

Лабораторна робота №3

Створення сценаріїв оболонки Bash у середовищі GNU/Linux

Мета роботи: навчитись створювати сценарії оболонки Bash у середовищі GNU/Linux а також навчитись працювати з одним із текстових редакторів на вибір: nano, vim, mcedit.

Загальні відомості

Іноді існує необхідність періодично запускати певний ряд команд у командному рядку. Замість того щоб вводити їх по черзі їх можна записати у спеціальний файл, який називається сценарієм оболонки або просто скриптом, після чого скрипт можна запускати. Іншими словами, сценарії командного рядка — це набори тих самих команд, які можна вводити з клавіатури, зібрані у файли і об'єднані якоюсь спільною метою. У цьому результаті роботи команд можуть бути або самостійну цінність, або бути вхідними даними інших команд. Сценарії - це потужний спосіб автоматизації дій, що часто виконуються.

У командній оболонці Bash (стандартній командній оболонці GNU/Linux) теж можна створювати скрипти. Bash-скрипти є текстовими файлами, подібно скриптам Python чи JavaScript, і можуть бути виконані у командному рядку. Кожен скрипт починається з так званого шебанг-рядка, який дає зрозуміти операційній системі що файл є скриптом. У Bash шебанг-рядок має наступний вигляд:

```
#!/bin/bash
```

За шебанг-рядком слідують набори команд. Команди вводяться одна за одною, кожна з нового рядка. Найбільш простою є команда echo:

```
echo "Hello from CS!"
```

В результаті її виконання буде виведений наступний текст:

```
Hello from CS!
```

Команди оболонки відокремлюються розривом рядка, коментарі виділяють знаком решітки. Виглядає це наступним чином:

```
#!/bin/bash
# This is a comment
pwd
whoami
```

Файли скриптів Bash зазвичай зберігають із розширенням `.sh`, щоб можна було легко зрозуміти тип файлу у лістингу.

Для того щоб файл скрипту можна було запустити, файл необхідно зробити виконуваним, інакше виникне помилка "Permission denied". Зробити це можна за допомогою наступної команди:

```
chmod +x myscript.sh
```

де `myscript.sh` - це назва створеного скрипта.

Для того щоб запустити скрипт, необхідно написати його ім'я, дописавши крапку та слеш перед ним:

```
./myscript.sh
```

Як і у звичайних мов програмування, у Bash-скриптах можна використовувати змінні. Звертатись до змінних можна за допомогою знаку долару. Приклад:

```
#!/bin/bash
# using variables
age=12
child="Roman"
echo "$child is a good boy, he is $age years old"
```

У прикладі застосовано змінні `age` та `child`. Об'явлення змінних відбувається за допомогою знаку `=`. Використані змінні серед аргументів команди `echo`.

Виводи команд можна зберігати у файли. Подвійний знак більше призводить до переспрямування запису виводу тексту команди у файл, замість відображення їх на екран. До прикладу можна взяти команду `ls`:

```
ls >> list.txt
```

У прикладі команда виводить перелік файлів у текстовий файл `list.txt`. Подібним чином можна застосувати і команду `echo`:

```
echo "Hello from CS!" >> hello.txt
```

Результатом виконання команди стане запис тексту "Hello from CS!" у файл `hello.txt`. Можна спробувати запустити команду кілька раз, щоб переконатись що доданий текст з'явиться на останньому рядку текстового файлу.

Завдання

1. Створити новий Docker-контейнер "lab3" з Ubuntu GNU/Linux (рекомендується використати для цього образ "linuxation/ostamp" з лабораторної роботи №2, його можна знайти у вкладці "Images" зліва).
(за бажанням) Можна створити папку lab3 на комп'ютері та підмонтувати її за шляхом `/mnt/lab3`.

2. Створити у ній скрипт `lab3.sh` за допомогою редакторів `nano`, `vim` чи `mcedit`. Також для цього можна використати `Visual Studio Code` чи іншу IDE (але лише якщо підмонтовано папку з реального комп'ютера). В жодному разі не використовуйте Блокнот Windows (він псує скрипти, роблячи їх невиконуємими).
3. Зробити скрипт виконуваним
4. Реалізувати у скрипті вивід привітання на екран (напис типу "`hi vntu`")
5. Реалізувати у скрипті створення файлу `lab3.txt`
6. Реалізувати у скрипті запис дати у файл `lab3.txt`
7. Реалізувати цикл із виведенням стрічки "`LAB N IS DONE`" та з записом їх у файл `lab3.txt`, де `N` - це число від 1 до 5
8. Підготувати звіт по лабораторній роботі з кодом скрипту, файлом `lab3.txt` та скріншотом результату виконання.

Додаткові завдання (по два на вибір)

1. Створити `bash`-скрипт, що приймає в якості аргументів ряд чисел та знаходить серед них найбільше.
2. Створити `bash`-скрипт, який зчитує всі слова у аргументах та обертає у зворотному порядку порядок символів у них.
3. Створити `bash`-скрипт, що зчитує всі слова у аргументах і підраховує кількість символів у них.
4. Створити `bash`-скрипт, що зчитує в якості аргументу число, створює текстовий файл та циклічно записує у файл всі парні числа від 0 до вказаного числа.
5. Створити `bash`-скрипт, що відображає найкоротше та найдовше ім'я користувача у системі.
6. Створити `bash`-скрипт, що генерує довільний 8-ми символний пароль з цифр, літер верхнього та нижнього регістру.
7. Створити `bash`-скрипт, що перевіряє наявність команд `curl` та `git` у системі і при їх наявності вивести їх місцезнаходження у системі

Зміст звіту

- короткі відповіді на контрольні питання;
- код скрипту `lab3.sh`;
- вміст файлу `lab3.txt`;
- скріншот результату виконання скрипту;
- висновки.