“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”**

Виконав(ла/ли) студент(ка/и)

групи КСМ-13а

European TRO: Козаченко

М.О., Савіч М.М.

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

**Завдання для попередньої підготовки**

***Готував матеріал студент Савіч Матвій.***

1. 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 | Власність файлу |
| User owner | Власник користувача |
| File and directory permissions | Права доступу до файлів та каталогів |
| Default permissions | Стандартні права доступу, які надаються файлам та каталогам при їх початковому створенні |
| Group owner | Власник групи |
| GID (Group ID) | Унікальний ідентифікатор групи в операційній системі |
| UID (User ID) | Ідентифікатор користувача |

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

Команда id використовується для відображення інформації про ідентифікатор користувача (UID - User ID) та групи (GID - Group ID). Вона виводить інформацію про поточного користувача системи, таку як ідентифікатор користувача (UID), ім'я користувача, ідентифікатор головної групи (GID), ім'я головної групи та список додаткових груп, до яких належить користувач.

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

Щоб це зробити треба скористатися командою ls з опцією –l.

* 1. Як змінити власника групи?

Для зміни власника групи файлу або каталогу використовується команда chown (change owner). Щоб змінити власника треба скористатися такою командою:

sudo chown :нова\_група файл\_або\_каталог

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

Для перегляду типу поточного файлу у терміналі можна використати команду file.

Приклади:

1. file example.txt – текстовий файл;
2. file mydirectory – каталог;
3. file myarchive.zip - ZIP-архів;
4. file mydocument.pdf - PDF-документ;
5. file myimage.png - графічний файл (зображення PNG).
   1. Для чого використовуються дозволи Setuid та Setgid?

Дозволи Setuid та Setgid використовуються для надання додаткових привілеїв виконання виконуваним файлам.

* Коли встановлено Setuid на виконуваний файл, він виконується в контексті власника файлу замість користувача, який його запустив. Це дозволяє виконувати деякі операції з підвищеними привілеями. Зазвичай встановлюється на програми, що потребують підвищених привілеїв для виконання конкретних завдань.
* Коли встановлено Setgid на виконуваний файл або каталог, він виконується в контексті групи-власника файлу або каталогу замість групи користувача, який його запустив. Це дозволяє спільний доступ до файлів та ресурсів між користувачами, які входять у ту ж групу. Часто встановлюється на каталоги, які користується група користувачів і яким потрібен спільний доступ до файлів в цьому каталозі. Це допомагає уникнути проблем із зміною власника файлу на іншу групу.
  1. Для чого в системі потрібен так званий “липкий біт” (Sticky Bit). Наведіть приклади коли цей дозвіл доцільно використовувати.

Це спеціальний дозвіл у файловій системі , який має кілька варіантів застосування.

Основне призначення липкого біта - обмеження можливості видалення або перейменування файлів у каталозі для користувачів, які не є власниками цих файлів або самого каталогу. Липкий біт також може використовуватися для створення тимчасових файлів. У папках з встановленим липким бітом, будь-який користувач може створити файли, але може видалити або змінити тільки ті файли, які він сам створив. Коли липкий біт встановлений на директорії, це дозволяє користувачам змінювати тільки власні файли в цій директорії, навіть якщо вони мають права запису до неї.

Приклад:

Директорія /tmp - це одне з типових місць, де застосовується липкий біт. Користувачі можуть створювати файли та директорії в /tmp, але вони можуть видалити лише ті файли або директорії, які вони самі створили. Це забезпечує деяку рівновагу між доступом користувачів до тимчасових файлів і запобігає ненавмисному видаленню чи модифікації файлів, які можуть бути важливими для інших користувачів або для системи.

**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:

**\*Тут табл\***

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.

**Відповіді на контрольні запитання**

***Готував матеріал студент Савіч Матвій.***

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

Приклади:

1. Додавання прав доступу:

* Додавання права читання (read) для групи до файлу example.txt:

chmod g+r example.txt

* Додавання прав виконання (execute) для усіх користувачів до каталогу my\_directory:

chmod a+x my\_directory

* Додавання прав запису (write) для власника та групи до файлу data.csv:

chmod ug+w data.csv

1. Видалення прав доступу:

* Видалення права виконання (execute) для інших користувачів з файлу script.sh:

chmod o-x script.sh

* Видалення права читання (read) для групи та інших користувачів з каталогу documents:

chmod go-r documents

1. Зміна конкретних прав доступу:

* Зміна прав доступу до файлу config.ini, дозволяючи власнику виконання, групі читання інших запису:

chmod u+x,g+r,o+w config.ini

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

Приклади:

1. Додавання прав доступу:

* Додавання права читання (read) для групи до файлу example.txt:

chmod g+r example.txt

* Цей приклад може бути перетворений в числовий метод як:

chmod 640 example.txt

У цьому випадку:

* 6 встановлює права для власника: read (4) + write (2) = 6.
* 4 встановлює права для групи: read (4).
* 0 встановлює відсутність прав для інших користувачів.

1. Видалення прав доступу:

* Видалення права виконання (execute) для усіх користувачів з каталогу my\_directory:

chmod a-x my\_directory

* Це може бути зроблено числовим методом так:

chmod 644 my\_directory

У цьому випадку:

* 6 встановлює права для власника: read (4) + write (2) = 6.
* 4 встановлює права для групи: read (4).
* 4 встановлює права для інших користувачів: read (4).

1. Зміна конкретних прав доступу:

* Зміна прав доступу до файлу `script.sh`, дозволяючи власнику виконання, групі читання та іншим запису:

chmod u+x,g+r,o+w script.sh

* Це можна зробити числовим методом так:

chmod 754 script.sh

У цьому випадку:

* 7 встановлює права для власника: read (4) + write (2) + execute (1) = 7.
* 5 встановлює права для групи: read (4) + execute (1) = 5.
* 4 встановлює права для інших користувачів: read (4).

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

Поясніть.

Так, можливо виконати файл. Права на читання не є обов'язковими для виконання файлу, якщо права на виконання встановлені для користувача чи групи, до яких належить користувач, який намагається виконати цей файл.

1. Яке призначення команди umask?

Команда umask встановлює або виводить маску режиму доступу за умовчанням для новостворених файлів та каталогів у поточному середовищі оболонки. Маска режиму доступу (UMask) визначає права, які автоматично віднімаються від режиму доступу, наданого новому файлу або каталогу під час їх створення. Основна функція umask полягає в контролі стандартних прав доступу, що застосовуються до нових файлів та каталогів, створюваних користувачем в операційній системі.

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

Вони не будуть збережені.

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

Так, існує шаблон, який використовується для встановлення прав доступу за замовчуванням при створенні нових файлів та каталогів. Цей шаблон називається umask. Щоб змінити права доступу за замовчуванням для нових файлів та каталогів, ви можете використати команду umask в оболонці (shell) для налаштування маски режиму доступу.

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

Для створення тимчасового файлу, який більше ніколи не буде потрібен після закриття програми, існує спеціальний каталог, призначений для таких цілей - це каталог для тимчасових файлів (Temporary directory). звичайних випадках, цей каталог зазвичай знаходиться за шляхом /tmp. Якщо програмі потрібно створити одноразовий тимчасовий файл, який більше не буде використовуватися після закриття програми, ви можете створити цей файл у каталозі для тимчасових файлів /tmp.

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

Щоб створити жорстке посилання треба застосувати команду ln.

ln /шлях\_до\_початкового\_файлу /нове\_ім'я\_для\_посилання

Основна особливість жорстких посилань полягає в тому, що вони посилаються на ту саму вузол файлу, що й початковий файл, тобто жорстке посилання та оригінальний файл ділять один і той же блок даних на диску.

Ситуації в яких їх доцільно використовувати:

1. Забезпечення доступу до одного файлу з різних місць в файловій системі.

Вони дозволяють вам мати кілька шляхів до того самого файлу. Наприклад, ви можете створити посилання в папці, що зручна для вас, і мати доступ до того ж файлу з іншої локації.

1. Збереження місця на диску.

Якщо у вас є велика кількість однакових файлів або ви хочете зекономити місце на диску, ви можете створити жорсткі посилання замість створення кількох копій файлів.

1. Забезпечення збереження файлу після його переміщення або перейменування.

Якщо ви створили жорстке посилання на файл, ви можете змінити або перемістити початковий файл, і посилання залишиться активним.

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

Щоб створити символічне посилання треба застосувати команду ln.

ln -s /шлях\_до\_цільового\_файлу\_або\_каталогу /нове\_ім'я\_для\_посилання

Ситуації в яких їх доцільно використовувати:

1. Створення посилання на файли або каталоги в інших місцях файлової системи.

Вони дозволяють вам мати посилання на файл або каталог з будь-якого місця в файловій системі, включаючи інші файлові системи або мережеві ресурси.

1. Спрощення доступу до файлів та каталогів.

Вони забезпечують можливість створювати короткі або зручні посилання на довгі або складні шляхи файлової системи.

1. Робота з симлінками у скриптах або програмах.

Символічні посилання дуже корисні при написанні скриптів або програм, оскільки вони дозволяють зручно посилатися на різні файли та каталоги.

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

- файл оригінал;

Якщо ви видалите оригінальний файл, до якого були створені посилання, це призведе до видалення фактичних даних, на які вказував оригінальний файл. Якщо це був єдиний файл з цими даними, вони будуть втрачені, навіть якщо у вас є посилання на ці дані.

- символічне посилання;

Видалення символічного посилання не вплине на дані в оригінальному файлі або в інших посиланнях. Символічне посилання є просто шляхом до оригінального файлу, а не самими даними. При видаленні символічного посилання ви просто втратите зв'язок з оригінальним файлом, але сам файл залишиться на місці.

- жорстке посилання.

Жорстке посилання посилається на ті самі дані, що й оригінальний файл. Якщо ви видалите жорстке посилання, це не призведе до видалення фактичних даних, на які вказував оригінальний файл або інші посилання. Фактичні дані будуть зберігатися на диску до тих пір, поки є хоча б одне посилання на ці дані. Якщо ви видалите всі посилання на ці дані, вони вже не будуть доступні, навіть якщо самі дані все ще зберігаються на диску. Вони будуть видалені після того, як останнє посилання на них буде видалено.

**Висновки**