“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної та програмної 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 КСМ-23А

Команда КГК:

Корольов Є.Ю.,

Горохов Д.С. та

Коваленко С.О.

Перевірила 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:**

***Готував матеріал студент Корольов Є. та Коваленко С.***

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

|  |  |
| --- | --- |
| Термін | Опис та Команди |
| Owner | Власник файлу чи каталогу.  Команда: chown user file.txt |
| Group | Група користувачів, які мають доступ до файлу.  Команда: chown :group file.txt |
| Other | Інші користувачі, які не є власниками чи членами групи. |
| Read (r) | Дозвіл на читання файлу чи каталогу.  Команда: chmod u+r file.txt |
| Write (w) | Дозвіл на зміну вмісту файлу.  Команда: chmod g+w file.txt |
| Execute (x) | Дозвіл на виконання файлу або доступ до каталогу.  Команда: chmod o+x script.sh |
| Permission string | Послідовність символів, що відображає права доступу (rwxr-xr--). |
| chown | Зміна власника файлу чи групи.  Приклад: chown user:group file.txt |
| chmod | Зміна прав доступу до файлу чи каталогу.  Приклад: chmod 755 file.txt |
| Special directories | Спеціальні каталоги, як /etc, /home, /var. |

1. Вивчіть матеріали онлайн-курсу “NDG Linux Essentials” (netacad.com):

* Chapter 17 - Ownership and Permissions
* Chapter 18 - Special Directories and Files

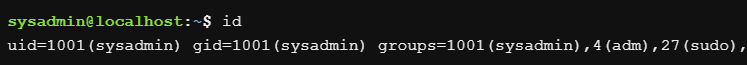
1. Пройдіть тестування у курсі NDG Linux Essentials за такими темами:

* Chapter 17 Exam
* Chapter 18 Exam

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

The id command can be useful for verifying which user account you are using and which groups you have available to use. By viewing the output of this command, you can see the user's identity information expressed both as a number and as a name.

The output of the id command displays the UID and user account name of the current user followed by the GID and group name of the primary group and the GIDs and group names of all group memberships.



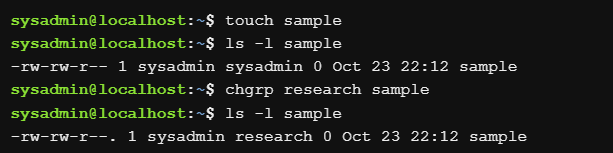
* 1. Як переглянути які права доступу має власник файлу?

The file ownership can be confirmed using the long listing -l option of the ls command.



* 1. \*Як змінити власника групи?

As the root user, the chgrp command can be used to change the group owner of any file to any group. As a user without administrative privileges, the chgrp command can only be used to change the group owner of a file to a group that the user is already a member of.



* 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



***-*** 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.

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

When the setuid permission is set on an executable binary file (a program) the binary file is run as the owner of the file, not as the user who executed it. This permission is set on a handful of system utilities so that they can be run by normal users, but executed with the permissions of root, providing access to system files that the normal user doesn't normally have access to.

The setgid permission is similar to setuid, but it makes use of the group owner permissions. There are two forms of setgid permissions: setgid on a file and setgid on a directory. The behavior of setgid depends on whether it is set on a file or directory.

* 1. \*\*Для чого в системі потрібен так званий “липкий біт” (Sticky Bit). Наведіть приклади коли цей дозвіл доцільно використовувати.

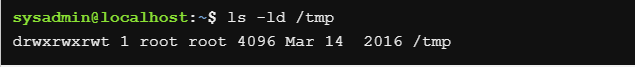
The sticky bit permission is used to prevent other users from deleting files that they do not own in a shared directory. Recall that any user with write permission on a directory can create files in that directory, as well as delete any file in the directory, even if they do not own the file!

The sticky bit permission allows for files to be shared with other users, by changing write permission on the directory so that users can still add and delete files in the directory, but files can only be deleted by the owner of the file or the root user.

A good example of the use of sticky bit directories would be the /tmp and /var/tmp directories. These directories are designed as locations where any user can create a temporary file.

Because these directories are intended to be writable by all users, they are configured to use the sticky bit. Without this special permission, users would be able to delete any files in this directory, including those that belong to other users.

The output of the ls -l command displays the sticky bit by a t character in the execute bit of the others permission group.



To set the sticky bit permission symbolically, execute a command like the following:

chmod o+t <*directory*>

To set the sticky bit permission numerically, add 1000 to the directory's existing permissions (assume the directory in the following example originally had 775 for its permissions):

chmod 1775 <*file*|*directory*>

To remove the sticky permission symbolically, run:

chmod o-t <*directory*>

‌⁠​​⁠​To remove the sticky bit permission numerically, subtract 1000 from the directory's existing permissions:

chmod 0775 <*directory*>

1. Підготувати в електронному вигляді початковий варіант звіту:

* Титульний аркуш, тема та мета роботи
* Словник термінів
* Відповіді на п.4.1-4.6 з завдань для попередньої підготовки

**Хід роботи:**

***Готував матеріал студент Корольов Є. та Горохов Д.(401 ауд.)***

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
2. Запустіть операційну систему Linux Ubuntu. Виконайте вхід в систему та запустіть термінал ***(якщо виконуєте ЛР у 401 ауд.)***.
3. Запустіть віртуальну машину Ubuntu\_PC ***(якщо виконуєте завдання ЛР через академію netacad)***
4. Запустіть свою операційну систему сімейства Linux ***(якщо працюєте на власному ПК та її встановили)*** та запустіть термінал.
5. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу NDG Linux Essentials - Lab 15: System and User Security та Lab 16: Creating Users and Groups. Створіть таблицю для опису цих команд

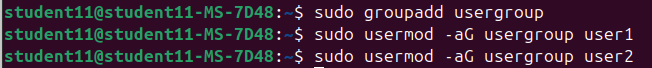
|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| id | The id command can be useful for verifying which user account you are using and which groups you have available to use. By viewing the output of this command, you can see the user's identity information expressed both as a number and as a name. |
| groups | The output of the groups command may not be as detailed as the output of the id command, but if all you need to know is what groups you can switch to by using the newgrp command, then the groups command provides the information that you need. The id command output does show your current primary group, so it is useful for verifying that the newgrp command succeeded. |
| chgrp | To change the group owner of an existing file the chgrp command can be used. |
| chown | The chown command allows the root user to change the user ownership of files and directories. A regular user cannot use this command to change the user owner of a file, even to give the ownership of one of their own files to another user. However, the chown command also permits changing group ownership, which can be accomplished by either root or the owner of the file. |
| umask | The umask command is a feature that is used to determine default permissions that are set when a file or directory is created. Default permissions are determined when the *umask value* is subtracted from the maximum allowable default permissions |
| passwd | Now consider the passwd command. When this command runs, it modifies the /etc/shadow file, which seems impossible because other commands that the sysadmin user runs that try to access this file fail. |

* 1. Виконайте наступні практичні завдання у терміналі наступні дії (продемонструвати скріншоти):
* створіть трьох нових користувачів;

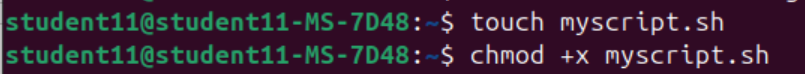




* створіть нову групу користувачів, туди додайте двох, з трьох створених користувачів;

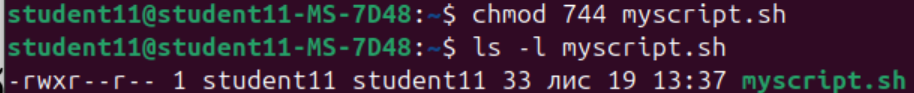


* створіть новий файл, який буде доступний на зчитування, редагування та виконання власником файлу, наприклад найпростіший скриптовий сценарій;



* 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;
* директорія, доступна для всіх трьох користувачів:







* директорія, доступна лише для власника:



* директорія, для перегляду групи, без редагування:





* \*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?

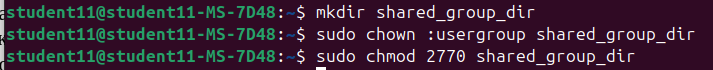


000: усі доступи(читання, запис, виконання) заборонені.

chmod 4 emptyfile - власник отримує читання, група і інші залишаються без доступу.

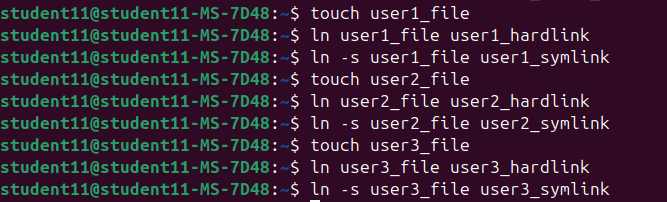
chmod 44 emptyfile - (040) група отримує читання, (004) інші отримують читання.

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



2770 - встановлює setgrid змушує всі файли успадкувати групу каталогу. Тобто усі файли належать групі usergroup.

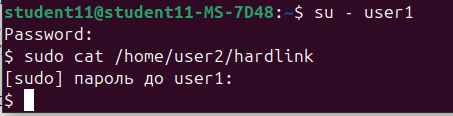
* \*\*під кожним користувачем створіть по одному новому файлу, та жорстке та символічне посилання на нього;



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

Перед виконання користувачам необхідно надати права доступу:





файл з жорстким посиланням пустий, але переглянути іншими користувачами можливо.

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



chmod works with three types of permissions: owner, group, and other.

setgid makes files inherit a directory group.

Hard links allow you to work with files even after deleting the original.

Only the owner or a user with the appropriate rights can delete a file.

**Контрольні запитання:**

***Готувала матеріали студентка Коваленко С.***

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

If you want to modify some of the current permissions, the symbolic method is usually easier to use. With this method, you specify which permissions you want to change on the file, and the other permissions remain as they are.

When specifying the new\_permission argument of the chmod command using the symbolic method three types of information are required.

Start by using one or more of the following characters to indicate which permission group(s) to apply the changes to:

u user owner

g group owner

o others

a all (user owner, group owner, and others)

‌⁠​​⁠

Then choose one of the following operators to indicate how to modify the permissions:

+ add

- remove

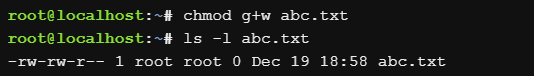
= equals

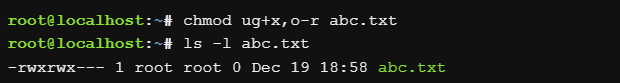
Lastly, use the following characters to specify the permissions type(s) to change:

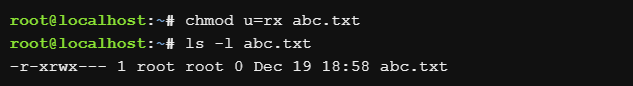
r read

w write

x execute







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

The numeric method (also called the octal method) is useful when changing many permissions on a file. It is based on the octal numbering system in which each permission type is assigned a numeric value:

4 Read

2 Write

1 Execute

By using a combination of numbers from 0 to 7, any possible combination of read, write and execute permissions can be specified for a single permission group set. For example:

7 rwx

6 rw-

5 r-x

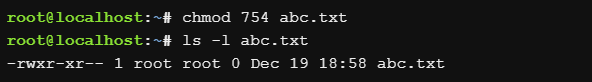
4 r--

3 -wx

2 -w-

1 --x

0 ---



1. Яке призначення команди umask?

The umask command is a feature that is used to determine default permissions that are set when a file or directory is created. Default permissions are determined when the umask value is subtracted from the maximum allowable default permissions. The maximum default permissions are different for files and directories:

file rw-rw-rw-

directories rwxrwxrwx



1. Порівняйте жорсткі та символічні посилання?

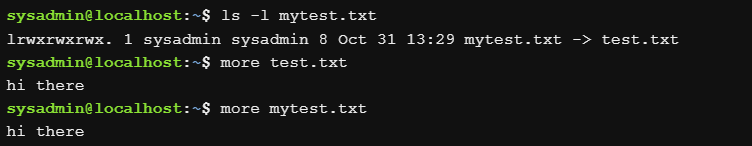
While hard and symbolic links have the same result, the different techniques produce different advantages and disadvantages. In fact, the advantage of one technique compensates for a disadvantage of the other technique.

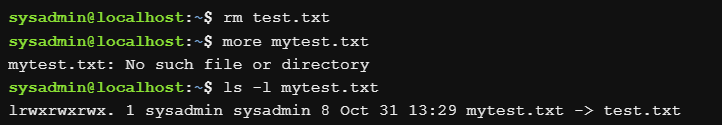
Advantage: Hard links don’t have a single point of failure.

One of the benefits of using hard links is that every file name for the file content is equivalent. If you have five files hard linked together, then deleting any four of these files would not result in deleting the actual file contents.

Recall that a file is associated with a unique inode number. As long as one of the hard linked files remains, then that inode number still exists, and the file data still exists.

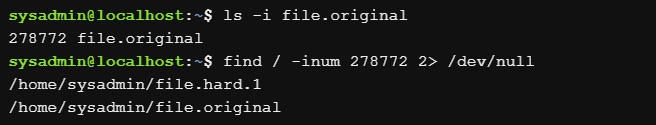
Symbolic links, however, have a single point of failure: the original file. Consider the following example in which access to the data fails if the original file is deleted. The mytest.txt file is a symbolic link to the text.txt file.





Advantage: Soft links are easier to see.

Sometimes it can be difficult to know where the hard links to a file exist. If you see a regular file with a link count that is greater than one, you can use the find command with the -inum search criteria to locate the other files that have the same inode number. To find the inode number you would first use the ls -i command.



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

Execute permission allows a file to be run as a program or script. If a file has this permission, the system allows the user to pass the file to an interpreter or kernel for execution. Read-only means that the user cannot view the contents of the file. Commands will not be able to open the file for viewing or editing. Execute permission allows you to run a file without having to read its contents directly. When a file is executed, the process accesses the file through internal system mechanisms that are independent of the user's read permissions.

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

Access rights and permissions to files and directories are stored in the file system.

When you change access rights using commands such as chmod, these changes are written to the file system and are saved even after the current session is terminated (exiting the terminal, rebooting the system).

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

The umask command is a feature that is used to determine default permissions that are set when a file or directory is created. Default permissions are determined when the umask value is subtracted from the maximum allowable default permissions. The maximum default permissions are different for files and directories.



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

A hard link can be created using the ln command .



One of the benefits of using hard links is that every file name for the file content is equivalent. If you have five files hard linked together, then deleting any four of these files would not result in deleting the actual file contents.

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

Soft links are much more visual, not requiring any extra commands beyond the ls command to determine the link:



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 correct directory to create a one-time temporary file that is not needed after the application is closed is /tmp

This is a special directory for temporary files, accessible to all users of the system.

Files created in /tmp are usually automatically deleted by the system after a reboot or after a certain period of time.

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.

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Original link | Symbolic link | Hard link |
| Deleting the original | Data is stored via a hard link | Symbolic link becomes broken | Hard link remains functional |
| Deleting symlink | No changes | Link is deleted | No changes |
| Deleting hard link | No changes (if there are other links) | No changes | Only this link is deleted |

Висновки:

In the course of the work, practical skills in working with the Bash command shell were consolidated, including the use of basic commands to change file owners and access rights. Commands that are key to system administration and resource access control were discussed.

The features of working with special Linux directories and files, such as /etc, /dev, /tmp, /var and others, were also studied. Their purpose, functionality, and security principles that ensure the isolation of data and processes between users were considered.

This work allowed us to gain an understanding of the basic aspects of file and access rights management, which is an important component of Linux system administration. The knowledge gained is the foundation for further study of more complex topics related to file system security and optimisation.