“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?

*The id command is used in operating systems such as Linux and Unix to display information about user and group identifiers.*

*Usually, when you run the id command without arguments on the command line, it returns information about your current user ID (UID), group ID (GID), and a list of other groups you belong to. This information can be useful for determining access rights to files and directories.*

*You can also specify another user or group as an argument to the id command to get information about their UIDs. For example, the id someuser command returns the user ID and group information for the user named someuser.*

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

*To view the permissions of a file, you need to use the ls command with the -l argument, which indicates that detailed information about each file in the current directory is displayed. In addition, you need to specify the name of the file whose permissions you want to view.*

* 1. Як змінити власника групи?

*To change the group owner of a file on Linux or Unix, you can use the chown command with the -g option, which allows you to change the group of the file owner.*

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

*To view the type of the current file in the terminal, you can use the file command. It returns information about the file type according to its contents, not just its name extension.*

* 1. Для чого використовуються дозволи Setuid та Setgid?

*The Setuid (SUID) permission allows you to run an executable file in the context of the file owner, rather than in the context of the user who runs it. This allows users to gain additional privileges that are necessary to perform certain tasks. For example, the passwd command has Setuid permission, so a user can change his password even if he does not have write permission to the /etc/shadow file where passwords are stored.*

*Setgid (SGID) permission allows you to set the owner group for executable files and directories. When SGID permission is set on a directory, all new files and directories created in that directory will have the same owner group as the directory. This allows a group of users to work together on files and directories without allowing other users in the external environment to access those files and directories.*

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

*Sticky Bit is a special permission in the Unix/Linux operating system that can be set for directories. If the Sticky Bit permission is set for a directory, only the owner of the file can delete or rename it.*

*There are several scenarios for using the Sticky Bit permission:*

*Ensuring security in the file structure system: The Sticky Bit permission allows you to ensure data security and prevent accidental deletion. For example, the /tmp directory is set to Sticky Bit permission to ensure that users who create temporary files keep their data secure. This ensures that other users cannot accidentally delete or modify files that do not belong to them.*

*Sharing a directory: When a directory is set to Sticky Bit permission, it allows a group of users to work together on files and directories without allowing other users in the external environment to access those files and directories. This can be useful in cases where users have limited data access rights and need to share shared resources.*

*Ensuring data privacy: If you set the Sticky Bit permission in a directory with sensitive data, you can ensure data privacy and prevent accidental deletion.*

**Хід роботи**

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

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

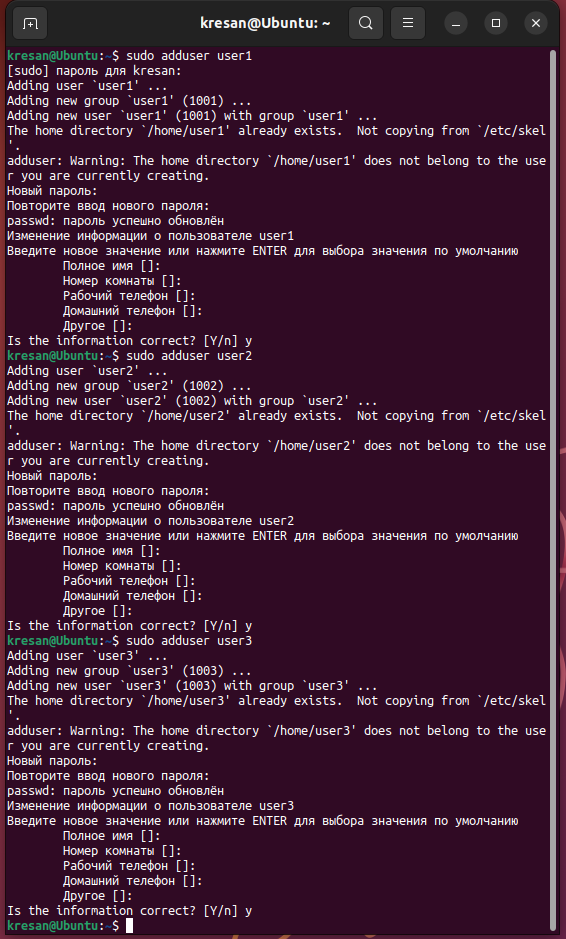
| **Назва команди** | **Її призначення та функціональність** |
| --- | --- |
| cd /tmp | The "cd /tmp" command allows you to change to the /tmp directory to access these temporary files and data stored there. Also, changing the current directory to /tmp can be useful for running some commands that require access to this directory. |
| mkdir priv-dir pub-dir | the "mkdir priv-dir pub-dir" command allows you to create two new directories with the appropriate names and use them to store private and public information in the operating system. |
| touch priv-dir/priv-file  touch pub-dir/pub-file | The "touch priv-dir/priv-file" command creates a new file named "priv-file" in the "priv-dir" directory. This file can be used to store confidential or private information that needs to be protected from unauthorized access.  the command "touch pub-dir/pub-file" will create a new file named "pub-file" in the "pub-dir" directory. This file can be used to store public information that can be accessed by different users. |
| ls -l priv-dir  ls -l pub-dir | the command "ls -l priv-dir" displays a list of files and directories that are located in the "priv-dir" directory along with additional information about each file or directory.  the command "ls -l pub-dir" will display a list of files and directories that are located in the "pub-dir" directory along with additional information about each file or directory. |
| ls -la | the "ls -la" command displays a list of all files and directories that are in the current directory, including hidden files, along with additional information about each file or directory. |
| ls -ld priv-dir/  chmod o-rx priv-dir/ | the command "ls -ld priv-dir/" displays information about the directory "priv-dir", including access rights, owner and other information, but not including a list of files that are in this directory.  The "chmod" command is used to change access rights to files and directories. The "o-rx" argument indicates that "other" users (i.e., users who are not the owner or group member of the file) are not allowed to read and execute. The command "chmod o-rx priv-dir/" will remove read and execute permissions for "other" users (i.e., those who are not owners or group members) for the directory "priv-dir". |
| ls -ld priv-dir/ | The "ls -ld priv-dir/" command displays information about the "priv-dir" directory, including access rights, owner, and group. The "-d" argument indicates that you want to display information about the "priv-dir" directory, not its contents. |
| ls -ld pub-dir/  chmod o+w pub-dir/  ls -ld pub-dir/ | The "ls -ld pub-dir/" command displays information about the "pub-dir" directory, including access rights, owner, and group. The "-d" argument indicates that you want to display information about the "pub-dir" directory, not its contents.  The "chmod" command is used to change access rights to files and directories. The "o+w" argument indicates that "other" users (i.e., users who are not owners or members of the file group).  the command "chmod o+w pub-dir/" will add write permissions for "other" users to the "pub-dir" directory. |
| ls -l priv-dir/priv-file  chmod g-rw,o-r priv dir/priv-file  ls -l priv-dir/priv-file | The command "ls -l priv-dir/priv-file" displays information about the file "priv-file", which is located in the directory "priv-dir".  The "chmod" command is used to change access rights to files and directories. The argument "g-rw,o-r" indicates the removal of read and write permissions for the group and "other" users of the file, respectively.  the command "chmod g-rw,o-r priv-dir/priv-file" will remove the read and write permissions for the file for the group and "other" users. |
| ls -l pub-dir/pub-file  chmod a=rw pub-dir/pub-file  ls -l pub-dir/pub-file | The command "ls -l pub-dir/pub-file" displays information about the file "pub-file", which is located in the directory "pub-dir".  The "chmod" command is used to change access rights to files and directories. The "a=rw" argument indicates that read and write permissions are granted to all (owner, group, and "other" users) of the file.  the command "chmod a=rw pub-dir/pub-file" will grant read and write permissions to all (owner, group and "other" users) of the file. |
| echo "date" > test.sh | The "echo" command outputs text to standard output, which can be redirected to a file using the ">" character, which creates or overwrites a file. |
| ./test.sh  ls -l test.sh | The command "./test.sh" starts the execution of the file "test.sh", which contains the command "date". When this command is executed, it prints the current date and time to the standard output.  The "ls -l test.sh" command displays detailed information about the "test.sh" file, including permissions, owner, group, file size, date and time of last modification, and file name. Usually, when you run the "ls -l" command, you will see the following file attributes: access rights, number of hard links, owner, group, file size, date and time of the last modification, and file name. |
| chmod u+x test.sh  ls -l test.sh  ./test.sh | The command "chmod u+x test.sh" gives the file owner the right to execute the file "test.sh".  The command "ls -l test.sh" displays information about the file "test.sh", including its access rights. After the command "chmod u+x test.sh" is executed, you will see that the permissions of the file "test.sh" have changed, and now the owner of the file has the right to execute the file.  The command "./test.sh" starts the execution of the file "test.sh" in the current directory. Since we have granted the owner of the file the right to execute, the execution of the file "test.sh" will be successful. The result of the command will be the output of the current date and time to the standard output. |
| stat test.sh | The "stat test.sh" command displays detailed information about the "test.sh" file, including the time of the last modification of the file, the time of the last access to the file, the time of the last change in the file status, the index node number, file size, file type, file permissions, file owner, and file group. |
| chmod 775 test.sh  ls -l test.sh | The "chmod 775 test.sh" command grants the file owner read, write, and execute access to the file, and the group and other users read and execute access to the file "test.sh".  After running the "chmod 775 test.sh" command, the "ls -l test.sh" command will display information about the "test.sh" file, including changed access rights. |
| su - | The "su -" command allows the user to change their user ID (UID) and effective group (GID) to those of another user or system administrator. |
| cd /tmp  ls -ld pub-dir  ls -l pub-dir/pub-file | cd /tmp: this command changes the current working directory to /tmp, which is the system directory for temporary files in most Linux distributions.  ls -ld pub-dir: This command displays information about the pub-dir directory located in the current working directory. The -l option allows you to display detailed information about each item in the directory, and -d indicates that you want to display information about the pub-dir directory, not its contents.  ls -l pub-dir/pub-file: This command displays detailed information about the file pub-file located in the pub-dir directory. The -l option allows you to display detailed information about a file, and the pub-dir/pub-file path indicates that you want to display information about the pub-file file located in the pub-dir directory. |
| chown root:root pub-dir  ls -ld pub-dir | chown root:root pub-dir: This command changes the owner and owner group for the pub-dir directory to the root user and the root group. The owner and owner group can only be changed by a user with administrator rights (root) or a user who has the right to change the owner/group.  ls -ld pub-dir: This command displays information about the pub-dir directory. The -l option allows you to display detailed information about each item in the directory, and the -d option indicates that you want to display information about the pub-dir directory, not its contents. After executing the chown command, the information about the owner and group of owners of the pub-dir directory may change. |
| chown bin pub-dir/pub-file  ls -l pub-dir/pub-file | chown bin pub-dir/pub-file: This command changes the owner of the pub-file file located in the pub-dir directory to the user bin. Ownership can only be changed by a user with administrator rights (root) or a user who has the right to change ownership.  ls -l pub-dir/pub-file: This command displays detailed information about the file pub-file located in the pub-dir directory. The -l option allows you to display detailed information about the file, and the pub-dir/pub-file path indicates that you want to display information about the pub-file file located in the pub-dir directory. After executing the chown command, information about the owner of the pub-file file may change. |
| ls -ld priv-dir  ls -l priv-dir/priv-file | ls -ld priv-dir: This command displays information about the priv-dir directory. The -l option allows you to display detailed information about each item in the directory, and the -d option indicates that you want to display information about the priv-dir directory, not its contents.  ls -l priv-dir/priv-file: This command displays detailed information about the priv-file file located in the priv-dir directory. The -l option allows you to display detailed information about a file, and the priv-dir/priv-file path indicates that you want to display information about the priv-file file located in the priv-dir directory. |
| 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: This command displays information about the priv-dir directory. The -l option allows you to display detailed information about each item in the directory, and the -d option indicates that you want to display information about the priv-dir directory, not its contents.  ls -l priv-dir/priv-file: This command displays detailed information about the priv-file file located in the priv-dir directory. The -l option allows you to display detailed information about a file, and the priv-dir/priv-file path indicates that you want to display information about the priv-file file located in the priv-dir directory.  chgrp -R users priv-dir: This command changes the ownership group of the priv-dir directory and all its contents to users. The -R option specifies a recursive pass through all subdirectories and contents of the directory. The group can be changed only by a user with administrator rights (root) or by a user who has the right to change the ownership group.  ls -ld priv-dir: this command displays information about the priv-dir directory again to check if the permissions have changed since the chgrp command was run.  ls -l priv-dir/priv-file: this command again displays detailed information about the priv-file file to check if the access rights have changed since the chgrp command was executed. |
| ls -ld /tmp  ls -ld /var/tmp | ls -ld /tmp: This command displays information about the /tmp directory. The -l option allows you to display detailed information about each item in the directory, and -d indicates that you want to display information about the /tmp directory, not its contents.  ls -ld /var/tmp: this command displays information about the /var/tmp directory. The -l option allows you to display detailed information about each item in the directory, and -d indicates that you want to display information about the /var/tmp directory, not its contents. |
| ls -l /etc/shadow | The ls -l /etc/shadow command displays information about the /etc/shadow file, including its owner, owner group, access rights, and modification date. The /etc/shadow file contains the encrypted passwords of users in the system. |
| ls -l /usr/bin/passwd | The ls -l /usr/bin/passwd command displays detailed information about the /usr/bin/passwd file, including its owner, owner group, access rights, modification date, and size. |
| cd | The cd command without arguments takes the user to the home directory, that is, to the directory that corresponds to his or her username. This command is useful when you need to quickly return to the home directory from any other directory in the system. |
| echo "data" > source | The command creates a new file named source in the current directory and writes the string data to it. The > operator redirects the output from the command line to the file, i.e. in this case, the contents of the string will be written to the source file. If a file with this name already exists, the contents of the file will be replaced with the new contents. |
| ls -li source | The ls -li source command displays information about the source file, including its inode node index, number of links, owner and owner group, and file name. |
| ln source hardlink  ls -li source hardlink | The ln source hardlink command creates a hard link named hardlink to a file named source.  A hard link to a file is a different name that points to the same inode node that is responsible for the file. Thus, if we change the contents of the source file, the changes will be reflected in the hardlink file and vice versa.  The ls -li source hardlink command displays information about both files in a format similar to the ls -li command. The inode node index of both files will be the same because it is a hard link. The number of references to the source file will increase by one because we have created another link to this file. |
| ln hardlink hardlinktwo  ls -li hardlink hardlinktwo source | The ln hardlink hardlinktwo command creates another hard link to the file named hardlink, but with the name hardlinktwo.  The ls -li hardlink hardlinktwo source command displays information about all three files in the current directory in a format similar to the output of the ls -li command. The inode node index of all three files will be the same because they are hard links to the same file. The number of references to the hardlink file will increase by one because we have created another link to this file named hardlinktwo. |
| rm hardlinktwo  ls -li source hardlink | The rm hardlinktwo command deletes the file named hardlinktwo, but it does not affect the source and hardlink files to which it was referenced. Since the source and hardlink files still exist and are hardlinked to the same inode node, they will continue to actually represent the same file, regardless of whether there is any other link to that file.  The ls -li source hardlink command displays information about the two files in the current directory in a format similar to the output of the ls -li command. The inode node index of the two files will be the same because they are hard links to the same file. The number of references to the source file will be two, since it has two hard links named hardlink and source. |
| rm hardlink  ls -li source | The rm hardlink command deletes the hardlink named hardlink. Because it was the only hard link to the file named source, after removing the hard link, the file named source is no longer accessible through the hard link named hardlink.  The ls -li source command displays information about the file named source in the current directory in a format similar to the output of the ls -li command. The inode node index for the file named source will remain the same, since we did not change the file itself, but only removed one of its hard links. The number of references to the file named source will be one, since the hard link named hardlink has been removed. |
| ln -s source softlink  ls -li source softlink | The ln -s source softlink command creates a symbolic link named softlink that points to a file named source.  The ls -li source softlink command displays information about the two files in the current directory in a format similar to the output of the ls -li command. The inode node index for the source file will be the same as in the case of a hard link, but it will be different for the softlink file, since it is a symbolic link. Only one will be displayed in the number of references to the source file, since symbolic links do not increase the number of references to the file. |
| ln -s /proc crossdir  ls -l crossdir | The ln -s /proc crossdir command creates a symbolic link to the /proc directory named crossdir.  The ls -l crossdir command displays information about a file or directory named crossdir in the current directory in a format similar to the output of the ls -l command. The output will indicate that crossdir is a symbolic link to the /proc directory. This means that any attempt to access files or directories through the crossdir symbolic link will be redirected to the corresponding files or directories in the /proc directory. |

\*\*\*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;**

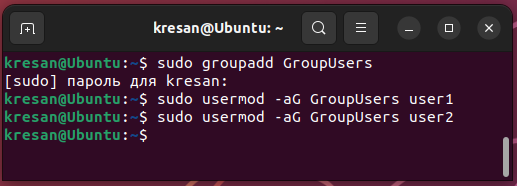
*To create three new users, run the adduser command. After entering this command in the terminal, you will be prompted to enter a password and additional user information if you wish. After entering this information, Linux will create a new user with the following parameters.*



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

*To create a new user group and add two of the three created users to it in the Linux terminal, follow these steps:*

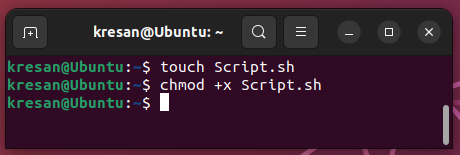
1. *Create a new user group using the groupadd command.*
2. *Add users to the new group using the sudo usermod -aG command.*



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

*To create a new file that will be available for reading, editing, and execution by the file owner, for example, a simple script in a Linux terminal, you should follow these steps:*

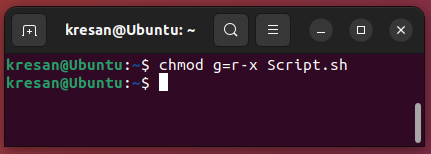
* + 1. *Create a new file using the touch command.*
    2. *Open the file for editing and add the contents of the script.*
    3. *Save the file and set its execution rights using the chmod +x command.*



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

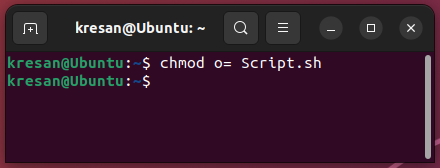
*To grant view and execute permissions to users in the owner's group without granting them edit permissions, follow these steps:*

1. *Run the ls -l command. This command displays the current access rights to the file.*
2. *Change file permissions using the chmod command.*
3. *Set access rights for users of the file owner's group without granting them editing permission. To achieve this, you should set the r-x permissions, which will allow you to read and execute the file, but not to edit it. To do this, run the command chmod g=r-x.*



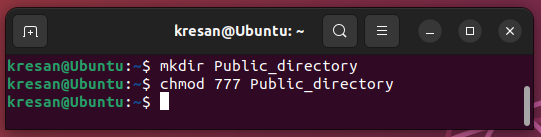
* + **для інших користувачів заборонити доступ до цього файлу;**

*To deny access to a file to other users who do not belong to the owner's group, run the chmod o= command, where o= sets the access rights for other users to no access (i.e., they cannot read, write, or execute the file).*

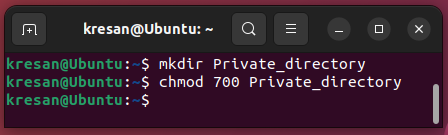


* + **подібні дії виконайте для директорій - створіть директорію, яка буде доступна для всіх трьох користувачів, створіть директорію, яку буде доступна тільки для власника, створіть директорію, яку користувачі групи власника зможуть переглядати, але не редагувати;**

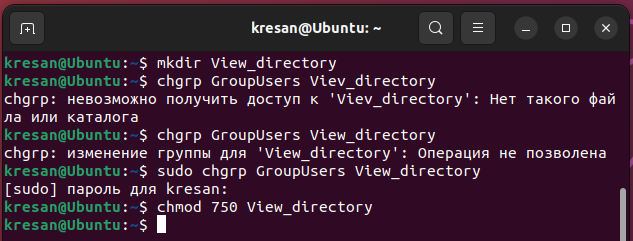
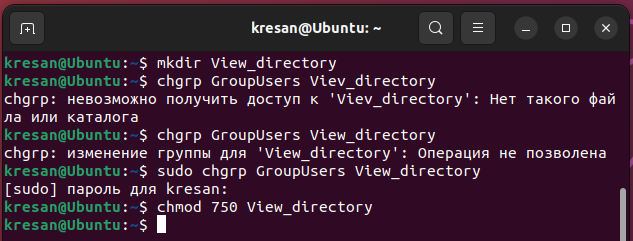
*Create a directory accessible to all three users:*



*Create a directory that is accessible only to the owner:*

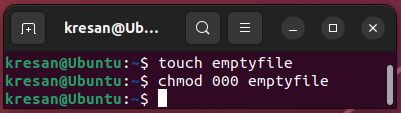


*Create a directory that users of the owner's group can view but not edit:*

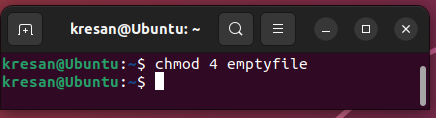


* + **створіть порожній файл під назвою emptyfile за допомогою команди touch emptyfile. Тепер “обнуліть” дозволи для файлу з chmod 000 emptyfile. Що станеться, якщо змінити дозволи для emptyfile, передавши лише одне значення для chmod у числовому режимі, наприклад, chmod 4 emptyfile? Що буде, якщо ми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

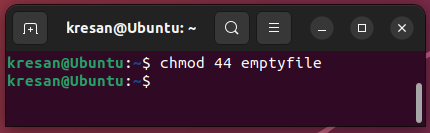
*After executing the chmod 000 emptyfile command, all permissions on the file will be reset to zero, meaning that any user will not be able to read, edit, or execute the file.*



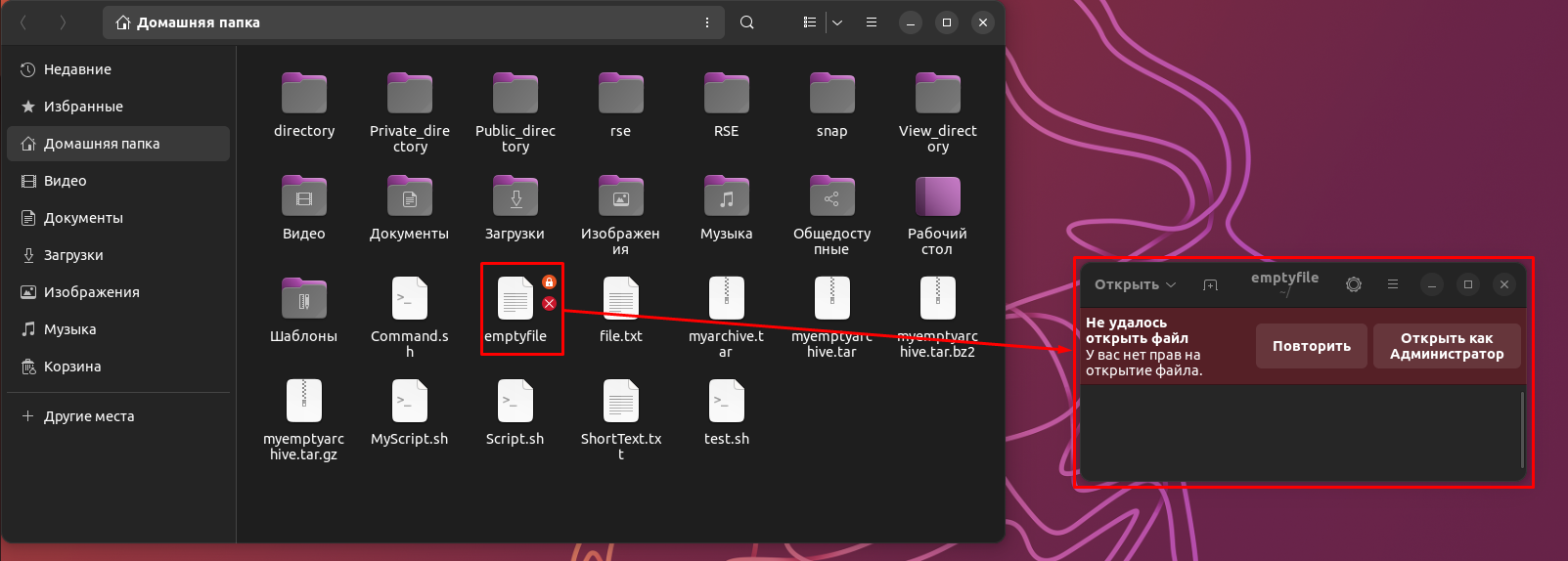
*If we execute the chmod 4 emptyfile command, we will set only read permission for the owner of the file, because 4 corresponds to the binary value 100, which sets read permission only for the owner. For the group and other users, the permissions will remain unchanged (i.e., they will be zero).*



*If we run the command chmod 44 emptyfile, we will set read permissions for the owner and group of the file, and for other users the permissions will remain unchanged.*



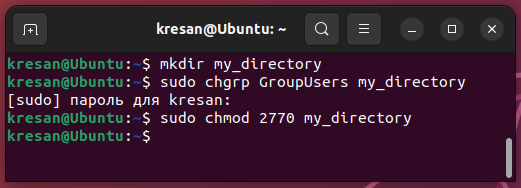
*When using the numeric mode in chmod, we can set different permissions for the owner, group, and other users using different combinations of numbers. Each number corresponds to a bit value of permissions (4 for read, 2 for write, 1 for execute), and you can add these numbers to set different combinations of permissions.*



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

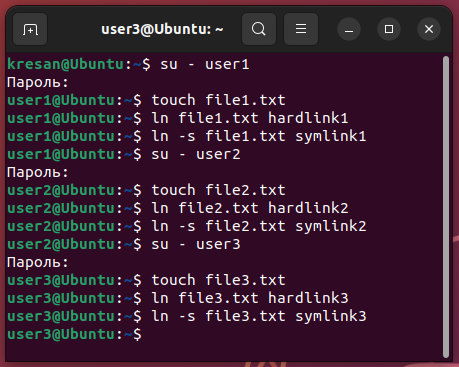
*Option 2 sets the SetGID bit, which means that all files and folders created in the directory will belong to the user group to which the directory belongs. Option 7 sets ownership to the owner and user group only, and allows them to read, write, and execute files. Option 0 sets zero permissions for all other users.*

*Now all files and directories that are created in the my\_directory directory will belong to your user group, and other users will not be able to delete files that were created in this directory.*

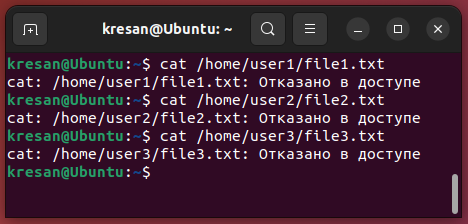


* + **під кожним користувачем створіть по одному новому файлу, та жорстке та символічне посилання на нього;**

*To create new files and hard links to them, use the following commands:*



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



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

*Other users who are not the owners of the files cannot delete them without the appropriate permissions. If files were created with permissions that allow group owners to perform operations on the file, other users in that group may have some rights to the file, such as viewing it, but cannot delete it without additional permissions. Thus, the owner of the file or a user with the appropriate permissions can delete these files, but other users cannot.*

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

**Відповіді на контрольні запитання**

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

*The symbolic method is a way to change file and directory permissions in Unix/Linux that uses symbolic mnemonics to set new permissions.*

*Examples of changing access rights using the symbolic method:*

*To change the permissions of file.txt so that the owner can read, edit, and execute it, the group can read it, and all other users cannot access it, run the following command: chmod u+rwx,g+r,o-rwx file.txt*

*To change the permissions of the mydir directory so that all users can read and execute files in the directory, but only the owner can edit files, run the following command: chmod u+rwx,g+rx,o+rx mydir*

*To change the permissions of the script.sh file so that all users can execute it, but cannot edit or read it, run the following command: chmod a+x script.sh*

*In these examples, the chmod command uses symbolic notation to set permissions on files and directories. u indicates the owner of the file, g indicates the user group, o indicates other users, and a indicates all users. r indicates the right to read the file, w indicates the right to edit the file, and x indicates the right to execute the file.*

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

*The numeric method (or octal method) is a way to change file and directory permissions in Unix/Linux that uses octal numbers to set new permissions. Each right (read, write, execute) has its own numeric code, and the resulting codes are combined into an octal number that represents the desired set of rights.*

*Examples of changing access rights using the numeric method:*

*To change the permissions of a file.txt file so that the owner can read, edit, and execute it, a group can read it, and all other users cannot access it, run the following command: chmod 750 file.txt*

*In this case, the number 7 represents the owner's access rights (read, write, and execute), the number 5 represents the group's access rights (read and execute), and the number 0 represents the access rights of other users (no access).*

*To change the access rights to the mydir directory so that all users can read and execute files in this directory, but only the owner can edit files, run the following command: chmod 755 mydir*

*In this case, the number 7 is the owner's access rights (read, write, and execute), the number 5 is the group's access rights (read and execute), and the number 5 is the other users' access rights (read and execute only).*

*To change the access rights to the script.sh file so that all users can execute it, but cannot edit or read it, run the following command: chmod 111 script.sh*

*In this case, all three numeric values are set to execute permission only, which corresponds to 1 in the octal system. Thus, only users can execute the script.sh file, but cannot read or edit it.*

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

*For example, you can execute a file that has runtime permissions but no read permissions. This is possible because to execute a file, you do not need to read its contents. When you execute a file, the operating system simply executes the instructions stored in the file and displays the result on the screen or in another output stream.*

*For example, if a file has execute permission but no read permission, you can still run the file as an executable script. However, if you try to read the contents of the file before executing it, you will receive an error from the operating system.*

1. Яке призначення команди umask?

*The umask command on Unix/Linux sets the default permission mask for newly created files and directories in the current session. The default permissions mask is subtracted from the maximum permissions allowed, that is, it determines which permissions will not be set by default when creating new files and directories.*

*The default permission mask value can be set from 0 to 777 in octal format. Each digit represents a permission mask for the corresponding user group (owner, group, and others). For example, if the default permission mask is set to 022, it means that write permissions for the group and other users will be subtracted from the maximum permitted access rights.*

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

*Changes to access rights and permissions in the current session are usually temporary and will not be saved when you close the session or restart the system.*

*If you change access rights and permissions in the current session, they are applied to all files and directories that you interact with during this session. These changes will not be applied to other sessions or to other users, unless you change permissions on system files or use commands that use these permissions as part of their functionality.*

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

*Yes, most operating systems have a template that is used for permissions when creating new files and directories. Typically, when a new file or directory is created, the system uses default values that can be customized using the umask command. The umask value sets a bit mask that subtracts certain access permissions when creating new files and directories.*

*Typically, the default values for new files are -rw-rw-rw- or 666, which means that all users have read and write permissions, but not execute permissions. For new directories, the default is usually -rwxrwxrwx or 777, which means that all users have full access to the directory.*

*To change the default permissions, you can set a new umask value. For example, if you want to set the permissions for new files to -rw-r--r-- or 644, you can run the following command: 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?

*The best place to create temporary files on most operating systems is /tmp or /var/tmp. These directories are usually designated specifically for temporary files and have the correct permissions to ensure that the system is secure and efficient. To create a temporary file, you can use the mktemp command, which automatically creates a unique file with a random name and permissions. For example, the mktemp /tmp/tempfile.XXXXXX command will create a temporary file with the variable part XXXXXX in the /tmp directory. After using the temporary file, you need to delete it with the rm command to free up space on the system.*

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

*A hard link is a file name that points to the same file system node as the original file name. Files referenced by hard links look like a single file because they have the same inode and point to the same block of data on the disk.*

*To create a hard link, use the ln command with the -s option, with the source file and the name of the hard link as arguments. For example, to create a hard link file2 to file1 in the current directory, run the following command: ln file1 file2*

*Hard links are usually used in situations where you need to have access to the same file from different names or when you need to create file backups. They allow you to reduce the amount of disk space because they refer to the same block of data on the disk. Also, if the source file is moved or renamed, a hard link to it will still work because it refers to an inode, not a filename. However, hard links cannot be created on a different disk or file system, and they cannot point to directories.*

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

*Symbolic links (symlink or soft link) are a special type of file that is created to point to another file or directory. It contains a reference to the file name, not to its sometimes-named structure, and therefore can refer to a file or directory regardless of its location in the file system.*

*To create a symbolic link, use the ln command with the -s option: ln -s /path/to/original\_file /path/to/symbolic\_link*

*For example, if we want to create a symbolic link to the file /home/user/Documents/report.txt in the /home/user/Downloads directory, we can run the following command: ln -s /home/user/Documents/report.txt /home/user/Downloads/report-link*

*Now you will have a file named report-link in the /home/user/Downloads directory that will link to the report.txt file in the /home/user/Documents directory.*

*Symbolic links can be useful in the following situations:*

* *When creating a link to a long path to a file with a short name to make it easier to access the file.*
* *When placing files that are frequently moved on external media or network directories to ensure stable access to these files, regardless of their location.*
* *When creating many identical copies of a single file to reduce disk space.*
* *When creating a directory link to have two different names for the same directory.*

1. Порівняйте жорсткі та символічні посилання?

*A hard link and a symbolic link are two ways to create a link to a file or directory in Linux.*

*A hard link is created by creating a new file system entry that points to the same file system node as the original file. A hard link can be used to access the original file from different directories, including the same directory where the original file is located. It cannot be created for directories and cannot cross file system boundaries.*

*A symbolic link is a record that contains the path to the original file. It is created using the ln command with the -s option. A symbolic link can cross file system boundaries and can be created for directories. It can be used to create a convenient path to the original file from different directories.*

*One of the main differences between a hard link and a symbolic link is that if you delete the original file, the hard link to it will still exist and have access to the file's data, but if you delete the original file that the symbolic link refers to, the symbolic link will be invalid.*

*Hard links are short and efficient because they don't use additional memory to store additional information about the link. Symbolic links provide more flexibility and convenience because they can be used to create links to directories and file systems that overlap.*

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

*If you delete the original file, the hard link will continue to exist and will refer to the same content.*

* + символічне посилання;

*If you delete the symbolic link, the original file will remain in place, but the symbolic link will be invalid and point to nothing.*

* + жорстке посилання.

*However, if you delete the hard link, the original file and the symbolic link will not be saved on your hard disk and will be invalid.*

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

**Висновки**

When working with the Bash shell, a user can use a variety of commands to work with files and directories in the Linux operating system. Knowing the commands for working with file owners and permissions is important because it allows you to control which users have access to certain files and directories and what they can do with them.