“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?

The id command is used to display information about the user ID (UID - User ID) and group (GID - Group ID). It displays information about the current system user, such as User ID (UID), username, Principal Group Identifier (GID), principal group name, and a list of additional groups to which the user belongs.

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

To do this, use the ls command with the –l option.

* 1. Як змінити власника групи?

The chown (change owner) command is used to change the owner of a file group or directory. To change the owner, use the following command:

sudo chown :new\_group file\_or\_directory

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

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

Examples:

1. file example.txt – text file;
2. file mydirectory – directory;
3. file myarchive.zip - ZIP archive;
4. file mydocument.pdf - PDF document;
5. file myimage.png - graphic file (PNG image).
   1. Для чого використовуються дозволи Setuid та Setgid?

Setuid and Setgid permissions are used to grant additional execution privileges to executable files.

* • When Setuid is set on an executable file, it runs in the context of the file's owner instead of the user who launched it. This allows you to perform some operations with elevated privileges. It is usually installed on programs that require elevated privileges to perform specific tasks.
* • When Setgid is set on an executable file or directory, it runs in the context of the group that owns the file or directory instead of the group of the user who launched it. It allows sharing of files and resources between users belonging to the same group. Often installed on directories shared by a group of users who need to share files in that directory. This helps avoid problems with changing the owner of a file to another group.
  1. Для чого в системі потрібен так званий “липкий біт” (Sticky Bit). Наведіть приклади коли цей дозвіл доцільно використовувати.

This is a special permission in the file system that has several application options.

The main purpose of the sticky bit is to restrict the ability to delete or rename files in a directory to users who do not own the files or the directory itself. The sticky bit can also be used to create temporary files. In folders with the sticky bit set, any user can create files, but can only delete or modify files that he himself created. When the sticky bit is set on a directory, it allows users to modify only their own files in that directory, even if they have write access to it.

Example:

The /tmp directory is one of the typical places where the sticky bit is applied. Users can create files and directories in /tmp, but they can only delete files or directories that they themselves have created. This provides some balance between user access to temporary files and prevents accidental deletion or modification of files that may be important to other users or to the system.

**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

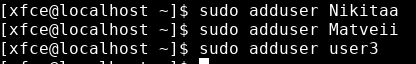
- Lab 18: Special Directories and Files

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

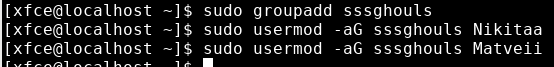
|  |  |
| --- | --- |
| NDG Linux Essentials - Lab 17 | |
| Command name | Its purpose and functionality |
| cd /tmp | Change to the /tmp directory. |
| mkdir priv-dir pub-dir | Create two directories named priv-dir and pub-dir. |
| touch priv-dir/priv-file<br>touch pub-dir/pub-file | Create two files: priv-file in priv-dir and pub-file in pub-dir. |
| ls -l priv-dir<br>ls -l pub-dir | Display details (long format) of directories priv-dir and pub-dir. |
| ls -ld priv-dir<br>ls -l priv-dir/priv-file | Display details (long format) for priv-dir and priv-file. |
| chmod o-rx priv-dir/ | Deny read and execute access for others to the priv-dir directory. |
| ls -ld priv-dir/ | Check changes in permissions for the priv-dir directory. |
| chmod a+x file<br>chmod g-w file<br>chmod go+r file<br>chmod o=rwx | Change file permissions using the symbolic method. |
| ls -ld pub-dir/<br>chmod o+w pub-dir/<br>ls -ld pub-dir/ | Allow write access for others to the pub-dir directory. |
| ls -l priv-dir/priv-file<br>chmod g-rw,o-r priv-dir/priv-file<br>ls -l priv-dir/priv-file | Deny access for the group and others to the priv-file file. |
| ls -l pub-dir/pub-file<br>chmod a=rw pub-dir/pub-file<br>ls -l pub-dir/pub-file | Grant read and write permission to everyone for the pub-file file. |
| echo "date" > test.sh | Create a file test.sh with the content "date". |
| ./test.sh | Attempt to execute the test.sh file, resulting in an error. |
| chmod u+x test.sh<br>./test.sh | Grant execute permission to the user and execute the test.sh file. |
| NDG Linux Essentials - Lab 18 | |
| ls -ld /tmp | View details of the /tmp directory. |
| ls -ld /var/tmp | View details of the /var/tmp directory. |
| ls -l /etc/shadow | View access rights for the /etc/shadow file. |
| ls -l /usr/bin/passwd | View access rights for the /usr/bin/passwd file. |
| ls -l /usr/bin/wall | View access rights for the /usr/bin/wall file. |
| echo "data" > source | Create a file named source with the content "data". |
| ls -li source | View information about the inode and details of the source file. |
| ln source hardlink | Create a hard link to the source file. |
| ls -li source hardlink | View information about the inode and details of the source and hardlink files. |
| ln hardlink hardlinktwo | Create another hard link to the source file. |
| ls -li hardlink hardlinktwo source | View information about the inode and details of the source, hardlink, and hardlinktwo files. |
| rm hardlinktwo | Remove one of the hard links (hardlinktwo). |
| ls -li source hardlink | View information about the inode and details of the source and hardlink files. |
| rm hardlink | Remove the other hard link (hardlink). |
| ls -li source | View information about the inode and details of the source file. |
| ln -s source softlink | Create a symbolic link (softlink) to the source file. |
| ls -li source softlink | View information about the inode and details of the source and softlink files. |
| ln -s /proc crossdir | Create a symbolic link (crossdir) to the /proc directory. |
| ls -l crossdir | View details of the symbolic link crossdir. |

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;







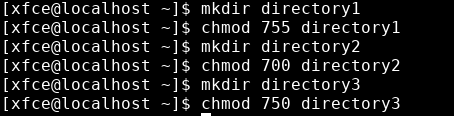
- для користувачів групи власника надайте дозволи на перегляд та виконання (без дозволу на редагування) цього файлу;



- 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?



If we use chmod 4 empyfile:

* In our case, the number 4 indicates the right to read for the owner of the file. However, if all permissions on the file were previously reset (chmod 000), then any attempt to set the permissions on the file will fail. So chmod 4 emptyfile in this case won't change anything in the file's permissions because all permissions are currently set to none.

If we use chmod 44 emptyfile:

* The first number indicates the read permission for the owner of the file, and the second number also indicates the read permission for the group to which the file belongs. But since I put chmod 000 before it, the command chmod 44 emptyfle won't change anything.

The chmod command in numeric mode uses three numbers to set permissions on a file or directory. These numbers represent the combination of access rights for the owner, group, and other users, in that order.

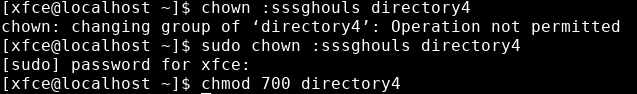
Each digit in numeric mode indicates certain access rights:

* The first number is responsible for the access rights for the owner of the file.
* The second number is responsible for the access rights for the group to which the file belongs.
* The third number indicates access rights for other users.

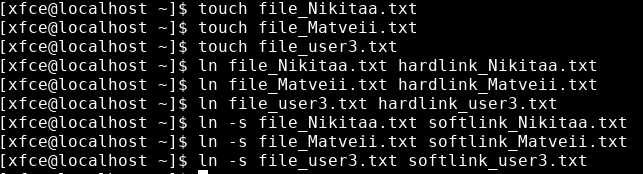
Each digit can have a value from 0 to 7, which is represented as a combination of three bits.

* The number 0 indicates that a certain access right is absent.
* The number 1 indicates the right of execution.
* The number 2 indicates the write right.
* The number 3 indicates write and execute rights.
* The number 4 indicates the right to read.
* The number 5 indicates read and execute rights.
* The number 6 indicates read and write rights.
* The number 7 indicates read, write and execute rights.

- 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;

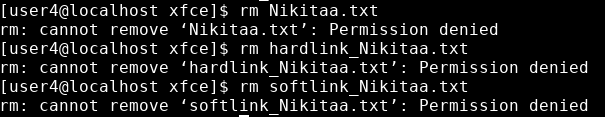


- спробуйте іншими користувачами переглянути ці файли;





- спробуйте іншими користувачами видалити ці файли, зробіть висновки.



Conclusion:

Access and ability to delete files by other users depends on the settings of access rights to these files and the rights of users trying to delete these files.

**Відповіді на контрольні запитання**

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

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

Examples:

1. Adding access rights:

* Adding read permission for a group to a file example.txt:

chmod g+r example.txt

* Adding execute rights for all users to the directory my\_directory:

chmod a+x my\_directory

* • Adding write rights for the owner and group to the data.csv file:

chmod ug+w data.csv

1. Removal of access rights:

* Removing the execute right for other users from the script.sh file:

chmod o-x script.sh

* Removing the read right for the group and other users from the documents directory:

chmod go-r documents

1. Change of specific access rights:

* Changing the access rights of the config.ini file, allowing the execution owner, group to read other entries:

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

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

Examples:

1. Adding access rights:

* Adding read permission for the group to the example.txt file:

chmod g+r example.txt

* This example can be converted to a numerical method as:

chmod 640 example.txt

In this case:

* 6 sets the rights for the owner: read (4) + write (2) = 6.
* 4 sets the rights for the group: read (4).
* 0 sets no rights for other users.

1. Removal of access rights:

* Removing the right to execute (execute) for all users from the directory my\_directory:

chmod a-x my\_directory

* This can be done numerically as follows:

chmod 644 my\_directory

In this case:

* 6 sets the rights for the owner: read (4) + write (2) = 6.
* 4 sets the rights for the group: read (4).
* 4 sets the rights for other users: read (4).

1. Change of specific access rights:

* Changing the access rights of the `script.sh` file, allowing the execution owner, the read group, and others to write:

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

* This can be done numerically as follows:

chmod 754 script.sh

In this case:

* 7 sets the rights for the owner: read (4) + write (2) + execute (1) = 7.
* 5 sets the rights for the group: read (4) + execute (1) = 5.
* 4 sets the rights for other users: read (4).

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

Поясніть.

Yes, it is possible to execute the file. Read permissions are not required to execute a file if execute permissions are set for the user or group to which the user trying to execute the file belongs.

1. Яке призначення команди umask?

The umask command sets or unmasks the default access mode for newly created files and directories in the current shell environment. The access mode mask (UMask) specifies the permissions that are automatically subtracted from the access mode granted to a new file or directory when it is created. The main function of umask is to control the standard access rights applied to new files and directories created by the user in the operating system.

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

They will not be saved.

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

Yes, there is a template that is used to set default permissions when creating new files and directories. This template is called umask. To change the default access rights for new files and directories, you can use the umask command in the shell to set the access mode mask.

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

To create a temporary file that will never be needed again after closing the program, there is a special directory designed for such purposes - it is a directory for temporary files (Temporary directory). in normal cases, this directory is usually under the path /tmp. If an application needs to create a one-time temporary file that will not be used again after the application is closed, you can create this file in the /tmp directory for temporary files.

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

To create a hard link, use the ln command.

ln /start\_file\_path /new\_name\_for\_link

The main feature of hard links is that they refer to the same file node as the original file, meaning that the hard link and the original file share the same block of data on disk.

Situations in which it is advisable to use them:

1. Providing access to one file from different places in the file systemі.

They allow you to have multiple paths to the same file. For example, you can create a link in a folder that is convenient for you and have access to the same file from another location.

1. Saving disk space.

If you have a large number of identical files or want to save disk space, you can create hard links instead of creating multiple copies of files.

1. Ensuring that a file is preserved after it is moved or renamed.

If you've created a hard link to a file, you can change or move the original file and the link will remain active.

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

To create a symbolic link, use the ln command.

ln -s /path\_to\_target\_file\_or\_directory /new\_name\_to\_link

Situations in which it is advisable to use them:

1) Creating links to files or directories elsewhere in the file system.

They allow you to link to a file or directory from anywhere in the file system, including other file systems or network resources.

2) Simplifying access to files and directories.

They provide the ability to create short or convenient links to long or complex file system paths.

3) Work with symlinks in scripts or programs.

Symbolic links are very useful when writing scripts or programs because they allow you to conveniently refer to different files and directories.

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

- original file;

If you delete the original file that was referenced, it will delete the actual data that the original file pointed to. If this was the only file with this data, it will be lost, even if you have a link to this data.

- symbolic link;

Deleting a symbolic link will not affect the data in the original file or in other links. A symbolic link is simply a path to the original file, not the data itself. When you delete a symbolic link, you simply lose the connection to the original file, but the file itself remains in place.

- hard link.

A hard link refers to the same data as the original file. If you delete a hard link, it will not delete the actual data that the original file or other links pointed to. Actual data will be stored on disk as long as there is at least one reference to that data. If you remove all references to this data, it will no longer be available, even though the data itself is still stored on disk. They will be removed after the last link to them is removed.

**Висновки**

We got practical skills of working with the Bash command shell, got acquainted with basic actions when changing file owners, got acquainted with basic actions when changing file access rights, got acquainted with special directories and files in Linux.