“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

Литвинюк Євген

Меліхов Данило

Бумажний Микола

Перевірив викладач

Сушанова В.С.

Київ 2024

**Робота студентів групи БІКС-13 Команда OCG: Литвинюк Є., Меліхов Д., Бумажний М..**

**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(Виконав Литвинюк Євген):**

1. \*Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань призначення команд та їх параметрів.
2. Command (команда):

* id: Виводить інформацію про ідентифікатори користувача та групи.
* chgrp: Змінює власника групи файлу.
* file: Переглядає тип поточного файлу.

1. File Permissions (права доступу до файлів):

* chmod: Змінює права доступу до файлу або директорії.

1. Special Directories and Files (спеціальні директорії та файли):

* setuid: Дозволяє програмам запускатися в контексті власника файлу.
* setgid: Дозволяє програмам запускатися в контексті групи файлу.
* Sticky Bit (липкий біт): Обмежує можливість видалення файлів з директорії.

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

* Команда id виводить інформацію про ідентифікатори користувача та групи.
  1. Як переглянути які права доступу має власник файлу?
* Це можна зробити за допомогою команди ls -l, яка показує список файлів та їх права доступу.
  1. \*Як змінити власника групи?
* Для зміни власника групи використовується команда chgrp.
  1. \*Як можна переглянути у терміналі який тип поточного файлу? Наведіть приклади для різних типів файлів.
* Для перегляду типу поточного файлу використовується команда file.
* Приклади:

file script.sh - виведе тип файлу script.sh.

file /dev/sda - виведе тип блочного пристрою /dev/sda.

file /etc/passwd - виведе тип звичайного текстов

* 1. \*\*Для чого використовуються дозволи Setuid та Setgid?
* Setuid: Дозволяє програмам запускатися в контексті власника файлу, незалежно від того, хто запускає цю програму. Це корисно для програм, які потребують підвищених привілеїв певного користувача, наприклад, програми для адміністрування системи.
* Setgid: Дозволяє програмам запускатися в контексті групи файлу, незалежно від того, хто запускає цю програму. Це корисно для програм, які потребують спільного доступу до файлів або ресурсів для групи користувачів.
  1. \*\*Для чого в системі потрібен так званий “липкий біт” (Sticky Bit). Наведіть приклади коли цей дозвіл доцільно використовувати.
* Липкий біт (Sticky Bit) застосовується до директорій і має наступні основні цілі:
* Забороняє видалення файлів з директорії користувачам, крім власника файлу, власника директорії та адміністратора системи.
* Цей дозвіл зазвичай використовується для тимчасових директорій, таких як /tmp, де користувачі можуть створювати файли, але не можуть видаляти або редагувати файли інших користувачів.

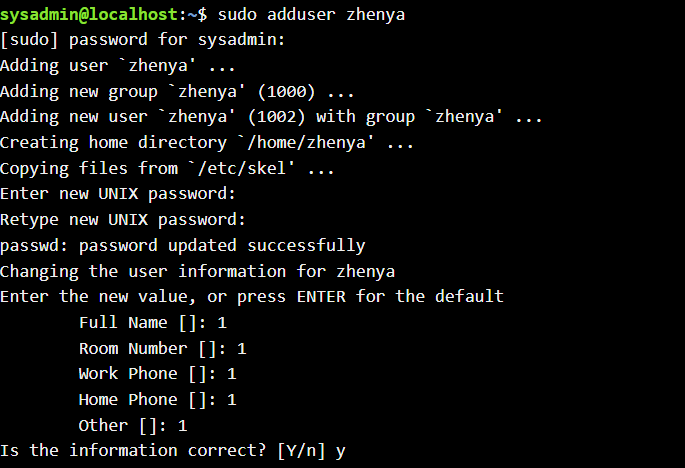
**Хід роботи(Виконав Меліхов Данило):**

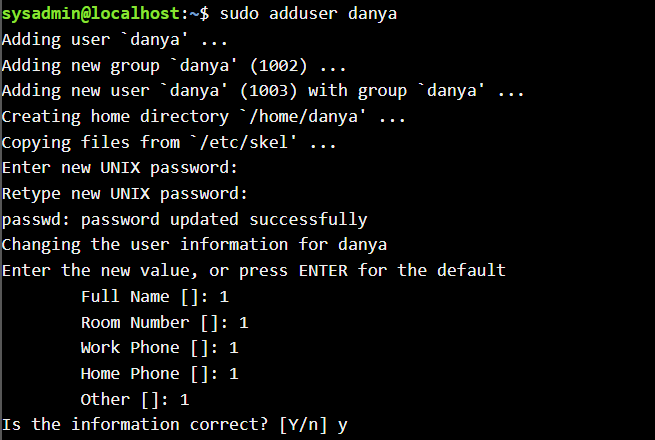
1. Опрацюйте всі приклади команд, що представлені у лабораторних роботах курсу ***NDG Linux Essentials: Lab 17: Ownership and Permissions*** та ***Lab 18: Special Directories and Files.*** Створіть таблицю команд вивчених у п.2 ходу роботи у наступному вигляді:

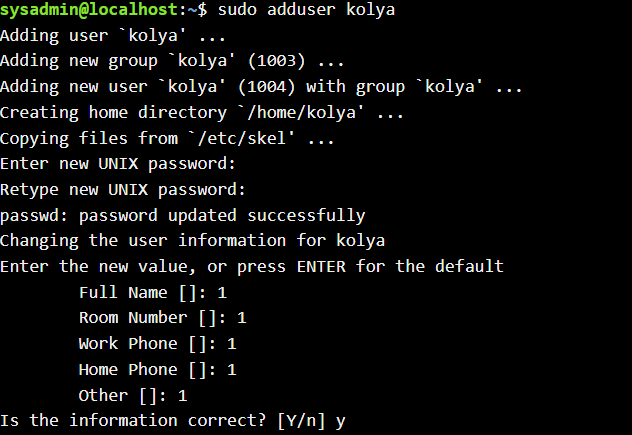
|  |  |
| --- | --- |
| **Назва команди** | **Призначення та функціональність** |
| **‘file'** | Визначає тип файлу. |
| **‘ln’** | Створює символічні посилання на файли. |
| **‘setuid’** | Встановлює біт setuid на виконуваний файл. |
| **‘setgid’** | Встановлює біт setgid на виконуваний файл. |
| **‘id’** | Виводить інформацію про ідентифікатори користувача та групи. |
| **‘ls -l’** | Виводить список файлів з їхніми правами доступу. |
| **‘chmod’** | Змінює права доступу до файлу або директорії. |
| **‘chown’** | Змінює власника файлу або директорії. |
| **‘chgrp’** | Змінює власника групи файлу або директорії. |
| **‘chmod +t’** | Встановлює липкий біт на директорію. |

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

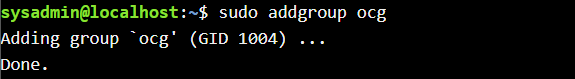
* створіть трьох нових користувачів;







* 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;



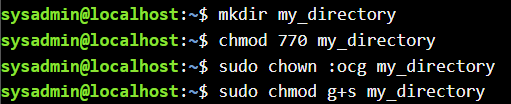
* \*подібні дії виконайте для директорій - створіть директорію, яка буде доступна для всіх трьох користувачів, створіть директорію, яку буде доступна тільки для власника, створіть директорію, яку користувачі групи власника зможуть переглядати, але не редагувати;



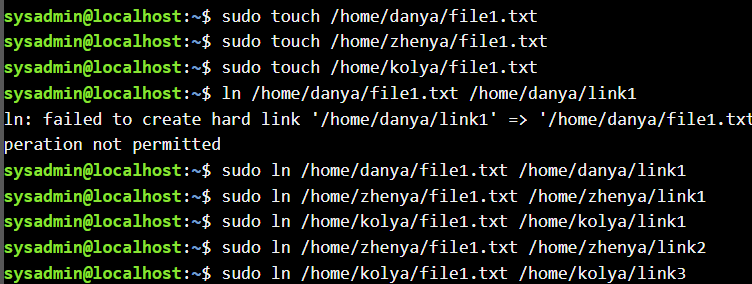
* \*створіть порожній файл під назвою emptyfile за допомогою команди touch emptyfile. Тепер “обнуліть” дозволи для файлу з chmod 000 emptyfile. Що станеться, якщо змінити дозволи для emptyfile, передавши лише одне значення для chmod у числовому режимі, наприклад, chmod 4 emptyfile? Що буде, якщо ми використаємо два числа, наприклад chmod 44 emptyfile? Що ми можемо дізнатися про те, як chmod зчитує числове значення?
* Коли ми використовуємо команду chmod для зміни прав доступу до файлу, ми можемо вказувати ці права в числовому форматі.
* Якщо ми використаємо команду chmod 4 emptyfile, це означає, що ми встановлюємо тільки право на читання для власника файлу.
* Якщо ми використаємо команду chmod 44 emptyfile, це означає, що ми встановлюємо право на читання для власника файлу та групи.
* Коли ми використовуємо команду chmod з одним числом, це означає, що ми встановлюємо права доступу тільки для власника файлу. З двома числами, перше число відповідає власнику, а друге - групі.
* Ця команда зчитує числове значення таким чином:
* Перше число вказує права доступу для власника.
* Друге число вказує права доступу для групи.
* Третє число вказує права доступу для інших користувачів.



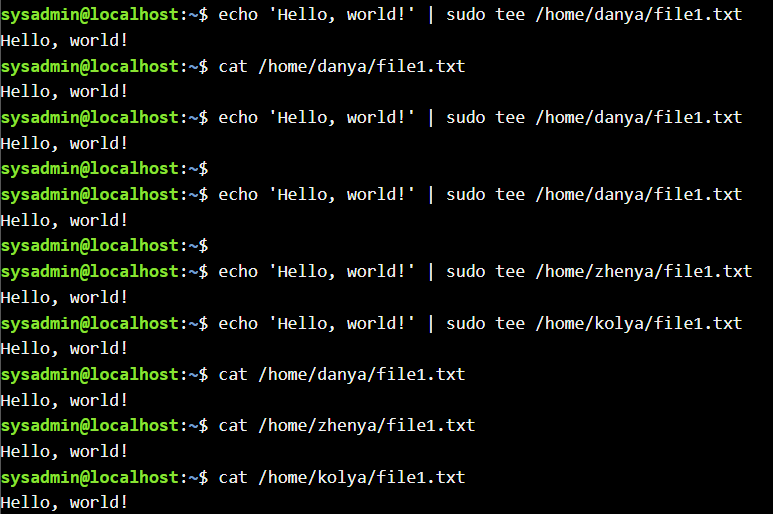
* \*\*створіть каталог під назвою, де всі файли автоматично будуть належати Вашій групі користувачів і можуть бути видалені лише користувачем, який їх створив?



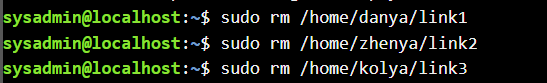
* \*\*під кожним користувачем створіть по одному новому файлу, та жорстке та символічне посилання на нього;



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



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



**Контрольні запитання(Виконав Бумажний Микола):**

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

**chmod u+x file.txt** # Додає право виконання власнику файла

**chmod g-w file.txt** # Забирає право запису групі до файла

**chmod o=r file.txt** # Встановлює тільки право читання для інших користувачів

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

**chmod 755 file.txt** # Встановлює права rwx для власника та rx для групи та інших

**chmod 644 file.txt** # Встановлює права rw для власника та групи, та r для інших

**chmod 600 file.txt** # Встановлює тільки права rw для власника файла

1. *Яке призначення команди umask?*

Команда umask встановлює маску режиму доступу за замовчуванням для нових файлів, яка віднімається від максимальних прав. Наприклад, umask 022 встановлює права -rw-r--r-- для нових файлів.

1. *Порівняйте жорсткі та символічні посилання?*

Жорстке посилання (hard link):

* Вказує на ту саму вузлову структуру файлової системи, що і оригінальний файл.
* Не може бути створений для директорій.
* Якщо видалити оригінальний файл, посилання все ще вказуватиме на той самий файл.

Символічне посилання (symbolic link):

* Посилається на ім'я файлу або директорії.
* Може бути створений для директорій.
* Якщо видалити оригінальний файл, посилання стане недійсним.

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

Ні, не можна. Щоб виконати файл, користувач має мати право на читання файлу, оскільки операційна система повинна зчитати файл, щоб виконати його.

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

Ні, зміни, внесені в права доступу та дозволи в поточній сесії, не будуть збережені в наступних сеансах роботи.

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 маска umask, яка визначає права доступу за замовчуванням для нових файлів. Щоб змінити права за замовчуванням, ви можете встановити нове значення umask у вашому файлі конфігурації shell, наприклад, .bashrc або .profile.

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

Жорстке посилання можна створити за допомогою команди ln:

**ln original\_file.txt hard\_link.txt**

Жорсткі посилання корисно використовувати, коли вам потрібно мати кілька імен файлів, які вказують на один і той же файл у файловій системі. Вони дозволяють вам звертатися до файлу з різних місць в файловій системі, не копіюючи фактичний вміст файлу.

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

Символічне посилання можна створити за допомогою команди ln з опцією -s:

**ln -s original\_file.txt symbolic\_link.txt**

Символічні посилання дозволяють створювати посилання на файли або директорії в будь-якому місці файлової системи. Вони особливо корисні, коли вам потрібно мати посилання на файл або директорію, яка знаходиться у різних місцях файлової системи або навіть на іншому пристрої.

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

Найкраще місце для створення тимчасового файлу - це директорія /tmp. Вона призначена саме для таких цілей.

1. \*\*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, обидва посилання (як символічне, так і жорстке) залишаться, але вони будуть вказувати на недійсний файл.

* символічне посилання;

Якщо ви видалите символічне посилання, оригінальний файл залишиться незмінним, а жорстке посилання буде вказувати на оригінальний файл.

* жорстке посилання.

Якщо ви видалите жорстке посилання, оригінальний файл залишиться незмінним, а символічне посилання стане недійсним.

**Висновок:**

У ході виконання лабораторної роботи ми ознайомилися з основними концепціями власності та дозволів у Linux, а також засвоїли базові команди для роботи з файлами, директоріями та посиланнями. Ми вивчили різні методи зміни прав доступу до файлів, а саме символічний та числовий методи, із зазначенням прикладів їх використання. Також ми вивчили призначення команди umask та навчилися створювати жорсткі та символічні посилання на файли.

Наша робота показала, що використання різних методів встановлення прав доступу дозволяє гнучко налаштовувати рівні доступу до файлів та директорій в залежності від потреб користувачів та вимог безпеки системи. Також ми дослідили вплив видалення оригінального файлу на його посилання, що є важливим аспектом при роботі з посиланнями в Linux.

Ця робота допомогла нам краще зрозуміти концепції власності та дозволів в Linux, а також засвоїти основні команди для роботи з файлами та директоріями. Тепер ми готові до подальшого вивчення системного адміністрування Linux.