Государственный Университет Молдовы

Факультет Математики и Информатики

Департамент Информатики

“Inițiere în Cloud Computing și Sisteme de Operare Distribuite”

Лабораторная работа 4  
“Команды Git, часть II. SourceTree ”

Проверил: Препелица Аурелия

Выполнил: Чобану Артём

Кишинев 2021

# Требования к лабораторной работе:

- Установка Git, GitBash

- Установка SourceTree

- Генерация SSH-ключа и добавление в аккаунт GitHub

Выполнение команд git с использованием графического интерфейса SourceTree

1. Создание публичного репозитория GitHub и веток слежения:

* git remote add
* git push -u origin master

2. Слияние / перебазирование веток git:

• git merge –squash

• git rebase

3. Разрешение конфликтов.

4. Перезапись комита:

• git –amend

5. Копирование изменений коммита:

• git cherry-pick

6. Отмена (реверсирование) изменений коммита:

• git revert

7. Отмена локальных изменений статуса репозитория Git:

• git reset (--soft, --mixed, --hard)

* Установка Git, GitBash:

Graphical user interface, application, email, website

Description automatically generated

* Установка SourceTree:

Graphical user interface, text, application

Description automatically generated

* Генерация SSH – ключа и добавление в аккаунт GitHub:

Сначала перейдём в директорию .ssh:

Text

Description automatically generated

Создадим ключ, введя название файла, а также введя секретную фразу:

Text

Description automatically generated

Создавшийся ключ:

Graphical user interface, text

Description automatically generated

Добавим ключ на GitHub:

Text

Description automatically generated with low confidence

Вставим ключ из файла github\_rsa:

Graphical user interface, text, application, email

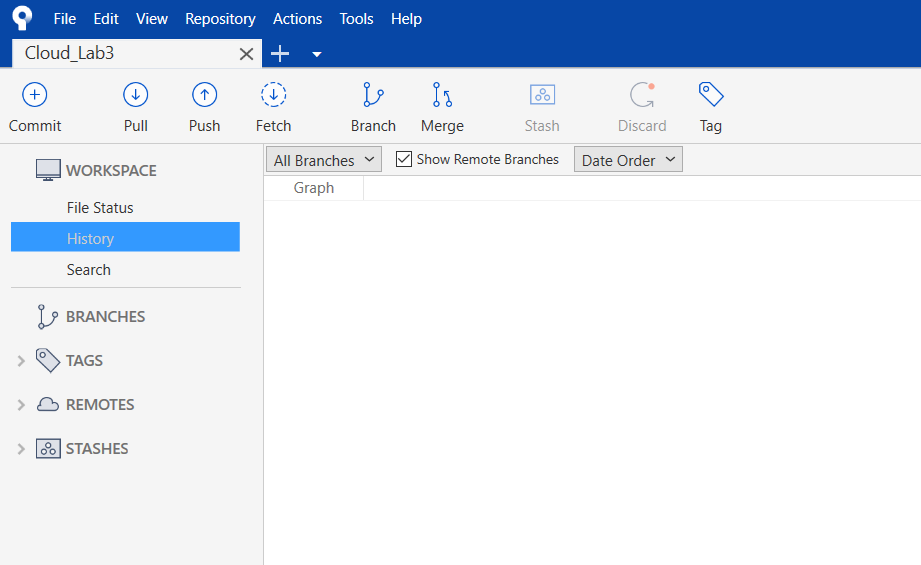
Description automatically generated

Теперь ключ отображается в списке:

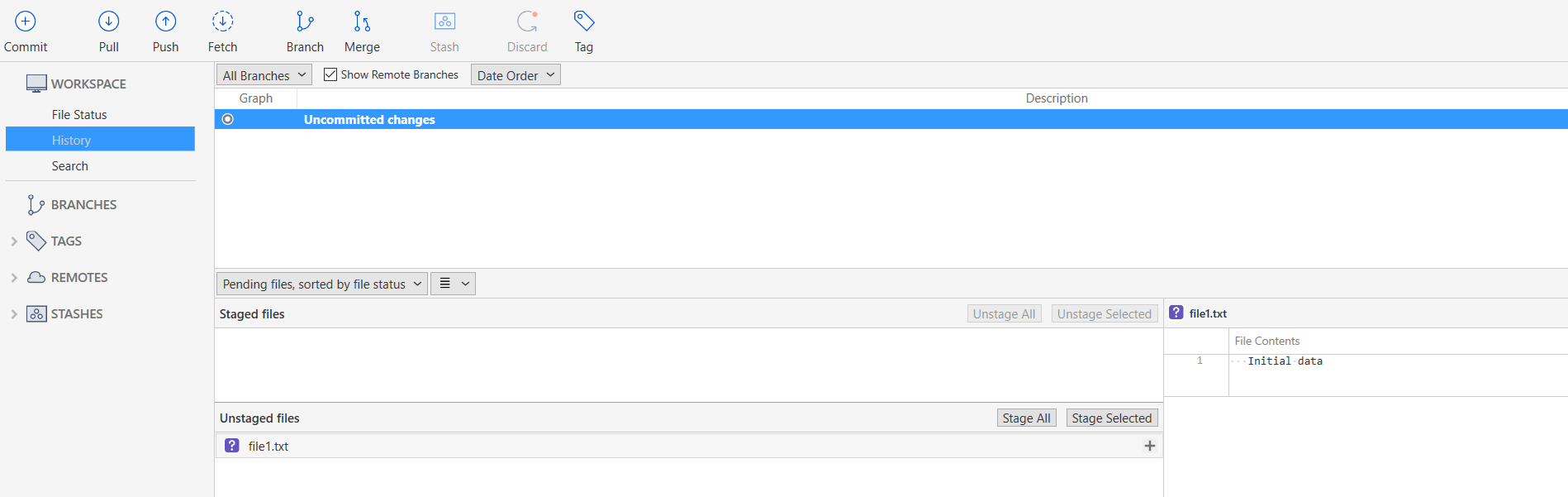
Graphical user interface, text, application, email

Description automatically generated

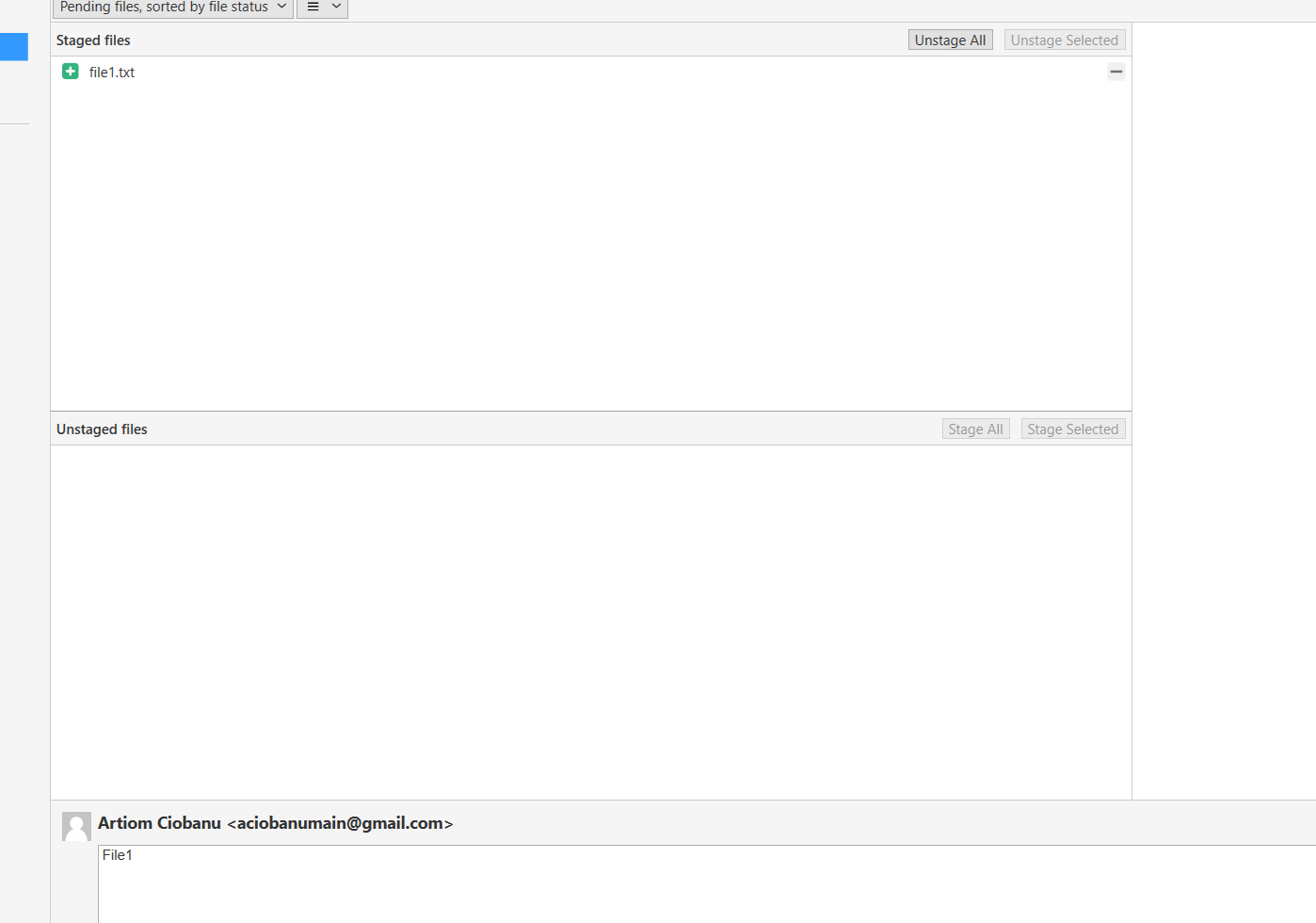
Создадим новый репозиторий:



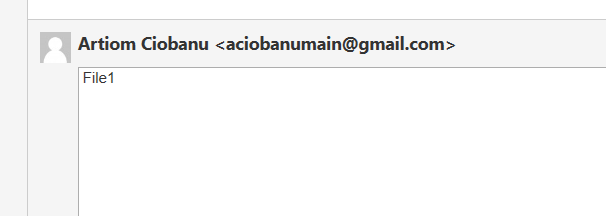
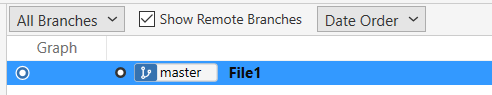
Создадим файл для начального коммита:



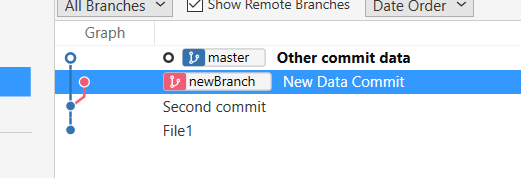
Добавим файл к коммиту:



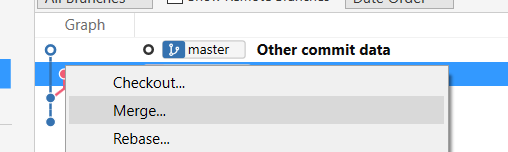
Сообщение для коммита и сам готовый коммит в ветке:

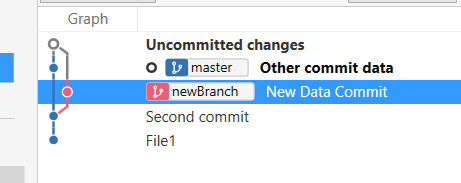
 

Создадим ещё несколько новых коммитов и новую ветку:

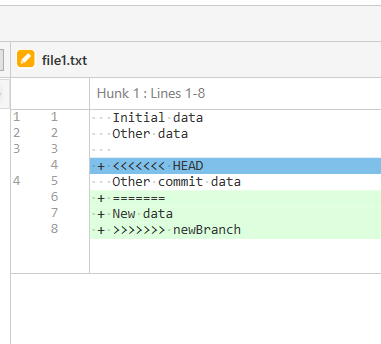


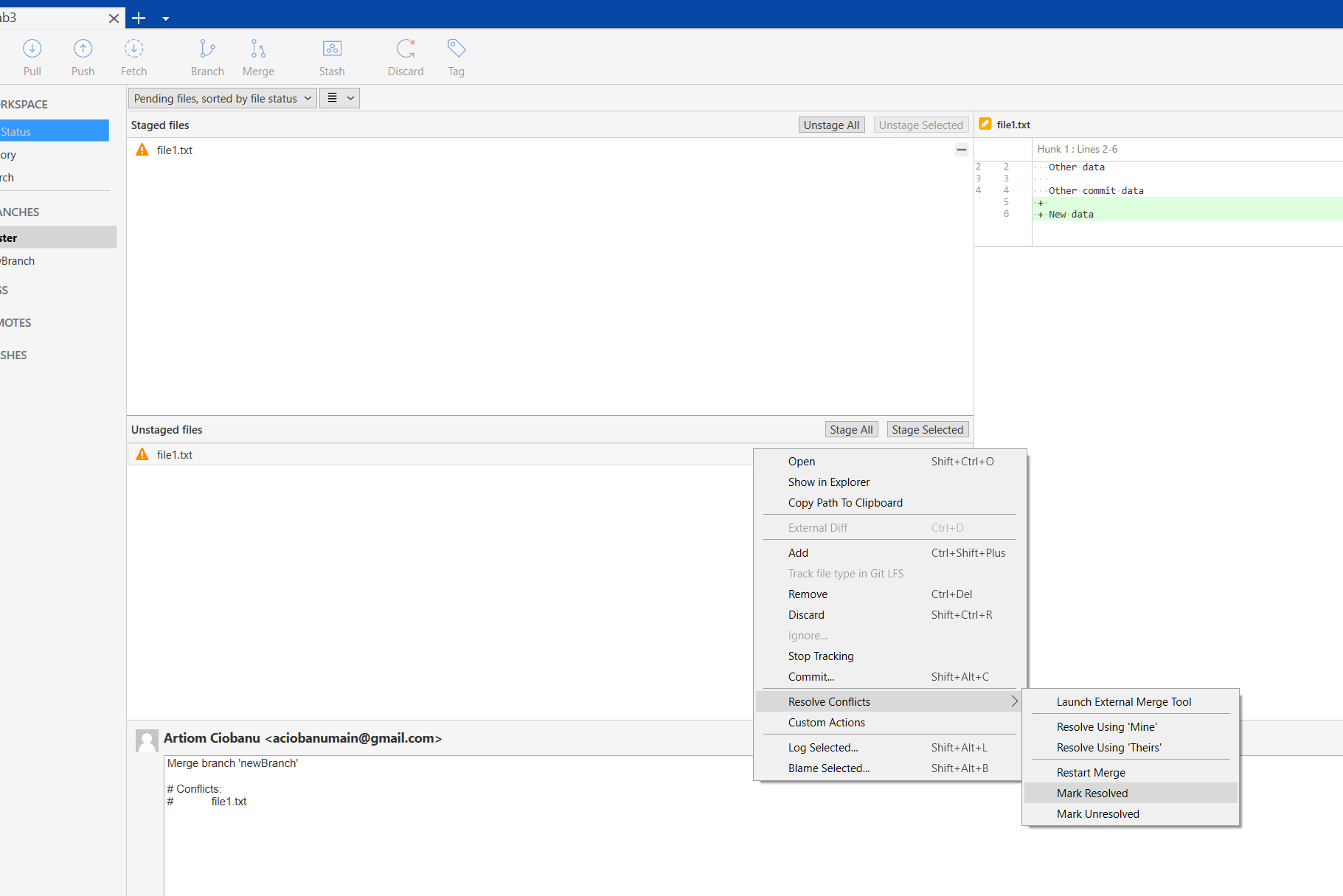
Выполним слияние веток:



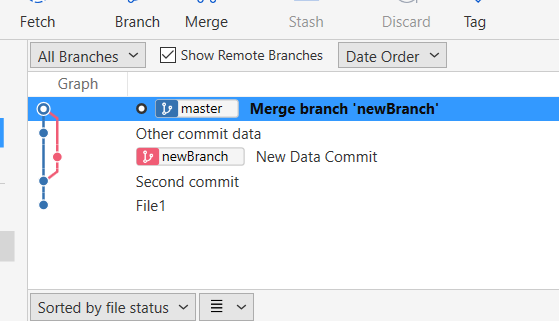


Необходимо разрешить конфликты:

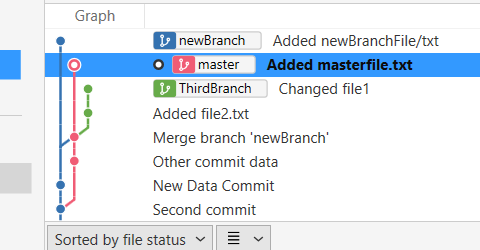




Конфликт разрешён:



Продолжим создавать новые ветки:



Создание публичного репозитория GitHub и веток слежения:

* git remote add
* git push -u origin master

Слияние / перебазирование веток git:

• git merge –squash

• git rebase

Разрешение конфликтов.

Перезапись комита:

• git –amend

Копирование изменений коммита:

• git cherry-pick

Отмена (реверсирование) изменений коммита:

• git revert

Отмена локальных изменений статуса репозитория Git:

• git reset (--soft, --mixed, --hard)

# Вывод

Во время работы над данной лабораторной работой я работал с системой контроля версий Git используя консоль. Я использовал команды commit, push, pull, fetch, checkout и другие.

# Библиография

<https://docs.github.com/en/github/importing-your-projects-to-github/importing-source-code-to-github/adding-an-existing-project-to-github-using-the-command-line>