

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Кафедра Програмного забезпечення автоматизованих систем

Лабораторна робота №2
Тема: Робота з системами керування Git

Виконав:

ст. групи ПІ-14-1

Вінтоняк Володимир

Перевірив:

Бандура В.В.

Мета: здобути необхідні знання та навички для роботи з системою керування версіями Git.

Завдання:

1. Зареєструватися на GitHub.
2. Завантажити ПЗ для роботи з Git.
3. Встановити на робочий компютер.
4. Створити репозиторій на GitHub.
5. Створити локальний репозиторій.
6. Додати в локальний репозиторій файл.
7. Закомітати даний файл в лок.репозиторій.
8. Закинути зміни з лок.репозиторія в репозиторій, який ви створили в п.4.
9. Додати два файли в лок.репозиторій. Закомітайте і запушуйте на GitHub.
10. Відновіться до стану, який був в п.8.
11. Зробіть клон репозиторія з GitHub в іншу папку на вашому комп'ютері.
12. Створіть нову гілку і додайте в неї новий файл.
13. Закомітайте зміни в новій бранчі. Синхронізуйте з репозиторієм на GitHub.
14. Об'єднайте створену гілку з основною.
15. Змініть файл, який ви додали в п.12 в основній бранчі.
16. Переключіться на гілку, яку ви створили в п.12 і також зробіть зміни в файлі, що додали в п.12.
17. Об'єднайте гілку з основною. Конфлікт файлу вирішіть в ручну.

Короткі теоретичні відомості:

GitHub — один з найбільших веб-сервісів для спільної розробки програмного забезпечення. Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub, Inc (раніше Logical Awesome).


Сервіс безкоштовний для проектів з відкритим вихідним кодом, з наданням користувачам усіх своїх можливостей (включаючи SSL), а для окремих індивідуальних проектів пропонуються різні платні тарифні плани. 21 вересня 2011 року кількість користувачів стала більшою за мільйон.


Хід виконання роботи:


1. Зареєструватися на GitHub.

Join GitHub

The best way to design, build, and ship software.

 **Step 1:**
Set up a personal account

 **Step 2:**
Choose your plan

 **Step 3:**
Go to your dashboard

Create your personal account

Username

This will be your username — you can enter your organization's username next.

Email Address

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

You'll love GitHub

- Unlimited** collaborators
- Unlimited** public repositories
- ✓ Great communication
- ✓ Friction-less development
- ✓ Open source community

2. Завантажити ПЗ для роботи з Git.

GitHub Desktop

[Overview](#) [Release Notes](#) [Help](#)

Simple collaboration from your desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise.

Available for Mac and [Windows](#)


[Download GitHub Desktop](#)
Windows 7 or later

By clicking the Download button you agree to the End-User License Agreement

(23%) Installing GitHub

Installing GitHub

This may take several minutes. You can use your computer to do other tasks during the installation.


 **Name:** [GitHub](#)

From: [github-windows.s3.amazonaws.com](#)

Downloading: 25.9 MB of 109 MB

Cancel

3. Встановити на робочий комп'ютер.



Filter repositories

Tutorial

Welcome

Log in

Configure

Repositories

Configure git

This will be used in the commits you create. Keep in mind that if you publish commits, anyone will have access to this email.

This will change your global gitconfig.

✓ Continue

✗ Skip


4. Створити репозиторій на GitHub.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

 RichKa

 /

alg_1

Great repository names are short and memorable. Need inspiration? How about [miniature-octo-fiesta](#).

Description (optional)

lab_1

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None**

Add a license: **None**

Create repository

5. Створити локальний репозиторій.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit
$ git init
Initialized empty Git repository in C:/My Documents/GitHub/NewGit/.git/
```

6. Додати в локальний репозиторій файл.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git add one.txt
```

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git status
On branch master
```

Initial commit

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

    new file:   one.txt
```

7. Закомітати даний файл в лок.репозиторій.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git commit -m 'First commit, one.txt added'
[master b82f87a] First commit, one.txt added
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 one.txt
```

8. Закинути зміни з лок.репозиторія в репозиторій, який ви створили в п.4.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git remote add origin https://github.com/RichKa/NewGit.git
fatal: remote origin already exists.
```

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git push -u origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 295 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/RichKa/NewGit.git
 ee435c9..b82f87a  master -> master
Branch master set up to track remote branch master from origin.
```

9. Додати два файли в лок.репозиторій. Закомітайте і запусуйте на GitHub.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit (master)
$ git add two.txt three.txt

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   three.txt
        new file:   two.txt

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit (master)
$ git commit -m '2 commit, two files added'
[master 4164776] 2 commit, two files added
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 three.txt
create mode 100644 two.txt

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit (master)
$ git push -u origin master
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 293 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To https://github.com/RichKa/NewGit.git
   b82f87a..4164776  master -> master
Branch master set up to track remote branch master from origin.
```

10. Відновіться до стану, який був в п.8.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit (master)
$ git log
commit 4164776d865688e49a7e4fe20c9e3dfce6e9fab3
Author: Volodymyr Uintonyak <v.vint7@gmail.com>
Date:   Mon Feb 29 20:28:19 2016 +0200

    2 commit, two files added

commit b82f87ac4510d911c920d71eaa009a7fa5831f2f
Author: Volodymyr Uintonyak <v.vint7@gmail.com>
Date:   Mon Feb 29 20:20:11 2016 +0200

    First commit, one.txt added

commit ee435c95106b8115b0550979b36a31b5e259caf8
Author: Volodymyr Uintonyak <v.vint7@gmail.com>
Date:   Mon Feb 29 18:46:16 2016 +0200

    Initial commit
```

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit (master)
$ git checkout 4164
Note: checking out '4164'.

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 performing another checkout.

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

    git checkout -b <new-branch-name>

HEAD is now at 4164776... 2 commit, two files added
```

11. Зробіть клон репозиторія з GitHub в іншу папку на вашому комп'ютері.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit (<4164776...>)
$ git clone https://github.com/RichKa/NewGit.git NewGit1
Cloning into 'NewGit1'...
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 8 (delta 1), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), done.
Checking connectivity... done.
```

12. Створіть нову гілку і додайте в неї новий файл.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <<4164776...>>
$ git checkout -b Second
Switched to a new branch 'Second'

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <Second>
$ git add NewFile.txt
```

13. Закомітайте зміни в новій бранчі. Синхронізуйте з репозиторієм на GitHub.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <Second>
$ git commit -m '3 commit, NewFile.txt added'
[Second 5036c1b] 3 commit, NewFile.txt added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 NewFile.txt

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <Second>
$ git push origin Second
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 259 bytes | 0 bytes/s, done.
Total 2 (delta 1), reused 0 (delta 0)
To https://github.com/RichKa/NewGit.git
 * [new branch]      Second -> Second
```

14. Об'єднайте створену гілку з основною.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <Second>
$ git merge master
Already up-to-date.
```

15. Змініть файл, який ви додали в п.12 в основній бранчі.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git add NewFile.txt

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git commit -m '4 commit, NewFile.txt edited'
[master c2d3f54] 4 commit, NewFile.txt edited
1 file changed, 1 insertion(+)
create mode 100644 NewFile.txt

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 302 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/RichKa/NewGit.git
 4164776..c2d3f54 master -> master

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
```

16. Переключіться на гілку, яку ви створили в п.12 і також зробіть зміни в файлі, що додали в п.12.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git checkout second
Switched to branch 'second'

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second>
$ git add NewFile.txt

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second>
$ git commit -m '5 commit, NewFile edited'
[second 6e57a02] 5 commit, NewFile edited
1 file changed, 1 insertion(+)

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second>
$ git push origin second
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 300 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/RichKa/NewGit.git
 * [new branch]      second -> second
```

17. Об'єднайте гілку з основною. Конфлікт файлу вирішіть в ручну.

```
Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second>
$ git merge master
Auto-merging NewFile.txt
CONFLICT (add/add): Merge conflict in NewFile.txt
Automatic merge failed; fix conflicts and then commit the result.

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second!MERGING>
$ git merge master
error: merge is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second!MERGING>
$ git add NewFile.txt

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second!MERGING>
$ git commit -m '6 commit, newfile conflict'
[second 1be0e96] 6 commit, newfile conflict

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second>
$ git merge master
Already up-to-date.

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second>
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git push origin master
Everything up-to-date

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <master>
$ git checkout second
Switched to branch 'second'

Uova@Uintonyak-PC MINGW64 /c/My Documents/GitHub/NewGit <second>
$ git push origin second
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 352 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/RichKa/NewGit.git
  6e57a02..1be0e96  second -> second
```

Висновок: на даній лабораторній роботі я здобув необхідні знання та навички для роботи з системою керування версіями Git.