

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: «Системи контролю версій. Розподілена система контролю версій “Git”»

Виконав:

студент групи ІА-34
Сухоручкін Гліб

Перевірив:

Асистент кафедри ІСТ
Мягкий М.Ю.

Зміст:

1. Вступ
2. Теоретичні відомості
3. Хід роботи
4. Висновки

Вступ

Метою цієї роботи є 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).

Про систему управління версіями Git

Лінус Торвальдс, т.зв. Батько Лінуksа, розробив і впровадив першу версію Гіт для надання можливості розробникам ядра Лінуks проводити контроль версій не тільки в BitKeeper.

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)

Хід роботи

1. Створення локального репозиторію.

```
(2025-09-21 / 17:42:36)-(gitlab)-(glebtik@glebtiks-pc)
▶ git init
Initialized empty Git repository in C:/Users/glebtik/Documents/gitlab/.git/
```

Рис. 1 – Створення локального репозиторію gitlab

2. Додавання та зафіксування довільного файлу test.txt.

```
(2025-09-21 / 17:46:04)-(gitlab)-(glebtik@glebtiks-pc)
▶ echo "abc" >test.txt

(2025-09-21 / 17:46:45)-(gitlab)-(glebtik@glebtiks-pc)
▶ git add .

(2025-09-21 / 17:46:48)-(gitlab)-(glebtik@glebtiks-pc)
▶ git commit -m "Create test.txt"
[master (root-commit) 5614613] Create test.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.txt
```

Рис. 2 – Додавання та зафіксування довільного файлу test.txt до локального репозиторію

3. Створення гілки і перехід на неї

```
(2025-09-21 / 17:48:58)-(gitlab)-(glebtik@glebtiks-pc)
▶ git branch one

(2025-09-21 / 17:48:59)-(gitlab)-(glebtik@glebtiks-pc)
▶ git checkout one
Switched to branch 'one'

(2025-09-21 / 17:49:03)-(gitlab)-(glebtik@glebtiks-pc)
▶ git status
On branch one
nothing to commit, working tree clean
```

Рис. 3 – Створення гілки one і перехід на неї

4. Створення гілки і перехід на неї за допомогою git checkout

```
(2025-09-21 / 17:50:30)-(gitlab)-(glebtik@glebtiks-pc)
▶ git checkout -b two
Switched to a new branch 'two'

(2025-09-21 / 17:51:36)-(gitlab)-(glebtik@glebtiks-pc)
▶ git branch
  master
  one
* two
```

Рис. 4 – Створення гілки two і перехід на неї за допомогою git checkout

5. Створення довільного файлу та додавання його до першої гілки

```

(2025-09-21 / 17:54:30)-(gitlab)-(glebtik@glebtiks-pc)
▶ echo 1 > f1.txt

(2025-09-21 / 17:54:35)-(gitlab)-(glebtik@glebtiks-pc)
▶ git checkout one
Switched to branch 'one'

(2025-09-21 / 17:54:38)-(gitlab)-(glebtik@glebtiks-pc)
▶ git add f1.txt

(2025-09-21 / 17:54:40)-(gitlab)-(glebtik@glebtiks-pc)
▶ git commit -m "Create f1.txt"
[one 0b9badc] Create f1.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f1.txt

```

Рис. 5 – Створення довільного файлу f1.txt та додавання його до гілки one

6. Створення довільного файлу та додавання його до другої гілки

```

(2025-09-21 / 17:54:42)-(gitlab)-(glebtik@glebtiks-pc)
▶ echo 2 > f2.txt

(2025-09-21 / 17:55:08)-(gitlab)-(glebtik@glebtiks-pc)
▶ git checkout two
Switched to branch 'two'

(2025-09-21 / 17:55:15)-(gitlab)-(glebtik@glebtiks-pc)
▶ git add f2.txt

(2025-09-21 / 17:55:19)-(gitlab)-(glebtik@glebtiks-pc)
▶ git commit -m "Create f2.txt"
[two 0961918] Create f2.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f2.txt

```

Рис. 6 – Створення довільного файлу f2.txt та додавання його до гілки two

7. Злів змін з першою гілкою

```

(2025-09-21 / 17:55:19)-(gitlab)-(glebtik@glebtiks-pc)
▶ git branch
  master
  one
* two

(2025-09-21 / 17:56:07)-(gitlab)-(glebtik@glebtiks-pc)
▶ git merge one
Merge made by the 'ort' strategy.
 f1.txt | Bin 0 -> 8 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 f1.txt

```

Рис. 7 – Злів змін гілки two з гілкою one

8. Виведення історії змін на екран

```
(2025-09-21 / 17:56:22)-(gitlab)-(glebtik@glebtiks-pc)-(4s623ms)
git log --all --graph
*   commit 8f2ea2bff49ece82de8118b0b9869f7532937e84 (HEAD -> two)
    \ Merge: 0b9badc
     | Author: GlebTiK <glebtik1@gmail.com>
     | Date:   Sun Sep 21 17:56:17 2025 +0300
     |
     | Merge branch 'one' into two
     |
    *   commit 0b9badc5b1d4723ac5eefaa6a6ccd3270e1d41a7 (one)
        | Author: GlebTiK <glebtik1@gmail.com>
        | Date:   Sun Sep 21 17:54:42 2025 +0300
        |
        | Create f1.txt
        |
    *   commit 09619181511dd1b1ff7ee905314ce026c7ddb077
        | Author: GlebTiK <glebtik1@gmail.com>
        | Date:   Sun Sep 21 17:55:19 2025 +0300
        |
        | Create f2.txt
        |
    *   commit 561461336c67373002d2c28c840d9253ac511495 (master)
        | Author: GlebTiK <glebtik1@gmail.com>
        | Date:   Sun Sep 21 17:46:52 2025 +0300
        |
        | Create test.txt
```

Рис. 8 – Виведення історії змін на екран

Висновок

У роботі опановано базові операції Git — створення локального репозиторію, додавання та фіксація файлів, робота з гілками (створення/перехід), злиття змін і перегляд історії.