“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»**

Виконавли студенти

групи РПЗ-03

Команда: Губенко Є.О.,

Заїка С.В. та Кресан Р.А.

Перевірив викладач

Сушанова В.С.

Київ 2022

**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*** ***Заїка С. В.***

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

*Команда id використовується в операційних системах, таких як Linux і Unix, для відображення інформації про ідентифікатори користувачів і груп.*

*Зазвичай, коли ви виконуєте команду id без аргументів в командному рядку, вона повертає інформацію про ваш поточний ідентифікатор користувача (UID), ідентифікатор групи (GID) та список інших груп, до яких ви належите. Ця інформація може бути корисною для визначення прав доступу до файлів та каталогів.*

*Крім того, можна вказати іншого користувача або групу в якості аргумента команди id, щоб отримати інформацію про їхні ідентифікатори. Наприклад, команда id someuser поверне інформацію про ідентифікатор користувача та групу для користувача з іменем someuser.*

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

*Для перегляду прав доступу до файлу, вам необхідно використовувати команду ls з аргументом -l, який вказує на відображення детальної інформації про кожен файл в поточному каталозі. Крім того, потрібно вказати назву файлу, права доступу до якого ви хочете переглянути.*

* 1. Як змінити власника групи?

*Щоб змінити власника групи для файлу в операційній системі Linux або Unix, ви можете використати команду chown з опцією -g, яка дозволяє змінити групу власника файлу.*

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

*Щоб переглянути тип поточного файлу у терміналі, можна використати команду file. Вона повертає інформацію про тип файлу згідно з його змістом, а не тільки з розширенням його імені.*

* 1. Для чого використовуються дозволи Setuid та Setgid?

*Дозвіл Setuid (SUID) дозволяє запускати виконуваний файл в контексті власника файлу, а не в контексті користувача, який запускає його. Це дозволяє користувачам отримувати додаткові привілеї, необхідні для виконання певних завдань. Наприклад, команда passwd має дозвіл Setuid, тому користувач може змінити свій пароль, навіть якщо він не має дозволу на запис до файлу /etc/shadow, де зберігаються паролі.*

*Дозвіл Setgid (SGID) дозволяє встановлювати групу власника для виконуваних файлів та директорій. Якщо в директорії встановлено дозвіл SGID, всі нові файли та директорії, створені в цій директорії, будуть мати ту саму групу власника, що й директорія. Це дозволяє групі користувачів спільно працювати з файлами та директоріями, не дозволяючи іншим користувачам зовнішнього середовища отримати доступ до цих файлів та директорій.*

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

*Sticky Bit - це спеціальний дозвіл в операційній системі Unix/Linux, який можна встановлювати для директорій. Якщо дозвіл Sticky Bit встановлено для директорії, то тільки власник файлу може його видалити або перейменувати.*

*Існує декілька сценаріїв використання дозволу Sticky Bit:*

*Забезпечення безпеки в системі файлової структури: Дозвіл Sticky Bit дозволяє забезпечити безпеку даних та запобігти їх випадковому видаленню. Наприклад, в директорії /tmp встановлюється дозвіл Sticky Bit, щоб забезпечити безпеку даних користувачів, які створюють тимчасові файли. Це дозволяє забезпечити, що інші користувачі не можуть випадково видалити або змінити файли, які не належать їм.*

*Спільний доступ до директорії: Якщо в директорії встановлено дозвіл Sticky Bit, то це дозволяє групі користувачів спільно працювати з файлами та директоріями, не дозволяючи іншим користувачам зовнішнього середовища отримати доступ до цих файлів та директорій. Це може бути корисно в тих випадках, коли користувачі мають обмежені права доступу до даних та потребують спільного доступу до спільних ресурсів.*

*Забезпечення приватності даних: Якщо в директорії з важливими даними встановлено дозвіл Sticky Bit, то це дозволяє забезпечити приватність даних та запобігти їх випадковому видаленню.*

**Хід роботи**

***Готував матеріал студент Губенко Є.О.***

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

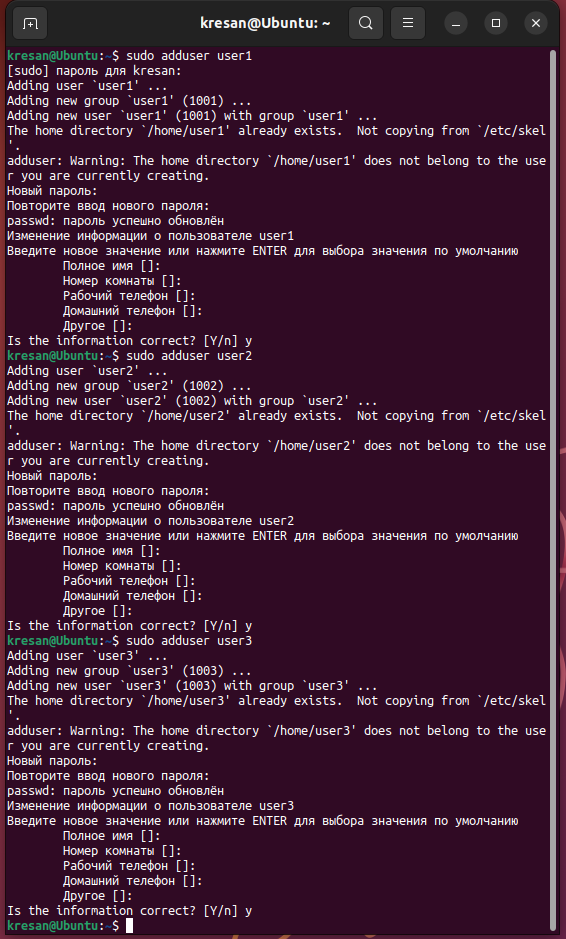
|  |  |
| --- | --- |
| **Назва команди** | **Її призначення та функціональність** |
| cd /tmp | команда "cd /tmp" дозволяє перейти до директорії /tmp, щоб отримати доступ до цих тимчасових файлів і даних, що зберігаються там. Також, зміна поточної директорії до /tmp може бути корисною для виконання деяких команд, що потребують доступу до цієї директорії. |
| mkdir priv-dir pub-dir | команда "mkdir priv-dir pub-dir" дозволяє створити дві нові директорії з відповідними іменами та використовувати їх для збереження приватної та публічної інформації в операційній системі. |
| touch priv-dir/priv-file  touch pub-dir/pub-file | команда "touch priv-dir/priv-file" створить новий файл з ім'ям "priv-file" у директорії "priv-dir". Цей файл може бути використаний для зберігання конфіденційної або приватної інформації, яку необхідно захистити від несанкціонованого доступу.  команда "touch pub-dir/pub-file" створить новий файл з ім'ям "pub-file" у директорії "pub-dir". Цей файл може бути використаний для збереження публічної інформації, на яку можуть мати доступ різні користувачі. |
| ls -l priv-dir  ls -l pub-dir | команда "ls -l priv-dir" виведе список файлів та директорій, які знаходяться в директорії "priv-dir" разом з додатковою інформацією про кожний файл або директорію.  команда "ls -l pub-dir" виведе список файлів та директорій, які знаходяться в директорії "pub-dir" разом з додатковою інформацією про кожний файл або директорію. |
| ls -la | команда "ls -la" виведе список всіх файлів та директорій, які знаходяться в поточній директорії, включаючи приховані файли, разом з додатковою інформацією про кожний файл або директорію. |
| ls -ld priv-dir/  chmod o-rx priv-dir/ | команда "ls -ld priv-dir/" виведе інформацію про директорію "priv-dir", включаючи права доступу, власника та іншу інформацію, але не включаючи список файлів, які знаходяться в цій директорії.  Команда "chmod" призначена для зміни прав доступу до файлів та директорій. Аргумент "o-rx" вказує на вилучення прав на читання та виконання для "інших" користувачів (тобто користувачів, які не є власниками або членами групи файлу). Команда "chmod o-rx priv-dir/" вилучить права на читання та виконання для "інших" користувачів (тобто тих, які не є власниками або членами групи) для директорії "priv-dir". |
| ls -ld priv-dir/ | команда "ls -ld priv-dir/" виводить інформацію про директорію "priv-dir", включаючи права доступу, власника та групу. Аргумент "-d" вказує, що потрібно вивести інформацію саме про директорію "priv-dir", а не про її вміст. |
| ls -ld pub-dir/  chmod o+w pub-dir/  ls -ld pub-dir/ | Команда "ls -ld pub-dir/" виведе інформацію про директорію "pub-dir", включаючи права доступу, власника та групу. Аргумент "-d" вказує, що потрібно вивести інформацію саме про директорію "pub-dir", а не про її вміст.  Команда "chmod" призначена для зміни прав доступу до файлів та директорій. Аргумент "o+w" вказує на додавання прав на запис для "інших" користувачів (тобто користувачів, які не є власниками або членами групи файлу).  команда "chmod o+w pub-dir/" додасть права на запис для "інших" користувачів для директорії "pub-dir". |
| ls -l priv-dir/priv-file  chmod g-rw,o-r priv dir/priv-file  ls -l priv-dir/priv-file | Команда "ls -l priv-dir/priv-file" виводить інформацію про файл "priv-file", який знаходиться в директорії "priv-dir".  Команда "chmod" призначена для зміни прав доступу до файлів та директорій. Аргумент "g-rw,o-r" вказує на видалення прав на читання та запис для групи та "інших" користувачів файлу відповідно.  команда "chmod g-rw,o-r priv-dir/priv-file" видалить права на читання та запис файлу для групи та "інших" користувачів. |
| ls -l pub-dir/pub-file  chmod a=rw pub-dir/pub-file  ls -l pub-dir/pub-file | Команда "ls -l pub-dir/pub-file" виводить інформацію про файл "pub-file", який знаходиться в директорії "pub-dir".  Команда "chmod" призначена для зміни прав доступу до файлів та директорій. Аргумент "a=rw" вказує на надання прав на читання та запис всім (власникові, групі та "іншим" користувачам) файлу.  команда "chmod a=rw pub-dir/pub-file" надасть права на читання та запис всім (власникові, групі та "іншим" користувачам) файлу. |
| echo "date" > test.sh | Команда "echo" виводить текст у стандартний вивід, який можна перенаправити до файлу з використанням символу ">", який створює або перезаписує файл. |
| ./test.sh  ls -l test.sh | Команда "./test.sh" запускає виконання файлу "test.sh", який містить команду "date". Коли ця команда виконується, вона виводить поточну дату та час у стандартний вивід.  Команда "ls -l test.sh" виводить детальну інформацію про файл "test.sh", зокрема, права доступу, власника, групу, розмір файлу, дату та час останньої модифікації та ім'я файлу. Зазвичай, при виконанні команди "ls -l" відображаються такі атрибути файлу, як права доступу, кількість жорстких посилань, власник, група, розмір файлу, дата та час останньої модифікації та ім'я файлу. |
| chmod u+x test.sh  ls -l test.sh  ./test.sh | Команда "chmod u+x test.sh" додає власнику файлу право на виконання файлу "test.sh".  Команда "ls -l test.sh" виведе інформацію про файл "test.sh", включаючи його права доступу. Після того, як виконана команда "chmod u+x test.sh", ви побачите, що права доступу до файлу "test.sh" змінилися, і тепер власник файлу має право на виконання файлу.  Команда "./test.sh" запускає виконання файлу "test.sh" в поточному каталозі. Оскільки ми додали власнику файлу право на виконання, то виконання файлу "test.sh" пройде успішно. Результатом виконання команди буде виведення поточної дати та часу в стандартний вивід. |
| stat test.sh | Команда "stat test.sh" виведе детальну інформацію про файл "test.sh", зокрема, час останньої зміни файлу, час останнього доступу до файлу, час останньої зміни статусу файлу, номер індексного вузла, розмір файлу, тип файлу, права доступу до файлу, власника та групу файлу. |
| chmod 775 test.sh  ls -l test.sh | Команда "chmod 775 test.sh" надає власнику файлу право на читання, запис та виконання, а групі та іншим користувачам - право на читання та виконання файлу "test.sh".  Після виконання команди "chmod 775 test.sh", команда "ls -l test.sh" виведе інформацію про файл "test.sh", включаючи змінені права доступу. |
| su - | Команда "su -" дозволяє користувачеві змінити свій ідентифікатор користувача (UID) та ефективну групу (GID) на ті, які належать іншому користувачеві або системному адміністратору. |
| cd /tmp  ls -ld pub-dir  ls -l pub-dir/pub-file | cd /tmp: ця команда змінює поточний робочий каталог на /tmp, що є системним каталогом для тимчасових файлів у більшості дистрибутивів Linux.  ls -ld pub-dir: ця команда виводить інформацію про каталог pub-dir, що знаходиться в поточному робочому каталозі. Опція -l дозволяє виводити детальну інформацію про кожен елемент каталогу, а -d вказує на те, що потрібно вивести інформацію саме про каталог pub-dir, а не про його вміст.  ls -l pub-dir/pub-file: ця команда виводить детальну інформацію про файл pub-file, що знаходиться в каталозі pub-dir. Опція -l дозволяє виводити детальну інформацію про файл, а шлях pub-dir/pub-file вказує на те, що потрібно вивести інформацію саме про файл pub-file, який знаходиться в каталозі pub-dir. |
| chown root:root pub-dir  ls -ld pub-dir | chown root:root pub-dir: ця команда змінює власника та групу власників для каталогу pub-dir на користувача root та групу root. Зміна власника та групи власників може здійснюватися тільки користувачем з правами адміністратора (root) або користувачем, який має право на зміну власника/групи власників.  ls -ld pub-dir: ця команда виводить інформацію про каталог pub-dir. Опція -l дозволяє виводити детальну інформацію про кожен елемент каталогу, а -d вказує на те, що потрібно вивести інформацію саме про каталог pub-dir, а не про його вміст. Після виконання команди chown можуть змінитися інформація про власника та групу власників каталогу pub-dir. |
| chown bin pub-dir/pub-file  ls -l pub-dir/pub-file | chown bin pub-dir/pub-file: ця команда змінює власника для файлу pub-file, що знаходиться в каталозі pub-dir, на користувача bin. Зміна власника може здійснюватися тільки користувачем з правами адміністратора (root) або користувачем, який має право на зміну власника.  ls -l pub-dir/pub-file: ця команда виводить детальну інформацію про файл pub-file, що знаходиться в каталозі pub-dir. Опція -l дозволяє виводити детальну інформацію про файл, а шлях pub-dir/pub-file вказує на те, що потрібно вивести інформацію саме про файл pub-file, який знаходиться в каталозі pub-dir. Після виконання команди chown може змінитися інформація про власника файлу pub-file. |
| ls -ld priv-dir  ls -l priv-dir/priv-file | ls -ld priv-dir: ця команда виводить інформацію про каталог priv-dir. Опція -l дозволяє виводити детальну інформацію про кожен елемент каталогу, а -d вказує на те, що потрібно вивести інформацію саме про каталог priv-dir, а не про його вміст.  ls -l priv-dir/priv-file: ця команда виводить детальну інформацію про файл priv-file, що знаходиться в каталозі priv-dir. Опція -l дозволяє виводити детальну інформацію про файл, а шлях priv-dir/priv-file вказує на те, що потрібно вивести інформацію саме про файл priv-file, який знаходиться в каталозі priv-dir. |
| ls -ld priv-dir  ls -l priv-dir/priv-file  chgrp -R users priv-dir  ls -ld priv-dir  ls -l priv-dir/priv-file | ls -ld priv-dir: ця команда виводить інформацію про каталог priv-dir. Опція -l дозволяє виводити детальну інформацію про кожен елемент каталогу, а -d вказує на те, що потрібно вивести інформацію саме про каталог priv-dir, а не про його вміст.  ls -l priv-dir/priv-file: ця команда виводить детальну інформацію про файл priv-file, що знаходиться в каталозі priv-dir. Опція -l дозволяє виводити детальну інформацію про файл, а шлях priv-dir/priv-file вказує на те, що потрібно вивести інформацію саме про файл priv-file, який знаходиться в каталозі priv-dir.  chgrp -R users priv-dir: ця команда змінює групу власності для каталогу priv-dir та всіх його вмісту на групу users. Опція -R вказує на рекурсивний прохід по всіх підкаталогах та вмісту каталогу. Зміна групи може здійснюватися тільки користувачем з правами адміністратора (root) або користувачем, який має право на зміну групи власності.  ls -ld priv-dir: ця команда знову виводить інформацію про каталог priv-dir, щоб перевірити, чи змінилися права доступу після виконання команди chgrp.  ls -l priv-dir/priv-file: ця команда знову виводить детальну інформацію про файл priv-file, щоб перевірити, чи змінилися права доступу після виконання команди chgrp. |
| ls -ld /tmp  ls -ld /var/tmp | ls -ld /tmp: ця команда виводить інформацію про каталог /tmp. Опція -l дозволяє виводити детальну інформацію про кожен елемент каталогу, а -d вказує на те, що потрібно вивести інформацію саме про каталог /tmp, а не про його вміст.  ls -ld /var/tmp: ця команда виводить інформацію про каталог /var/tmp. Опція -l дозволяє виводити детальну інформацію про кожен елемент каталогу, а -d вказує на те, що потрібно вивести інформацію саме про каталог /var/tmp, а не про його вміст. |
| ls -l /etc/shadow | Команда ls -l /etc/shadow виводить інформацію про файл /etc/shadow, зокрема, виводить його власника, групу власників, права доступу і дату зміни. Файл /etc/shadow містить захешовані паролі користувачів в системі. |
| ls -l /usr/bin/passwd | Команда ls -l /usr/bin/passwd виводить детальну інформацію про файл /usr/bin/passwd, зокрема, виводить його власника, групу власників, права доступу, дату зміни та розмір. |
| cd | Команда cd без аргументів переводить користувача в домашню директорію, тобто в директорію, яка відповідає його імені користувача. Ця команда корисна, коли потрібно швидко повернутися до домашньої директорії з будь-якої іншої директорії в системі. |
| echo "data" > source | Команда створює новий файл з ім'ям source у поточній директорії та записує в нього рядок data. Оператор > перенаправляє вивід з командного рядка до файлу, тобто в даному випадку вміст рядка буде записаний у файл source. Якщо файл з таким ім'ям вже існує, вміст файлу буде замінений на новий вміст. |
| ls -li source | команда ls -li source виведе інформацію про файл source, зокрема його індекс вузла іноди, кількість посилань, власника та групу власників та ім'я файлу. |
| ln source hardlink  ls -li source hardlink | Команда ln source hardlink створює жорстке посилання (hardlink) з іменем hardlink на файл з іменем source.  Жорстке посилання на файл є іншим ім'ям, яке посилається на той самий вузол іноди, який відповідає за даний файл. Таким чином, якщо ми змінимо вміст файлу source, зміни будуть відображені і у файлі hardlink і навпаки.  Команда ls -li source hardlink виведе інформацію про обидва файли у форматі, схожому на вивід команди ls -li. Індекс вузла іноди обох файлів буде однаковим, оскільки це жорстке посилання. Кількість посилань на файл source збільшиться на одиницю, оскільки ми створили ще одне посилання на цей файл. |
| ln hardlink hardlinktwo  ls -li hardlink hardlinktwo source | Команда ln hardlink hardlinktwo створює ще одне жорстке посилання на файл з іменем hardlink, але з ім'ям hardlinktwo.  Команда ls -li hardlink hardlinktwo source виведе інформацію про всі три файли, що містяться у поточній директорії, у форматі, схожому на вивід команди ls -li. Індекс вузла іноди усіх трьох файлів буде однаковим, оскільки вони є жорсткими посиланнями на один і той самий файл. Кількість посилань на файл hardlink збільшиться на одиницю, оскільки ми створили ще одне посилання на цей файл з ім'ям hardlinktwo. |
| rm hardlinktwo  ls -li source hardlink | Команда rm hardlinktwo видаляє файл з іменем hardlinktwo, але це не вплине на файли source та hardlink, на які він був ссилкою. Оскільки файли source та hardlink все ще існують і мають жорстке посилання на один і той же вузол іноди, вони продовжать фактично представляти той самий файл, незалежно від того, чи існує ще яке-небудь посилання на цей файл.  Команда ls -li source hardlink виведе інформацію про два файли, що містяться у поточній директорії, у форматі, схожому на вивід команди ls -li. Індекс вузла іноди усіх двох файлів буде однаковим, оскільки вони є жорсткими посиланнями на один і той самий файл. Кількість посилань на файл source буде дорівнювати двом, оскільки він має два жорстких посилання з іменами hardlink та source. |
| rm hardlink  ls -li source | Команда rm hardlink видаляє жорстке посилання з іменем hardlink. Оскільки це було єдине жорстке посилання на файл з іменем source, після видалення жорсткого посилання файл з іменем source більше не буде доступним через жорстке посилання з іменем hardlink.  Команда ls -li source виведе інформацію про файл з іменем source, що міститься у поточній директорії, у форматі, схожому на вивід команди ls -li. Індекс вузла іноди для файлу з іменем source залишиться тим самим, оскільки ми не змінювали сам файл, а тільки видалили одне з його жорстких посилань. Кількість посилань на файл з іменем source буде дорівнювати одиниці, оскільки жорстке посилання з іменем hardlink було видалено. |
| ln -s source softlink  ls -li source softlink | Команда ln -s source softlink створює символічне посилання з іменем softlink, що вказує на файл з іменем source.  Команда ls -li source softlink виведе інформацію про два файли, що містяться у поточній директорії, у форматі, схожому на вивід команди ls -li. Індекс вузла іноди для файлу source буде однаковим як у випадку з жорстким посиланням, але він буде різним для файлу softlink, оскільки він є символічним посиланням. У кількості посилань на файл source буде відображена лише одиниця, оскільки символічні посилання не збільшують кількість посилань на файл. |
| ln -s /proc crossdir  ls -l crossdir | Команда ln -s /proc crossdir створює символічне посилання на каталог /proc з іменем crossdir.  Команда ls -l crossdir виведе інформацію про файл або каталог з іменем crossdir, що міститься у поточній директорії, у форматі, схожому на вивід команди ls -l. У виводі буде вказано, що crossdir є символічним посиланням на каталог /proc. Це означає, що будь-яка спроба звернення до файлів або каталогів через символічне посилання crossdir буде перенаправлена до відповідних файлів або каталогів у каталозі /proc. |

\*\*\*Скріншоти виконання команд в терміналі можна не представляти, достатньо коротко описати команди в таблиці.

***Готував матеріал студент Кресан Руслан***

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

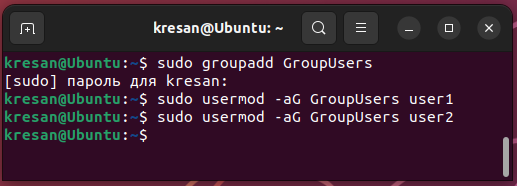
*Щоб створити трьох нових користувачів, слід виконати команду adduser. Після введення цієї команди в терміналі вам буде запропоновано ввести пароль та додаткову інформацію про користувача, якщо ви бажаєте. Після введення цієї інформації Linux створить нового користувача з вказаними параметрами.*



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

*Щоб створити нову групу користувачів і додати до неї двох з трьох створених користувачів в терміналі Linux, слід виконати наступні кроки:*

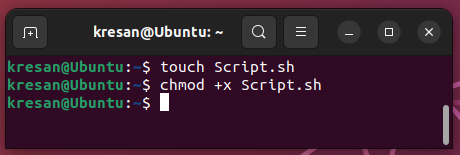
* *Створити нову групу користувачів за допомогою команди groupadd.*
* *Додати користувачів до нової групи за допомогою команди sudo usermod -aG.*



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

*Щоб створити новий файл, який буде доступний на зчитування, редагування та виконання власником файлу, наприклад, найпростіший скриптовий сценарій в терміналі Linux, слід виконати наступні кроки:*

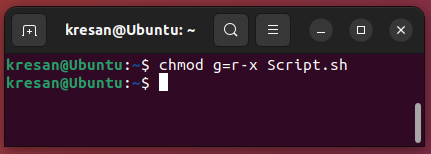
* + 1. *Створити новий файл за допомогою команди touch.*
    2. *Відкрити файл для редагування та додати вміст скрипту.*
    3. *Зберегти файл та встановити йому права на виконання за допомогою команди chmod +x.*



* + **для користувачів групи власника надайте дозволи на перегляд та виконання (без дозволу на редагування) цього файлу;**

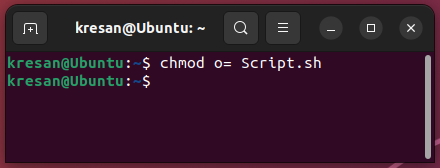
*Щоб надати дозволи на перегляд та виконання файлу для користувачів групи власника, не надаючи їм дозволу на редагування, слід виконати наступні кроки:*

1. *Виконати команду ls -l. Ця команда виведе поточні права доступу до файлу.*
2. *Замінити права доступу до файлу за допомогою команди chmod.*
3. *Встановити права доступу для користувачів групи власника файлу, не надаючи їм дозволу на редагування. Щоб цього досягти, слід встановити права r-x, які дозволять читати та виконувати файл, але не дозволять редагувати його. Для цього слід виконати команду chmod g=r-x.*



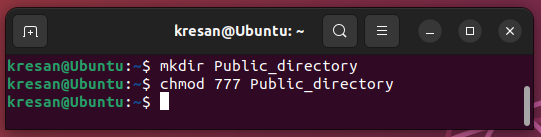
* + **для інших користувачів заборонити доступ до цього файлу;**

*Щоб заборонити доступ до файлу іншим користувачам, не належним до групи власника, слід виконати команду chmod o=, де o= встановлює права доступу для інших користувачів на немає доступу (тобто не можуть читати, записувати або виконувати файл).*

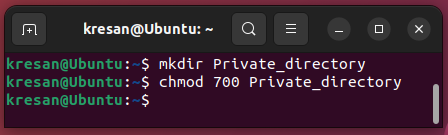


* + **подібні дії виконайте для директорій - створіть директорію, яка буде доступна для всіх трьох користувачів, створіть директорію, яку буде доступна тільки для власника, створіть директорію, яку користувачі групи власника зможуть переглядати, але не редагувати;**

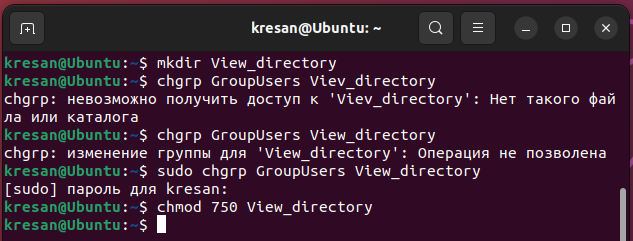
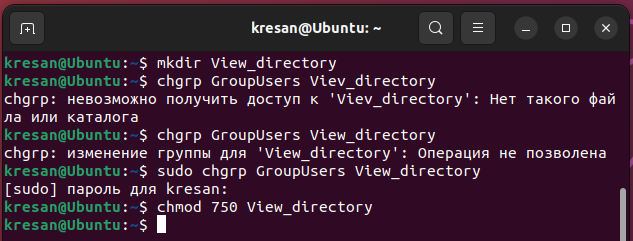
*Створення директорії, доступної для всіх трьох користувачів:*



*Створення директорії, яка доступна тільки для власника:*

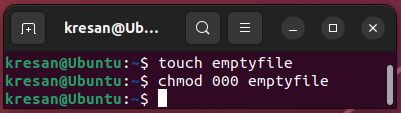


*Створення директорії, яку користувачі групи власника зможуть переглядати, але не редагувати:*

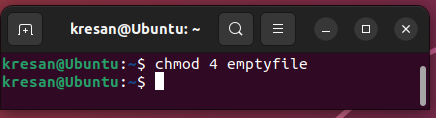


* + **створіть порожній файл під назвою emptyfile за допомогою команди touch emptyfile. Тепер “обнуліть” дозволи для файлу з chmod 000 emptyfile. Що станеться, якщо змінити дозволи для emptyfile, передавши лише одне значення для chmod у числовому режимі, наприклад, chmod 4 emptyfile? Що буде, якщо ми використаємо два числа, наприклад chmod 44 emptyfile? Що ми можемо дізнатися про те, як chmod зчитує числове значення?**

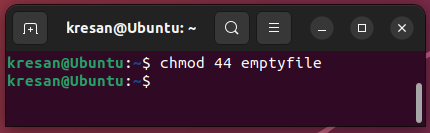
*Після виконання команди chmod 000 emptyfile всі дозволи на файл будуть обнулені, тобто будь-який користувач не зможе читати, редагувати або виконувати файл.*



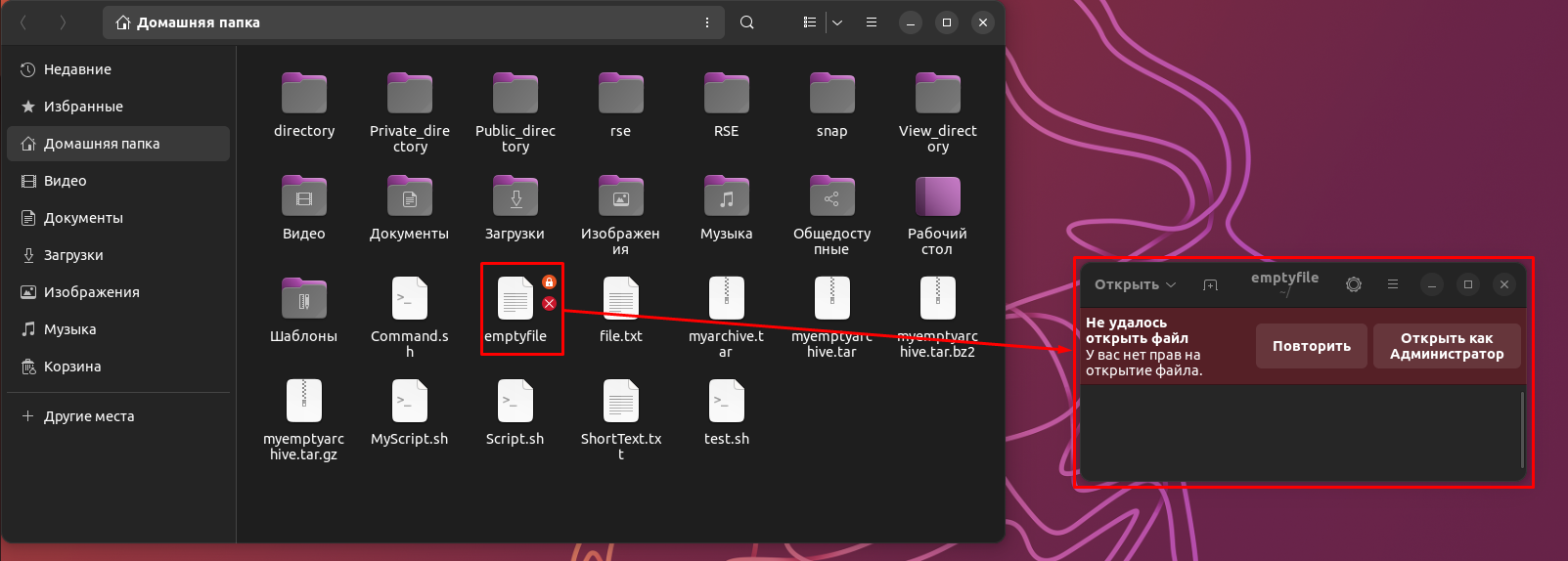
*Якщо ми виконаємо команду chmod 4 emptyfile, то ми встановимо тільки дозвіл на читання для власника файлу, тому що 4 відповідає бінарному значенню 100, що встановлює дозвіл на читання тільки для власника. Для групи та інших користувачів дозволи будуть залишатися незмінними (тобто будуть рівні нулю).*



*Якщо ми виконаємо команду chmod 44 emptyfile, то ми встановимо дозволи на читання для власника та групи файлу, а для інших користувачів дозволи будуть залишатися незмінними.*



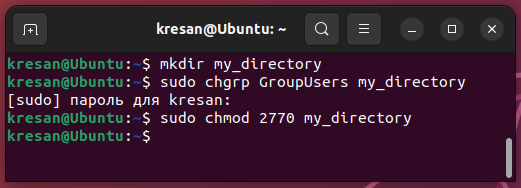
*При використанні числового режиму в chmod, ми можемо встановити різні дозволи для власника, групи та інших користувачів, використовуючи різні комбінації чисел. Кожне число відповідає бітовому значенню дозволів (4 для читання, 2 для запису, 1 для виконання), і можна додавати ці числа для встановлення різних комбінацій дозволів.*



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

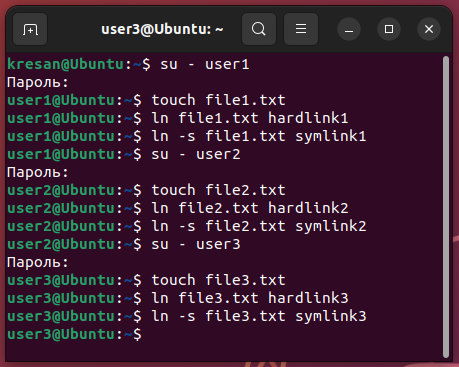
*Опція 2 встановлює SetGID-біт, що означає, що всі файли та папки, що створюються в директорії, належатимуть групі користувачів, до якої належить директорія. Опція 7 встановлює належність тільки власнику та групі користувачів, а також дозволяє їм читати, писати та виконувати файли. Опція 0 встановлює нульові дозволи для всіх інших користувачів.*

*Тепер всі файли та директорії, які створюються в директорії my\_directory, належатимуть до вашої групи користувачів, а інші користувачі не зможуть видалити файли, які були створені в цій директорії.*

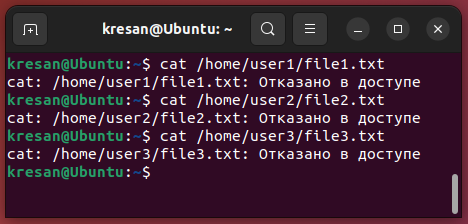


* + **під кожним користувачем створіть по одному новому файлу, та жорстке та символічне посилання на нього;**

*Для створення нових файлів та жорстких посилань на них, використаємо наступні команди:*



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



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

*Інші користувачі, які не є власниками файлів, не можуть їх видалити без відповідних дозволів. У разі, якщо файли були створені з дозволами, які дозволяють власникам групи виконувати операції з файлом, інші користувачі з цієї групи можуть мати деякі права на файл, наприклад, можуть його переглядати, але не можуть видалити без додаткових дозволів. Таким чином, власник файлу або користувач з відповідними дозволами можуть видалити ці файли, а інші користувачі не можуть цього зробити.*

***Готував матеріал студент Заїка С. В.***

**Відповіді на контрольні запитання**

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

*Символічний метод - це спосіб зміни прав доступу до файлів та директорій в Unix / Linux, який використовує символьні мнемоніки для встановлення нових прав.*

*Приклади зміни прав доступу за допомогою символічного методу:*

*Для зміни прав доступу до файла file.txt так, щоб власник міг зчитувати, редагувати і виконувати його, група могла зчитувати, а всі інші користувачі не мали доступу, виконайте наступну команду: chmod u+rwx,g+r,o-rwx file.txt*

*Для зміни прав доступу до директорії mydir так, щоб всі користувачі могли зчитувати та виконувати файли у цій директорії, але тільки власник міг редагувати файли, виконайте наступну команду: chmod u+rwx,g+rx,o+rx mydir*

*Для зміни прав доступу до файла script.sh так, щоб всі користувачі могли виконувати його, але не могли редагувати або зчитувати його, виконайте наступну команду: chmod a+x script.sh*

*У цих прикладах команда chmod використовує символьну нотацію для встановлення прав доступу до файлів та директорій. u вказує на власника файлу, g вказує на групу користувачів, o вказує на інші користувачі, а a вказує на всіх користувачів. r означає право на зчитування файлу, w - право на його редагування, а x - право на виконання файлу.*

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

*Числовий метод (або октальний метод) - це спосіб зміни прав доступу до файлів та директорій в Unix / Linux, який використовує восьмеричні числа для встановлення нових прав. Кожне право (зчитування, запис, виконання) має свій власний числовий код, а отримані коди об'єднуються в восьмеричне число, що відображає потрібний набір прав.*

*Приклади зміни прав доступу за допомогою числового методу:*

*Для зміни прав доступу до файла file.txt так, щоб власник міг зчитувати, редагувати і виконувати його, група могла зчитувати, а всі інші користувачі не мали доступу, виконайте наступну команду: chmod 750 file.txt*

*У цьому випадку число 7 відповідає за права доступу власника (зчитування, запис і виконання), число 5 відповідає за права доступу групи (зчитування та виконання), а число 0 відповідає за права доступу інших користувачів (без доступу).*

*Для зміни прав доступу до директорії mydir так, щоб всі користувачі могли зчитувати та виконувати файли у цій директорії, але тільки власник міг редагувати файли, виконайте наступну команду: chmod 755 mydir*

*У цьому випадку число 7 відповідає за права доступу власника (зчитування, запис і виконання), число 5 відповідає за права доступу групи (зчитування та виконання), а число 5 відповідає за права доступу інших користувачів (тільки зчитування та виконання).*

*Для зміни прав доступу до файла script.sh так, щоб всі користувачі могли виконувати його, але не могли редагувати або зчитувати його, виконайте наступну команду: chmod 111 script.sh*

*У цьому випадку у всіх трьох числових значеннях встановлено тільки дозвіл на виконання, що відповідає значенню 1 в октальній системі. Таким чином, тільки користувачі можуть виконувати файл script.sh, але не можуть читати або редагувати його.*

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

*Так, можна виконати файл, для якого є права на виконання, але не встановлені права на читання. Це можливо тому, що для виконання файлу не потрібно читати його вміст. При виконанні файлу, операційна система просто виконує інструкції, які зберігаються в файлі, і виводить результат на екран або в інший вихідний потік.*

*Наприклад, якщо у файлу є право на виконання, але немає права на читання, ви все ще зможете запустити цей файл як скрипт на виконання. Однак, якщо ви спробуєте прочитати вміст файлу до того, як виконати його, то отримаєте помилку від операційної системи.*

1. Яке призначення команди umask?

*Команда umask в Unix / Linux встановлює значення маски дозволів за замовчуванням для новостворених файлів та директорій в поточній сесії. Маска дозволів за замовчуванням віднімається від максимально дозволених прав доступу, тобто вона визначає, які дозволи за замовчуванням не будуть встановлюватись при створенні нових файлів та директорій.*

*Значення маски дозволів за замовчуванням можна встановити від 0 до 777 в вісімковому форматі. Кожна цифра представляє маску дозволів для відповідної групи користувачів (власник, група та інші). Наприклад, якщо маска дозволів за замовчуванням встановлена як 022, це означає, що з максимально дозволених прав доступу будуть відняті дозволи на запис для групи та інших користувачів.*

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

*Зміна прав доступу та дозволів в поточній сесії зазвичай є тимчасовою і не буде збережена після закриття сесії або перезавантаження системи.*

*Якщо ви змінюєте права доступу та дозволи в поточній сесії, то вони застосовуються до всіх файлів та директорій, з якими ви взаємодієте під час цієї сесії. Ці зміни не будуть застосовані до інших сеансів взаємодії з системою або до інших користувачів, крім випадку, якщо ви змінюєте права доступу в системних файлів або використовуєте команди, які використовують ці права як частину своєї функціональності.*

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

*Так, у більшості операційних систем є шаблон, який використовується для прав доступу при створенні нових файлів та директорій. Зазвичай, коли створюється новий файл або директорія, система використовує значення за замовчуванням, які можна налаштувати за допомогою команди umask. Значення umask встановлює бітову маску, яка віднімає певні дозволи доступу при створенні нових файлів та директорій.*

*Зазвичай значення за замовчуванням для нових файлів мають права -rw-rw-rw- або 666, що означає, що всі користувачі мають дозвіл на читання та запис, але не на виконання. Для нових директорій значення за замовчуванням зазвичай є -rwxrwxrwx або 777, що означає, що всі користувачі мають повний доступ до директорії.*

*Щоб змінити права дозволу за замовчуванням, можна встановити нове значення umask. Наприклад, якщо ви хочете встановити права доступу для нових файлів на -rw-r--r-- або 644, ви можете виконати наступну команду: umask 022*

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 тимчасових файлів у більшості операційних систем - це /tmp або /var/tmp. Ці каталоги зазвичай призначені саме для тимчасових файлів та мають правильні права доступу, щоб забезпечити безпеку та ефективність системи. Для створення тимчасового файлу можна використовувати команду mktemp, яка автоматично створює унікальний файл з випадковою назвою та правами доступу. Наприклад, команда mktemp /tmp/tempfile.XXXXXX створить тимчасовий файл зі змінною частиною XXXXXX у каталозі /tmp. Після використання тимчасового файлу його потрібно видалити за допомогою команди rm, щоб звільнити місце в системі.*

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

*Жорстке посилання (hard link) - це ім'я файлу, яке посилається на той же самий вузол файлової системи, що й початкове ім'я файлу. Файли, на які посилаються жорсткі посилання, виглядають як один файл, оскільки вони мають одинакові inode та вказують на один і той же блок даних на диску.*

*Для створення жорсткого посилання використовується команда ln з опцією -s, аргументами якої є вихідний файл та ім'я жорсткого посилання. Наприклад, щоб створити жорстке посилання file2 на файл file1 в поточному каталозі, необхідно виконати команду: ln file1 file2*

*Жорсткі посилання зазвичай використовуються в ситуаціях, коли потрібно мати доступ до одного файлу з різних імен або коли потрібно створити бекапи файлів. Вони дозволяють зменшити кількість дискового простору, оскільки вони посилаються на один і той же блок даних на диску. Також, якщо вихідний файл буде переміщено або перейменовано, жорстке посилання на нього все ще буде працювати, оскільки воно посилається на inode, а не на ім'я файлу. Однак, жорсткі посилання не можуть бути створені на іншому диску або файловій системі, і вони не можуть посилатися на каталоги.*

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

*Символічні посилання (symlink або soft link) - це особливий тип файлу, який створюється для посилання на інший файл або каталог. Він містить посилання на ім'я файлу, а не на його інодінійну структуру, і тому може посилатися на файл або каталог незалежно від його місцезнаходження в файловій системі.*

*Для створення символічного посилання використовується команда* ***ln*** *з опцією* ***-s****: ln -s /шлях/до/оригінального\_файлу /шлях/до/символічного\_посилання*

*Наприклад, якщо ми хочемо створити символічне посилання на файл* ***/home/user/Documents/report.txt*** *в каталозі* ***/home/user/Downloads****, ми можемо виконати наступну команду: ln -s /home/user/Documents/report.txt /home/user/Downloads/report-link*

*Тепер у каталозі* ***/home/user/Downloads*** *з'явиться файл з ім'ям* ***report-link****, який буде посилатися на файл* ***report.txt*** *у каталозі* ***/home/user/Documents****.*

*Символічні посилання можуть бути корисними в таких ситуаціях:*

* *При створенні посилання на довгий шлях до файлу з короткою назвою, щоб полегшити доступ до файлу.*
* *При розміщенні файлів, які часто переміщуються, на зовнішніх носіях або в мережевих каталогах, щоб забезпечити стабільний доступ до цих файлів, незалежно від їх місцезнаходження.*
* *При створенні багатьох ідентичних копій одного файлу, щоб зменшити обсяг зайнятого місця на диску.*
* *При створенні посилання на каталог, щоб мати дві різні назви для одного й того ж каталогу.*

1. Порівняйте жорсткі та символічні посилання?

*Жорстке та символічне посилання - це два способи створення посилань на файл або каталог у Linux.*

*Жорстке посилання створюється шляхом створення нового запису в файловій системі, який посилається на той самий вузол файлової системи, що й оригінальний файл. Жорстке посилання може бути використане для доступу до оригінального файлу з різних директорій, включаючи ту саму директорію, де знаходиться оригінальний файл. Він не може бути створений для каталогів і не може пересікати межі файлових систем.*

*Символічне посилання - це запис, який містить шлях до оригінального файлу. Воно створюється за допомогою команди ln з опцією -s. Символічне посилання може пересікати межі файлових систем та бути створене для каталогів. Воно може бути використане для створення зручного шляху до оригінального файлу з різних директорій.*

*Одна з основних відмінностей між жорстким та символічним посиланням полягає у тому, що якщо ви видаляєте оригінальний файл, то жорстке посилання на нього все ще буде існувати і матиме доступ до даних файлу, але якщо ви видаляєте оригінальний файл, на який посилається символічне посилання, то символічне посилання буде недійсним.*

*Жорсткі посилання короткі та ефективні, оскільки не використовують додаткової пам'яті, щоб зберігати додаткову інформацію про посилання. Символічні посилання надають більшу гнучкість та зручність, оскільки вони можуть бути використані для створення посилань на каталоги та файлові системи, які перетинаються.*

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

*При видаленні оригінального файлу, жорстке посилання продовжить існувати і буде посилатися на той самий вміст*

* + символічне посилання;

*Якщо видалити символічне посилання, то оригінальний файл залишиться на місці, але символічне посилання буде недійсним і вказуватиме на ніщо.*

* + жорстке посилання.

*Однак, якщо ви видалите жорстке посилання, то оригінальний файл і символічне посилання не будуть зберігатись на жорсткому диску, і будуть недійсними.*

***Готував матеріал студент Губенко Є.О***

**Висновки**

Під час роботи з командною оболонкою Bash, користувач може використовувати різноманітні команди для роботи з файлами та каталогами в операційній системі Linux. Знання команд для роботи з власниками та правами доступу до файлів є важливим, оскільки це дозволяє контролювати, які користувачі мають доступ до певних файлів та каталогів, і що вони можуть з ними робити.