



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

Ейсмонт А.В.

Перевірив:

Мягкий М.Ю.

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. Система керування версіями дозволяє додавати зміни в файлах в репозиторій і таким чином після кожної фіксації змін мانی 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.

Такі системи найбільш широко використовуються при розробці програмного забезпечення для зберігання вихідних кодів програми, що розробляється. Однак вони можуть з успіхом застосовуватися і в інших областях, в яких ведеться робота з великою кількістю електронних документів, що безперервно змінюються. Зокрема, системи керування версіями застосовуються у САПР, зазвичай у складі систем керування даними про виріб (PDM). Керування версіями використовується у інструментах конфігураційного керування (Software Configuration Management Tools).

Робота з Git може виконуватися з командного рядка, а також за допомогою візуальних оболонок. Командний рядок використовується програмістами, як можливість виконання всіх доступних команд, а також можливості складання складних макросів. Візуальні оболонки як правило дають більш наглядне представлення репозиторію у вигляді дерева, та більш зручний спосіб роботи з репозиторієм, але, дуже часто доступний не весь набір команд Git, а лише саме ті, що найчастіше використовуються.

Прикладами візуальних оболонок для роботи з Git є Git Extension, SourceTree, GitKraken, GitHub Desktop та інші. 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»).

Таким чином, якщо розглядати основний робочий процес програміста в команді, то він виглядає наступним чином: На початку роботи з проєктом виконується клонування, після цього, в рамках виконання поставленої задачі, створюється бранч і всі зміни в коді, зроблені в рамках цієї задачі фіксуються в репозиторії (періодично виконується синхронізація з основним репозиторієм). Далі, коли задача виконана, то виконується об'єднання гілки з основною гілкою і фінальна синхронізація з центральним репозиторієм.

Хід роботи:

1. Створити порожній репозиторій

```
C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git init
Initialized empty Git repository in C:/Users/Home/Desktop/УНІБЕР/5 СЕМЕСТР/ТРПЗ/.git/

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git commit --allow-empty -m 'init'
[master (root-commit) 25cad14] 'init'
```

2. Створити дві додаткові гілки

```
C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git branch aa

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git checkout -b bb
Switched to a new branch 'bb'

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git branch
  aa
* bb
  master
```

3. Створити конфлікт

```
C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>echo "111" > f.txt

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git add f.txt

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git commit -m "1"
[bb 247ccca] 1
 1 file changed, 1 insertion(+)
 create mode 100644 f.txt

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git checkout aa
Switched to branch 'aa'

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>echo "222" > f.txt

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git add f.txt

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git commit -m "2"
[aa 47e22fd] 2
 1 file changed, 1 insertion(+)
 create mode 100644 f.txt

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТРПЗ>git merge bb
Auto-merging f.txt
CONFLICT (add/add): Merge conflict in f.txt
```

4. Вирішити конфлікт(вирішуємо вручну)

```
C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТПЗ>notepad f.txt

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТПЗ>git add f.txt

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТПЗ>git merge --continue
[aa 109ff77] Merge branch 'bb' into aa

C:\Users\Home\Desktop\УНІБЕР\5 СЕМЕСТР\ТПЗ>git log --all --graph
*   commit 109ff77df85200eda19f0c81377f3b816a8b87b8 (HEAD -> aa)
|  \
|   Merge: 47e22fd 247ccca
|   Author: Eismont Anastasiia <eismont.anastasiia@lkl.kpi.ua>
|   Date:   Sat Sep 27 11:23:14 2025 +0300
|
|       Merge branch 'bb' into aa
|
| *   commit 247ccca64790e0a5284def465989635ef900c3dc (bb)
| |  \
| |   Author: Eismont Anastasiia <eismont.anastasiia@lkl.kpi.ua>
| |   Date:   Sat Sep 27 11:20:16 2025 +0300
| |
| |       1
| |
| *   commit 47e22fdb0fa983dc6eff25930495cba3ad4e54f0
| |  /
| |   Author: Eismont Anastasiia <eismont.anastasiia@lkl.kpi.ua>
| |   Date:   Sat Sep 27 11:21:31 2025 +0300
| |
| |       2
| |
| *   commit 25cad14655258152c7490ff9ceb4c9437b5bdc81 (master)
| |   Author: Eismont Anastasiia <eismont.anastasiia@lkl.kpi.ua>
| |   Date:   Sat Sep 27 11:18:24 2025 +0300
| |
| |       'init'
```

Висновок: Під час виконання лабораторної роботи я навчилася виконувати основні операції в роботі з децентралізованими системами контролю версій на прикладі роботи з сучасною системою Git. А саме створення репозиторію, гілок, створення конфлікту та його вирішення.