



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №1
ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ
Системи контролю версій. Розподілена система контролю
версій «Git»

Виконала:
студентка групи ІА-32
Коляда Марія

Київ 2025

Тема: Системи контролю версій. Розподілена система контролю версій «Git».

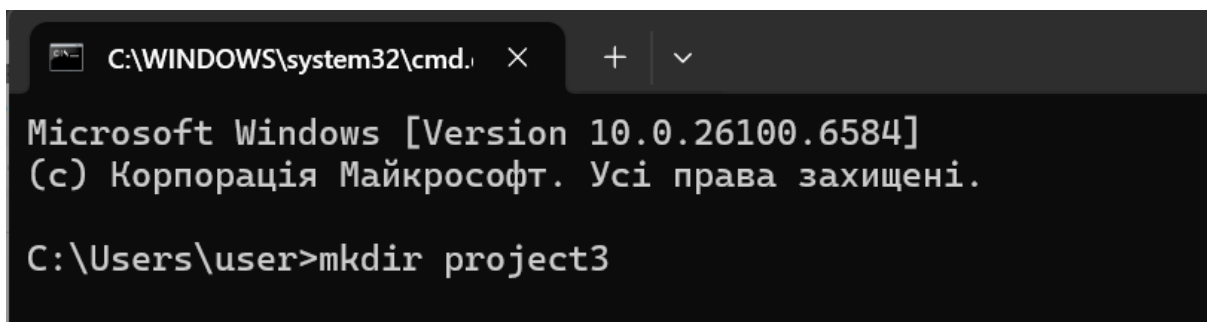
Мета: Навчитися виконувати основні операції в роботі з децентралізованими системами контролю версій на прикладі роботи з сучасною системою Git.

Хід роботи

1. mkdir project3

Створюємо нову теку (папку) з назвою *project3*.

Це майбутня робоча директорія проєкту.

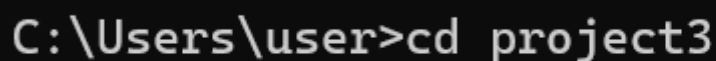


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.26100.6584]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\user>mkdir project3
```

2. cd project3

Переходимо у створену теку, щоб усі наступні команди виконувались саме там.



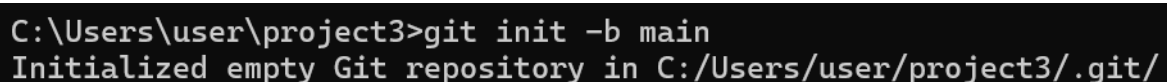
```
C:\Users\user>cd project3
```

3. git init -b main

Ініціалізуємо новий Git-репозиторію у цій теці.

Команда створює приховану папку *.git*, де зберігається історія змін.

Параметр *-b main* одразу задає головну гілку (*main*), замість стандартної *master*.



```
C:\Users\user\project3>git init -b main
Initialized empty Git repository in C:/Users/user/project3/.git/
```

4. echo "File1" > file.txt

Створюємо новий файл *file.txt* і записує в нього текст File1.

Це просто приклад створення початкового файлу в проєкті.

```
C:\Users\user\project3>echo "File1" > file.txt
```

5. **git add file.txt**

Додаємо файл *file.txt* до «індексу» (staging area), тобто готує його до збереження в історії змін (коміту)

```
C:\Users\user\project3>git add file.txt
```

6. **git commit -m "Commit1"**

Фіксуємо зміни, що були додані командою `git add`, створюючи перший коміт (Commit1).

```
C:\Users\user\project3>git commit -m "Commit1"
[main (root-commit) 9de6b27] Commit1
1 file changed, 1 insertion(+)
create mode 100644 file.txt
```

7. **git branch branch1**

Створюємо нову гілку (branch) з назвою branch1. `git branch` показує, що існують дві гілки – `main` (активна) і `branch1`.

```
C:\Users\user\project3>git branch branch1

C:\Users\user\project3>git branch
branch1
* main
```

8. **git checkout -b branch2**

Створюємо нову гілку `branch2` і одразу переходимо у неї.

```
C:\Users\user\project3>git checkout -b branch2
Switched to a new branch 'branch2'

C:\Users\user\project3>git branch
  branch1
* branch2
  main
```

9. **git checkout branch1 → echo "зміна 1" >> file.txt → git add . → git commit -m "зміна 1"**

Переходимо у гілку branch1, додаємо новий рядок "зміна 1" у файл і створюємо коміт.

```
C:\Users\user\project3>git checkout branch1
Switched to branch 'branch1'

C:\Users\user\project3>echo "зміна 1" >> file.txt

C:\Users\user\project3>git add .
```

```
C:\Users\user\project3>git commit -m "зміна 1"
[branch1 b15d396] зміна 1
 1 file changed, 1 insertion(+)
```

10. **git checkout branch2 → echo "зміна2" >> file.txt → git add . → git commit -m "зміна2 branch"**

Аналогічно – у гілці branch2 додаємо зміну "зміна2".

```
C:\Users\user\project3>git checkout branch2
Switched to branch 'branch2'

C:\Users\user\project3>echo "зміна2" >> file.txt

C:\Users\user\project3>git add .
```

```
C:\Users\user\project3>git commit -m "зміна2 branch"
[branch2 ceb00c8] зміна2 branch
1 file changed, 1 insertion(+)
```

11.git merge branch1

Спроба об'єднати зміни з branch1 у поточну гілку branch2.

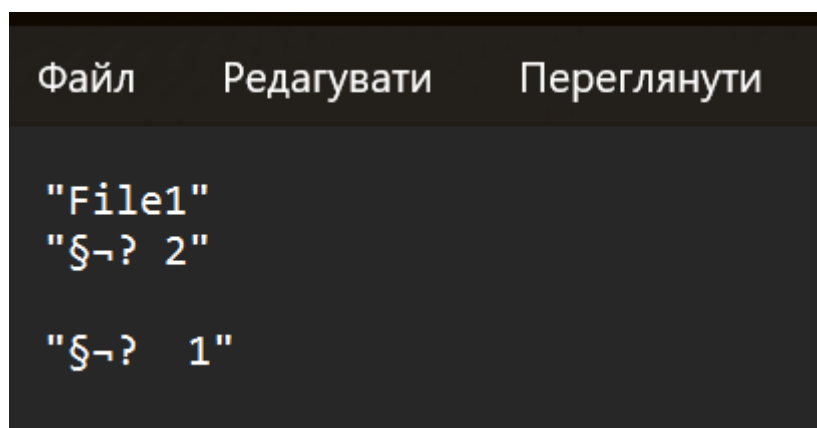
В результаті виникає конфлікт (conflict), бо один і той самий файл (file.txt) змінювався в обох гілках по-різному.

```
C:\Users\user\project3>git merge branch1
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
```

12.notepad file.txt

Відкриваємо файл, щоб вручну виправити конфлікт.

```
C:\Users\user\project3>notepad file.txt
```



13.. git add file.txt → git merge --continue

Після ручного виправлення конфлікту зміни додаємо до індексу і завершується злиття.

```
C:\Users\user\project3>git add file.txt

C:\Users\user\project3>git merge --continue
[branch2 24123e3] Merge branch 'branch1' into branch2
```

Висновок: під час виконання роботи було опановано основні операції роботи з децентралізованою системою контролю версій Git. Було створено новий репозиторій, додано файли та зафіксовано зміни за допомогою команди `git commit`. Також створено окремі гілки (`branch1` і `branch2`), у яких виконувалися незалежні зміни файлів. Під час об'єднання гілок відпрацьовано процес злиття (`merge`) та вирішення конфліктів, що виникають при різних версіях одного файлу. Отримано практичні навички створення, додавання, фіксації, злиття змін і роботи з гілками в Git.