



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

Лабораторна робота №1

з дисципліни «Технології розроблення програмного забезпечення»

Тема: Системи контролю версій. Git.

Виконала:
студентка групи ІА-23
Проценко. В. І.

Перевірив:
Мягкий М. Ю.

Київ 2024

Теоретичні відомості

Системи контролю версій — це інструменти, які дозволяють відстежувати зміни в файлах і керувати різними версіями проєктів. Вони використовуються для збереження історії змін, співпраці між розробниками та управління конфліктами.

Основні концепції Git:

- **Репозиторій:**
Локальний репозиторій — містить повну історію проєкту на комп'ютері розробника.
Віддалений репозиторій — копія репозиторію на сервері, яка використовується для співпраці.
- **Гілки:**
Гілки дозволяють працювати над різними частинами проєкту одночасно, без ризику впливу на основну версію.
Основна гілка зазвичай називається `main` або `master`.
Інші гілки використовуються для розробки нових функцій, виправлення помилок тощо.
- **Коміти (Commits):**
Кожен коміт — це знімок проєкту на певний момент часу, що зберігає зміни.
Коміт містить опис змін та ідентифікатор автора.
- **Merge і Rebase:**
`Merge` — злиття двох гілок для об'єднання змін.
`Rebase` — перенесення комітів з однієї гілки на іншу для збереження лінійної історії.
- **Cherry-pick** — перенесення вказаних комітів на поточну гілку.
- **Основні команди Git:**
`git init` — створює новий локальний репозиторій.
`git clone <url>` — копіює віддалений репозиторій на локальний комп'ютер.
`git status` — показує поточний стан репозиторію.
`git add <file>` — додає зміни до індексу (стейджинг).
`git commit -m "<message>"` — фіксує зміни з описом.
`git push` — відправляє зміни на віддалений сервер.
`git pull` — отримує останні зміни з віддаленого репозиторію.
`git branch` — відображає список гілок.
`git branch <branch>` — створення гілки.
`git checkout <branch>` / `git switch <branch>` — перемикається на іншу гілку.
`git merge <branch>` — зливає гілку в поточну.

Виконання

```
PS D:\nikap\Documents\GIT\githowto\repositories\test> git init
Initialized empty Git repository in D:/nikap/Documents/GIT/githowto/repositories/test/.git/
PS D:\nikap\Documents\GIT\githowto\repositories\test> git add 1.txt
PS D:\nikap\Documents\GIT\githowto\repositories\test> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   1.txt

PS D:\nikap\Documents\GIT\githowto\repositories\test> git commit -m "Init"
[main (root-commit) 431f926] Init
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 1.txt
PS D:\nikap\Documents\GIT\githowto\repositories\test> gi

PS D:\nikap\Documents\GIT\githowto\repositories\test> git branch branch1
PS D:\nikap\Documents\GIT\githowto\repositories\test> git checkout -b branch2
Switched to a new branch 'branch2'
PS D:\nikap\Documents\GIT\githowto\repositories\test> git status
On branch branch2
nothing to commit, working tree clean
PS D:\nikap\Documents\GIT\githowto\repositories\test> git branch
  branch1
* branch2
  main
```

```
PS D:\nikap\Documents\GIT\githowto\repositories\test> git switch branch1
Switched to branch 'branch1'
PS D:\nikap\Documents\GIT\githowto\repositories\test> git status
On branch branch1
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:   1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Folder/

no changes added to commit (use "git add" and/or "git commit -a")
PS D:\nikap\Documents\GIT\githowto\repositories\test>
PS D:\nikap\Documents\GIT\githowto\repositories\test>
PS D:\nikap\Documents\GIT\githowto\repositories\test>
PS D:\nikap\Documents\GIT\githowto\repositories\test>
PS D:\nikap\Documents\GIT\githowto\repositories\test> git add .
PS D:\nikap\Documents\GIT\githowto\repositories\test> git commit -m "1 commit"
[branch1 fdb11c] 1 commit
 3 files changed, 1 insertion(+)
 create mode 100644 Folder/2.txt
 create mode 100644 Folder/3.txt
```

```

PS D:\nikap\Documents\GIT\githowto\repositories\test> git add 1.txt
PS D:\nikap\Documents\GIT\githowto\repositories\test> git commit -m "change to 1.txt"
[branch1 d467a23] change to 1.txt
 1 file changed, 1 insertion(+), 1 deletion(-)
PS D:\nikap\Documents\GIT\githowto\repositories\test> git log branch1
commit d467a2323da619b6cb1343f3f8eff3849155a5f7 (HEAD -> branch1)
Author: Veronika Protsenko <protsenko.veronika@lil.kpi.ua>
Date: Thu Sep 19 11:57:24 2024 +0200

    change to 1.txt

commit fdba11c9d3242925fac762d24f7253bb4b52761d
Author: Veronika Protsenko <protsenko.veronika@lil.kpi.ua>
Date: Thu Sep 19 11:54:18 2024 +0200

    1 commit

commit 431f926f26397a42e076661dd319160f2a16d67f (main, branch2)
Author: Veronika Protsenko <protsenko.veronika@lil.kpi.ua>
Date: Thu Sep 19 11:47:28 2024 +0200

    Init

```

```

PS D:\nikap\Documents\GIT\githowto\repositories\test> git switch main
Switched to branch 'main'
PS D:\nikap\Documents\GIT\githowto\repositories\test> git branch branch3
PS D:\nikap\Documents\GIT\githowto\repositories\test> git add .
PS D:\nikap\Documents\GIT\githowto\repositories\test> git commit -m "3 lines 1.txt"
[main 4f22f4f] 3 lines 1.txt
 1 file changed, 3 insertions(+)
PS D:\nikap\Documents\GIT\githowto\repositories\test> git log branch1
commit d467a2323da619b6cb1343f3f8eff3849155a5f7 (branch1)
Author: Veronika Protsenko <protsenko.veronika@lil.kpi.ua>
Date: Thu Sep 19 11:57:24 2024 +0200

    change to 1.txt

commit fdba11c9d3242925fac762d24f7253bb4b52761d
Author: Veronika Protsenko <protsenko.veronika@lil.kpi.ua>
Date: Thu Sep 19 11:54:18 2024 +0200

    1 commit

commit 431f926f26397a42e076661dd319160f2a16d67f (branch3, branch2)
Author: Veronika Protsenko <protsenko.veronika@lil.kpi.ua>
Date: Thu Sep 19 11:47:28 2024 +0200

    Init

```

```
PS D:\nikap\Documents\GIT\githowto\repositories\test> git cherry-pick d467a2323da619b6cb1341
Auto-merging 1.txt
CONFLICT (content): Merge conflict in 1.txt
error: could not apply d467a23... change to 1.txt
hint: After resolving the conflicts, mark them with
hint: "git add/rm <paths>", then run
hint: "git cherry-pick --continue".
hint: You can instead skip this commit with "git cherry-pick --skip".
hint: To abort and get back to the state before "git cherry-pick",
hint: run "git cherry-pick --abort".
hint: Disable this message with "git config advice.mergeConflict false"
PS D:\nikap\Documents\GIT\githowto\repositories\test> git add 1.txt
PS D:\nikap\Documents\GIT\githowto\repositories\test> git commit -m "Resolved conflict"
[main df6e2fb] Resolved conflict
Date: Thu Sep 19 11:57:24 2024 +0200
1 file changed, 2 insertions(+), 1 deletion(-)
```

Висновок: У ході виконання лабораторної роботи ми освоїли основні команди і концепції системи контролю версій Git та виконали поставлене викладачем завдання зі створення локального репозиторію, гілок та маніпулювання комітами.