“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №10**

з дисципліни: «Операційні системи»

**Тема: “Спеціальні каталоги та файли в Linux”**

Виконав студент

групи РПЗ-93а

Бровченко Р.А.

Перевірив викладач

Сушанова В.С.

Київ 2022

**1. Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.

2. 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.**

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

|  |  |
| --- | --- |
| Термін | Призначення |
| **setuid permission** | Permission when the binary file is run as the owner of the file, not as the user who executed it. |
| **chmod u+s file** | To add the setuid permission symbolically |
| **chmod u-s file** | To remove the setuid permission symbolically |
| ***setgid* permission** | The *setgid* permission is similar to setuid, but it makes use of the group owner permissions. |
| ***chmod g+s <file|directory>*** | To add the setgid permission symbolically |
| ***chmod g-s <file|directory>*** | To remove the setgid permission symbolically |
| ***sticky bit*** | *Sticky bit* permission is used to prevent other users from deleting files that they do not own in a shared directory. |
| ***links*** | link |
| ***inode number*.** | A unique identification number of file. |
| **symbolic link** | A *soft link*, is simply a file that points to another file. |

2. На базі розглянутого матеріалу дайте відповіді на наступні питання:

2.1. Для чого використовуються дозволи Setuid та Setgid?

Коли дозвіл **setuid** встановлено для виконуваного двійкового файлу (програми), двійковий файл запускається як власник файлу, а не як користувач, який його виконав. Цей дозвіл встановлюється на кількох системних утилітах, щоб їх могли запускати звичайні користувачі, але виконуватися з правами root, забезпечуючи доступ до системних файлів, до яких звичайний користувач зазвичай не має доступу.

Дозвіл **setgid** на файл дуже схожий на setuid. Він дозволяє користувачеві запускати виконуваний двійковий файл у спосіб, який надає їм додатковий (тимчасовий) груповий доступ. Система дозволяє користувачеві, який виконує команду, фактично належати до групи, якій належить файл, але тільки в програмі setgid.

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

Sticky bit – це право для захисту файлів від випадкового видалення в середовищі, де кілька користувачів мають права на запис в той самий каталог. Якщо застосовується закріплений sticky bit, користувач може видалити файл тільки якщо він є користувачем-власником файлу або каталогу, в якому міститься файл. З цієї причини він застосовується як роздільна здатність для каталогу /tmp і може бути корисний також для каталогів спільних груп.

**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 18: Special Directories and Files

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

|  |  |
| --- | --- |
| **Назва команди** | **Її призначення та функціональність** |
| ls -ld /var/tmp | Використання параметра -d для команди ls відображає інформацію про каталог; у поєднанні з параметром -l він показує право власності та дозволи на файли каталогу. |
| ln | Створення жорсткого посилання |
| rm | Видалення файлу |
| ln -s source softlink | Параметр -s для команди ln створює символьне посилання замість жорсткого посилання. |

**Відповіді на контрольні запитання:**

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

*Для створення жорстких посилань використовується команда ln з двома аргументами. Першим аргументом є ім’я існуючого файлу, з яким потрібно зв’язатися, яке називається цільовим, а другий аргумент – це нове ім’я файлу для посилання на ціль.*

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

*Інший тип посилань, який можна створити, відомий як символічне посилання або програмне посилання. Символічні посилання не збільшують кількість посилань файлів, з якими вони пов’язані.*

*Файли символічних посилань мають власний індекс і тип файлу. Замість того, щоб зв’язувати і надавати спільний номер inode, вони посилаються на ім’я файлу. На відміну від жорстких посилань, м'які посилання можуть бути пов'язані з каталогами і можуть перетинати пристрої та розділи до своїх цілей.*

**3. Порівняйте жорсткі та символічні посилання?**

|  |  |  |
| --- | --- | --- |
| **Основа для порівняння** | **Тверде посилання** | **М'яке посилання** |
| Основні | Доступ до файлу можна отримати через безліч різних імен, відомих як жорсткі посилання. | Доступ до файлу можна отримати через різні посилання, що вказують на цей файл, відомий як м'яке посилання |
| Перевірка посилання, коли вихідний файл видалений | Доступ до файлу може бути доступним. | Недійсний |
| Команда, що використовується для створення | ln | ln -s |
| номер inode | Те саме | Інший |
| Можна пов’язати | До власної перегородки. | До будь-якої іншої файлової системи, навіть мережевої. |
| Споживання пам'яті | Менше | Більше |
| Відносний шлях | Не застосовується | Дозволено |

**4. 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.**

*Якщо у вас є п’ять файлів, жорстко пов’язаних між собою, то видалення будь-яких чотирьох із цих файлів не призведе до видалення фактичного вмісту файлу. Іноді може бути важко дізнатися, де існують жорсткі посилання на файл.*

*Якщо ви бачите звичайний файл із кількістю посилань, більшим за одиницю, ви можете скористатися командою find із критерієм пошуку -inum, щоб знайти інші файли з таким самим номером індексу.*

*Якщо вихідний файл видалено, усі файли, пов’язані з ним, не працюватимуть. М’які посилання легше побачити. М’які посилання є набагато візуальнішими, не вимагаючи жодних додаткових команд, крім команди ls, щоб визначити посилання.*

**Висновок:** під час виконання лабораторної роботи, я ознайомився з спеціальними каталогами та файлами в Linux.