National Technical University of Ukraine

“Igor Sikorsky Kyiv Polytechnic Institute”

Faculty of Informatics and Computer Science

Department of Information Systems and Technologies

Laboratory work № __5

from the discipline «LINUX »

Subject: « Linux streams.»

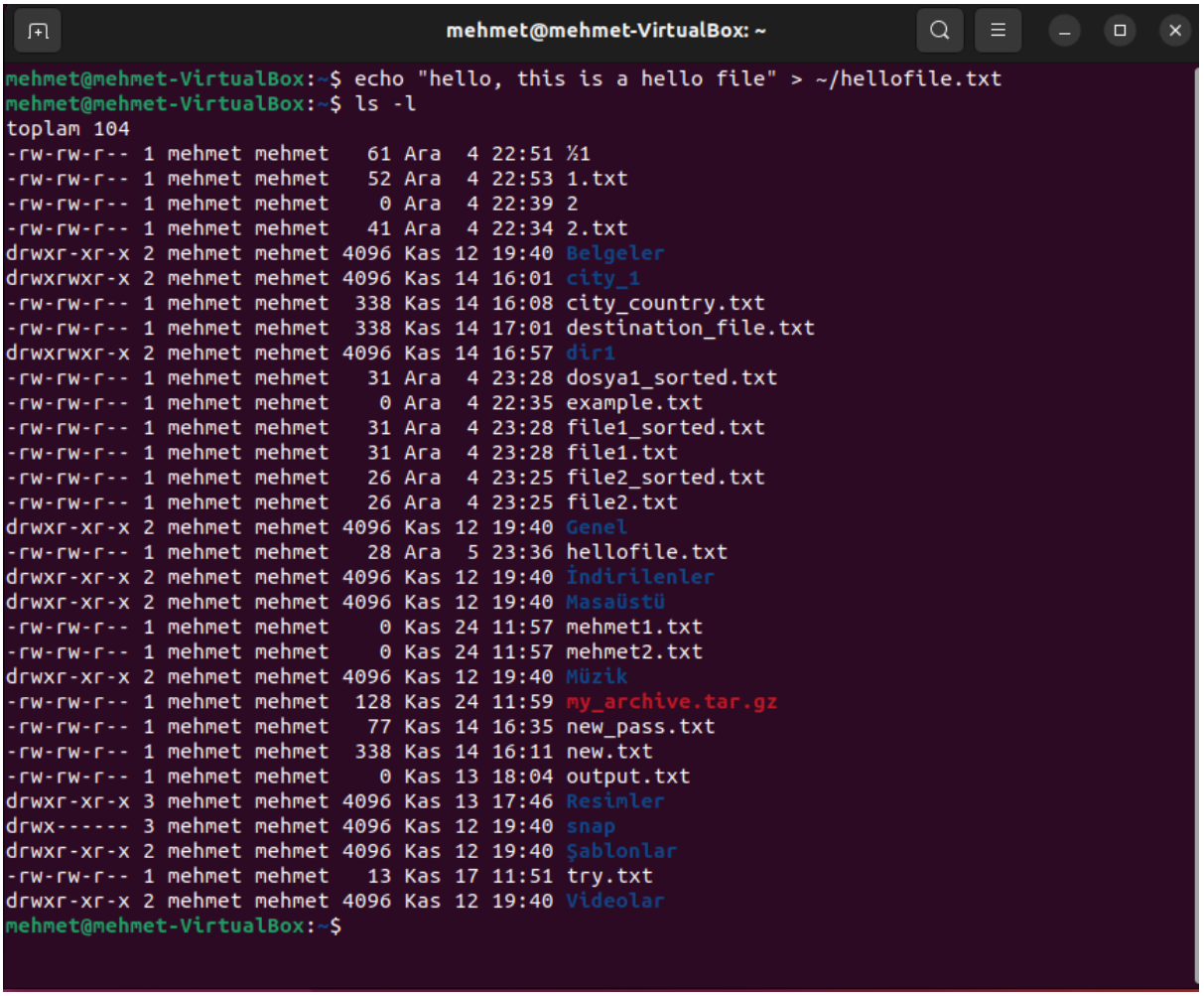
Performed by: student of group IM-14 FIOT Full name Mehmet KULUBECİOGLU	Checked:: Senior lecturer of Department ST Maryna Khmeliuk

KYIV 2023

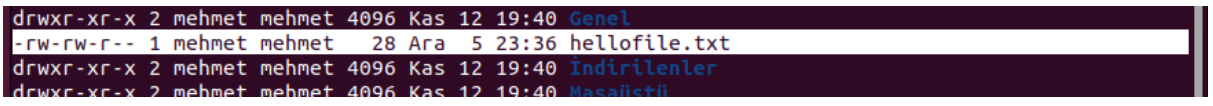
TASK 1 :

1) Create one text file in your HOME directory. Display the full catalog listing.

We can see that we have created a hellofile.txt in home directory and displayed the full catalog list.

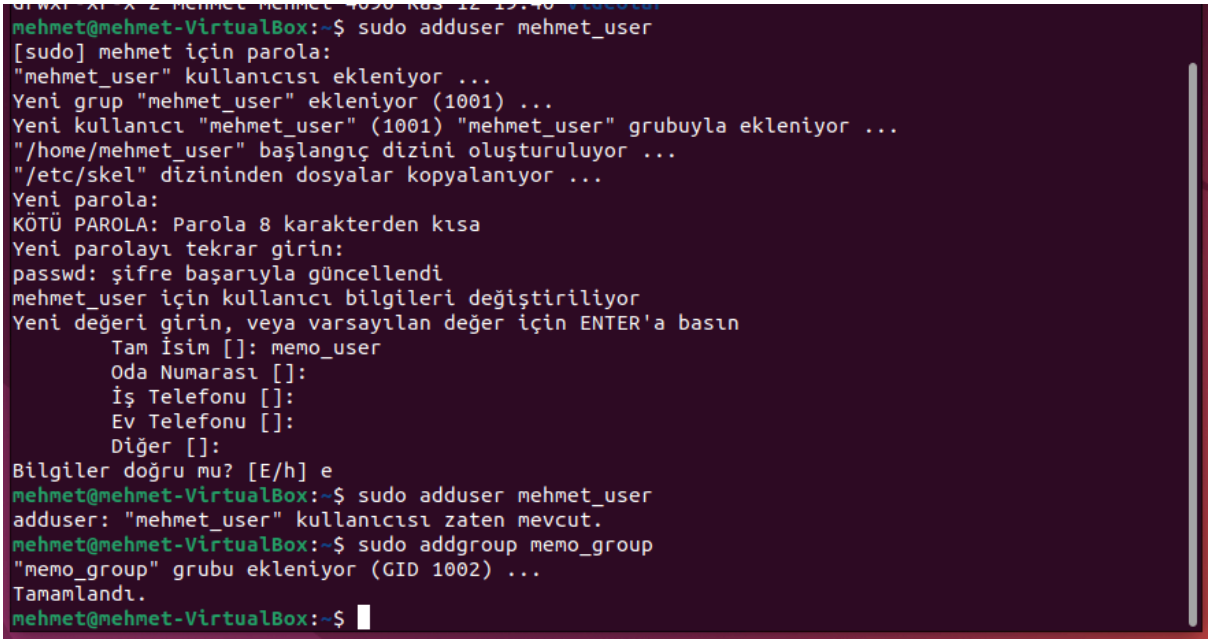


Here,we can see our hellofile.txt file.



2) Create a user and a group. Add the owner of the file and the new user to the group.

Here we have created a user name ‘new_user’ and set the required details as presented in the screenshot below. Also a group named ‘new_group’ is created.



Now we add the owner of the file (mehmet) and the mehmet_user to the memo_group group.

```
mehmet@mehmet-VirtualBox:~$ sudo usermod -aG memo_group mehmet
mehmet@mehmet-VirtualBox:~$ sudo usermod -aG memo_group memo_user
usermod: 'memo_user' kullanıcısı yok
mehmet@mehmet-VirtualBox:~$ sudo usermod -aG memo_group mehemt
usermod: 'mehemt' kullanıcısı yok
mehmet@mehmet-VirtualBox:~$ sudo usermod -aG memo_group mehmet
mehmet@mehmet-VirtualBox:~$ sudo usermod -aG mehmet_user
Usage: usermod [options] LOGIN

Options:
  -b, --badnames                allow bad names
  -c, --comment COMMENT         new value of the GECOS field
  -d, --home HOME_DIR           new home directory for the user account
  -e, --expiredate EXPIRE_DATE  set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -g, --gid GROUP               force use GROUP as new primary group
  -G, --groups GROUPS           new list of supplementary GROUPS
  -a, --append                  append the user to the supplemental GROUPS
                                mentioned by the -G option without removing
                                the user from other groups
  -h, --help                    display this help message and exit
  -l, --login NEW_LOGIN         new value of the login name
  -L, --lock                    lock the user account
  -m, --move-home               move contents of the home directory to the
                                new location (use only with -d)
  -o, --non-unique              allow using duplicate (non-unique) UID
  -p, --password PASSWORD       use encrypted password for the new password
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -s, --shell SHELL             new login shell for the user account
  -u, --uid UID                 new UID for the user account
  -U, --unlock                  unlock the user account
  -v, --add-subuids FIRST-LAST  add range of subordinate uids
  -V, --del-subuids FIRST-LAST  remove range of subordinate uids
  -w, --add-subgids FIRST-LAST  add range of subordinate gids
```

3) Analyze and explain what access rights the owner of the file, his group and other users have to this file.

~ We can see the owner name and his group,

The rights of the owner is = rw- (read & write)

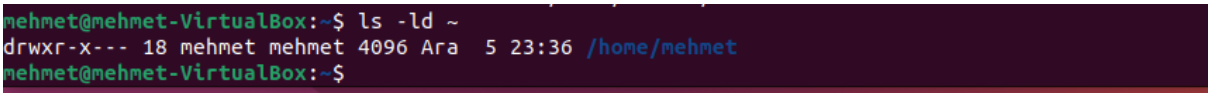
The rights of the group is = rw- (read & write)

The rights of others is = r- - (read only)

```
mehmet@mehmet-VirtualBox:~$ ls -l ~/hellofile.txt
-rw-rw-r-- 1 mehmet mehmet 28 Ara  5 23:36 /home/mehmet/hellofile.txt
mehmet@mehmet-VirtualBox:~$
```

4) Analyze the access rights to your home directory. Are there any restrictions on working with files in this directory?

~Here we list the rights of our current directory, which is all(rwx) for user, (r-x) read & execution for group and no rights for others.

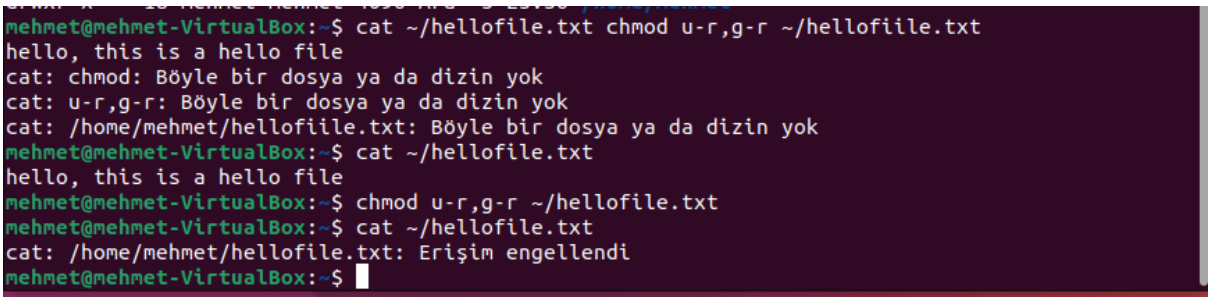


5) Display the contents of the file on the screen. Explain why the operation was successful.

6) Remove the right to read the file to the owner and the group, try to display the text of the file on the screen. Explain why the operation is not performed.

Firstly we display the contents of the hellofile.txt file using cat command.Then we change the permissions of the file as - removing the read right for user and group.

Then we try to display the contents of the file, it is denied as the read permission is removed for user.



7) Remove write permissions to the file. Try adding text to the file. Explain the result.

~ The write permission of the hellofile.txt file for user is removed using the chmod command, then we try to add some text to the file - as we can see the permission is denied as we have used(-w) and remove the write right for the user.

```
mehmet@mehmet-VirtualBox:~$ chmod u-w ~/hellofile.txt
mehmet@mehmet-VirtualBox:~$ echo "some text" >> ~/hellofile.txt
bash: /home/mehmet/hellofile.txt: Erişim engellendi
mehmet@mehmet-VirtualBox:~$
```

8) Remove the right to modify the directory. Delete the file. Explain the result.

Here we modified the permissions of the directory and tried to remove the hellofile.txt. We see that the action is denied as we have used -w for user.

```
mehmet@mehmet-VirtualBox:~$ chmod u-w ~
mehmet@mehmet-VirtualBox:~$ rm hellofile.txt
rm: yazma korumalı normal dosya 'hellofile.txt' kaldırılsın mı? Y
rm: 'hellofile.txt' silinemedi: Erişim engellendi
mehmet@mehmet-VirtualBox:~$
```

9) Create a subdirectory. Create a text file in it. Analyze the access rights to the subdirectory and explain the possibilities for using the subdirectory.

A directory named ‘subdir’ is created with a text file in it with contents. We check the rights of the directory which shows all rights to user and group , read and attribute execution rights for others. We can use the directory as we need it to because as the user we have all permissions(rwx).

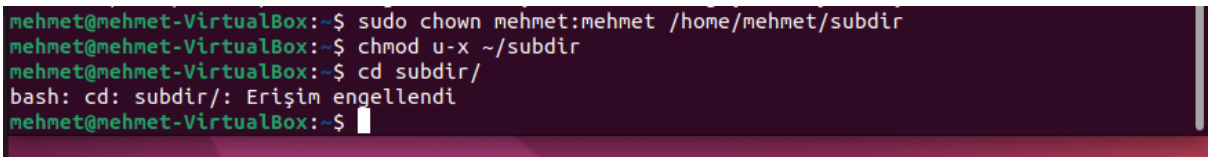
```
mehmet@mehmet-VirtualBox:~$ mkdir subdir
mkdir: `subdir' dizini oluşturulamıyor: Dosya var
mehmet@mehmet-VirtualBox:~$ echo "This is a testt file))" > ~/subdir/testfile.txt
bash: /home/mehmet/subdir/testfile.txt: Erişim engellendi
mehmet@mehmet-VirtualBox:~$ ls -ld ~/subdir
drwxr-xr-x 2 root root 4096 Ara  6 00:22 /home/mehmet/subdir
mehmet@mehmet-VirtualBox:~$
```

10) Remove the owner's right to "execute" the subdirectory.

11) Try to make the subdirectory current. Explain the result.

Here we remove the owner's right to "execute" the /subdir/

Here we try to change our current directory to the subdir directory but the permission is denied as we have removed the owner's right for execution.



12) View the contents of the subdirectory. Explain the result.

We can list the contents of the directory as the read permission is still allowed

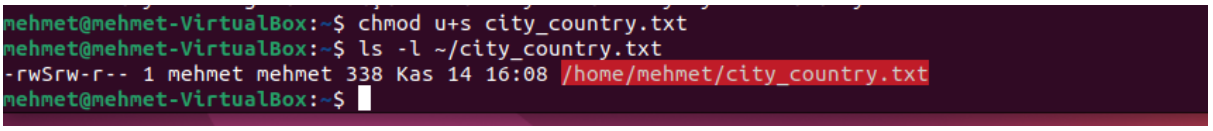
CONCLUSION: (Part I)

To sum up everything, we can derive from the first task the fundamental uses of commands like - File permissions control read, write, and execute access for the owner, group, and others; Directory permissions additionally control the ability to list contents (read), create or delete files inside (write), and enter (execute); Modifying permissions affects users' ability to perform various operations on files and directories; Security considerations involve granting the minimum necessary permissions to users and groups; Regularly reviewing and adjusting permissions is essential for maintaining a secure and functional file system.

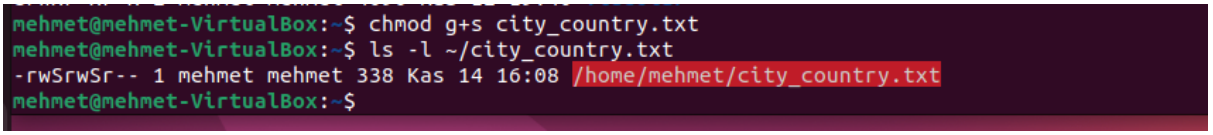
Task 2

Familiarize yourself with the SUID, SGID, and STICKY BIT. Learn to use them.

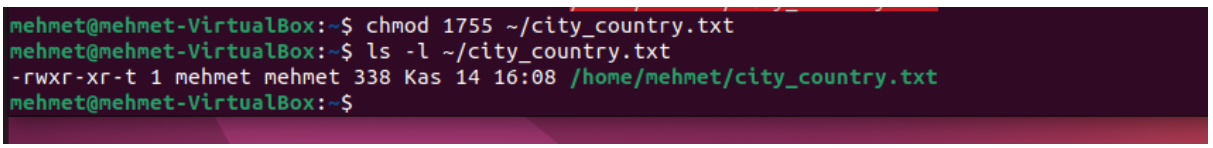
Here we use the +s to the user. When the SUID permission is set on an executable file, the program will run with the permissions of the file owner rather than the permissions of the person executing the program. We see capital ‘S’ which means the user did not have execution rights before SUID was set.



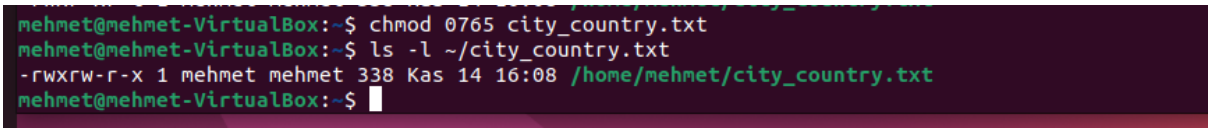
Here we use the +s to the group. Similar to SUID, but it sets the group ownership of the process to the group owner of the file. We see capital ‘S’ which means the group did not have execution rights before SUID was set.



Here we use the octal value for sticky bit with 755 permissions. When the Sticky Bit is set on a directory, only the file owner can delete or rename their own files within that directory, even if others have write permissions to the directory



Here we remove all the extra bits from the city.txt file with 765 permissions.



CONCLUSION: (Part II)

In summary, these permissions like - SUID, SGID & STICKY BIT features provide a way to fine-tune access to files and directories, enhancing security and facilitating collaboration in a multi-user environment. However, they should be used carefully, as incorrect application may lead to security vulnerabilities. We must always consider the principle of least privilege when assigning permissions.