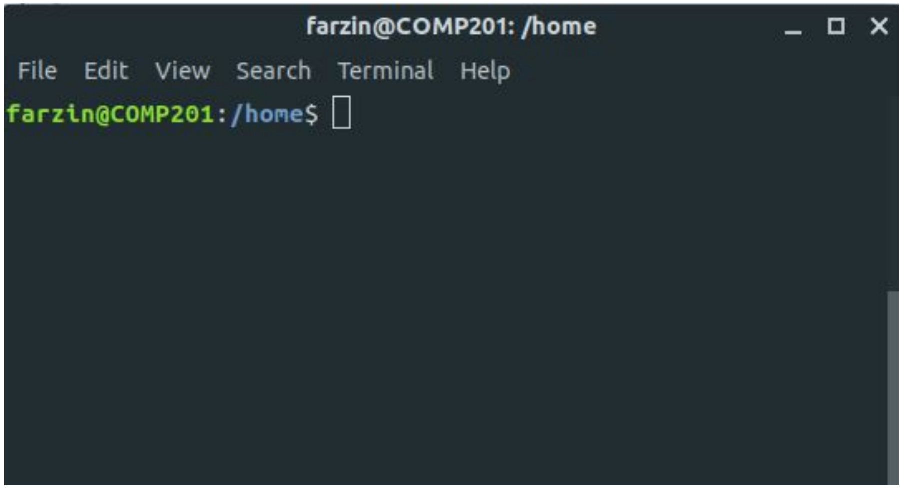
Introduction to Linux Shell

COMP201 - Lab1
Spring 2023

KOÇ UNIVERSITY

What is shell?



```
farzin@COMP201:/home$
```

- The Linux shell is the interface between you and operating system that controls the hardware.
- The most commonly used shell is called BASH – Bourne Again Shell
- username@hostname:curr_dir$
  - username: farzin
  - hostname: COMP201
  - curr_dir: /home

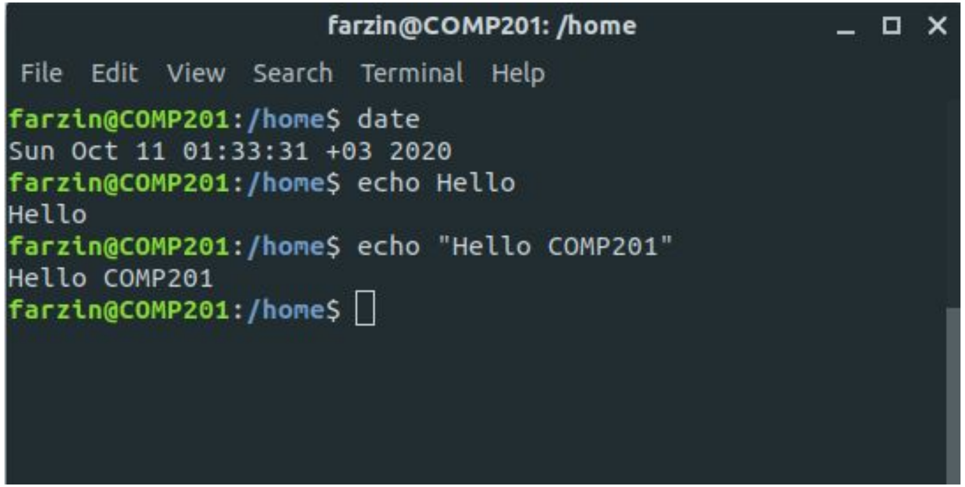
How to connect?

```
$ ssh USERNAME@linuxpool.ku.edu.tr
```

(don’t type in the “$”! This means type into terminal)

1. Type your password when prompted.
2. If you see a warning about SSH host keys, click or enter “yes.”

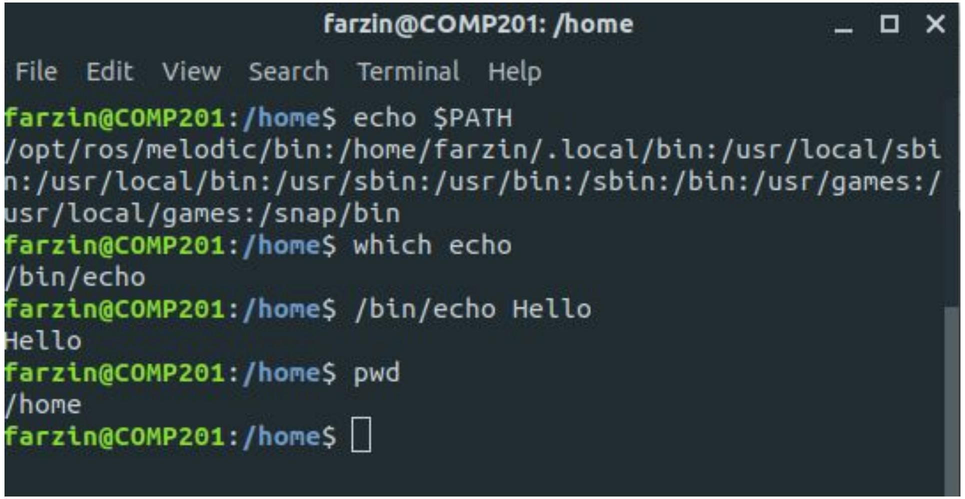
Executing system programs



```
farzin@COMP201:/home$ date
Sun Oct 11 01:33:31 +03 2020
farzin@COMP201:/home$ echo Hello
Hello
farzin@COMP201:/home$ echo "Hello COMP201"
Hello COMP201
farzin@COMP201:/home$ 
```

- Execute programs
- $date
  - This program prints current date and time
- $echo
  - This program prints the input argument

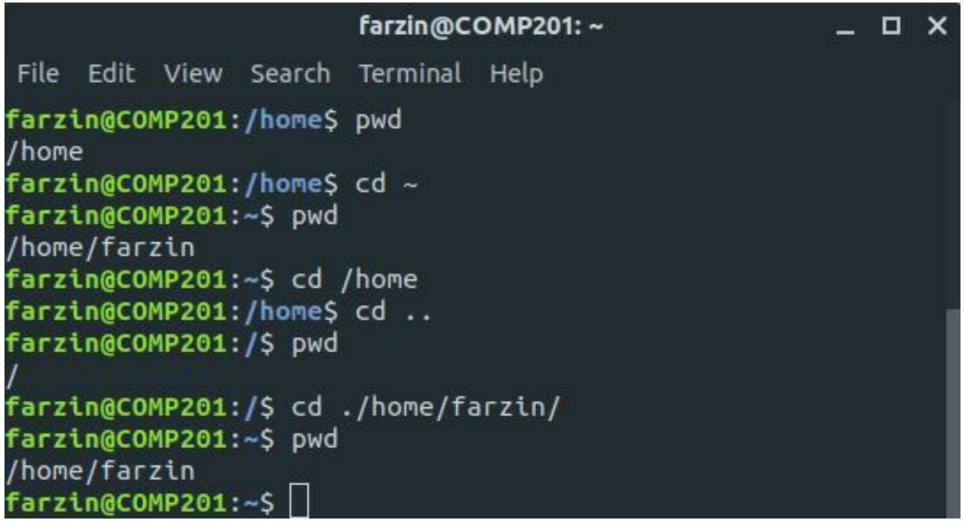
Path and $PATH



```
farzin@COMP201:/home$ echo $PATH
/opt/ros/melodic/bin:/home/farzin/.local/bin:/usr/local/sbi
n:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
usr/local/games:/snap/bin
farzin@COMP201:/home$ which echo
/bin/echo
farzin@COMP201:/home$ /bin/echo Hello
Hello
farzin@COMP201:/home$ pwd
/home
farzin@COMP201:/home$ 
```

- $PATH
  - A variable that contains addresses where system look for programs to execute
- $which
  - Prints which file is being executed given an input program name
- $pwd
  - This program prints current working directory
  - Stands for "print working directory"

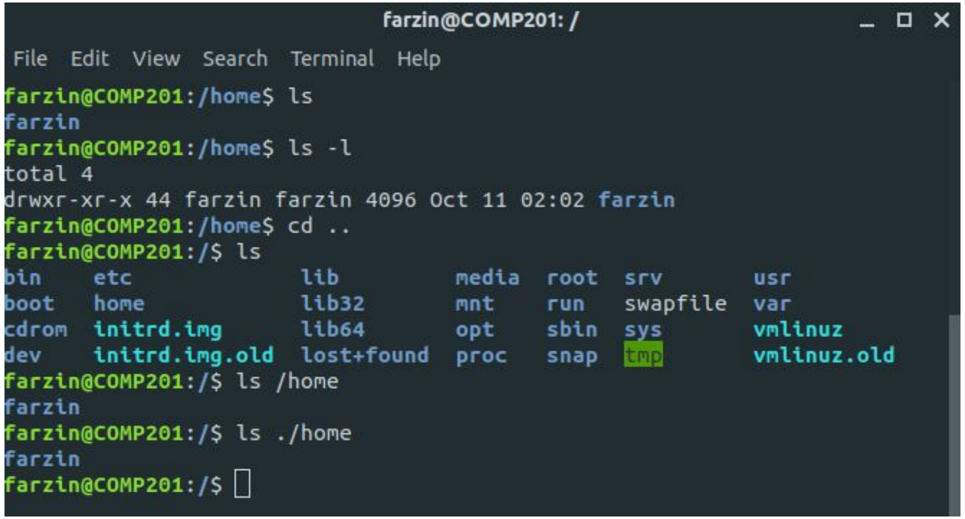
Path



```
farzin@COMP201:/home$ pwd
/home
farzin@COMP201:/home$ cd ~
farzin@COMP201:~$ pwd
/home/farzin
farzin@COMP201:~$ cd /home
farzin@COMP201:/home$ cd ..
farzin@COMP201:/$ pwd
/
farzin@COMP201:/$ cd ./home/farzin/
farzin@COMP201:~$ pwd
/home/farzin
farzin@COMP201:~$
```

- $cd
  - Changes the working directory
  - .. is the parent directory
  - . is the current directory
  - Tilda (~) is the /home/usr directory
- Absolute vs Relative path
  - Relative: ./home/farzin
  - Absolute: /home/farzin

Listing files and directories



```
farzin@COMP201:/home$ ls
farzin
farzin@COMP201:/home$ ls -l
total 4
drwxr-xr-x 44 farzin farzin 4096 Oct 11 02:02 farzin
farzin@COMP201:/home$ cd ..
farzin@COMP201:/$ ls
bin    etc              lib          media  root  srv       usr
boot   home             lib32        mnt    run   swapfile  var
cdrom  initrd.img       lib64        opt    sbin  sys       vmlinuz
dev    initrd.img.old   lost+found   proc   snap  tmp       vmlinuz.old
farzin@COMP201:/$ ls /home
farzin
farzin@COMP201:/$ ls ./home
farzin
farzin@COMP201:/$ 
```

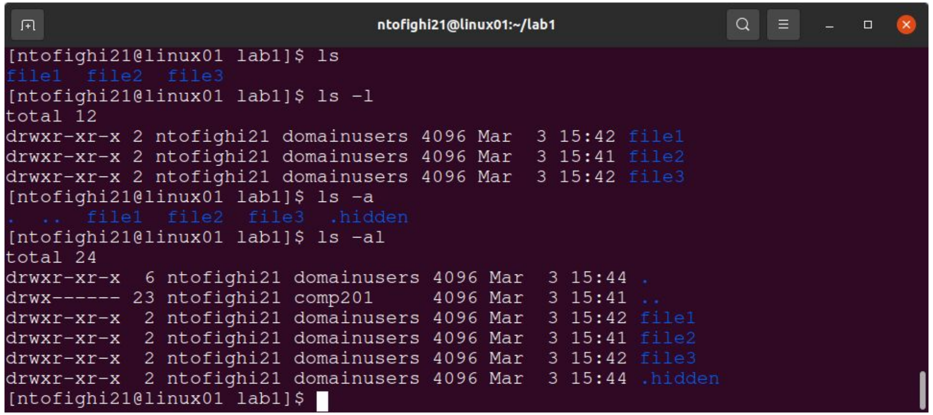
- $ ls
  - Prints files and directories under current working directory

Options with Commands in Linux

- Many Linux commands have options that can be used to modify their behavior.
- Options are usually preceded by one or two dashes, followed by a letter or a word.
- Options can be used to:
  - Control the output of a command
  - Specify a file or directory to work with
  - Modify the command's behavior in other ways

Options with Commands in Linux

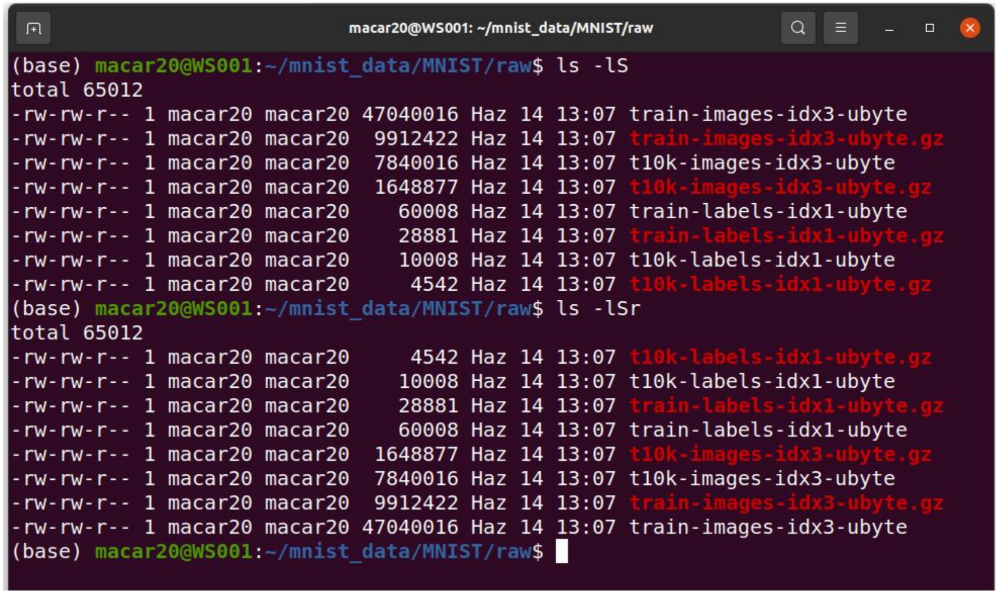
- Let's look at an example: **ls command**.
- By default, it lists the contents of the current directory.
- But we can use options to modify its behavior.
- For example,
    - -l option to display the contents of the directory in a long format, which includes additional information such as file permissions, owner, and size.
    - -a option to display all files, including hidden files (which are usually not displayed by default).
- To use both options together, we can type ls -la



```
[ntofighi21@linux01 lab1]$ ls
file1  file2  file3
[ntofighi21@linux01 lab1]$ ls -l
total 12
drwxr-xr-x 2 ntofighi21 domainusers 4096 Mar  3 15:42 file1
drwxr-xr-x 2 ntofighi21 domainusers 4096 Mar  3 15:41 file2
drwxr-xr-x 2 ntofighi21 domainusers 4096 Mar  3 15:42 file3
[ntofighi21@linux01 lab1]$ ls -a
.  ..  file1  file2  file3  .hidden
[ntofighi21@linux01 lab1]$ ls -al
total 24
drwxr-xr-x  6 ntofighi21 domainusers 4096 Mar  3 15:44 .
drwx------ 23 ntofighi21 comp201     4096 Mar  3 15:41 ..
drwxr-xr-x  2 ntofighi21 domainusers 4096 Mar  3 15:42 file1
drwxr-xr-x  2 ntofighi21 domainusers 4096 Mar  3 15:41 file2
drwxr-xr-x  2 ntofighi21 domainusers 4096 Mar  3 15:42 file3
drwxr-xr-x  2 ntofighi21 domainusers 4096 Mar  3 15:44 .hidden
[ntofighi21@linux01 lab1]$ 
```

To learn more about the options available for a particular command →**man command**
Provides detailed information on how to use the command and its options → man ls

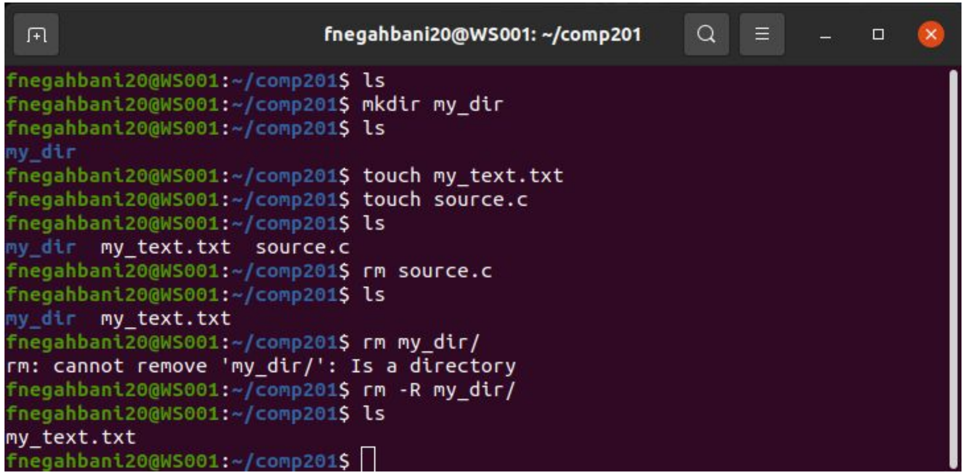
Listing files and directories


macar20@WS001: ~/mnist_data/MNIST/raw

```
(base) macar20@WS001:~/mnist_data/MNIST/raw$ ls -lS
total 65012
-rw-rw-r-- 1 macar20 macar20 47040016 Haz 14 13:07 train-images-idx3-ubyte
-rw-rw-r-- 1 macar20 macar20  9912422 Haz 14 13:07 train-images-idx3-ubyte.gz
-rw-rw-r-- 1 macar20 macar20  7840016 Haz 14 13:07 t10k-images-idx3-ubyte
-rw-rw-r-- 1 macar20 macar20  1648877 Haz 14 13:07 t10k-images-idx3-ubyte.gz
-rw-rw-r-- 1 macar20 macar20    60008 Haz 14 13:07 train-labels-idx1-ubyte
-rw-rw-r-- 1 macar20 macar20    28881 Haz 14 13:07 train-labels-idx1-ubyte.gz
-rw-rw-r-- 1 macar20 macar20    10008 Haz 14 13:07 t10k-labels-idx1-ubyte
-rw-rw-r-- 1 macar20 macar20     4542 Haz 14 13:07 t10k-labels-idx1-ubyte.gz
(base) macar20@WS001:~/mnist_data/MNIST/raw$ ls -lSr
total 65012
-rw-rw-r-- 1 macar20 macar20     4542 Haz 14 13:07 t10k-labels-idx1-ubyte.gz
-rw-rw-r-- 1 macar20 macar20    10008 Haz 14 13:07 t10k-labels-idx1-ubyte
-rw-rw-r-- 1 macar20 macar20    28881 Haz 14 13:07 train-labels-idx1-ubyte.gz
-rw-rw-r-- 1 macar20 macar20    60008 Haz 14 13:07 train-labels-idx1-ubyte
-rw-rw-r-- 1 macar20 macar20  1648877 Haz 14 13:07 t10k-images-idx3-ubyte.gz
-rw-rw-r-- 1 macar20 macar20  7840016 Haz 14 13:07 t10k-images-idx3-ubyte
-rw-rw-r-- 1 macar20 macar20  9912422 Haz 14 13:07 train-images-idx3-ubyte.gz
-rw-rw-r-- 1 macar20 macar20 47040016 Haz 14 13:07 train-images-idx3-ubyte
(base) macar20@WS001:~/mnist_data/MNIST/raw$
```

- You can use “-S” option to display files sorted by their sizes, and “-r” option for reverse sorting.

Making directories, files, and removing them



fnegahbani20@WS001: ~/comp201

```
fnegahbani20@WS001:~/comp201$ ls
fnegahbani20@WS001:~/comp201$ mkdir my_dir
fnegahbani20@WS001:~/comp201$ ls
my_dir
fnegahbani20@WS001:~/comp201$ touch my_text.txt
fnegahbani20@WS001:~/comp201$ touch source.c
fnegahbani20@WS001:~/comp201$ ls
my_dir  my_text.txt  source.c
fnegahbani20@WS001:~/comp201$ rm source.c
fnegahbani20@WS001:~/comp201$ ls
my_dir  my_text.txt
fnegahbani20@WS001:~/comp201$ rm my_dir/
rm: cannot remove 'my_dir/': Is a directory
fnegahbani20@WS001:~/comp201$ rm -R my_dir/
fnegahbani20@WS001:~/comp201$ ls
my_text.txt
fnegahbani20@WS001:~/comp201$ 
```

- $ mkdir <folder_name>
  - Makes a new directory in the given working directory with the given “folder_name”.
- $ touch
  - Creates a file with desired extension and name
- $ rm
  - Removes a file or folder.
  - For removing folders you need to use -R option

Chmod

- Chmod (short for "change mode") is a command in Linux that allows users to change the read, write, and execute permissions of files and directories.

- The syntax for chmod is as follows:
  - `chmod [options] MODE FILENAME`

- The mode is a combination of the letters "r" (read), "w" (write), and "x" (execute),
- Permissions can be granted to three different user groups:
  - The file owner
  - The group owner
  - All users

File Permission in Linux

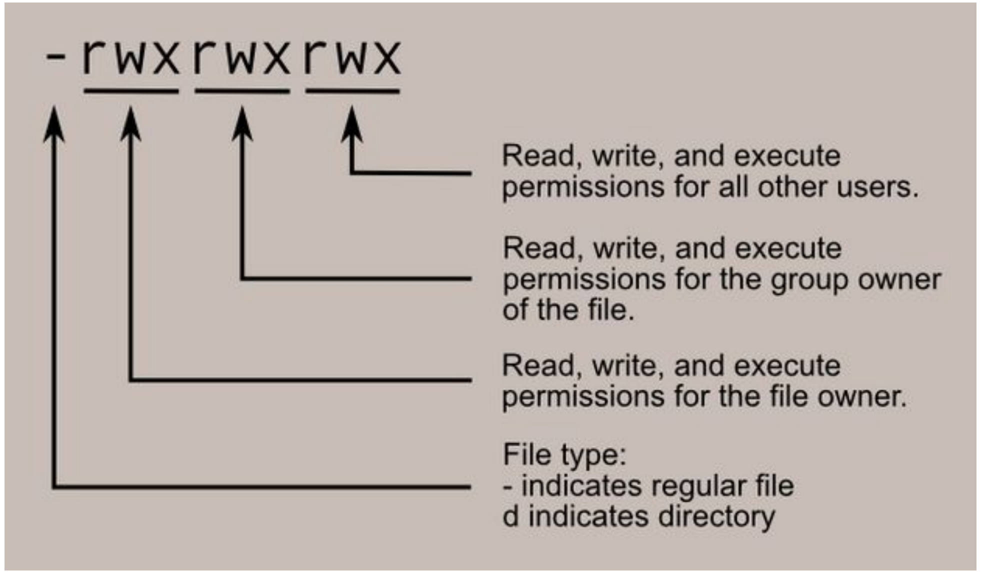
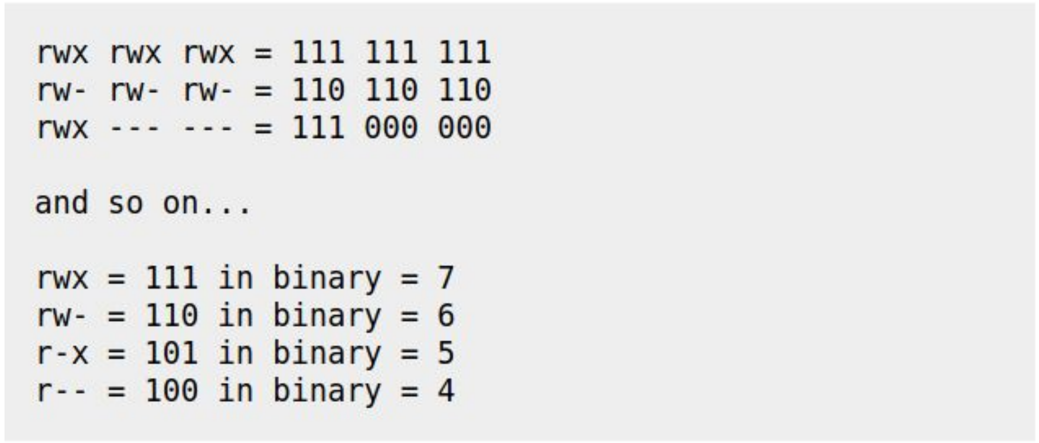

-rwxrwxrwx
Read, write, and execute permissions for all other users.
Read, write, and execute permissions for the group owner of the file.
Read, write, and execute permissions for the file owner.
File type:
- indicates regular file
d indicates directory

Image source: http://linuxcommand.org/lc3_lts0090.php

File Permission in Linux



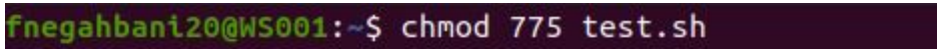
```
rwx rwx rwx = 111 111 111
rw- rw- rw- = 110 110 110
rwx --- --- = 111 000 000

and so on...

rwx = 111 in binary = 7
rw- = 110 in binary = 6
r-x = 101 in binary = 5
r-- = 100 in binary = 4
```

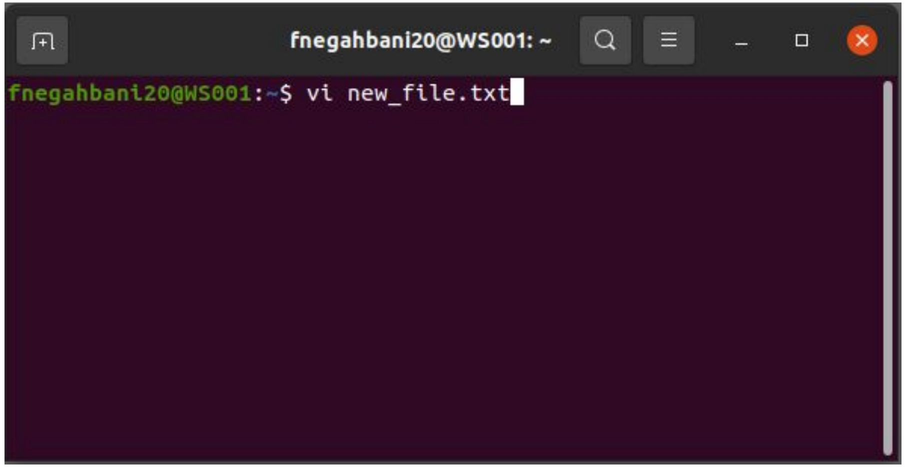
Image source: http://linuxcommand.org/lc3_lts0090.php

Initially, test.sh cannot be executed, to grant -rwx rwx r-x permission to test.sh file:



```
fnegahbani20@WS001:~$ chmod 775 test.sh
```

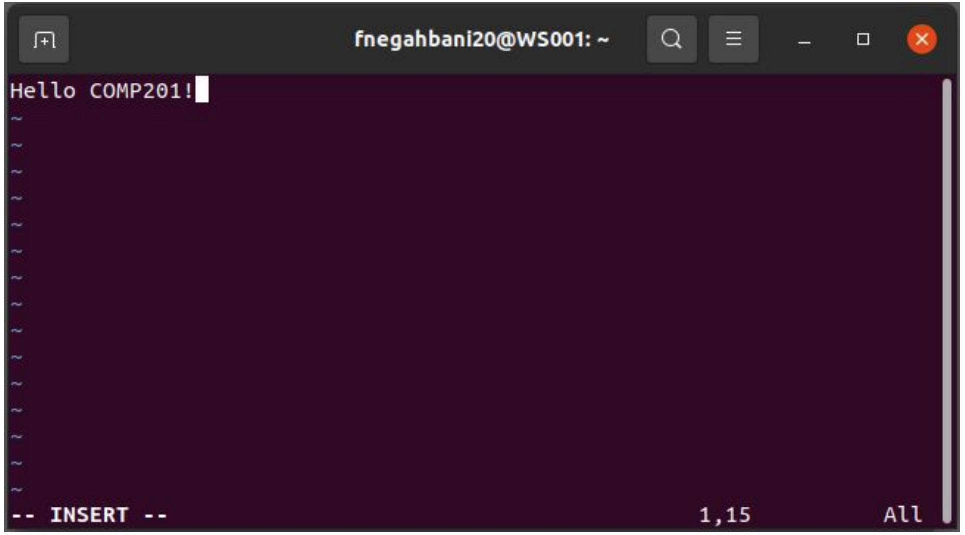
What is Vi?



```
fnegahbani20@WS001:~$ vi new_file.txt
```

- Vi is the default text editor in the UNIX operating system.
- Using vi, we can create a new file, read, and edit an existing file.
- To open vi, type “vi” or “vi filename”. If the file “filename” doesn’t exist, it will be created when you save it.

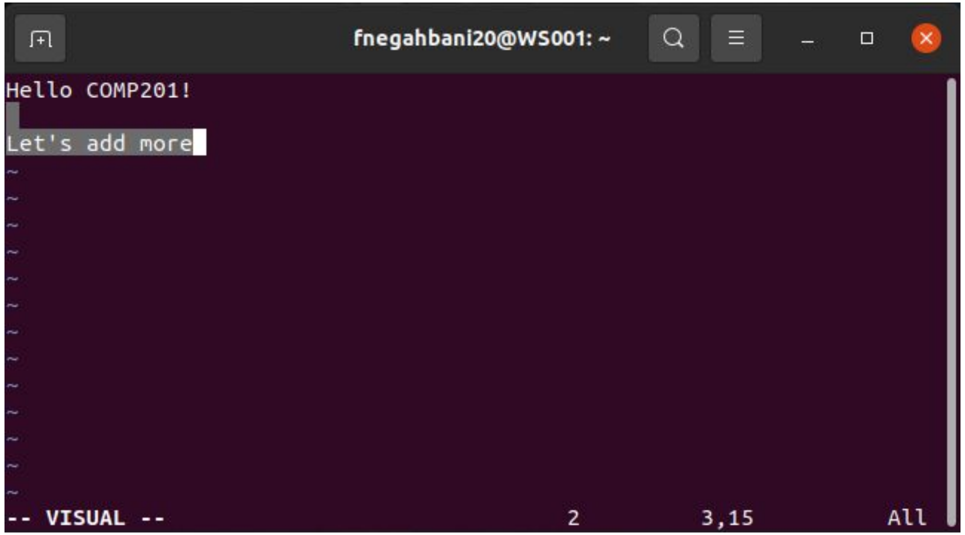
Operation Modes in vi or vim



```
Hello COMP201!
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                              1,15          All
```

- Normal mode
  - The default mode in vi.
  - In some source, like https://www.cs.colostate.edu/helpdocs/vi.html, it is also called command mode.
  - Every character you type is interpreted as a command.
- Insert mode
  - The one on the left picture.
  - To switch from normal mode to insert mode, type ‘i’ in the normal mode.
  - Every character you type is put to the file.
  - To switch back to normal mode, press <Esc>

Operation Modes in vi or vim

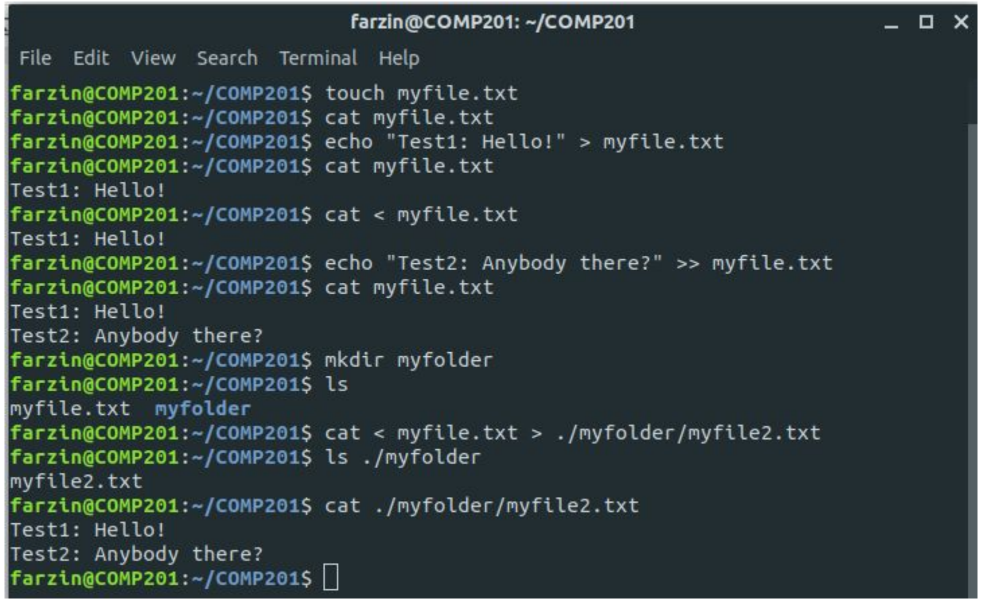


```
Hello COMP201!

Let's add more
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- VISUAL --                    2         3,15          All
```

- Visual mode
  - To switch from normal mode to visual mode, type ‘v’.
  - You can select blocks of text.
  - Type d to delete the block, c to delete the block and switch to insert mode to replace the deleted block with another string.
  - To switch back to normal mode, type <Esc>.
- Exit without saving
  - To exit from a file without saving it, go to the Normal mode (command mode) by pressing <Esc> then type :q!

Redirection

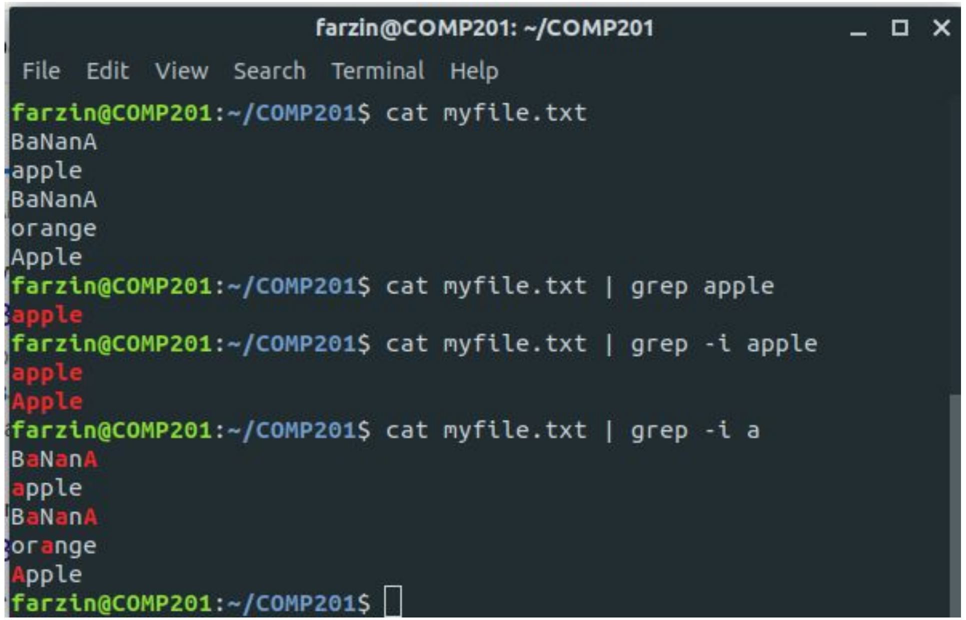


farzin@COMP201: ~/COMP201
File Edit View Search Terminal Help

```
farzin@COMP201:~/COMP201$ touch myfile.txt
farzin@COMP201:~/COMP201$ cat myfile.txt
farzin@COMP201:~/COMP201$ echo "Test1: Hello!" > myfile.txt
farzin@COMP201:~/COMP201$ cat myfile.txt
Test1: Hello!
farzin@COMP201:~/COMP201$ cat < myfile.txt
Test1: Hello!
farzin@COMP201:~/COMP201$ echo "Test2: Anybody there?" >> myfile.txt
farzin@COMP201:~/COMP201$ cat myfile.txt
Test1: Hello!
Test2: Anybody there?
farzin@COMP201:~/COMP201$ mkdir myfolder
farzin@COMP201:~/COMP201$ ls
myfile.txt  myfolder
farzin@COMP201:~/COMP201$ cat < myfile.txt > ./myfolder/myfile2.txt
farzin@COMP201:~/COMP201$ ls ./myfolder
myfile2.txt
farzin@COMP201:~/COMP201$ cat ./myfolder/myfile2.txt
Test1: Hello!
Test2: Anybody there?
farzin@COMP201:~/COMP201$ 
```

- $cat
  - Print the content of the given file
- “< file” and “> file”
  - You can wire the input and output of a program to a file
  - “>> file” appends to end of file

Piping


farzin@COMP201: ~/COMP201
File Edit View Search Terminal Help

```
farzin@COMP201:~/COMP201$ cat myfile.txt
BaNanA
apple
BaNanA
orange
Apple
farzin@COMP201:~/COMP201$ cat myfile.txt | grep apple
apple
farzin@COMP201:~/COMP201$ cat myfile.txt | grep -i apple
apple
Apple
farzin@COMP201:~/COMP201$ cat myfile.txt | grep -i a
BaNanA
apple
BaNanA
orange
Apple
farzin@COMP201:~/COMP201$
```

- Pipe character “ | “
  - Connects output of a program to input of another one
- $grep
  - Searches for a particular information
  - By default it is case sensitive
- Try “grep --help” and find what does -i option do

SCP

- SCP (Secure Copy) is a command-line tool in Linux used to securely transfer files between hosts over a network.

- The syntax for SCP is as follows:
    - `scp  [options]  SOURCE  DESTINATION`

- -r: Copy directories.

SCP

Do not forget the colon

- From local machine to LinuxPool:
  - (on local machine): `$ scp -r FILENAME USERNAME@linuxpool.ku.edu.tr:`

- From LinuxPool to local machine:
  - (on local machine): `$ scp -r USERNAME@linuxpool.ku.edu.tr:PATH/TO/FILE ./`

Useful commands:

- clear: Clearing the contents of the terminal screen
- ctrl+r: Searching for previously executed commands
- Tab: auto-completion
- * (asterisk): Used as a wildcard to represent any combination of characters in a command or filename

Other Resources

- MIT MS The Shell
- Stanford CS107 Unix videos 1-15, 24, 25
- UNIX Tutorial for Beginners