

# лабораторная работа №2

Студент: **Алших маслем Ахмад**

Группа: **НФИБД-02-20**

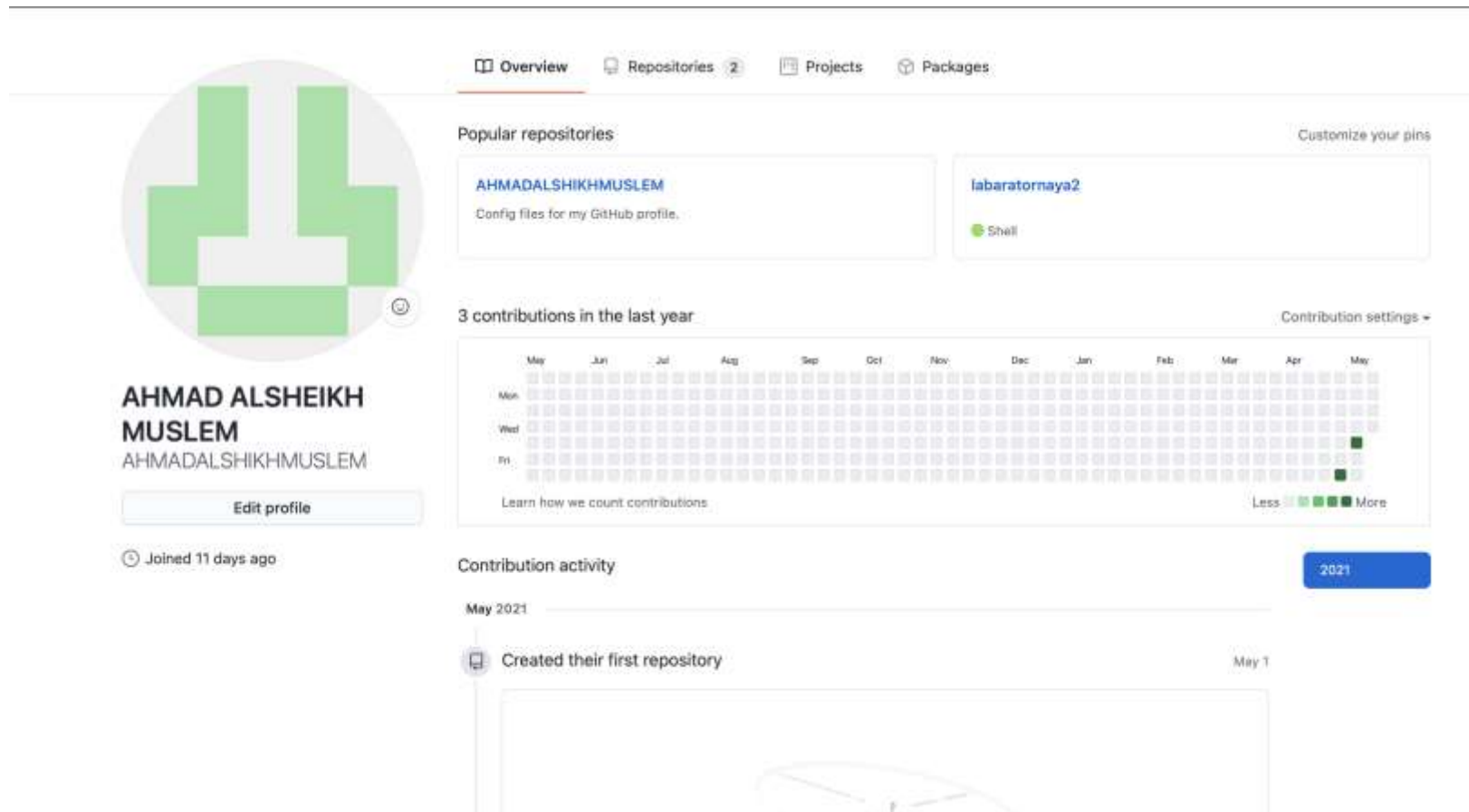
# Цель:

Изучить идеологию и применение средств контроля версий.

## Ход работы:

1) Создал учетную запись на github.

---



The screenshot shows the GitHub profile of a user named AHMAD ALSHEIKH MUSLEM. The profile includes a green and white geometric avatar, the username AHMAD ALSHEIKH MUSLEM, and the GitHub handle AHMADALSHIKHMUSLEM. A button labeled "Edit profile" is visible. The user joined 11 days ago. The "Overview" tab is selected, showing popular repositories: AHMADALSHIKHMUSLEM (with a description "Config files for my GitHub profile.") and labaratornaya2 (with a description "Shell"). Below this, a "3 contributions in the last year" section shows a calendar grid with three green squares indicating contributions in May. A "Contribution activity" section shows a timeline for May 2021, with a note "Created their first repository" on May 1st.

Overview Repositories 2 Projects Packages

Popular repositories Customize your pins

AHMADALSHIKHMUSLEM  
Config files for my GitHub profile.

labaratornaya2  
Shell

3 contributions in the last year Contribution settings

May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May

Mon  
Tue  
Wed  
Thu

Learn how we count contributions Less More

Contribution activity 2021

May 2021

Created their first repository May 1

2) Обозначил рабочий каталог как **test**,  
создав ее командой **mkdir**. После перешел в  
данный каталог командой **cd**.

Инициализировал систему **git** командой **git  
init**.

Создаю заготовку для файла **README.md**:

**echo "# lab02" >> README.md**

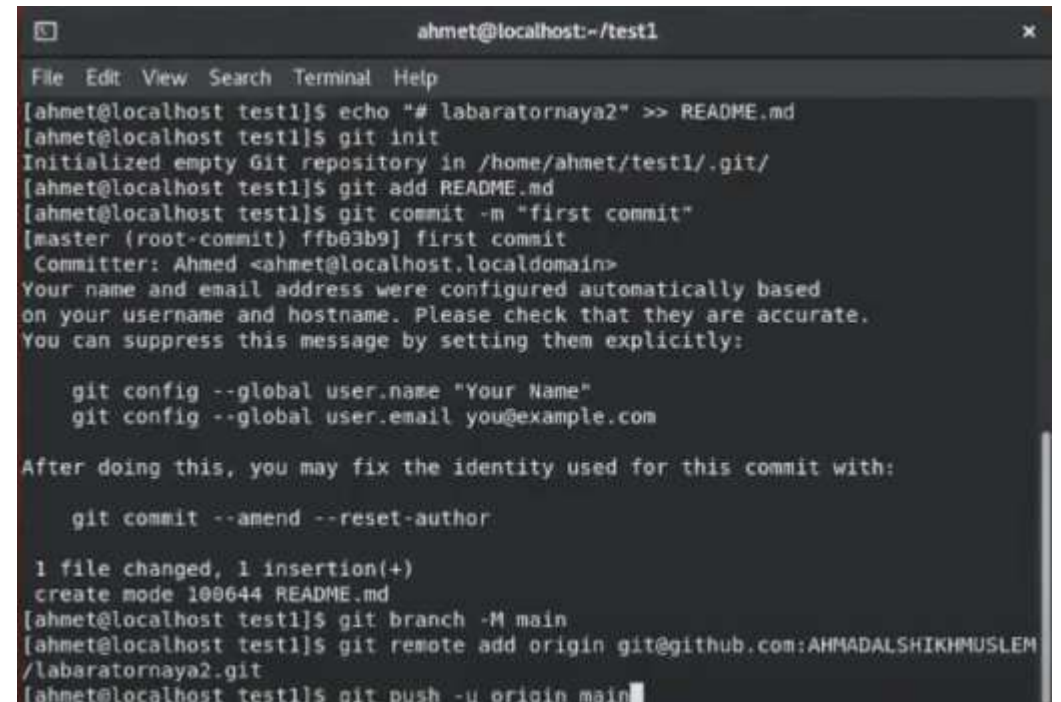
**git add README.md**

– Делаю первый коммит и выкладываем на github:

**git commit -m "first commit"**

**git remote add origin git@github.com:<username>/sciproc-  
intro.git**

**git push -u origin master**



```
ahmet@localhost:~/test1
File Edit View Search Terminal Help
[ahmet@localhost test1]$ echo "# labaratornaya2" >> README.md
[ahmet@localhost test1]$ git init
Initialized empty Git repository in /home/ahmet/test1/.git/
[ahmet@localhost test1]$ git add README.md
[ahmet@localhost test1]$ git commit -m "first commit"
[master (root-commit) ffb03b9] first commit
Committer: Ahmed <ahmet@localhost.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 README.md
[ahmet@localhost test1]$ git branch -M main
[ahmet@localhost test1]$ git remote add origin git@github.com:AHMADALSHIKHMUSLEM
/labaratornaya2.git
[ahmet@localhost test1]$ git push -u origin main
```

3) Для Добавил файл лицензий и начало просмотрим список имеющихся шаблонов

```
ahmet@localhost:~/test1
File Edit View Search Terminal Help
[ahmet@localhost test1]$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt
--2021-05-05 14:32:49-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Resolving creativecommons.org (creativecommons.org)... 104.20.150.16, 104.20.151.16, 172.67.34.140, ...
Connecting to creativecommons.org (creativecommons.org)|104.20.150.16|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: 'legalcode.txt'

legalcode.txt      [ <=> ] 18.22K --.-KB/s  in 0s

2021-05-05 14:32:50 (62.6 MB/s) - 'legalcode.txt' saved [18657]

[ahmet@localhost test1]$ curl -L -s https://www.gitignore.io/api/list
```

## 4)Теперь скачиваем шаблон для С

```
ahmet@localhost:~/test1
File Edit View Search Terminal Help
[ahmet@localhost test1]$ curl -L -s https://www.gitignore.io/api/c >> .gitignore
[ahmet@localhost test1]$ ls
legalcode.txt  README.md
[ahmet@localhost test1]$
```

## 6)Отправим на github

## 5)Добавил новые файлы и выполнил КОММИТ

```
ahmet@localhost:~/test1
File Edit View Search Terminal Help
[ahmet@localhost test1]$ curl -L -s https://www.gitignore.io/api/c >> .gitignore
[ahmet@localhost test1]$ ls
legalcode.txt  README.md
[ahmet@localhost test1]$ git add .
[ahmet@localhost test1]$ git commit -m "push"
[main c4064b3] push
Committer: Ahmed <ahmet@localhost.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 455 insertions(+)
create mode 100644 .gitignore
create mode 100644 legalcode.txt
[ahmet@localhost test1]$ clear
```

```
ahmet@localhost:~/test1
File Edit View Search Terminal Help
[ahmet@localhost test1]$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 6.43 KiB | 6.43 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:AHMADALSHIKHMUSLEM/labaratornaya2.git
   ffb03b9..c4064b3  main -> main
[ahmet@localhost test1]$
```

7) Инициализировал git-flow, установив префикс для ярлыков в v.

```
[ahmet@localhost test1]$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
```

8) Создаю релиз с версией 1.0.0

```
[ahmet@localhost test1]$ git flow release start 1.0.0
Switched to a new branch 'release/1.0.0'

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.0.0'
```

9) Записал версию Добавил в индекс

```
ahmet@localhost:~/test1
File Edit View Search Terminal Tabs Help

ahmet@localhost:~/test1 x ahmet@localhost:/home/ahmet x

[ahmet@localhost test1]$ echo "1.0.0" >> VERSION
[ahmet@localhost test1]$ git add .
[ahmet@localhost test1]$ git commit -m "first version"
[release/1.0.0 89b3ff5] first version
Committer: Ahmed <ahmet@localhost.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

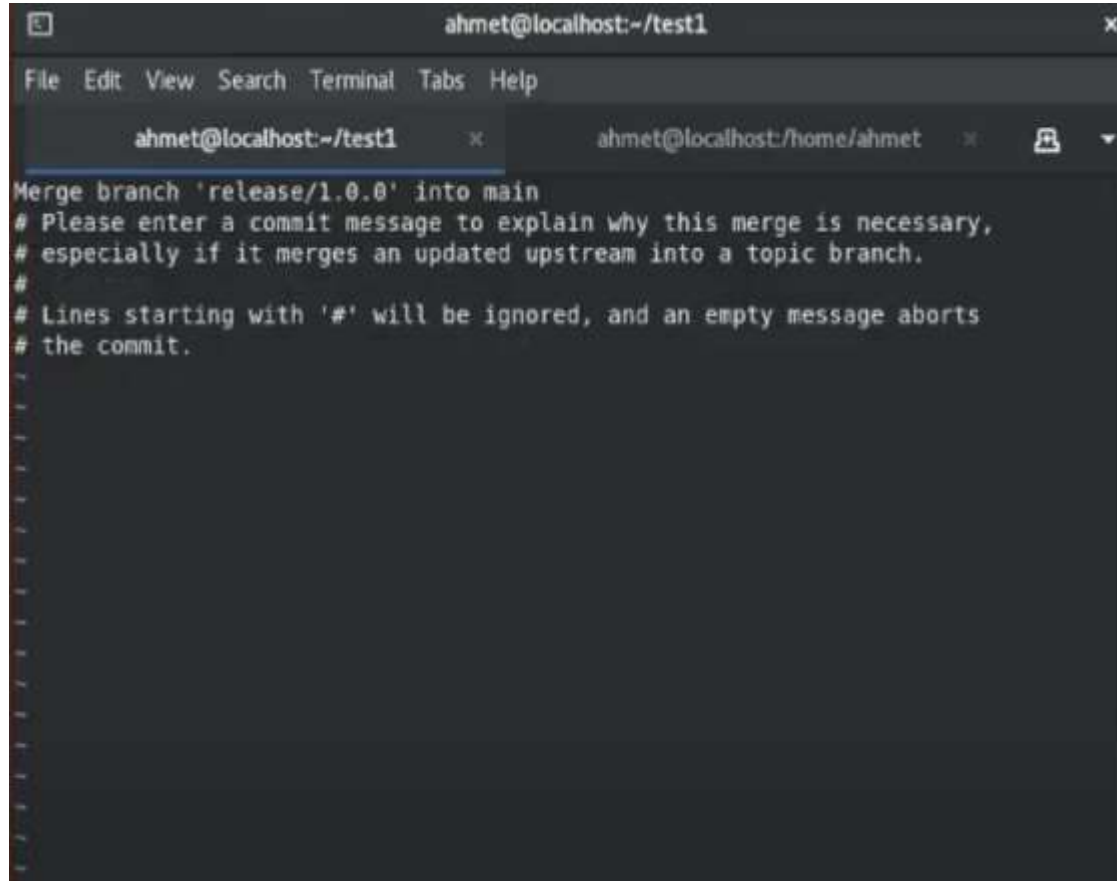
    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 79 insertions(+)
create mode 100644 VERSION
create mode 100755 gitflow-installer.sh
[ahmet@localhost test1]$
```

## 10) Залил релизную ветку в основную ветку



A terminal window titled 'ahmet@localhost:~/test1' with a menu bar (File, Edit, View, Search, Terminal, Tabs, Help). It shows a merge operation: 'Merge branch 'release/1.0.0' into main'. Below this, it prompts for a commit message: '# Please enter a commit message to explain why this merge is necessary, # especially if it merges an updated upstream into a topic branch. # Lines starting with '#' will be ignored, and an empty message aborts the commit.' The terminal is currently empty for input.

## 11) Отправил данные на github

```
[ahmet@localhost test1]$ git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.30 KiB | 1.30 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To github.com:AHMADALSHIKHMUSLEM/labaratornaya2.git
   c4064b3..b808c01  main -> main
   * [new branch]      develop -> develop
   * [new branch]      release/1.0.0 -> release/1.0.0
[ahmet@localhost test1]$ git push --tags
Everything up-to-date
[ahmet@localhost test1]$
```

## Вывод:

Изучил и понял как работать с системой контроля версий, с помощью командной строки, а именно с Git. Разобрал команды