

“Київський фаховий коледж зв’язку”
Циклова комісія Комп’ютерної та програмної 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 РПЗ-03
**Кошіль Владислав та
Фещенко Євгеній.**
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 **Кошіль Владислав**.

Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеликий словник базових англійських термінів з питань класифікації віртуальних середовищ.

Термін англійською	Термін українською
Interpreter	Інтерпретатор
Multithreading	Многопоточність
Encryption	Шифрування
Source code	Програмний код

Дайте відповіді на наступні питання:

Готував матеріал студент **Фещенко Євгеній**.

1. Яке призначення команди id?

The 'id' command is used to display the user and group IDs of the current user or specified user. It prints information about the user and group of the specified username, or if no username is given, it displays information about the current user. Specifically, it shows the user's UID (user ID), GID (primary group ID), and supplementary group IDs. This command is useful for checking the current user's permissions and access levels.

2. Як переглянути які права доступу має власник файлу?

To see what access rights the file owner has, you can use the 'ls' command with the '-l' option to show the long format listing of the file. The file owner's access rights are displayed in the first column of the output, which consists of ten characters. The first character indicates the file type, while the remaining nine characters represent the permissions for the file owner, the group owner, and other users.

3. Як змінити власника групи?

You can change the group owner of a file or directory using the 'chgrp' command followed by the new group name and the file/directory name

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

In the terminal, you can use the `file` command to see the type of a file. Here are some examples:

- To see the type of a text file called "example.txt": `file example.txt`. The output might be something like "ASCII text".
- To see the type of a binary file called "program": `file program`. The output might be something like "ELF 64-bit LSB executable".
- To see the type of a compressed file called "archive.zip": `file archive.zip`. The output might be something like "Zip archive data".

5. Для чого використовуються дозволи Setuid та Setgid?

The Setuid and Setgid permissions are used to allow users to execute a file with the permissions of the file owner or group owner, respectively.

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

The "Sticky Bit" permission is a special permission bit in Unix-based systems that is used to restrict the deletion of files in a directory. When the sticky bit is set on a directory, only the owner of the file, the owner of the directory, or the root user can delete or rename the file.

To set the sticky bit on a directory, you can use the chmod command with the "+t" option, followed by the name of the directory.

Хід роботи

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:

Готував матеріал студент **Кошіль Владислав**.

- Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її.
- Запустіть віртуальну машину Ubuntu_PC
- Запустіть свою операційну систему сімейства Linux

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

Готував матеріал студент **Фещенко Євгеній**.

Назва команди	Її призначення та функціональність
chmod	Changes the permissions of files and directories in Linux.
chown	Changes the owner of a file or directory in Linux.
chgrp	Changes the group of a file or directory in Linux.

-d	Sets the default permissions for new directories in Linux.
-l	Lists the contents of a directory in long format in Linux.
-ld	Shows the permissions of a directory in Linux.

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

Готував матеріал студент **Кошіль Владислав**.

- створіть трьох нових користувачів;

```

Adding new group `user2' (1002) ...
Adding new user `user2' (1002) with group `user2' ...
The home directory `/home/user2' already exists. Not copying from `/etc/skel'.
adduser: Warning: The home directory `/home/user2' does not belong to the user you are currently creating.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for user2
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
quadfordt@Ubuntu:~$ sudo adduser user3
Adding user `user3' ...
Adding new group `user3' (1003) ...
Adding new user `user3' (1003) with group `user3' ...
The home directory `/home/user3' already exists. Not copying from `/etc/skel'.
adduser: Warning: The home directory `/home/user3' does not belong to the user you are currently creating.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for user3
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
quadfordt@Ubuntu:~$ _

```

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

```

quadfordt@Ubuntu:~$ sudo groupadd gruppa
quadfordt@Ubuntu:~$ sudo usermod -a -G gruppa user1
quadfordt@Ubuntu:~$ sudo usermod -a -G gruppa user2

```

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

nano script.sh

```

GNU nano 6.4 script.sh
echo "Hello, world!"

```

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

- для інших користувачів заборонити доступ до цього файлу;

```
quadfordt@Ubuntu:~$ sudo chmod 0 script.sh
quadfordt@Ubuntu:~$ _
```

- 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;

```
quadfordt@Ubuntu:~$ sudo chmod 0 script.sh
quadfordt@Ubuntu:~$ mkdir shared_directory
quadfordt@Ubuntu:~$ chmod 777 shared_directory
quadfordt@Ubuntu:~$ mkdir private_directory
quadfordt@Ubuntu:~$ chmod 777 private_directory
quadfordt@Ubuntu:~$ mkdir group_directory
quadfordt@Ubuntu:~$ chmod 777 group_directory
quadfordt@Ubuntu:~$
```

- створіть порожній файл під назвою `emptyfile` за допомогою команди `touch emptyfile`. Тепер “обнулите” 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?

```
quadfordt@Ubuntu:~$ touch emptyfile
quadfordt@Ubuntu:~$
```

```
quadfordt@Ubuntu:~$ chmod 000 emptyfile
quadfordt@Ubuntu:~$
```

If we change the permissions for `emptyfile` by passing only one value for `chmod` in numeric mode, for example, **`chmod 4 emptyfile`**, it means that read permission is set only for the owner of the file. The second and third numbers in this case are zeros, which means that all users will be denied write and execute permission. The minus sign before the first decimal number means that it is a file, not a directory. In this case, no user can write to or execute this file. If we use the **`ls -l emptyfile`** command, we can see the new permissions:

```
quadfordt@Ubuntu:~$ ls -l emptyfile
-----r-- 1 quadfordt quadfordt 0  3 15:41 emptyfile
quadfordt@Ubuntu:~$
```

Using two numbers for the `chmod` command, such as **`chmod 44 emptyfile`**, means that the owner and user group have read and write permission to the file. Other users do not have read and write permission. In this case, the owner and the user group have read and write permissions on the file, but other users do not have write and execute permissions. If we run the **`ls -l emptyfile`** command, we can see the new permissions:

```
quadfordt@Ubuntu:~$ ls -l emptyfile
----r--r-- 1 quadfordt quadfordt 0  3 15:41 emptyfile
quadfordt@Ubuntu:~$ _
```

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

```
quadfordt@Ubuntu:~$ mkdir -m 700 mydir
```

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

```
quadfordt@Ubuntu:~$ touch /home/quadfordt/user1_file.txt
quadfordt@Ubuntu:~$ touch /home/quadfordt/user2_file.txt
quadfordt@Ubuntu:~$ touch /home/quadfordt/user3_file.txt
```

```
quadfordt@Ubuntu:~$ ln /home/quadfordt/user1_file.txt /home/quadfordt/user1_hardlink
quadfordt@Ubuntu:~$ ln -s /home/quadfordt/user1_file.txt /home/quadfordt/user1_hardlink
ln: failed to create symbolic link '/home/quadfordt/user1_hardlink': File exists
quadfordt@Ubuntu:~$ ln -s /home/quadfordt/user1_file.txt /home/quadfordt/user1_symlink
quadfordt@Ubuntu:~$ ln /home/quadfordt/user1_file.txt /home/quadfordt/user2_hardlink
quadfordt@Ubuntu:~$ ln -s /home/quadfordt/user1_file.txt /home/quadfordt/user2_symlink
quadfordt@Ubuntu:~$ ln /home/quadfordt/user1_file.txt /home/quadfordt/user3_hardlink
quadfordt@Ubuntu:~$ ln -s /home/quadfordt/user1_file.txt /home/quadfordt/user4_symlink
quadfordt@Ubuntu:~$ ln -s /home/quadfordt/user1_file.txt /home/quadfordt/user3_symlink
quadfordt@Ubuntu:~$
```

- спробуйте іншими користувачами переглянути ці файли;
- спробуйте іншими користувачами видалити ці файли, зробіть висновки.

Відповіді на контрольні запитання

Готував матеріал студент *Фещенко Євгеній та Кошіль Владислав.*

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

Here are some examples of changing access rights using the Symbolic Method:

1. Give read, write, and execute permission to the owner of the file:

```
chmod u+rw filename
```

2. Remove write permission from the group and others for a file:

```
chmod go-w filename
```

3. Give execute permission to the group and others for a directory:

```
chmod go+x directoryname
```

4. Give read and write permission to the owner, and only read permission to the group and others for a file:

```
chmod u+rw,g+r,o+r filename
```

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

Some examples of changing access rights using the numeric method are:

- To set the owner's permissions to read, write, and execute, and the group's and everyone else's permissions to read and execute: `chmod 755 filename`
- To set the owner's permissions to read and write, and the group's and everyone else's permissions to read only: `chmod 644 filename`
- To set the owner's permissions to read, write, and execute, and the group's and everyone else's permissions to no access: `chmod 700 filename`

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

Yes, it is possible to execute a file for which you have runtime permissions but no read permissions (--x). This is because the execute permission (--x) allows a user to run a file, but not to read or modify it. However, in order to execute the file, you must know its name and the directory it is in, which requires read permission on the directory.

4. Яке призначення команди umask?

The umask command is used to set the default file permission mask for newly created files and directories. It is a three-digit octal number that specifies the permissions that will be removed from the default permissions when a new file or directory is created.

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

No, changes made to access rights and permissions in the current session do not persist to the next session unless they are explicitly written to disk or saved in a configuration file. When the system is restarted or the user logs out and back in again, the default permissions will be applied again. To make changes that persist across sessions, the user must modify the relevant configuration files or system settings.

6. 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 called umask that the system uses for default permissions when creating new files. The umask value is a bitmask that specifies which permissions should be turned off by default when a new file is created.

To change the default permissions, you can use the umask command followed by the desired umask value. For example, to set the default permissions to allow read, write, and execute for the owner and read and execute for group and others.

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

The correct directory to create a one-time temporary file that will never be needed again after the program is closed is the `/tmp` directory. The `/tmp` directory is a standard directory used for storing temporary files in Unix and Unix-like operating systems. It is usually cleared on system startup, so any files stored there will only persist for the duration of the current session.

To create a temporary file in the `/tmp` directory, the program can use the `mktemp` command or create a file with a unique name using a random number or process ID, and then delete the file when it is no longer needed.

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

To create a hard link, you can use the `ln` command with the `-n` option.

Hard links can be useful when you need to maintain multiple copies of the same file, while only actually storing one physical copy on disk. Hard links also provide a way to create a backup of a file, since any changes to the original file will also be reflected in the hard link.

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

To create a symbolic link, the `ln` command is used with the `-s` option, followed by the original file and the desired name and location of the symbolic link.

Symbolic links are useful when a file needs to be accessed from multiple locations or when a file needs to be moved to a different location without breaking any dependencies. They also make it easy to create shortcuts

to frequently used files or directories.

10. Порівняйте жорсткі та символічні посилання?

Hard links:

- Point to the same physical location on the disk as the original file.
- Have the same inode number as the original file.
- Cannot be created for directories or files on different file systems.
- Cannot point to a file or directory that has been deleted.
- Changes to the original file will affect all hard links to that file.
- Cannot be used to link to a file on a different file system.

Symbolic links:

- Point to the path name of the original file.
- Have a different inode number from the original file.
- Can be created for directories or files on different file systems.
- Can point to a file or directory that has been deleted.
- Changes to the original file will affect the symbolic link to that file.
- Can be used to link to a file on a different file system.

11. 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.**

If you delete the original file, both the symbolic link and the hard link will become invalid, as they both reference the inode of the original file. The symbolic link will still exist as a file in the filesystem, but will no longer point to anything. The hard link, however, will still be a valid file and will contain the same data as before, but it will no longer be possible to access it because the file name and directory entry for the original file no longer exist.

If you delete the symbolic link, it will simply be removed from the filesystem. The original file and the hard link will not be affected.

If you delete the hard link, the original file will still exist as long as there are other links pointing to it. However, the hard link itself will be removed from the filesystem. The symbolic link will not be affected.

Висновок: During this lab, we gained practical skills in working with the Bash command shell; learned the basic steps for changing file owners; learned the basic steps for changing file permissions; and learned about special directories and files in Linux.