



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота №1
Технологія розробки програмного забезпечення
«Системи контролю версій Git.»

Виконала студентка
групи ІА-23:
Кашуб'як С. М.

Перевірив:
Мягкий М. Ю.

Київ 2024

Тема: Системи контролю версій Git.

```
[sofia@MacBook-Pro-Sofia ТРПЗ % mkdir test  
[sofia@MacBook-Pro-Sofia ТРПЗ % git init
```

Рис 1. Команди mkdir test та git init

mkdir test

- Задача команди: створити нову директорію з назвою test в поточній робочій директорії.

git init

- Задача команди: ініціалізувати новий порожній Git-репозиторій в поточній папці.

- Результат: створюється підпапка .git, яка зберігає всю інформацію про репозиторій (коміти, гілки, налаштування тощо).

Ініціалізовано порожнє Git сховище в /Users/sofia/Documents/КПІ/ТРПЗ/.git/

Рис 2. Результат команди git init

```
sofia@MacBook-Pro-Sofia ТРПЗ % git status  
На гілці master  
  
Поки що немає комітів  
  
Невідстежувані файли:  
  (скористайтесь "git add <файл>...", щоб додати до майбутнього коміту)  
  .DS_Store  
  
нічого не додано до коміту, але є невідстежувані файли (скористайтесь "git add" для відстежування)
```

Рис 3. Команда git status

git status

- Задача команди: показати поточний стан репозиторію.

- Результат: інформує про те, які файли відстежуються чи ні.

У цьому випадку .DS_Store — невідстежуваний файл.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git commit --allow-empty -m "Initial commit"  
[master (кореневий коміт) d01b526] Initial commit
```

Рис 4. Команда git commit --allow-empty -m "Initial commit"

git commit --allow-empty -m "Initial commit"

- Задача команди: створити перший коміт без додавання будь-яких файлів до репозиторію.
- Навіщо це потрібно: щоб мати базовий (початковий) коміт, з якого можна почати роботу.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git branch
* master
```

Рис 5. Команда git branch

git branch

- Задача команди: показує всі існуючі гілки в репозиторії та позначає, на якій гілці знаходиться HEAD (поточна гілка).
- Результат: я побачила гілку master, на якій працюю.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git branch branch1
sofia@MacBook-Pro-Sofia ТРПЗ % git branch
branch1
* master
```

Рис 6. Команда git branch branch1

git branch branch1

- Задача команди: створити нову гілку з назвою branch1.
- Навіщо це потрібно: для того, щоб розділити роботу на окрему гілку.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git checkout -b branch2
Переключено на нову гілку "branch2"
```

Рис 7. Команда git checkout -b branch2

git checkout -b branch2

- Задача команди: створити нову гілку branch2 і відразу перемикутись на неї.

- Еквівалент: це еквівалент команд `git branch branch2 + git checkout branch2`.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git checkout master  
Переключено на гілку "master"
```

Рис 8. Команда `git checkout master`

`git checkout master`

- Задача команди: повернути на гілку `master`.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git switch -c branch3  
Переключено на нову гілку "branch3"
```

Рис 9. Команда `git switch -c branch3`

`git switch -c branch3`

- Задача команди: створити нову гілку `branch3` і перемикнутись на неї.

- Різниця від `git checkout`: `git switch` є сучаснішою альтернативою для роботи з гілками.

```
sofia@MacBook-Pro-Sofia ТРПЗ % echo 1 > master.txt
```

Рис 10. Команда `echo 1 > master.txt`

`echo 1 > master.txt`

- Задача команди: створити файл `master.txt`, записуючи в нього число 1.

```
sofia@MacBook-Pro-Sofia ТРПЗ % echo 1 > 1.txt  
sofia@MacBook-Pro-Sofia ТРПЗ % echo 2 > 2.txt  
sofia@MacBook-Pro-Sofia ТРПЗ % echo 3 > 3.txt
```

Рис 11. Команди
`echo 1 > 1.txt, echo 2 > 2.txt, echo 3 > 3.txt`

echo 1 > 1.txt, echo 2 > 2.txt, echo 3 > 3.txt

- Задача команди: створити файли 1.txt, 2.txt та 3.txt, кожен з яких містить відповідне число.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git add .
```

Рис 12. Команда git add .

git add .

- Задача команди: додати всі файли в поточній директорії до стейджингу (staging area), щоб їх можна було зафіксувати в наступному коміті.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git commit 3.txt -m "Add 3.txt"
[branch3 4f5b269] Add 3.txt
 1 file changed, 1 insertion(+)
 create mode 100644 3.txt
```

Рис 13. Команда git commit 3.txt -m "Add 3.txt"

git commit 3.txt -m "Add 3.txt"

- Задача команди: створити коміт, додаючи до репозиторію файл 3.txt з повідомленням Add 3.txt.

```
[sofia@MacBook-Pro-Sofia ТРПЗ % git checkout branch2
[A      .DS_Store
A      1.txt
A      2.txt
A      master.txt
Переключено на гілку "branch2"
```

Рис 14. Команда git checkout branch2

git checkout branch2

- Задача команди: перемикутись на гілку branch2.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git commit 2.txt -m "Add 2.txt"
[branch2 4a7874c] Add 2.txt
 1 file changed, 1 insertion(+)
 create mode 100644 2.txt
```

Рис 15. Команда git commit 2.txt -m "Add 2.txt"

git commit 2.txt -m "Add 2.txt"

- Задача команди: створити коміт на гілці branch2, додаючи файл 2.txt до репозиторію.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git checkout branch1
A      .DS_Store
A      1.txt
A      master.txt
Переключено на гілку "branch1"
```

Рис 16. Команда git checkout branch1

git checkout branch1

- Задача команди: перемикнути на гілку branch1.

```
[sofia@MacBook-Pro-Sofia ТРПЗ % git commit 1.txt -m "Add 1.txt"
[branch1 4465cee] Add 1.txt
1 file changed, 1 insertion(+)
create mode 100644 1.txt
```

Рис 17. Команда git commit 1.txt -m "Add 1.txt"

git commit 1.txt -m "Add 1.txt"

- Задача команди: створити коміт на гілці branch1, додаючи файл 1.txt до репозиторію.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git checkout master
A      .DS_Store
A      master.txt
Переключено на гілку "master"
```

Рис 18. Команда git checkout master

git checkout master

- Задача команди: перемикнути назад на гілку master.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git commit master.txt -m "Add master.txt"
[master a73769d] Add master.txt
1 file changed, 1 insertion(+)
create mode 100644 master.txt
```

Рис 19. Команда git commit master.txt -m "Add master.txt"

git commit master.txt -m "Add master.txt"

- Задача команди: додати файл master.txt до репозиторію в гілці master.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git log --all --oneline
a73769d (HEAD -> master) Add master.txt
4465cee (branch1) Add 1.txt
4a7874c (branch2) Add 2.txt
4f5b269 (branch3) Add 3.txt
d01b526 Initial commit
```

Рис 19. Команда git log --all --oneline

git log --all --oneline

- Задача команди: показати стислий список всіх комітів у репозиторії з усіх гілок (включаючи їх ідентифікатори та повідомлення).

```
sofia@MacBook-Pro-Sofia ТРПЗ % echo A > 1.txt
sofia@MacBook-Pro-Sofia ТРПЗ % echo A > 2.txt
sofia@MacBook-Pro-Sofia ТРПЗ % echo A > 3.txt
```

Рис 20. Команди echo A > 1.txt, echo A > 2.txt, echo A > 3.txt

echo A > 1.txt, echo A > 2.txt, echo A > 3.txt

- Задача команди: перезаписати файли 1.txt, 2.txt, 3.txt, замінюючи їх вміст на А.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git commit -m "Add file 1, 2, 3"
[master 039fb0d] Add file 1, 2, 3
4 files changed, 3 insertions(+)
 create mode 100644 .DS_Store
 create mode 100644 1.txt
 create mode 100644 2.txt
 create mode 100644 3.txt
```

Рис 21. Команда git commit -m "Add file 1, 2, 3"

git commit -m "Add file 1, 2, 3"

- Задача команди: створити коміт, додаючи зміни у файли 1.txt, 2.txt, 3.txt.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git switch a73769d --detach
HEAD зараз на a73769d Add master.txt
sofia@MacBook-Pro-Sofia ТРПЗ % echo A > 1.txt
sofia@MacBook-Pro-Sofia ТРПЗ % echo A > 2.txt
sofia@MacBook-Pro-Sofia ТРПЗ % echo A > 3.txt
sofia@MacBook-Pro-Sofia ТРПЗ % git add .
sofia@MacBook-Pro-Sofia ТРПЗ % git commit -m "Add file 1, 2, 3"
[від'єднаний HEAD 00e1dfe] Add file 1, 2, 3
3 files changed, 3 insertions(+)
create mode 100644 1.txt
create mode 100644 2.txt
create mode 100644 3.txt
```

Рис 22. Команда git switch a73769d --detach

git switch a73769d --detach

- Задача команди: перемикнути репозиторій на коміт a73769d без прив'язки до жодної гілки (від'єднаний HEAD).

```
sofia@MacBook-Pro-Sofia ТРПЗ % git log master
[commit 039fb0d6d1442d0eff85c5c1aec9a439b74d1e8d (master)]
Author: kashubiaksofiia <kashsophia@gmail.com>
Date: Thu Oct 10 14:42:24 2024 +0300
```

Add file 1, 2, 3

```
commit a73769d45d13ef3ea701614738e53d3910ddd708
Author: kashubiaksofiia <kashsophia@gmail.com>
Date: Thu Oct 10 14:38:11 2024 +0300
```

Add master.txt

```
commit d01b52619e6c107218868411bb6afcb8dc9de456
Author: kashubiaksofiia <kashsophia@gmail.com>
Date: Thu Oct 10 14:23:50 2024 +0300
```

Initial commit

Рис 23. Команда git log master


```
sofia@MacBook-Pro-Sofia ТРПЗ % git checkout branch1
Попередження: ви залишаєте позаду 1 коміт, не підключений до
жодної з ваших гілок:
```

```
00e1dfe Add file 1, 2, 3
```

Якщо ви хочете зберегти його, створивши нову гілку, то зараз є чудова нагода зробити це за допомогою:
`git branch <назва-нової-гілки> 00e1dfe`

Переключено на гілку "branch1"

Рис 23. Команда `git checkout branch1`

git checkout branch1

- Задача команди: перемикнути на гілку branch1.
- Попередження: залишили від'єднаний коміт позаду (про це Git попередив).

```
Переключено на гілку "branch1"
sofia@MacBook-Pro-Sofia ТРПЗ % git merge master
Автоматичне злиття 1.txt
КОНФЛІКТ (додано/додано): конфлікт злиття в 1.txt
Автоматичне злиття не вдалося; виправте конфлікти і потім зробіть коміт результату.
```

Рис 24. Команда `git merge master`

git merge master

- Задача команди: злити зміни з гілки master в branch1.
- Конфлікт: виник конфлікт у файлі 1.txt.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git merge --continue
U      1.txt
помилка: Створення коміту неможливе, оскільки у вас є не злиті файли.
підказка: Виправте їх у робочому дереві, а потім скористайтесь командою "git add/rm <файл>"
підказка: щоб позначити вирішення і зробити коміт.
збій: Вихід через невирішений конфлікт.
sofia@MacBook-Pro-Sofia ТРПЗ % git add .
sofia@MacBook-Pro-Sofia ТРПЗ % git merge --continue
[branch1 f5e9c07] Merge branch 'master' into branch1
```

Рис 25. Команда `git merge --continue`

git merge --continue

- Задача команди: після виправлення конфлікту завершує процес злиття.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git rebase master
Автоматичне злиття 2.txt
КОНФЛІКТ (додано/додано): конфлікт злиття в 2.txt
помилка: не вдалося застосувати 4a7874c... Add 2.txt
підказка: Resolve all conflicts manually, mark them as resolved with
підказка: "git add/rm <conflicted_files>", then run "git rebase --continue".
підказка: You can instead skip this commit: run "git rebase --skip".
підказка: To abort and get back to the state before "git rebase", run "git rebase --abort".
підказка: Disable this message with "git config advice.mergeConflict false"
Не вдалося застосувати 4a7874c... Add 2.txt
sofia@MacBook-Pro-Sofia ТРПЗ % git add .
sofia@MacBook-Pro-Sofia ТРПЗ % git rebase --continue
[від'єднаний HEAD 2e814c4] Add 2.txt
1 file changed, 2 insertions(+), 1 deletion(-)
Успішно перебазовано та оновлено refs/heads/branch2.
```

Рис 26. Команда git rebase master

git rebase master

- Задача команди: перебазувати зміни з branch2 поверх гілки master.
- Конфлікт: виникає конфлікт у файлі 2.txt.

```
sofia@MacBook-Pro-Sofia ТРПЗ % git log
commit 4f5b269a36cf22d7ff3bd738c0cb6be099650b85 (HEAD -> branch3)
Author: kashubiaksofiia <kashsophia@gmail.com>
Date: Thu Oct 10 14:34:00 2024 +0300

Add 3.txt

commit d01b52619e6c107218868411bb6afcb8dc9de456
Author: kashubiaksofiia <kashsophia@gmail.com>
Date: Thu Oct 10 14:23:50 2024 +0300

Initial commit
sofia@MacBook-Pro-Sofia ТРПЗ % git cherry-pick 039fb0d a73769d
Автоматичне злиття 3.txt
КОНФЛІКТ (додано/додано): конфлікт злиття в 3.txt
помилка: не вдалося застосувати 039fb0d... Add file 1, 2, 3
підказка: Після вирішення конфліктів позначте їх за допомогою
підказка: "git add/rm <визначник шляху>", а потім виконайте
підказка: "git cherry-pick --continue".
підказка: Замість цього ви можете пропустити цей коміт за допомогою "git cherry-pick --skip".
підказка: Щоб перервати процес і повернутися до стану перед "git cherry-pick",
підказка: виконайте "git cherry-pick --abort".
підказка: Disable this message with "git config advice.mergeConflict false"
sofia@MacBook-Pro-Sofia ТРПЗ % git add .
sofia@MacBook-Pro-Sofia ТРПЗ % git cherry-pick --continue
[branch3 437bf23] Add file 1, 2, 3
Date: Thu Oct 10 14:42:24 2024 +0300
4 files changed, 4 insertions(+), 1 deletion(-)
create mode 100644 .DS_Store
create mode 100644 1.txt
create mode 100644 2.txt
[branch3 8211c1c] Add master.txt
Date: Thu Oct 10 14:38:11 2024 +0300
1 file changed, 1 insertion(+)
create mode 100644 master.txt
```

Рис 27. Команда git cherry-pick

git cherry-pick 039fb0d a73769d

- Задача команди: перенести зміни з комітів 039fb0d та a73769d у поточну гілку branch3.
- Конфлікт: конфлікт у 3.txt, який потрібно вирішити.

```

sofia@MacBook-Pro-Sofia TПП3 % git log --all --oneline
8211c1c (HEAD -> branch3) Add master.txt
437bf23 Add file 1, 2, 3
2e814c4 (branch2) Add 2.txt
f5e9c07 (branch1) Merge branch 'master' into branch1
039fb0d (master) Add file 1, 2, 3
a73769d Add master.txt
4465cee Add 1.txt
4f5b269 Add 3.txt
d01b526 Initial commit
sofia@MacBook-Pro-Sofia TПП3 % git log --all --oneline --graph
* 8211c1c (HEAD -> branch3) Add master.txt
* 437bf23 Add file 1, 2, 3
* 4f5b269 Add 3.txt
| * 2e814c4 (branch2) Add 2.txt
| | * f5e9c07 (branch1) Merge branch 'master' into branch1
| | | \
| | | /
| | /
| /
| * 039fb0d (master) Add file 1, 2, 3
| * a73769d Add master.txt
| /
| * 4465cee Add 1.txt
| /
* d01b526 Initial commit

```

Рис 28. Всі коміти репозиторію

git log --all --oneline --graph

- Задача команди: відобразити графічну структуру комітів у репозиторії.

Висновок: у цій лабораторній роботі продемонстровано процес роботи з Git для створення та управління гілками, внесення змін у файли, обробки конфліктів, а також застосування основних операцій, таких як злиття (merge), перебазування (rebase), та вибіркоче застосування комітів (cherry-pick).