**Київський національний університет імені Тараса Шевченка**

**факультет радіофізики, електроніки та комп’ютерних систем**

Лабораторна робота №1

з дисципліни інженерія комп’ютерного забезпечення

**Тема:** «Distributed version-control system Git»

Роботу виконав

студент 3 курсу

системний адміністратор

Коваленко Володимир Вікторович

Київ 2023

**Мета роботи:** ознайомитися системами керування версіями. Дослідити та отримати практичні навики щодо створення найпростішої програми та власного репозиторію.

# **Настроювання Git.**

## Відкрийте папку, в якій знаходяться файли проекту. З контекстного меню виберіть Git Bash Here , і у вікні, що відкрилося введіть команду: $ git init

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

## Авторизуйтесь для внесення подальших змін, вкажіть своє ім’я та електронну пошту

Зображення, що містить текст, електроніка, знімок екрана, програмне забезпечення

Автоматично згенерований опис

## Створіть в папці з проектом файл «.gitignore» для постійного виключення з файлів зі змінами, що відстежуються, тимчасових файлів та файлів, що автоматично генеруються компіляторам та компоновщиком.

Зображення, що містить знімок екрана, текст, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

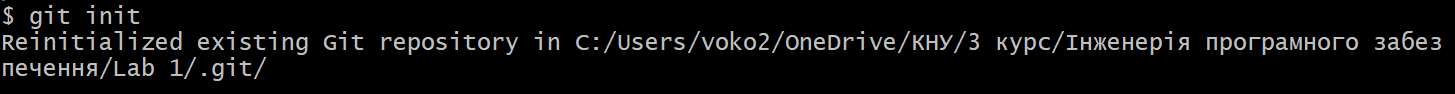
Створіть у в файлі «.gitignore» список файлів та папок, які не потрібно

відслідковувати

Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис

## Імпортування папки в Git. Необхідно зайти в поточну папку проекту і виконати команду git init



В поточній папці проекту кожен файл може знаходитись в двох станах, під

вірсійним контролем і без нього, тобто коли файл не відслідковується

(untracked). За допомогою команди git status можна подивитись в якому стані знаходиться репозиторій.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Додайте всі файли, що знаходяться в папці для відслідковування за допомогою команди git add

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Після редагування файлу Lab 1.docx:

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

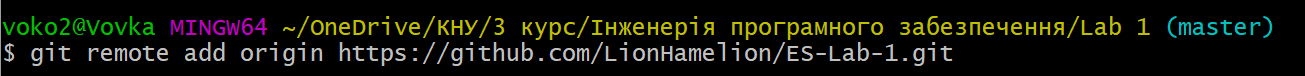
Далі необхідно виконати команду git commit –m “first modification”

Commit робить “знімок” проіндексованих файлів в робочому каталозі.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

## Додавання імен віддалених репозиторіїв.



Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

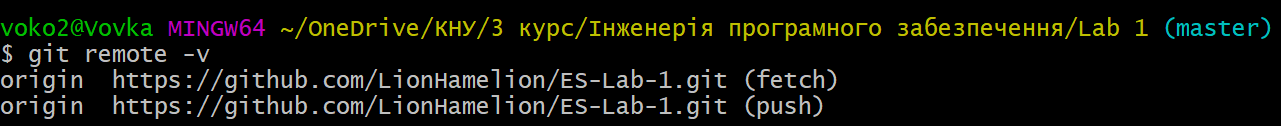
Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

# **Клонування репозиторію з сервера.**

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис



# **Завдання 1**

Зробити клонування репозиторію коду космічної місії польоту на місяць. https://github.com/chrislgarry/Apollo-11

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

# **Основні команди Git**

## Перевірка стану робочої папки здійснюється за допомогою команди

Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис

## Включення файлів і папок в перелік, що відслідковується виконується командою git add \*cpp \*dfm \*h .

Створимо новий файл hello.cpp

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

Зображення, що містить текст, Шрифт, знімок екрана, ряд

Автоматично згенерований опис

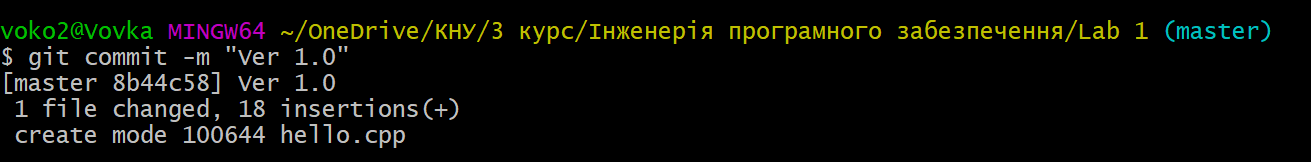
## Додайте змінений файл до файлів, що підготовлені до фіксації

стану (staged files), за допомогою команди git add,

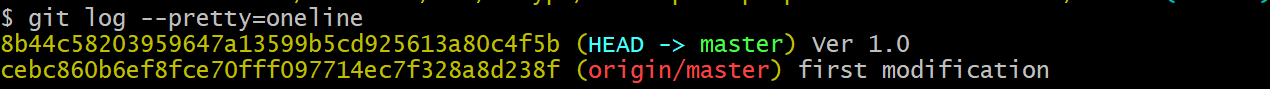
Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Перевірте стан файлів за допомогою команди git status і збережіть знімок поточного стану файлів, що відслідковуються за допомогою команди git commit: $ git commit -m “Ver 1.0”.



## Перегляньте історію змін за допомогою команди git log:

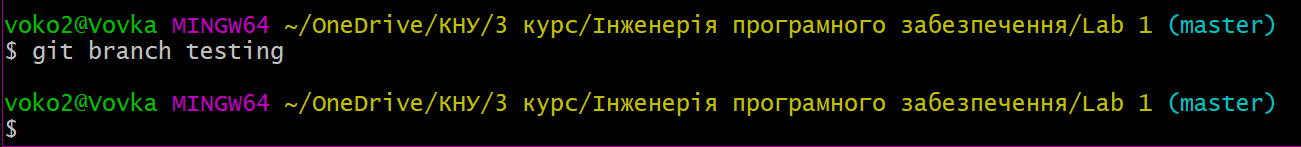


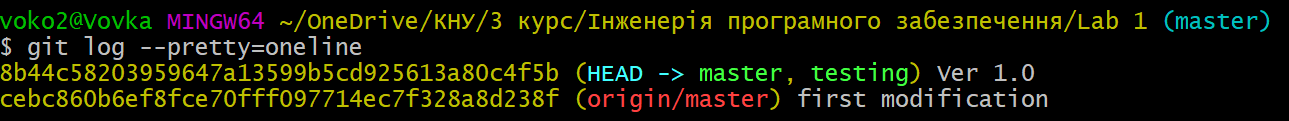
Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис

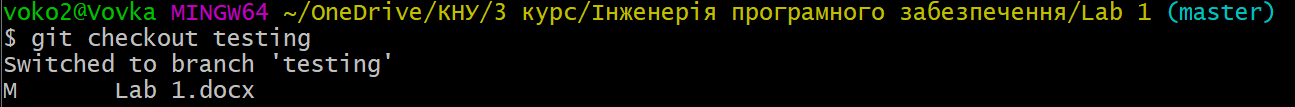
## Створення нової гілки

Створіть гілку testing за допомогою команди $ git branch testing

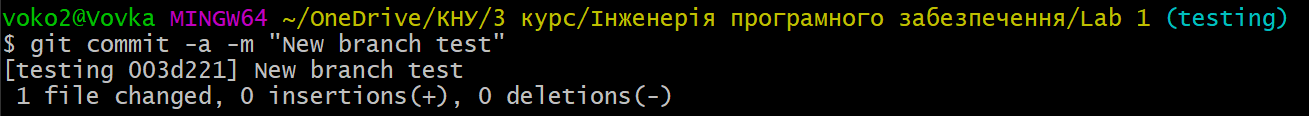




Перехід на нову гілку виконується за допомогою команди git checkout.



## Внесіть зміни в один-два текстові файли нової гілки, що відслідковуються і зафіксуйте зміни за допомогою команди



## Об’єднування гілок.

Перейдіть на головну гілку master за допомогою команди $ git checkout master.