



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні систем та технологій

Лабораторна робота №1
із дисципліни «*ТПІЗ*»
Тема: «Системи контролю версій. Git.»

Виконав:
Студент групи ІА-24
Бакалець А.І.

Перевірів:
Мягкий М. Ю.

Хід роботи

1. Ініціалізація git репозиторію

```
D:\Training\lab1>git init
Initialized empty Git repository in D:/Training/lab1/.git/

D:\Training\lab1>
```

2. Внесення і додавання змін

```
D:\Training\lab1>type nul > text.txt

D:\Training\lab1>git add .

D:\Training\lab1>
```

Крапка означає, що ми додаємо до стеїджу всі зміни

3. Створення комміту

```
D:\Training\lab1>git commit -m "init"
[master (root-commit) 2ee2ca7] init
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 text.txt

D:\Training\lab1>
```

4. Створення нової гілки

```
D:\Training\lab1>git branch songs-text

D:\Training\lab1>git branch
* master
  songs-text

D:\Training\lab1>
```

Можна використати тотожні команди

```
git switch -c name-of-branch
git checkout -b name-of-branch
```

5. Змінюємо поточну гілку і коммітимо зміни

```
D:\Training\lab1>git switch songs-text
Switched to branch 'songs-text'
```

```
≡ text.txt
    You, 3 seconds ago | 1 author (You)
1   SONG 1
2   SONG 2
3   SONG 3
4   SONG 4    You, 1 second ago
```

```
D:\Training\lab1>code .
```

```
D:\Training\lab1>git status
```

```
On branch songs-text
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git restore <file>..." to discard changes in working directory)
```

```
modified:   text.txt
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
D:\Training\lab1>git commit -ma "add songs text"
```

```
error: pathspec 'add songs text' did not match any file(s) known to git
```

```
D:\Training\lab1>git commit -am "add songs text"
```

```
[songs-text b4dd935] add songs text
```

```
1 file changed, 4 insertions(+)
```

```
D:\Training\lab1>|
```

6. Виконуємо fast-forward merge

```
D:\Training\lab1>git switch master
Switched to branch 'master'
```

```
D:\Training\lab1>git merge songs-text
```

```
Updating 2ee2ca7..b4dd935
```

```
Fast-forward
```

```
text.txt | 4 ++++
```

```
1 file changed, 4 insertions(+)
```

7. Перевіряємо історію коммітів

```
D:\Training\lab1>git log --oneline
```

```
b4dd935 (HEAD -> master, songs-text) add songs text
```

```
2ee2ca7 init
```

8. Створюємо ще одну гілку books-text і після цього внесемо певні зміни в master

```
D:\Training\lab1>git branch books-text
```

```
You, 1 second ago | 1 author (You)
1  SONG 1
2  SONG 2
3  SONG 3
4  SONG 4
5  SONG 5
6  SONG 6
7  :-> You, 1 second ago • Uncommitted changes
```

```
D:\Training\lab1>git commit -am "add necessary changes"
[master 5fbef14] add necessary changes
1 file changed, 4 insertions(+), 1 deletion(-)
```

9. Вносимо зміни в books-text і мерджимо

```
You, 23 seconds ago | 1 author (You)
1  BOOK 1
2  BOOK 2
3  BOOK 3
4  BOOK 4
5  You, 1 second ago • Uncommi
6
```

```
D:\Training\lab1>git checkout books-text
Switched to branch 'books-text'
```

```
D:\Training\lab1>git commit -am "add some books"
[books-text bc009f7] add some books
1 file changed, 5 insertions(+), 4 deletions(-)
```

```
D:\Training\lab1>git switch master
Switched to branch 'master'
```

```
D:\Training\lab1>git merge books-text
Auto-merging text.txt
CONFLICT (content): Merge conflict in text.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
text.txt
You, 16 seconds ago | 1 author (You) | Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
1  <<<<<< HEAD (Current Change)
2  SONG 1
3  SONG 2
4  SONG 3
5  SONG 4
6  SONG 5
7  SONG 6
8  :->
9  =====
10 BOOK 1
11 BOOK 2
12 BOOK 3
13 BOOK 4
14
15 >>>>>> books-text (Incoming Change)
16
```

```
D:\Training\lab1>git commit -am "merge #1"
[master a617263] merge #1
```

10. Видалимо зайвий branch

```
D:\Training\lab1>git branch -d songs-text
Deleted branch songs-text (was b4dd935).
```

11. Подимивось на попередні комміти з Detached-HEAD

```
D:\Training\lab1>git log --oneline
a617263 (HEAD -> master) merge #1
bc009f7 (books-text) add some books
5fbeb14 add necessary changes
b4dd935 add songs text
2ee2ca7 init
```

```
D:\Training\lab1>git checkout b4dd935
Note: switching to 'b4dd935'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable `advice.detachedHead` to `false`

HEAD is now at b4dd935 add songs text

```
You, 11 minutes ago | 1 author (You)
1  SONG 1
2  SONG 2
3  SONG 3
4  SONG 4      You, 11 minutes ago • add songs text
```

```
D:\Training\lab1>git switch -
Previous HEAD position was b4dd935 add songs text
Switched to branch 'master'
```

12. Використаємо cherry-pick

```

D:\Training\lab1>git switch -c feature-branch
Switched to a new branch 'feature-branch'

D:\Training\lab1>git commit -am "add feature text"
[feature-branch 18ad434] add feature text
1 file changed, 1 insertion(+)

D:\Training\lab1>git checkout master
Switched to branch 'master'

D:\Training\lab1>git cherry-pick 18ad434
[master 0dc2ff8] add feature text
Date: Fri Sep 20 01:15:49 2024 +0300
1 file changed, 1 insertion(+)

```

13. Зробимо rebase

```

D:\Training\lab1>git commit -am "add some bad staff"
[feature-branch 991eefc] add some bad staff
1 file changed, 4 insertions(+)

D:\Training\lab1>git commit -am "add more bad stuff"
[feature-branch 80aa11d] add more bad stuff
1 file changed, 2 insertions(+)

D:\Training\lab1>git switch master
Switched to branch 'master'

D:\Training\lab1>git rebase feature-branch
warning: skipped previously applied commit 0dc2ff8
hint: use --reapply-cherry-picks to include skipped commits
hint: Disable this message with "git config advice.skippedCherryPicks false"
Successfully rebased and updated refs/heads/master.

```

```

≡ text.txt
You, 1 minute ago | 1 author (You)
1  SONG 1
2  SONG 2
3  SONG 3
4  SONG 4
5  SONG 5
6  SONG 6
7  :->
8  HERE is the FEATURE
9  ERROR TEXT 1
10 ERROR TEXT 2
11
12
13
14
15 |

```

14. Видалимо помилкові комміти за допомогою reset

```
D:\Training\lab1>git reset --hard 991eefc
HEAD is now at 991eefc add some bad staff

D:\Training\lab1>git log --oneline
991eefc (HEAD -> master) add some bad staff
18ad434 add feature text
a617263 merge #1
bc009f7 (books-text) add some books
5fbeb14 add necessary changes
b4dd935 add songs text
2ee2ca7 init
```

15. Використаємо revert для зміни поганого комміту

```
You, 37 seconds ago | 1 author (You)
SONG 1
SONG 2
SONG 3
SONG 4
SONG 5
SONG 6
:->
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<< HEAD (Current Change)
HERE is the FEATURE
ERROR TEXT 1

=====
>>>>>> parent of 18ad434 (add feature text) (Incoming Change)
```

```

D:\Training\lab1>git revert HEAD~1
Auto-merging text.txt
CONFLICT (content): Merge conflict in text.txt
error: could not revert 18ad434... add feature text
hint: After resolving the conflicts, mark them with
hint: "git add/rm <paths>", then run
hint: "git revert --continue".
hint: You can instead skip this commit with "git revert --skip".
hint: To abort and get back to the state before "git revert",
hint: run "git revert --abort".

D:\Training\lab1>git status
On branch master
You are currently reverting commit 18ad434.
  (fix conflicts and run "git revert --continue")
  (use "git revert --skip" to skip this patch)
  (use "git revert --abort" to cancel the revert operation)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
    both modified:   text.txt

no changes added to commit (use "git add" and/or "git commit -a")

D:\Training\lab1>git commit -am "revert bad commit"
[master b59a70e] revert bad commit
 1 file changed, 5 deletions(-)

D:\Training\lab1>

D:\Training\lab1>git log --oneline
b59a70e (HEAD -> master) revert bad commit
991eefc add some bad stuff
18ad434 add feature text
a617263 merge #1
bc009f7 (books-text) add some books
5fbef14 add necessary changes
b4dd935 add songs text
2ee2ca7 init

```

Висновок: виконуючи дану лабораторну роботу, я розглянув основні команди для роботи з СКВ гіт.