**Київський національний університет імені Тараса Шевченка факультет радіофізики, електроніки та комп’ютерних систем**

Лабораторна робота № 1

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

Роботу виконав

студент 3 курсу

КІ-МА

Сподар Назар Іванович

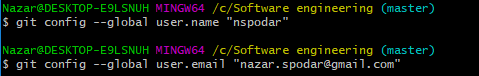
Київ 2019

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



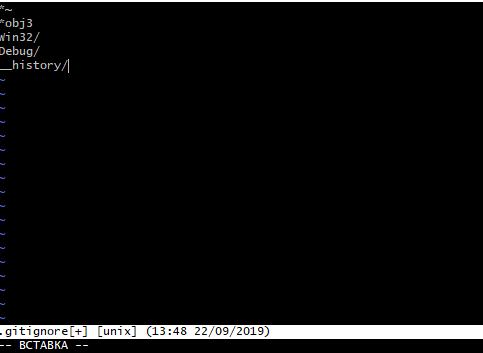
2. Авторизуйтесь для внесення подальших змін, вкажіть своє ім’я та електронну пошту.



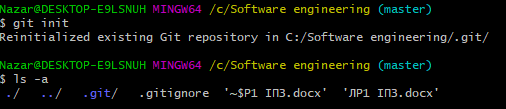
3. Створіть в папці з проектом файл «.gitignore» для постійного виключення з файлів зі змінами, що відстежуються, тимчасових файлів та файлів, що автоматично генеруються компіляторам та компоновщиком.

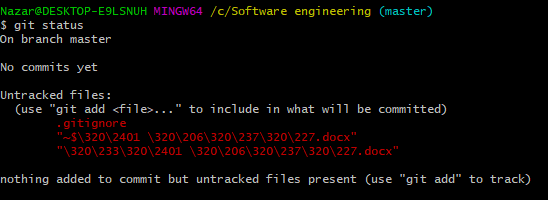


Створіть у в файлі «.gitignore» список файлів та папок, які не потрібно

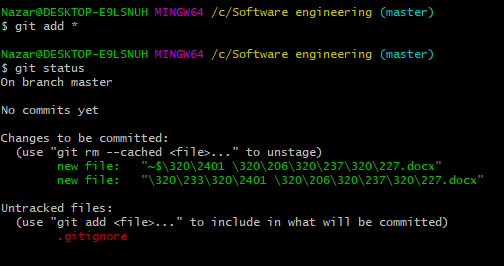


4.​ ​ Імпортування папки в Git. Необхідно зайти в поточну папку проекту і виконати команду git init





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, збережіть зміни і знову виконайте команду git status. Червоним кольором буде показано не проіндексовані файли. Для того щоб їх проіндексувати знову необхідно виконати команду add <file>

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

5. Додавання імен віддалених репозиторіїв. Одним з найпопулярніших репозиторіїв є github. На ньому необхідно зареєструватись і створити account. Далі в консольному вікні

необхідно виконати команду:

Клонування репозиторію з сервера.

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

Основні команди Git

5. Перевірка стану робочої папки здійснюється за допомогою команди $ git status.

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.

7. 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'.

9. Перегляньте історію змін за допомогою команди git log.

10. Створення нової гілки. Створіть гілку testing за допомогою команди $ git branch testing.

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"

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.

Перегляньте результат об’єднання виправлених файлів в текстовому редакторі.