“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №10**

з дисципліни: «Операційні системи»

**Тема: «Зміна власників і прав доступу до файлів в Linux. Спеціальні каталоги та файли в Linux»**

Виконали

студенти

групи БІКС-13

Команда «chipichapa»:

Ємець В.О.,

Лапчик С.С.,

Савустьян М.В.

Перевірив викладач

Сушанова В.С.

Робота студентів групи БІКС-13 Команда «chipichapa»: Ємець В., Лапчик С., Савустьян М.

Київ 2024

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими діями при зміні власників файлів, .прав доступу до файлів
3. Знайомство з спеціальними каталогами та файлами в Linux.

**Матеріальне забезпечення занять:**

1. ЕОМ типу IBM PC. 2. ОС сімейства Windows та віртуальна машина Virtual Box (Oracle). 3. ОС GNU/Linux (будь-який дистрибутив). 4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки:**

***Готувала матеріал студентка Лапчик Софія***

\*Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань призначення команд та їх параметрів.  
chmod – command to change permissions.  
stat – displays more detailed information about a file.  
-i – prints the index number of the file.  
-R – to change the group ownership of all of the files of a directory structure (option to the chgrp command).

1. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. Яке призначення команди id?  
      The id command is used to display information about the current user, group, and user and group IDs. This information can be useful for understanding access rights to files and directories,
   2. Як переглянути які права доступу має власник файлу?  
      Using the ls command: The ls command with the -l flag can display detailed information about file permissions, including owner, group, and other user permissions. Using the stat command: The stat command can display more detailed information about a file, including the access rights of the owner, group, and other users.
   3. \*Як змінити власника групи?  
      You can change the group owner of a file using the chown command. For example, to change the group owner of the file /home/myfile to the www-data group, run the following command:  
      sudo chown :www-data /home/myfile
   4. \*Як можна переглянути у терміналі який тип поточного файлу?Наведіть приклади для різних типів файлів  
      The file command can determine the type of a file based on its contents.  
      file /home/myfile  
      The mimeinfo command can determine the type of a MIME file. A MIME type is a standardized way of specifying a file type.  
      mimeinfo /home/myfile  
      -rw-r--r-- 1 centos centos 1234 May 24 2024 /home/myfile  
      /home/myfile: ASCII text, with CRLF line terminators  
      drwxr-xr-x 1 centos centos 4096 May 24 2024 /home/mydir
   5. \*\*Для чого використовуються дозволи Setuid та Setgid?  
      Setuid: When a program is given Setuid permission, it runs with the UID of the file's owner, regardless of which user runs it. This can be useful for programs that need access to files or directories owned by root. For example, the sudo command has Setuid permission, which allows users to execute commands with root privileges after entering a password. Setgid: When a program is given Setgid permission, it runs with the GID of the file's owner, regardless of which user runs it. This can be useful for applications that need access to files or directories that belong to a specific group. For example, the nginx command usually has Setgid permission, which allows it to create files with the nginx group.
   6. \*\*Для чого в системі потрібен так званий “липкий біт” (Sticky Bit). Наведіть приклади коли цей дозвіл доцільно використовувати.  
      "Sticky Bit" is a permission flag that can be set for directories. When the "sticky bit" is set for a directory, files deleted from that directory are retained until the system is rebooted. This can be useful for directories that contain important files that should not be accidentally deleted.  
      For example, the /tmp directory usually has the "sticky bit" set. This means that files deleted from /tmp are retained until the system is rebooted, giving users the ability to recover them if they were accidentally deleted. Directories used to store temporary files created by users. Directories used to store logs. Directories that contain important configuration files.

**Хід роботи:**

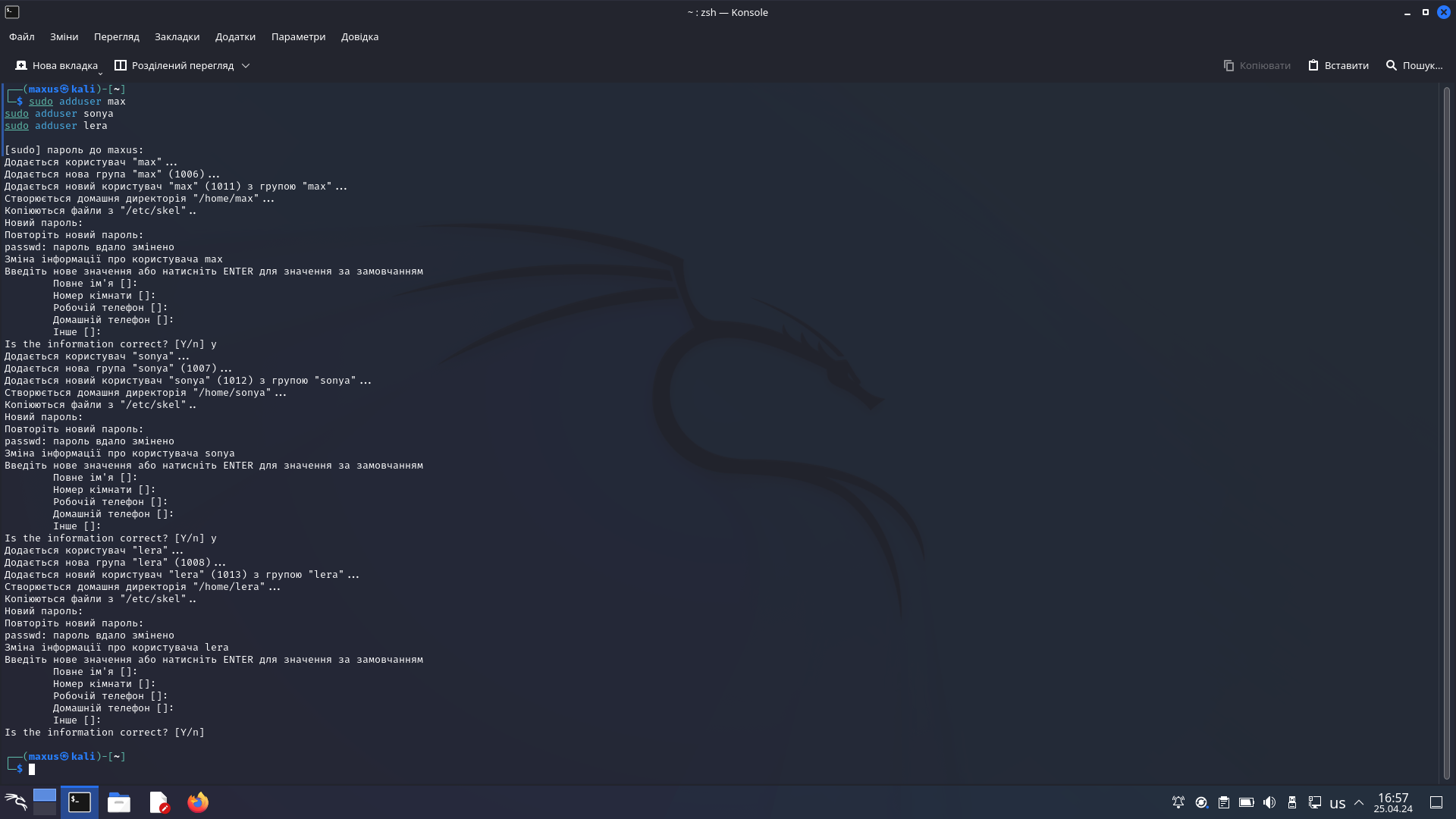
***Готувала матеріал студентка Лапчик Софія***

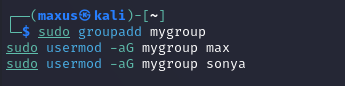
1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
   1. Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її. Виконайте вхід в систему під користувачем: CentOS, пароль для входу: reverse ***(якщо виконуєте ЛР у 401 ауд.)*** та зпустіть термінал.
   2. Запустіть віртуальну машину Ubuntu\_PC ***(якщо виконуєте завдання ЛР через академію netacad)***
   3. Запустіть свою операційну систему сімейства Linux ***(якщо працюєте на власному ПК та її встановили)*** та запустіть термінал.
2. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу ***NDG Linux Essentials: Lab 17: Ownership and Permissions*** та ***Lab 18: Special Directories and Files.*** Створіть таблицю команд вивчених у п.2 ходу роботи у наступному вигляді:

|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| chmod | command to change permissions |
| -l | file ownership can be confirmed using the long listing option of the ls command. |
| -a | hidden files, which begin with the dot . character are listed using this option |
| -d | option for the ls command lists directory information |
| -i | prints the index number of the file |
| ln | to create another hard link to the source file |
| rm | used to remove files |
| stat | displays more detailed information about a file |
| -R | to change the group ownership of all of the files of a directory structure (option to the chgrp command) |

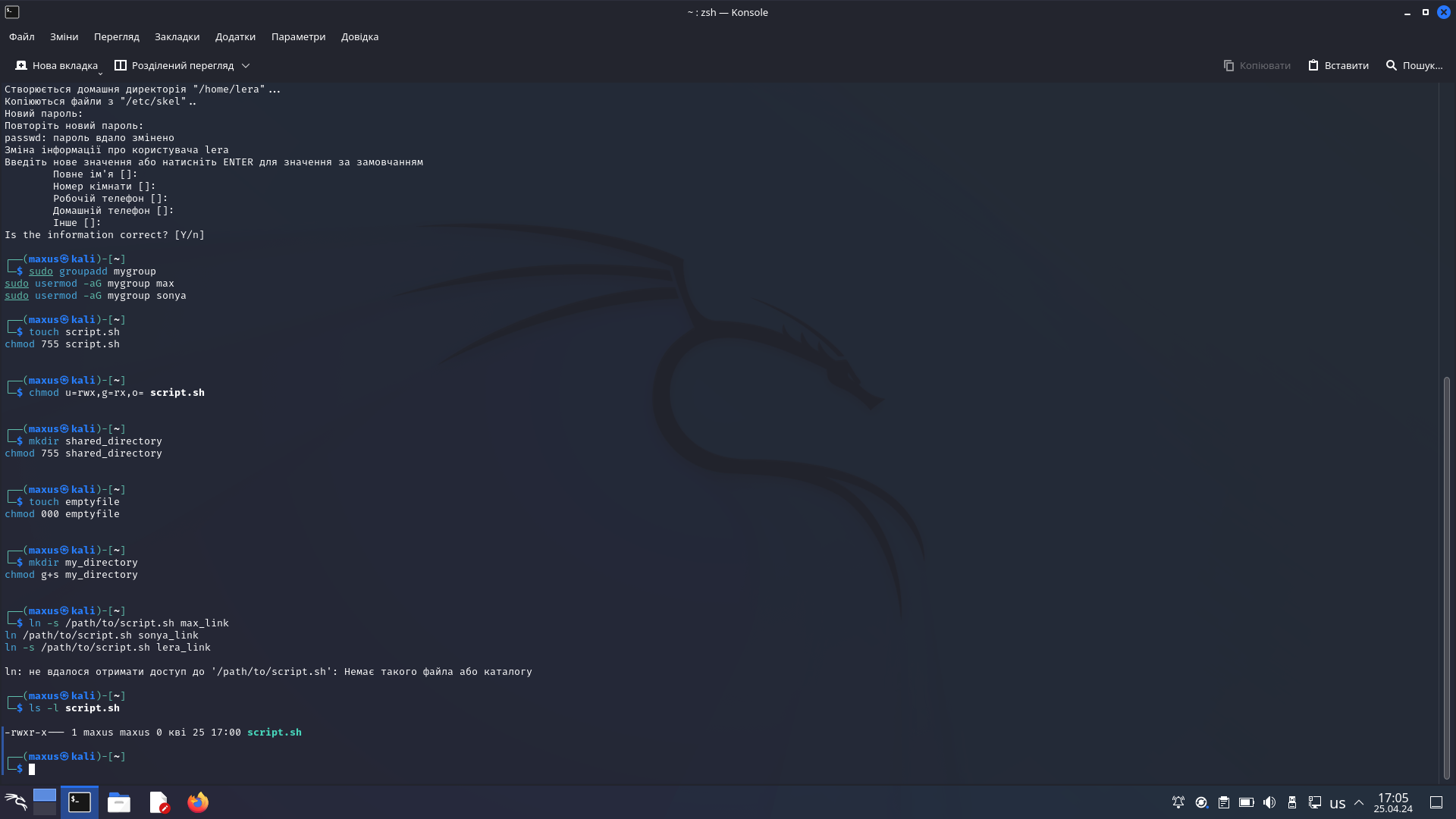
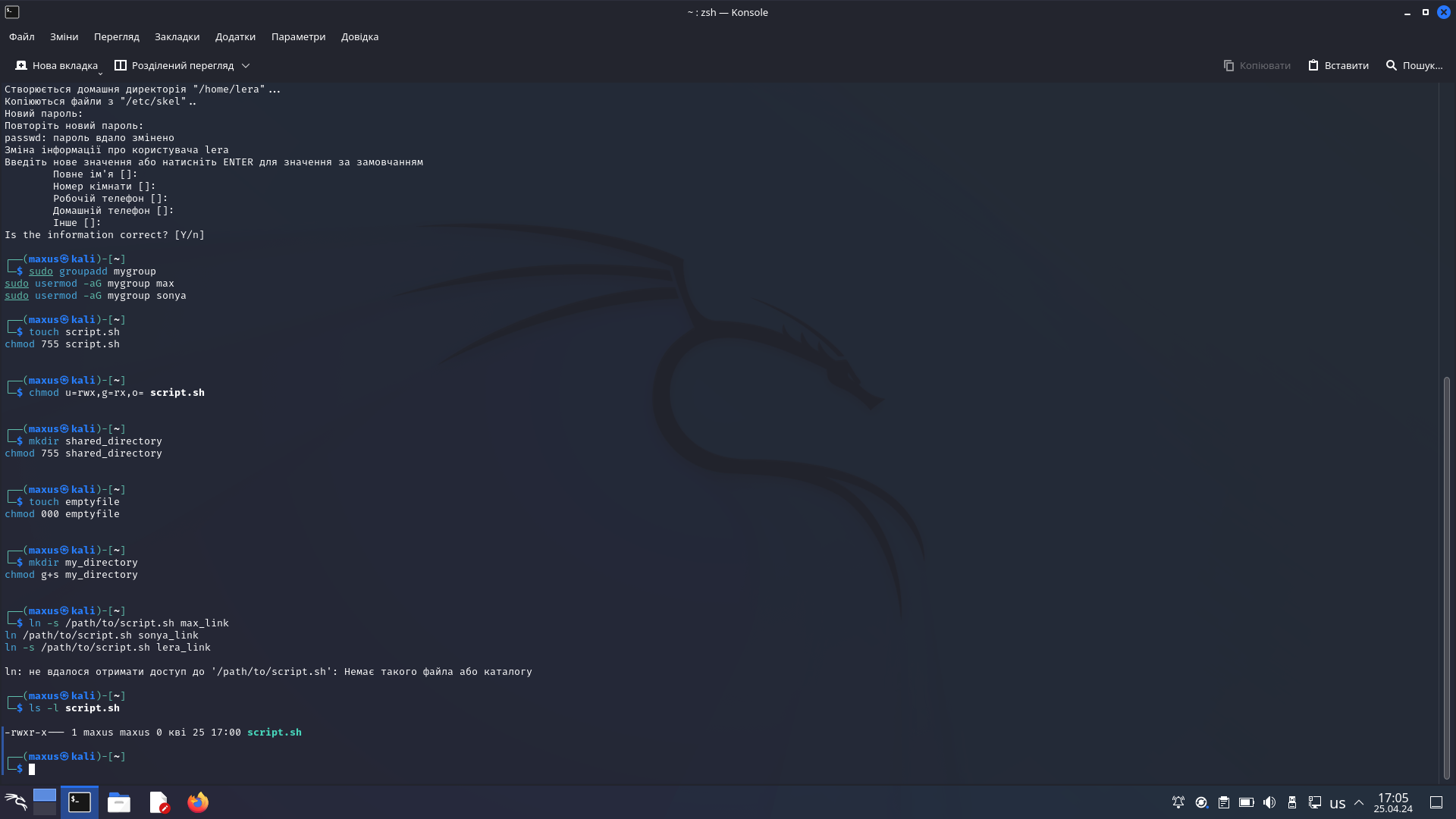
***Готував матеріал студент Савустьян Максим***

1. Виконайте наступні практичні завдання у терміналі наступні дії (продемонструвати скріншоти):

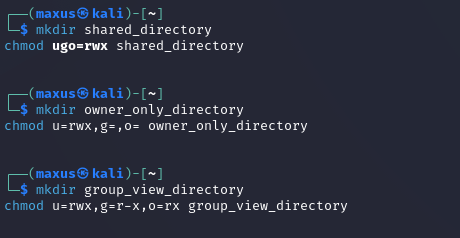
* створіть трьох нових користувачів;
* 
* створіть нову групу користувачів, туди додайте двох, з трьох створених користувачів;



* створіть новий файл, який буде доступний на зчитування, редагування та виконання власником файлу, наприклад найпростіший скриптовий сценарій; для користувачів групи власника надайте дозволи на перегляд та виконання (без дозволу на редагування) цього файлу; для інших користувачів заборонити доступ до цього файлу;

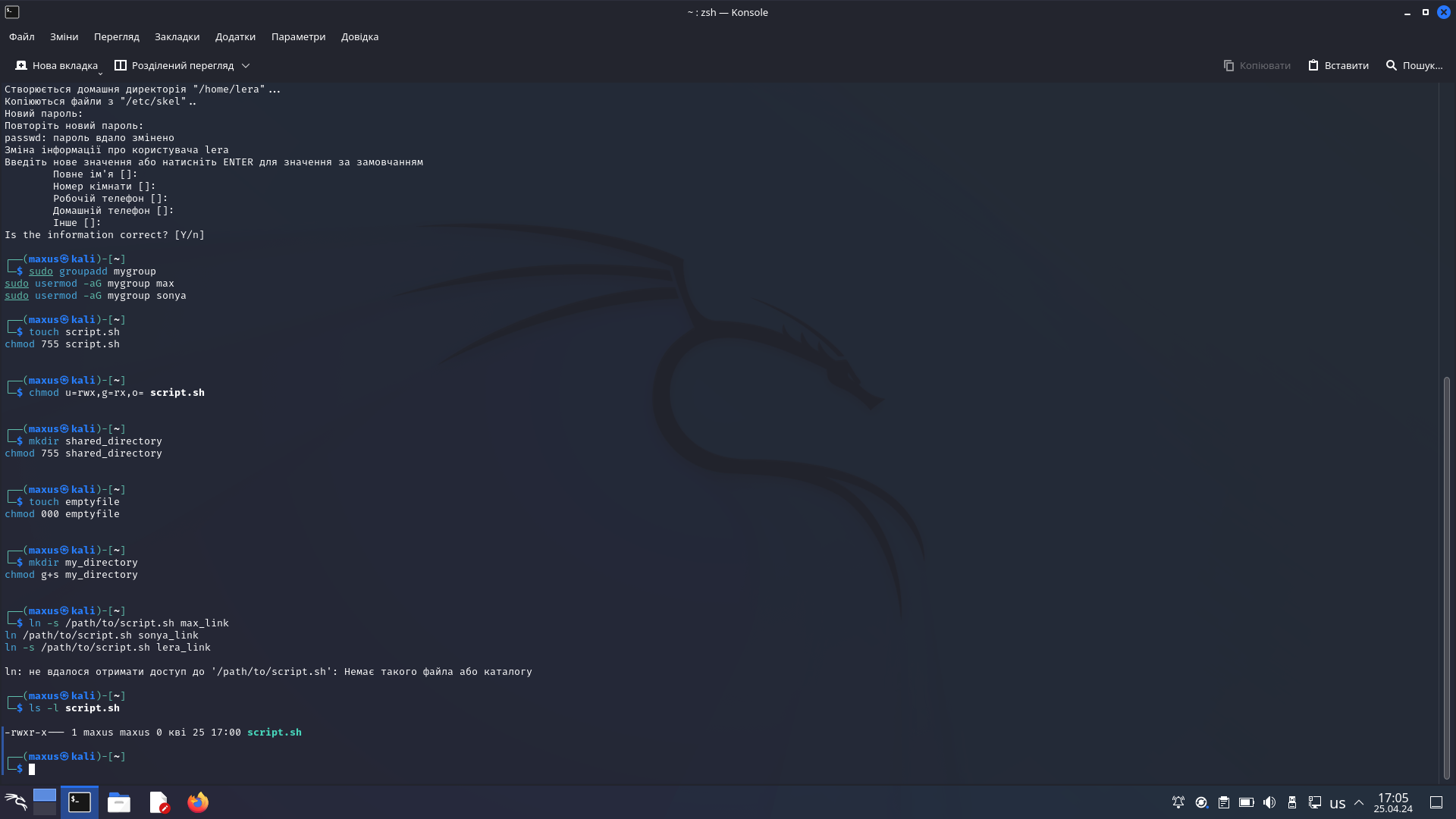


* \*подібні дії виконайте для директорій - створіть директорію, яка буде доступна для всіх трьох користувачів, створіть директорію, яку буде доступна тільки для власника, створіть директорію, яку користувачі групи власника зможуть переглядати, але не редагувати;



We created 3 derivatives with the following requirements

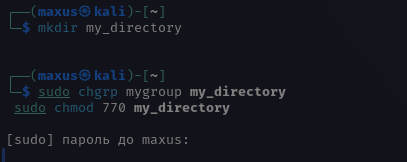
* \*створіть порожній файл під назвою emptyfile за допомогою команди touch emptyfile. Тепер “обнуліть” дозволи для файлу з chmod 000 emptyfile. Що станеться, якщо змінити дозволи для emptyfile, передавши лише одне значення для chmod у числовому режимі, наприклад, chmod 4 emptyfile? Що буде, якщо ми використаємо два числа, наприклад chmod 44 emptyfile? Що ми можемо дізнатися про те, як chmod зчитує числове значення?



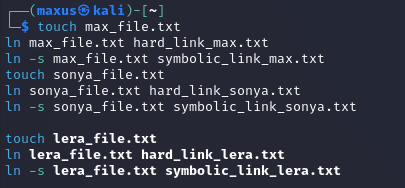
chmod 4 emptyfile, this will set read-only permission for the file owner. That is, the owner of the file will be able to read the contents of the file, but will not be able to edit or execute.

If there are two numbers, for example, like `chmod 44 emptyfile`, the first set of numbers corresponds to the permissions of the file owner, and the second set corresponds to the permissions for the file group. In this case, both sets of digits are 4, which means read permission.

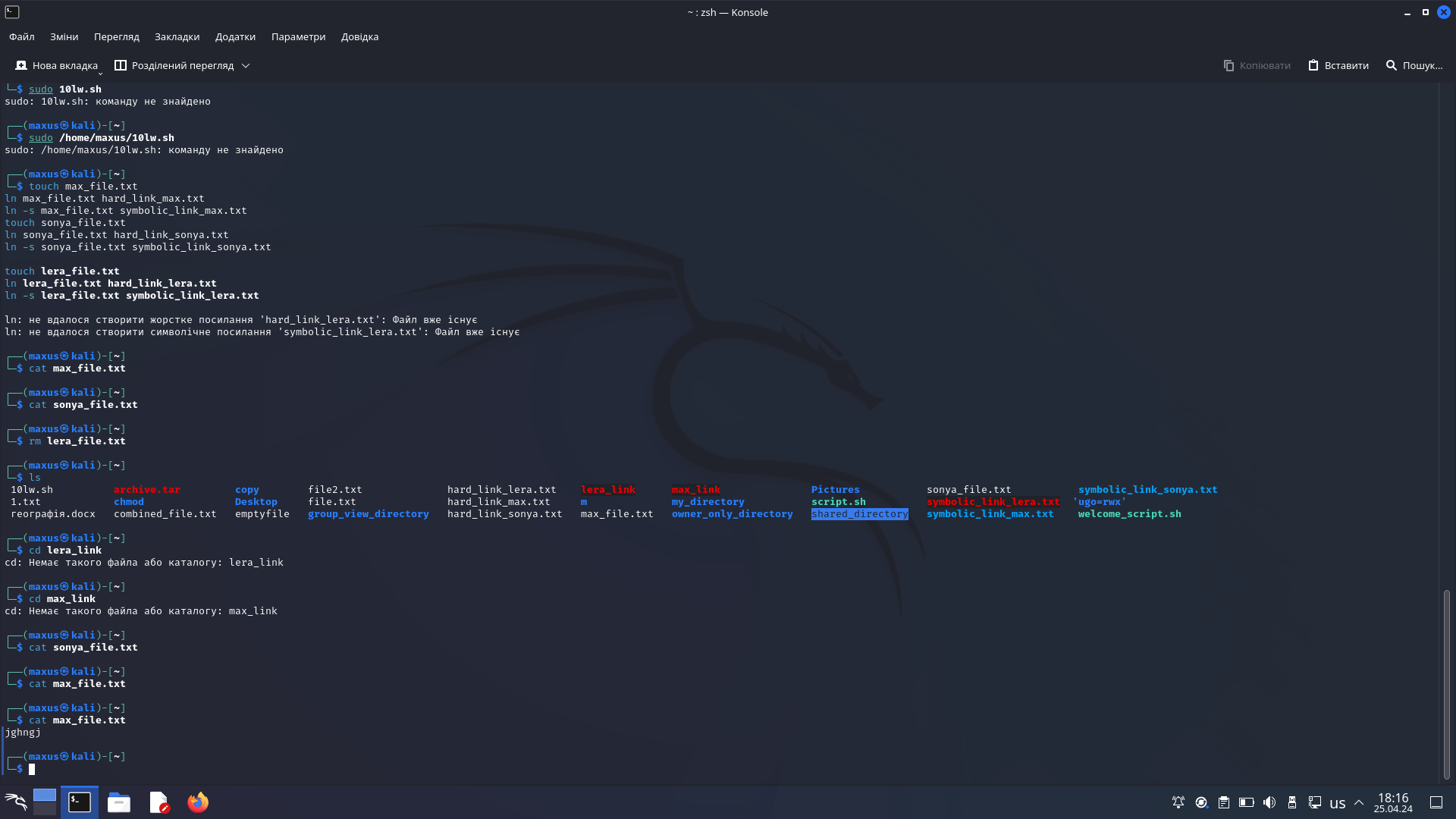
* \*\*створіть каталог під назвою, де всі файли автоматично будуть належати Вашій групі користувачів і можуть бути видалені лише користувачем, який їх створив.



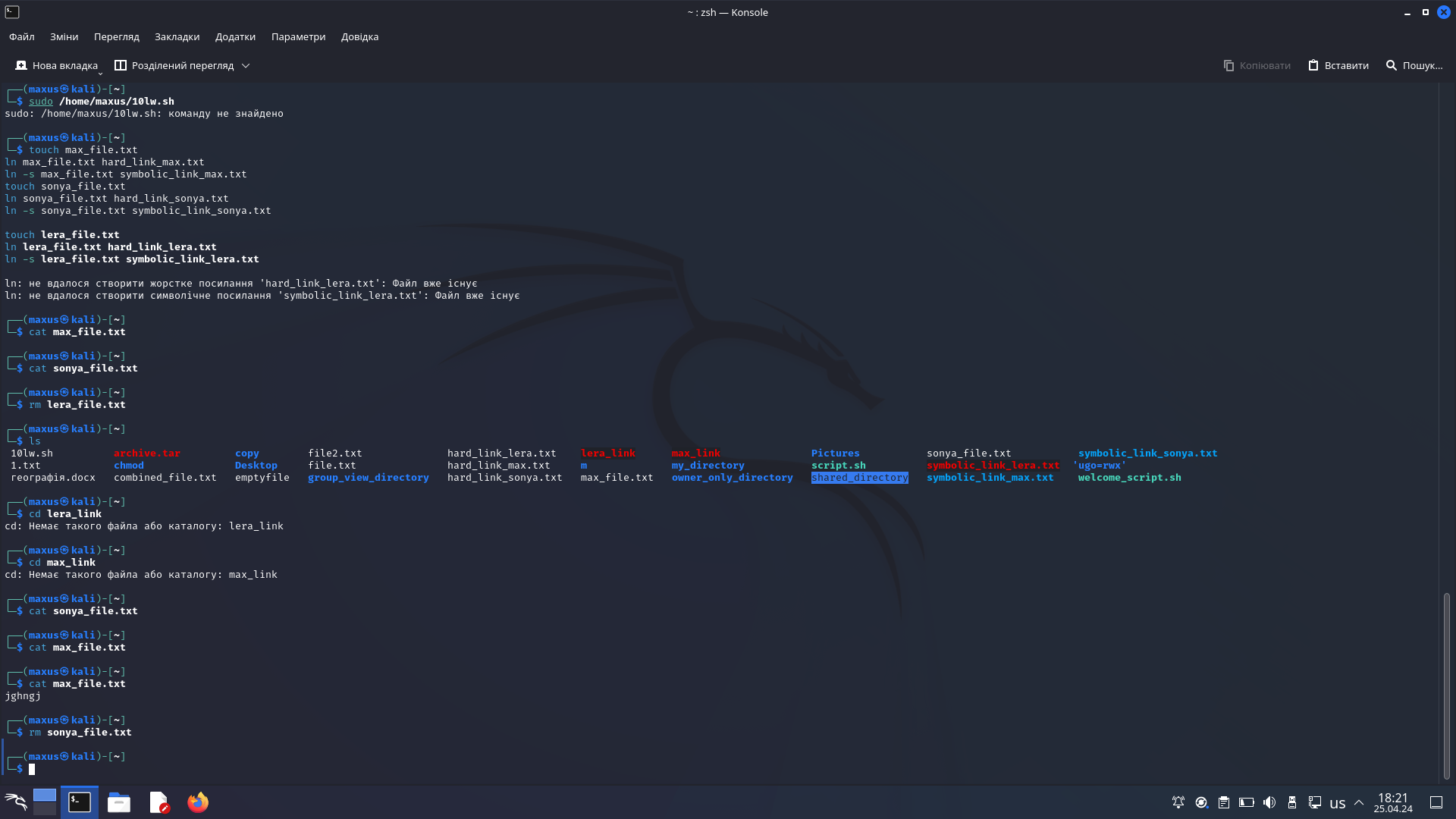
* \*\*під кожним користувачем створіть по одному новому файлу, та жорстке та символічне посилання на нього;



* \*\*спробуйте іншими користувачами переглянути ці файли;



* \*\*спробуйте іншими користувачами видалити ці файли, зробіть висновки.



Conclusion: If other users have the appropriate write permissions to the directory where the files are located, they can easily delete these files and see them.

PS: I guess I deleted them because I had access to each one, but I didn't?

**Контрольні запитання:**

***Готувала матеріал студентка Ємець Валерія***

1. Наведіть приклади зміни прав доступу символічним методом (Symbolic Method)?

1) To allow everyone (user, group, and others) to read a file, use the command chmod ugo+r filename

2) To prevent others (not the owner or group) from writing to or executing files within a directory, use the command chmod o-wx directory\_name

3) If you want to set the same permissions on another file as an existing file, use the `--reference` option

1. Наведіть приклади зміни прав доступу числовим методом (numeric method, octal method)?

The numeric representation consists of three digits: `###`. The first digit represents special permissions (setuid, setgid, or sticky bit). The remaining two digits represent permissions for the owner, group, and others.

Examples:

- `rwx` (full permission): Use `4 + 2 + 1 = 7`.

- `rw-` (read and write, no execute): Use `4 + 2 + 0 = 6`.

- `r--` (read only): Use `4 + 0 + 0 = 4`.

- `r-x` (read and execute, no write): Use `4 + 0 + 1 = 5`.

1. Яке призначення команди umask?

The umask command in Linux is used to set default permissions for files or directories the user creates.

The umask command specifies the permissions that the user does not want to be given out to the newly created file or directory. umask works by doing a Bitwise AND with the bitwise complement(where the bits are inverted, i.e. 1 becomes 0 and 0 becomes 1) of the umask. The bits which are set in the umask value, refer to the permissions, which are not assigned by default, as these values are subtracted from the maximum permission for files/directories.

1. Порівняйте жорсткі та символічні посилання?

Hard links cannot link directories and cross file system boundaries. Soft or symbolic links are just like hard links. It allows to associate multiple filenames with a single file. However, symbolic links allows to create links between directories and can cross file system boundaries. Also symbolic links are not updated, but hard links always refer to the source, even if moved or removed.

1. \*Чи можна виконати файл, для якого є права на виконання, але не встановлені права на читання (--x)? Поясніть.

Yes, it is indeed possible to execute a file that has execute privileges but not read privileges (--x) under certain conditions. When a file has execute (--x) permission but not read (r--) permission, it means that users are allowed to execute the file but not read its contents. This can happen, for example, when you have a script or executable program that you want users to be able to run but not view its source code.

1. \*Якщо ми змінюємо права доступу та дозволи в поточній сесії чи будуть вони збережені в наступній?

In the current session, any access rights and permissions you modify will not be automatically saved for the next session. Each session is independent, and changes made during one session do not persist to subsequent sessions.

1. \*Чи є якийсь шаблон, яким система користується щодо прав та доступів при створенні нових файлів. Як можна змінити права дозволу за замовчуванням?

When creating new files, the system uses a set of default permissions based on the umask value. The umask determines which permissions are not set when a new file is created. By default, it restricts certain permissions to ensure security and privacy.

You can change the default permissions for newly created files using umask, setting group ownership, using POSIX ACLs.

1. \*Яким чином можна створити жорстке посилання? В яких ситуаціях їх доцільно використовувати?

To create a hard links on a Linux or Unix-like system:

1) Create hard link between sfile1file and link1file, run: ln sfile1file link1file

2) To make symbolic links instead of hard links, use: ln -s source link

3) To verify soft or hard links on Linux, run: ls -l source link

Hard links use in shared libraries, version control and backups.

1. \*Яким чином можна створити символічне посилання? В яких ситуаціях їх доцільно використовувати?

In Linux, you can create a symbolic link using the ln command with the -s option.

Symbolic links use in shared libraries, version control and binary files, relocating documents and backup strategies.

1. \*\*Уявіть, що програмі потрібно створити одноразовий тимчасовий файл, який більше ніколи не знадобиться після закриття програми. Який правильний каталог для створення цього файлу?

It’s best to use a temporary directory. In Linux/Unix-like Systems you can create using the mktemp command with the -d option.

1. \*\*Є файл оригінал та для нього створено два посилання - символічне та жорстке. Що відбудеться з іншими файлами, якщо видалити:

* файл оригінал;
* символічне посилання;
* жорстке посилання.

1) If you delete the original file, it will be permanently removed from the filesystem. Any data associated with it will be lost. Other links (symbolic or hard) pointing to this original file will become dangling links, meaning they will reference a non-existent file.

2) Deleting a symbolic link does not affect the original file or any other links. The original file remains intact, and the symbolic link itself is removed. Symbolic links are essentially pointers to the original file, so their deletion doesn’t impact the file’s data.

3) When you delete a hard link, the behavior depends on the remaining links:

* If there are other hard links to the same file, the data remains intact.
* When the last hard link is deleted (i.e., the link count reaches 0), the file’s data is truly removed from the filesystem.

**Висновок:** we gained practical skills in working with the Bash command shell, learned the basic steps for changing file owners, file access rights, and special directories and files in Linux.