

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»»

Виконала:

студентка групи ІА-31

Кизим Є. К.

Київ 2025

ВСТУП

Ця лабораторна робота присвячена системам контролю версій, а саме роботі з розподіленою системою контролю версій «Git». Метою лабораторної роботи було навчитися виконувати основні операції в роботі з децентралізованими системами контролю версій на прикладі роботи з сучасною системою Git.

Зміст

| | |
|----------------------------------|----------|
| Теоретичні відомості..... | 4 |
| Хід роботи..... | 5 |
| Висновки | 7 |

Теоретичні відомості

У даній лабораторній роботі було використано:

git init <ім'я_каталогу> – створює новий порожній Git-репозиторій у вказаній папці.

cd <каталог> – перехід у потрібний каталог у файловій системі.

echo "текст" > файл.txt – створення нового файлу і запис тексту в нього.

git add <файл/каталог> – додавання змін (нові або змінені файли) до індексу Git, щоб їх можна було зафіксувати у наступному коміті.

git add . – додавання до індексу всі нові та змінені файли в поточній директорії та вкладених теках (крапка означає «весь каталог»).

git commit -m "повідомлення" – створення коміту (фіксація змін) з коротким описом у лапках.

mkdir <каталог> – створення нової директорії.

rm -r <каталог> – видалення теки разом з усім її вмістом (не rmdir, згідно лабораторного практикуму, тому що ця команда видаляє тільки порожню теку).

git branch <назва> – створення нової гілки з вказаною назвою.

git checkout <назва_гілки> – переключення на вказану гілку.

git merge <назва_гілки> – об'єднання в поточну гілку зміни з іншої гілки.

git log – історія комітів.

Хід роботи

1. Створений локальний репозиторій.

```
user@DESKTOP-AVDF5MP MINGW64 ~  
$ git init lab1  
Initialized empty Git repository in C:/Users/user/lab1/.git/  
  
user@DESKTOP-AVDF5MP MINGW64 ~  
$ cd lab1
```

2. Створений і закомічений в репозиторій файл.

```
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)  
$ echo "test" > hello.txt  
  
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)  
$ git add hello.txt  
warning: in the working copy of 'hello.txt', LF will be replaced  
by CRLF in 'hello.txt'.  
  
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)  
$ git commit -m "Hello added"  
[master (root-commit) 6cf3956] Hello added  
1 file changed, 1 insertion(+)  
create mode 100644 hello.txt
```

3. Створена в проєкті директорія.

```
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)  
$ mkdir child_dir  
  
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)  
$ cd child_dir  
  
user@DESKTOP-AVDF5MP MINGW64 ~/lab1/child_dir (master)  
$ echo "inner" > inner.txt  
  
user@DESKTOP-AVDF5MP MINGW64 ~/lab1/child_dir (master)  
$ cd ..  
  
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)  
$ git add child_dir  
warning: in the working copy of 'child_dir/inner.txt', LF will be replaced  
by CRLF in 'child_dir/inner.txt'.  
warning: use 'git add -f' to force adding it  
  
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)  
$ git commit -m "Add child directory with content"  
[master 132e385] Add child directory with content  
1 file changed, 1 insertion(+)  
create mode 100644 child_dir/inner.txt
```

4. В репозиторії існує дві гілки.

```
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)
$ git branch feature_branch

user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)
$ git checkout feature_branch
Switched to branch 'feature_branch'
```

5. Гілка зі змінами злита з основною гілкою

```
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (feature_branch)
$ rm -r child_dir

user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (feature_branch)
$ git add .

user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (feature_branch)
$ git commit -m "Remove chid_dir"
[feature_branch 8ae9da4] Remove chid_dir
1 file changed, 1 deletion(-)
delete mode 100644 child_dir/inner.txt

user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (feature_branch)
$ git checkout master
Switched to branch 'master'

user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)
$ git merge feature_branch
Updating 132e385..8ae9da4
Fast-forward
 child_dir/inner.txt | 1 -
1 file changed, 1 deletion(-)
delete mode 100644 child_dir/inner.txt
```

Результат:

```
user@DESKTOP-AVDF5MP MINGW64 ~/lab1 (master)
$ git log
commit 8ae9da4342f64f5b2af96c0efb62b5281e68667c (HEAD -> master, feature_branch)
Author: omorrey <babanuna888@gmail.com>
Date: Fri Sep 12 17:22:34 2025 +0300

    Remove chid_dir

commit 132e385f02dbb645467a9040af7f9559bda0cc79
Author: omorrey <babanuna888@gmail.com>
Date: Fri Sep 12 17:16:24 2025 +0300

    Add child directory with content

commit 6cf3956d5b63719cd2ab197ea78a4e62dc3ed89a
Author: omorrey <babanuna888@gmail.com>
Date: Fri Sep 12 17:11:38 2025 +0300

    Hello added
```

Висновки: Під час виконання лабораторної роботи я навчилася виконувати основні операції в роботі з децентралізованими системами контролю версій на прикладі роботи з сучасною системою Git.