“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.

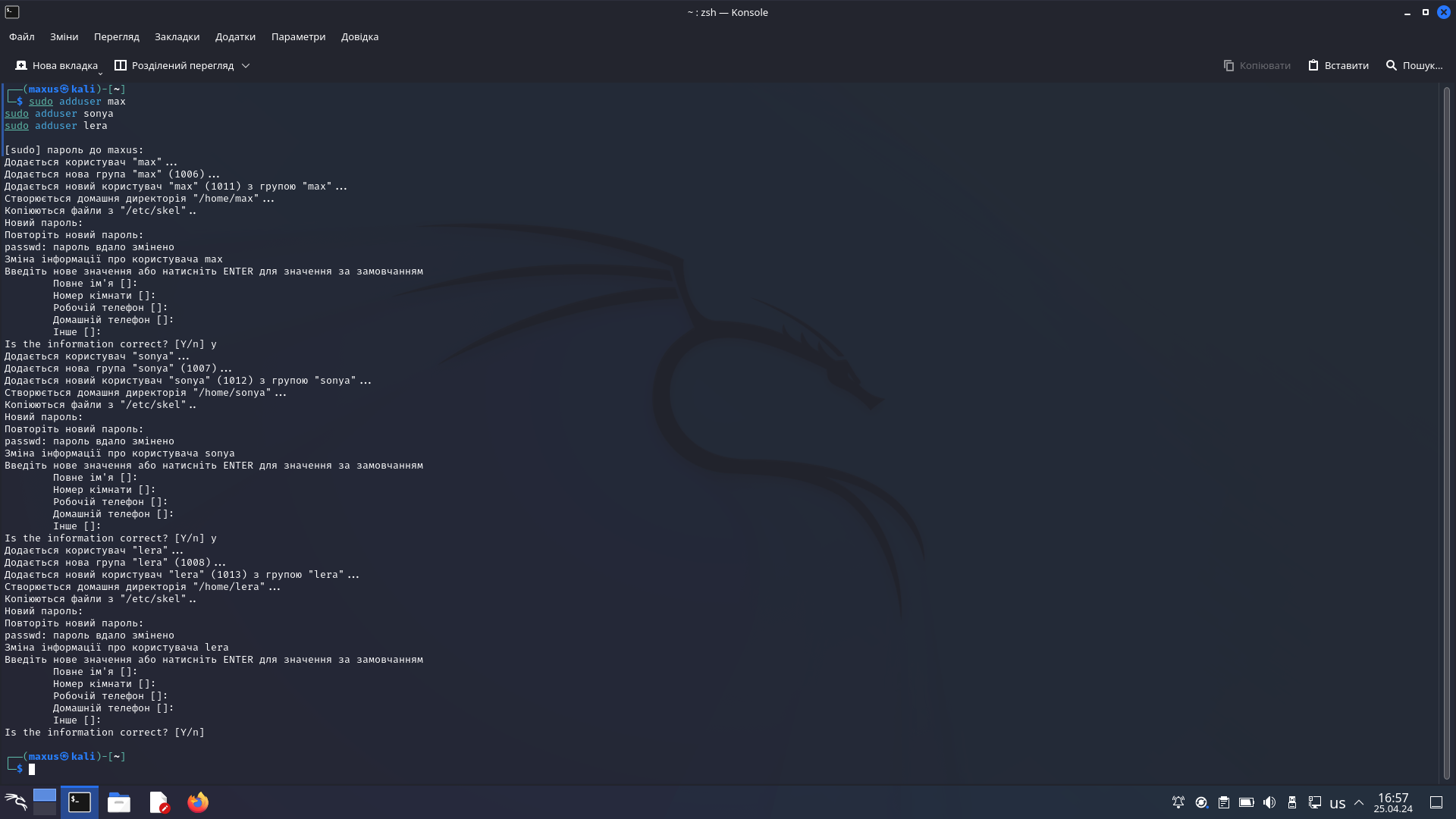
**Хід роботи:**

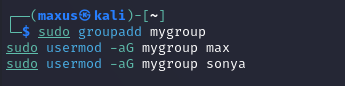
***Готувала матеріал студентка Лапчик Софія***

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:

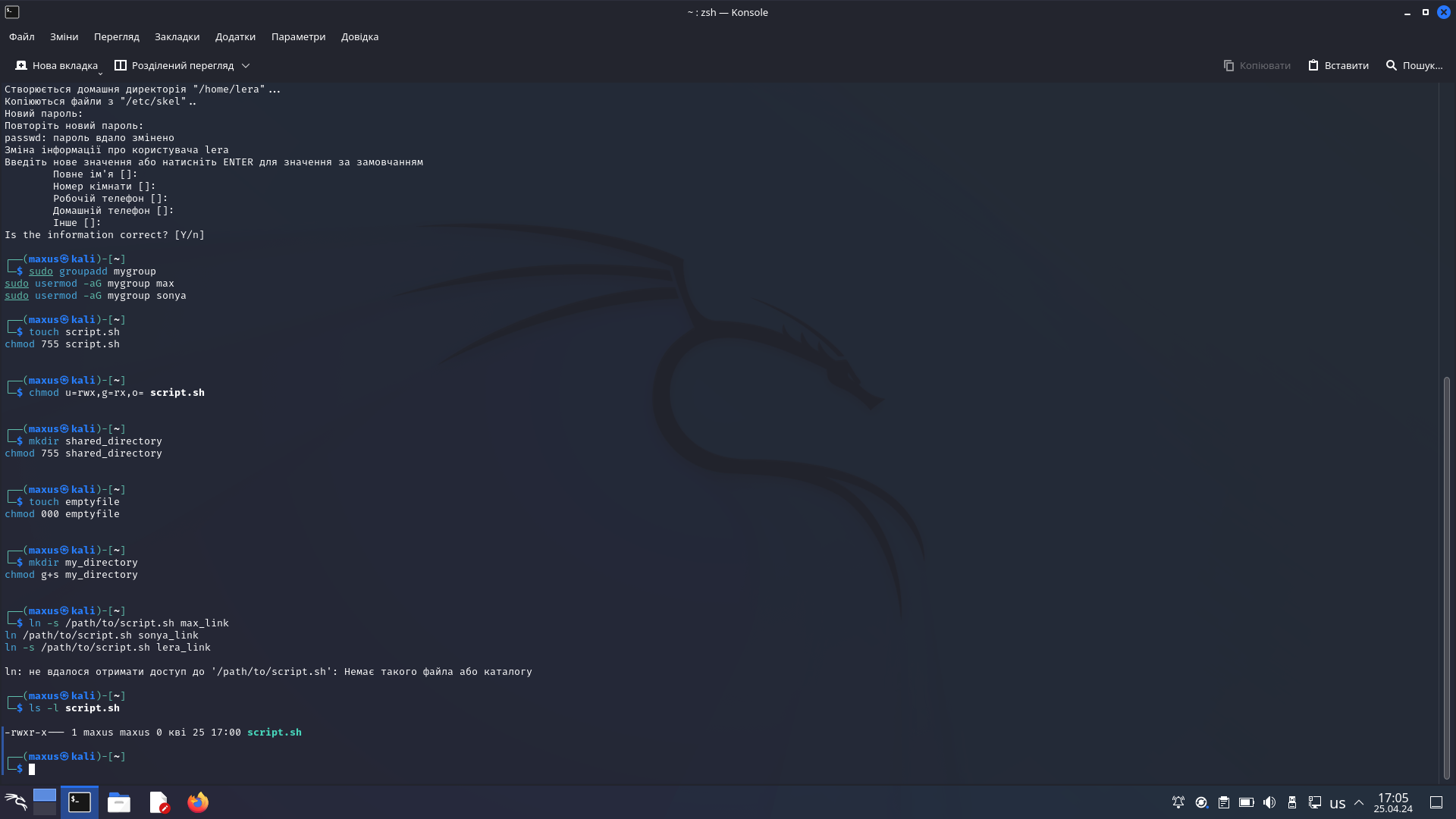
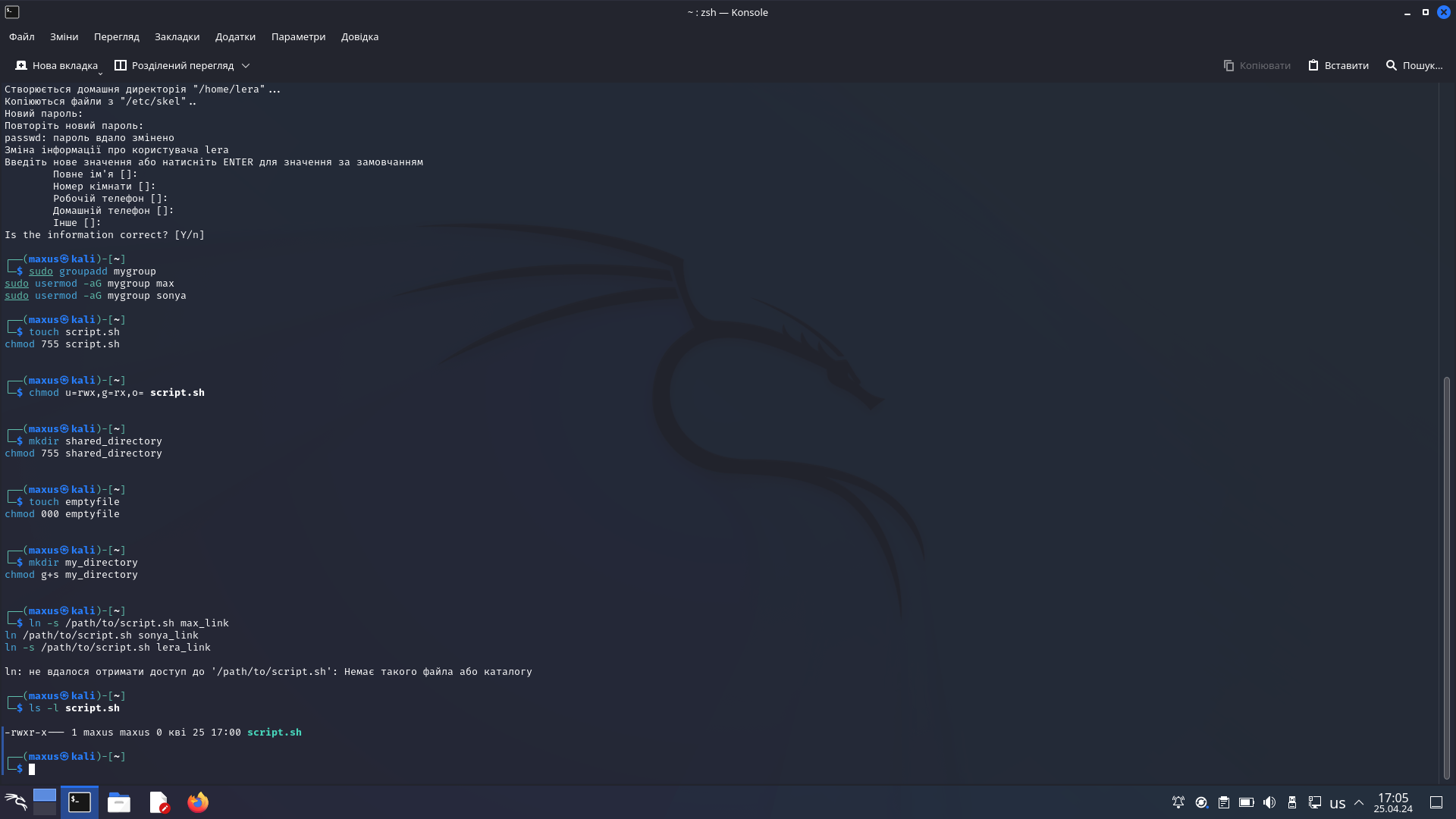
|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| chmod | command to change permissions |
| -l | file ownership can be confirmed using the long listing option of the ls command. |
| -a | hidden files, which begin with the dot . character are listed using this option |
| -d | 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) |

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

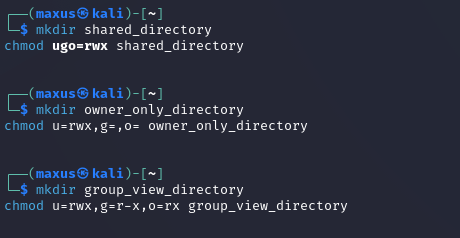
* створіть трьох нових користувачів;
* 
* створіть нову групу користувачів, туди додайте двох, з трьох створених користувачів;



* 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;

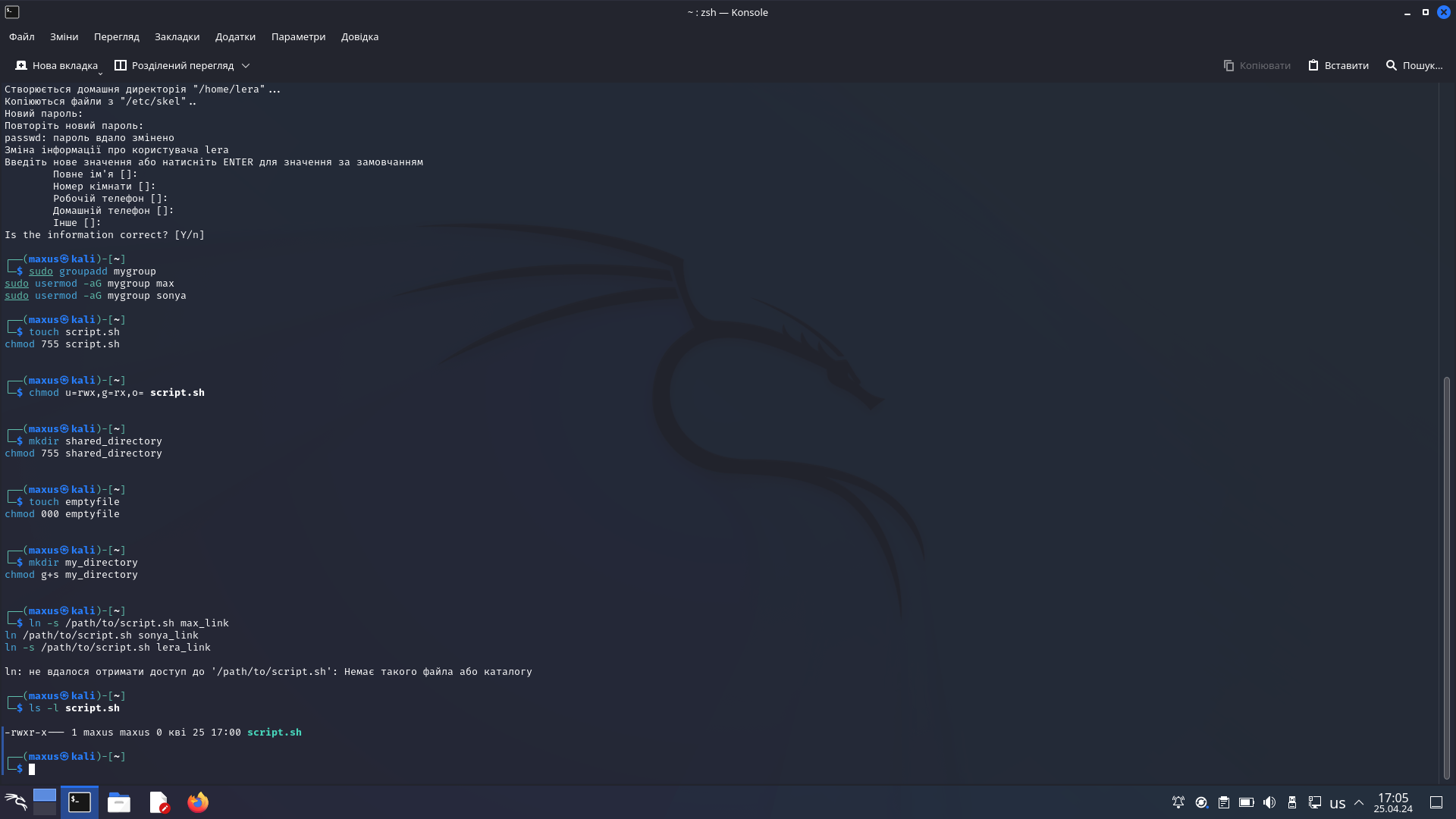


* \*подібні дії виконайте для директорій - створіть директорію, яка буде доступна для всіх трьох користувачів, створіть директорію, яку буде доступна тільки для власника, створіть директорію, яку користувачі групи власника зможуть переглядати, але не редагувати;



We created 3 derivatives with the following requirements

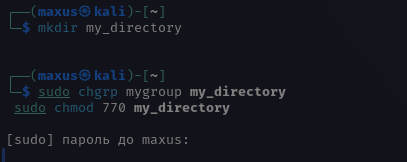
* \*створіть порожній файл під назвою emptyfile за допомогою команди touch emptyfile. Тепер “обнуліть” дозволи для файлу з chmod 000 emptyfile. Що станеться, якщо змінити дозволи для emptyfile, передавши лише одне значення для chmod у числовому режимі, наприклад, chmod 4 emptyfile? Що буде, якщо ми використаємо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



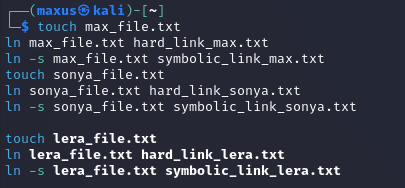
chmod 4 emptyfile, this will set read-only permission for the file owner. That is, the owner of the file will be able to read the contents of the file, but will not be able to edit or execute.

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.

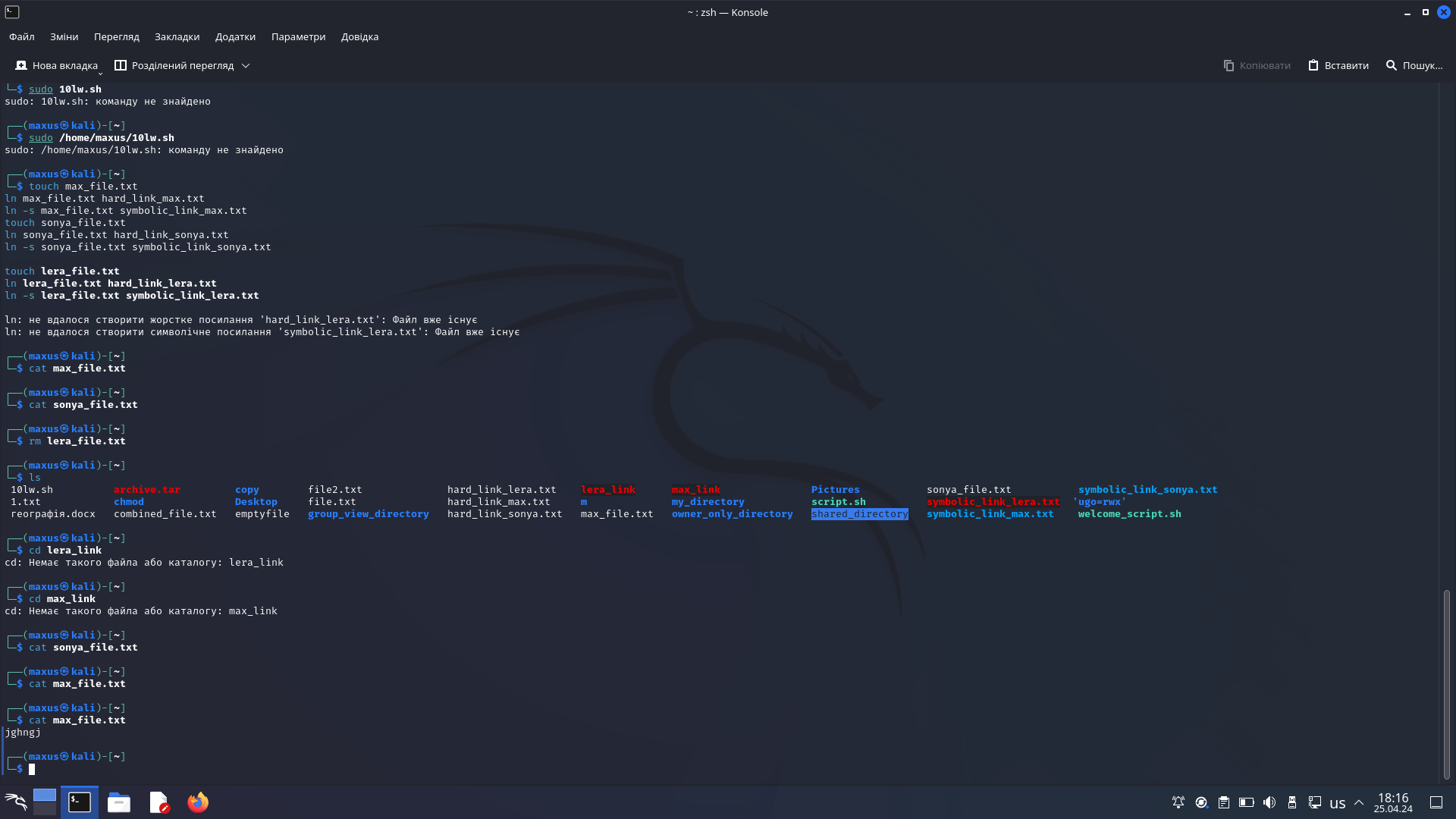
* \*\*створіть каталог під назвою, де всі файли автоматично будуть належати Вашій групі користувачів і можуть бути видалені лише користувачем, який їх створив.



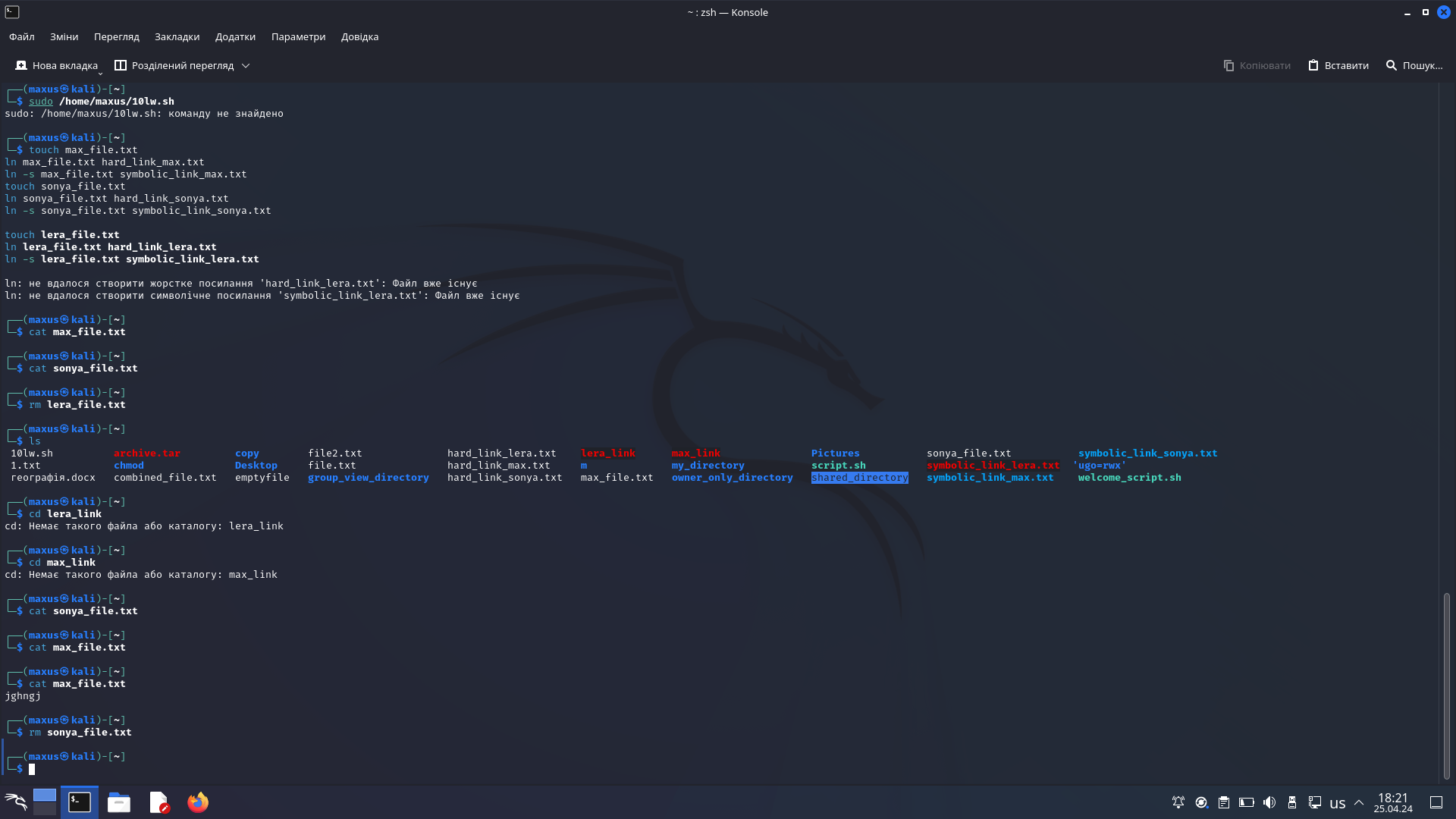
* \*\*під кожним користувачем створіть по одному новому файлу, та жорстке та символічне посилання на нього;



* \*\*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:**

***Готував матеріал студент***

1. Наведіть приклади зміни прав доступу символічним методом (Symbolic Method)?
2. Наведіть приклади зміни прав доступу числовим методом (numeric method, octal method)?
3. Яке призначення команди umask?
4. Порівняйте жорсткі та символічні посилання?
5. \*Чи можна виконати файл, для якого є права на виконання, але не встановлені права на читання (--x)? Поясніть.
6. \*Якщо ми змінюємо права доступу та дозволи в поточній сесії чи будуть вони збережені в наступній?.
7. \*Чи є якийсь шаблон, яким система користується щодо прав та доступів при створенні нових файлів. Як можна змінити права дозволу за замовчуванням?
8. \*Яким чином можна створити жорстке посилання? В яких ситуаціях їх доцільно використовувати?
9. \*Яким чином можна створити символічне посилання? В яких ситуаціях їх доцільно використовувати?
10. \*\*Уявіть, що програмі потрібно створити одноразовий тимчасовий файл, який більше ніколи не знадобиться після закриття програми. Який правильний каталог для створення цього файлу?
11. \*\*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.

**Висновок:**