

КІТ-33 Стрельцов Віталій Вікторович
група ПІБ студента

04.05.2015
Дата виконання

11.05.2015
Дата прийому

Оцінка

ЗВІТ

Лабораторна робота №3

Тема: Основи роботи з вітками в GIT. Основи злиття.
Розв'язування конфліктів.

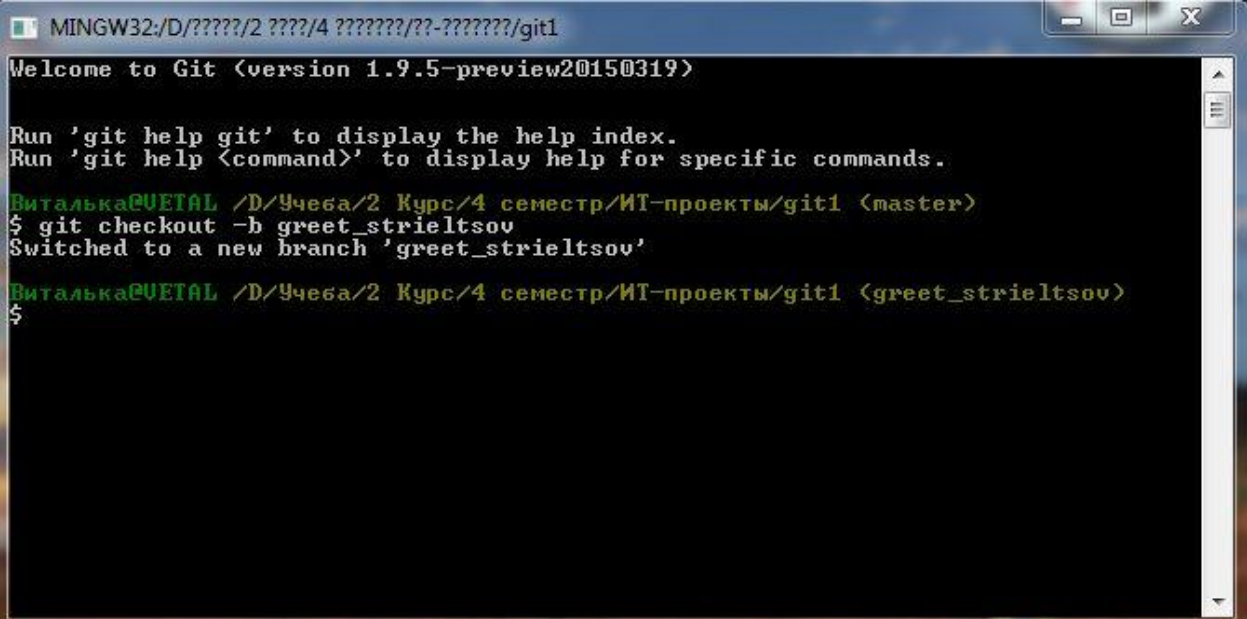
Харків 2015

Мета: Познайомитись з основами роботи з вітками в Git. Освоїти процеси злиття віток та розв'язування конфліктів.

Хід роботи

1. Створення гілок і їх злиття

- 1) Створюємо робочий каталог на диску С з назвою «git». Зайдемо в цей каталог, клік правою кнопкою миші, натискаємо на «Git Init Here». Клікаємо правою кнопкою миші на каталозі і вибираємо «Git Bash». Відкриється консоль для роботи з Git.
- 2) Створюємо нову гілку `greet_strieltsov`, використовуючи команду `git checkout -b <branchname>`.



```
MINGW32:/D/????/2 ???/4 ?????/??-????/git1
Welcome to Git (version 1.9.5-preview2@150319)

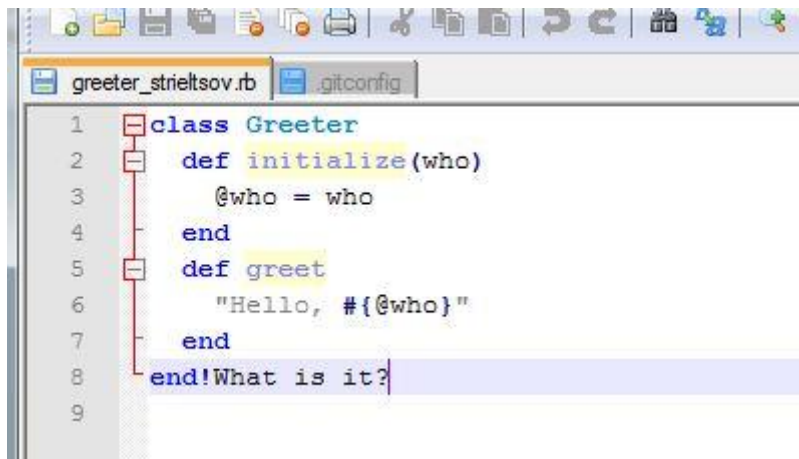
Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$ git checkout -b greet_strieltsov
Switched to a new branch 'greet_strieltsov'

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$
```

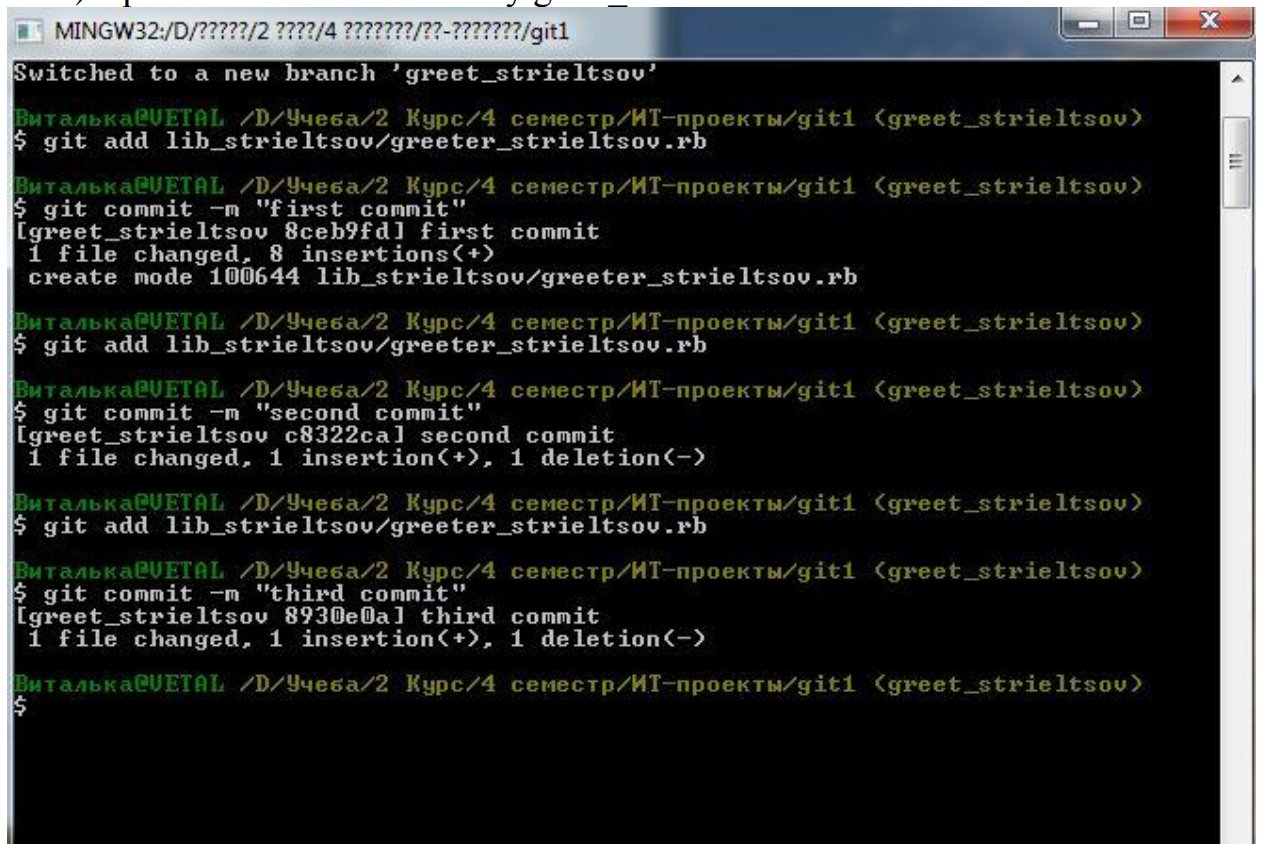
- 3) Додаємо новий каталог `lib_strieltsov` і файл `greeter_strieltsov.rb` із вмістом:

```
class Greeter
  def initialize (who)
    who = who
  end
  def greet
    "Hello, # {@ who}"
  end
end
```



```
1 class Greeter
2   def initialize(who)
3     @who = who
4   end
5   def greet
6     "Hello, #{@who}"
7   end
8 end!What is it?
```

4) Зробимо 3 commits в гілку greet_strieltsov.



```
MINGW32:/D/????/2 ???/4 ?????/??-????/git1
Switched to a new branch 'greet_strieltsov'
Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git add lib_strieltsov/greeter_strieltsov.rb
Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git commit -m "first commit"
[greet_strieltsov 8ceb9fd] first commit
1 file changed, 8 insertions(+)
create mode 100644 lib_strieltsov/greeter_strieltsov.rb
Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git add lib_strieltsov/greeter_strieltsov.rb
Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git commit -m "second commit"
[greet_strieltsov c8322ca] second commit
1 file changed, 1 insertion(+), 1 deletion(-)
Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git add lib_strieltsov/greeter_strieltsov.rb
Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git commit -m "third commit"
[greet_strieltsov 8930e0a] third commit
1 file changed, 1 insertion(+), 1 deletion(-)
Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$
```

5) Продемонструємо перемикання на гілку master.

```

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git checkout master
Switched to branch 'master'

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$

```

6) Створюємо файл README і зробимо commit в master.

```

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$ git add README.txt

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$ git commit -m "4th commit"
[master f9fe65f] 4th commit
1 file changed, 1 insertion(+)
create mode 100644 README.txt

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$

```

7) Проводимо злиття гілок greet і master - продемонструємо історію commits.

```

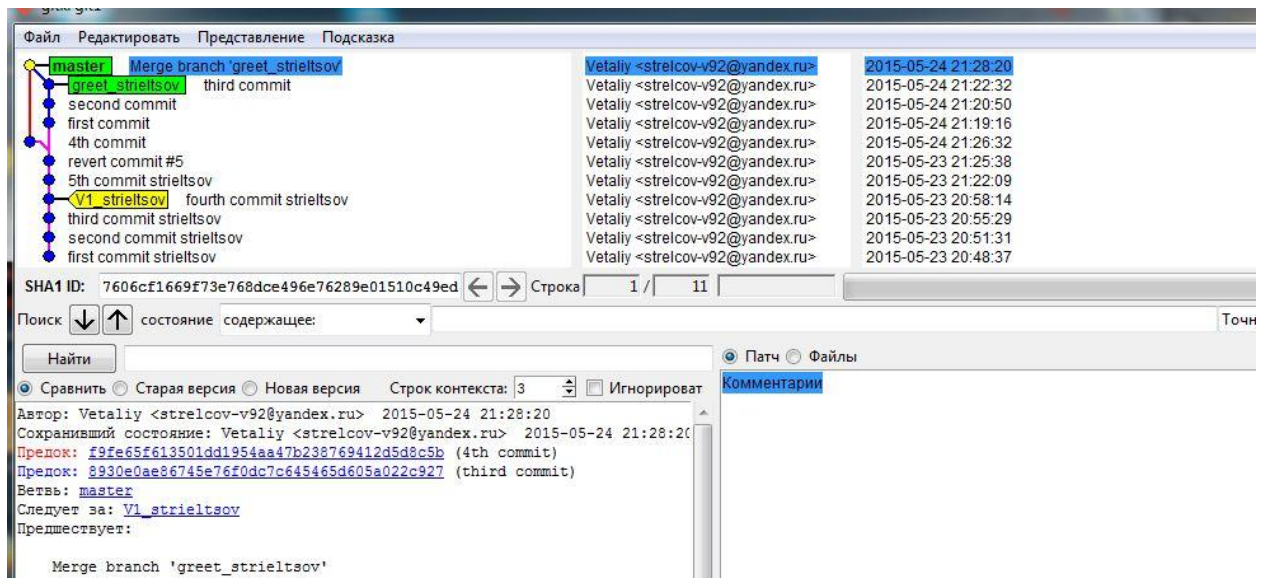
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$ git hist --all
fatal: Invalid object name '--pretty=format'.

Warning: Your console font probably doesn't support Unicode. If you experience s
trange characters in the output, consider switching to a TrueType font such as L
ucida Console!

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$ git hist --all
* 7606cf1 2015-05-24 | Merge branch 'greet_strieltsov' (HEAD, master) [Uetaliy]
* 8930e0a 2015-05-24 | third commit (greet_strieltsov) [Uetaliy]
* c8322ca 2015-05-24 | second commit [Uetaliy]
* 8ceb9fd 2015-05-24 | first commit [Uetaliy]
* f9fe65f 2015-05-24 | 4th commit [Uetaliy]
* b38c541 2015-05-23 | revert commit #5 [Uetaliy]
* 5e75c24 2015-05-23 | 5th commit strieltsov [Uetaliy]
* 897e6ae 2015-05-23 | fourth commit strieltsov (tag: U1_strieltsov) [Uetaliy]
* caf0fab 2015-05-23 | third commit strieltsov [Uetaliy]
* 90f5494 2015-05-23 | second commit strieltsov [Uetaliy]
* efce154 2015-05-23 | first commit strieltsov [Uetaliy]
<END>

```

8) Зобразимо у графічному редакторі граф комітів, де вершинами графа будуть коди комітів.



2. Конфлікти

- 1) Змодельюємо штучно конфлікт, при злитті описаному в попередній частині.

```

MINGW32/D/????/2 ???/4 ?????/??-????/git1

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 <master>
$ git checkout greet_strieltsov
Switched to branch 'greet_strieltsov'

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 <greet_strieltsov>
$ git add README.txt

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 <greet_strieltsov>
$ git commit -m "5th commit"
[greet_strieltsov c0a5a21] 5th commit
1 file changed, 1 insertion(+)
create mode 100644 README.txt

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 <greet_strieltsov>
$ git checkout master
Switched to branch 'master'

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 <master>
$ git add README.txt

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 <master>
$ git commit -m "6th commit"
[master 2dia306] 6th commit
1 file changed, 1 insertion(+), 1 deletion(-)

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 <master>
$ git merge greet_strieltsov
Auto-merging README.txt
CONFLICT (add/add): Merge conflict in README.txt
Automatic merge failed; fix conflicts and then commit the result.

```

- 2) Використовуючи команду reset повернемо покажчик на стан гілок до їх злиття.

```

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 <master!MERGING>
$ git reset --hard 8930e0a
HEAD is now at 8930e0a third commit

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 <master>
$

```

- 3) Виконаємо кроки 4-6 з частини 1. Виконаємо злиття змін командою rebase. Зобразимо у графічному редакторі граф комітів, де вершинами графа будуть коди комітів.

```
MINGW32/D/????/2 ???/4 ?????/??-????/git1
Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git add README.txt
warning: You ran 'git add' with neither '-A (--all)' or '--ignore-removal',
whose behaviour will change in Git 2.0 with respect to paths you removed.
Paths like 'README.txt' that are
removed from your working tree are ignored with this version of Git.

* 'git add --ignore-removal <paths>', which is the current default,
ignores paths you removed from your working tree.

* 'git add --all <paths>' will let you also record the removals.

Run 'git status' to check the paths you removed from your working tree.

Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git add README1.txt

Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git commit -m "7th commit"
[greet_strieltsov 4a26a15] 7th commit
1 file changed, 1 insertion(+)
create mode 100644 README1.txt

Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git add README1.txt

Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git commit -m "8th commit"
[greet_strieltsov cd7b2fd] 8th commit
1 file changed, 1 insertion(+), 1 deletion(-)

Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git add README1.txt

Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git commit -m "8th commit"
[greet_strieltsov 70194bb] 8th commit
1 file changed, 1 insertion(+), 1 deletion(-)

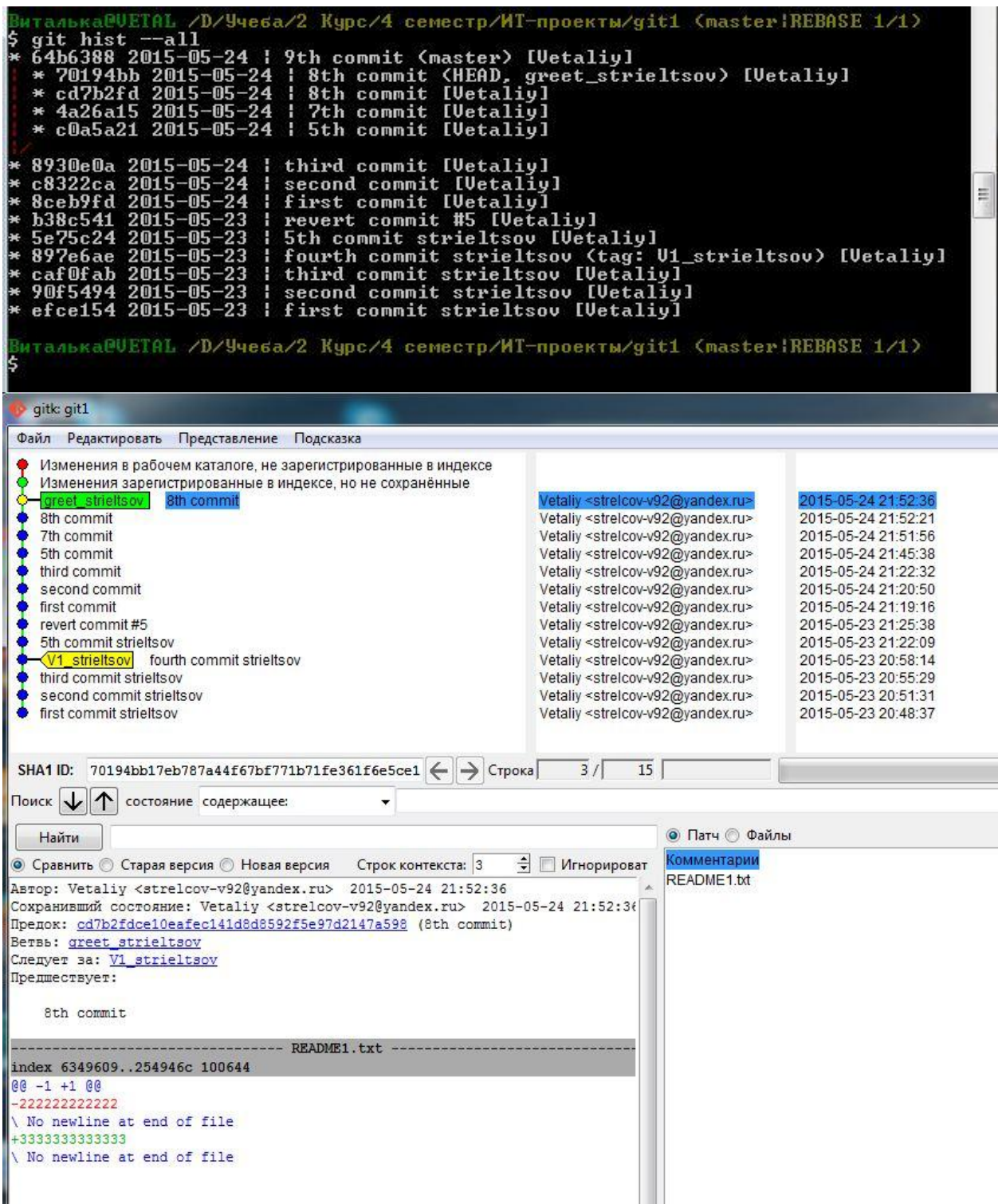
Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (greet_strieltsov)
$ git checkout master
Switched to branch 'master'

Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$ git add README.txt

Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$ git commit -m "9th commit"
[master 64b6388] 9th commit
1 file changed, 1 insertion(+)
create mode 100644 README.txt

Виталька@VETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/git1 (master)
$ git rebase greet_strieltsov
First, rewinding head to replay your work on top of it...
Applying: 9th commit
Using index info to reconstruct a base tree...
Falling back to patching base and 3-way merge...
Auto-merging README.txt
CONFLICT (add/add): Merge conflict in README.txt
Failed to merge in the changes.
Patch failed at 0001 9th commit
The copy of the patch that failed is found in:
d:/Учеба/2 Курс/4 семестр/ИТ-проекты/git1/.git/rebase-apply/patch
Warning: Your console font probably doesn't support Unicode. If you experience s
trange characters in the output, consider switching to a TrueType font such as L
ucida Console!

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --abort".
```

Як бачимо граф rebase має лінійну структуру, а граф merge – розгалужену.

- 4) Для початку роботи зареєструємось на віддаленому сервісі github та ініціалізуємо віддалений репозиторій.
- 5) Створюємо перший робочий каталог на диску C з назвою «github». Зайдемо в цей каталог, клік правою кнопкою миші, натискаємо на «Git Init Here». Клікаємо правою кнопкою миші на каталозі і вибираємо «Git Bash». Відкриється консоль для роботи з Git. Створюємо там декілька файлів та заливаємо весь вміст папки у віддалений репозиторій.

```
MINGW32:/D/????/2 ???/4 ?????/??-????/github1
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github1 (master)
$ git add README.txt

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github1 (master)
$ git commit -m "first commit"
[master (root-commit) e7a7bb2] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.txt

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github1 (master)
$ git remote add origin https://github.com/Uetaliy/strieltsov.git

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github1 (master)
$ git push -u origin master
Username for 'https://github.com': Uetaliy
Password for 'https://Uetaliy@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 221 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Uetaliy/strieltsov.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github1 (master)
$
```

- б) Створюємо другий робочий каталог на диску С з назвою «github2». Зайдемо в цей каталог, клік правою кнопкою миші, натискаємо на «Git Init Here». Клікаємо правою кнопкою миші на каталозі і вибираємо «Git Bash». Відкриється консоль для роботи з Git. Цей репозиторій «забирає» проект із віддаленого репозиторія до себе в локальний репозиторій.

```
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github2 (master)
$ git remote add origin https://github.com/Uetaliy/strieltsov.git

Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github2 (master)
$ git pull -u origin master
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/Uetaliy/strieltsov
 * branch            master       -> FETCH_HEAD
 * [new branch]      master       -> origin/master

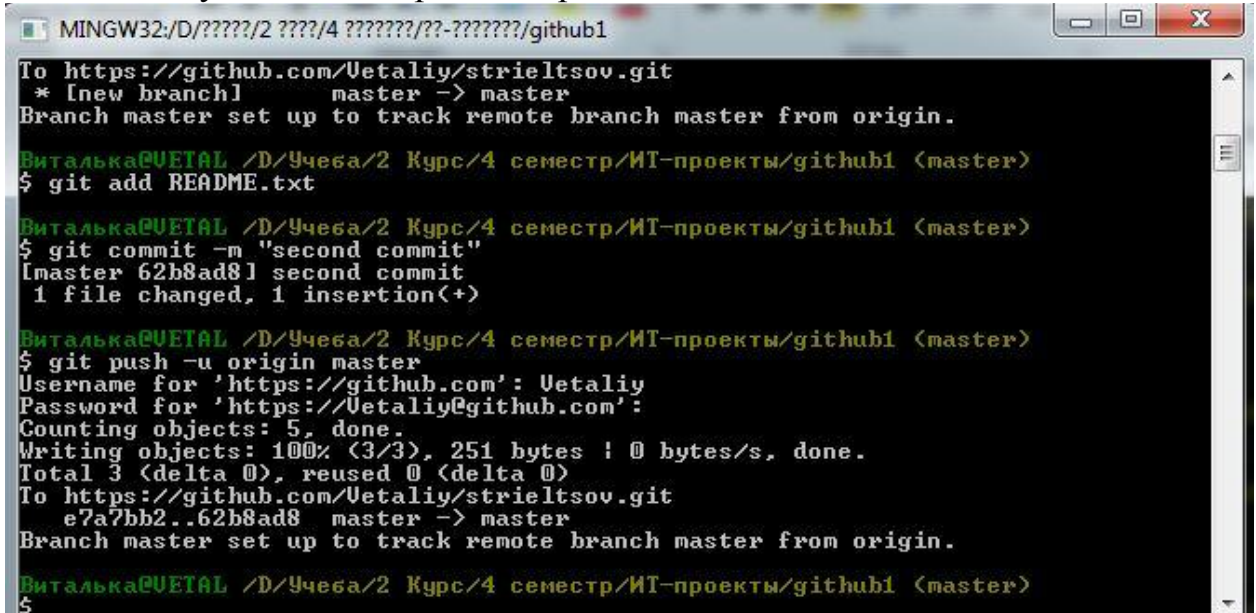
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github2 (master)
$
```

- 7) Виконаємо команду git status.

```
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github2 (master)
$ git status
On branch master
nothing to commit, working directory clean

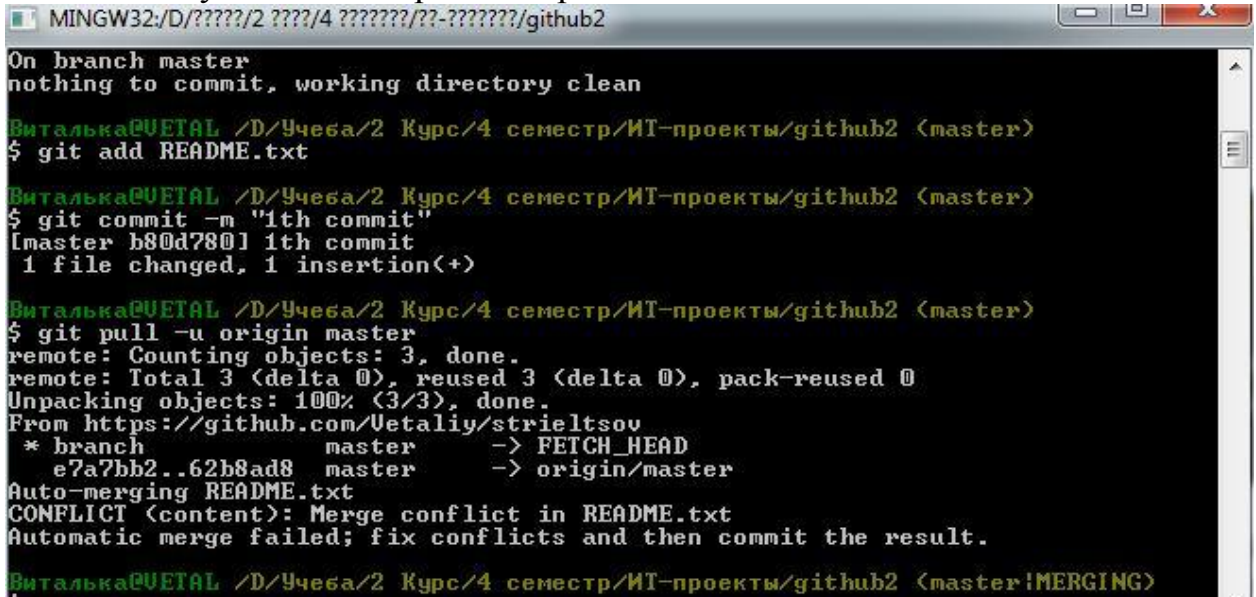
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github2 (master)
$
```


- 8) Проведемо у першому локальному репозиторії три коміти і «зальємо» зміни у віддалений репозиторій.



```
MINGW32/D:/????/2 ???/4 ?????/??-????/github1
To https://github.com/Uetaliy/strieltsov.git
* [new branch]      master -> master
Branch master set up to track remote branch master from origin.
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github1 <master>
$ git add README.txt
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github1 <master>
$ git commit -m "second commit"
[master 62b8ad8] second commit
1 file changed, 1 insertion(+)
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github1 <master>
$ git push -u origin master
Username for 'https://github.com': Uetaliy
Password for 'https://Uetaliy@github.com':
Counting objects: 5, done.
Writing objects: 100% (3/3), 251 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Uetaliy/strieltsov.git
e7a7bb2..62b8ad8  master -> master
Branch master set up to track remote branch master from origin.
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github1 <master>
$
```

- 9) Проведемо у другому локальному репозиторії три коміти і «зальємо» зміни у віддалений репозиторій.



```
MINGW32/D:/????/2 ???/4 ?????/??-????/github2
On branch master
nothing to commit, working directory clean
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github2 <master>
$ git add README.txt
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github2 <master>
$ git commit -m "1th commit"
[master b80d780] 1th commit
1 file changed, 1 insertion(+)
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github2 <master>
$ git pull -u origin master
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/Uetaliy/strieltsov
* branch      master      -> FETCH_HEAD
e7a7bb2..62b8ad8  master    -> origin/master
Auto-merging README.txt
CONFLICT (content): Merge conflict in README.txt
Automatic merge failed; fix conflicts and then commit the result.
Виталька@UETAL /D/Учеба/2 Курс/4 семестр/ИТ-проекты/github2 <master!MERGING>
$
```

Як бачимо нам виводиться повідомлення про конфлікт.

Якщо ми зайдемо на віддалений сервіс github, то побачимо, що «залились» лише зміни проведені у першому локальному репозиторії. Отже, можна зробити висновок, що після виникненні конфлікту зміни не зберігаються.

Висновки: На даній лабораторній роботі я ознайомився з основами роботи з вітками в Git, а також освоїв процеси злиття віток та розв'язування конфліктів.