National University
of computer and emerging sciences

CL 2006 OPERATING SYSTEM

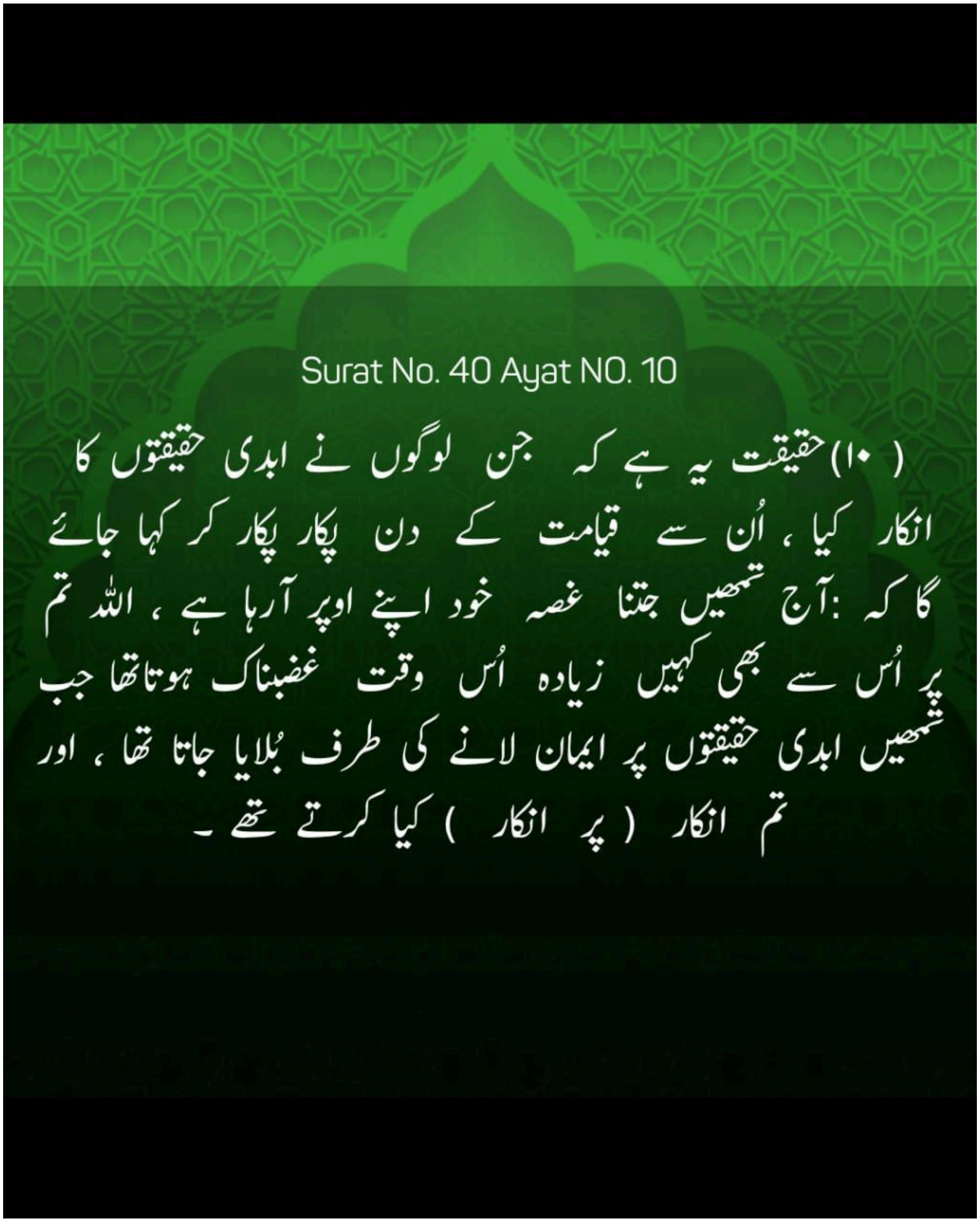
Lab 02

BS Software Engineering

Fall-2024

Course Instructors: Syed Daniyal Hussain Shah, Hifza Umar

Course Coordinator: Syed Daniyal Hussain Shah



Surat No. 40 Ayat NO. 10

(۱۰)حقیقت یہ ہے کہ جن لوگوں نے ابدی حقیقتوں کا
انکار کیا ، اُن سے قیامت کے دن پکار پکار کر کہا جائے
گا کہ :آج تمھیں جتنا غصہ خود اپنے اوپر آرہا ہے ، اللہ تم
پر اُس سے بھی کہیں زیادہ اُس وقت غضبناک ہوتا تھا جب
تمھیں ابدی حقیقتوں پر ایمان لانے کی طرف بُلایا جاتا تھا ، اور
تم انکار (پر انکار) کیا کرتے تھے ۔

Lab Policy and Rules:

1. **100% attendance is mandatory.** In case of an emergency, you can avail yourself of up to 3 absences. So, don't waste your absences , save them for emergencies. If you are debarred from the course due to low attendance, do not come to me to correct it.

2**. Disturbing the class environment during lab sessions.** such as by talking, using mobile phones, eating, etc. will result in penalties (e.g., deduction of marks).

3. **Lab tasks will not be accepted if you are absent.** If you miss a lab, you cannot submit that lab task.

4**. Lab demos for each lab task will be conducted at the end of each session.** If you miss the demo for any reason, no retake will be allowed, and that lab will be marked as 0.

5. **All quizzes will be unannounced**. Be prepared for surprise quizzes. Quizzes will cover content from previous labs as well as theory lectures from the corresponding theory course.

6. **You can take help from the internet for lab tasks,** but simply copying and pasting without understanding will be marked as 0. You should be able to explain the syntax or material used in your tasks.

7**. Do not ask for personal favors.** If you have concerns, such as short attendance, please speak with the relevant authority (e.g., academics).

8**. Students on warning:** Now is the time to study and earn a good grade. Do not ask for extra marks at the end if you are unable to clear your warning.

Single Commands

- **To clear the screen**
  Syntax: $ clear
- **To view commands history**
  Syntax: $ history
- **To display first 10 lines of a file**
  Syntax: $ head file.txt
- **To display first 6 lines of a file**
  Syntax: $ head —n 6 file.txt
- **To display first 5 lines from 2 files**
  Syntax: $ head —n 5 title .txt file2.txt
- **To display last 10 lines of a file**
  Syntax: $ tail file.txt
- **To display last 6 lines of a file**
  Syntax: $ tail —n 6 file.txt
- **The wc command without passing any parameter will display a basic result Of "file_txt' file. Result Of wc command will be (number of lines), (number or words) and (number of chars/bytes) of the file.**
  Syntax: $ wc file.txt
- **To display the number or characters in a file**
  Syntax: $ wc —c file.txt
- **To display the number of lines**
  Syntax: $ wc—l file.txt
- **To display the number of words**
  Syntax: $ wc —w file.txt
- **TO display number Of lines with numbers**
  Syntax: $ nl file.txt
- **To sort the content of file**
  Syntax: $ sort file.txt
- **To sort the content Of file in reverse orderz**
  Syntax: $ sort -r file-txt

- **To display the calendar.**
  Syntax: $ cal
- **To display system date.**
  Syntax: $ date
- **To display the login user details**
- Syntax: $ whoami
- **To display the contents of a file, line by line and page by page. we can use less command.**
  Syntax: $ less file.txt
- Note: to move up or down line by line use Arrow keys. And to move up page by page use 'b' key and for move down use Space key. To quit form the less command use ‘q' key.
- **To copy the file to another file**
  $ cp file1 -txt file2.txt
- **To copy a file within the other directory**
  $ cp file2.txt "Documents/directory1
- **To move a file to another directory**
  $ rmv file1.txt directory2
  $ mv —Documents/file1.txt directory3
- **To rename a file:**
- $ mv file1.txt file2.txt
- $ mv —Documents/file1.txt —/Documents/file2.txt
- **To delete a file**
  $ rm file1.txt
- $ rm Dir1/file2.txt

find command: Find command is used to search and locate the files and directories.

Find a specific file by name in current working directory and sub Directories of current working directory.

$ find —name filename

E.g: $ find —name file1.txt

Find a specific file by name in specified path(Absolute or Relative) and all sub

Directories of specified path.

$ find path —name filename

E.g: $ find ~/ Videos —name file2.txt

Find specific directory by name in current working directory and sub Directories Of current working directory.

$ find —name directory

E.g: $ find —name D1

Find a specific directory by name in specified path(AbsoJute or Relative) and all sub Directories of specified path.

$ find path —name filename

E.g: find ../Music —name M1

Ownership and Permissions

In Linux. we use permissions to control what a user can do with a file or directory.

Types of permissions

Linux uses three types of permissions:

- **Read:** For a file, the read permission allows a user to view the Contents Of a file. For directories. the read permission allows the user to view the names of files and other directories stored in it.
- **Write:** For a file. the write permission allows a user to create. modify a file. For directories. the permission allows a user to modify Its contents (create. delete rename files in it).

However. this permission has no effect on directories unless the execute permission is also enabled.

- **Execute:** When set on a file. it works files that are executable like script files. C executable files etc. However. the permission has no effect on a file unless the read permission is also enabled. On the other hand. tor directories. the execute permission allows the user to enter the directory.

Permission Classes

For any file or directory. there are three types of "permission classes". You can assign different permissions to these classes, and thus control who can access and modify files. The permission classes are as follows:

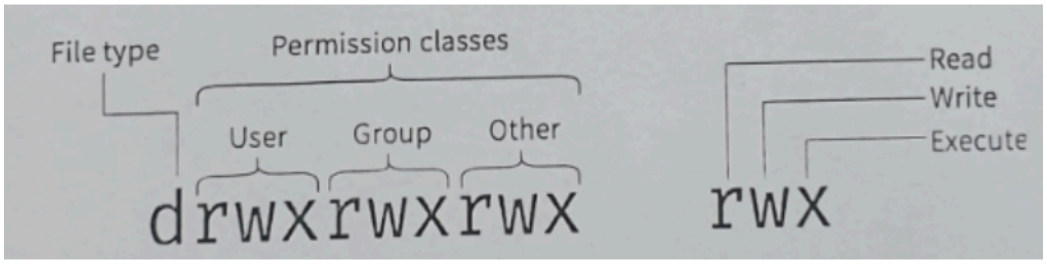
- **User:** Permissions in this class affect the owner of the file.
- **Group:** Permissions in this class affect the group which owns the file. However. if the owner user is in this group, the "user" permissions apply. instead of the group permissions.
- **Other:** Permissions in this class affect all other users on the system.

Viewing permissions

The easiest way to view the permissions of files in a given directory is to run:

```
ls -l
```

After the execution of command (ls Output of command shows, the "file mode" column which displays the file type and permissions in a compact way, as shown below:



The first character represents the type of file, e.g A normal file is represented with a -, directories with a d etc. I-hen. we have the permissions often corresponding to the “user". "group" and "other" classes. The triplet rwx represents whether these classes have the read. write and execute permissions. If a - appears in this triplet. it means (hat the corresponding permission is disabled.

Using Chmod Command to Change File Permissions

As all Linux users. you will at some point need to modify the permission settings of a file/directory.

The command that executes such tasks is the chmod command.

The basic syntax is:

chmod [permission] [file_name]

There are two ways to define permission:

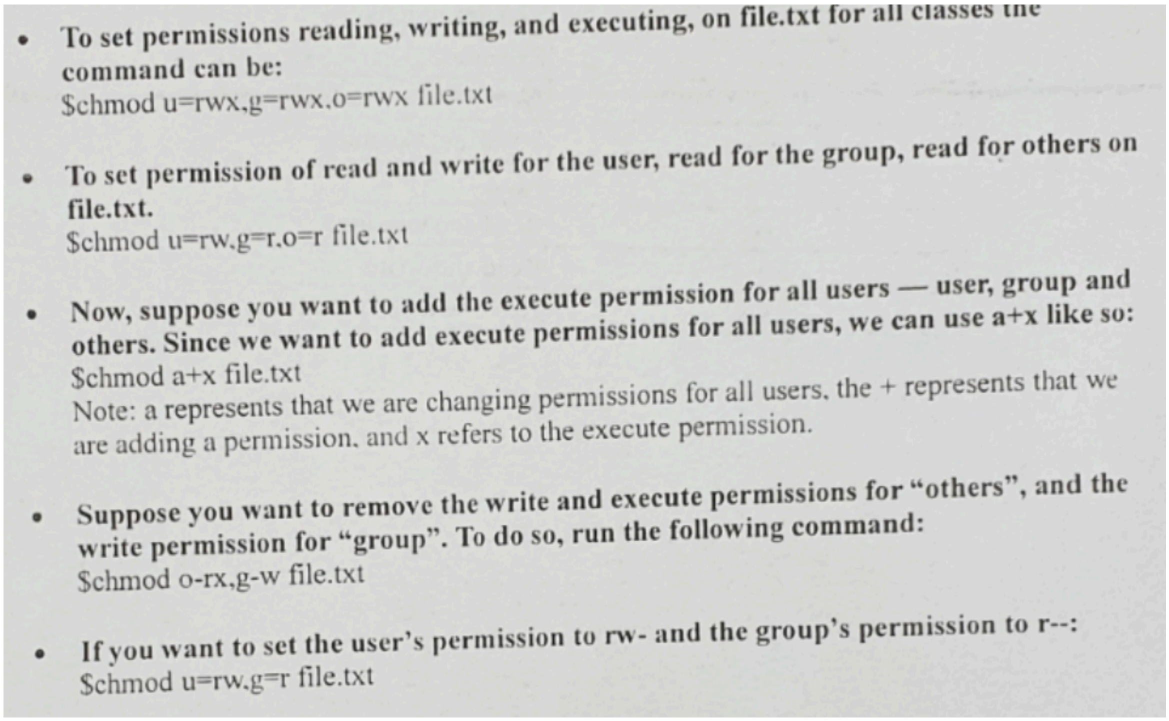
l. using symbols (characters)

2. using the octal notation method

Define File Permission with Symbolic Mode

To specify permission settings using alphanumerical characters. you'll need to define accessibility for the user or owner (u). group (g). and others (o).

Example Commands using Symbolic Mode:

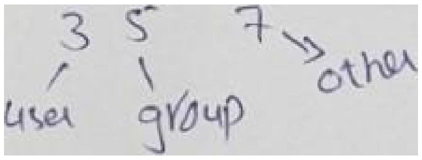


- To set permissions reading, writing, and executing, on file.txt for all classes the command can be:
  $chmod u=rwx,g=rwx,o=rwx file.txt
- To set permission of read and write for the user, read for the group, read for others on file.txt.
  $chmod u=rw,g=r,o=r file.txt
- Now, suppose you want to add the execute permission for all users — user, group and others. Since we want to add execute permissions for all users, we can use a+x like so:
  $chmod a+x file.txt
  Note: a represents that we are changing permissions for all users, the + represents that we are adding a permission, and x refers to the execute permission.
- Suppose you want to remove the write and execute permissions for "others", and the write permission for "group". To do so, run the following command:
  $chmod o-rx,g-w file.txt
- If you want to set the user's permission to rw- and the group's permission to r--:
  $chmod u=rw,g=r file.txt

Define File Permission with octal/numeric Mode

Another way to specify permission is by using the octal/numeric format. This option is faster, as it requires less typing, although it is not as straightforward as the previous method instead of letters. the octal format represents privileges with numbers:

- read has the value of4
- write has the value of 2
- execute has the value of I
- no permission has the value of 0



The privileges are summed up and depicted by one number. Therefore. the possibilities are:

- 7 — for read. write. and execute permission
- 6 — for read and write privileges

chmod 774 file.txt

above command gives read, write and execute to user and groups while read only to others.

su vs. sudo commands

This is a key difference between **su and sudo**. su(short for substitute user) switches you to the root user account and requires the root account’s password. sudo(short for superuser do) runs a single command with root privileges – it doesn’t switch to the root user or require a separate root user password.

The Root User

Both su and sudo are used to run commands with root user permissions. The root user is basically equivalent to the administrator user on Windows – the root user has maximum permissions and can do anything to the system. Normal users on Linux run with reduced permissions – for example, they can’t create users, install software. To do something that requires these permissions, you’ll have to acquire them with su or sudo. Eg: sudo passwd root

su command

The su command switches to the super user or root user from current login user. You’ll have to enter the root account’s password. This isn’t all the su command does, though – you can use it to switch to any user account. If you execute the **su Ahmed** command, you’ll be prompted to enter Ahmed user password and the shell will switch to Ahmed’s user account. Once you’re done running commands in the root shell, you should type **exit** to leave the root shell and go back to limited-privileges mode. Following command will switch to root user account after entering the root user password.

$ su -

sudo command

sudo runs a single command with root privileges. With sudo, one or more users are granted superuser privileges on an as needed basis. To execute a command as the superuser, the desired command is simply preceded with the sudo command. After the command is entered, the user is prompted for the their own password rather than the superuser's. By default, password for fifteen minutes and won't ask for a password again until the fifteen minutes are up. Following syntax will execute the command having superuser privileges after entering own password by the user.

$ sudo command

Enabling the Root User

By default root user is disabled. To enable the root user account, use the following command to set a password for it.

$ sudo passwd root

Sudo will prompt you for your current user account's password before you can set a new password of root password. Use your new password to log in as root with the su command.

National University
of computer and emerging sciences

User Management in Linux

Adding a User in Linux

- To add users, using the useradd command: $sudo useradd -m ali
  For example, if you want to add the user named Ahmed, then the command will be like: $sudo useradd -m Ahmed Note: useradd create a home directory, we've used the -m switch.

Setting the Password of User

- At this point, the user has been created, but they don't have a password and can't log in. So, to assign a password to the newly created user, run the passwd command: $sudo passwd For example, to set the password of Ahmed: $sudo passwd Ahmed.
  Note: The command will ask for the new password, and ask you to confirm it:

Adding a user and setting its password in a single command

- To add a new user and also set its password, we can use –p option to set its password:
  For example $sudo useradd –m Ahmed –p 123

Deleting a User Account

- We can use the userdel command to delete user account and all the related directories/files. $ sudo userdel -r Asif

Every user has some user ID(a unique ID is assigned to a user after its creation).

- Following command can be used to view the id of a user Ahmed.

  $ id -u Ahmed

Users and groups

In order to make managing users easier, you can add users into a "group". A group can have zero or more users. A particular user is associated with a "default group"(primary group) having same name as of username, and can also be a member of other groups(Secondary groups) on the system.

- **Following command can be used to view the name of primary group assigned to a specific user.**

  For Example, we want to view the primary group of user "userone".

  $ id –gn userone
- **Following command can be used to view the id of primary group assigned to a specific user(Note: each group in linux is also assigned a id for group identification)**. For Example, we want to view the id of primary group of user "userone".

  $ id –g userone
- **Following command can be used to view the list the names of groups(all groups), in which user is a member.**

  $ id -Gn userone

  Note: First group that will be displayed in the list will be primary group(primary group has the same name as of user name) and rest of groups will be secondary groups.

- **Following command can be used to view the list the ID's of groups(all groups), in which user is a member.**
  $ id -G userone
- **Following command can be used to make a user a member of a group. In following example user Ahmed has been made as a member of sudo group.** $ sudo usermod –a -G sudo Ahmed
- **Following command can be used to remove a user from a group. In following example user Ahmed has been removed from sudo group.**
  $ sudo gpasswd –d Ahmed sudo OR $ sudo deluser Ahmed sudo
- **Following command can be used to add/create a new group. For example g2 is created/added.**
  $ sudo groupadd g2
- **Following command can be used to remove a group. For example g2 is removed.**
  $ sudo groupdel g2

<u>Software/Package Management</u>

apt-get is a command which helps in handling packages/softwares in Linux. Its main task is to retrieve the information and packages from the authenticated sources for installation, upgrade and removal of packages. Here APT stands for the Advanced Packaging Tool.

Most Used Commands:

- **update :** The update command is used to download package information from all configured sources. So when you run update command, it downloads the package information from the Internet. It is useful to get info on an updated version of packages. You need to perform an update before you upgrade. $sudo apt-get update

- **upgrade :** This command is used to install the latest versions of the packages currently installed on the user's system. The installed packages which have new packages available are retrieved and installed. You need to perform an update before the upgrade, so that apt-get knows that new versions of packages are available. $sudo apt-get upgrade
- **install :** This command is used to install packages. In the following example vlc mediaplayer is being installed. $sudo apt-get install vlc
- **remove :** This is similar to install, with the difference being that it removes the packages instead of installing. $sudo apt-get remove vlc

Process Management

Linux bash terminal has a number of commands that can be used to find information about the running processes.

- **$ ps command**

  The ps command lists running processes. The output includes information about the shell(bash terminal) and the process running in the shell.
- **$ ps –e command**

  The ps command with –e option will display all the processes in the system.
- **$ ps –ef command**

  The ps command with –f option will provide full-format listing, which provides detailed information about the processes.
- **$ ps –C ProcessName Command**

Using ps command with –C option and giving the process name, it will as a result give us the PID(process id) of specified process name.

- **$ top command**

  The top command provide the view of processes according to resource usage and see the processes that are taking up the system resources.

- **$ kill ProcessID command**

  The kill command signal can kill a process, given its process ID.

Other Shell Commands

$ whoami command

The whoami command prints the user name who currently logged-in.

$ hostname command

A hostname command is used to display the computer's hostname.

$ cal command

cal command is used to display the calendar.

$ which CommandName

command which command in Linux is a command which is used to locate the executable file associated with the given command. For example: “$which ls” will give us the output “/bin/ls”. Means it displays the path where that executable ls is stored.

$ echo SomeString command

echo is a command that outputs the string it is being passed as arguments. For example: $echo “Hello” will give us the output “Hello”

pipe command

In linux, the pipe command(represented as | symbol), sends the output of one command to another command as an input. For Example: In the following example output of ls –l command is send to less command as an input. $ ls –l | less

Tasks:

Create a file example.txt with at least 20 lines in home directory.

I. Display the first 5 lines of the file.
II. Display the first 10 lines of the file.
III. Display the last 5 lines of the file.
IV. Display lines 6-10.
V. Display the first 100 bytes:
VI. Display the last 100 bytes:

Create another file example2.txt with at least 30 lines.

VII. Display the first 3 lines of both files.
VIII. Display the last 3 lines of both files.
IX. Search for a file named `example.txt` in your home directory.
X. Find all `.txt` files in the `/home/user/documents/` directory.
XI. Find all files larger than 100MB in the `/var/log/` directory.
XII. Find all files smaller than 1KB in your current directory.
XIII. Find all empty files in `/var/` directory.
XIV. Find all empty directories in your home directory.
XV. Change the permissions of `example.txt` so that only the owner has read, write, and execute permissions, while the group and others have no permissions. Using octal number.
XVI. Change the permissions of `example.txt` so that everyone (owner, group, others) has read and write permissions. Using octal number

XVII. Change the permissions of `example.txt` so that it is read-only for the owner, group, and others. Using octal number.

XVIII. Change the permissions of `example.txt` so that others (everyone except the owner and group) have no permissions at all.

XIX. Change the permissions of `example.txt` so that the group and others lose their write permissions but retain any other existing permissions.

XX. Use symbolic notation to give the owner execute permission, the group read and execute permissions, and others only execute permission.

XXI. Change the permissions of example.txt so that everyone (owner, group, others) can execute the file, without affecting other permissions. Without using u, g and o.