**Київський національний університет імені Тараса Шевченка**

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

лабораторна робота № 1

З дисципліни “Інженерія програмного забезпечення”

Тема:«Distributed version-control system Git»

Роботу виконав

студент 3 курсу

МА

Пастушенко Ілля Сергійович

Київ 2023

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

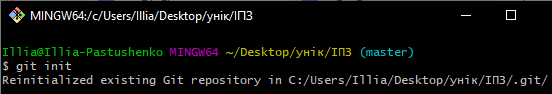
**Завдання на лабораторну роботу:**

**Ознайомитись з теоретичними відомостями**, ретельно опрацювати матеріал. Вміти давати пояснення термінам та поняттям: система керування версіями; централізовані та розподілені системи контролю версіями; репозиторій; приватні та відкриті репозиторії; GitHub.

**Практична частина.**

**Настроювання Git.**

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

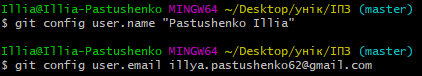


У результаті в папці з проектом з'являється папка з іменем «.git», що містить усі необхідні файли локального репозиторію.

2. Авторизуйтесь для внесення подальших змін, вкажіть своє ім’я та електронну пошту, наприклад:

$ *git config user.name "Bohdan Sus"*

$ *git config user.email [bogdaniht@gmail.com](mailto:suse@univ.kiev.ua)*



Якщо ви плануєте постійно працювати з проектами на даному комп'ютері, то додайте до команд авторизації ключ *global*:

$ *git config --global user.name "Bohdan Sus"*

$ *git config --global user.email [bogdaniht@gmail.com](mailto:suse@univ.kiev.ua)*

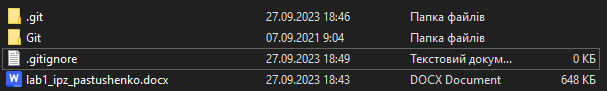
*(Ці настройки достатньо зробити один раз).*

*Перевірити настройки можна за допомогою команди git config –list*

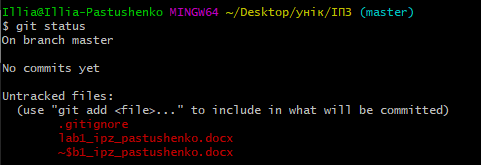
*Щоб отримати допомогу необхідно набрати*

*git help назва команди*

3. 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

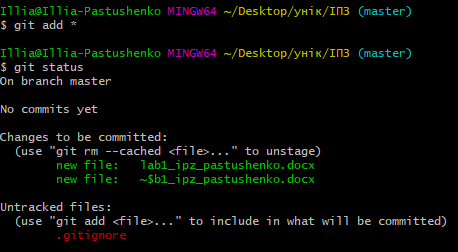


4. Імпортування папки в Git.

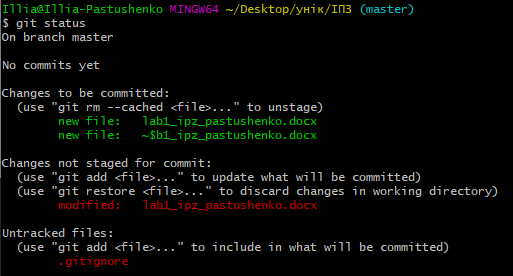


Додайте всі файли, що знаходяться в папці для відслідковування за допомогою команди *add \**

Виконайте команду *git status* і перевірте стан файлів.



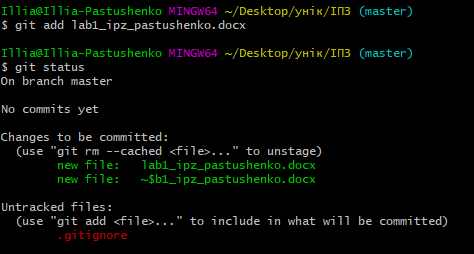
Відредагуйте якийсь файл проекту, збережітьь зміни і знову виконайте команду *git status*



Червоним кольором буде показао не проіндексовані файли. Для того щоб їх проіндексувати знову необхідно виконати команду

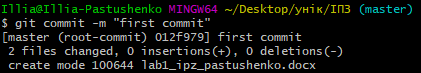
*add <file>*

*git status* показує стан файлу після останнього виконання команди *git add*



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

Commit робить “знімок” проіндексованих файлів в робочому каталозі.



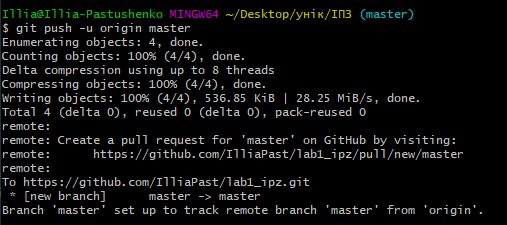
5. Додавання імен віддалених репозиторіїв.

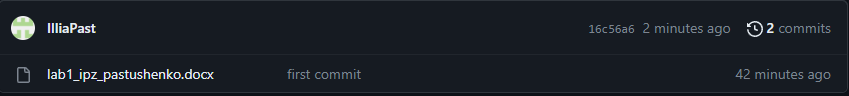
Далі в консольному вікні необхідно виконати команду, наприклад:

git remote add origin <https://github.com/bohdansus/labs2023.git>

git push -u origin master



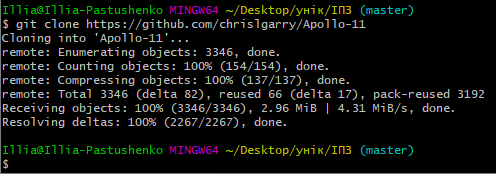


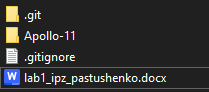


**Завдання 1.**

Зробити клонування репозиторію коду космічної місії польоту на місяць.

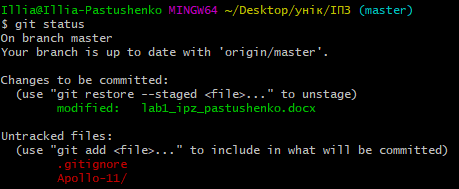
<https://github.com/chrislgarry/Apollo-11>



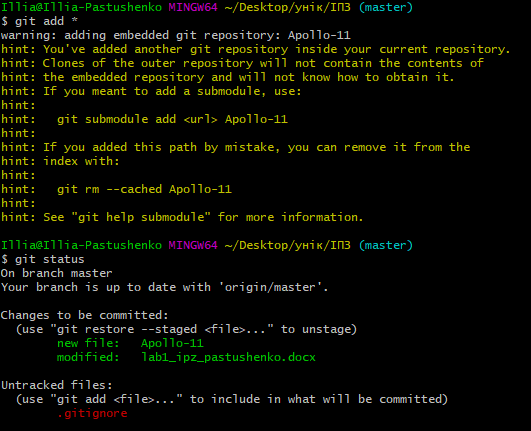


**Основні команди Git**

1. Перевірка стану робочої папки здійснюється за допомогою команди



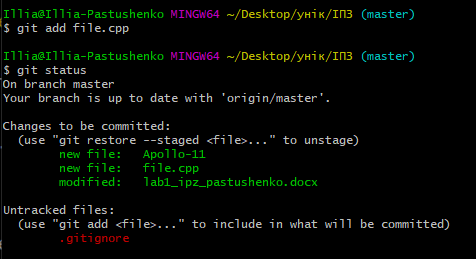
6. Включення файлів і папок в перелік, що відслідковується виконується командою git add:



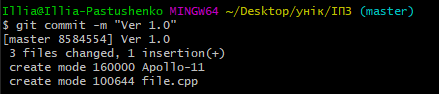
Внесіть зміни в один з текстових файлів, наприклад, додайте коментар до якого-небудь оператора вихідного тексту програми на мові C ++, і збережіть змінений файл. Знову виконайте команду *git status*. Перелік файлів робочої папки, що відслідковуються, які не були змінені і підготовлені до фіксації їх стану командою *commit* буде виведений зеленим шрифтом. Ім'я зміненого файлу і файлів, які не включені в ті, що відслідковуються, будуть показані червоним шрифтом.

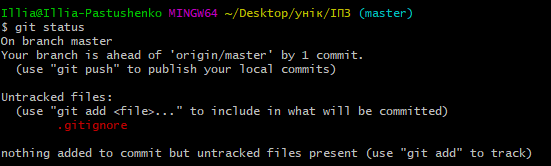
1. Додайте змінений файл до файлів, що підготовлені до фіксації стану (staged files), за допомогою команди

$ *git add* *ім’я\_файла* .

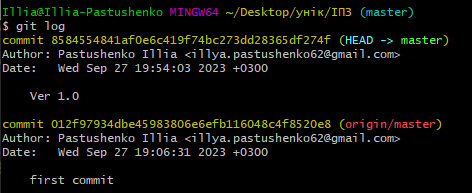


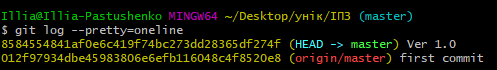
Перевірте стан файлів командою git status і збережіть знімок поточного стану відслідковуються файлів командою git commit: $ git commit -m 'Ver 1.0'.



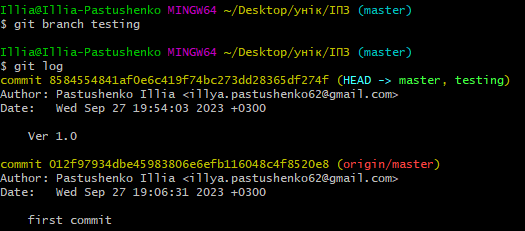


1. Перегляньте історію змін за допомогою команди *git log*:

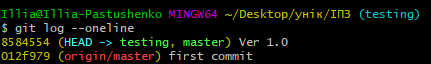




10. **Створення нової гілки**





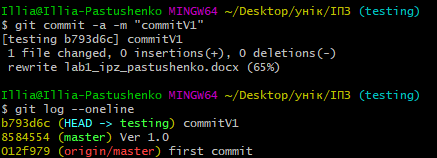


11. Внесіть зміни в один-два текстові файли нової гілки, що відслідковуються і зафіксуйте зміни за допомогою команди

*$ git commit -a -m "текст\_комментаря\_commit"*

Ключ «-*a*» в команді *git commit* дає можливість пропустити команду *git add,* що додає змінені файли в область індексуваня, та готує їх до фіксації.

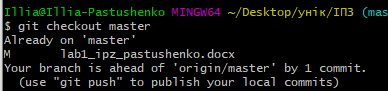
Перегляньте історію змін гілки *testing* за допомогою команди *git log --oneline* .



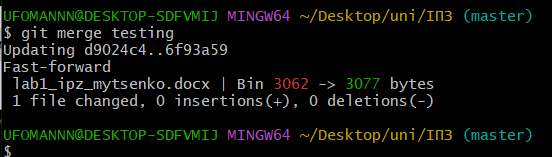
12. Об’єднування гілок.

Перейдіть на головну гілку *master* за допомогою команди

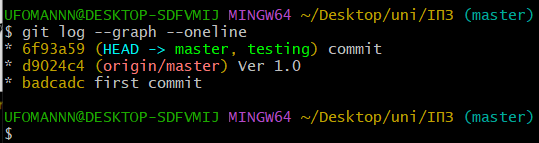
*$ git checkout master .*



Виконайте об’єднування гілки *testing* з гілкою *master* за допомогою команди *$ git merge testing.*

**

Перегляньте історію змін за допомогою команди *$ git log --graph –oneline*

**

**Завдання 2.**

Написати програму для розрахунку чисел Фібоначчі (будь якою мовою програмування), завантажити на Github, посилання на репозиторій вставити в звіт до лабораторної роботи.

*<https://github.com/vadimmits/test1.1>*

Висновок: У ході виконання цієї лабораторної роботи було здійснено дослідження та отримано практичні навички щодо використання розподіленої системи контролю версій Git.