“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 КСМ-13а

Команда “Viper”:

Малієнко А. М.,

Мішин А. О.

та Нерощин Д.О.

Перевірив викладач

Сушанова В.С.

Київ 2023

**Мета роботи:**

**1. Отримання практичних навиків роботи з командною оболонкою Bash.**

**2. Знайомство з базовими діями при зміні власників файлів.**

**3. Знайомство з базовими діями при зміні прав доступу до файлів**

**4. Знайомство з спеціальними каталогами та файлами в Linux.**

**Матеріальне забезпечення занять**

**1. ЕОМ типу IBM PC.**

**2. ОС сімейства Windows (Windows 7).**

**3. Віртуальна машина – Virtual Box (Oracle).**

**4. Операційна система GNU/Linux – CentOS.**

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

***Готувала матеріал студентка Малієнко А.***

***Завдання для попередньої підготовки.***

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

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

***- Chapter 17 - Ownership and Permissions***

***- Chapter 18 - Special Directories and Files***

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

***- Chapter 17 Exam***

***- Chapter 18 Exam***

***4. На базі розглянутого матеріалу дайте відповіді на наступні питання:***

***4.1. Яке призначення команди id?***

The `id` command in Unix-like operating systems is used to display the user and group IDs associated with the current user or a specified username. It also shows additional information such as the user's primary group and supplementary groups.

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

To view the permissions of a file's owner, you can use the `ls` command with the `-l` option in Unix-like operating systems. Specifically, the owner's permissions are displayed in the first set of characters in the file listing.

The command in English would be:

*ls -l filename*

This command provides a detailed listing of the file, including information about the file's permissions, ownership, size, modification time, and more. The owner's permissions are represented by the characters in the first position of the output. For example:

*$ ls -l myfile.txt*

*-rw-r--r-- 1 owner group 1024 Nov 1 10:00 myfile.txt*

In this example, the owner has read and write permissions (rw-) for the file. The three sets of characters represent permissions for the owner, group, and others, respectively.

***4.3. Як змінити власника групи?***

To change the owner of a group, you can use the `chown` command in Unix-like operating systems. The `chown` command allows you to change the ownership of files and directories, including changing the group ownership.

Here's the basic syntax to change the group owner:

*sudo chown :newgroup filename*

Replace `newgroup` with the desired group name, and `filename` with the name of the file or directory for which you want to change the group ownership.

For example, if you want to change the group ownership of a file named "example.txt" to a group called "newgroup," you would run:

*sudo chown :newgroup example.txt*

Make sure to use `sudo` to execute the command with elevated privileges, especially if you are changing ownership of system files or directories.

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

To view the type of the current file in the terminal, you can use the `file` command.

Example:

*file example.txt*

*file my\_program*

*file image.jpg*

*file archive.zip*

***4.5. Для чого використовуються дозволи Setuid та Setgid?***

1. Setuid (Set User ID):

Purpose: When the setuid permission is set on an executable file, the program runs with the effective user ID of the file owner.

Use Case: This is often used for programs that require elevated privileges to perform specific tasks. For example, the `passwd` command needs to update the password file, a task that regular users do not have permission to do. Therefore, `passwd` is setuid to allow users to change their passwords without giving them full root access.

2. Setgid (Set Group ID):

Purpose: When the setgid permission is set on an executable file, the program runs with the effective group ID of the file's group owner.

Use Case: This is commonly used in scenarios where a program needs to access files that are shared among members of a specific group. For example, the `cp` command is setgid to allow users to create a copy of a file while preserving the group ownership of the original file.

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

The "sticky bit" is a special permission in Unix-like operating systems that serves a specific purpose for directory permissions. When the sticky bit is set on a directory, it ensures that only the owner of a file within that directory can delete or rename their own files. This provides a level of privacy and security, particularly in shared or public directories where multiple users have write access. An example use case is the `/tmp` directory, where setting the sticky bit ensures users can only delete their own temporary files, enhancing overall system security.

***5. 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***

***Готував матеріал студент Нерощин Д.***

***Хід роботи.***

***1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:***

***1.1. 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.***

***1.2. Запустіть віртуальну машину Ubuntu\_PC (якщо виконуєте завдання ЛР через академію netacad)***

***1.3. 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***

***- Lab 18: Special Directories and Files***

***3. Створіть таблицю команд вивчених у п.2 ходу роботи у наступному вигляді:***

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

4. 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.

**Контрольні запитання**

Символічний метод:

Давайте розглянемо приклад використання символічного методу для зміни прав доступу до файлу "example.txt". Наприклад, я хочу, щоб власник мав права на читання, запис та виконання, група - тільки на читання, а інші не мали жодних прав.

chmod u=rwx,g=r,o= example.txt

Числовий метод:

Той самий приклад можна виразити числовим методом. Наприклад, я встановлю числові права 744 для файлу "example.txt", що відповідає правам власника на читання, запис та виконання, а групи та іншим - на читання.

chmod 744 example.txt

Виконання файлу без прав на читання (--x):

Так, можливо виконати файл, навіть якщо відсутні права на читання. Це через те, що виконання дозволяє виконати вміст файлу, навіть якщо його не можна прочитати.

Призначення команди umask:

Учні, команда umask визначає біти, які будуть відняті від прав доступу при створенні нового файлу чи каталогу. Вона встановлює маску того, які права не будуть надані.

Збереження прав доступу:

Важливо зазначити, що зміни, які ви вносите до прав доступу в поточній сесії, не зберігаються після того, як ви закриєте свою робочу сесію. Вони діють лише на протязі поточного використання терміналу чи сесії.

Шаблон за замовчуванням:

Система використовує шаблон за замовчуванням для визначення прав доступу при створенні нових файлів. Цей шаблон визначається змінною umask.

Створення тимчасового файлу:

Як ваш учитель, я б порадив створювати тимчасові файли у каталозі "/tmp". Цей каталог призначений саме для таких файлів, і його регулярно очищає система.

Жорстке посилання:

Жорсткі посилання створюються командою ln. Наприклад, ln source\_file hard\_link. Вони вказують на той самий блок даних на диску.

Символічне посилання:

Для створення символічного посилання використовуйте команду ln -s source\_file symbolic\_link. Вони вказують на ім'я файлу, а не на його фізичний блок.

Порівняння посилань:

Жорсткі посилання вказують на той самий блок на диску, тоді як символічні вказують на ім'я файлу. Жорсткі можуть перехрещуватися в межах файлової системи, символічні - між ними.

Видалення файлів і посилань:

Якщо видалити оригінальний файл, обидва посилання залишаться, але будуть вказувати на ніщо. Якщо видалити символічне посилання, сам файл залишиться. Якщо видалити жорстке посилання, дані на диску залишаться, доки існує хоча б одне посилання.