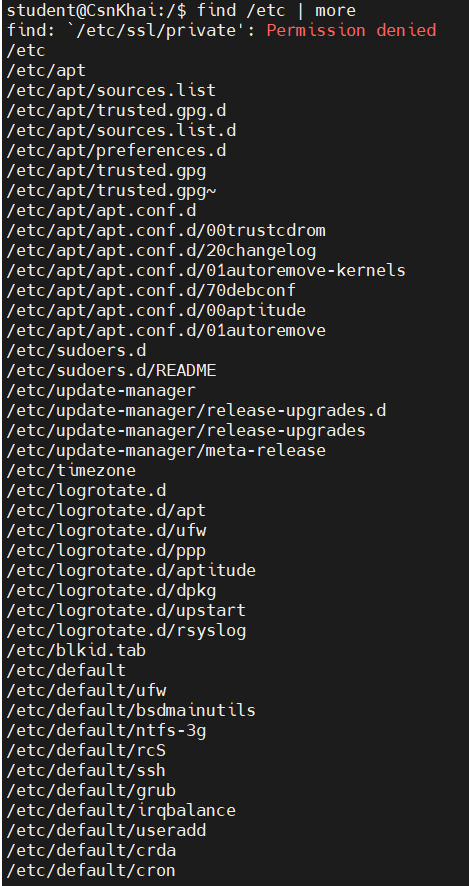
1. List all objects in /etc that contain the ss character sequence. How can I duplicate a similar command using a bunch of **grep**:



or



1. Organize a screen-by-screen print of the contents of the /etc directory. Hint: You must use stream redirection operations.



1. What are the types of devices and how to determine the type of device? Give examples.

Linux supports three types of hardware device: character, block and network. Character devices are read and written directly without buffering, for example the system's serial ports /dev/cua0 and /dev/cua1. Block devices can only be written to and read from in multiples of the block size, typically 512 or 1024 bytes. Block devices are accessed via the buffer cache and may be randomly accessed, that is to say, any block can be read or written no matter where it is on the device. Block devices can be accessed via their device special file but more commonly they are accessed via the file system. Only a block device can support a mounted file system. Network devices are accessed via the BSD socket interface and the networking subsytems.

Examples of Block Devices:

* /dev/sda: A typical hard drive.
* /dev/sdb: Another hard drive or storage device.
* /dev/nvme0n1: A NVMe-based SSD.

Examples of Character Devices:

* /dev/tty: Terminal device.
* /dev/ttyS0: Serial port device.
* /dev/zero: Null device (returns zeros when read).

1. How to determine the type of file in the system, what types of files are there?

With command “file” we can determine the type of file in the system.

Seven standard file types are:

* Regular
* Directory
* Symbolic link
* FIFO special
* Block special
* Character special
* Socket

1. \* List the first 5 directory files that were recently accessed in the **/etc**

directory.

