# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **КНІТ**

Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 1

**З дисципліни:** *“Моделювання та аналіз програмного забезпечення”*

**На тему:** *“Налаштування Git”*

**Лектор:** доц. каф. ПЗ Сердюк П.В.

**Виконав:** ст. гр. ПЗ-23 Михалевич П.-І. В.

**Прийняла:** асист. каф. ПЗ Цимбалюк Т.М.

« » 2022 р.

∑= .

Львів – 2022

**Тема роботи:** Налаштування Git.

**Мета роботи:** Навчитися працювати з системою керування репозиторіями програмного коду GitLab.

# TЕОРЕТИЧНІ ВІДОМОСТІ

Git – це система керування версій файлів. Ця система дуже зручна для використання на проектах, оскільки в будь який момент можна повернутися до попередніх версій розробки, також ця система зручна для спільної роботи, можна легко обмінюватися даними з іншими розробниками на проекті. Основний принцип системи полягає у створенні комітів (у комітах зберігається стан проекту у даний момент часу) для збереження вже зробленої частини роботи та у створенні гілок для різноманітних завдань. Для кожного завдання створюється своя гілка, на якій виконується вся розробка, після чого гілка об’єнується з головною і т.д.

Основні комманди:

*git commit –* створити коміт.

*git push –* відіслати зміни на сервер.

*git pull –* стягнути зміни з серверу.

*git branch –* створити нову гілку на основі поточної.

*git merge –* обєднати поточку гілку із заданою.

# ЗАВДАННЯ

1. Переглянути відео по Git.
2. Створити аккаунт на GitLab (тільки GitLab - не GitHub, і не жоден інший).
3. Створити репозиторій назвати MAPZ (тільки така назва).
4. Створити папку на комп’ютері та склонувати репозиторій.
5. Створити структура папок локально на компютері- верхня MAPZ, всередині Lab0, Lab1 ,.....
6. Надати доступ до вашого репозиторію pavlo\_serdyuk, з правами доступу до коду (низькі права не дозволяють переглядати код).
7. Додати посилання на ваш репозиторій у XL файл, доступ до кого має забезпечити староста групи.
8. Проробити усі дії на сайті https://learngitbranching.js.org/ до 5 рівня (Advanced не потрібно) і повторити їх у себе локально на комп’ютері.
9. Усі дії робити у папці Lab0. Для комітів використовувати текстові файли, щоб зробити новий коміт, достатньо змінити лінійку тексту всередині файлу.
10. Для кожного завдання використовуйте новий текстовий файл.
11. Звіти повинні містити скріншоти з операціями - для цього можна використати лог, що відобразить історію, або любу візуальну надбудову Git - GitKraken, SourceTree, TortoiseGit.
12. Звіти завантажити на ВНС.

# ХІД ВИКОНАННЯ

1.Переглянув відео.

2.Створив аккаунт на GitLab ([https://gitlab.com/pavlo- ivan.mykhalevych.pz.2020](https://gitlab.com/pavlo-%20ivan.mykhalevych.pz.2020)).

3.Створив репозиторій ([https://gitlab.com/pavlo- ivan.mykhalevych.pz.2020](https://gitlab.com/pavlo-%20ivan.mykhalevych.pz.2020)/mapz).

4-7. Створив репозиторій локально, дав дозвіл та додав посилання

# 7-11.Завдання з сайту:

**Завдання 1.**

**Пункт 1.**

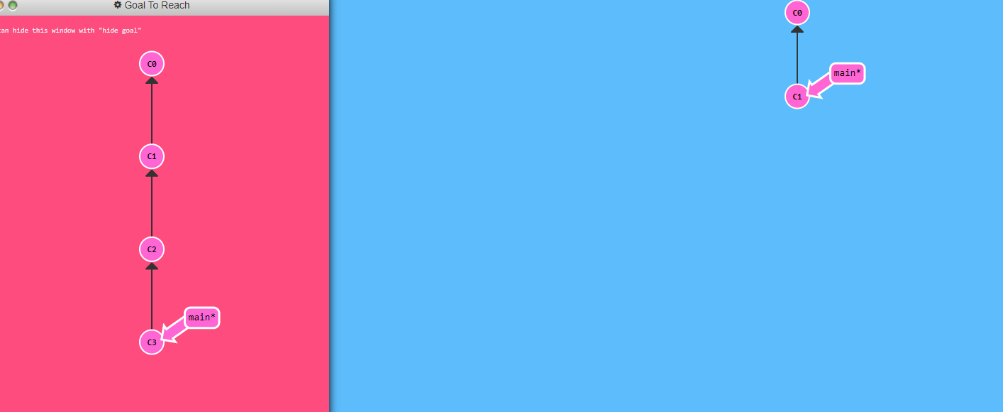
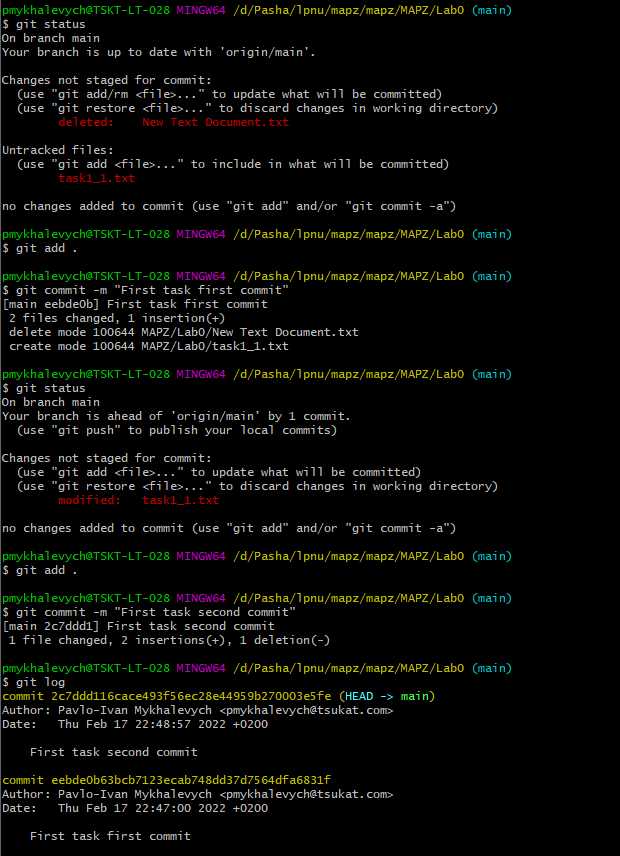
Створення перших комітів.

Рис. 1.1.1. Завдання до 1 завдання 1 пункту

Рис. 1.1.2. Командна стрічка до 1 завдання 1 пункту

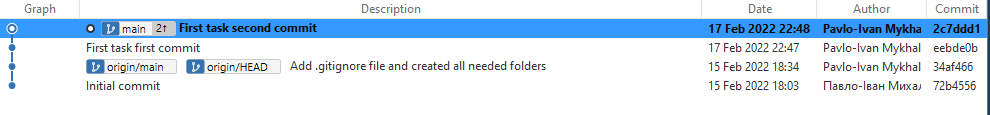


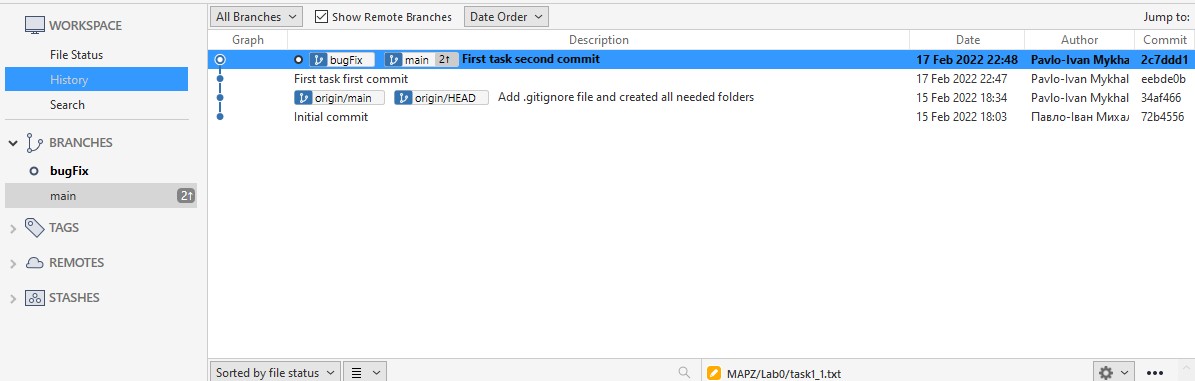
Рис. 1.1.3. Історія комітів до 1 завдання 1 пункту

**Пункт 2.**

Створення нової гілки.



Рис. 1.2.1. Завдання до 1 завдання 2 пункту

 Рис. 1.2.2. Історія комітів до 1 завдання 2 пункту

**Пункт 3.**

Використання команди merge.

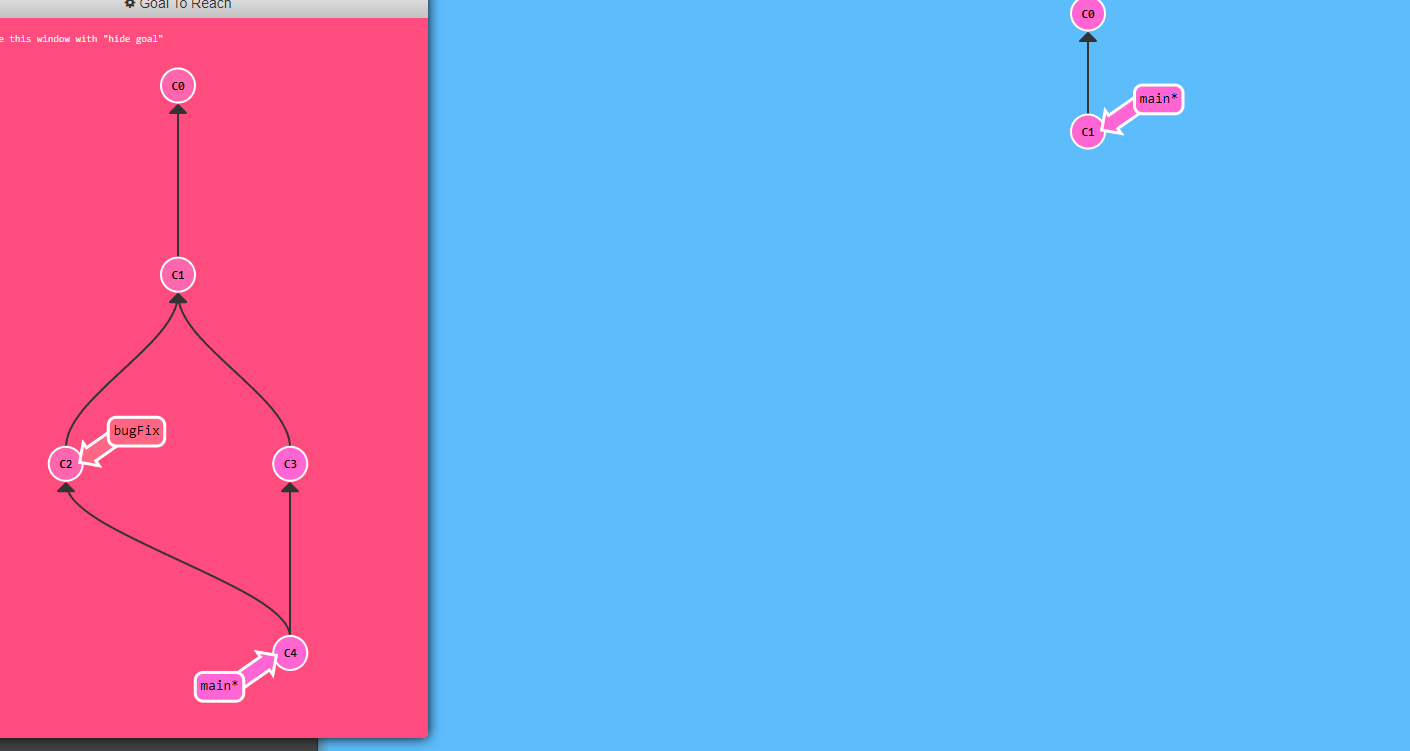
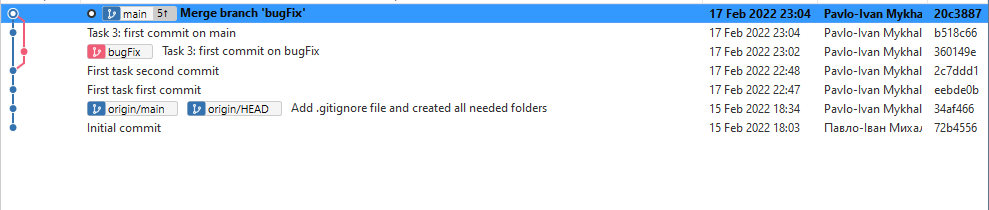


Рис. 1.3.1. Завдання до 1 завдання 3 пункту

 Рис. 1.3.2. Історія комітів до 1 завдання 3 пункту

**Пункт 4.**

Використання команди rebase.

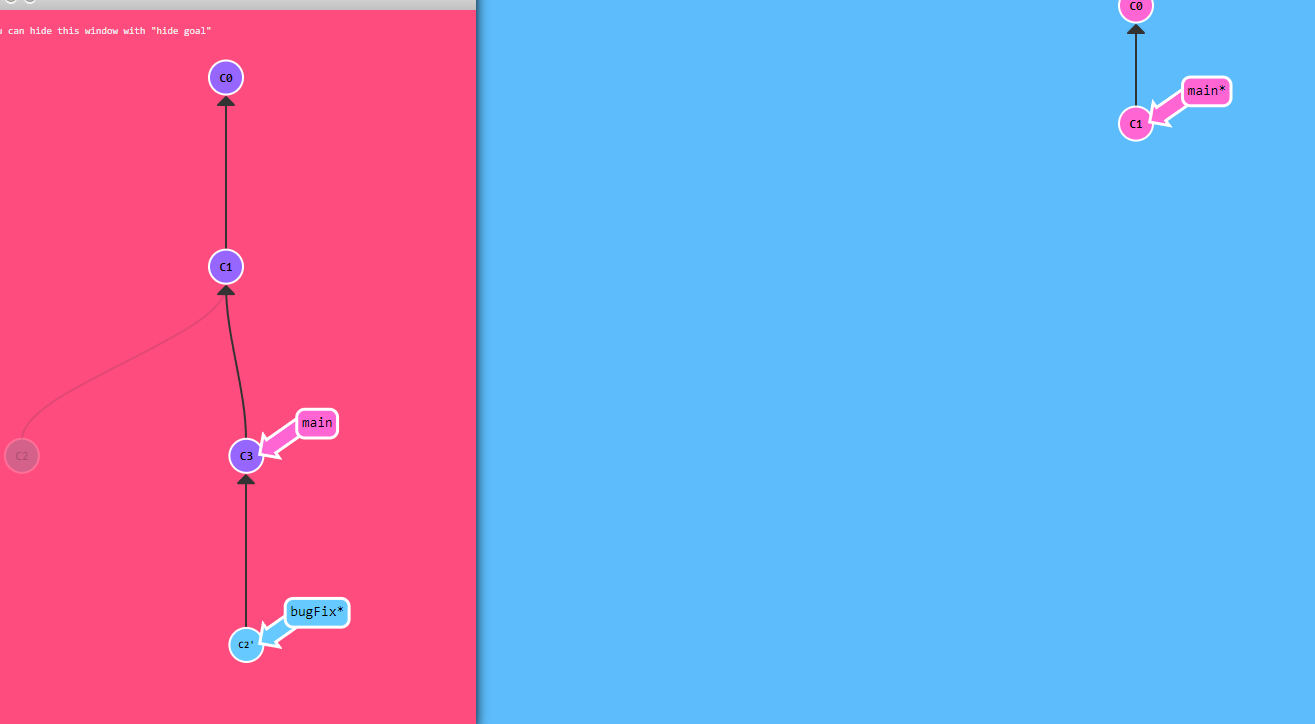
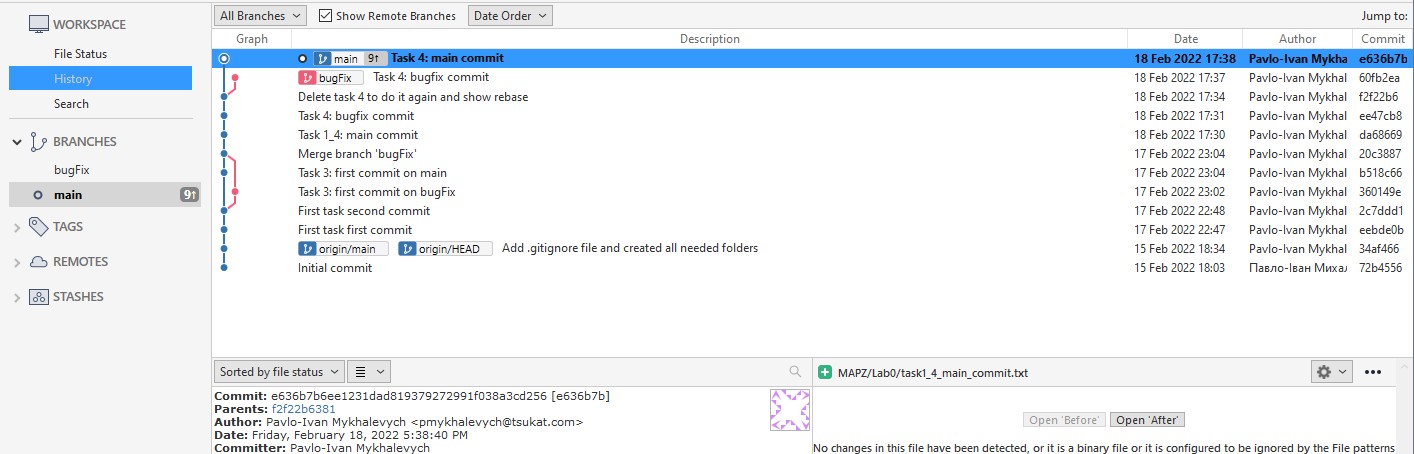


Рис. 1.4.1. Завдання до 1 завдання 4 пункту

Рис. 1.4.2. Історія комітів до виконання 1 завдання 4 пункту

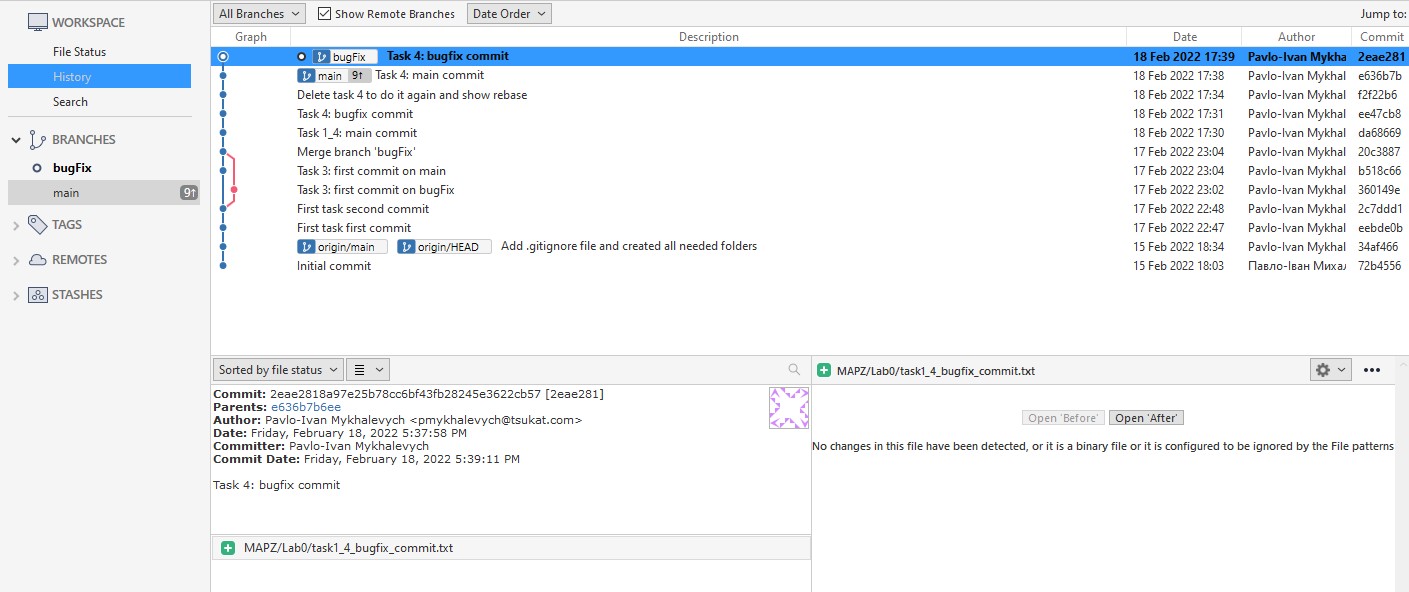


Рис. 1.4.3. Історія комітів після виконання 1 завдання 4 пункту

# Завдання 2.

**Пункт** **1.**

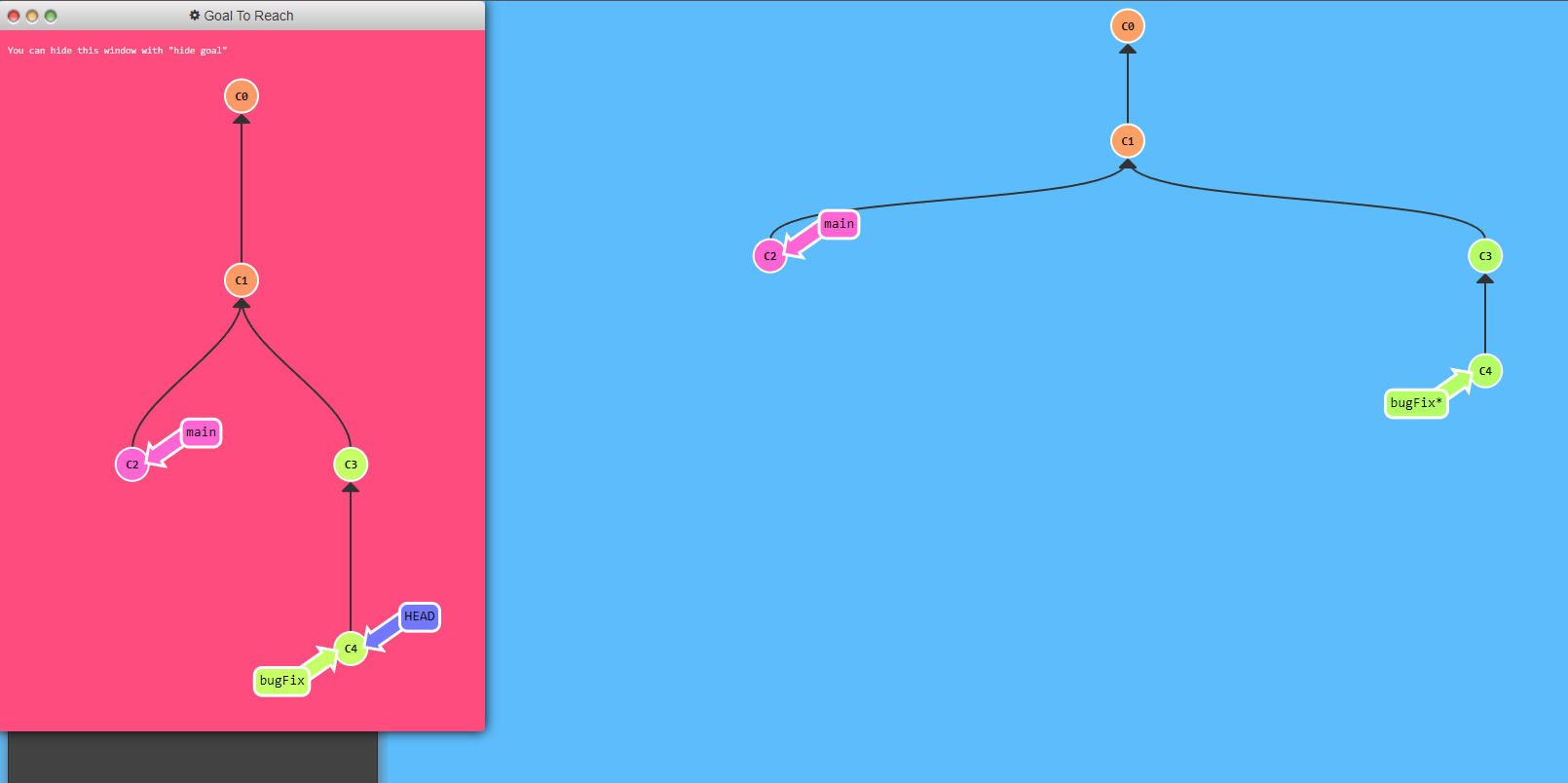
Detach HEAD. 

Рис. 2.1.1. Завдання до 2 завдання 1 пункту

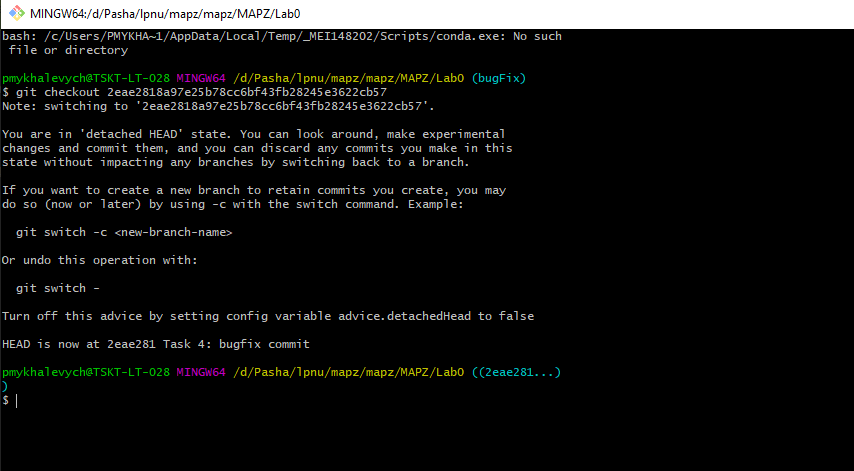


Рис. 2.1.2. Командна стрічка до 2 завдання 1 пункту

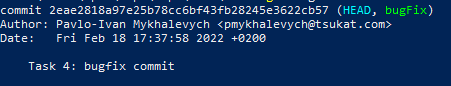
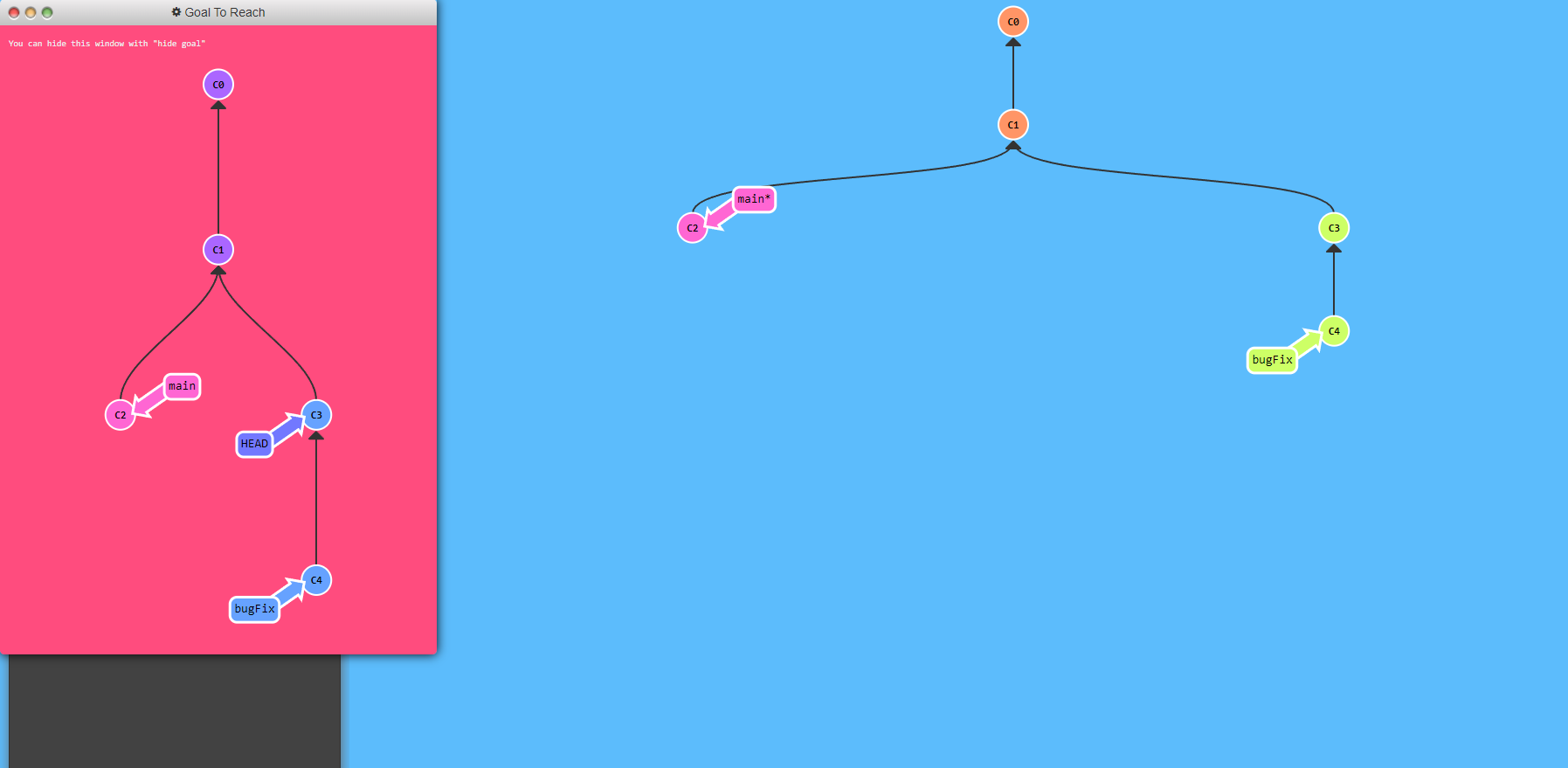


Рис. 2.1.2. На що вказує HEAD після виконання завдання

**Пункт 2.**

Команда ^ та ~.

 Рис. 2.2.1. Завдання до 2 завдання 2 пункту

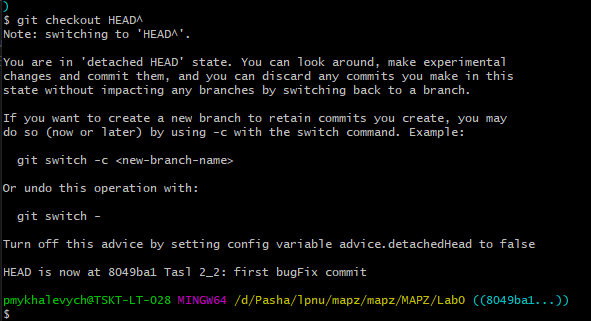
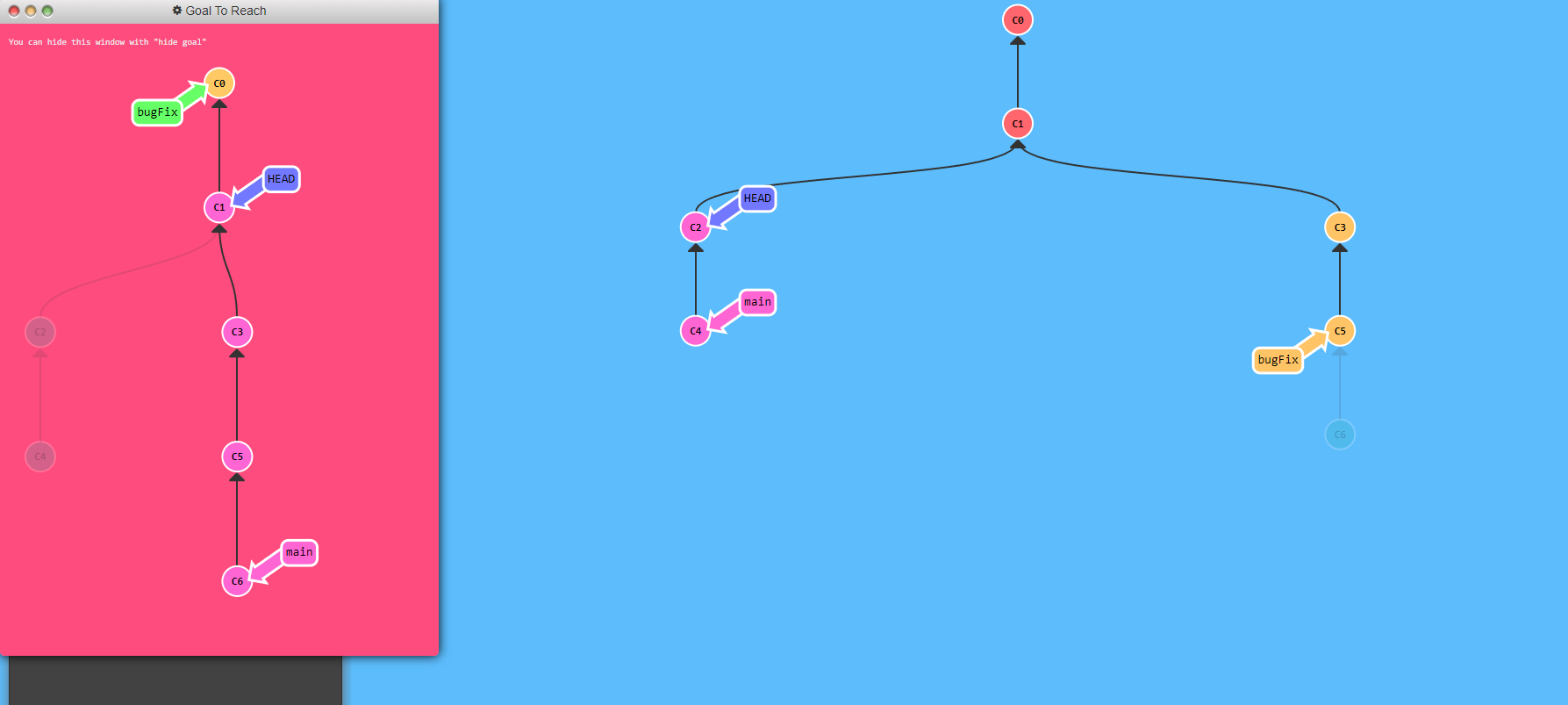


Рис. 2.2.2. Виконання 2 завдання 2 пункту

**Пункт 3.**

Насильне перенесення гілок.

 Рис. 2.3.1. Завдання до 2 завдання 3 пункту

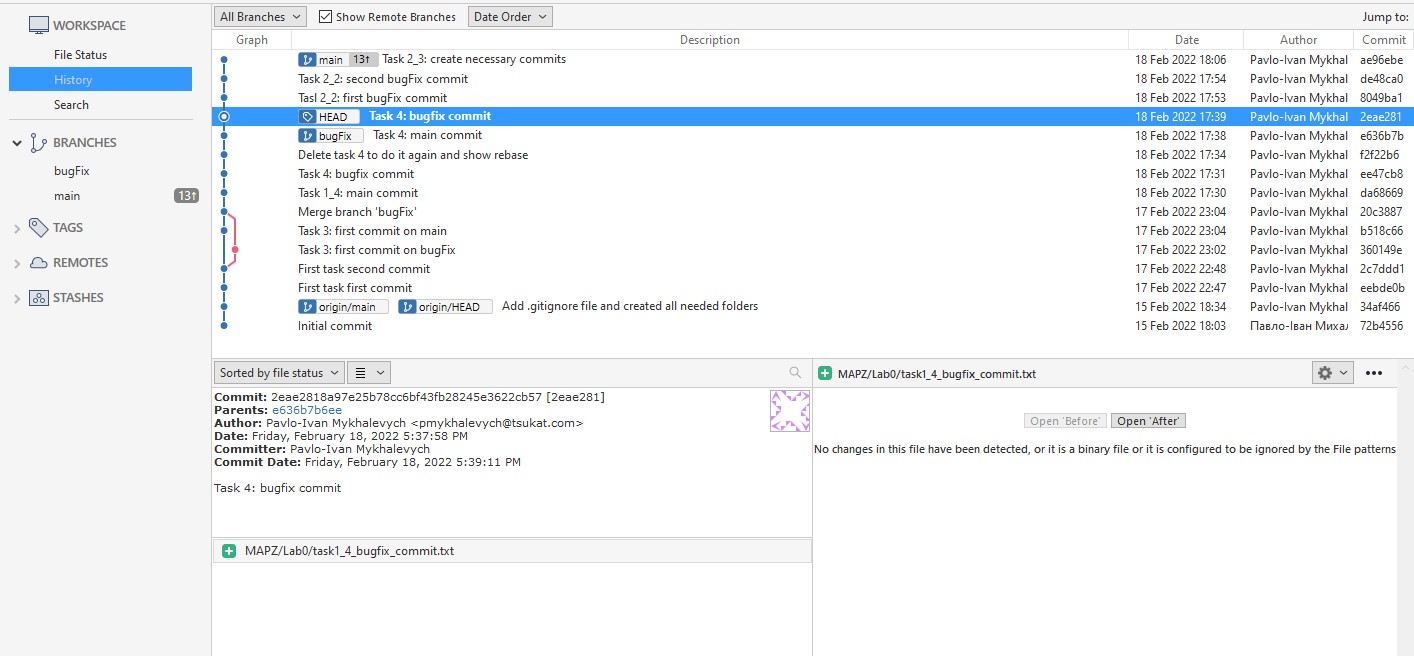
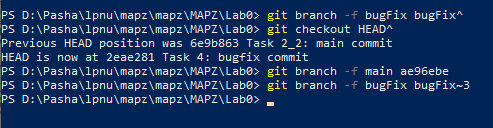
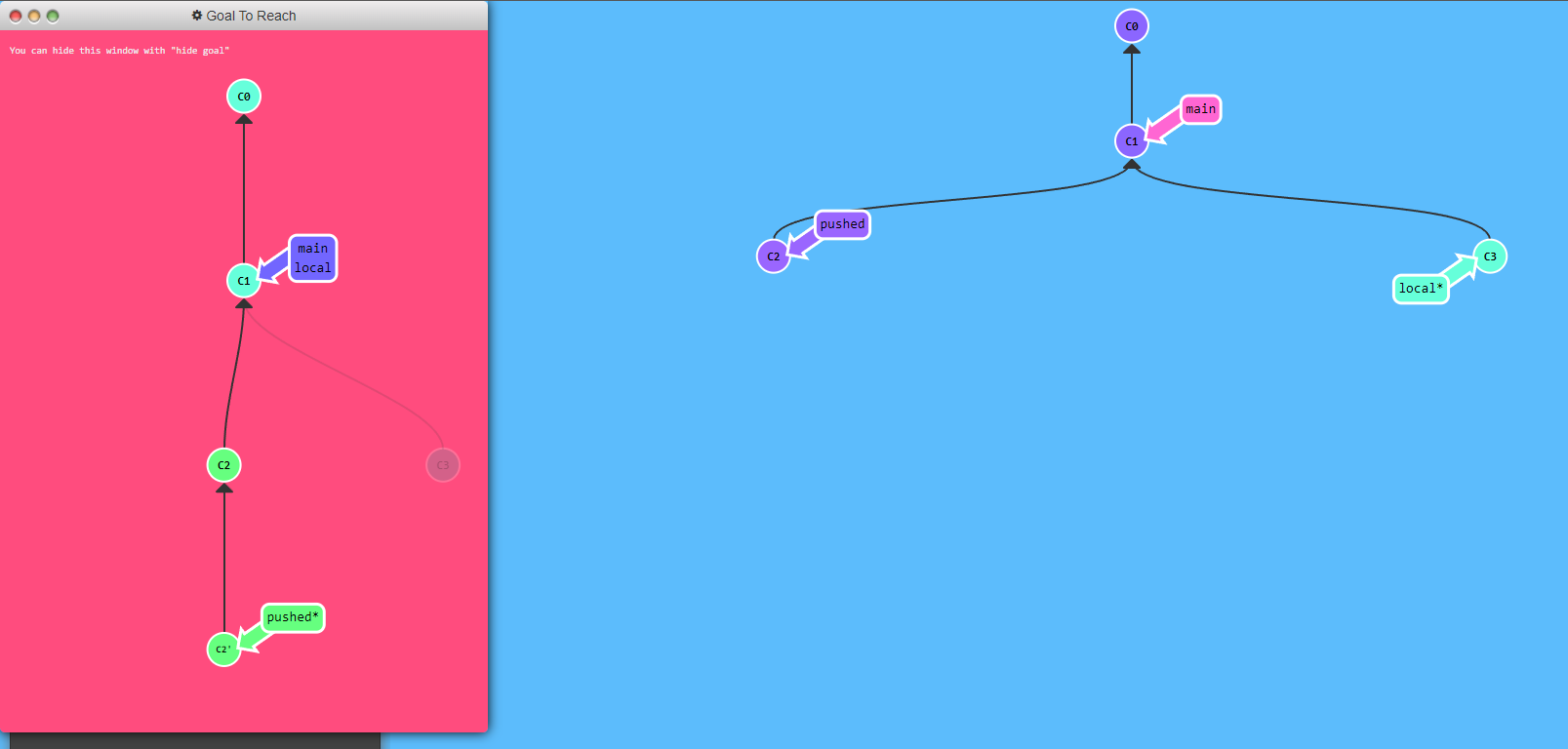


Рис. 2.3.2. Історія комітів та виконання 2 завдання 3 пункту

**Пункт 4.**

Використання команд revert та reset.

 Рис. 2.4.1. Завдання до 2 завдання 4 пункту

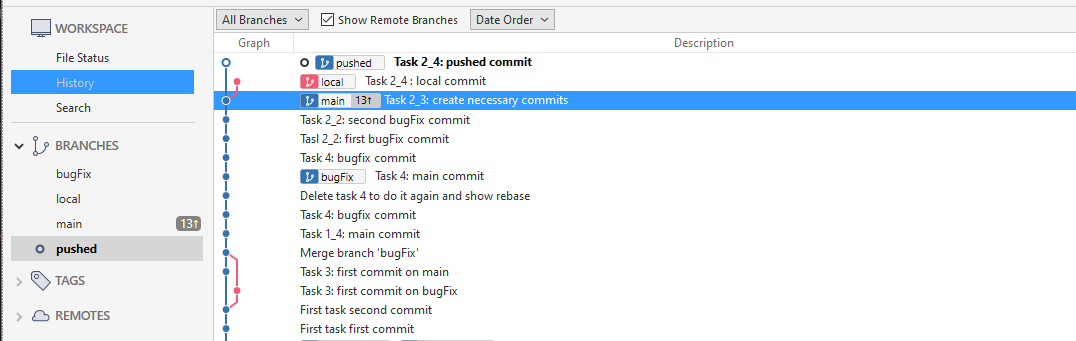


Рис. 2.4.2. Історія комітів до виконання 2 завдання 4 пункту

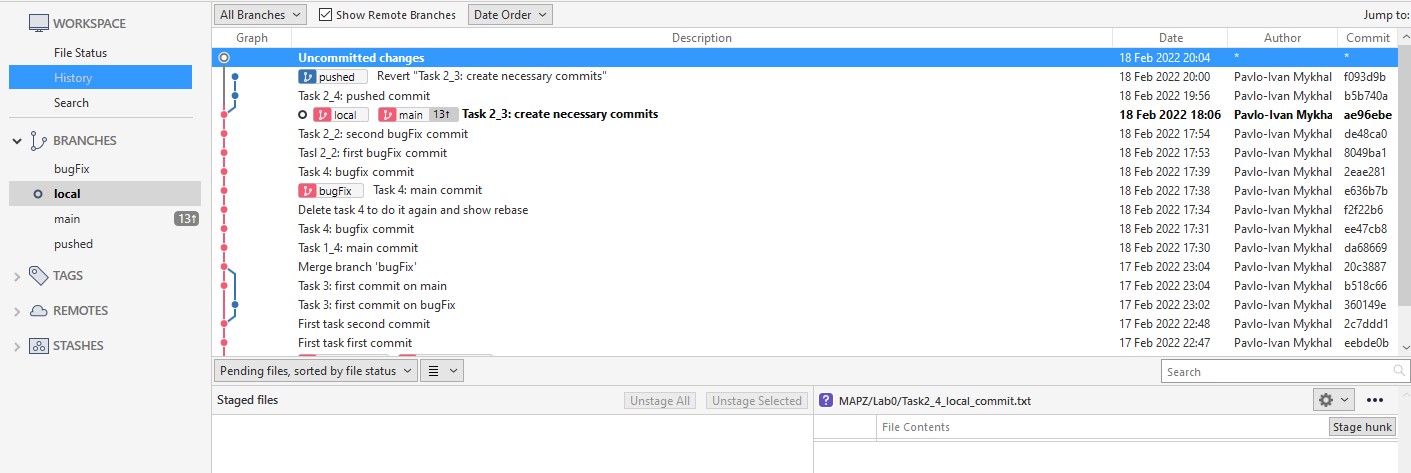
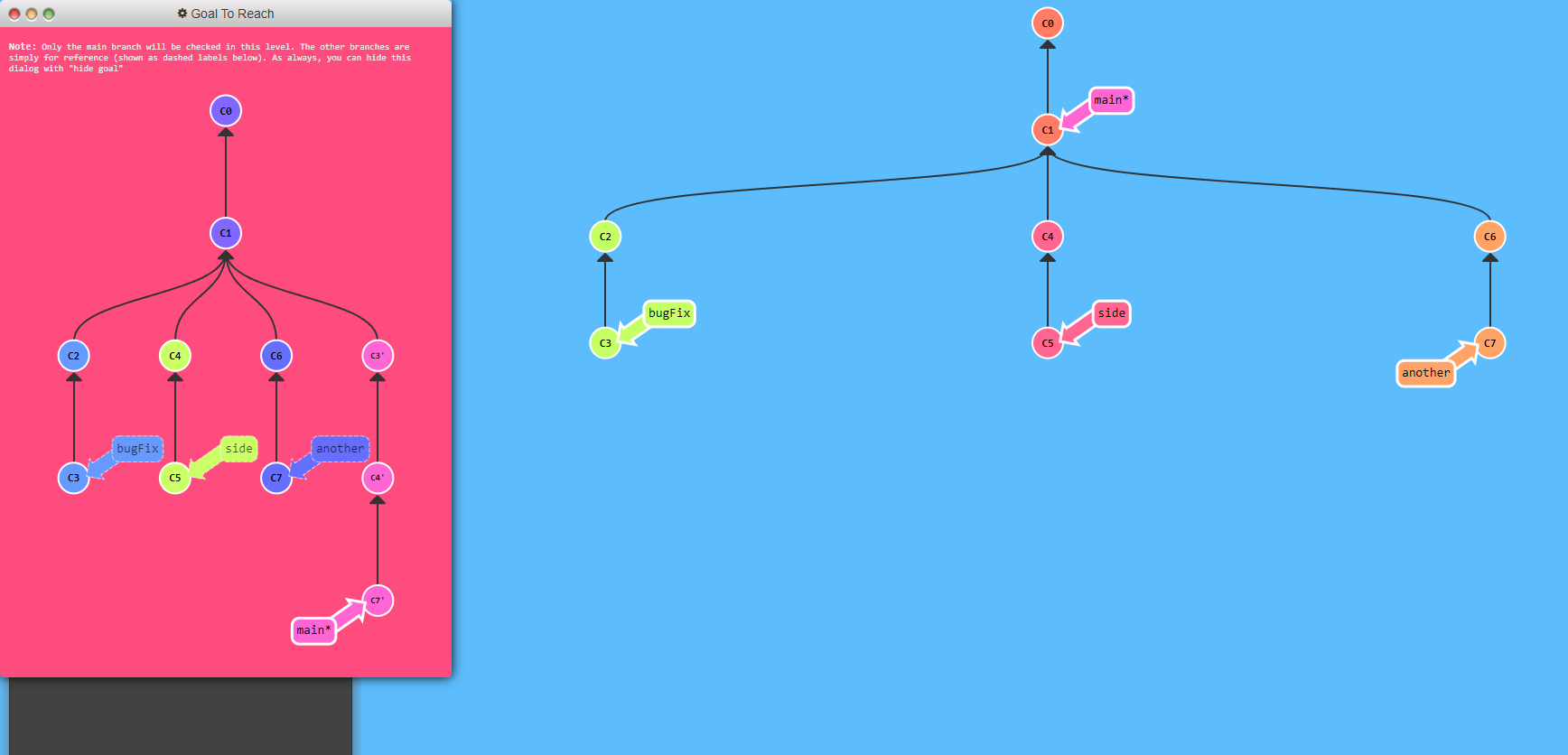


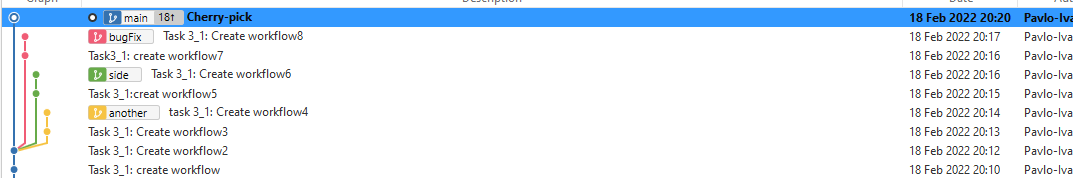
Рис. 2.4.3. Історія комітів після виконання 2 завдання 4 пункту

# Завдання 3.

**Пункт 1.**

Використання команди cherry-pick.

 Рис. 3.1.1. Завдання до 3 завдання 1 пункту



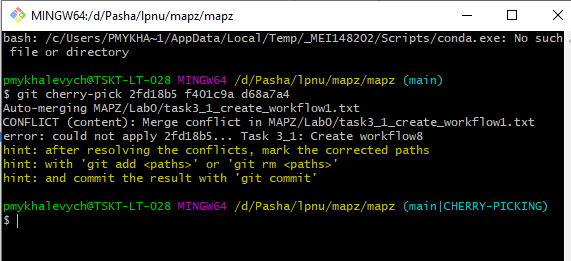
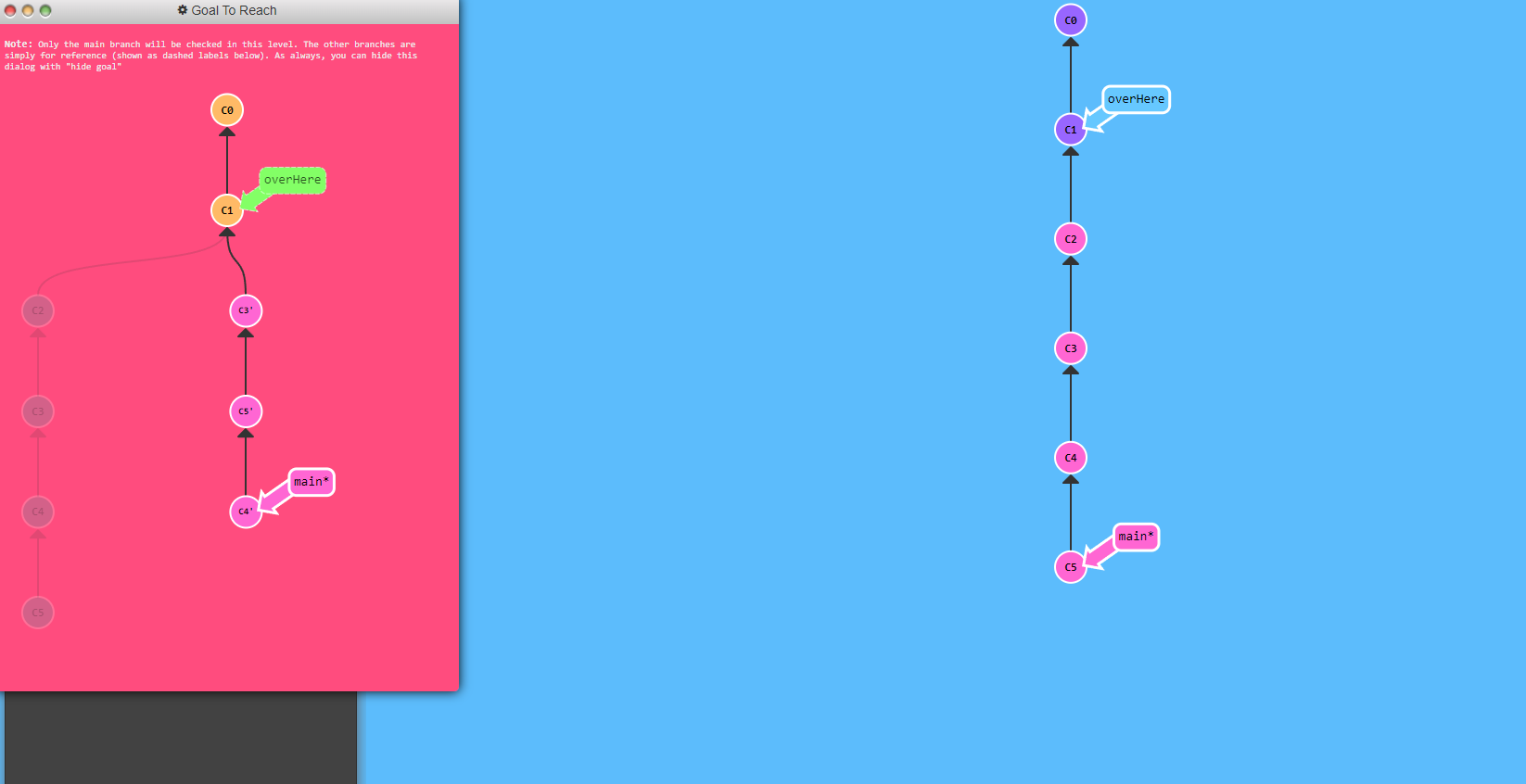
Рис. 3.1.2. Історія комітів до 3 завдання 1 пункту

Рис. 3.1.3. Виконання 3 завдання 1 пункту

**Пункт 2.**

Використання команди rebase -i.

 Рис. 3.2.1. Завдання до 3 завдання 2 пункту

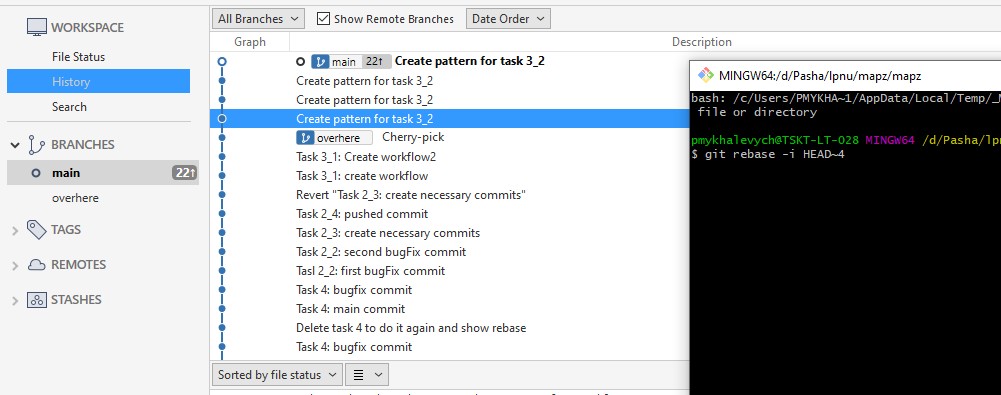


Рис. 3.2.2. Історія комітів до виконання 3 завдання 2 пункту

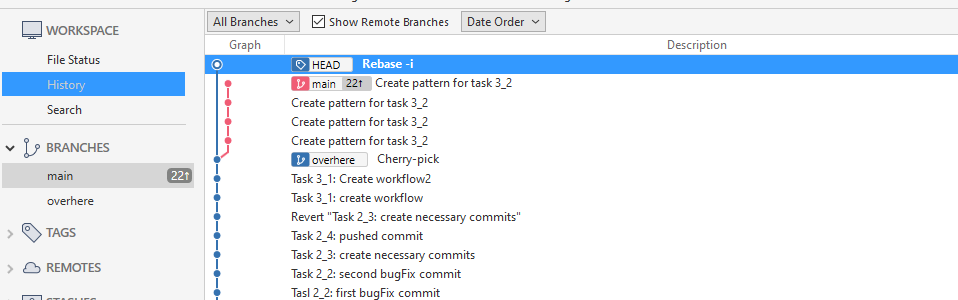
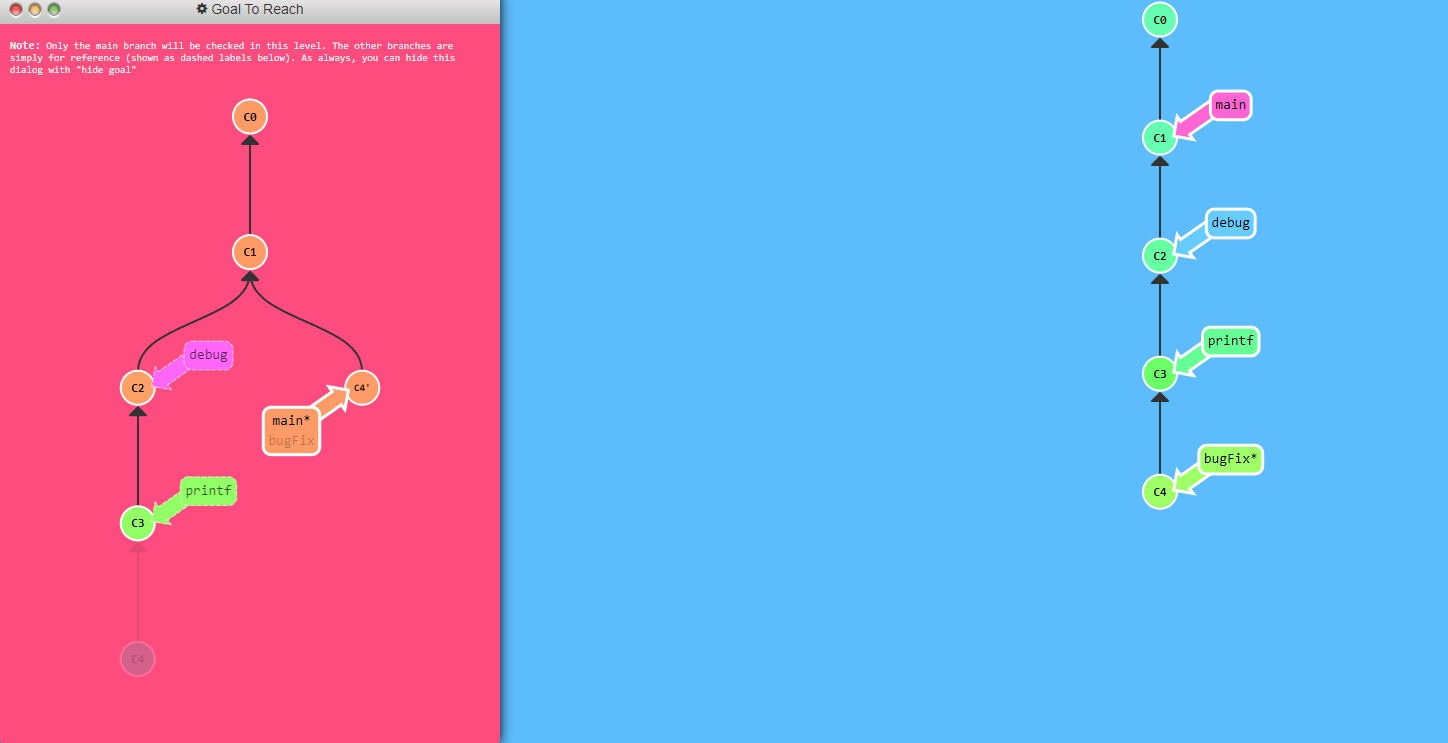


Рис. 3.2.3. Історія комітів після виконання 3 завдання 2 пункту

**Завдання 4.**

**Пункт 1.**

Використання команди cherry-pick.

 Рис. 4.1.1. Завдання до 4 завдання 1 пункту

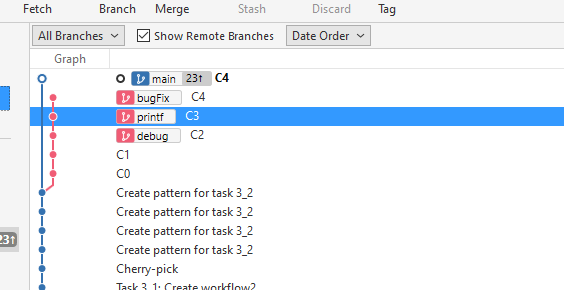
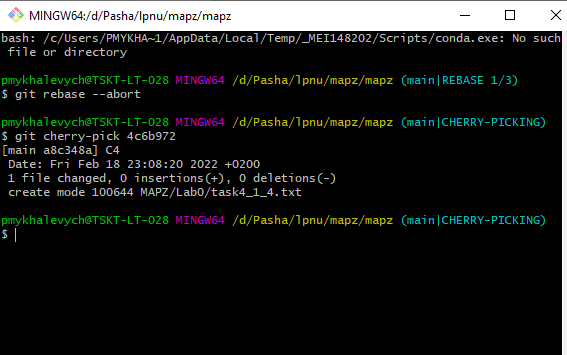
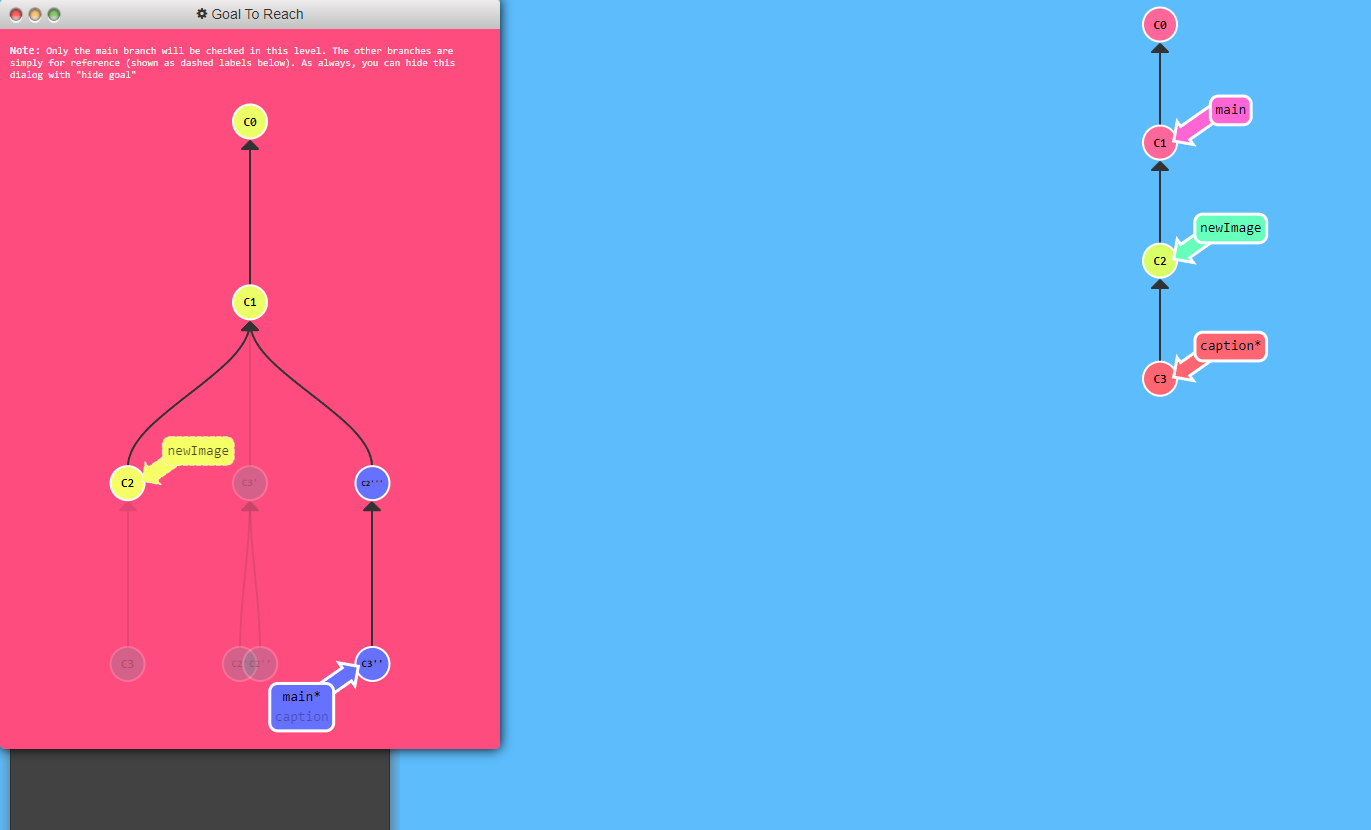


Рис. 4.1.2. Виконання 4 завдання 1 пункту

**Пункт 2.**

Використання команди cherry-pick та rebase -i.

 Рис. 4.2.1. Завдання до 4 завдання 2 пункту

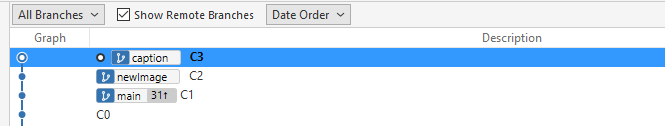
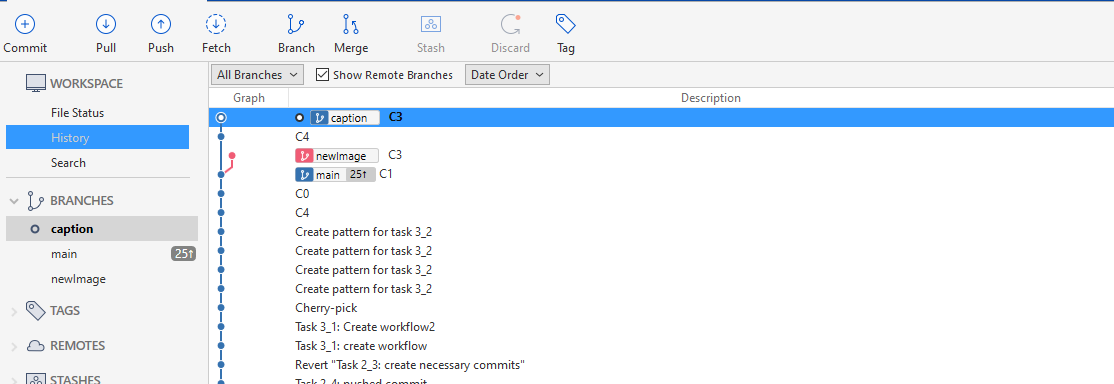


Рис. 4.2.2. Виконання 4 завдання 2 пункту

Рис. 4.2.3. Виконання 4 завдання 2 пункту

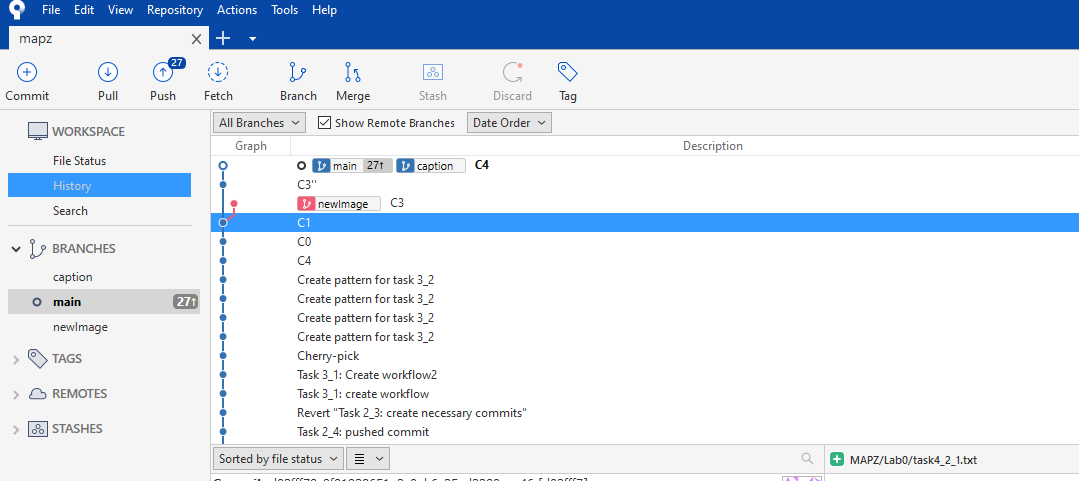
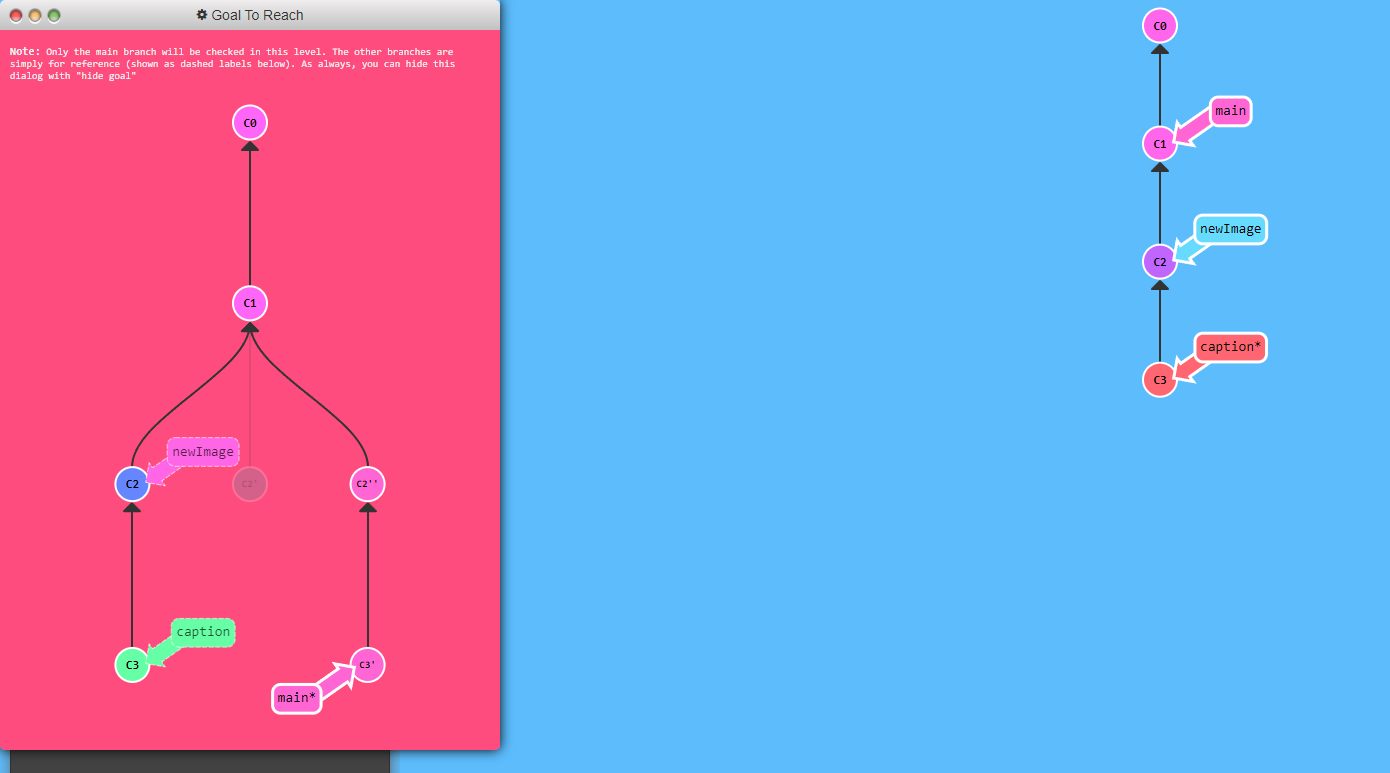


Рис. 4.2.1. Історія комітів після виконання 4 завдання 2 пункту

**Пункт 3.**

Використання команди cherry-pick.

 Рис. 4.3.1. Завдання до 4 завдання 3 пункту

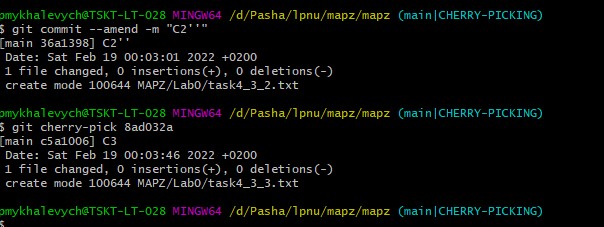


Рис. 4.3.2. Виконання 4 завдання 3 пункту

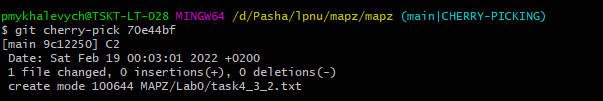


Рис. 4.3.3. Виконання 4 завдання 3 пункту

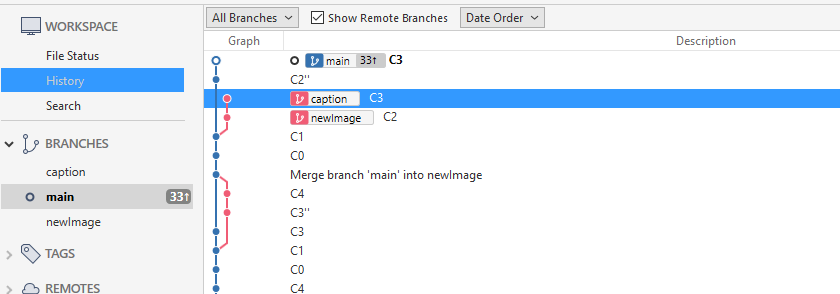
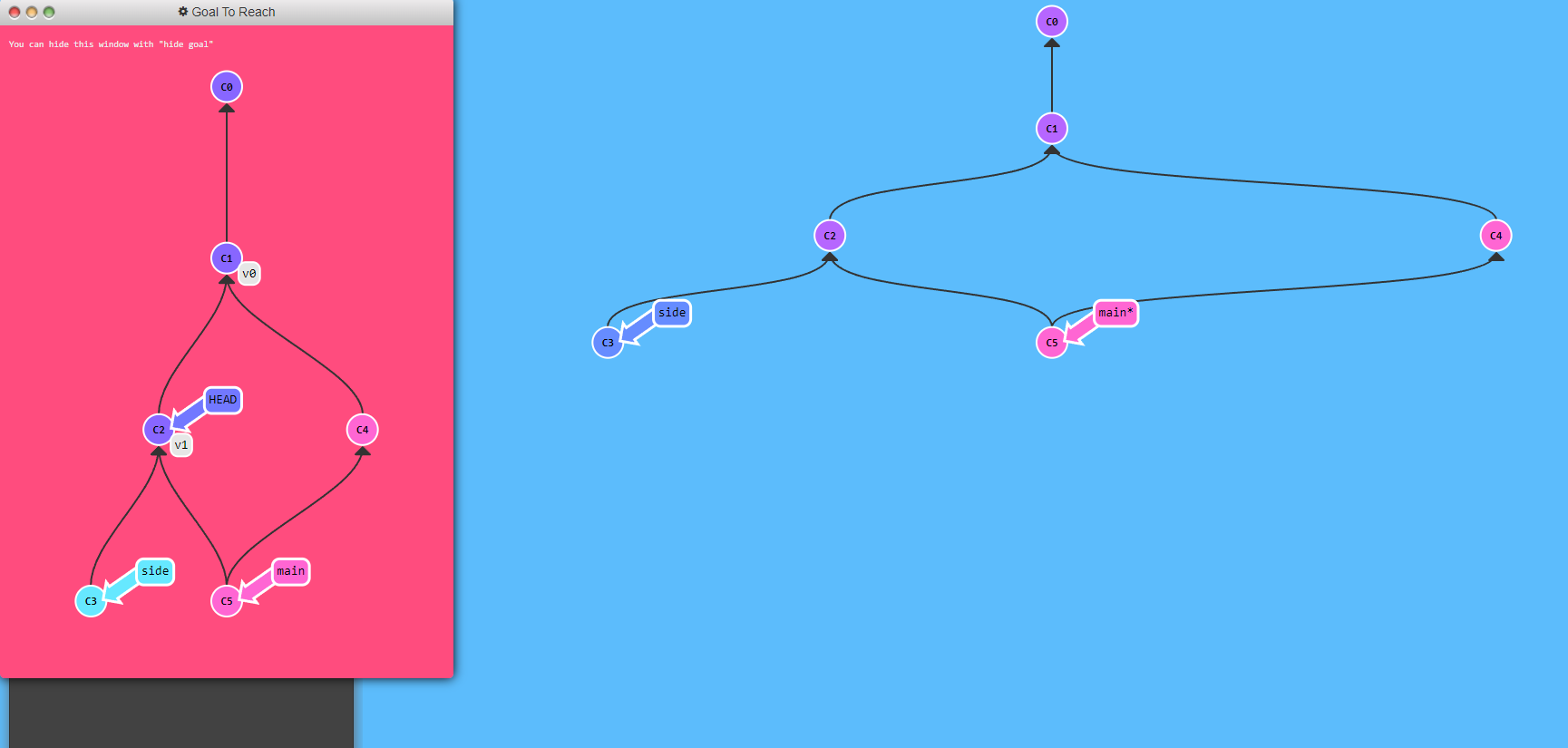


Рис. 4.3.4. Історія комітів після виконання 4 завдання 3 пункту

**Пункт 4.**

Використання тегів у Git.

 Рис. 4.4.1. Завдання до 4 завдання 4 пункту

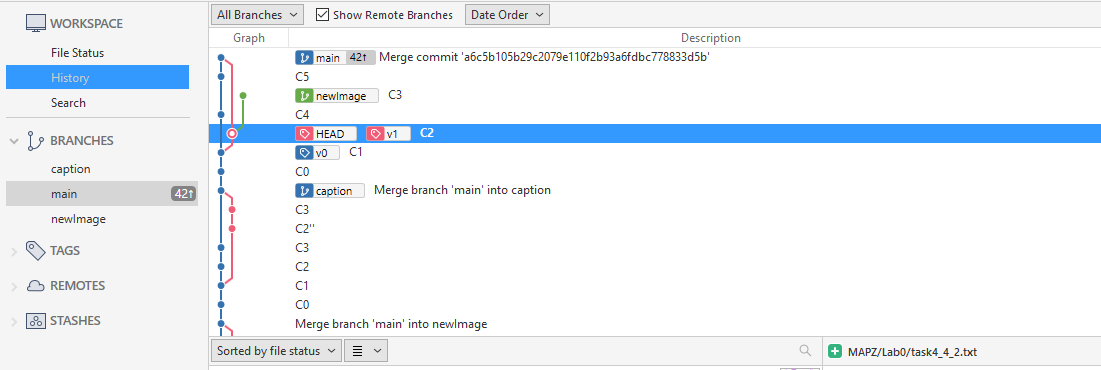


Рис. 4.4.2. Історія комітів після виконання до 4 завдання 4 пункту

**Пункт 5.**

Використання команди *git describe.*

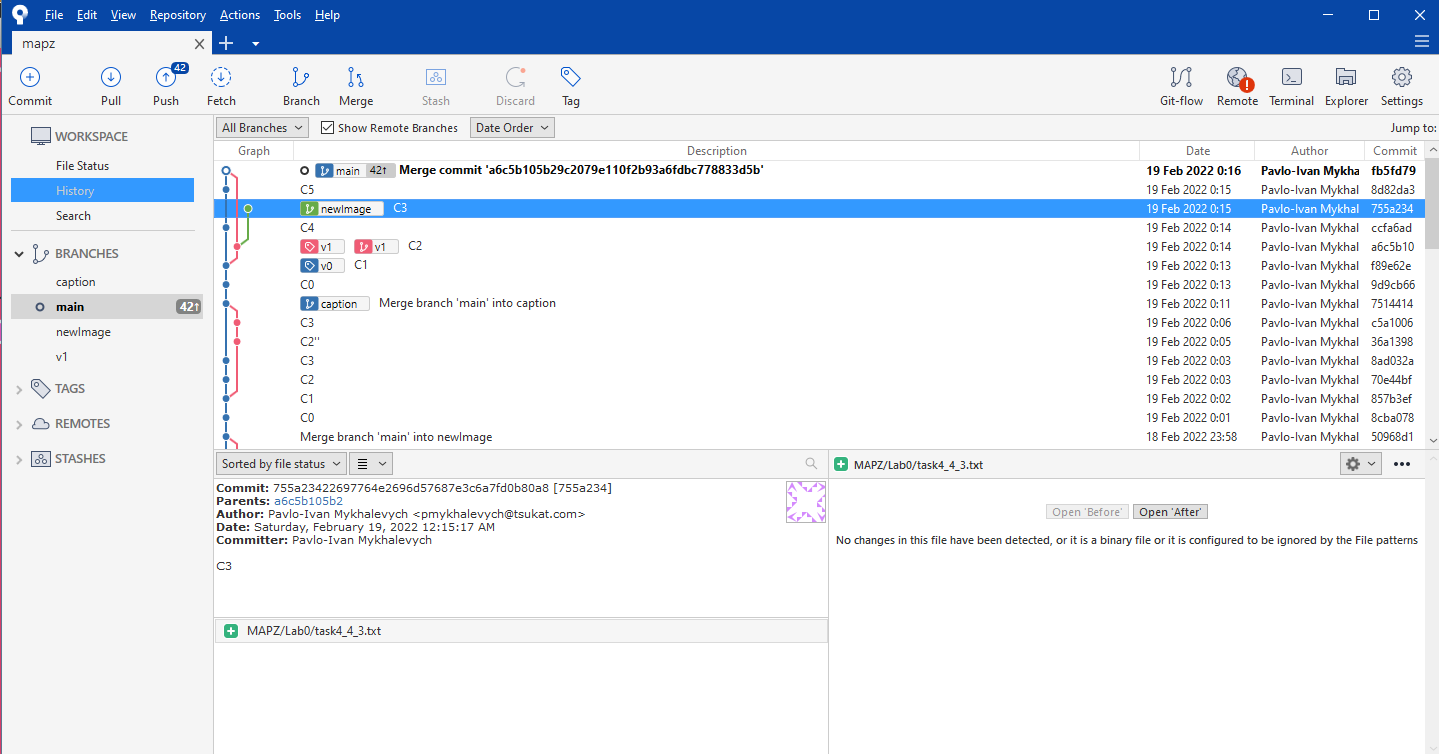


Рис. 4.4.2. Історія комітів до 4 завдання 4 пункту

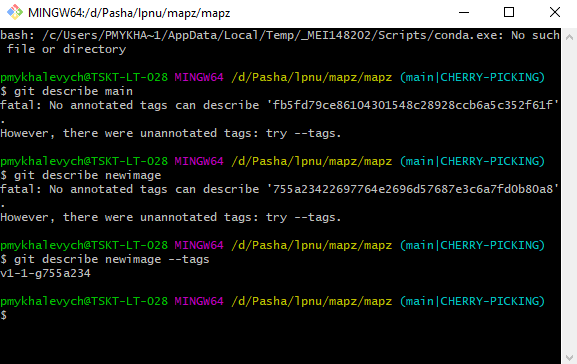


Рис. 4.4.2. Виконання 4 завдання 4 пункту

# ВИСНОВКИ

На даній лабораторній роботі я вивчив, що таке Git та як з ним працювати,навчився створювати репозиторій. Я вивчив основні команди Git, такі як: git commit, git branch, git checkout, git merge, git rebase, git revert та інші. Зрозумів, що при виборі не тих команд, можна втратити всі зміни, тому варто уважно вибирати команди.