

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ЛАБОРАТОРНАЯ РАБОТА № 2

дисциплина: *Архитектура компьютера*

Студент: Мизинов М.Г.

Группа: НКАбд-04-25

№ ст. билета: 1032253540

МОСКВА

2025 г.

СОДЕРЖАНИЕ

Список иллюстраций.....	3
Список таблиц.....	4
Основная часть.....	5
1. Цель работы	5
2. Теоретическое введение	5
3. Задание	7
4. Выполнение лабораторной работы.....	7
4.1 Настройка github	7
4.2. Базовая настройка Git.....	7
4.3. Создание SSH-ключа.....	7
4.4. Создание рабочего пространства и репозитория курса на основе шаблона..	8
4.5. Создание репозитория курса на основе шаблона	9
4.6. Настройка каталога курса	10
5. Задание для самостоятельной работы	10
Выводы	11
Список литературы.....	12

Список иллюстраций

Рисунок 1 – git к консоли.....	7
Рисунок 2 – настройка параметров.....	7
Рисунок 3 – генерация ключа.....	8
Рисунок 4 – Добавление ключа	8
Рисунок 5 – создание каталога	8
Рисунок 6 – создание репозитория	9
Рисунок 7 – клонирование репозитория.....	9
Рисунок 8 – формирование каталогов.....	10
Рисунок 9 – git push	10
Рисунок 10 – загрузка лабораторных работ	10

Список таблиц

Таблица 3.1 Описание некоторых команд системы контроля версий Git.....	6
--	---

Основная часть

1. Цель работы

Целью работы является изучение идеологии и применения средств контроля версий, приобретение практических навыков по работе с системой контроля версий git.

2. Теоретическое введение

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 Демидова А. В., Велиева

Т. Р., Геворкян М. Н. 19 средствами файловой системы ОС, обеспечивая, таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. Наиболее часто используемые команды git представлены в таблице 3.1.

Таблица 3.1 Описание некоторых команд системы контроля версий Git

Команда	Описание
git init	создание основного дерева репозитория
git pull	получение обновлений (изменений) текущего дерева из центрального репозитория
git push	отправка всех произведённых изменений локального дерева в центральный репозиторий
git status	просмотр списка изменённых файлов в текущей директории
git diff	просмотр текущих изменений
git add	добавить все изменённые и/или созданные файлы и/или каталоги

3. Задание

На основе методических указаний провести работу с базовыми командами системы контроля версий git, выучить применение команд для разных случаев использования, настроить GitHub.

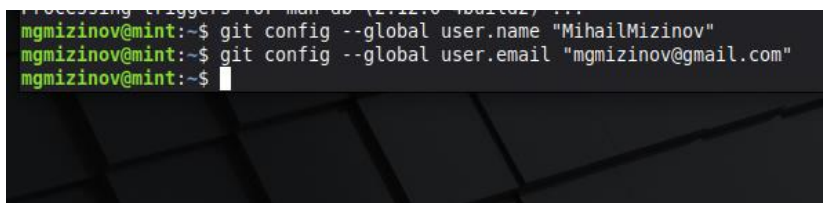
4. Выполнение лабораторной работы

4.1 Настройка github

Для выполнения лабораторных работ был использован Github.

4.2. Базовая настройка Git

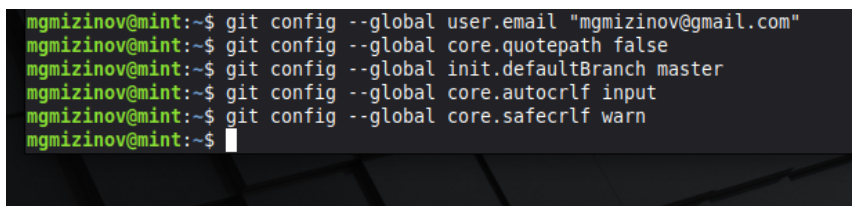
Сначала сделаем предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и e-mail владельца репозитория (рис. 1):



```
Processing triggers for man-db (2.11.2-1build1) ...  
mgmizinov@mint:~$ git config --global user.name "MihailMizinov"  
mgmizinov@mint:~$ git config --global user.email "mgmizinov@gmail.com"  
mgmizinov@mint:~$
```

Рисунок 1 – git к консоли

Настроим utf-8 в выводе сообщений git. Зададим имя начальной ветки (будем называть её master). Параметр autocrlf. Параметр safecrlf (рис. 2):



```
mgmizinov@mint:~$ git config --global user.email "mgmizinov@gmail.com"  
mgmizinov@mint:~$ git config --global core.quotepath false  
mgmizinov@mint:~$ git config --global init.defaultBranch master  
mgmizinov@mint:~$ git config --global core.autocrlf input  
mgmizinov@mint:~$ git config --global core.safecrlf warn  
mgmizinov@mint:~$
```

Рисунок 2 – настройка параметров

4.3. Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый) (рис. 3):

```
mgmizinov@mint:~$ ssh-keygen -C "Mihail Mizinov mgmizinov@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/mgmizinov/.ssh/id_ed25519):
Created directory '/home/mgmizinov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mgmizinov/.ssh/id_ed25519
Your public key has been saved in /home/mgmizinov/.ssh/id_ed25519.pub
The key fingerprint is:
Mihail Mizinov mgmizinov@gmail.com

The key's randomart image is:
+--[ED25519 256]--+
|      + . = * +      |
|      o = .ooo.      |
|      . + . . .      |
|      o o o o        |
|      S o = +        |
|      . . + BE|      |
|      o o . B +      |
|      ..O. =.=.+o|    |
|      ..o +o+o**.|    |
+-----[SHA256]-----+
mgmizinov@mint:~$
```

Рисунок 3 – генерация ключа

Далее необходимо загрузить сгенерированный открытый ключ (рис. 4).

Add new SSH Key

Title

Key type

Key

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHFa9kAXUtbCukBk

Mihail Mizinov mgmizinov@gmail.com

Рисунок 4 – Добавление ключа

4.4. Создание рабочего пространства и репозитория курса на основе шаблона

Откройте терминал и создайте каталог для предмета «Архитектура компьютера» (рис. 5):

```
mgmizinov@mint:~/.ssh$ mkdir -p ~/work/study/2025-2026/"Computer architecture"
mgmizinov@mint:~/.ssh$ cd
mgmizinov@mint:~$
```

Рисунок 5 – создание каталога

4.5. Создание репозитория курса на основе шаблона

Репозиторий на основе шаблона можно создать через web-интерфейс github (рис. 6):

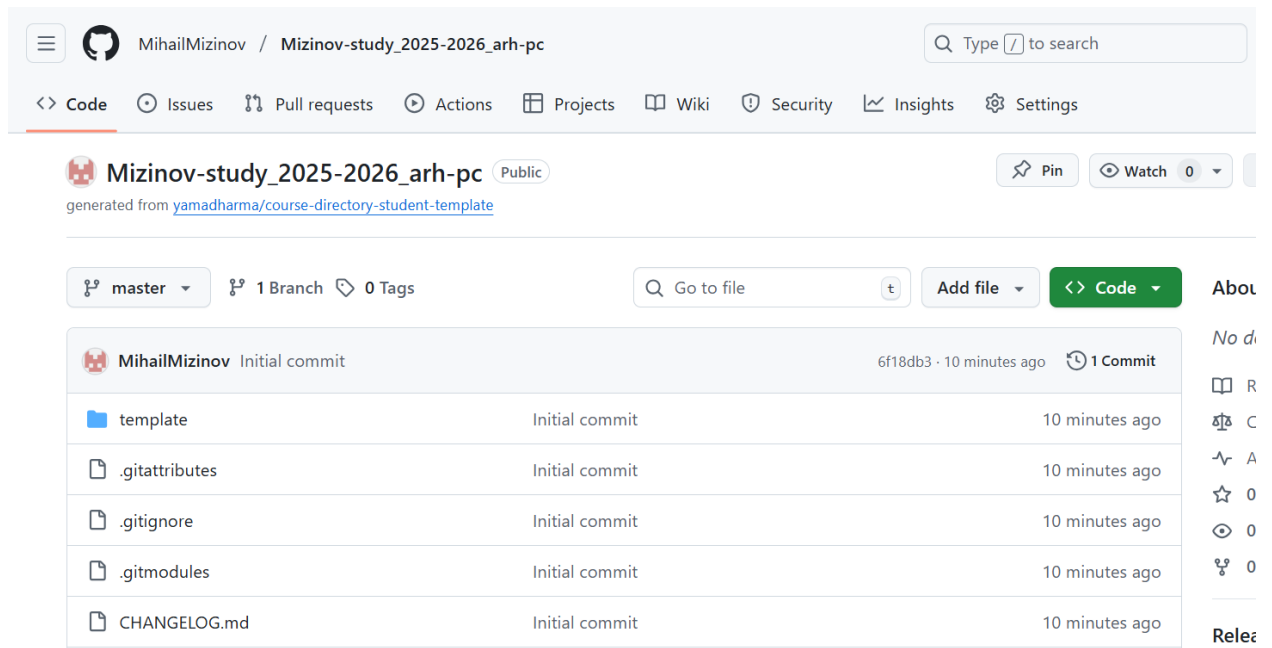


Рисунок 6 – создание репозитория

Клонируйте созданный репозиторий (рис. 7):

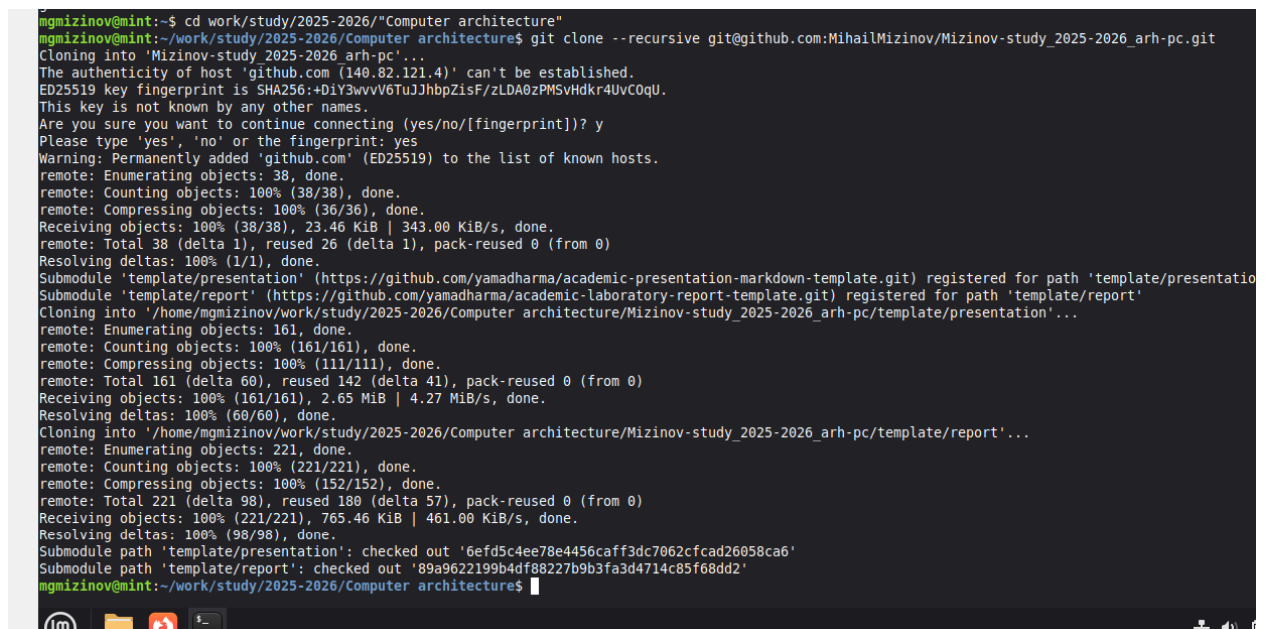


Рисунок 7 – клонирование репозитория

4.6. Настройка каталога курса

Создайте необходимые каталоги (рис. 8):

```
mgmizinov@mint:~$ cd work/study/2025-2026/"Computer architecture"/arh-pc
mgmizinov@mint:~/work/study/2025-2026/Computer architecture/arh-pc$ rm package.json
mgmizinov@mint:~/work/study/2025-2026/Computer architecture/arh-pc$ echo arch-pc > COURSE
```

Рисунок 8 – формирование каталогов

Отправьте файлы на сервер (рис. 9):

```
mgmizinov@mint:~/work/study/2025-2026/Computer architecture/study_2025-2026_arh-pc$ git add .
mgmizinov@mint:~/work/study/2025-2026/Computer architecture/study_2025-2026_arh-pc$ git commit -am 'feat(main): make course structure'
[master ca8ce73] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
mgmizinov@mint:~/work/study/2025-2026/Computer architecture/study_2025-2026_arh-pc$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 287 bytes | 287.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:MihailMizinov/Mizinov-study_2025-2026_arh-pc.git
 6f18db3..ca8ce73  master -> master
mgmizinov@mint:~/work/study/2025-2026/Computer architecture/study_2025-2026_arh-pc$
```

Рисунок 9 – git push

5. Задание для самостоятельной работы

Скопируйте отчеты по выполнению текущей и предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства в соответствующем каталоге рабочего (рис. 10):

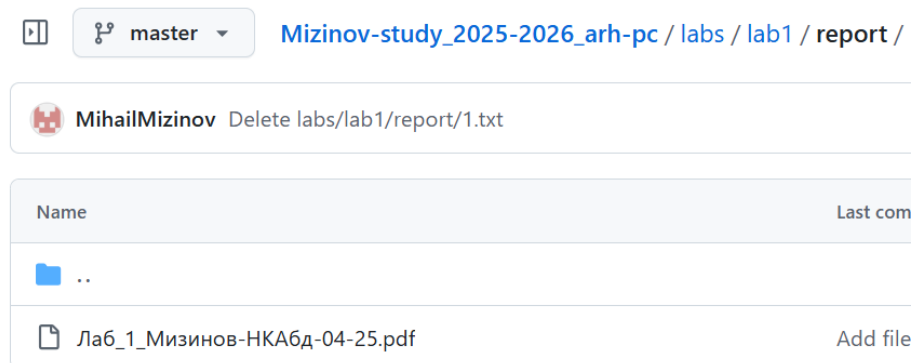


Рисунок 10 – загрузка лабораторных работ

Ссылка на github: https://github.com/MihailMizinov/Mizinov-study_2025-2026_arh-pc

Выводы

В результате выполнения лабораторной работы номер 2 были изучены основные принципы работы систем контроля версий, в частности, распределенной системы Git. На практике были приобретены навыки базовой настройки Git, создания SSH-ключей для безопасного взаимодействия с удаленным репозиторием и настройки рабочего пространства. Освоен процесс создания репозитория курса на основе шаблона на GitHub и его клонирования на локальную машину. Таким образом, работа заложила фундамент для дальнейшего использования Git при выполнении индивидуальных и групповых проектов.

Список литературы

- 1) Лабораторная работа №2.
https://esystem.rudn.ru/pluginfile.php/2945866/mod_resource/content/0/lab2.pdf
- 2) Википедия. <https://en.wikipedia.org/wiki/GitHub>