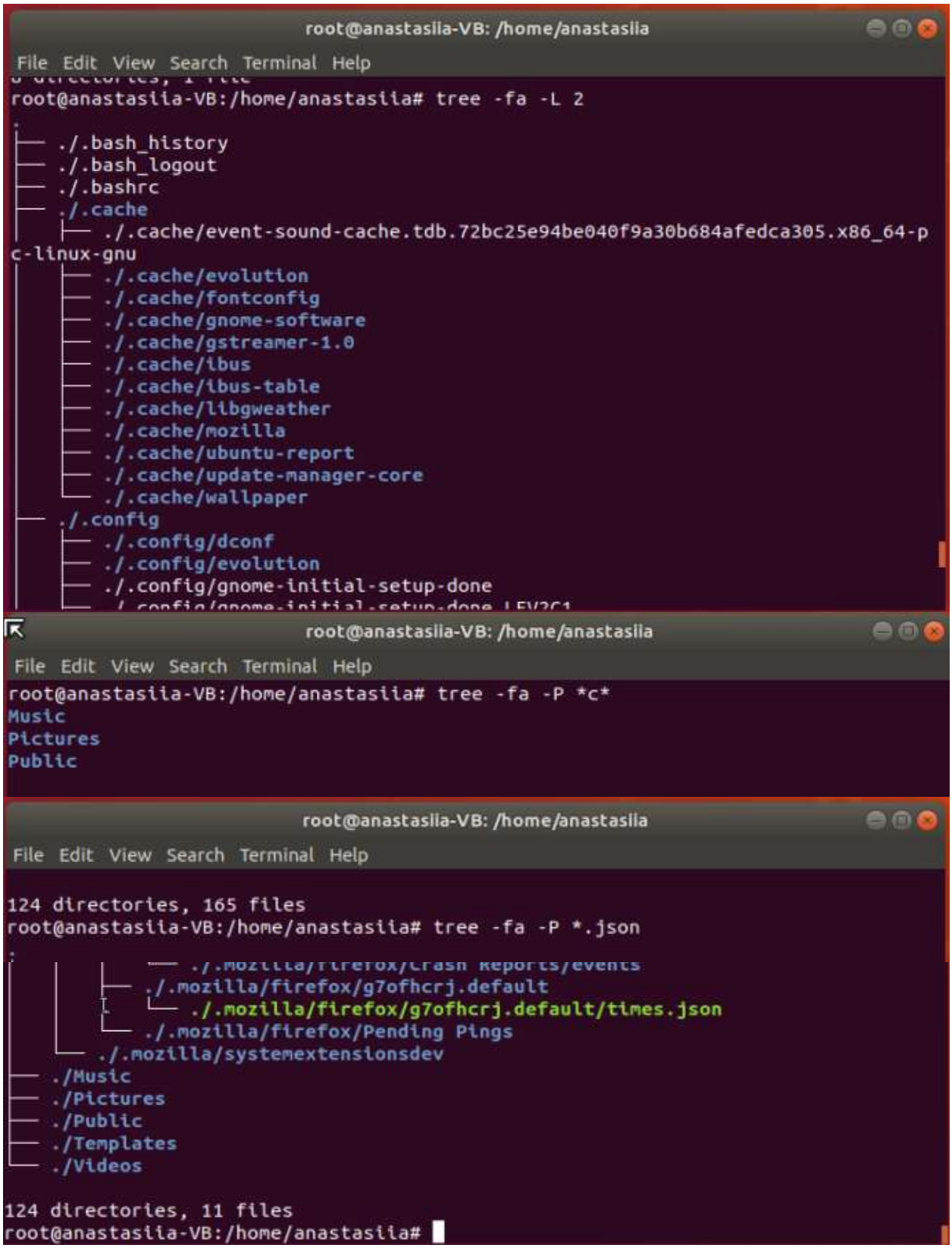
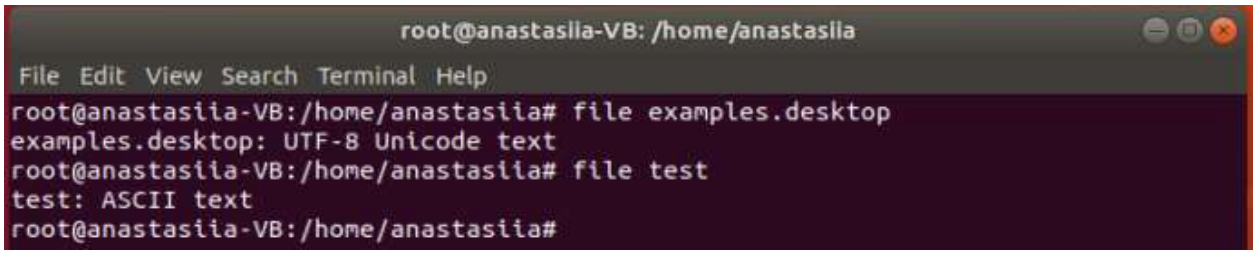
1. Examine the tree command. Master the technique of applying a template, for example, display all files that contain a character c, or files that contain a specific sequence of characters. List subdirectories of the root directory up to and including the second nesting level.

tree command lists contents of directories in a tree-like format. To list only thosefiles that match the wild-card pattern, use the following command: tree -P pattern And to max display depth of the directory tree, use: tree -L level



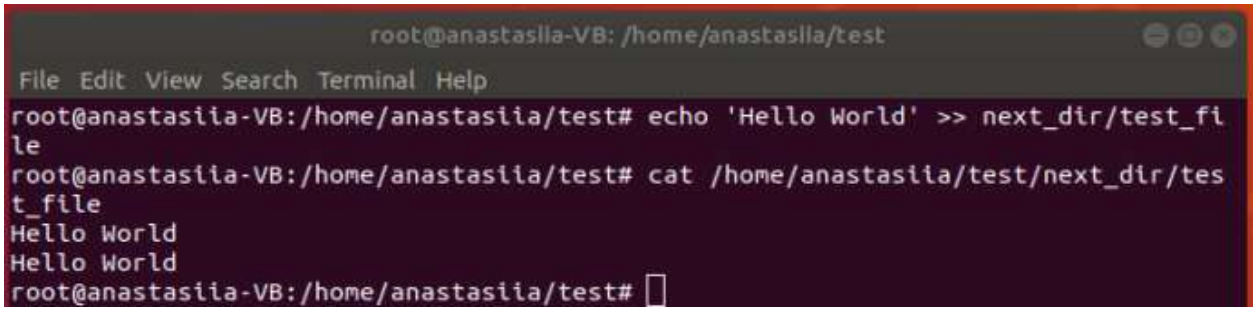
2. What command can be used to determine the type of file (for example, text or binary)? Give an example.

To determine the type if file, use file command. file tests each arfument in an attempt to classify it:



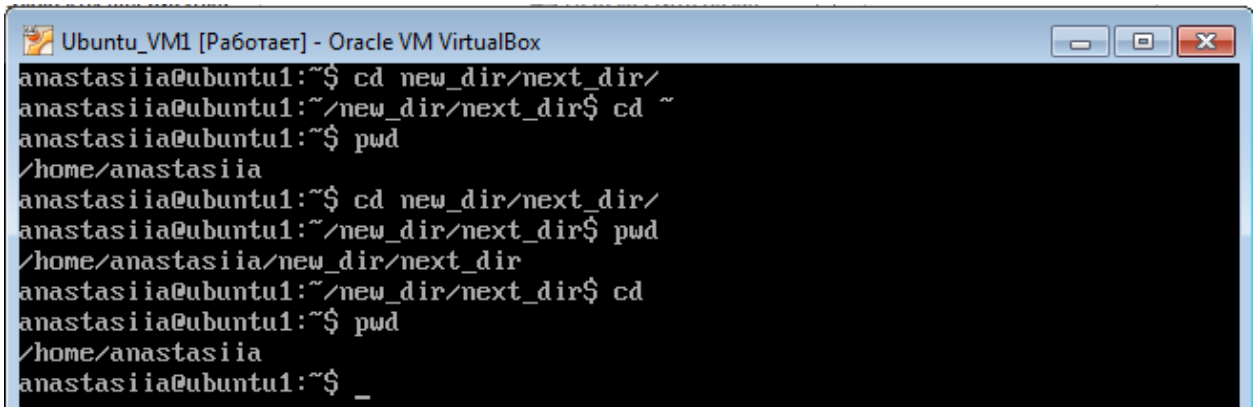
3. Master the skills of navigating the file system using relative and absolute paths. How can you go back to your home directory from anywhere in the filesystem?

An absolute path is defined as the specifying the location of a file or directory from the root directory(/). In other words we can say absolute path is a complete path from start of actual filesystem from / directory. Relative path is defined as path related to the present working directory(pwd).

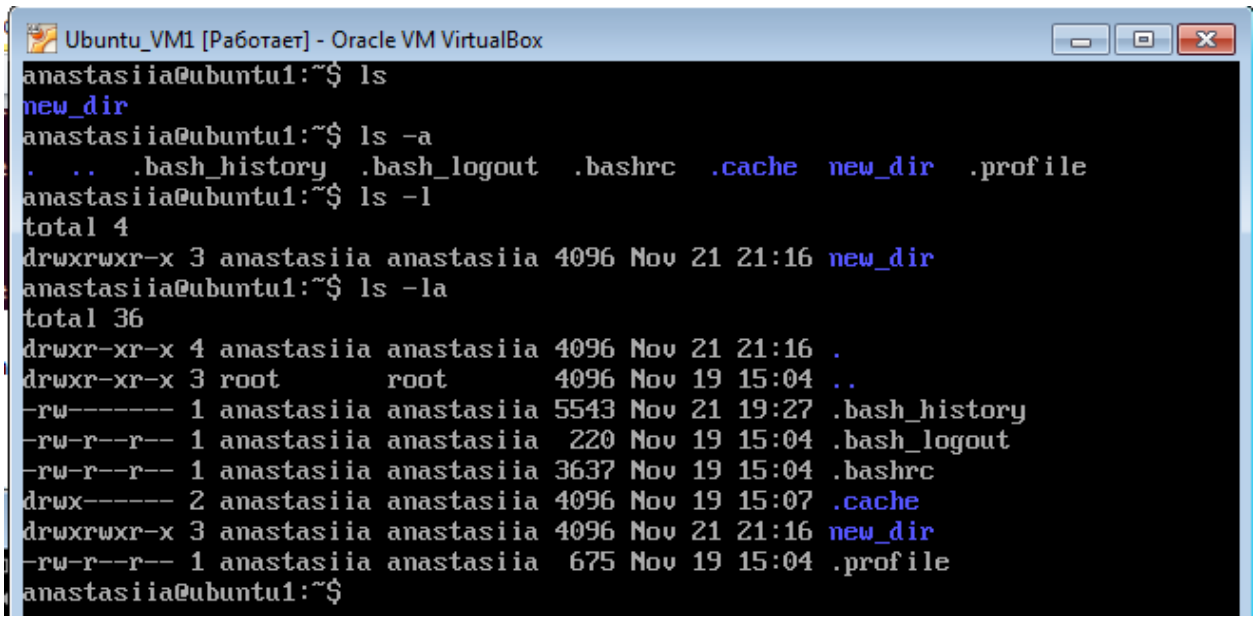


To go back to your home directory from anywhere in the filesystem, use: cd ~

or cd



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.

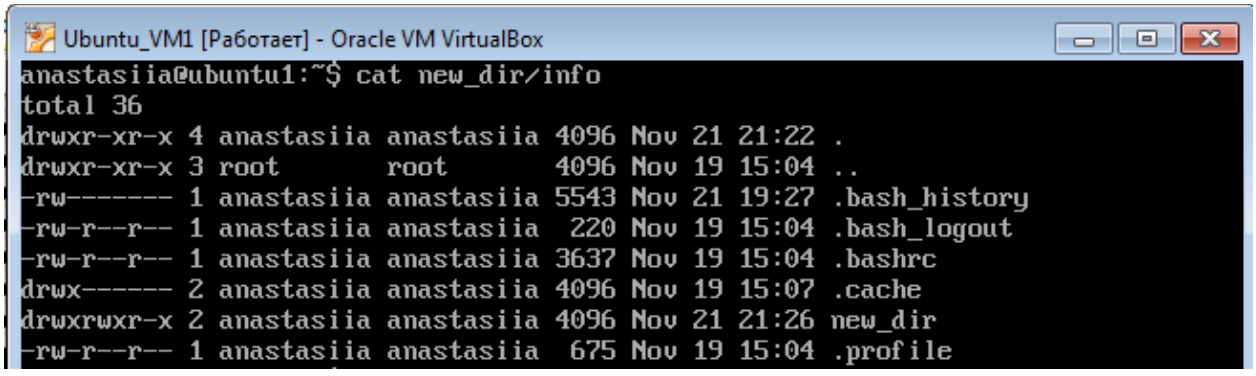


5. Perform the following sequence of operations:

- 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);

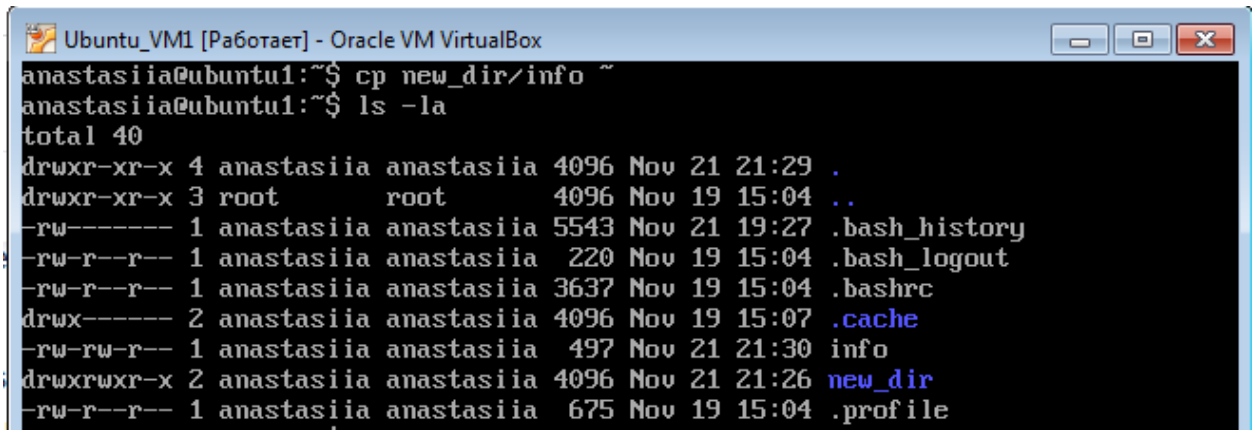
- view the created file;



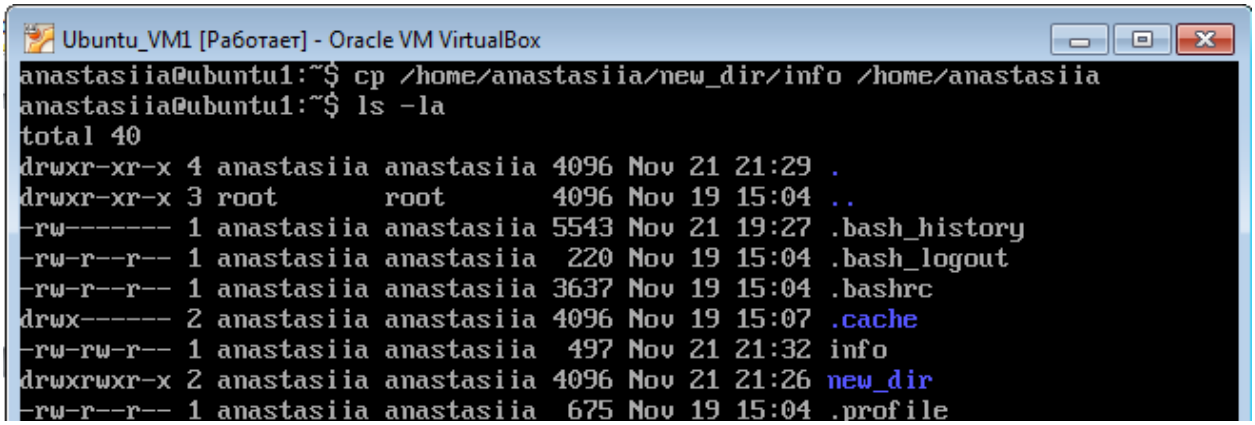
- copy the created file to your home directory using relative and absolute

addressing.

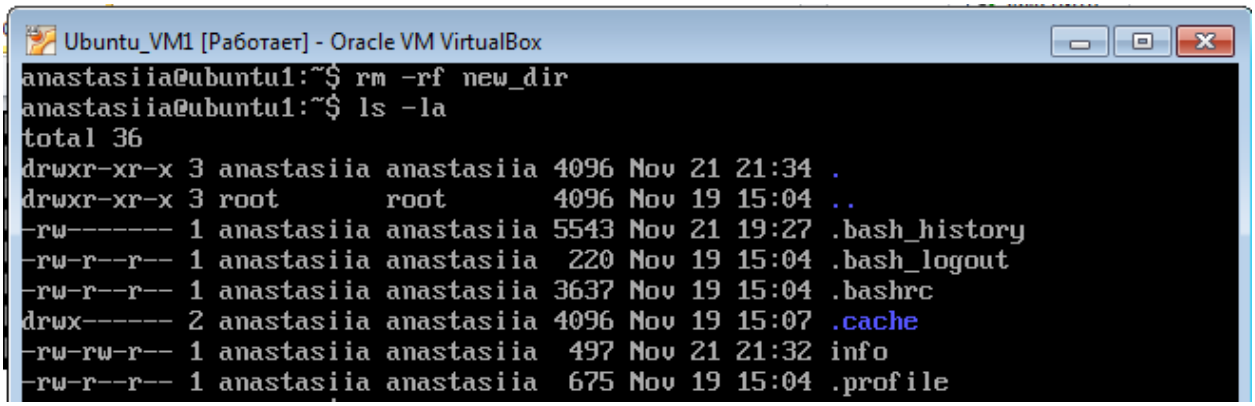
Relative addressing:



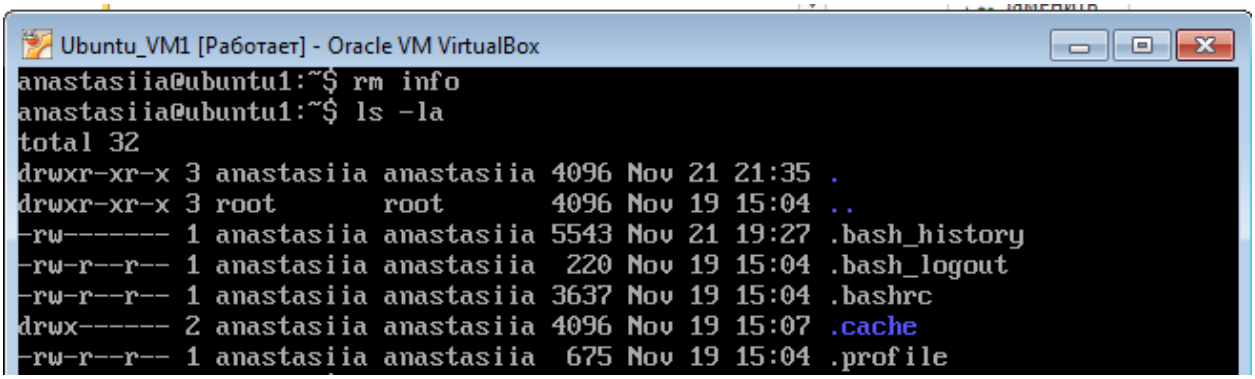
Absolute addressing:



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

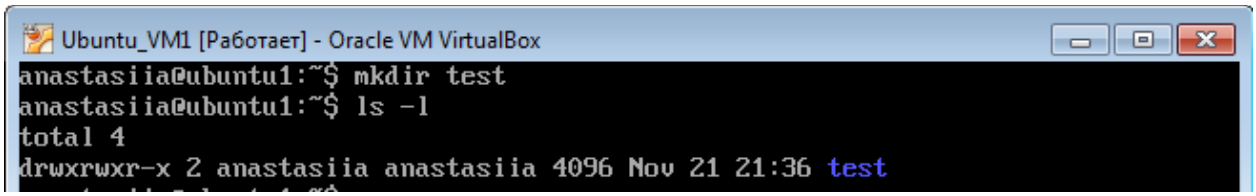


- delete the file copied to the home directory.



6. Perform the following sequence of operations:

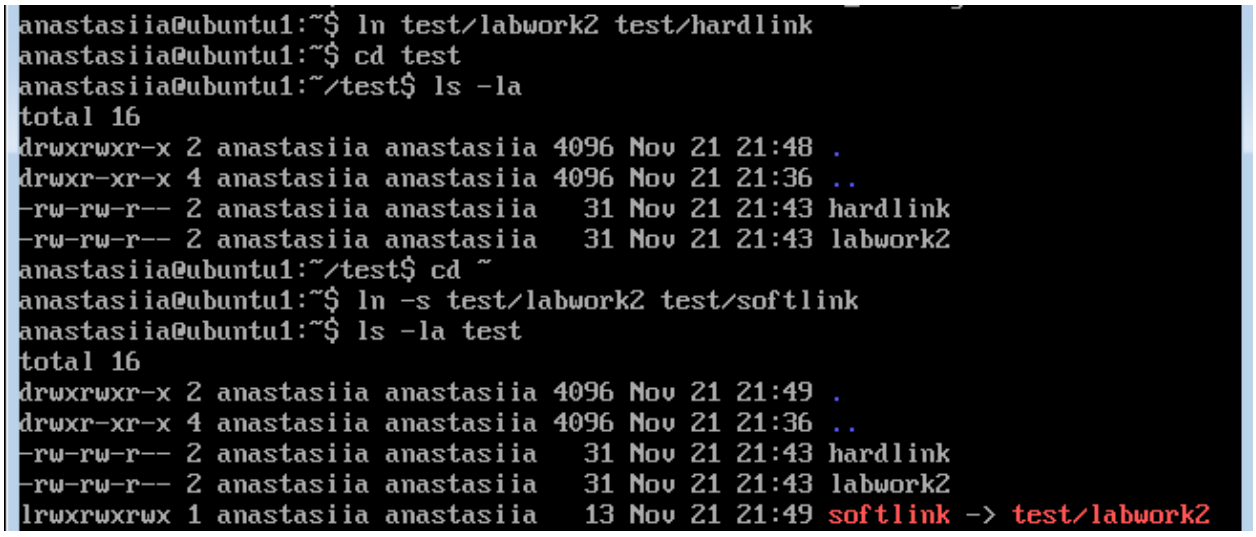
- create a subdirectory test in the home directory;



- 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;



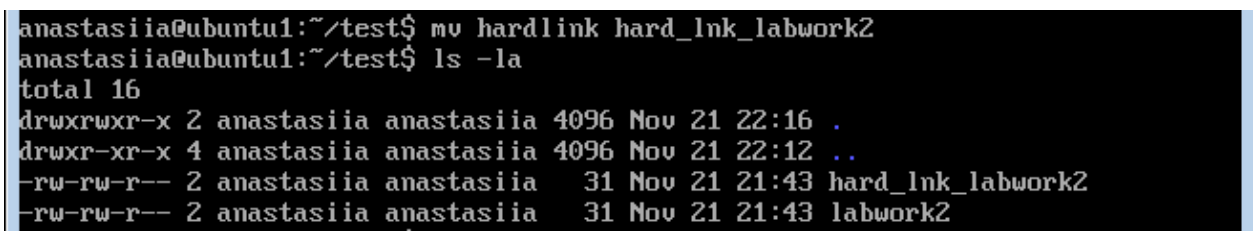
- how to define soft and hard link, what do these

concepts;

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

why

- rename the hard link file to hard_lnk_labwork2;



- rename the soft link file to symb_lnk_labwork2 file;

```
anastasiia@ubuntu1:~/test$ mv softlink soft_lnk_labwork2
anastasiia@ubuntu1:~/test$ ls -la
total 16
drwxrwxr-x 2 anastasiia anastasiia 4096 Nov 21 22:19 .
drwxr-xr-x 4 anastasiia anastasiia 4096 Nov 21 22:18 ..
-rw-rw-r-- 2 anastasiia anastasiia   31 Nov 21 21:43 hard_lnk_labwork2
-rw-rw-r-- 2 anastasiia anastasiia   31 Nov 21 21:43 labwork2
lrwxrwxrwx 1 anastasiia anastasiia    8 Nov 21 22:18 soft_lnk_labwork2 -> labwor
k2
```

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

```
anastasiia@ubuntu1:~/test$ rm labwork2
anastasiia@ubuntu1:~/test$ ls -la
total 12
drwxrwxr-x 2 anastasiia anastasiia 4096 Nov 21 22:20 .
drwxr-xr-x 4 anastasiia anastasiia 4096 Nov 21 22:18 ..
-rw-rw-r-- 1 anastasiia anastasiia   31 Nov 21 21:43 hard_lnk_labwork2
lrwxrwxrwx 1 anastasiia anastasiia    8 Nov 21 22:18 soft_lnk_labwork2 -> labwor
k2
```

When deleting the source file, an error occurs that such a file does not exist, because the source file was actually deleted. Removing the link leaves the original file in place.

When deleting the source file, nothing happened and the link still points to the desired disk section, this is the main difference between a hard link and a symbolic one. We can conclude that the linux hardlink is a regular file. Each file has at least one link, but for some we can create multiple links.

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

```
anastasiia@ubuntu1:~$ locate -w squid
/usr/share/vim/vim74/syntax/squid.vim
anastasiia@ubuntu1:~$ locate -w traceroute
/etc/alternatives/traceroute6
/etc/alternatives/traceroute6.8.gz
/lib/modules/4.4.0-142-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
```

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

these partitions.

```
anastasiia@ubuntu1:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.4G  4.0K  2.4G   1% /dev
tmpfs           484M  372K  484M   1% /run
/dev/sda1       122G  1.8G  114G   2% /
none            4.0K     0  4.0K   0% /sys/fs/cgroup
none            5.0M     0  5.0M   0% /run/lock
none            2.4G     0  2.4G   0% /run/shm
none            100M     0  100M   0% /run/user
anastasiia@ubuntu1:~$ lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0   128G  0 disk
├─sda1   8:1    0 123.1G  0 part /
├─sda2   8:2    0     1K  0 part
└─sda5   8:5    0   4.9G  0 part [SWAP]
sr0     11:0    1  1024M  0 rom
```

```
anastasiia@ubuntu1:~$ mount
/dev/sda1 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/cgroup type tmpfs (rw)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
udev on /dev type devtmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
none on /run/user type tmpfs (rw,noexec,nosuid,nodev,size=104857600,mode=0755)
none on /sys/fs/pstore type pstore (rw)
systemd on /sys/fs/cgroup/systemd type cgroup (rw,noexec,nosuid,nodev,none,name=
systemd)
```

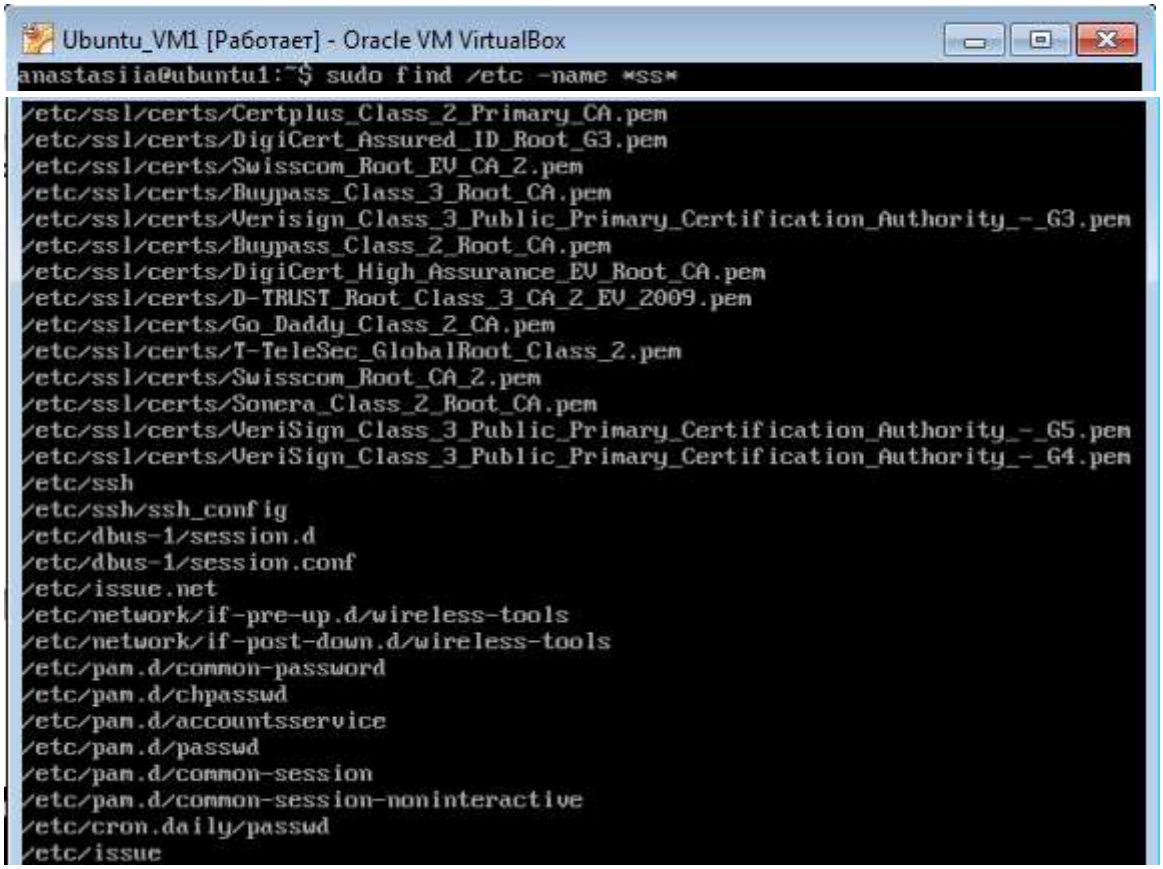
9. Count the number of lines containing a given sequence of characters in a given file.

```
anastasiia@ubuntu1:~$ touch info.txt
anastasiia@ubuntu1:~$ ls -la > info.txt
anastasiia@ubuntu1:~$ cat info.txt
total 36
drwxr-xr-x 4 anastasiia anastasiia 4096 Nov 21 22:45 .
drwxr-xr-x 3 root       root       4096 Nov 19 15:04 ..
-rw------- 1 anastasiia anastasiia 5543 Nov 21 19:27 .bash_history
-rw-r--r-- 1 anastasiia anastasiia  220 Nov 19 15:04 .bash_logout
-rw-r--r-- 1 anastasiia anastasiia 3637 Nov 19 15:04 .bashrc
drwx------ 2 anastasiia anastasiia 4096 Nov 19 15:07 .cache
-rw-rw-r-- 1 anastasiia anastasiia    0 Nov 21 22:45 info.txt
-rw-r--r-- 1 anastasiia anastasiia  675 Nov 19 15:04 .profile
drwxrwxr-x 2 anastasiia anastasiia 4096 Nov 21 22:20 test
anastasiia@ubuntu1:~$ grep good info.txt | wc -l
0
anastasiia@ubuntu1:~$ grep bash info.txt | wc -l
3
```

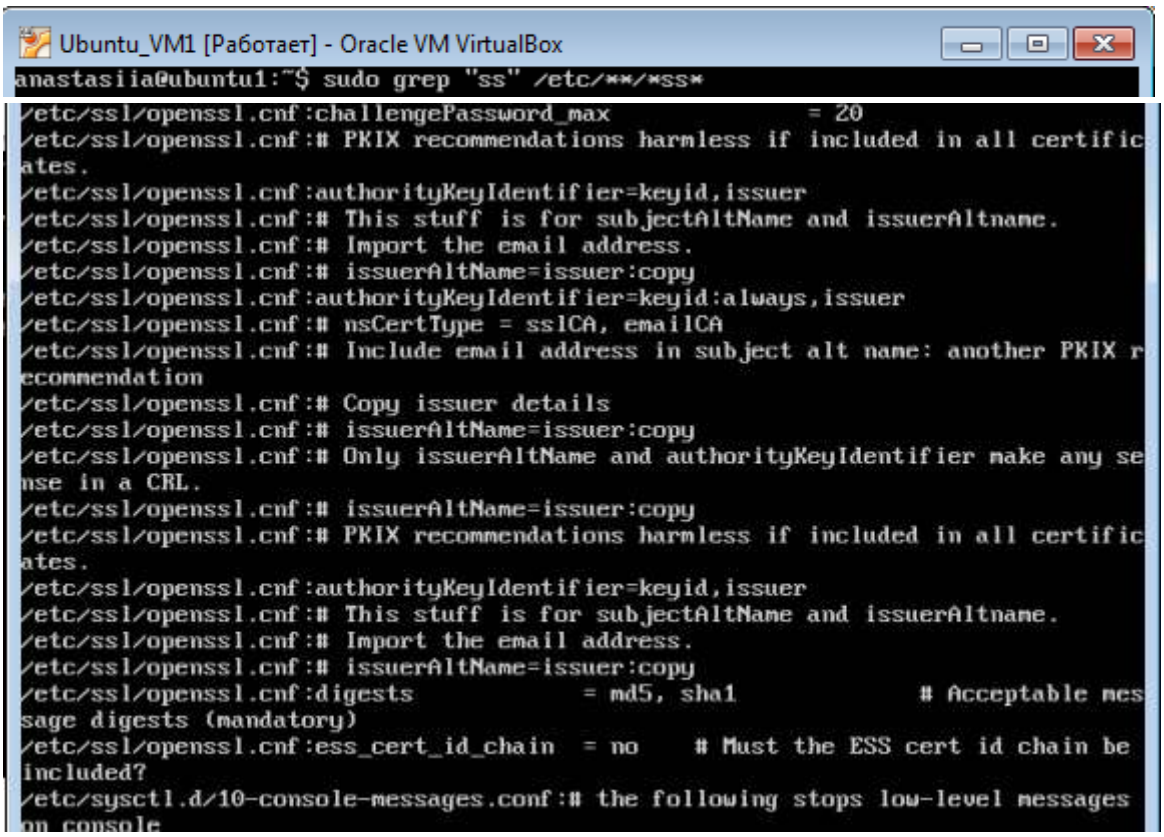
10. Using the find command, find all files in the /etc directory containing the host character sequence.

```
anastasiia@ubuntu1:~$ sudo find /etc -name *host*
/etc/hosts
/etc/host.conf
/etc/init/hostname.conf
/etc/dbus-1/system.d/org.freedesktop.hostname1.conf
/etc/hostname
```

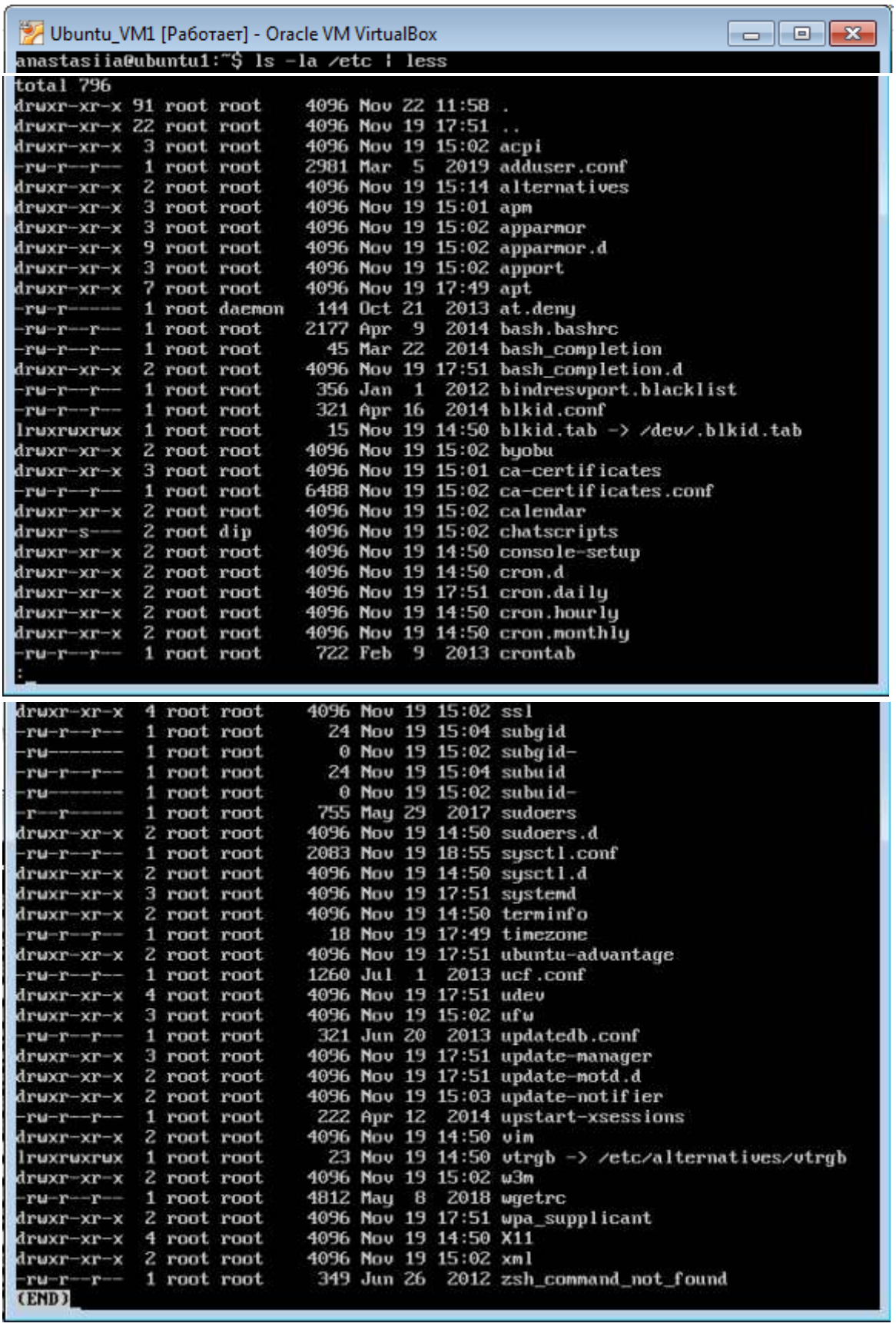
11. List all objects in /etc that contain the ss character sequence. How can I duplicate a similar command using a bunch of grep?



With grep command

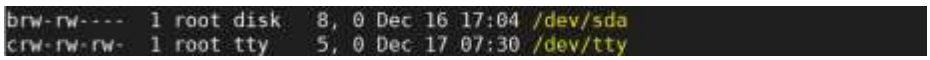


12. Organize a screen-by-screen print of the contents of the /etc directory. Hint: You must use stream redirection operations.



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

```
prw-rw-r-- 1 student student 0 Dec 17 08:51 myPipe
srwxrwxr-x 1 student student 0 Dec 17 08:58 mySocket1.sock
srwxrwxr-x 1 student student 0 Dec 17 08:57 mySocket.sock
```



The columns are as follows from left to right:

- Permissions
- Owner
- Group
- Major Device Number
- Minor Device Number
- Timestamp
- Device Name

Remember in the ls command you can see the type of file with the first bit on each line. Device files are denoted as the following:

c - character

b - block

p - pipe

s - socket

Character Device

These devices transfer data, but one a character at a time. You'll see a lot of pseudo devices (/dev/null) as character devices, these devices aren't really physically connected to the machine, but they allow the operating system greater functionality.

Block Device

These devices transfer data, but in large fixed-sized blocks. You'll most commonly see devices that utilize data blocks as block devices, such as harddrives, filesystems, etc.

Pipe Device

Named pipes allow two or more processes to communicate with each other, these are similar to character devices, but instead of having output sent to a device, it's sent to another process.

Socket Device

Socket devices facilitate communication between processes, similar to pipe devices but they can communicate with many processes at once.

Device Characterization

Devices are characterized using two numbers, **major device number** and **minor device number**. You can see these numbers in the above ls example, they are separated by a comma. For example, let's say a device had the device numbers: **8, 0**:

The major device number represents the device driver that is used, in this case 8, which is often the major number for sd block devices. The minor number tells the kernel which unique device it is in this driver class, in this case 0 is used to represent the first device (a).

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

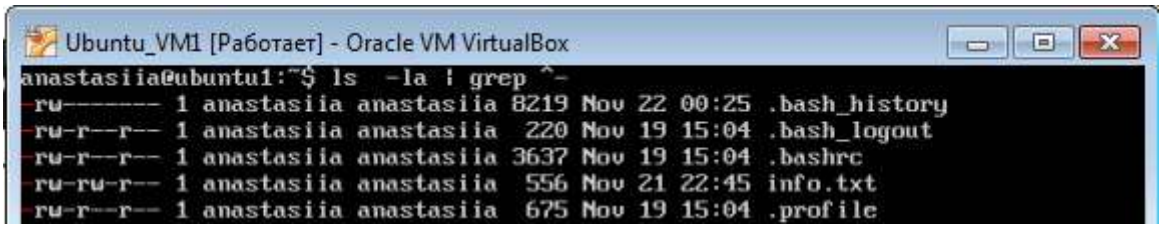
Totally 7 types of files are available in Linux with 3 Major categories. The details are below.

- Regular File
- Directory File
- Special Files (There are five types of files in the special category)
- Link File
- Character Device File
- Socket File
- Named Pipe File
- Block File

Regular file

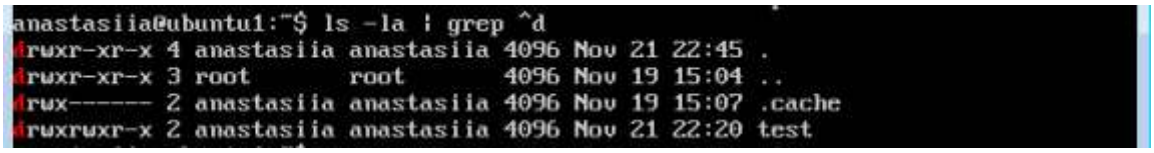
The regular file is a common file type found everywhere on Linux system. These include text files, script files, images, binary files, and shared libraries,etc.

"-" This refers to the identification symbol for the regular file. You can use the rm command to remove a regular file.



Directory

The directory is the second most common file type found on a Linux system. This can be created with the mkdir command.



Link File

A link is a mechanism for creating a shortcut to the original file or directory. It contains information about another file or directory.

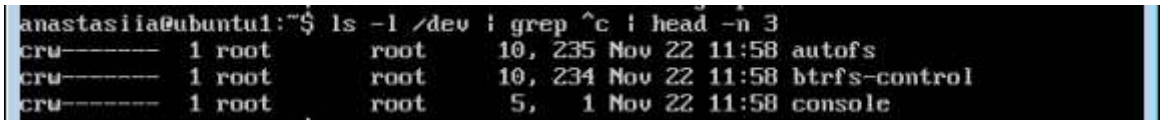
Links allow more than one filename to reference the same file.

There are two types of link files available, it's soft link and hard link.

Character Device file

Character device files allow the user and application program to communicate directly with the hardware device. It's not allow programs to read or write single characters at a time.

It is available under the /dev directory.



Block Device file

Block devices provide buffered access to hardware devices, it's similar to character devices. Unlike character devices, block devices will always allow the programmer to read or write a block of any size at a time.

```
# ls -la | grep ^b
brw-rw----.  1 root disk      7,   0 Jan 28 14:05 loop0
brw-rw----.  1 root disk      7,   1 Jan 28 14:05 loop1
brw-rw----.  1 root disk      7,   2 Jan 28 14:05 loop2
```

Socket file

A socket is a special file used for inter-process communication, which enables communication between two processes.

```
# ls -la | grep ^s
srw-rw-rw- 1 root root 0 Jan  5 16:36 system_bus_socket
```

Named Pipe file (FIFO)

a named pipe (also known as a FIFO) is one of the methods for inter-process communication.

Named pipes are special files that can exist anywhere in the file system. They can be created with the command mkfifo.

A named pipe is marked with a p as the first letter of the mode string.

```
# ls -la | grep ^p
prw-------.  1 root    root        0 Jan 28 14:06 replication-notify-fifo|
prw-------.  1 root    root        0 Jan 28 14:06 stats-mail|
```

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