

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и
естественных наук Кафедра прикладной
информатики и теории вероятностей

Лабораторная работа № 2

дисциплина: Архитектура компьютеров и операционные системы.
Раздел "Архитектура компьютеров"

Студент: Михедько Наталья

Группа: НПИбд-03-25

№ ст. билета: 1032253858

МОСКВА
2025 г.

Цель работы:

Научиться работать с системой контроля версий(GIT)

Порядок выполнения работы:

- Настройки git

Настроим git для правильной работы, после выведем список всех настроек, чтобы убедиться в том, что мы настроили все правильно.

- Добавим имя пользователя и почту, чтобы в коммитах было видно, кто именно сделал изменения. Без этого git не сможет нас идентифицировать и дальнейшая работа будет невозможна.

```
natasha-mihedko@natasha-mihedko-VirtualBox:~$ git config --global user.name "Mihedko"
natasha-mihedko@natasha-mihedko-VirtualBox:~$ git config --global user.email "1032253858@pfur.ru"
natasha-mihedko@natasha-mihedko-VirtualBox:~$
```

(рис. 1.1 настройка имени и почты в git)

Далее настраиваем правильный вывод данных. Эта настройка поможет нам избежать будущих проблем с чтением файлов, в которых находятся символы, которых

```
natasha-mihedko@natasha-mihedko-VirtualBox:~$ git config --global core.quotePath false
natasha-mihedko@natasha-mihedko-VirtualBox:~$
```

нет в ASCII таблице, например, русских букв.

(рис. 1.2 настройка для правильного отображения символов)

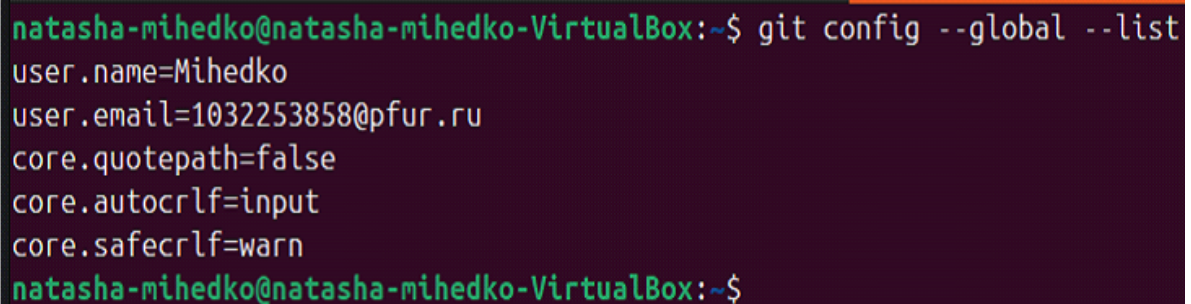
После этого называем ветку нашего первоначального репозитория, в которую будут коммитить все наши изменения. Она автоматически создается после того как мы введем команду git init.

```
natasha-mihedko@natasha-mihedko-VirtualBox:~$ git config --global core.autocrlf input
natasha-mihedko@natasha-mihedko-VirtualBox:~$ git config --global core.safecrlf warn
natasha-mihedko@natasha-mihedko-VirtualBox:~$
```

(рис. 1.4 настройка для предупреждения, если в файлах будут различаться переносы строк)

Далее настраиваем правильное форматирование перехода на следующую строку, так как на разных ОС реализация переноса сделана по-разному. Например, в Windows используется - CRLF(использует два символа: /r/n), а для Linux/macOS - LF(использует один символ: /n). Данная настройка заставит git предупреждать вас, если обнаружится несоответствие, которое может вызвать проблемы при совместной работе в разных ОС.

- Выведем все наши настройки, которые мы сделали.



```
natasha-mihedko@natasha-mihedko-VirtualBox:~$ git config --global --list
user.name=Mihedko
user.email=1032253858@pfur.ru
core.quotepath=false
core.autocrlf=input
core.safecrlf=warn
natasha-mihedko@natasha-mihedko-VirtualBox:~$
```

(рис. 1.5 вывод все настроек, которые мы произвели)

- Создание ssh ключа
Создадим SSH ключ и выведем файлы, потому что их у нас будет два. Открытый и Закрытый ключи. Именно открытый ключ мы позже используем, чтобы работать с github по SSH, чтобы убедиться в правильности наших действий.
- Введем команду для создания SSH ключа и добавим комментарий, в котором напомним имя, фамилию и почту.
-

После ввода этой команды, терминал предложит нам выбрать, куда сохранить наш ключ, нажмем Enter, чтобы он сохранил его в директорию ~/.ssh. После этого вводим

```
Enter file in which to save the key (/home/natasha-mihedko/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/natasha-mihedko/.ssh/id_ed25519
Your public key has been saved in /home/natasha-mihedko/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:0r3+plIXXQkuv4D2rP765FT0YkdGTxrnMyJlaoyhM9s Mihedko 1032253858@pfur.ru
The key's randomart image is:
+--[ED25519 256]--+
|      . . . . |
|      . +.. |
|      . . B +. |
|      ...= 0 o.. |
|      .+S+.* B o. |
|      .* =.X o o |
|      . E.O =   |
|      ..* +     |
|      .=*B.    |
+-----[SHA256]-----+
natasha-mihedko@natasha-mihedko-VirtualBox: ~$
```

пароль два раза, чтобы защитить наш ключ. И наши ключи созданы!

(рис. 2.2 задаем пароль и расположение файла с ключами)

-

Выведем все наши файлы, чтобы убедиться в правильности наших действий.

```
natasha-mihedko@natasha-mihedko-VirtualBox: ~$ cd ~/.ssh
natasha-mihedko@natasha-mihedko-VirtualBox: ~/.ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
natasha-mihedko@natasha-mihedko-VirtualBox: ~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIL2wq+1yIGnhs4iPN+uQkFdIN8WPk5KNfwBW7r9U0JGM
Mihedko 1032253858@pfur.ru
natasha-mihedko@natasha-mihedko-VirtualBox: ~/.ssh$
```

(рис. 2.3 проверка на наличие созданных ключей)

Я перешел в папку, куда были сохранены мои ключи. Далее вывел содержимое публичного ключа. Все наши комментарии были добавлены успешно, так что можем идти дальше. Хочу отметить, что открытым считается ключ с расширением (.pub). Именно его мы и будем использовать для работы с git. А ключ, который не имеет данного расширения - приватный. Его нельзя никому показывать.

- Добавление SSH ключа в наши настройки github, чтобы работать с репозиторием.

Перейдем на наш github, зайдём в настройки, назовем наш ключ - Title, добавим его и сохраним.

Переходим на наш github

Заходим в настройки и нажимаем на кнопку SSH and GPG keys.

-




Mihedko (Mihedko)

Your personal account

 Public profile


 Account

 Appearance

 **Accessibility**

 Notifications


Access

 Billing and licensing 

 Emails

 Password and authentication

 Sessions

 SSH and GPG keys

 Organizations

 Enterprises

 Moderation 

Code, planning, and automation

 Repositories

 Codemirror

(в настройках профиля ищем кнопку для добавления SSH-ключа)

-
- Заранее переходим в терминал и пишем команду, чтобы скопировать наш публичный SSH ключ и вставить его в поле для ключа

```
natasha-mihedko@natasha-mihedko-VirtualBox: ~/.ssh$ cat ~/.ssh/id_ed25519.pub | xclip  
clip
```


(рис. 3.5 команда для копирования публичного SSH-ключа)

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

**Mihedko**
SHA256: 0r3+p1DXXQkuv4D2+P765FT0Ykd6TXmNyJ1aoyhM9s
Added on Oct 14, 2025
Never used — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or [troubleshoot common SSH problems](#).

(рис. 3.6 нажав, как показано на рис. 3.2 нажимаем на кнопку new SSH key)

- Далее вводим название Title и вставляем наш ключ и нажимаем на кнопку ADD SSH KEY. И еще одно важное действие позади!

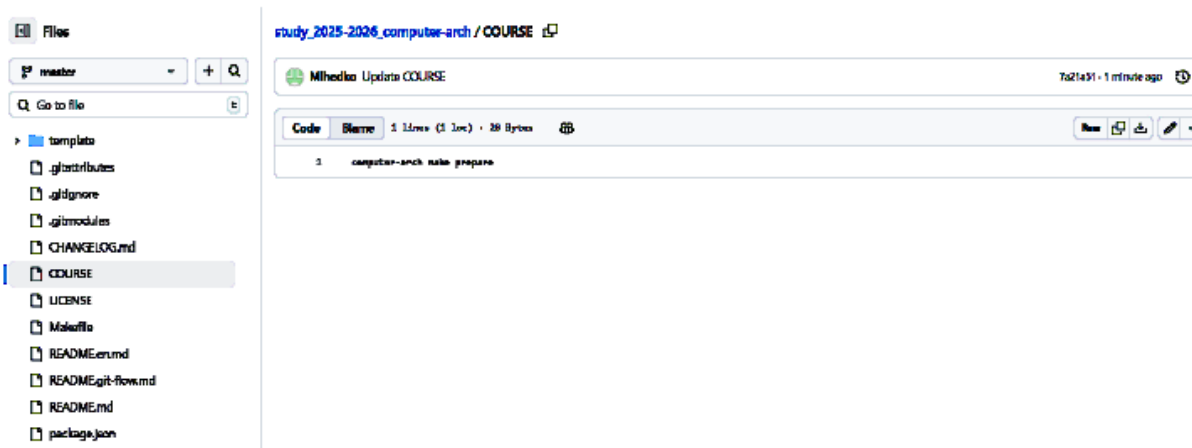
- Создание репозитория

Перейдем по ссылке

<https://github.com/yamadharma/course-directory-student-template#>, создадим репозиторий, дав ему название, которое от нас просят и сделай коммиты.

-

- Переходим по ссылке, ищем кнопку USE THIS TEMPLATE, нажимаем на нее. Далее нажимаем на кнопку CREATE A NEW REPOSITORY.



(рис. 5.1 создаем репозиторий)

- В появившемся окне ищем поле, куда мы введем название нашего репозитория и пишем то, что показано на фото и создаем наш репозиторий. Запомним это имя, ведь именно туда мы и будем добавлять наши файлы с лабораторными, которые мы сделаем чуть позже.

- клонирование репозитория в нашу папку

После всего этого мы переходим в терминал, переходим в каталог ~/work/study/2025-2026/Архитектура компьютера/

Ссылку на репозиторий, который мы будем клонировать можно найти на github, перейдя по самой первой ссылке [#">https://github.com/yamadharma/course-directory-student-template #](https://github.com/yamadharma/course-directory-student-template). Копируем именно SSH.

- Возвращаемся в терминал и пишем команду для клонирования репозитория в папку Архитектура компьютеров, которую мы создали /. После этого выведем содержимое папки и убедимся, что все работало.

```
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026$ cd computerarch
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch$ ls
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch$ git clone --recursive git@github.com:Mihedko/study_2025-2026_computer-arch
```

(рис. 6.2 клонирование репозитория в папку)

```
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch$ ls
study_2025-2026_computer-arch
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch$ cd
study_2025-2026_computer-arch/
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/study_2025-2026_computer-arch$ ls
CHANGELOG.md  LICENSE  package.json  README.git-flow.md  template
COURSE        Makefile  README.en.md  README.md
```

(рис. 6.3 убеждаемся в том, что репозиторий клонирован)

- Использование главных команд add, commit, push
- Найдем файл COURSE, убедимся в том, что он пустой.
После этого запишем в него COURSE make prepare и отправим изменение на github.
- Поиск в скопированном репозитории файла COURSE.

```
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/study_2025-2026_computer-arch$ ls
CHANGELOG.md  LICENSE  package.json  README.git-flow.md  template
COURSE        Makefile  README.en.md  README.md
```

(рис. 7.1 поиск файла COURSE)

Записываем в файл COURSE строчку COURSE make prepare.

```
dy_2025-2026_computer-arch$ cat COURSE
computer-arch make prepare
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/study_2025-2026_computer-arch$
```

(рис. 7.3 запись строки в файл)

- Добавляем все изменения с помощью команды (git add)
-

.), сохраняем все изменения с помощью команды (git commit), отправляем все сохраненные изменения на github с помощью команды (git push)

```
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/study_2025-2026_computer-arch$ git add .
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/study_2025-2026_computer-arch$ git commit -am 'feat(main)make course structure'
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/study_2025-2026_computer-arch$ git push
Everything up-to-date
```

(рис. 7.4 добавление изменений, сохранение изменений и отправка на github)

Самостоятельной работа

1) Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs/lab02/report).

```
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/arch-pc/labs/labs02/report$ touch л02_МИХЕДЬКО_ОТЧЕТ.pdf
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/arch-pc/labs/labs02/report$ ls л02_МИХЕДЬКО_ОТЧЕТ.pdf
л02_МИХЕДЬКО_ОТЧЕТ.pdf
```


2) Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.

```
natasha-mihedko@natasha-mihedko-VirtualBox:~$ cd Downloads
natasha-mihedko@natasha-mihedko-VirtualBox:~/Downloads$ cp "Л01_Михедько_отчет.pdf" ~
/work/study/2025-2026/computerarch/arch-pc/labs/labs02/report
natasha-mihedko@natasha-mihedko-VirtualBox:~/Downloads$ cd
natasha-mihedko@natasha-mihedko-VirtualBox:~$ cd ~/work/study/2025-2026/computerarch/a
rch-pc/labs/labs02/report
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/arch-pc
/labs/labs02/report$
```

Рисунок. 2 Копируем первый отчет в соответствующий каталог

3) Загрузите файлы на github.

```
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/arch-pc
/labs/labs02/report$ git add .
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/arch-pc
/labs/labs02/report$ git commit -am 'feat(main)make course structure'
[master 0d81d93] feat(main)make course structure
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 arch-pc/labs/labs02/report/Л01_Михедько_отчет.pdf
 create mode 100644 arch-pc/labs/labs02/report/Л02_МИХЕДЬКО_отчет.pdf
natasha-mihedko@natasha-mihedko-VirtualBox:~/work/study/2025-2026/computerarch/arch-pc
/labs/labs02/report$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 2.17 MiB | 2.93 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Mihedko/study_2025-2026_computer-arch
 7a21a51..0d81d93 master -> master
```

Рисунок. 3 Загружаем все на github

Вывод: Я освоил систему контроля версий Git, выполнив следующие шаги: настроил локальное окружение, создал удалённый репозиторий на GitHub и успешно загрузил в него отчёты по лабораторным работам. Этот процесс позволил мне эффективно управлять версиями кода и документов, обеспечивая их безопасность и доступность для совместной работы.