

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Отчет

По лабораторной работе №2.

дисциплина: архитектура компьютера

Студент: Назармамадов У.Дж.

Группа: НКАбд-04–23

МОСКВА

2023 г.

Содержание:

- 1. Цель работы.....**
- 2. Задание.....**
- 3. Теоретическое введение.....**
- 4. Выполнение лабораторной работы.....**
- 5. Выводы.....**
- 6. Источники.....**

1. Цель работы:

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

2. Задание:

1. Настройки GitHub.
2. Базовая настройка Git.
3. Создание SSH-ключа.
4. Создание рабочего пространства и репозитория курса на основе шаблона.
5. Создание репозитория курса на основе шаблона.
6. Настройка каталога курса.

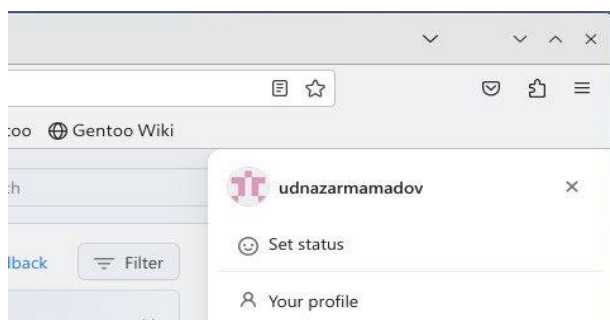
3. Теоретическое задание:

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

4. Выполнение лабораторной работы:

4.1 Настройка GitHub:

Создавать аккаунт в github мне не нужно, потому что он у меня уже есть. Поэтому в первую очередь я захожу в свой аккаунт.



4.2 Базовая настройка Git:

Открываю терминал и ввожу туда определенные команды для предварительной конфигурации git, где я указываем свое имя и адрес электронной почты:

```
udnazarmamadov@dk8n61 ~ $ git config --global user.name "<Umed Nazarmamadov>"
udnazarmamadov@dk8n61 ~ $ git config --global user.email "<umednazarmamadov40@gmail.com>"
udnazarmamadov@dk8n61 ~ $
```

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов:

```
udnazarmamadov@dk8n61 ~ $ git config --global core.quotePath false
udnazarmamadov@dk8n61 ~ $
```

Задаю имя для начальной ветки:

```
udnazarmamadov@dk8n61 ~ $ git config --global init.defaultBranch master
udnazarmamadov@dk8n61 ~ $
```

Задаю параметр autocrlf:

```
udnazarmamadov@localhost:~$ git config --global core.autocrlf
udnazarmamadov@localhost:~$
```

Задаю параметр safecrlf:

```
udnazarmamadov@localhost:~$ git config --global core.safecrlf
udnazarmamadov@localhost:~$
```

4.3 Создание SSH-ключа:

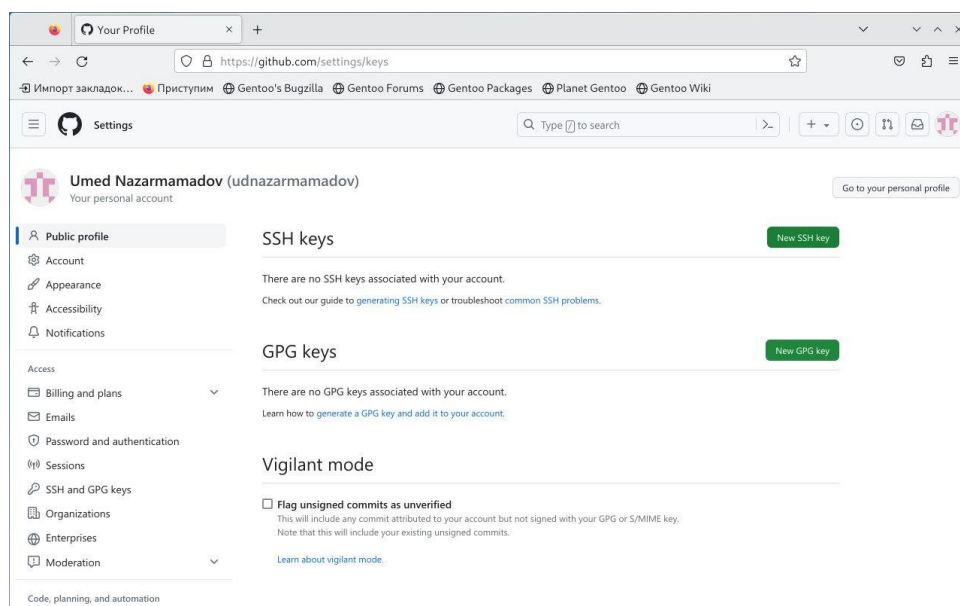
Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Ввожу команду `ssh-keygen -C "Имя_Фамилия, work@email"`. Ключи сохраняются в каталоге `~/.ssh/`.

```
udnazarmamadov@dk8n61 ~ $ ssh-keygen -C "Umed Nazarmamadov <umednazarmamadov40@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/u/d/udnazarmamadov/.ssh/id_rsa):
Created directory '/afs/.dk.sci.pfu.edu.ru/home/u/d/udnazarmamadov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/u/d/udnazarmamadov/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/u/d/udnazarmamadov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:alAyLR5Wodfr6Pjt6fBb+hzwlUPrPIzBW/KkP42bZTg Umed Nazarmamadov <umednazarmamadov40@gmail.com>
The key's randomart image is:
+----[RSA 3072]-----+
|      o.      |
|      + .     |
|    B + . .   |
|   o B . . o . |
|   o S = *    |
| . + o . & o   |
|    = . + + Eoo|
| + + = . .o*. |
| ..ooOooo +o  |
+-----[SHA256]-----+
```

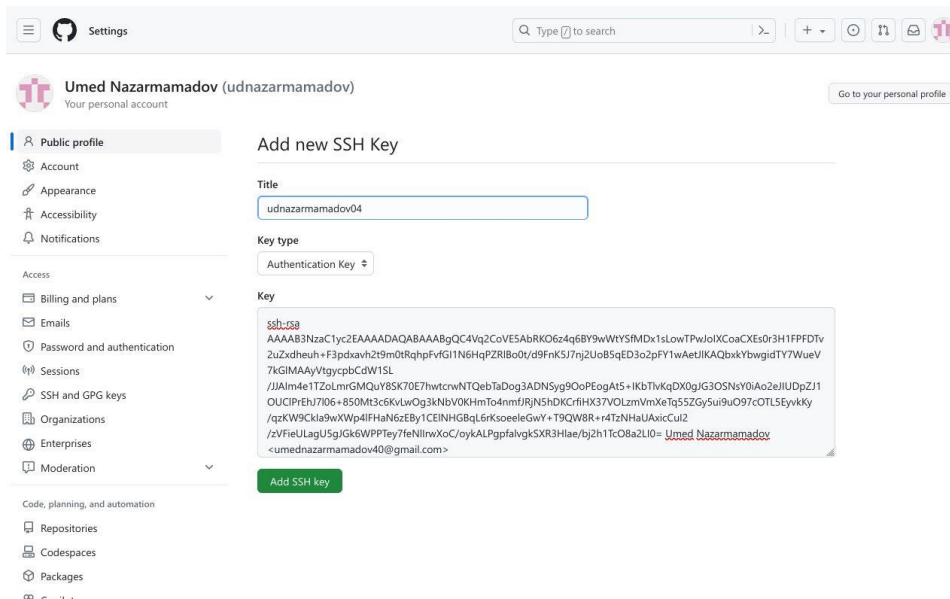
Копирую открытый ключ из директории, с помощью утилиты `xclip`:

```
udnazarmamadov@localhost:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Открываю github. В своем профиле выбираю страницу “SSH and GPG keys”. Нажимаю кнопку “New SSH keys”:



Вставляю в поле “Key” скопированный ключ. Указываю имя ключа и добавляю этот ключ нажимая на кнопку “Add SSH-key”.



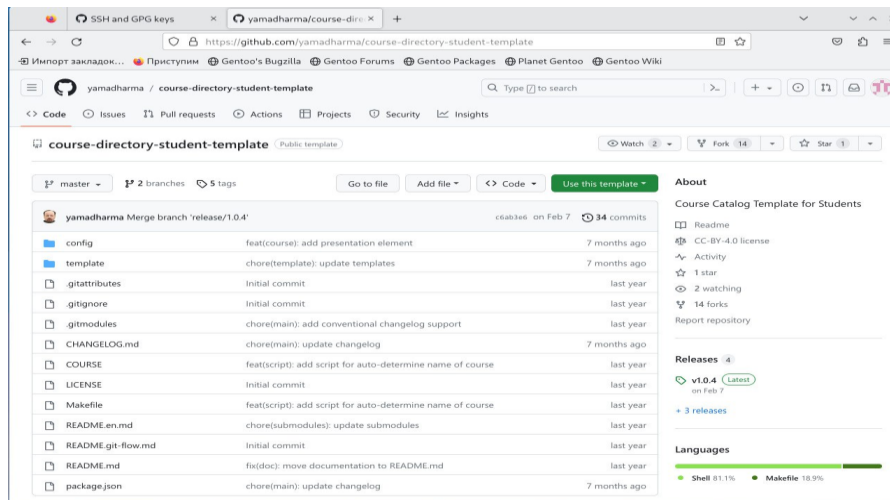
4.4 Создание рабочего пространство и репозитории курса на основе шаблона:

Открываю терминал чтобы создать директорию, рабочее пространство, с помощью утилиты `mkdir`, благодаря ключу `-p` создаю все директории после домашней `~/work/study/2023-2024/"Архитектура компьютера"` рекурсивно. Проверяю работу команды с помощью утилиты `ls`:

```
udnazarmamadov@dk8n61 ~ $ mkdir -p work/study/2022-2023/"Архитектура компьютера"
udnazarmamadov@dk8n61 ~ $ ls
$ grell public public_html work Видео Документы Загрузки Изображения Музыка Общедоступные 'Рабочий стол' Шаблоны
```

4.5 Создание репозитория курса на основе шаблона:

Перехожу на страницу репозитория в браузере с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю “Use this template”:



В открывшемся окне задаю имя репозитория study_2023–2024_arhpc и нажимаю на кнопку создать:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * udnazarmamadov / **Repository name *** study_2023-2024_arh-pc
 study_2023-2024_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about [literate-couscous](#) ?

Description (optional)

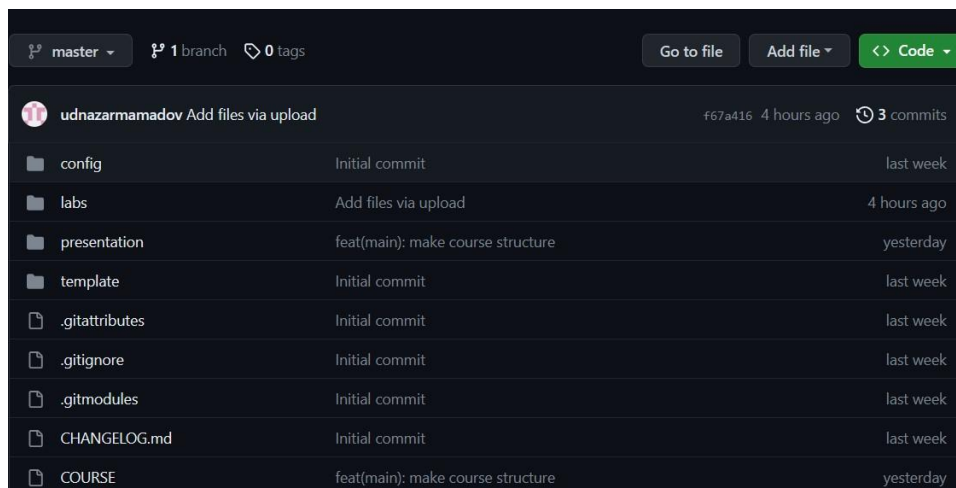
☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

You are creating a public repository in your personal account.

[Create repository](#)

Проверяем наше действие:



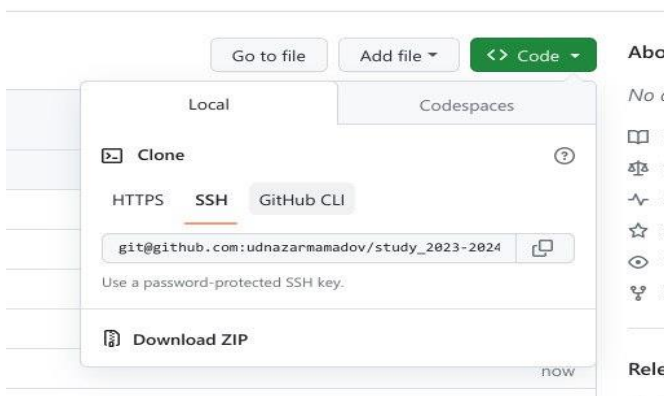
Открываю терминал и перехожу в созданный каталог с помощью утилиты cd:

```
udnazarmamadov@dk8n61 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"
udnazarmamadov@dk8n61 ~/work/study/2023-2024/Архитектура компьютера $
```

Копирую созданный репозиторий с помощью команды git clone --recursive git@github.com:/study_2023-2024_arh-pc.git arch-pc:

```
udnazarmamadov@dk8n61 ~/work/study/2023-2024/Архитектура компьютера $ git clone --recursive git@github.com:udnazarmamadov/study_2023-2024_arh-pc.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA8zPMsvHdkr4Uvc0QJ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.94 КБ | 16.94 МБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/u/d/udnazarmamadov/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КБ | 1.16 МБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/u/d/udnazarmamadov/work/study/2023-2024/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
```

Копирую ссылку для клонирования на странице созданного репозитория:



4.6 Настройка каталога курса:

Перехожу в каталог arch-pc с помощью команды cd:

```
udnazarmamadov@dk2n22 ~/work/study/2023-2024/Архитектура компьютера $ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
udnazarmamadov@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $
```

Удаляю лишние файлы с помощью утилиты rm:

```
udnazarmamadov@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ rm package.json
udnazarmamadov@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $
```

Создаю необходимые каталоги:

```
udnazarmamadov@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ echo arch-pc > COURSE
udnazarmamadov@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ make
```

Отправляю файлы на сервер, добавляю все созданные каталоги с помощью git add, комментирую и сохраняю изменения на сервере как добавление курса с помощью git commit:

```
udnazarmamadov@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git commit -am 'feat(main): make course structure'
[master 96805a1] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
create mode 100644 labs/lab02/presentation/presentation.md
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/core.py
```

Отправляю все на сервер:

```
udnazarmamadov@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.14 КиБ | 2.78 МиБ/с, готово.
Всего 35 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:udnazarmamadov/study_2023-2024_arh-pc.git
   6a5fa20..96805a1  master -> master
```

Проверяю правильность работы на самом сайте Github:

udnazarmamadov feat(main): make course structure 96805a1 · 2 minutes ago 🕒 History		
Name	Last commit message	Last commit date
..		
lab01	feat(main): make course structure	2 minutes ago
lab02	feat(main): make course structure	2 minutes ago
lab03	feat(main): make course structure	2 minutes ago
lab04	feat(main): make course structure	2 minutes ago
lab05	feat(main): make course structure	2 minutes ago
lab06	feat(main): make course structure	2 minutes ago
lab07	feat(main): make course structure	2 minutes ago
lab08	feat(main): make course structure	2 minutes ago
lab09	feat(main): make course structure	2 minutes ago
lab10	feat(main): make course structure	2 minutes ago
lab11	feat(main): make course structure	2 minutes ago

5. Вывод:

При выполнении данной лабораторной работы я научился работать с github-ом с помощью консоли в Linux. Создать репозитории и каталогов в нем, не обязательно находится в самом сайте github, а можно сделать это прямо в консоли с помощью определенных команд.

6. Список литературы:

1. Архитектура ЭВМ.
2. Git – gitattributes Документация.