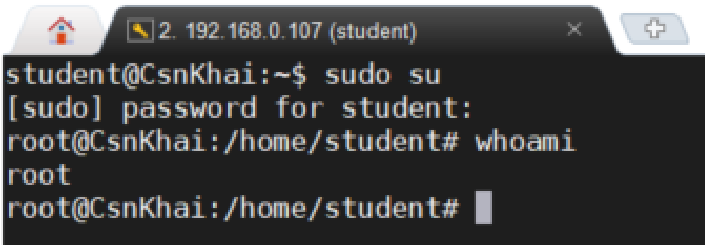
Task.Linux.1

Pashynskyi Maxim

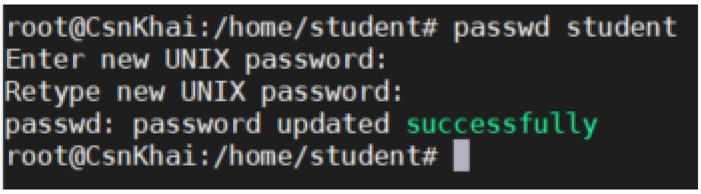
Task1.Part1

1) **Log in to the system as root.**



```
student@CsnKhai:~$ sudo su
[sudo] password for student:
root@CsnKhai:/home/student# whoami
root
root@CsnKhai:/home/student#
```

2) **Use the passwd command to change the password. Examine the basic parameters of the command. What system files does it change *?**

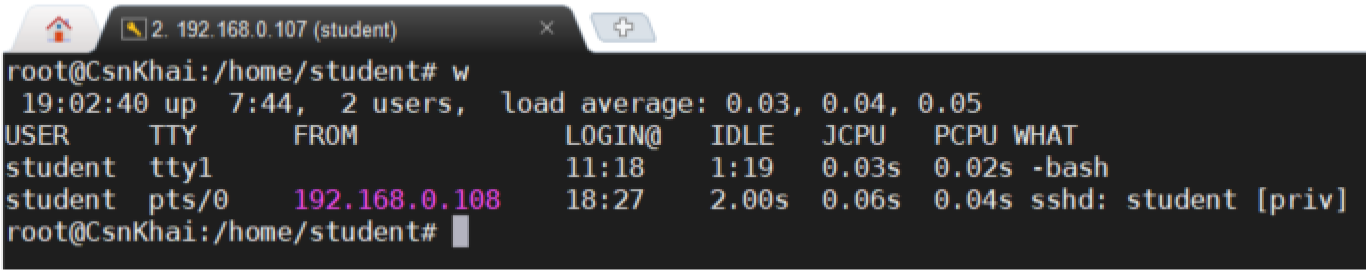


```
root@CsnKhai:/home/student# passwd student
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@CsnKhai:/home/student#
```

Basic parameters of the ***passwd*** command are different options (***-l*** for locking account, ***-u*** for unlocking, ***-d*** for password deletion, etc.) and user login. When a user without privileges uses ***passwd***, he can only change his own password. On the other hand, sudoer or root can change other users' passwords.

The ***passwd*** command execution changes */etc/shadow* file replacing user password hash stored there.

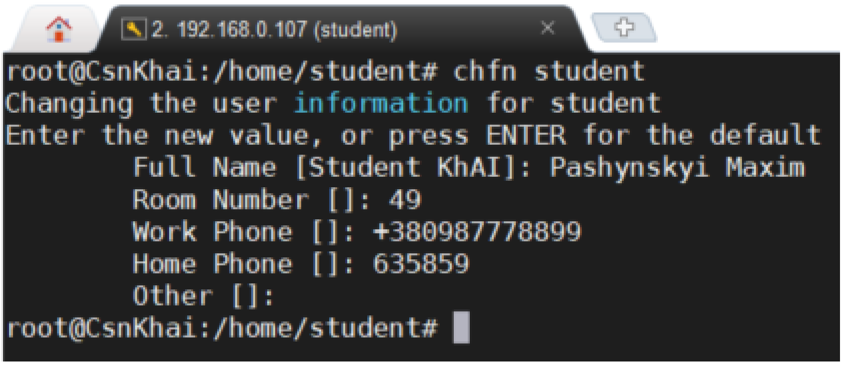
3) **Determine the users registered in the system, as well as what commands they execute. What additional information can be gleaned from the command execution?**



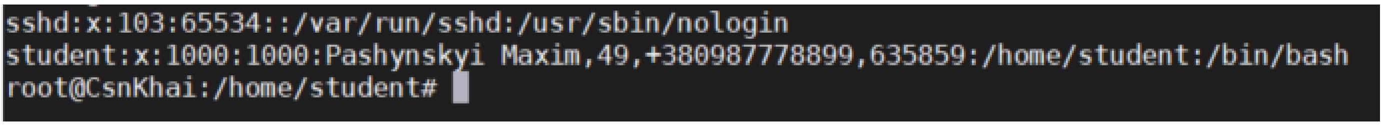
```
root@CsnKhai:/home/student# w
 19:02:40 up  7:44,  2 users,  load average: 0.03, 0.04, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
student  tty1                      11:18    1:19   0.03s  0.02s -bash
student  pts/0    192.168.0.108    18:27    2.00s  0.06s  0.04s sshd: student [priv]
root@CsnKhai:/home/student#
```

Additional info from ***w*** command execution: FROM (host IP-address), LOGIN@ (logon time), IDLE (inactivity time), JCPU (time used by all processes attached), PCPU (time used by process, displayed in WHAT field), WHAT (what user doing).

4) Change personal information about yourself.



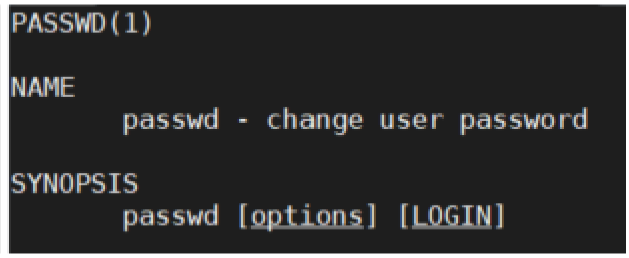
```
root@CsnKhai:/home/student# chfn student
Changing the user information for student
Enter the new value, or press ENTER for the default
        Full Name [Student KhAI]: Pashynskyi Maxim
        Room Number []: 49
        Work Phone []: +380987778899
        Home Phone []: 635859
        Other []:
root@CsnKhai:/home/student# 
```



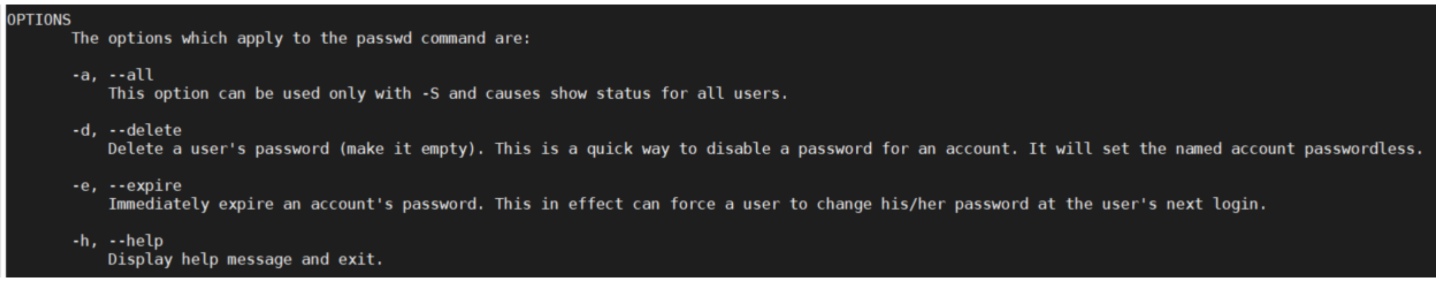
```
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
student:x:1000:1000:Pashynskyi Maxim,49,+380987778899,635859:/home/student:/bin/bash
root@CsnKhai:/home/student# 
```

5) Become familiar with Linux help system and the man and ifo commands. Get help on the previously discussed commands. Give examples.

man command:

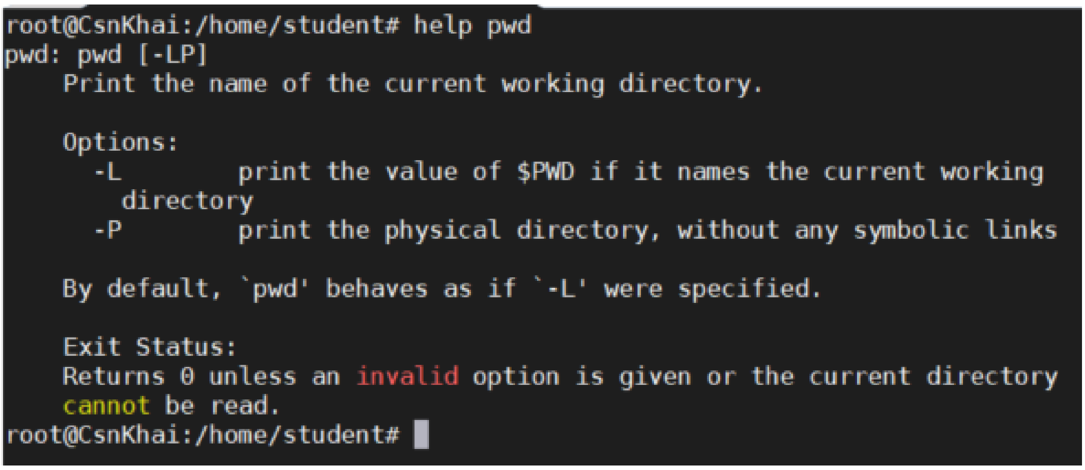


```
PASSWD(1)

NAME
       passwd - change user password

SYNOPSIS
       passwd [options] [LOGIN]
```



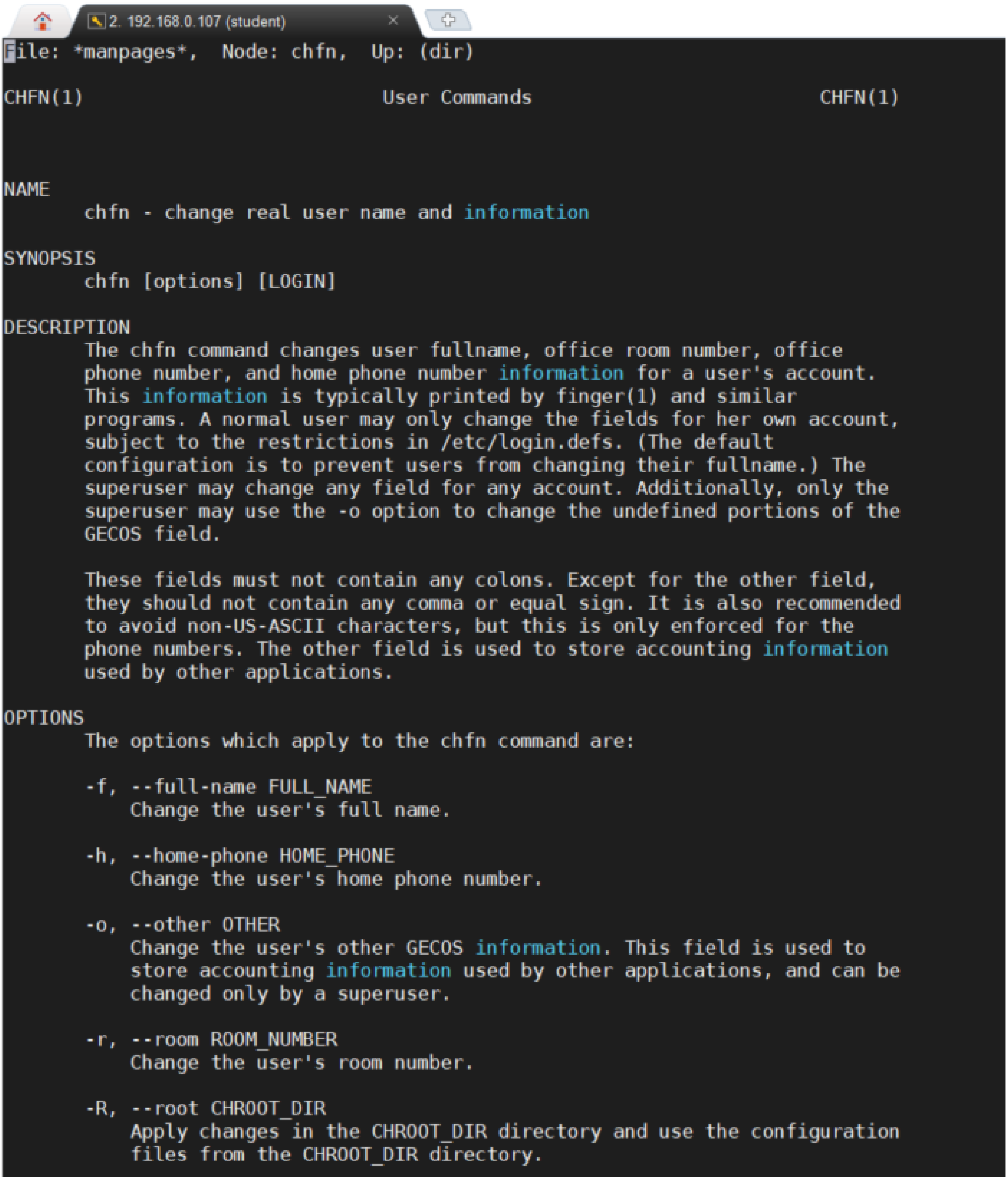
```
OPTIONS
       The options which apply to the passwd command are:

       -a, --all
           This option can be used only with -S and causes show status for all users.

       -d, --delete
           Delete a user's password (make it empty). This is a quick way to disable a password for an account. It will set the named account passwordless.

       -e, --expire
           Immediately expire an account's password. This in effect can force a user to change his/her password at the user's next login.

       -h, --help
           Display help message and exit.
```

help command:



```
root@CsnKhai:/home/student# help pwd
pwd: pwd [-LP]
    Print the name of the current working directory.

    Options:
      -L        print the value of $PWD if it names the current working
        directory
      -P        print the physical directory, without any symbolic links

    By default, `pwd' behaves as if `-L' were specified.

    Exit Status:
    Returns 0 unless an invalid option is given or the current directory
    cannot be read.
root@CsnKhai:/home/student# 
```

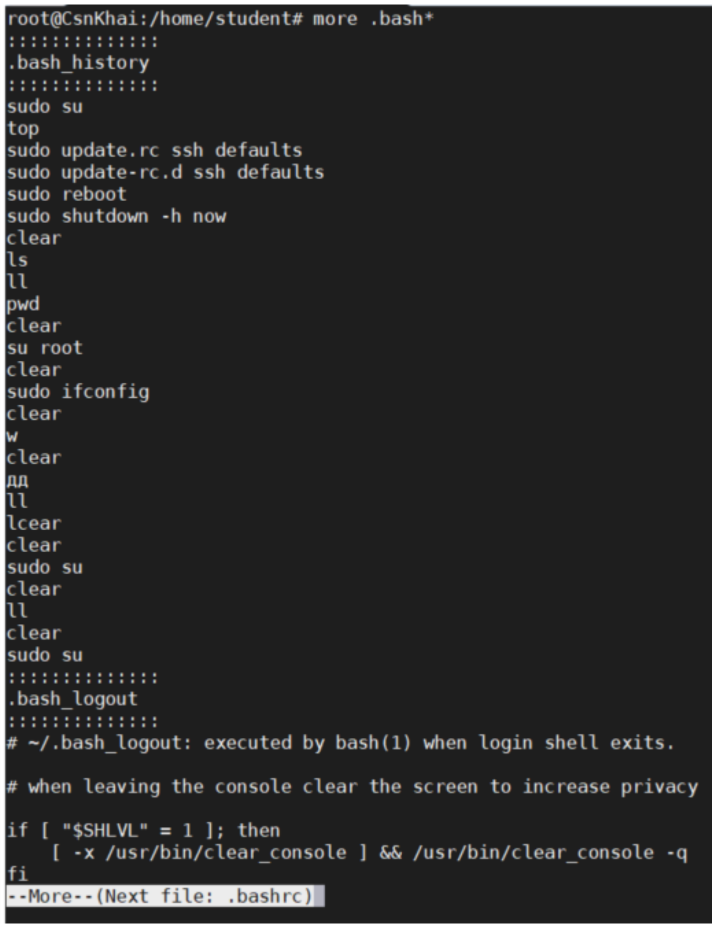
info command:



```
File: *manpages*,  Node: chfn,  Up: (dir)

CHFN(1)                          User Commands                          CHFN(1)



NAME
       chfn - change real user name and information

SYNOPSIS
       chfn [options] [LOGIN]

DESCRIPTION
       The chfn command changes user fullname, office room number, office
       phone number, and home phone number information for a user's account.
       This information is typically printed by finger(1) and similar
       programs. A normal user may only change the fields for her own account,
       subject to the restrictions in /etc/login.defs. (The default
       configuration is to prevent users from changing their fullname.) The
       superuser may change any field for any account. Additionally, only the
       superuser may use the -o option to change the undefined portions of the
       GECOS field.

       These fields must not contain any colons. Except for the other field,
       they should not contain any comma or equal sign. It is also recommended
       to avoid non-US-ASCII characters, but this is only enforced for the
       phone numbers. The other field is used to store accounting information
       used by other applications.

OPTIONS
       The options which apply to the chfn command are:

       -f, --full-name FULL_NAME
           Change the user's full name.

       -h, --home-phone HOME_PHONE
           Change the user's home phone number.

       -o, --other OTHER
           Change the user's other GECOS information. This field is used to
           store accounting information used by other applications, and can be
           changed only by a superuser.

       -r, --room ROOM_NUMBER
           Change the user's room number.

       -R, --root CHROOT_DIR
           Apply changes in the CHROOT_DIR directory and use the configuration
           files from the CHROOT_DIR directory.
```

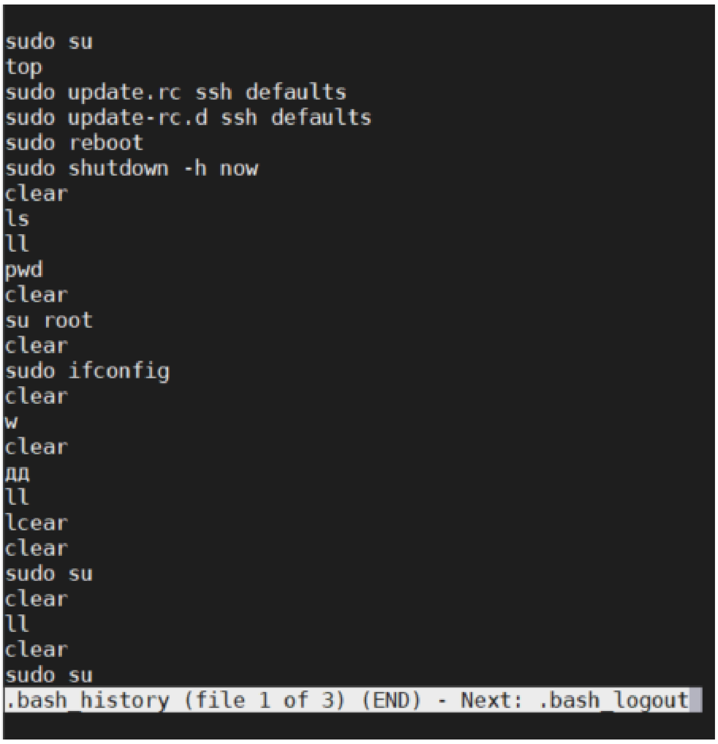
6) Explore the more and less commands using the help system. View the contents of files .bash* using commands.

*more .bash**



```
root@CsnKhai:/home/student# more .bash*
::::::::::::::
.bash_history
::::::::::::::
sudo su
top
sudo update.rc ssh defaults
sudo update-rc.d ssh defaults
sudo reboot
sudo shutdown -h now
clear
ls
ll
pwd
clear
su root
clear
sudo ifconfig
clear
w
clear
дд
ll
lcear
clear
sudo su
clear
ll
clear
sudo su
::::::::::::::
.bash_logout
::::::::::::::
# ~/.bash_logout: executed by bash(1) when login shell exits.

# when leaving the console clear the screen to increase privacy

if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
--More--(Next file: .bashrc)
```

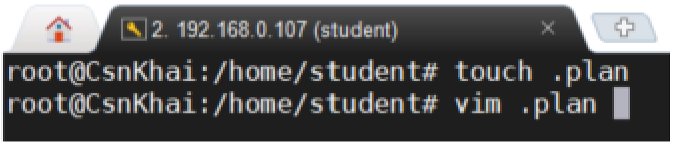
*less .bash**



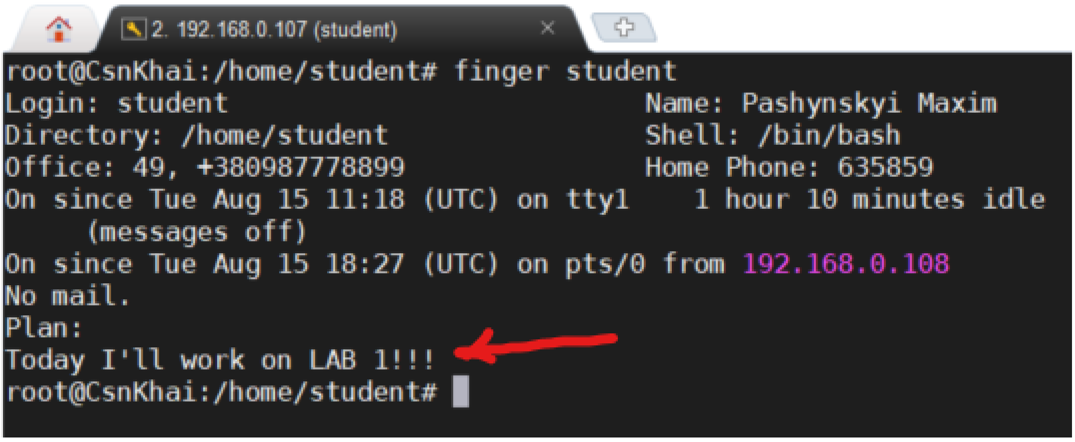
```
sudo su
top
sudo update.rc ssh defaults
sudo update-rc.d ssh defaults
sudo reboot
sudo shutdown -h now
clear
ls
ll
pwd
clear
su root
clear
sudo ifconfig
clear
w
clear
дд
ll
lcear
clear
sudo su
clear
ll
clear
sudo su
.bash_history (file 1 of 3) (END) - Next: .bash_logout
```

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

From the *finger* command man page I figured out that plans are being stored in the .plan file in users' home directory. So let's create and edit it:

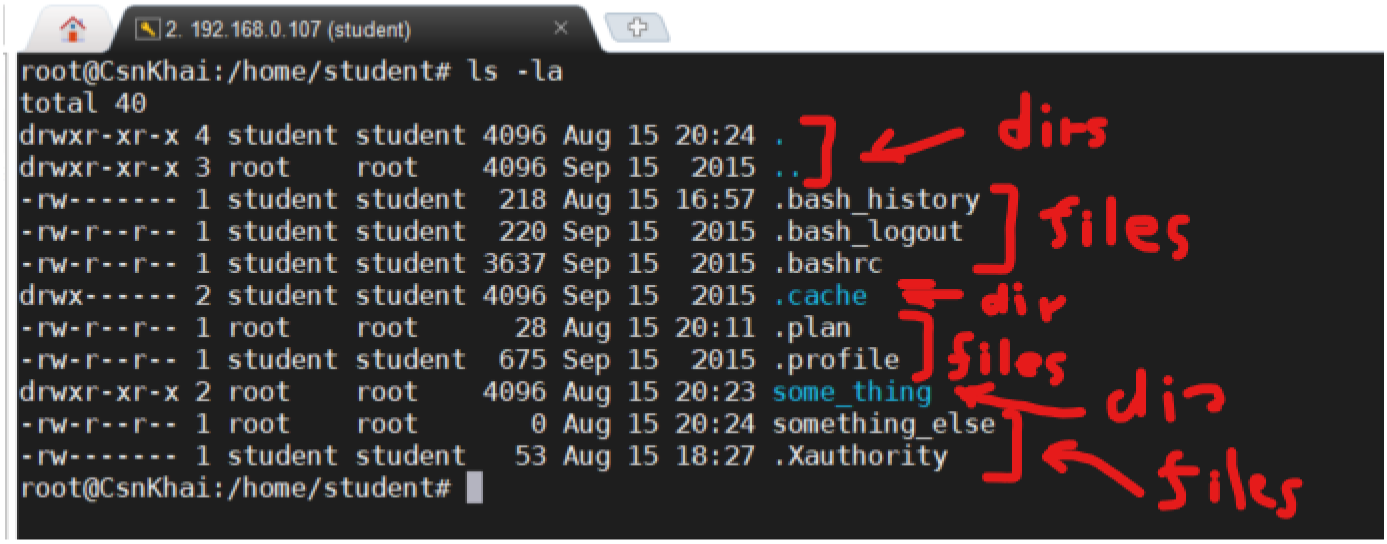


```
root@CsnKhai:/home/student# touch .plan
root@CsnKhai:/home/student# vim .plan
```



```
root@CsnKhai:/home/student# finger student
Login: student                          Name: Pashynskyi Maxim
Directory: /home/student                Shell: /bin/bash
Office: 49, +380987778899               Home Phone: 635859
On since Tue Aug 15 11:18 (UTC) on tty1    1 hour 10 minutes idle
     (messages off)
On since Tue Aug 15 18:27 (UTC) on pts/0 from 192.168.0.108
No mail.
Plan:
Today I'll work on LAB 1!!!
root@CsnKhai:/home/student#
```

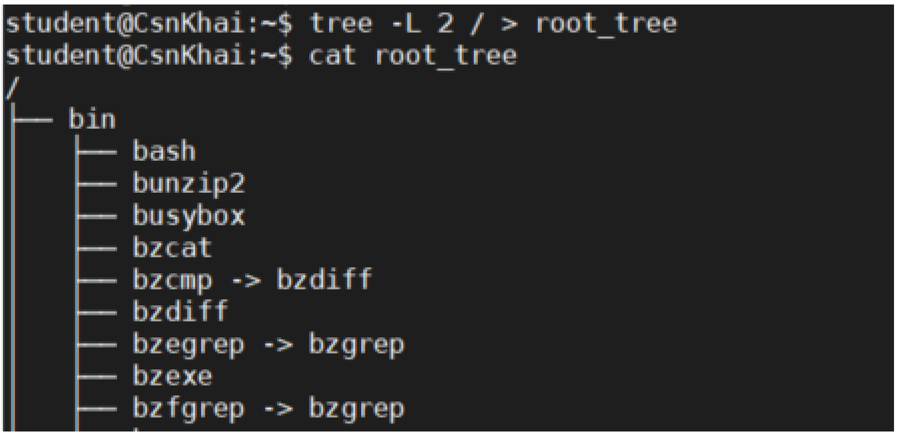
8) List the contents of the home directory using the ls command, define its files and directories. Hint: Use the help system to familiarize yourself with the ls command.



```
root@CsnKhai:/home/student# ls -la
total 40
drwxr-xr-x 4 student student 4096 Aug 15 20:24 .
drwxr-xr-x 3 root    root    4096 Sep 15  2015 ..
-rw------- 1 student student  218 Aug 15 16:57 .bash_history
-rw-r--r-- 1 student student  220 Sep 15  2015 .bash_logout
-rw-r--r-- 1 student student 3637 Sep 15  2015 .bashrc
drwx------ 2 student student 4096 Sep 15  2015 .cache
-rw-r--r-- 1 root    root      28 Aug 15 20:11 .plan
-rw-r--r-- 1 student student  675 Sep 15  2015 .profile
drwxr-xr-x 2 root    root    4096 Aug 15 20:23 some_thing
-rw-r--r-- 1 root    root       0 Aug 15 20:24 something_else
-rw------- 1 student student   53 Aug 15 18:27 .Xauthority
root@CsnKhai:/home/student#
```

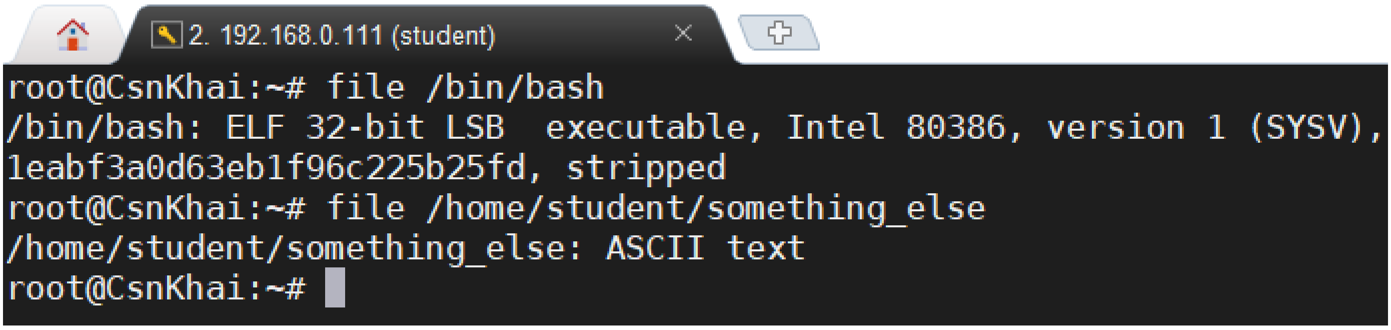
Task1.Part2

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.



```
student@CsnKhai:~$ tree -L 2 / > root_tree
student@CsnKhai:~$ cat root_tree
/
├── bin
│   ├── bash
│   ├── bunzip2
│   ├── busybox
│   ├── bzcat
│   ├── bzcmp -> bzdiff
│   ├── bzdiff
│   ├── bzegrep -> bzgrep
│   ├── bzexe
│   ├── bzfgrep -> bzgrep
```

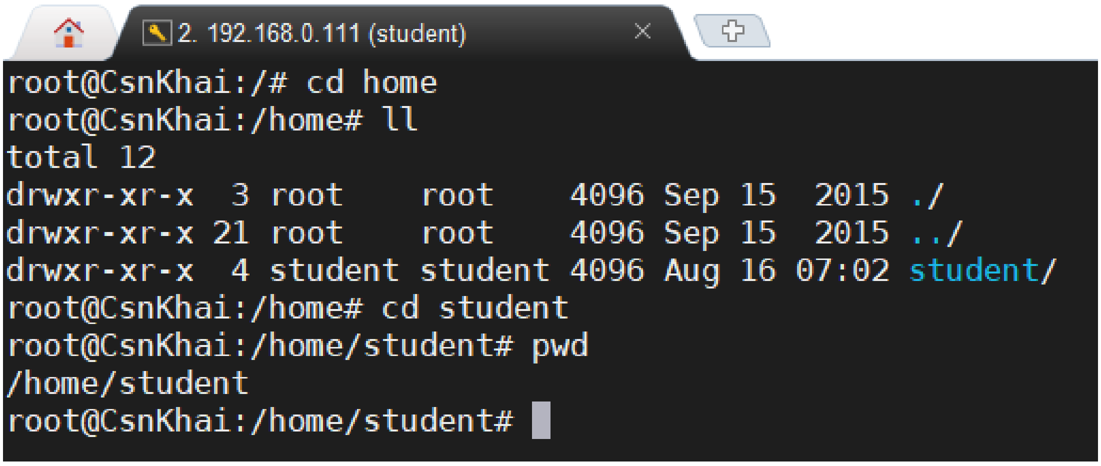
2) What command can be used to determine the type of file (for example, text or binary)? Give an example.



```
root@CsnKhai:~# file /bin/bash
/bin/bash: ELF 32-bit LSB  executable, Intel 80386, version 1 (SYSV),
1eabf3a0d63eb1f96c225b25fd, stripped
root@CsnKhai:~# file /home/student/something_else
/home/student/something_else: ASCII text
root@CsnKhai:~#
```

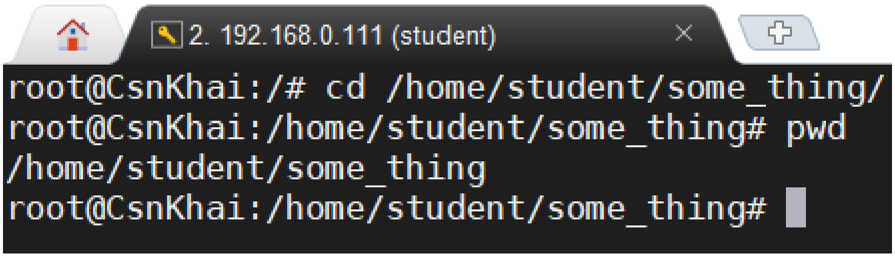
3) **Master the skills of navigating the file system using relative and absolute paths. How can you go back to your home directory from anywhere in the filesystem?**

Relative path navigation:



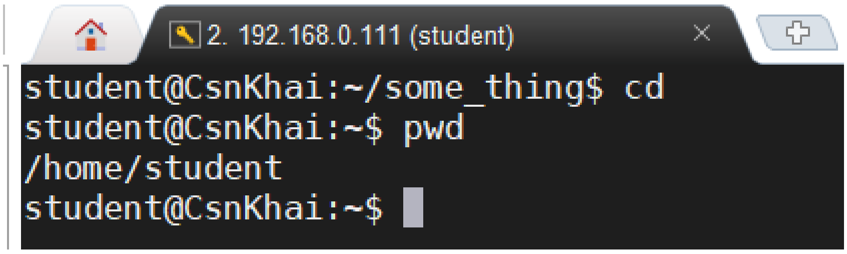
```
root@CsnKhai:/# cd home
root@CsnKhai:/home# ll
total 12
drwxr-xr-x  3 root    root    4096 Sep 15  2015 ./
drwxr-xr-x 21 root    root    4096 Sep 15  2015 ../
drwxr-xr-x  4 student student 4096 Aug 16 07:02 student/
root@CsnKhai:/home# cd student
root@CsnKhai:/home/student# pwd
/home/student
root@CsnKhai:/home/student#
```

Absolute path navigation:



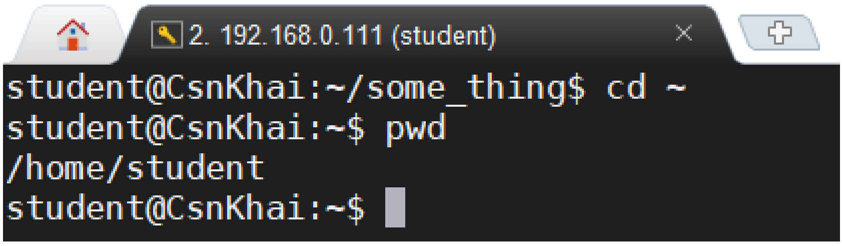
```
root@CsnKhai:/# cd /home/student/some_thing/
root@CsnKhai:/home/student/some_thing# pwd
/home/student/some_thing
root@CsnKhai:/home/student/some_thing#
```

Returning to home directory:



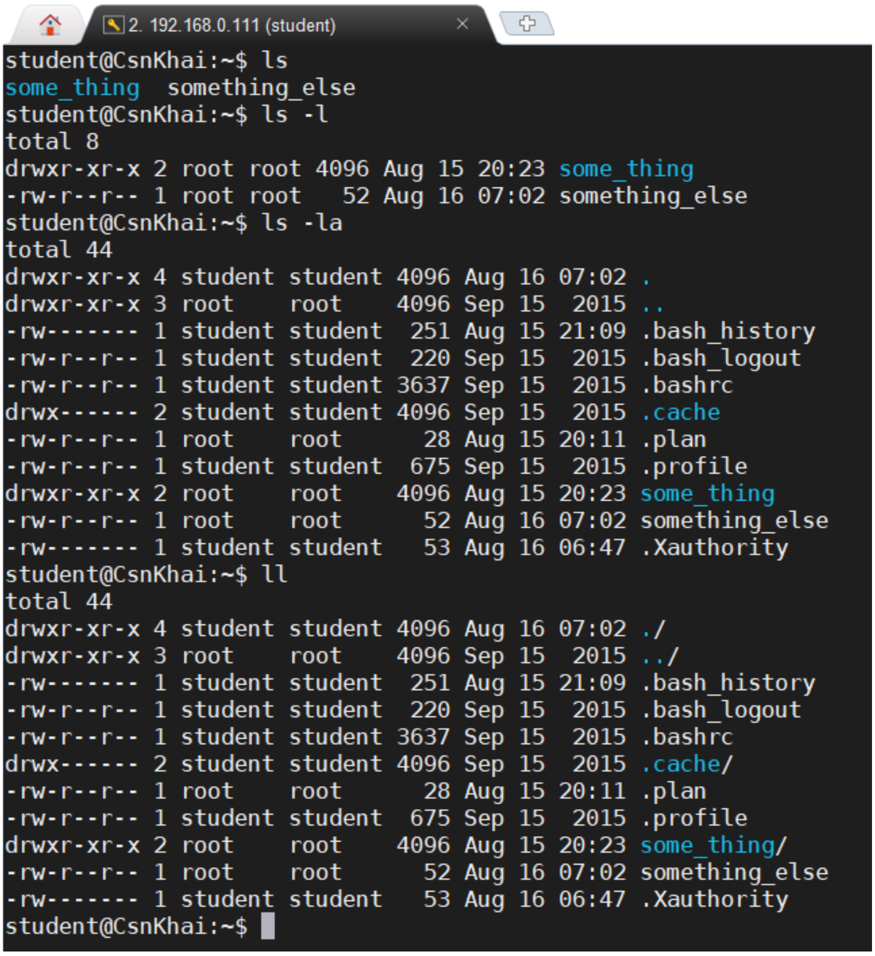
```
student@CsnKhai:~/some_thing$ cd
student@CsnKhai:~$ pwd
/home/student
student@CsnKhai:~$
```

or



```
student@CsnKhai:~/some_thing$ cd ~
student@CsnKhai:~$ pwd
/home/student
student@CsnKhai:~$
```

4) **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.**



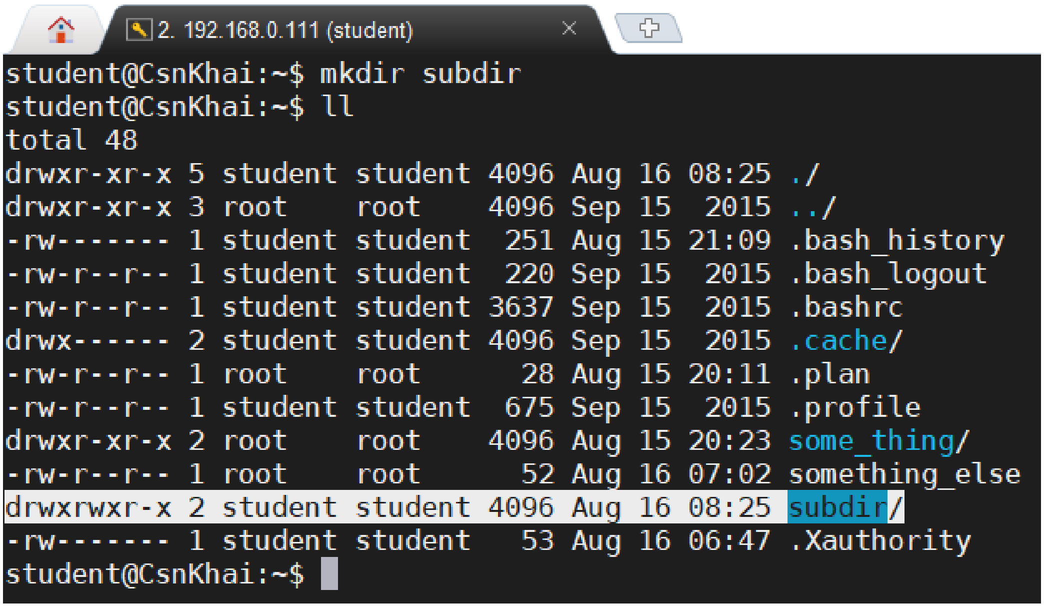
```
student@CsnKhai:~$ ls
some_thing  something_else
student@CsnKhai:~$ ls -l
total 8
drwxr-xr-x 2 root root 4096 Aug 15 20:23 some_thing
-rw-r--r-- 1 root root   52 Aug 16 07:02 something_else
student@CsnKhai:~$ ls -la
total 44
drwxr-xr-x 4 student student 4096 Aug 16 07:02 .
drwxr-xr-x 3 root    root    4096 Sep 15  2015 ..
-rw------- 1 student student  251 Aug 15 21:09 .bash_history
-rw-r--r-- 1 student student  220 Sep 15  2015 .bash_logout
-rw-r--r-- 1 student student 3637 Sep 15  2015 .bashrc
drwx------ 2 student student 4096 Sep 15  2015 .cache
-rw-r--r-- 1 root    root      28 Aug 15 20:11 .plan
-rw-r--r-- 1 student student  675 Sep 15  2015 .profile
drwxr-xr-x 2 root    root    4096 Aug 15 20:23 some_thing
-rw-r--r-- 1 root    root      52 Aug 16 07:02 something_else
-rw------- 1 student student   53 Aug 16 06:47 .Xauthority
student@CsnKhai:~$ ll
total 44
drwxr-xr-x 4 student student 4096 Aug 16 07:02 ./
drwxr-xr-x 3 root    root    4096 Sep 15  2015 ../
-rw------- 1 student student  251 Aug 15 21:09 .bash_history
-rw-r--r-- 1 student student  220 Sep 15  2015 .bash_logout
-rw-r--r-- 1 student student 3637 Sep 15  2015 .bashrc
drwx------ 2 student student 4096 Sep 15  2015 .cache/
-rw-r--r-- 1 root    root      28 Aug 15 20:11 .plan
-rw-r--r-- 1 student student  675 Sep 15  2015 .profile
drwxr-xr-x 2 root    root    4096 Aug 15 20:23 some_thing/
-rw-r--r-- 1 root    root      52 Aug 16 07:02 something_else
-rw------- 1 student student   53 Aug 16 06:47 .Xauthority
student@CsnKhai:~$
```

The *ls* command execution without any options will show only directory content names.

(-l) option is used to list directory content in long format (with permissions, owners, etc).

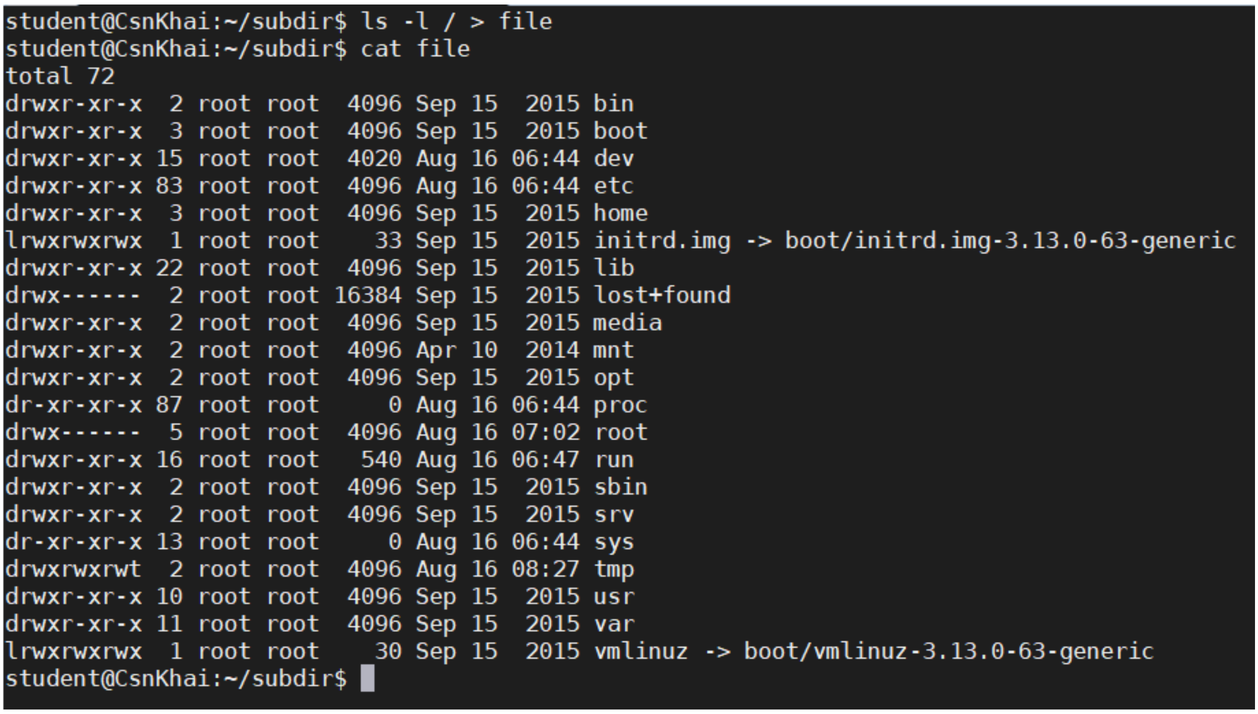
(-a) is used for listing all information including hidden files.

5) **Perform the following sequence of operations:**
   - create a subdirectory in the home directory;



```
student@CsnKhai:~$ mkdir subdir
student@CsnKhai:~$ ll
total 48
drwxr-xr-x 5 student student 4096 Aug 16 08:25 ./
drwxr-xr-x 3 root    root    4096 Sep 15  2015 ../
-rw------- 1 student student  251 Aug 15 21:09 .bash_history
-rw-r--r-- 1 student student  220 Sep 15  2015 .bash_logout
-rw-r--r-- 1 student student 3637 Sep 15  2015 .bashrc
drwx------ 2 student student 4096 Sep 15  2015 .cache/
-rw-r--r-- 1 root    root      28 Aug 15 20:11 .plan
-rw-r--r-- 1 student student  675 Sep 15  2015 .profile
drwxr-xr-x 2 root    root    4096 Aug 15 20:23 some_thing/
-rw-r--r-- 1 root    root      52 Aug 16 07:02 something_else
drwxrwxr-x 2 student student 4096 Aug 16 08:25 subdir/
-rw------- 1 student student   53 Aug 16 06:47 .Xauthority
student@CsnKhai:~$
```

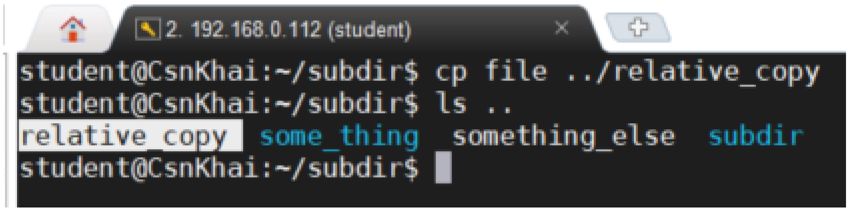
- in this subdirectory create a file containing information about directories located in the root directory (using I/O redirection operations);
- view the created file;



```
student@CsnKhai:~/subdir$ ls -l / > file
student@CsnKhai:~/subdir$ cat file
total 72
drwxr-xr-x  2 root root  4096 Sep 15  2015 bin
drwxr-xr-x  3 root root  4096 Sep 15  2015 boot
drwxr-xr-x 15 root root  4020 Aug 16 06:44 dev
drwxr-xr-x 83 root root  4096 Aug 16 06:44 etc
drwxr-xr-x  3 root root  4096 Sep 15  2015 home
lrwxrwxrwx  1 root root    33 Sep 15  2015 initrd.img -> boot/initrd.img-3.13.0-63-generic
drwxr-xr-x 22 root root  4096 Sep 15  2015 lib
drwx------  2 root root 16384 Sep 15  2015 lost+found
drwxr-xr-x  2 root root  4096 Sep 15  2015 media
drwxr-xr-x  2 root root  4096 Apr 10  2014 mnt
drwxr-xr-x  2 root root  4096 Sep 15  2015 opt
dr-xr-xr-x 87 root root     0 Aug 16 06:44 proc
drwx------  5 root root  4096 Aug 16 07:02 root
drwxr-xr-x 16 root root   540 Aug 16 06:47 run
drwxr-xr-x  2 root root  4096 Sep 15  2015 sbin
drwxr-xr-x  2 root root  4096 Sep 15  2015 srv
dr-xr-xr-x 13 root root     0 Aug 16 06:44 sys
drwxrwxrwt  2 root root  4096 Aug 16 08:27 tmp
drwxr-xr-x 10 root root  4096 Sep 15  2015 usr
drwxr-xr-x 11 root root  4096 Sep 15  2015 var
lrwxrwxrwx  1 root root    30 Sep 15  2015 vmlinuz -> boot/vmlinuz-3.13.0-63-generic
student@CsnKhai:~/subdir$ 
```

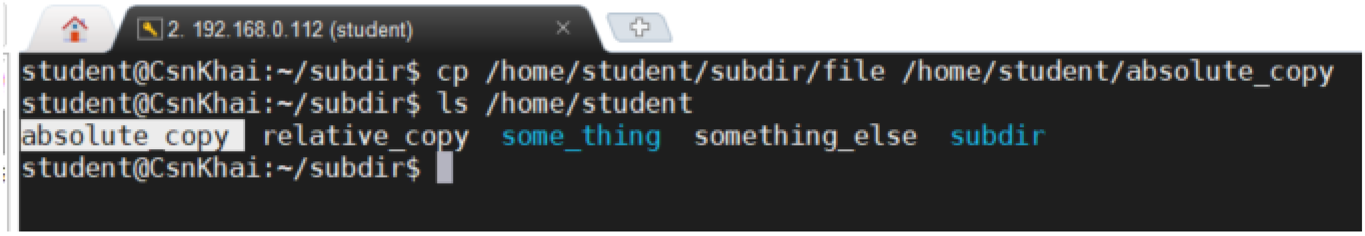
- copy the created file to your home directory using relative and absolute addressing;

relative addressing:



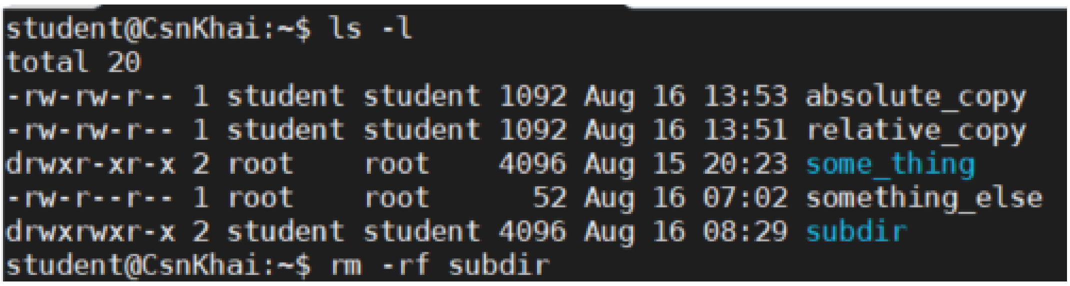
```
student@CsnKhai:~/subdir$ cp file ../relative_copy
student@CsnKhai:~/subdir$ ls ..
relative_copy  some_thing  something_else  subdir
student@CsnKhai:~/subdir$ 
```

absolute addressing:



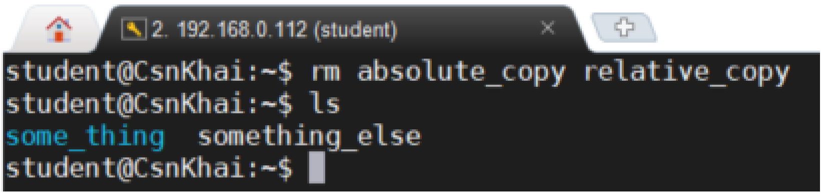
```
student@CsnKhai:~/subdir$ cp /home/student/subdir/file /home/student/absolute_copy
student@CsnKhai:~/subdir$ ls /home/student
absolute_copy  relative_copy  some_thing  something_else  subdir
student@CsnKhai:~/subdir$ 
```

- delete the previously created subdirectory with the file requesting removal;



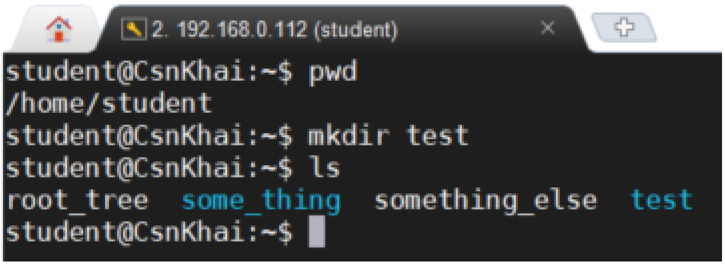
```
student@CsnKhai:~$ ls -l
total 20
-rw-rw-r-- 1 student student 1092 Aug 16 13:53 absolute_copy
-rw-rw-r-- 1 student student 1092 Aug 16 13:51 relative_copy
drwxr-xr-x 2 root    root    4096 Aug 15 20:23 some_thing
-rw-r--r-- 1 root    root      52 Aug 16 07:02 something_else
drwxrwxr-x 2 student student 4096 Aug 16 08:29 subdir
student@CsnKhai:~$ rm -rf subdir
```

- delete the file copied to the home directory.



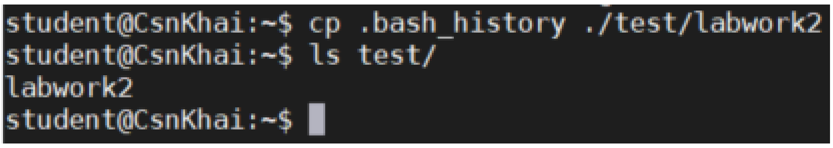
```
student@CsnKhai:~$ rm absolute_copy relative_copy
student@CsnKhai:~$ ls
some_thing  something_else
student@CsnKhai:~$ 
```

6) Perform the following sequence of operations:
   - create a subdirectory test in the home directory;



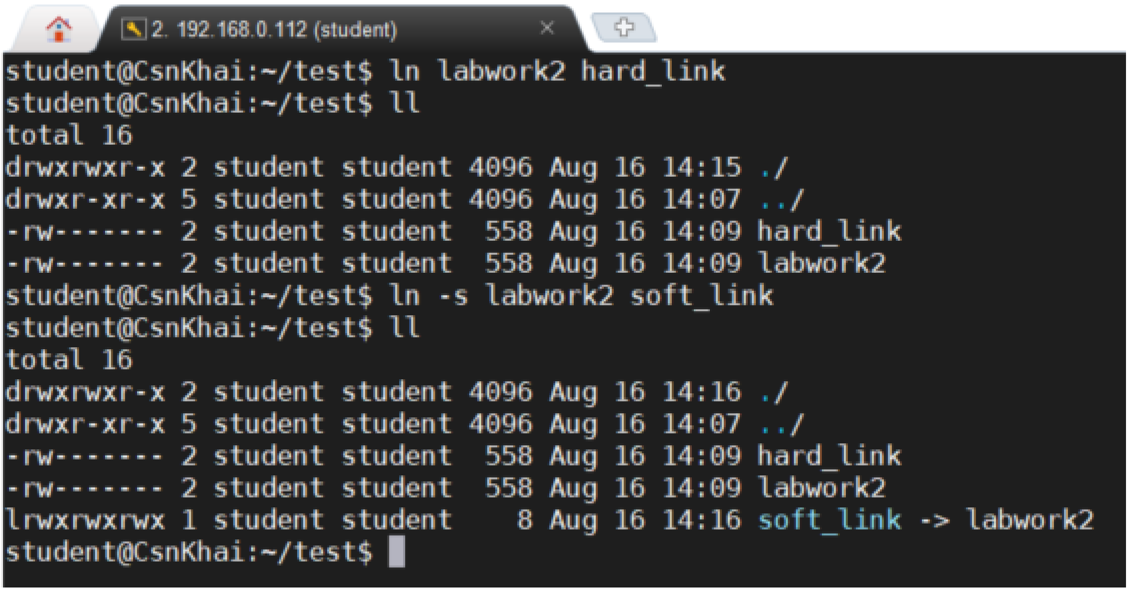
```
student@CsnKhai:~$ pwd
/home/student
student@CsnKhai:~$ mkdir test
student@CsnKhai:~$ ls
root_tree  some_thing  something_else  test
student@CsnKhai:~$ 
```

- copy the .bash_history file to this directory while changing its name to labwork2;



```
student@CsnKhai:~$ cp .bash_history ./test/labwork2
student@CsnKhai:~$ ls test/
labwork2
student@CsnKhai:~$ 
```

- create a hard and soft link to the labwork2 file in the test subdirectory;



```
student@CsnKhai:~/test$ ln labwork2 hard_link
student@CsnKhai:~/test$ ll
total 16
drwxrwxr-x 2 student student 4096 Aug 16 14:15 ./
drwxr-xr-x 5 student student 4096 Aug 16 14:07 ../
-rw------- 2 student student  558 Aug 16 14:09 hard_link
-rw------- 2 student student  558 Aug 16 14:09 labwork2
student@CsnKhai:~/test$ ln -s labwork2 soft_link
student@CsnKhai:~/test$ ll
total 16
drwxrwxr-x 2 student student 4096 Aug 16 14:16 ./
drwxr-xr-x 5 student student 4096 Aug 16 14:07 ../
-rw------- 2 student student  558 Aug 16 14:09 hard_link
-rw------- 2 student student  558 Aug 16 14:09 labwork2
lrwxrwxrwx 1 student student    8 Aug 16 14:16 soft_link -> labwork2
student@CsnKhai:~/test$ 
```

- how to define soft and hard link, what do these concepts;

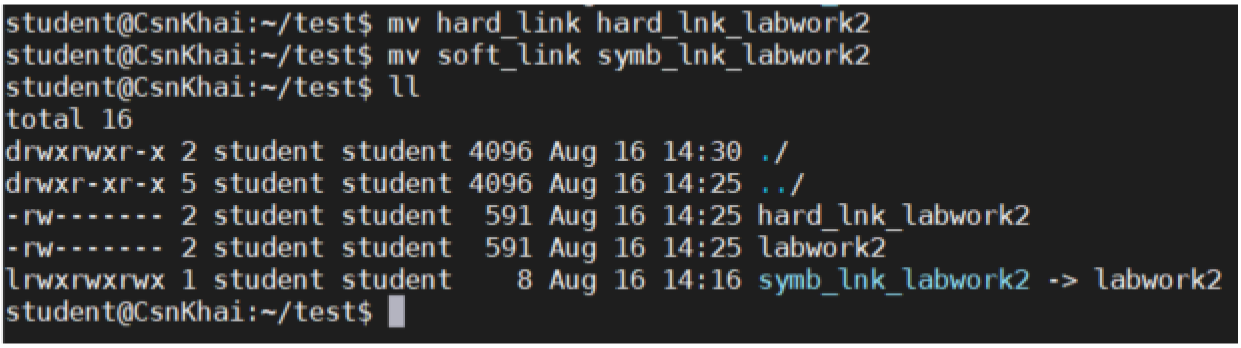
Hard link points a filename to data on a storage device.

Soft link points a filename to another filename, which then points to data on a storage device.

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

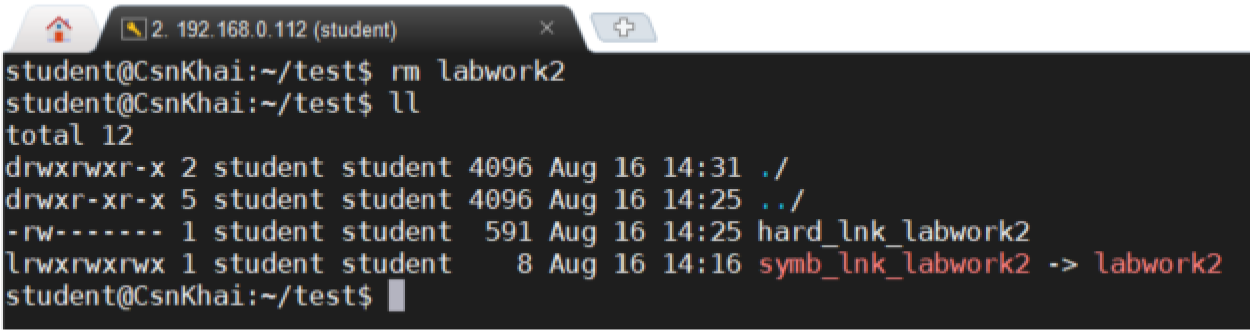
If I open a symbolic link with vim and change its content, changes will be made to the original file, because symlink just points to it.

- **rename the hard link file to hard_lnk_labwork2;**
- **rename the soft link file to symb_lnk_labwork2 file;**



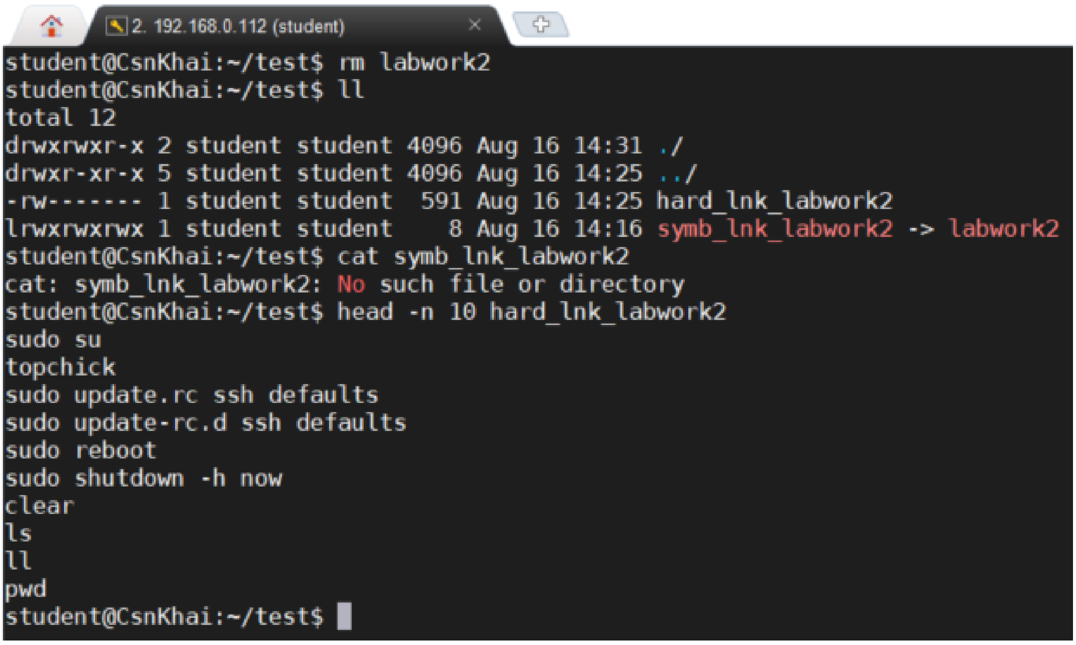
```
student@CsnKhai:~/test$ mv hard_link hard_lnk_labwork2
student@CsnKhai:~/test$ mv soft_link symb_lnk_labwork2
student@CsnKhai:~/test$ ll
total 16
drwxrwxr-x 2 student student 4096 Aug 16 14:30 ./
drwxr-xr-x 5 student student 4096 Aug 16 14:25 ../
-rw------- 2 student student  591 Aug 16 14:25 hard_lnk_labwork2
-rw------- 2 student student  591 Aug 16 14:25 labwork2
lrwxrwxrwx 1 student student    8 Aug 16 14:16 symb_lnk_labwork2 -> labwork2
student@CsnKhai:~/test$
```

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



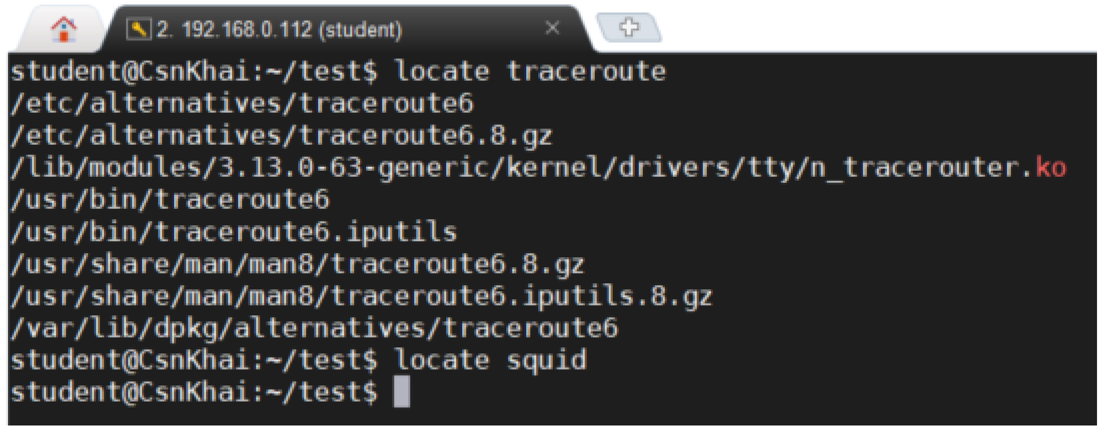
```
student@CsnKhai:~/test$ rm labwork2
student@CsnKhai:~/test$ ll
total 12
drwxrwxr-x 2 student student 4096 Aug 16 14:31 ./
drwxr-xr-x 5 student student 4096 Aug 16 14:25 ../
-rw------- 1 student student  591 Aug 16 14:25 hard_lnk_labwork2
lrwxrwxrwx 1 student student    8 Aug 16 14:16 symb_lnk_labwork2 -> labwork2
student@CsnKhai:~/test$
```

After deletion of the original file symlinks stop working and become broken in contrast to hard links, which continue to exist as separate data on disk.



```
student@CsnKhai:~/test$ rm labwork2
student@CsnKhai:~/test$ ll
total 12
drwxrwxr-x 2 student student 4096 Aug 16 14:31 ./
drwxr-xr-x 5 student student 4096 Aug 16 14:25 ../
-rw------- 1 student student  591 Aug 16 14:25 hard_lnk_labwork2
lrwxrwxrwx 1 student student    8 Aug 16 14:16 symb_lnk_labwork2 -> labwork2
student@CsnKhai:~/test$ cat symb_lnk_labwork2
cat: symb_lnk_labwork2: No such file or directory
student@CsnKhai:~/test$ head -n 10 hard_lnk_labwork2
sudo su
topchick
sudo update.rc ssh defaults
sudo update-rc.d ssh defaults
sudo reboot
sudo shutdown -h now
clear
ls
ll
pwd
student@CsnKhai:~/test$
```

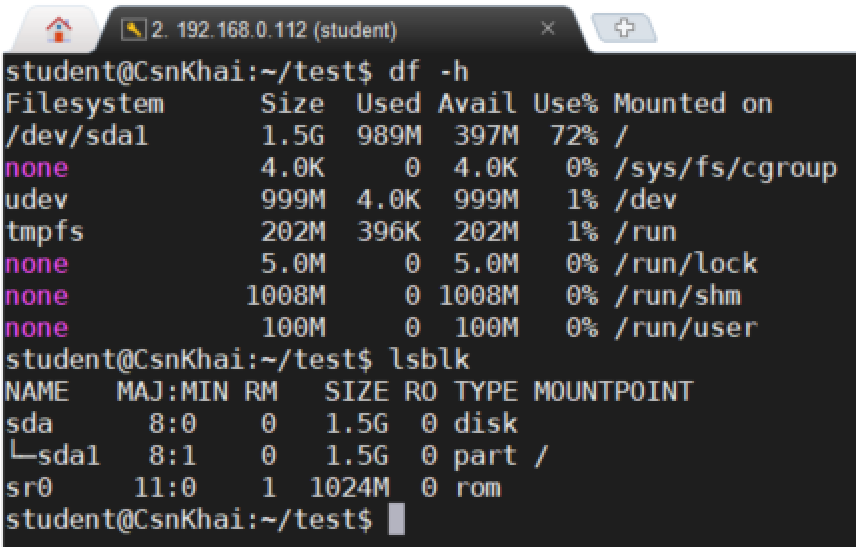
7) **Using the locate utility, find all files that contain the squid and traceroute sequence.**



```
student@CsnKhai:~/test$ locate traceroute
/etc/alternatives/traceroute6
/etc/alternatives/traceroute6.8.gz
/lib/modules/3.13.0-63-generic/kernel/drivers/tty/n_tracerouter.ko
/usr/bin/traceroute6
/usr/bin/traceroute6.iputils
/usr/share/man/man8/traceroute6.8.gz
/usr/share/man/man8/traceroute6.iputils.8.gz
/var/lib/dpkg/alternatives/traceroute6
student@CsnKhai:~/test$ locate squid
student@CsnKhai:~/test$
```

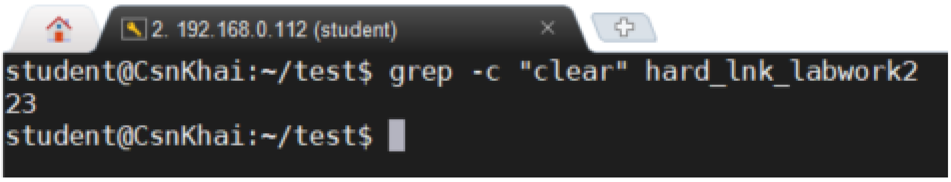
There are no files with squid sequence in the file system.

8) **Determine which partitions are mounted in the system, as well as the types of these partitions.**



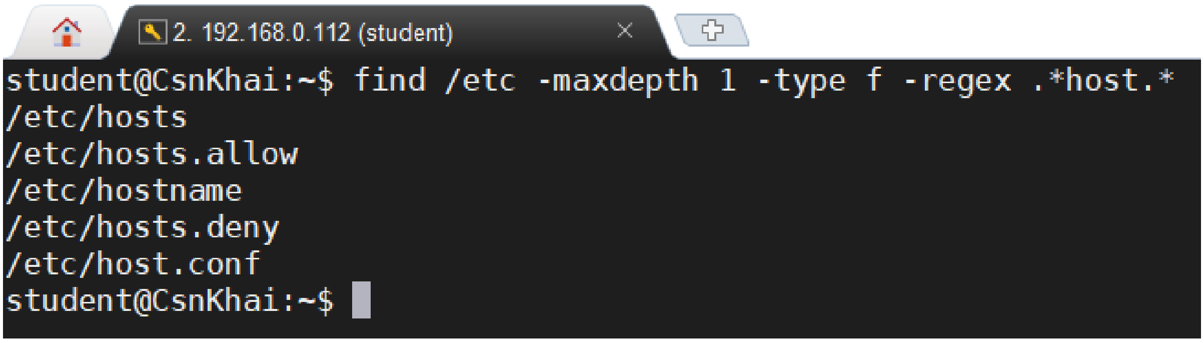
```
student@CsnKhai:~/test$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       1.5G  989M  397M  72% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
udev            999M  4.0K  999M   1% /dev
tmpfs           202M  396K  202M   1% /run
none            5.0M     0  5.0M   0% /run/lock
none           1008M     0 1008M   0% /run/shm
none            100M     0  100M   0% /run/user
student@CsnKhai:~/test$ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0   1.5G  0 disk
└─sda1   8:1    0   1.5G  0 part /
sr0     11:0    1  1024M  0 rom
student@CsnKhai:~/test$
```

9) **Count the number of lines containing a given sequence of characters in a given file.**



```
student@CsnKhai:~/test$ grep -c "clear" hard_lnk_labwork2
23
student@CsnKhai:~/test$
```

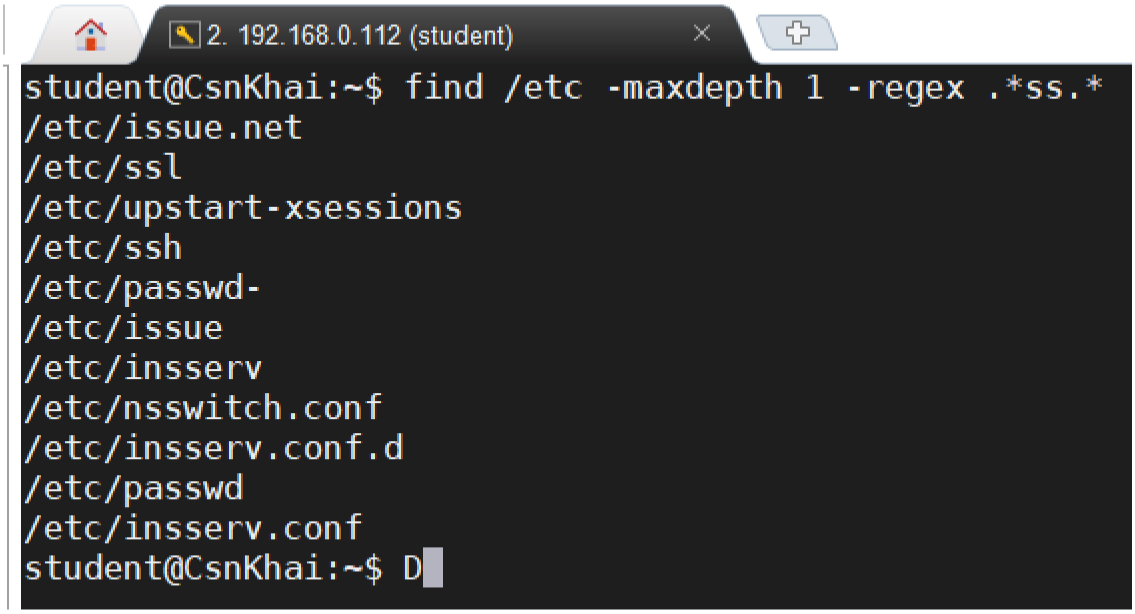
10) Using the find command, find all files in the /etc directory containing the host character sequence.



```
student@CsnKhai:~$ find /etc -maxdepth 1 -type f -regex .*host.*
/etc/hosts
/etc/hosts.allow
/etc/hostname
/etc/hosts.deny
/etc/host.conf
student@CsnKhai:~$
```

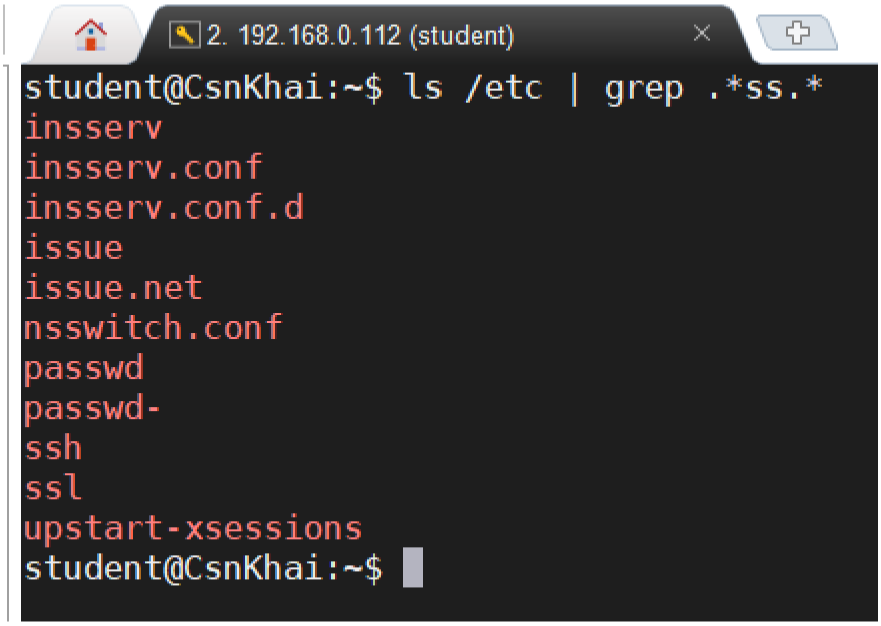
11) List all objects in /etc that contain the ss character sequence. How can I duplicate a similar command using a bunch of grep?

With ***find***:



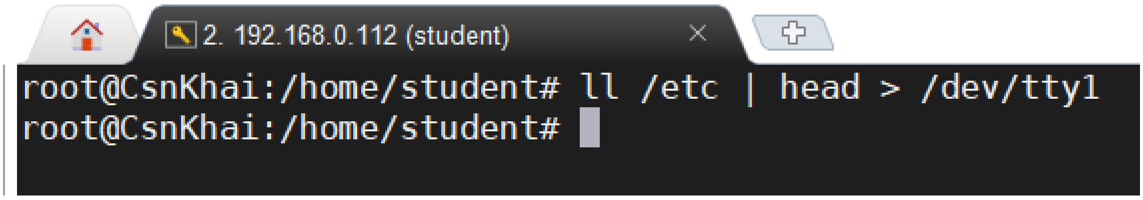
```
student@CsnKhai:~$ find /etc -maxdepth 1 -regex .*ss.*
/etc/issue.net
/etc/ssl
/etc/upstart-xsessions
/etc/ssh
/etc/passwd-
/etc/issue
/etc/insserv
/etc/nsswitch.conf
/etc/insserv.conf.d
/etc/passwd
/etc/insserv.conf
student@CsnKhai:~$ D
```

With ***grep***:

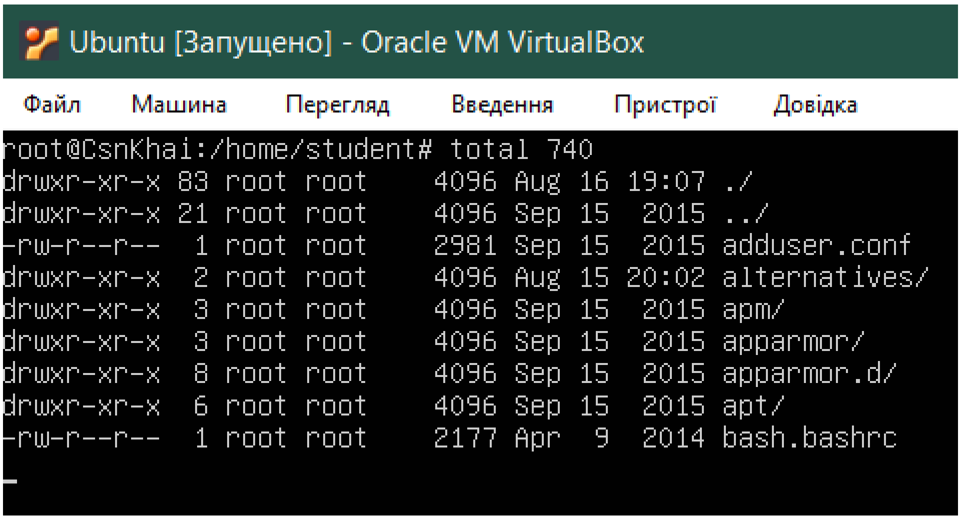


```
student@CsnKhai:~$ ls /etc | grep .*ss.*
insserv
insserv.conf
insserv.conf.d
issue
issue.net
nsswitch.conf
passwd
passwd-
ssh
ssl
upstart-xsessions
student@CsnKhai:~$
```

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



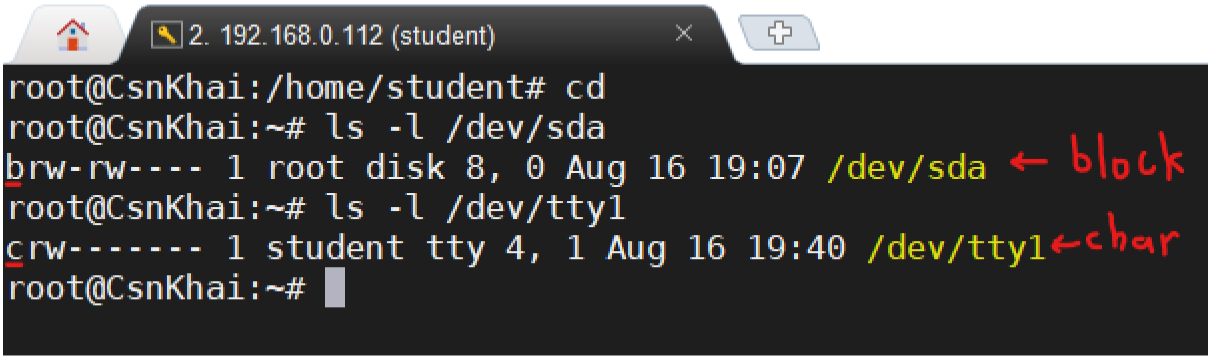
```
root@CsnKhai:/home/student# ll /etc | head > /dev/tty1
root@CsnKhai:/home/student#
```



```
root@CsnKhai:/home/student# total 740
drwxr-xr-x 83 root root    4096 Aug 16 19:07 ./
drwxr-xr-x 21 root root    4096 Sep 15  2015 ../
-rw-r--r--  1 root root    2981 Sep 15  2015 adduser.conf
drwxr-xr-x  2 root root    4096 Aug 15 20:02 alternatives/
drwxr-xr-x  3 root root    4096 Sep 15  2015 apm/
drwxr-xr-x  3 root root    4096 Sep 15  2015 apparmor/
drwxr-xr-x  8 root root    4096 Sep 15  2015 apparmor.d/
drwxr-xr-x  6 root root    4096 Sep 15  2015 apt/
-rw-r--r--  1 root root    2177 Apr  9  2014 bash.bashrc
_
```

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

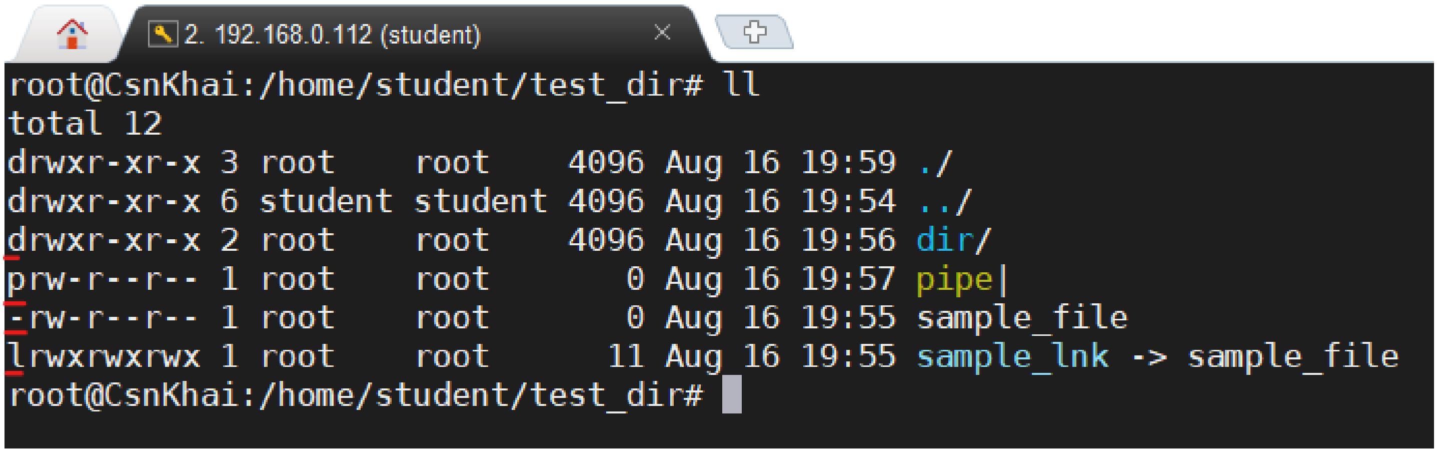
There are block devices (**b**) and character devices (**c**). To determine the type of device we can look on ***ls*** command output:



```
root@CsnKhai:/home/student# cd
root@CsnKhai:~# ls -l /dev/sda
brw-rw---- 1 root disk 8, 0 Aug 16 19:07 /dev/sda
root@CsnKhai:~# ls -l /dev/tty1
crw------- 1 student tty 4, 1 Aug 16 19:40 /dev/tty1
root@CsnKhai:~#
```

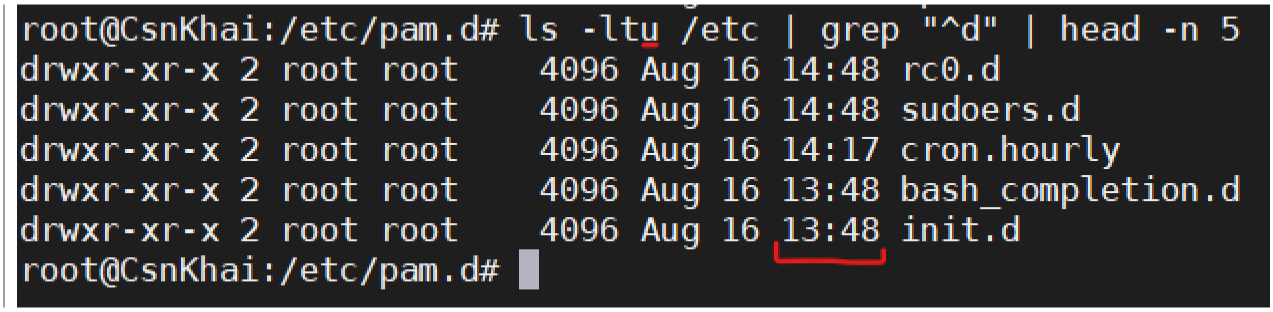
14) **How to determine the type of file in the system, what types of files are there?**

There are directories (**d**), regular files (**-**), symbolic links (**l**), sockets (**s**) and pipes (**p**).



```
root@CsnKhai:/home/student/test_dir# ll
total 12
drwxr-xr-x 3 root    root    4096 Aug 16 19:59 ./
drwxr-xr-x 6 student student 4096 Aug 16 19:54 ../
drwxr-xr-x 2 root    root    4096 Aug 16 19:56 dir/
prw-r--r-- 1 root    root       0 Aug 16 19:57 pipe|
-rw-r--r-- 1 root    root       0 Aug 16 19:55 sample_file
lrwxrwxrwx 1 root    root      11 Aug 16 19:55 sample_lnk -> sample_file
root@CsnKhai:/home/student/test_dir#
```

15) *** List the first 5 directory files that were recently accessed in the /etc directory.**



```
root@CsnKhai:/etc/pam.d# ls -ltu /etc | grep "^d" | head -n 5
drwxr-xr-x 2 root root    4096 Aug 16 14:48 rc0.d
drwxr-xr-x 2 root root    4096 Aug 16 14:48 sudoers.d
drwxr-xr-x 2 root root    4096 Aug 16 14:17 cron.hourly
drwxr-xr-x 2 root root    4096 Aug 16 13:48 bash_completion.d
drwxr-xr-x 2 root root    4096 Aug 16 13:48 init.d
root@CsnKhai:/etc/pam.d#
```