



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 розробки програмного
забезпечення**
СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ. GIT

Виконала:

Студентка групи ІА-22 Лахоцька
С. О.

Перевірив: Мягкий
М. Ю

Київ 2024

Зміст

Теоретичні відомості	3
Висновки.....	5
Вихідний код	6

Тема: Системи контролю версій. Git.

Мета: ознайомитись з теоретичними відомостями, опрацювати у терміналі команди Git.

Теоретичні відомості

1. Що таке система контролю версій (СКВ)

Система контролю версій (СКВ) — це інструмент, який дозволяє розробникам відстежувати зміни в коді, документації або інших файлах проєкту. Вона забезпечує зберігання історії змін, можливість працювати декільком розробникам над одним проєктом, відновлення попередніх версій файлів та вирішення конфліктів при внесенні змін.

2. Відмінності між розподіленою та централізованою СКВ

Централізована СКВ зберігає весь код на одному сервері, а для доступу до історії змін потрібне постійне підключення до сервера. Це менш стійкий підхід, оскільки сервер є єдиною точкою відмови.

У розподіленій СКВ кожен розробник має повну копію репозиторію, що дозволяє працювати автономно. Мережевий сервер потрібен лише для синхронізації.

3. Різниця між stage та commit в Git

Stage (стейджинг) — це підготовка файлів до збереження у вигляді змін в області staging area. Використовується команда `git add`.

Commit — це збереження цих змін в історії репозиторію разом із коментарем за допомогою команди `git commit`.

4. Як створити гілку в Git

Щоб створити нову гілку, використовується команда `git branch <ім'я_гілки>`. Для переходу на неї використовується `git checkout <ім'я_гілки>` або одночасно створення й перехід: `git switch -c <ім'я_гілки>`.

5. Як створити або скопіювати репозиторій Git з віддаленого серверу

Щоб скопіювати репозиторій, використовується команда `git clone <URL>`. Для створення нового репозиторію локально слід ініціалізувати його за допомогою команди `git init`, а потім додати віддалений сервер через `git remote add origin <URL>`.

6. Що таке конфлікт злиття, як створити конфлікт, як вирішити конфлікт?

Конфлікт злиття виникає, коли Git не може автоматично об'єднати зміни через суперечливі правки в одному файлі. Щоб створити конфлікт, достатньо внести зміни в той самий рядок файлу в двох різних гілках і спробувати виконати злиття через `git merge`. Для вирішення конфлікту потрібно вручну відкрити файл, вибрати потрібні зміни, зберегти файл, додати його у staging area через `git add`, і завершити коміт `git commit`.

7. Ситуації використання команд: merge, rebase, cherry-pick

Команда `merge` використовується для об'єднання гілок із збереженням їхньої історії, наприклад, під час завершення роботи над фічею. `Rebase` застосовується для перенесення змін із однієї гілки на іншу з метою збереження лінійної історії. `Cherry-pick` дозволяє вибірково перенести конкретний коміт із однієї гілки в іншу.

8. Як переглянути історію змін Git репозиторію в консолі?

Для перегляду історії використовується команда `git log`. Щоб побачити спрощену і візуально структуровану історію, можна використовувати `git log --oneline --graph --all`.

9. Як створити гілку в Git не використовуючи команду git branch

Можна одночасно створити і перейти на нову гілку за допомогою команди `git switch -c <ім'я_гілки>`.

10. Як підготувати всі зміни в поточній папці до коміту?

Для підготовки всіх змін у поточній папці виконується команда `git add`.

11. Як підготувати всі зміни в дочірній папці до коміту?

Щоб підготувати зміни тільки в певній дочірній папці, використовується команда `git add <шлях_до_папки>`.

12. Як переглянути перелік наявних гілок в репозиторії?

Для перегляду всіх гілок у репозиторії використовується команда `git branch`. Щоб побачити віддалені гілки, додається прапорець `-a`.

13. Як видалити гілку?

Для видалення локальної гілки виконується команда `git branch -d <ім'я_гілки>`. Якщо гілка не злиття, то використовується прапорець `-D`.

Висновки

У цій лабораторній роботі було опрацьовано команди для роботи з `git`.

Вихідний код

Вихідний код, де наведено приклад використання команд git у терміналі.

```
(base) Sofi@MacBookAir ~ % mkdir lab1TRPZ
(base) Sofi@MacBookAir ~ % cd lab1TRPZ
(base) Sofi@MacBookAir lab1TRPZ % git init
Initialized empty Git repository in /Users/Sofi/lab1TRPZ/.git/
(base) Sofi@MacBookAir lab1TRPZ % git branch
(base) Sofi@MacBookAir lab1TRPZ % git add .
(base) Sofi@MacBookAir lab1TRPZ % git commit -m "початковий коміт"
On branch main

Initial commit

nothing to commit (create/copy files and use "git add" to track)
(base) Sofi@MacBookAir lab1TRPZ % git branch main
fatal: not a valid object name: 'main'
(base) Sofi@MacBookAir lab1TRPZ % git branch "main"
fatal: not a valid object name: 'main'
(base) Sofi@MacBookAir lab1TRPZ % git checkout -b main
Switched to a new branch 'main'
(base) Sofi@MacBookAir lab1TRPZ % git branch
(base) Sofi@MacBookAir lab1TRPZ % git commit --hel
error: unknown option `hel'
usage: git commit [-a | --interactive | --patch] [-s] [-v] [-u<mode>] [--amend]
                [--dry-run] [(-c | -C | --squash) <commit> | --fixup
[ (amend|reword):]<commit>)]
                [-F <file> | -m <msg>] [--reset-author] [--allow-empty]
                [--allow-empty-message] [--no-verify] [-e] [--author=<author>]
                [--date=<date>] [--cleanup=<mode>] [--[no-]status]
                [-i | -o] [--pathspec-from-file=<file> --pathspec-file-nul]]
                [(--trailer <token>[(=|:)<value>])...] [-S[<keyid>]]
                [--] [<pathspec>...]

    -q, --quiet                suppress summary after successful commit
    -v, --verbose              show diff in commit message template
```

Commit message options

<code>-F, --file <file></code>	read message from file
<code>--author <author></code>	override author for commit
<code>--date <date></code>	override date for commit
<code>-m, --message <message></code>	commit message
<code>-c, --reedit-message <commit></code>	reuse and edit message from specified commit
<code>-C, --reuse-message <commit></code>	reuse message from specified commit
<code>--fixup [(amend reword):]<commit></code>	use autosquash formatted message to fixup or amend/reword specified commit
<code>--squash <commit></code>	use autosquash formatted message to squash specified commit
<code>--reset-author</code>	the commit is authored by me now (used with <code>-C/-c/--amend</code>)
<code>--trailer <trailer></code>	add custom trailer(s)
<code>-s, --signoff</code>	add a Signed-off-by trailer
<code>-t, --template <file></code>	use specified template file
<code>-e, --edit</code>	force edit of commit
<code>--cleanup <mode></code>	how to strip spaces and #comments from message
<code>--status</code>	include status in commit message template
<code>-S, --gpg-sign[=<key-id>]</code>	GPG sign commit

Commit contents options

<code>-a, --all</code>	commit all changed files
<code>-i, --include</code>	add specified files to index for commit
<code>--interactive</code>	interactively add files
<code>-p, --patch</code>	interactively add changes
<code>-o, --only</code>	commit only specified files
<code>-n, --no-verify</code>	bypass pre-commit and commit-msg hooks
<code>--dry-run</code>	show what would be committed
<code>--short</code>	show status concisely
<code>--branch</code>	show branch information

```

--ahead-behind      compute full ahead/behind values
--porcelain         machine-readable output
--long             show status in long format (default)
-z, --null         terminate entries with NUL
--amend            amend previous commit
--no-post-rewrite   bypass post-rewrite hook
-u, --untracked-files[=<mode>]
                    show untracked files, optional modes: all, normal, no.
(Default: all)

--pathspec-from-file <file>
                    read pathspec from file

--pathspec-file-nul with --pathspec-from-file, pathspec elements are
separated with NUL character

```

```
(base) Sofi@MacBookAir lab1TRPZ % git commit --help
```

```
zsh: suspended  git commit --help
```

```
(base) Sofi@MacBookAir lab1TRPZ % git commit --allow-empty
```

```
hint: Waiting for your editor to close the file...
```

```
[No write since last change]
```

```
zsh:1: command not found: q
```

```
shell returned 127
```

```
Press ENTER or type command to continue
```

```
[No write since last change]
```

```
zsh:1: command not found: q
```

```
shell returned 127
```

```
Press ENTER or type command to continue
```

```
[No write since last change]
```

```
zsh:1: command not found: q
```

```
shell returned 127
```

```
Press ENTER or type command to continue
```



```
zsh: suspended  git commit --allow-empty
(base) Sofi@MacBookAir lab1TRPZ % git status
On branch main
```

No commits yet

```
nothing to commit (create/copy files and use "git add" to track)
(base) Sofi@MacBookAir lab1TRPZ % git branch
(base) Sofi@MacBookAir lab1TRPZ % git commit -m --allow-empty
On branch main
```

Initial commit

```
nothing to commit (create/copy files and use "git add" to track)
(base) Sofi@MacBookAir lab1TRPZ % git status
On branch main
```

No commits yet

```
nothing to commit (create/copy files and use "git add" to track)
(base) Sofi@MacBookAir lab1TRPZ % git commit -m "First commit" --allow-empty
[main (root-commit) 252b289] First commit
Committer: Sofi <Sofi@MacBookAir.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
(base) Sofi@MacBookAir lab1TRPZ % git status
```

```
On branch main
nothing to commit, working tree clean
(base) Sofi@MacBookAir lab1TRPZ % git log
commit 252b2896cc2ee35e8efa6ab81dfcbe6b2da16d99 (HEAD -> main)
Author: Sofi <Sofi@MacBookAir.local>
Date: Thu Dec 19 21:17:23 2024 +0200
```

First commit

```
(base) Sofi@MacBookAir lab1TRPZ % git checkout -b feature
Switched to a new branch 'feature'
(base) Sofi@MacBookAir lab1TRPZ % touch file1.txt
(base) Sofi@MacBookAir lab1TRPZ % git add .
(base) Sofi@MacBookAir lab1TRPZ % git commit -m "Add first file"
[feature 4517d55] Add first file
Committer: Sofi <Sofi@MacBookAir.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1.txt
(base) Sofi@MacBookAir lab1TRPZ % git chekout main
git: 'chekout' is not a git command. See 'git --help'.
```

The most similar command is

```
checkout
```

```
(base) Sofi@MacBookAir lab1TRPZ % git checkout main
Switched to branch 'main'
```

```
(base) Sofi@MacBookAir lab1TRPZ % git merge feature
Updating 252b289..4517d55
Fast-forward
 file1.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1.txt
(base) Sofi@MacBookAir lab1TRPZ % git log
commit 4517d5589157d8ffc46b0248d657714db09d0fe3 (HEAD -> main, feature)
Author: Sofi <Sofi@MacBookAir.local>
Date: Thu Dec 19 21:19:01 2024 +0200
```

Add first file

```
commit 252b2896cc2ee35e8efa6ab81dfcbe6b2da16d99
Author: Sofi <Sofi@MacBookAir.local>
Date: Thu Dec 19 21:17:23 2024 +0200
```

First commit

```
(base) Sofi@MacBookAir lab1TRPZ % git branch exp1
(base) Sofi@MacBookAir lab1TRPZ % git branch exp2
(base) Sofi@MacBookAir lab1TRPZ % git checkout exp1
Switched to branch 'exp1'
(base) Sofi@MacBookAir lab1TRPZ % nano file1.txt
(base) Sofi@MacBookAir lab1TRPZ % git add file1.txt
(base) Sofi@MacBookAir lab1TRPZ % git commit -m "change on exp1"
[exp1 a132b02] change on exp1
Committer: Sofi <Sofi@MacBookAir.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 2 insertions(+)
```

```
(base) Sofi@MacBookAir lab1TRPZ % git checkout exp2
```

```
Switched to branch 'exp2'
```

```
(base) Sofi@MacBookAir lab1TRPZ % git log
```

```
commit 4517d5589157d8fffc46b0248d657714db09d0fe3 (HEAD -> exp2, main, feature)
```

```
Author: Sofi <Sofi@MacBookAir.local>
```

```
Date: Thu Dec 19 21:19:01 2024 +0200
```

```
Add first file
```

```
commit 252b2896cc2ee35e8efa6ab81dfcbe6b2da16d99
```

```
Author: Sofi <Sofi@MacBookAir.local>
```

```
Date: Thu Dec 19 21:17:23 2024 +0200
```

```
First commit
```

```
(base) Sofi@MacBookAir lab1TRPZ % nano file1.txt
```

```
(base) Sofi@MacBookAir lab1TRPZ % git add .
```

```
(base) Sofi@MacBookAir lab1TRPZ % git commit -m "change on exp2"
```

```
[exp2 172d246] change on exp2
```

```
Committer: Sofi <Sofi@MacBookAir.local>
```

Your name and email address were configured automatically based on your username and hostname. Please check that they are accurate. You can suppress this message by setting them explicitly. Run the following command and follow the instructions in your editor to edit your configuration file:

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 1 insertion(+)
```

```

(base) Sofi@MacBookAir lab1TRPZ % git checkout main
Switched to branch 'main'

(base) Sofi@MacBookAir lab1TRPZ % git merge expl
Updating 4517d55..a132b02
Fast-forward
   file1.txt | 2 ++
   1 file changed, 2 insertions(+)

(base) Sofi@MacBookAir lab1TRPZ % git merge exp2
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.

(base) Sofi@MacBookAir lab1TRPZ % nano file1.txt

(base) Sofi@MacBookAir lab1TRPZ % git merge --help

(base) Sofi@MacBookAir lab1TRPZ % git add .

(base) Sofi@MacBookAir lab1TRPZ % git merge --continue
hint: Waiting for your editor to close the file... error: There was a problem
with the editor 'vi'.

Please supply the message using either -m or -F option.

(base) Sofi@MacBookAir lab1TRPZ % git merge --continue
hint: Waiting for your editor to close the file... error: There was a problem
with the editor 'vi'.

Please supply the message using either -m or -F option.

(base) Sofi@MacBookAir lab1TRPZ % git merge --continue
hint: Waiting for your editor to close the file... error: There was a problem
with the editor 'vi'.

Please supply the message using either -m or -F option.

(base) Sofi@MacBookAir lab1TRPZ % git log
commit a132b026343ab1b1665ce5f4872106a5c847723b (HEAD -> main, expl)
Author: Sofi <Sofi@MacBookAir.local>
Date: Thu Dec 19 21:24:11 2024 +0200

    change on expl

commit 4517d5589157d8ffc46b0248d657714db09d0fe3 (feature)
Author: Sofi <Sofi@MacBookAir.local>
Date: Thu Dec 19 21:19:01 2024 +0200

    Add first file

```

```
commit 252b2896cc2ee35e8efa6ab81dfcbe6b2da16d99
```

```
Author: Sofi <Sofi@MacBookAir.local>
```

```
Date: Thu Dec 19 21:17:23 2024 +0200
```

First commit

```
(base) Sofi@MacBookAir lab1TRPZ % git checkout -b clean-up
```

```
Switched to a new branch 'clean-up'
```

```
(base) Sofi@MacBookAir lab1TRPZ % nano file1.txt
```

```
(base) Sofi@MacBookAir lab1TRPZ % git add .
```

```
(base) Sofi@MacBookAir lab1TRPZ % git commit -m "1"
```

```
[clean-up 7548775] 1
```

```
Committer: Sofi <Sofi@MacBookAir.local>
```

Your name and email address were configured automatically based on your username and hostname. Please check that they are accurate. You can suppress this message by setting them explicitly. Run the following command and follow the instructions in your editor to edit your configuration file:

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 2 insertions(+)
```

```
(base) Sofi@MacBookAir lab1TRPZ % echo "2" >> file1.txt
```

```
(base) Sofi@MacBookAir lab1TRPZ % nano file1.txt
```

```
(base) Sofi@MacBookAir lab1TRPZ % git add .
```

```
(base) Sofi@MacBookAir lab1TRPZ % git commit -m "2"
```

```
[clean-up 8a4e960] 2
```

```
Committer: Sofi <Sofi@MacBookAir.local>
```

Your name and email address were configured automatically based on your username and hostname. Please check that they are accurate. You can suppress this message by setting them explicitly. Run the following command and follow the instructions in your editor to edit

your configuration file:

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 1 insertion(+)
```

```
(base) Sofi@MacBookAir lab1TRPZ % echo "3" >> file1.txt
```

```
(base) Sofi@MacBookAir lab1TRPZ % git commit -am "3"
```

```
[clean-up b82c16f] 3
```

```
Committer: Sofi <Sofi@MacBookAir.local>
```

Your name and email address were configured automatically based on your username and hostname. Please check that they are accurate. You can suppress this message by setting them explicitly. Run the following command and follow the instructions in your editor to edit your configuration file:

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 1 insertion(+)
```

```
(base) Sofi@MacBookAir lab1TRPZ % git checkout main
```

```
Switched to branch 'main'
```

```
(base) Sofi@MacBookAir lab1TRPZ % git rebase clean-up
```

```
Successfully rebased and updated refs/heads/main.
```

```
(base) Sofi@MacBookAir lab1TRPZ % git log
```

```
commit b82c16f5ef196ed0ca25fe07f867b921f6ad21ee (HEAD -> main, clean-up)
```

```
Author: Sofi <Sofi@MacBookAir.local>
```

```
Date: Thu Dec 19 21:44:13 2024 +0200
```

commit 8a4e960be0cb217f049c6fedd238c11a19bc756a

Author: Sofi <Sofi@MacBookAir.local>

Date: Thu Dec 19 21:43:34 2024 +0200

2

commit 75487756b5d351e8850bbcac4b84ec2738472b52

Author: Sofi <Sofi@MacBookAir.local>

Date: Thu Dec 19 21:42:47 2024 +0200

1

commit a132b026343ab1b1665ce5f4872106a5c847723b (expl)

Author: Sofi <Sofi@MacBookAir.local>

Date: Thu Dec 19 21:24:11 2024 +0200

change on expl

commit 4517d5589157d8ffc46b0248d657714db09d0fe3 (feature)

Author: Sofi <Sofi@MacBookAir.local>

Date: Thu Dec 19 21:19:01 2024 +0200

Add first file

commit 252b2896cc2ee35e8efa6ab81dfcbe6b2da16d99

Author: Sofi <Sofi@MacBookAir.local>

Date: Thu Dec 19 21:17:23 2024 +0200

First commit

(base) Sofi@MacBookAir lab1TRPZ % git log --oneline --graph

* b82c16f (HEAD -> main, clean-up) 3

* 8a4e960 2

* 7548775 1

* a132b02 (expl) change on expl

* 4517d55 (feature) Add first file

* 252b289 First commit

(base) Sofi@MacBookAir lab1TRPZ %