

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

**[Технології розроблення програмного забезпечення](http://wiki.kpi.ua/index.php/%D0%A2%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%97 %D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F %D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE %D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F_(20102010)" \o "Технології розроблення програмного забезпечення)**

Лабораторна робота №1

«Системи контролю версій. Git.»

Виконав

студент групи ІА–24:

Кармазіна А. В.

Перевірив:

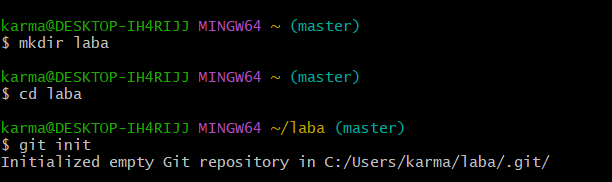
[Мягкий М. Ю.](http://epi.kpi.ua/Schedules/ViewSchedule.aspx?v=2c3407f3-135e-43f1-86f2-3cda6dabcaac" \o "посада Мягкий Михайло Юрійович)

Київ 2024

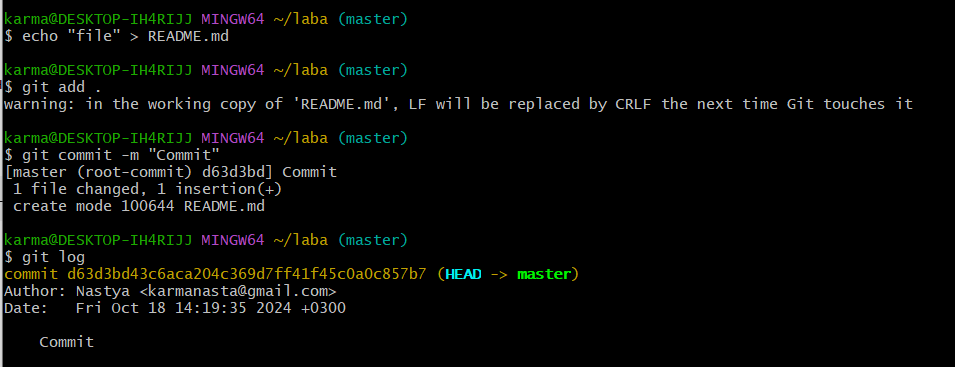
**Мета:** Ознайомитися з основами системи контролю версій Git, навчитися створювати та управляти репозиторіями, використовувати базові та більш складні команди Git для роботи з гілками, комітами, злиттям та історією змін.

**Хід роботи**

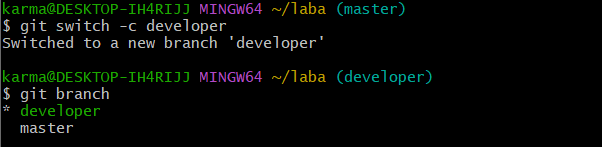
1. Створюємо нову директорію, та переміщуємся в неї.
2. Ініціалізуємо порожній Git репозиторій в поточній директорії. Створює підпапку .git, де будуть зберігатися всі версії файлів і налаштування.



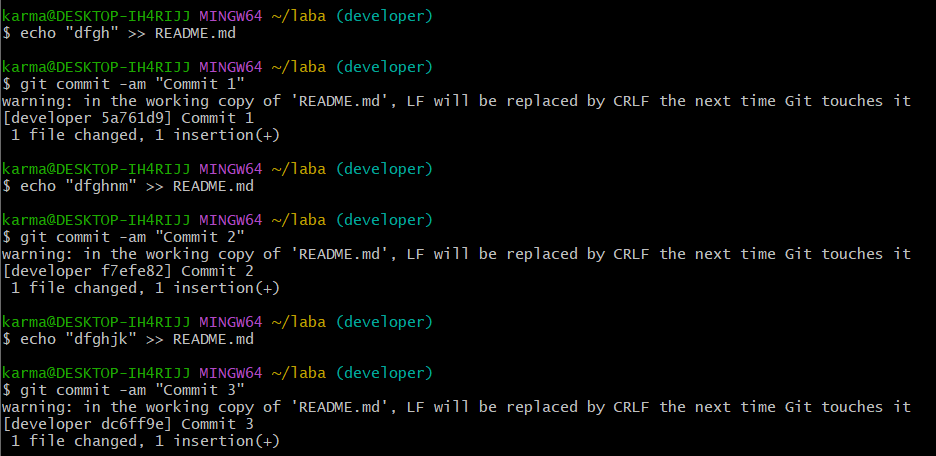
1. Створюємо файл README.md, записуючи в нього текст.
2. Додаємо файл в поточній директорії, готуючи їх до коміту.
3. Створюємо коміт (збереження стану проєкту) з повідомленням "Commit".
4. Виводимо на екран список всіх комітів у поточному репозиторії.



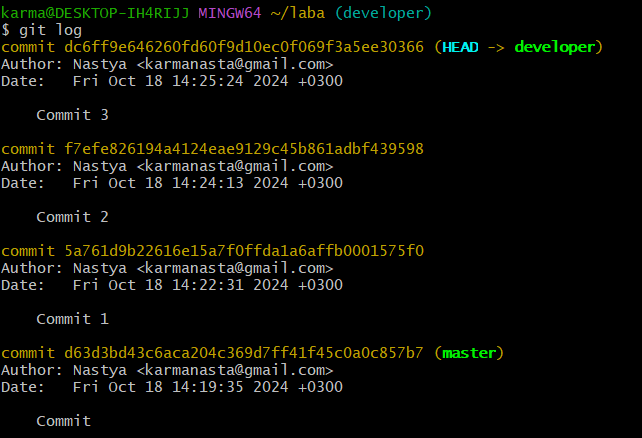
1. Створюємо нову гілку з назвою "developer" та перемикається на неї.
2. Виводимо список всіх гілок у поточному репозиторії, з позначкою \* перед активною гілкою.



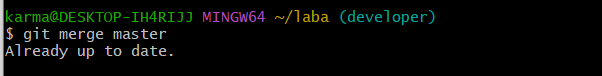
1. Додаємо текст у кінець файлу README.md тричі.
2. Комбінація команд git add та git commit. Дозволяє одночасно додати до індексу всі змінені файли та закомітити їх з повідомленням.

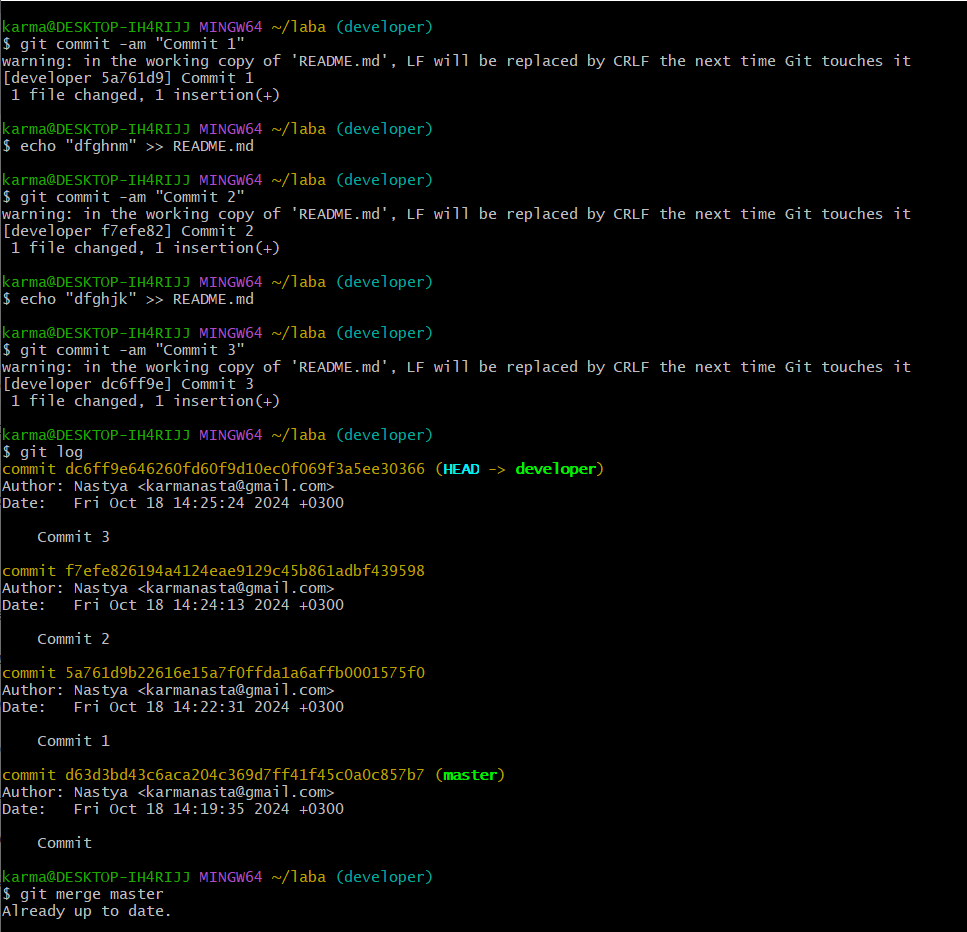
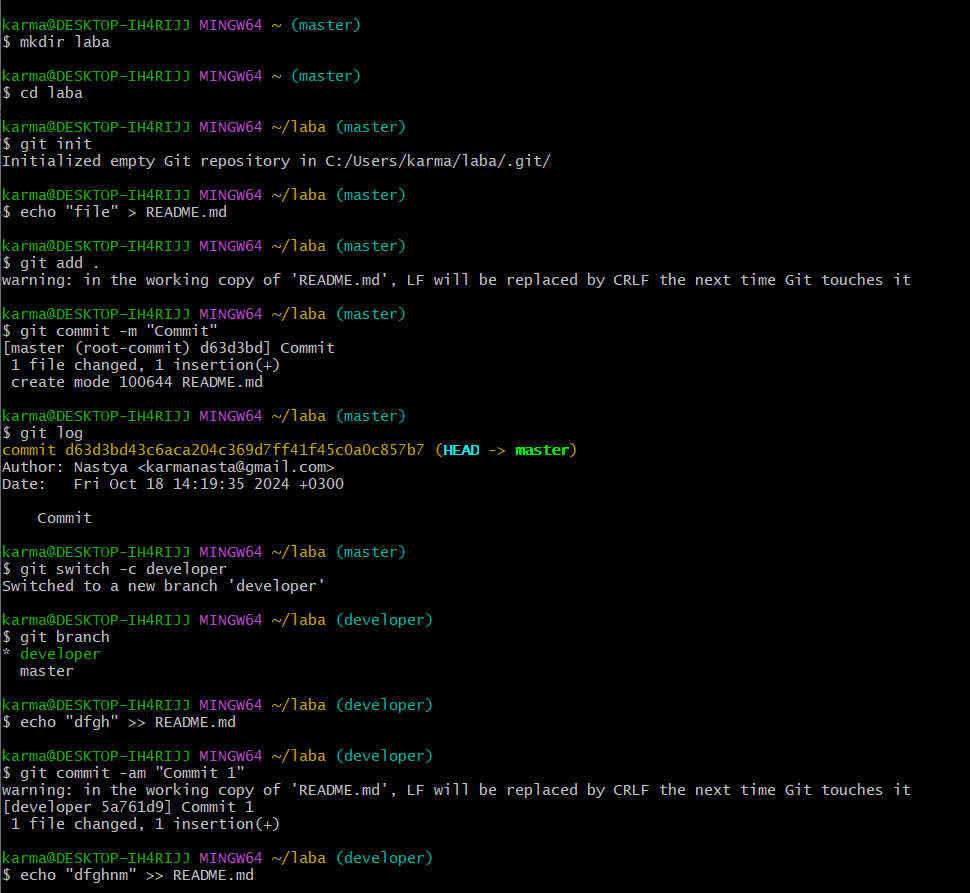


1. Виводимо на екран список всіх комітів у поточному репозиторії, дозволяючи переглянути історію змін.



1. Зливаємо (об'єднуємо) зміни з гілки master у поточну гілку developer. Якщо змін не було, то відобразиться повідомлення "Already up to date"



Команди 

**Висновок:**

У цій лабораторній роботі ми ознайомилися з основними командами Git, що дозволяють управляти версіями проєкту. Ми навчились ініціалізувати репозиторій, створювати файли та фіксувати зміни за допомогою комітів. Також було розглянуто роботу з гілками, зокрема створення нових гілок і злиття змін між ними. Це є важливим аспектом для організації командної роботи та паралельного розвитку різних функціональностей у проєктах.