



Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота № 1
з дисципліни «Технології розроблення програмного
забезпечення»
Тема: «**Git**»

Виконала
студентка групи ІА-34
Мартинюк Т.В.

Перевірів
Мягкий М.Ю.

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.

Теоретичні відомості

Система управління версіями (від англ. Version Control System або Source Control System) – програмне забезпечення яке призначено допомогти команді розробників керувати змінами в вихідному коді під час роботи [1]. Система керування версіями дозволяє додавати зміни в файлах в репозиторій і таким чином після кожної фіксації змін мани нову ревізію файлів. Це дозволяє повертатися до попередніх версій коду для аналізу внесених змін або пошуку, які зміни привели до появи помилки. Таким чином можна знайти хто, коли і які зміни зробив в коді, а також чому ці зміни були зроблені

Гіт є системою розподіленого контролю версій, коли кожен розробник має власний репозиторій, куди він вносить зміни. Далі система гіт синхронізує репозиторії із центральним репозиторієм. Це дозволяє проводити роботу незалежно від центрального репозиторію (на відміну від SVN, коли версіонування передбачало наявність зв'язку з центральним сервером), перекладає складності ведення гілок та склеювання змін більше на плечі системи, ніж розробників та ін. Зміни зберігаються у вигляді наборів змін (changeset), що отримує унікальний ідентифікатор (хеш-сума на основі самих змін).

Основна ідея Git, як і будь-якої іншої розподіленої системи контролю версій – кожен розробник має власний репозиторій, куди складаються зміни (версії) файлів, та синхронізація між розробниками виконується за допомогою синхронізації репозиторіїв. Відповідно, є ряд основних команд для роботи:

1. Клонувати репозиторій (`git clone`) – отримати копію репозиторію на локальну машину для подальшої роботи з ним;
2. Синхронізація репозиторіїв (`git fetch` або `git pull`) – отримання змін із віддаленого (вихідного, центрального, або будь-якого іншого такого ж) репозиторію;
3. Фіксація змін в репозиторій (`git commit`) – фіксація виконаних змін в програмному коді в локальний репозиторій розробника;
4. Синхронізація репозиторіїв (`git push`) – переслати зміни – `push` – передача власних змін до віддаленого репозиторію – Записати зміни – `commit` – створення нової версії;
5. Оновитись до версії – `update` – оновитись до певної версії, що є у репозиторії.
6. Об'єднання гілок (`git merge`) – об'єднання вказаною гілки в поточну (часто ще називається «злиттям»).

Хід роботи

1. Створюємо директорію та переходимо в неї.

```
C:\Users\Тетяна>mkdir lab11  
  
C:\Users\Тетяна>cd lab11
```

2. Створюємо локальний репозиторій.

```
C:\Users\Тетяна\lab11>git init  
Initialized empty Git repository in C:/Users/Тетяна/lab11/.git/  
  
C:\Users\Тетяна\lab11>git status  
On branch master  
  
No commits yet  
  
nothing to commit (create/copy files and use "git add" to track)
```

3. Робимо перший пустий запис у Git.

```
C:\Users\Тетяна\lab11>git commit --allow-empty -m "empty commit added"  
[master (root-commit) ff7bd69] empty commit added
```

4. Створюємо 2 гілки різними способами.

```
C:\Users\Тетяна\lab11>git branch b1  
  
C:\Users\Тетяна\lab11>git checkout -b b2  
Switched to a new branch 'b2'
```

5. Додаємо два файли у директорію.

```
C:\Users\Тетяна\lab11>echo "1" > f1.txt  
  
C:\Users\Тетяна\lab11>echo "2" > f2.txt
```

6. Зафіксуємо зміни, додавши файл f1.txt в гілку b1, а f2.txt в гілку b2.

```
C:\Users\Тетяна\lab11>git add .

C:\Users\Тетяна\lab11>git commit -m "f2 added" f2.txt
[b2 06498d9] f2 added
1 file changed, 1 insertion(+)
create mode 100644 f2.txt

C:\Users\Тетяна\lab11>git checkout b1
A      f1.txt
Switched to branch 'b1'

C:\Users\Тетяна\lab11>git commit -m "f1 added" f1.txt
[b1 8b2bdca] f1 added
1 file changed, 1 insertion(+)
create mode 100644 f1.txt
```

7. Переглядаємо історію змін у вигляді графа.

```
C:\Users\Тетяна\lab11>git log --graph
* commit 8b2bdca9111ddb63ab5d7d1fbaf94ed618e4fef4 (HEAD -> b1)
  Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
  Date:   Sat Sep 13 17:01:30 2025 +0300

      f1 added

* commit ff7bd6960c725689d5ceb0a5d7c0a7c85c900b3a (master)
  Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
  Date:   Sat Sep 13 16:59:07 2025 +0300

      empty commit added

C:\Users\Тетяна\lab11>git log --graph --all
* commit 8b2bdca9111ddb63ab5d7d1fbaf94ed618e4fef4 (HEAD -> b1)
  Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
  Date:   Sat Sep 13 17:01:30 2025 +0300

      f1 added

* commit 06498d9709cb4a4330feb261d28e6669f53ce0f4 (b2)
  Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
  Date:   Sat Sep 13 17:01:01 2025 +0300

      f2 added

* commit ff7bd6960c725689d5ceb0a5d7c0a7c85c900b3a (master)
  Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
  Date:   Sat Sep 13 16:59:07 2025 +0300

      empty commit added
```

8. Додаємо файл f2.txt в гілку b1 та фіксуємо зміни.

```
C:\Users\Тетяна\lab11>git add f2.txt

C:\Users\Тетяна\lab11>git commit -m "f2 adedd to b1" f2.txt
[b1 3f4e62e] f2 adedd to b1
1 file changed, 1 insertion(+)
create mode 100644 f2.txt
```

9. Намагаємось об'єднати гілки b1 та b2.

```
C:\Users\Тетяна\lab11>git merge b2
Auto-merging f2.txt
CONFLICT (add/add): Merge conflict in f2.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Отримуємо конфлікт через те, що один і той самий файл змінений у двох гілках, та вирішуємо його вручну.

10. Продовжуємо злиття гілок b1 та b2.

```
C:\Users\Тетяна\lab11>git add .

C:\Users\Тетяна\lab11>git merge --continue
[b1 a576377] Merge branch 'b2' into b1
```

11. Переглядаємо історію змін

```
C:\Users\Тетяна\lab11>git log --all --graph
*   commit a576377eae5534b7130c364391ec2fc4bf53112a (HEAD -> b1)
|   Merge: 3f4e62e 06498d9
|   Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
|   Date:   Sat Sep 13 17:06:03 2025 +0300
|
|       Merge branch 'b2' into b1
|
|   *   commit 06498d9709cb4a4330feb261d28e6669f53ce0f4 (b2)
|   |   Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
|   |   Date:   Sat Sep 13 17:01:01 2025 +0300
|   |
|   |       f2 added
|   |
|   *   commit 3f4e62e2c3aa9d8425465a550989e43ab1aac206
|   |   Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
|   |   Date:   Sat Sep 13 17:03:14 2025 +0300
|   |
|   |       f2 added to b1
|   |
|   *   commit 8b2bdca9111ddb63ab5d7d1fba94ed618e4fed
|   |   Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
|   |   Date:   Sat Sep 13 17:01:30 2025 +0300
|   |
|   |       f1 added
|   |
|   *   commit ff7bd6960c725689d5ceb0a5d7c0a7c85c900b3a (master)
|   |   Author: tetyanamartyniuk <145157063+tetyanamartyniuk@users.noreply.github.com>
|   |   Date:   Sat Sep 13 16:59:07 2025 +0300
|   |
|   |       empty commit added
```

Висновок: Під час виконання цієї роботи ми навчилися виконувати основні операції в роботі з git, зокрема створювати гілки, виконувати коміти, об'єднувати гілки, а також вирішувати конфлікти.