“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»**

Виконала студентка

групи РПЗ-13б

Дімітрова С.П.

Перевірив викладач

Сушанова В.С.

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).

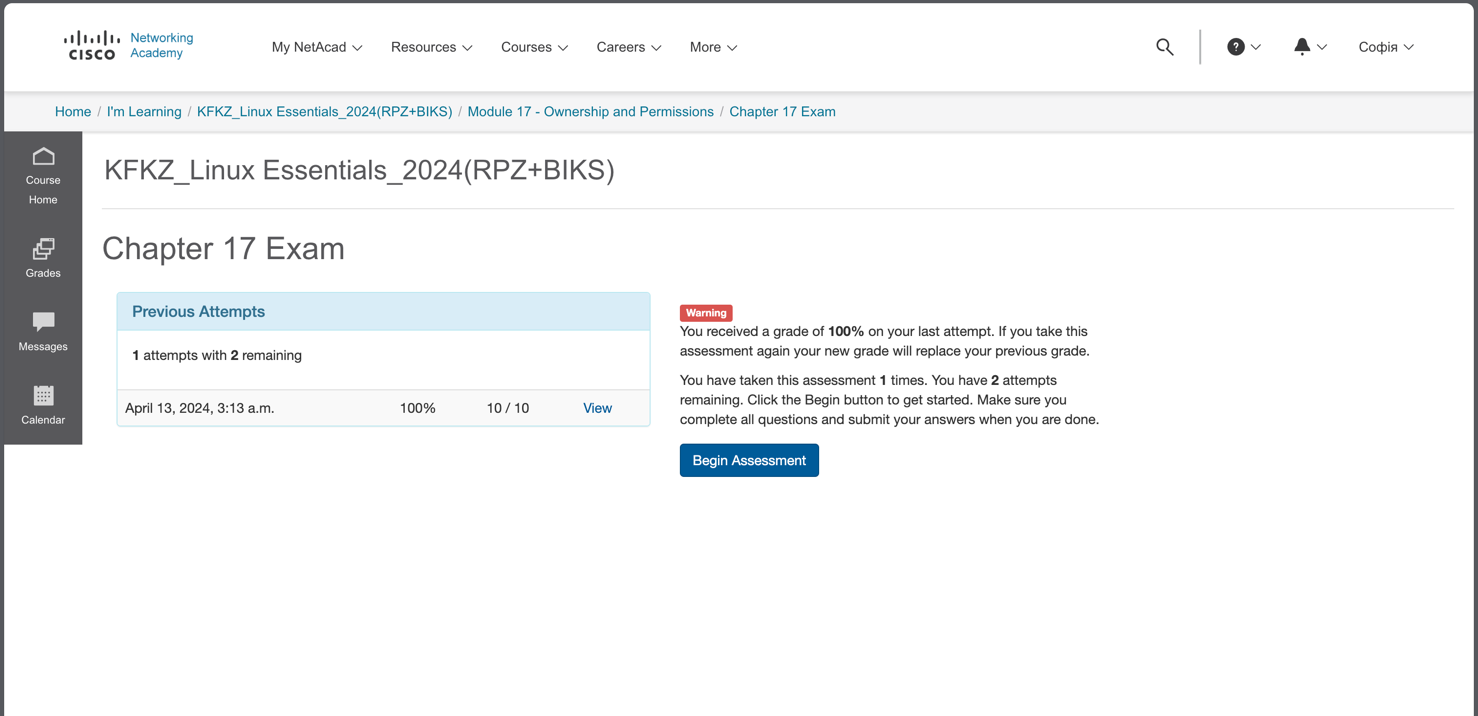
4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки:**

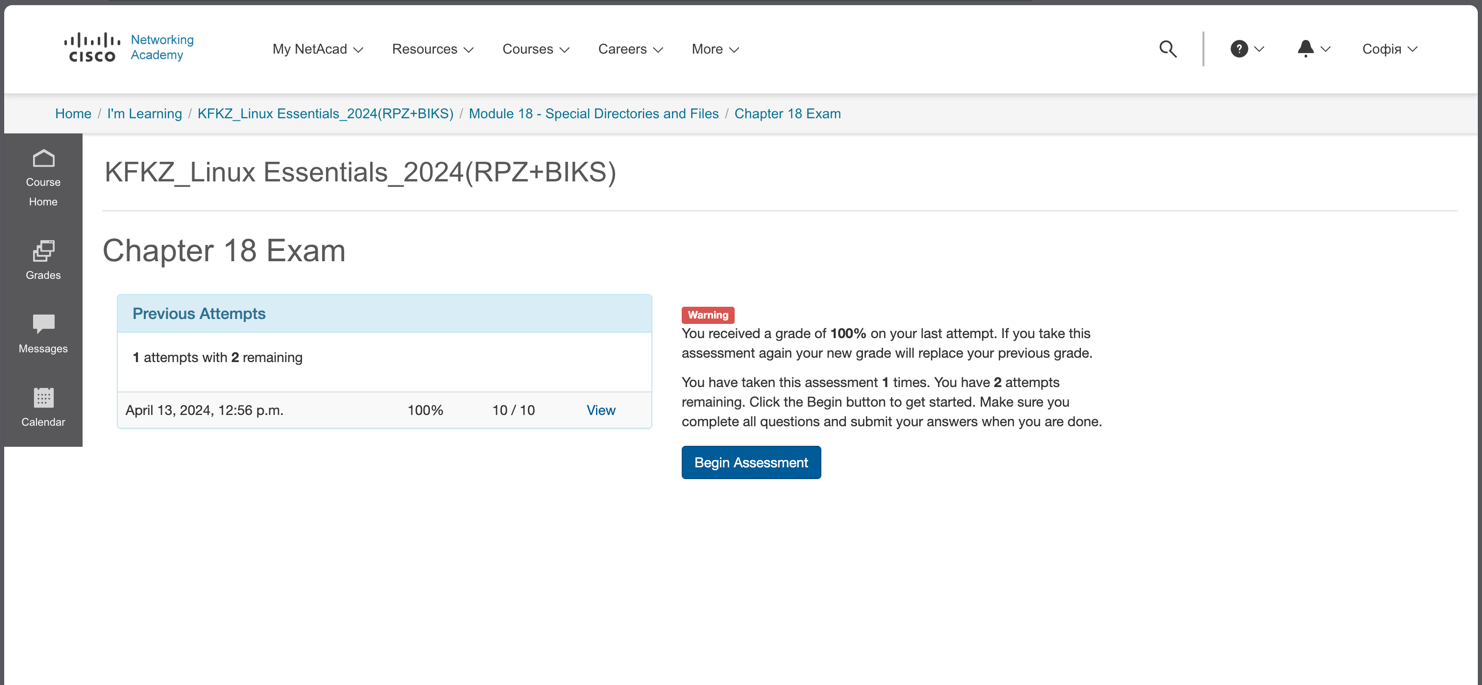
* \*Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань призначення команд та їх параметрів.

|  |  |
| --- | --- |
| **Термін англійською** | **Термін українською** |
| File ownership | Право власності на файл.  Вказує на користувача та групу, які мають право доступу до файлу. |
| Supplemental groups | Додаткові групи |
| Setuid (Set User ID) | Установка ідентифікатора користувача.  Якщо setuid встановлено, то при запуску файлу, процес отримує ідентифікатор користувача файлу, а не ідентифікатор користувача, яким запущений процес. |
| Setgid (Set Group ID) | Установка ідентифікатора групи.  Якщо setgid встановлено для файлу, то при запуску процесу, процес отримує ідентифікатор групи файлу, а не ідентифікатор групи, до якої належить користувач. |
| Sticky bit | Якщо sticky bit встановлено для директорії, тоді тільки власник файлу може його видалити або перейменувати. Цей біт використовується, наприклад, для захисту спільних директорій, де кілька користувачів можуть записувати файли. |
| Octal method | Вісімковий метод. Метод представлення прав доступу до файлу за допомогою вісімкових чисел. . Кожні три біти відповідають окремій категорії користувачів (власник, група та інші користувачі), а значення кожного біта відображає наявність або відсутність певних прав доступу (читання, запису та виконання). |
| Symbolic method | Символьний метод. Метод представлення прав доступу до файлу за допомогою символів. Кожна категорія користувачів (власник, група та інші користувачі) представлена символами (r - читання, w - запис, x - виконання), які вказують на наявність або відсутність певних прав доступу. |

* Вивчіть матеріали онлайн-курсу академії Cisco “NDG Linux Essentials”:
* Chapter 17 - Ownership and Permissions
* Chapter 18 - Special Directories and Files
* Пройдіть тестування у курсі NDG Linux Essentials за такими темами:
* Chapter 17 Exam

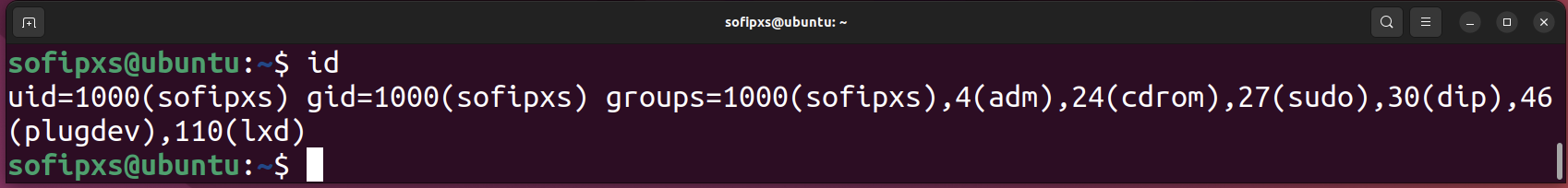


* Chapter 18 Exam



* На базі розглянутого матеріалу дайте відповіді на наступні питання:
  1. Яке призначення команди id?

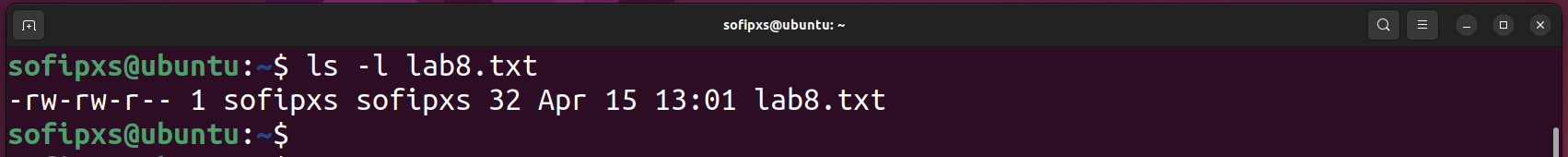
The *id* command is used to display information about the current user and their groups. This includes the user ID (UID), username, group ID (GID), primary group name, and any supplemental groups the user belongs to. It allows to check which user account is currently in use and which groups are available for use.



* 1. Як переглянути які права доступу має власник файлу?

To view the access rights of the file owner, use the *ls* command with the *-l* option, which displays detailed information about files and directories in a "long listing" format.

The first character - or *d* indicates the type of file. - indicates a regular file, and d indicates a directory.The second three characters indicate the access rights of the file owner. The first character indicates whether the owner has read permission (r). The second character indicates whether the owner has write permission (w). The third character indicates whether the owner has the right to execute (x).



Example: ls -l file.txt

The output will look like the following: -rw-rw-r-- 1 owner group 32 Apr 15 13:01 file.txt

In the example, -rw-rw-r-- shows the access rights of the owner of the file lab8.txt. Here is the decoding of these rights:

* + The first character - indicates the type of file, in this case it is a regular file (not a symbolic link, not a directory, etc.).
  + rw- means that the owner of the file (sofipxs) has read (r) and write (w) permissions, but no execute (-) permissions.
  + rw- means that the group to which the file belongs (also sofipxs) has read (r) and write (w) permissions, but no execute (-) permissions.
  + r-- means that other users who do not belong to the group of the file owner have read (r) permission only, but no write (-) or execute (-) permission.
  1. \*Як змінити власника групи?

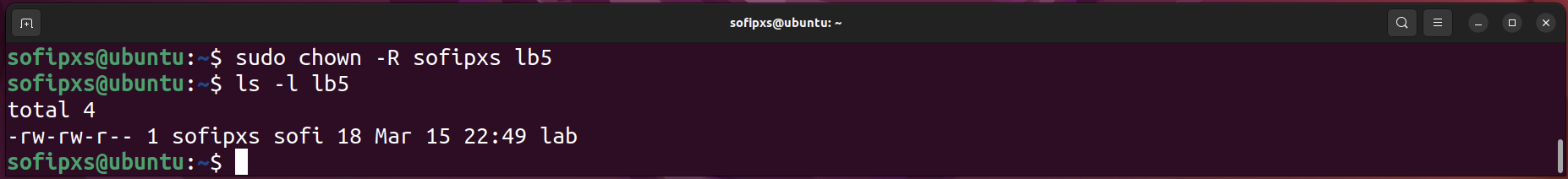
To change the group owner of a file, we can use the *chown* command (change owner). The chown command can only be executed by the root user and it can change both the user and group that owns a file.

If the root user wanted to change the user ownership of the filetest1 file to the user jane, then the following command could be executed:

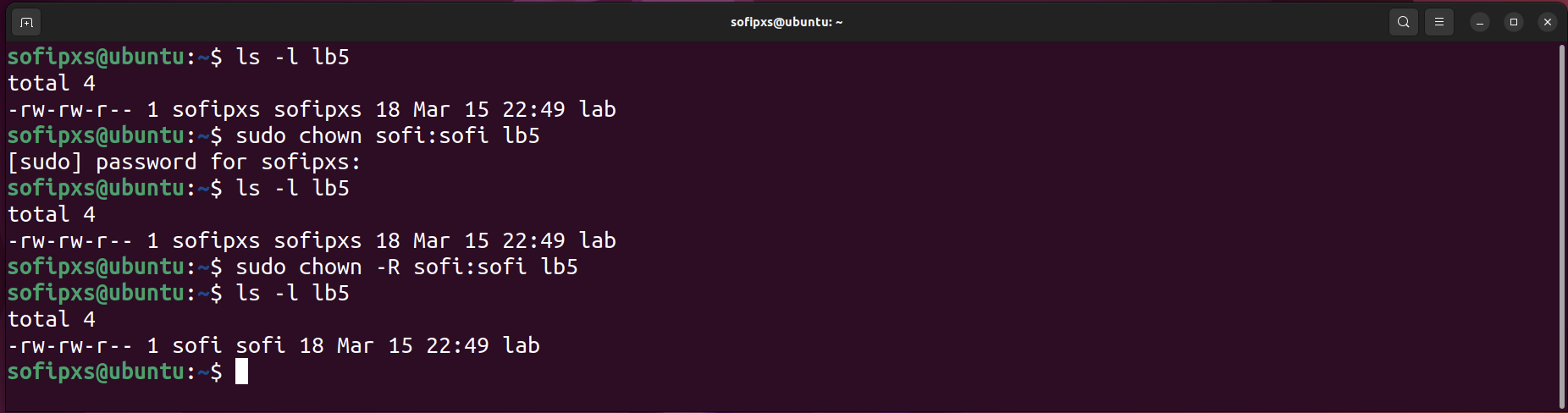
*root@localhost:~# chown jane /tmp/filetest1*

*root@localhost:~# ls -l /tmp/filetest1*

*-rw-rw-r-- 1 jane sysadmin 0 Dec 19 18:44 /tmp/filetest1*



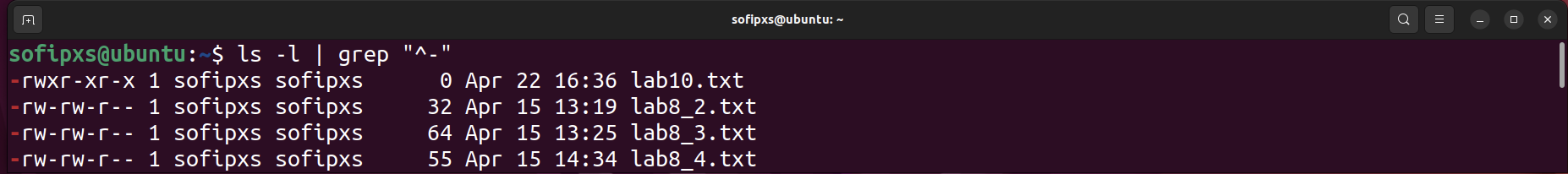
If we want to change the group owner of the file.txt file to the user new\_owner and the group new\_group, the command will look like this:   
*sudo chown -R new\_owner:new\_group file.txt*



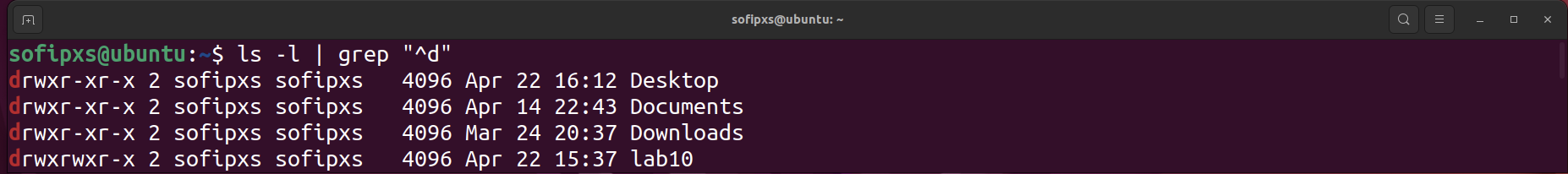
* 1. \*Як можна переглянути у терміналі який тип поточного файлу? Наведіть приклади для різних типів файлів.

The output of the ls -l command displays ten characters at the beginning of each line. These characters indicate the type of file and the permissions of the file. The first character of each line indicates the type of file. Possible values for file types:

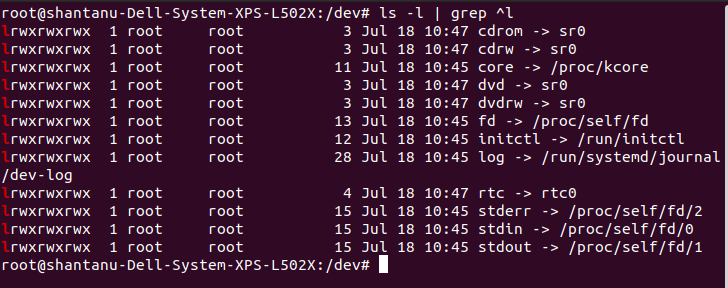
- : A regular file, which may be empty, or contain text or binary data.



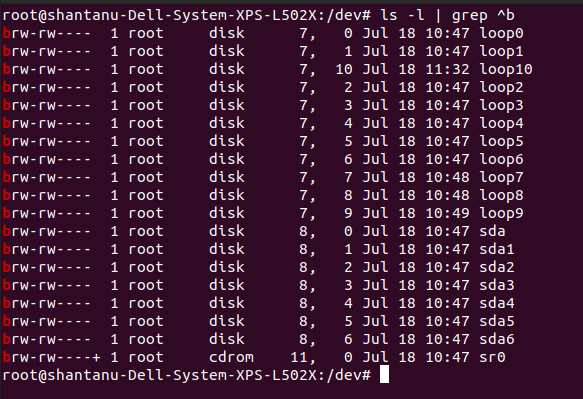
d: A directory file, which contains the names of other files and links to them.



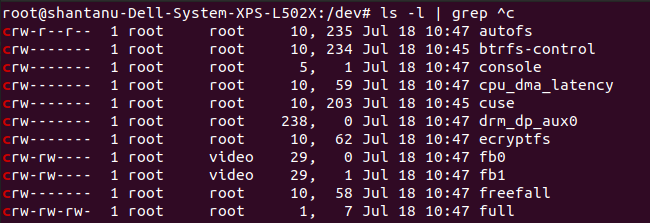
l: A symbolic link is a file name that refers (points) to another file.



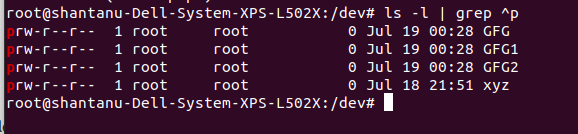
b: A block file is one that relates to a block hardware device where data is read in blocks of data.



c: A character file is one that relates to a character hardware device where data is read one byte at a time.



p: A pipe file works similar to the pipe symbol, allowing for the output of one process to communicate to another process through the pipe file, where the output of the one process is used as input for the other process.



s: A socket file allows two processes to communicate, where both processes are allowed to either send or receive data.

Lightbox

* 1. \*\*Для чого використовуються дозволи Setuid та Setgid?

Setuid (Set User ID) and Setgid (Set Group ID) permissions are used to set special access rights for executable files in Unix-like systems.

* *Setuid:* If an executable file has the Setuid bit set and is owned by a user, it runs with the permissions of the file owner, rather than those of the user who executes it. This is useful for programs that require special privileges, such as programs that work with system resources like network configuration or password changes.
* *Setgid:* If an executable file has the Setgid bit set and belongs to a group, it runs with the permissions of the file group, rather than those of the user who executes it. This can be useful for files that require shared access to certain resources, for example, shared directories where users in a group can modify files owned by that group.
  1. \*\*Для чого в системі потрібен так званий “липкий біт” (Sticky Bit). Наведіть приклади коли цей дозвіл доцільно використовувати.

The Sticky Bit is used in the system to set special access rights for directories. The primary purpose of the Sticky Bit is to restrict the ability to delete or rename files by other users in a directory that has the Sticky Bit set.

A few examples of situations where using a sticky bit might be appropriate:

* The `/tmp` directory: On many systems, the `/tmp` directory is used as a temporary repository for files created by different users. Setting the sticky bit to `/tmp` allows you to ensure that each user can only modify or delete their own files, not files created by other users.
* Public directories: If you have a directory in which files are frequently uploaded or created by many users, setting a sticky bit can prevent other users from accidentally deleting or moving files. This can be useful, for example, in web servers where files available for download are stored in a shared directory.
* Data protection: If you have a directory with sensitive data that should only be accessible by certain users, setting a sticky bit can ensure that files in that directory cannot be deleted or moved by other users.

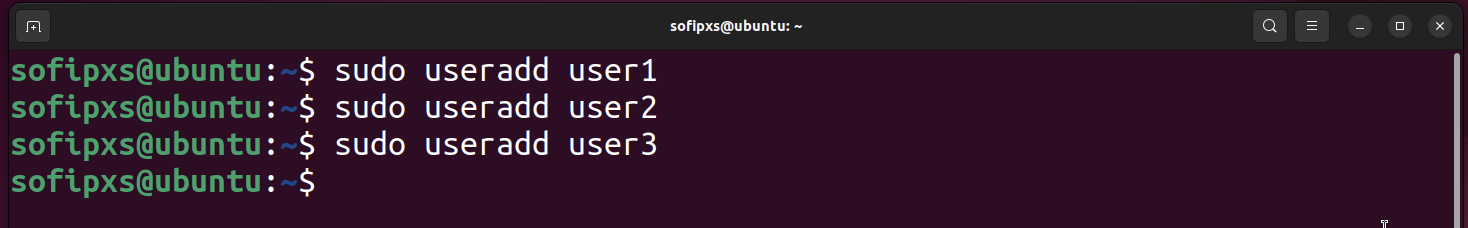
**Хід роботи:**

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 запустіть 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 для опису цих команд.

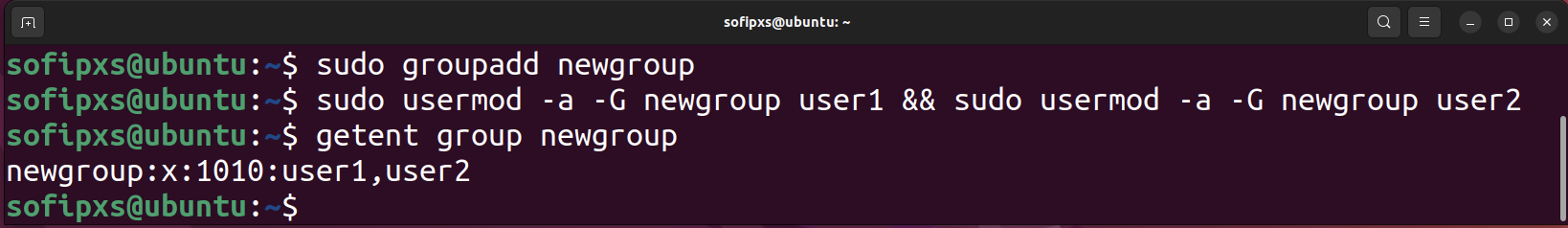
|  |  |
| --- | --- |
| **Назва команди** | **Її призначення та функціональність** |
| ***NDG Linux Essentials - Lab 17: Ownership and Permissions*** | |
| cd /tmp | Move to the /tmp directory |
| mkdir priv-dir pub-dir | Create two directories called priv-dir and pub-dir |
| touch priv-dir/priv-file  touch pub-dir/pub-file | Create two files, one file called priv-file in the priv-dir directory and another file called pub-file in the pub-dir directory |
| ls -l priv-dir  ls -l pub-dir | View the contents of the new directories. |
| ls -ld priv-dir/ | View permissions for the priv-dir directory |
| chmod o-rx priv-dir/ | Remove the others' permissions for read and execute |
| chmod a+x file | Gives everyone execute permission |
| chmod g-w file | Removes write permission for group owners |
| chmod go+r file | Adds read permission for group owner and others |
| chmod o=rwx | Sets others permissions to read, write and execute |
| chmod o+w pub-dir/ | Adds write permission for others to the pub-dir directory. |
| chmod g-rw,o-r priv-dir/priv-file | Removes read and write permissions for the group and read permission for others from the priv-file inside the priv-dir directory. |
| chmod a=rw pub-dir/pub-file | Sets the permissions for the owner, group, and others to read and write for the pub-file inside the pub-dir directory. |
| echo "date" > test.sh | Create a test.sh file in the /tmp containing the content "date" |
| ./test.sh | Attempt to execute the test.sh file; it should fail. |
| chmod u+x test.sh | Adds execute permission for the owner (user) to the test.sh file. |
| stat test.sh | Verify the octal value for the permissions (access) to test.sh |
| chmod 775 test.sh | Sets the permissions for the owner, group, and others to read, write, and execute for the test.sh file. |
| chown root:root pub-dir | Change the user and group owner of pub-dir to the root user and the root group |
| chown bin pub-dir/pub-file | Change the user owner of the pub-file to the bin user |
| ls -ld priv-dir | View the details of the priv-dir and its contents |
| ls -ld priv-dir  ls -l priv-dir/priv-filechgrp -R users priv-dir | To change the group ownership of all of the files of a directory structure, use the recursive -R option to the chgrp command. Change the group owner of the priv-dir and priv-file to the users group recursively with the chgrp command and view the updated files. |
| ***NDG Linux Essentials - Lab 18: Special Directories and Files*** | |
| ls -ld /tmp | List the details of the /tmp directorie.Using the -d option for the ls command lists directory information; combined with the -l option it shows ownership and permissions for the directory files. |
| ls -ld /var/tmp | List the details of the  /var/tmp directorie. |
| ls -l /etc/shadow | View the permissions on the /etc/shadow file. Specifically, the /etc/shadow file contains the encrypted passwords of all local user accounts and information about password aging (how long a password is valid). |
| ls -l /usr/bin/passwd | View the permissions of the /usr/bin/passwd file. Thus, the passwd command is able to update the /etc/shadow file, as it executes as the root user (recall that the root user can edit any file, regardless of the permissions on the file). |
| ls -l /usr/bin/wall | View the permissions on the /usr/bin/wall command |
| cd | Change to your home directory |
| echo "data" > source | Create a file named source containing the text "data" by using redirection |
| ls -li source | Using the -i option with the ls command prints the index number of the file. View the details and inode information of the source file |
| ln source hardlink | Create a hard link. |
| ls -li source hardlink | View the details and inode information of the source and new hard link file. |
| ln hardlink hardlinktwo  ls -li hardlink hardlinktwo source | Create another hard link to the source file. View the details and inode information of the source and new hard link files. |
| rm hardlinktwo | Remove the last link that was created. |
| ln -s source softlink  ls -li source softlink | The -s option for the ln command creates a symbolic link instead of a hard link. Create a symbolic link to the source file and view the details of both files. |
| ln -s /proc crossdir  ls -l crossdir | Create a symbolic link to the /proc directory and display the link. |

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;

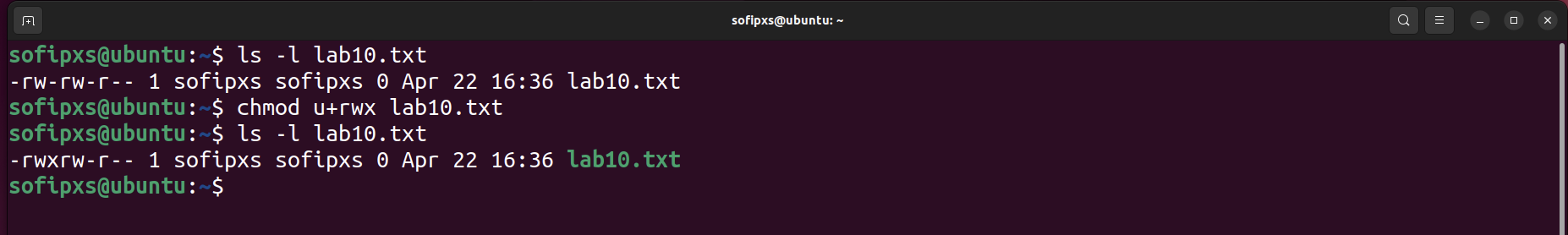


* створіть новий файл, який буде доступний на зчитування, редагування та виконання власником файлу, наприклад найпростіший скриптовий сценарій;
* для користувачів групи власника надайте дозволи на перегляд та виконання (без дозволу на редагування) цього файлу;
* для інших користувачів заборонити доступ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

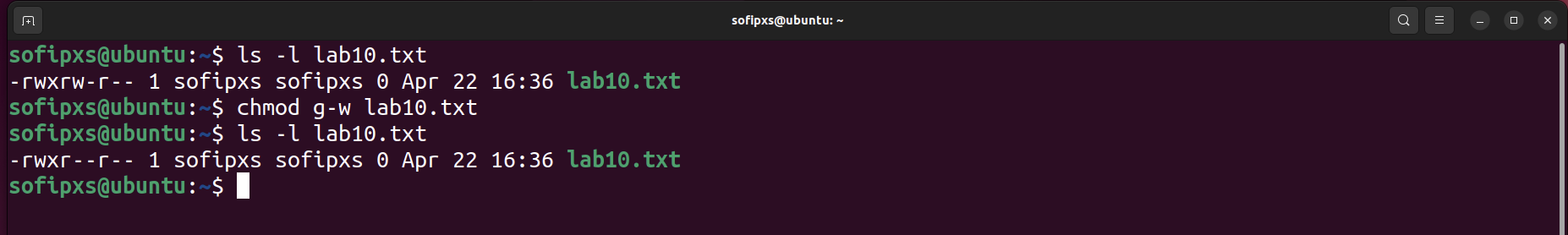
**Відповіді на контрольні запитання:**

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

* Change access rights for the file owner: *chmod u+rwx filename.* This command grants the owner (u) read (r), write (w), and execute (x) permissions for lab10.txt.



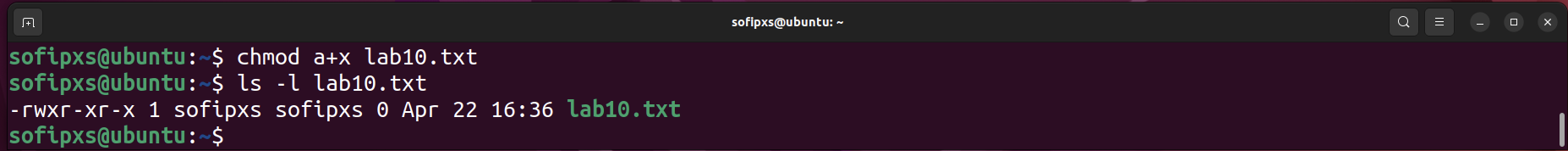
* Change access rights for a group: *chmod g-w file.txt*. This command   
  removes (-) write permission (w) for group (g) for lab10.txt.



* Combined change of access rights: *chmod u=rw,g+rx,o-w file.txt.* This command sets (=) read and write (rw) permissions for the owner (u), adds (+) read and execute (rx) permissions for the group (g), and removes (-) write permissions (w) for other users (o) for the lab10.txt.



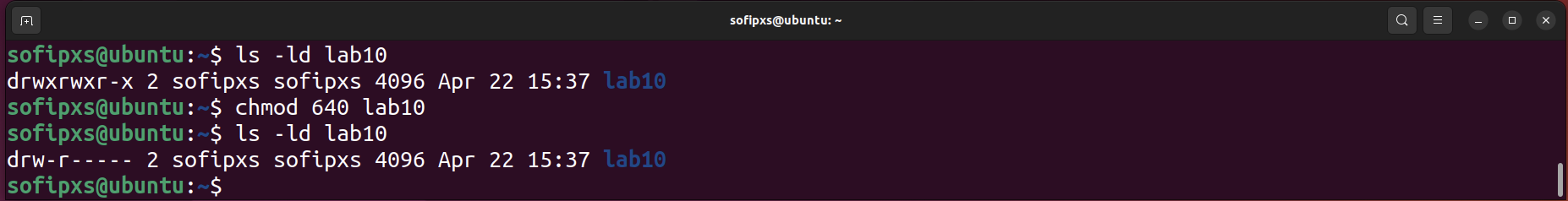
* + Gives everyone execute permission: *chmod a+x file.txt*



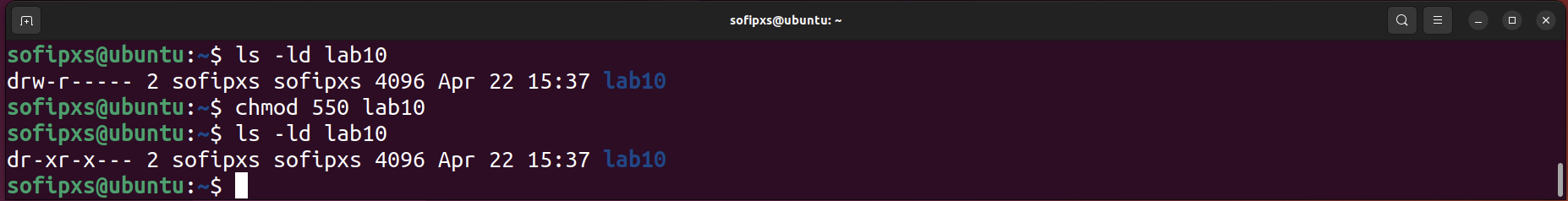
1. Наведіть приклади зміни прав доступу числовим методом (numeric method, octal method)?

In the numeric method, each of the access rights (read, write, execute) is represented by a three-digit number, where each digit corresponds to the access rights for the owner, group, and other users.

* Set access rights to "read and write" for the owner, "read" for the group and "no access" for the others: *chmod 640 file.txt*



* Set access rights to "read and execute" for owner and group, "read" for others: *chmod 550 file.txt*



* Set "read, write and execute" access rights for all: *chmod 777 file.txt*



1. Яке призначення команди umask?

The purpose of the *umask* command is to set the default file creation permissions mask. It determines which permissions are automatically subtracted from the maximum permissions when a new file or directory is created. It helps control the default permissions to enhance security and privacy.

1. Порівняйте жорсткі та символічні посилання?

* A *hard link* is a direct link to a file by its inode index number. A hard link is created to an existing file on the same file system and looks like an additional name for that file. All hard links point to the same file, and they cannot point to directories or files on other file systems.
* A *symbolic link* is a reference to a file or directory that contains the path to the target file. A symbolic link allows you to create links to files or directories that may be located elsewhere in the file system or even on other file systems.

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

No. It is not possible to execute a file that has execute privileges but not read privileges (--x). To execute a file, you must have at least read permission, because the operating system must read the contents of the file to execute its contents. If you do not have read permission (--x), the operating system cannot read the file and execute it.

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

No. Changes to access rights and permissions made in the current session are not saved for subsequent sessions by default. When the session is closed or the system is restarted, the changed access rights and permissions return to the values set in the file system configuration or default settings. To make changes to access rights permanent and save them between sessions, you need to make appropriate changes to the configuration files or use other mechanisms, such as scripts or automated procedures, to apply the changes at each startup.

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

Yes, there is a template or default set of permissions and access control settings used when creating new files in Linux. The default permissions are determined by the system's umask value, which is typically set in the shell configuration files (~/.bashrc, /etc/profile, etc.). You can change the default permissions by modifying the umask value in the appropriate configuration file.

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

To create hard links, the ln command is used with two arguments. The first argument is an existing file name to link to, called a target, and the second argument is the new file name to link to the target: *ln target link\_name*

Hard links are advisable to use when you want multiple filenames to refer to the same physical data on disk. They are useful for creating backups, referencing important files from different locations, or when you want to have different filenames for the same content.

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

To create a symbolic link, you can use the ln command with the -s option followed   
by the target file or directory and the desired name for the symbolic link:   
*ln -s target link\_name*

Symbolic links are appropriate to use when you want a file or directory to be accessible from a different location or when you want to create shortcuts or references to files. Symbolic links can point to files or directories across different file systems.

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

The /tmp directory is commonly used for creating temporary files that are not needed after the program is closed. The /tmp directory is meant for temporary files that can be accessed by all users on the system. It is often cleared on system reboot or during regular cleanup processes.

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

* файл оригінал;

The file will be permanently deleted. Symbolic links will become broken, and hard links will still be able to access the file's content until they are deleted or the system is rebooted.

* символічне посилання;

Only the symbolic link itself will be deleted. The original file and other hard links will remain unaffected.

* жорстке посилання.

The hard link will be deleted, but the original file and other hard links will still retain the file's content until they are deleted or the system is rebooted.

**Висновки:**