



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
із предмету Технології розроблення програмного забезпечення
‘Основи роботи з Git’

Виконала:
студентка групи ІА-33
Маляревич А. В.

Перевірів: Мягкий Михайло Юрійович

Київ 2025

Тема: Основи роботи з Git: створення, коміти, гілки та злиття

Мета: Ознайомитися з базовими можливостями системи Git, навчитися створювати та перемикатися між гілками, виконувати коміти, проводити злиття гілок та розв'язувати конфлікти, що виникають у процесі злиття.

Теоретичні відомості

`git init` – Створює новий локальний репозиторій у поточній папці.

`git status` – Відображає поточний стан репозиторію.

`git add` – Додає файл у *staging area* (індекс) для підготовки до коміту.

`git commit -m "повідомлення"` – Створює новий коміт із внесеними змінами.

`git branch <назва>` – Створює нову гілку.

`git checkout -b <назва>` а б о `git switch -c <назва>` – Створює нову гілку і відразу перемикає на неї.

`git checkout <назва>` а б о `git switch <назва>` – Перемикає на існуючу гілку.

`git merge <назва>` – Виконує злиття вибраної гілки в поточну.

`git log` – Показує історію комітів.

`git log --graph --oneline --all` – Виводить історію з візуалізацією гілок та злиттів.

Хід роботи

1. Створити новий каталог та ініціалізувати порожній Git-репозиторій.
2. Створити основну гілку `main` і зробити в ній перший (порожній) коміт.
3. Створити три додаткові гілки: `b1`, `b2`, `b3`.
4. У гілці `b3` створити три файли (`f1.txt`, `f2.txt`, `f3.txt`), додати їх до індексу та закомітити лише файл `f3.txt`.
5. Перейти на гілку `b1`, додати і закомітити файл `f1.txt`.
6. Перейти на гілку `b2`, додати і закомітити файл `f2.txt`.

7. Виконати перегляд історії комітів за допомогою команди `git log --all` та переконатися, що у кожній гілці збережені свої зміни.
8. Внести зміни у файл `f1.txt` у гілці `b2` та створити новий коміт.
9. Виконати злиття гілки `b1` у гілку `b2`.
10. Переглянути граф комітів за допомогою `git log --graph`.

Використання команд представлено у фото нижче:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\anzhe> mkdir hehehe

Directory: C:\Users\anzhe

Mode                LastWriteTime         Length Name
----                -
d-----          9/6/2025   3:13 PM             hehehe

PS C:\Users\anzhe> cd hehehe
PS C:\Users\anzhe\hehehe> git init
Initialized empty Git repository in C:/Users/anzhe/hehehe/.git/
PS C:\Users\anzhe\hehehe> git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
PS C:\Users\anzhe\hehehe> git checkout -b main
Switched to a new branch 'main'
PS C:\Users\anzhe\hehehe> git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
PS C:\Users\anzhe\hehehe> git commit --allow-empty -m "init"
[main (root-commit) 1c48ebb] init
PS C:\Users\anzhe\hehehe> git branch b1
PS C:\Users\anzhe\hehehe> git checkout -b b2
Switched to a new branch 'b2'
PS C:\Users\anzhe\hehehe> git checkout main
Switched to branch 'main'
PS C:\Users\anzhe\hehehe> git switch -c b3
Switched to a new branch 'b3'
PS C:\Users\anzhe\hehehe> echo "1" > f1.txt
PS C:\Users\anzhe\hehehe> echo "2" > f2.txt
PS C:\Users\anzhe\hehehe> echo "3" > f3.txt
PS C:\Users\anzhe\hehehe> git status
On branch b3
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        f1.txt
        f2.txt
        f3.txt

nothing added to commit but untracked files present (use "git add" to track)
```

```

PS C:\Users\anzhe\hehehe> git add .
PS C:\Users\anzhe\hehehe> git status
On branch b3
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f1.txt
    new file:   f2.txt
    new file:   f3.txt

PS C:\Users\anzhe\hehehe> git commit -m "f3" f3.txt
[b3 bcd7f1e] f3
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f3.txt
PS C:\Users\anzhe\hehehe> git status
On branch b3
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f1.txt
    new file:   f2.txt

PS C:\Users\anzhe\hehehe> git checkout b1
A      f1.txt
A      f2.txt
Switched to branch 'b1'
PS C:\Users\anzhe\hehehe> git commit -m "f1" f1.txt
[b1 b5761ff] f1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f1.txt
PS C:\Users\anzhe\hehehe> git checkout b2
A      f2.txt
Switched to branch 'b2'
PS C:\Users\anzhe\hehehe> git commit -m "f2" f2.txt
[b2 699d073] f2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f2.txt

```

```

PS C:\Users\anzhe\hehehe> git log --all
commit 699d0739e1edbb18cdc7e4f383e3dfcaf1da6005 (HEAD -> b2)
Author: Anzhela <anzhela.en.mal@gmail.com>
Date:   Sat Sep 6 15:21:07 2025 +0300

    f2

commit b5761ff86e39886612ec1af0a5e3101c98db73b9 (b1)
Author: Anzhela <anzhela.en.mal@gmail.com>
Date:   Sat Sep 6 15:20:46 2025 +0300

    f1

commit bcd7f1eabbcfae58afbd43c4b58b258f8ed68f07 (b3)
Author: Anzhela <anzhela.en.mal@gmail.com>
Date:   Sat Sep 6 15:19:46 2025 +0300

    f3

commit 1c48ebb4ae83a7b4f92007d3efc8ed179025e16e (main)
Author: Anzhela <anzhela.en.mal@gmail.com>
Date:   Sat Sep 6 15:15:03 2025 +0300

    init
PS C:\Users\anzhe\hehehe> echo "22"
22
PS C:\Users\anzhe\hehehe>

```

```

PS C:\Users\anzhe\hehehe> echo "22" > f1.txt
PS C:\Users\anzhe\hehehe> git add .
PS C:\Users\anzhe\hehehe> git commit -m "f1 to b2"
[b2 6a90c80] f1 to b2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f1.txt
PS C:\Users\anzhe\hehehe> git merge b1
warning: Cannot merge binary files: f1.txt (HEAD vs. b1)
Auto-merging f1.txt
CONFLICT (add/add): Merge conflict in f1.txt
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\anzhe\hehehe> code .
PS C:\Users\anzhe\hehehe> git status
On branch b2
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both added:      f1.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\anzhe\hehehe> cat f1.txt
1

22
PS C:\Users\anzhe\hehehe>

```

```

PS C:\Users\anzhe\hehehe>
PS C:\Users\anzhe\hehehe> git add .
PS C:\Users\anzhe\hehehe> git merge --continue
[b2 6a4ee53] Merge branch 'b1' into b2
PS C:\Users\anzhe\hehehe>
PS C:\Users\anzhe\hehehe> git log --graph
*   commit 6a4ee5309cfe5ceb9e42a60ef7be531a21643916 (HEAD -> b2)
|  Merge: 6a90c80 b5761ff
|  Author: Anzhela <anzhela.en.mal@gmail.com>
|  Date:   Sat Sep 6 15:29:47 2025 +0300
|
|      Merge branch 'b1' into b2
|
| *   commit b5761ff86e39886612ec1af0a5e3101c98db73b9 (b1)
| |  Author: Anzhela <anzhela.en.mal@gmail.com>
| |  Date:   Sat Sep 6 15:20:46 2025 +0300
| |
| |      f1
| |
| *   commit 6a90c8045d498307b0c5c45728ef0ca5335e2c8e
| |  Author: Anzhela <anzhela.en.mal@gmail.com>
| |  Date:   Sat Sep 6 15:25:10 2025 +0300
| |
| |      f1 to b2
| |
| *   commit 699d0739e1edbb18cdc7e4f383e3dfcaf1da6005
| |  Author: Anzhela <anzhela.en.mal@gmail.com>
| |  Date:   Sat Sep 6 15:21:07 2025 +0300
| |
| |      f2
| |
| *   commit 1c48ebb4ae83a7b4f92007d3efc8ed179025e16e (main)
| |  Author: Anzhela <anzhela.en.mal@gmail.com>
| |  Date:   Sat Sep 6 15:15:03 2025 +0300
| |
| |      init
| |
PS C:\Users\anzhe\hehehe>

```

Висновок: У лабораторній роботі я навчилася працювати з гілками в Git, виконувати коміти та злиття, а також розв'язувати конфлікти. Отримала практичні навички відстеження історії змін і структури проекту.