Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій у енергетиці

Лабораторна робота №5

з дисципліни «Операційна система UNIX»

«Створення сценаріїв в оболонці Bash»

Виконав:

студент 2-го курсу, НН ІАТЕ

групи ТР-23

Брукалюк С.П.

Перевірила:

д.т.н. Левченко Л.О.

КИЇВ 2023

**Мета роботи**: набути навичок створювання bash-скриптів в OC Linux

**Порядок роботи:**

1. Ознайомитися з теоретичними відомостями.

2. Виконати завдання за варіантом.

3. Представити скріншоти виконання завдання.

**Короткий теоретичний опис роботи**

**Bourne-again shell** (GNU Bash) - це реалізація Unix shell, написана на C в 1987 році Брайаном Фоксом (Brian Fox) для GNU Project. Синтаксис мови Bash є надбудовою синтаксису мови Bourne shell. Переважна більшість скриптів для Bourne shell можуть бути виконані інтерпретатором Bash без змін, за винятком скриптів, які використовують спеціальні змінні або вбудовані команди Bourne shell. Також синтаксис мови Bash включає ідеї, запозичені з Korn shell (ksh) і C shell (csh): редагування командного рядка, історія команд, стек директорій, змінні $ RANDOM і $ PPID, синтаксис POSIX *для підстановки команд: $* (...).

Командна оболонка *BASH* дозволяє створювати скрипти за допомогою групування декількох команд, які виконують певну дію.

***Скрипт*** - це звичайний текстовий файл, що містить системні або вбудовані команди оболонки. Такий файл може бути запущений на виконання наступним чином: ***$bash ім’я\_файла.***

Оболонка послідовно інтерпретує і виконує команди, задані в сценарії. Ці ж команди можуть бути виконані простим послідовним викликом їх в командному рядку оболонки. Для файлів сценаріїв оболонки *bash* прийнято встановлювати розширення ***.sh***. Тобто, для виконання скрипту необхідно запустити командну оболонку, передавши їй як параметр ім’я відповідного файлу. Є другий варіант запуску скрипту - вказати його ім'я в командній оболонці (тобто зробити з нього якийсь вид програми). Для цього треба в параметрах доступу визначити файл як *виконуваний*, і в перших рядках цього файлу явно вказати оболонку, для якої призначений цей скрипт, в такий спосіб:

**#!оболонка**

Будь-який сценарій для Bash починається з вказівки в першому рядку комбінації:

**#!/bin/bash**

Символи **#!** повідомляють системі про те, що наступний за ними аргумент – це програма, яка застосовується для виконання даного файлу. В даному випадку програма ***/bin/sh*** - командна оболонка, що застосовується за замовчуванням. Ця послідовність вказує на програму, яка використовується для обробки сценарію в командну оболонку bash. У загальному випадку символ «***#***» в скрипті означає коментар, що означає ігнорування рядка. Але якщо він є першим символом файлу і за ним слідує символ «**!**» та шлях до файлу (наприклад, /bin/bash, /bin/perl, /bin/ sh і т.д), командна оболонка запускає відповідний файл і передає йому його ім'я в якості аргументу.

Створити файл, який містить команди, можна допомогою будь-якого текстового редактора. У даній роботі рекомендується використовувати вбудований в ***mc*** редактор. Для створення нового файлу в *mc* використовуйте комбінацію клавіш Shift + F4. Створіть файл з ім'ям *first* з таким вмістом:

#!/bin/sh

# first

# Цей файл переглядає всі файли в поточному каталозі для пошуку рядка

# POSIX, а потім виводить імена знайдених файлів в стандартне виведення.

for file in \*

do

if grep -q POSIX $file

then

echo $ file

fi

done

exit 0

Виконання скрипту відбувається по рядкам.

***Перетворення сценарію у виконуваний файл***

Файл сценарію можна виконати двома способами. Перший - запустити оболонку з ім'ям файлу сценарію як параметром: **$ /bin/sh first.** Цей варіант працює.

Другий спосіб (більш доцільний) - запускати сценарій, ввівши його ім'я і тим самим присвоївши йому статус інших команд Linux. Зробити це можна за допомогою команди **chmod**, змінивши режим файлу (**file mode**) і зробивши його виконуваним для всіх користувачів:

**$ chmod u+x <ім’я сценарія>**

**$ chmod u+x first**

Додані режими: для власника (User), виконання (x - eXecutable).

Аналогічну функціональність реалізує наступна команда:

**chmod 744 <ім’я\_файла\_сценарія>**

Після цього можна виконувати файл за допомогою **команди $ first**

При цьому може з'явитися повідомлення про помилку, яке говорить про те, що команда не знайдена.

Запуск сценарію на виконання з поточного каталогу проводиться за допомогою наступної команди: **./ <ім’я\_файла\_сценарія>**

Виправити помилку можна запровадивши з клавіатури в **командному рядку ./first** в каталозі, що містить сценарій, щоб задати командній оболонці повний відносний шлях до файлу.

Зазначення шляху, який починається з символів «**./»**, дає ще одну перевагу - ви випадково не зможете виконати іншу команду з тим же ім'ям, що і у вашого файлу сценарію. Після того як ви переконаєтеся в коректній роботі вашого сценарію, можете перемістити його в більш відповідне місце, ніж поточний каталог.

***Функції***

Синтаксис оголошення функції:

function <ім'я> ()

{

<список>;

}

Функція визначається іменем <ім'я>. Тіло функції - <список> записується між дужками { }.

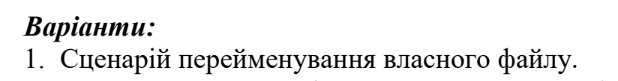
**Завдання**

1. Ознайомитися з теоретичними матеріалом по лабораторній роботі.

2. Опанувати команди, які використовують при написанні сценарію.

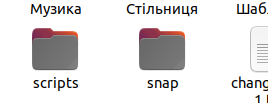
3. Підготувати звіт для викладача про виконання лабораторної роботи і представити його.

**Завдання 1-го варіанту**

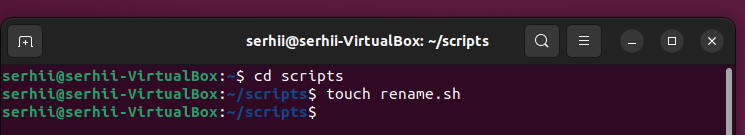
****

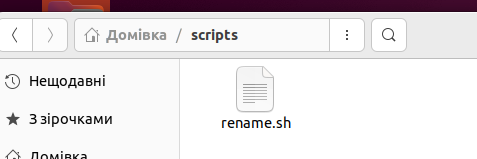
**Результат виконання роботи виконання роботи**

1.Створення нової папки з назвою “scripts”.

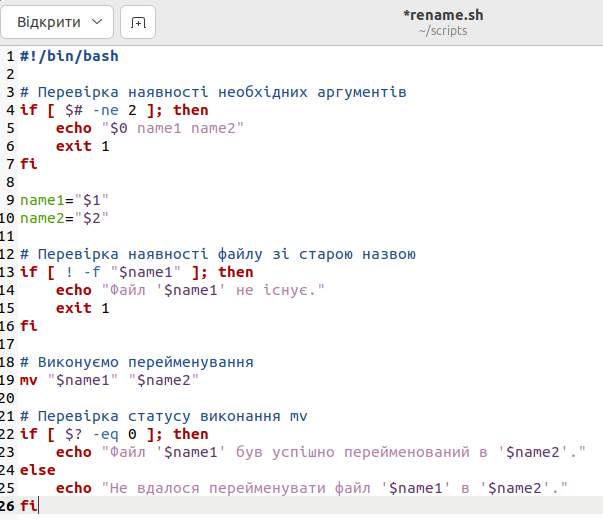
** й**

2.Перехід до цієї папки та створення файлу “rename.sh” за допомогою команди touch.





3.Написання сценарію для перейменування власного файлу з name1.txt на name2.txt.



3.Перетворення файлу rename.sh у виконуваний файл за допомогою команди chmod +x rename.sh.



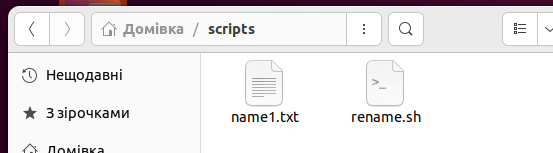
4.Протестуємо цей скрипт декількома сценаріями:

1).Спробуємо перейменувати, коли файл name1.txt не створено.

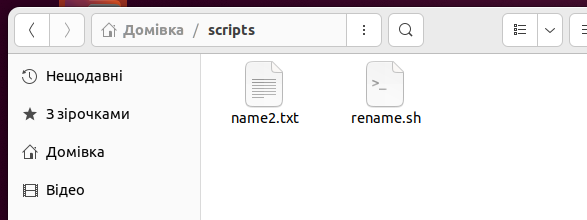


2). Перейменуємо вже створений файл name1.txt.









**Висновок**: В ході виконання лабораторної роботи було отримано нові знання та практичний досвід у створенні bash-скриптів для операційної системи Linux. Ми успішно розробили скрипт відповідно до варіанту 1, який дозволив нам впоратися із завданням перейменування власного файлу.