

**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: Скворцова Дарья

Группа: НКАбд-05-25

**МОСКВА**

2025 г.

## **Оглавление**

1 Цель работы .....	3
2 Задание .....	4
3 Теоретическое введение .....	5
4 Выполнение лабораторной работы .....	7
4.1 Техническое обеспечение .....	7
4.2 Базовая настройка Git .....	7
4.3 Создание рабочего пространства и репозитория курса на основе шаблона. ....	10
4.4 Создание репозитория курса на основе шаблона. ....	10
4.5 Настройка каталога курса. ....	13
5 Задания для самостоятельной работы. ....	14
6 Выводы. ....	15
Список литературы .....	16

# **1 Цель работы**

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

## **2 Задание**

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

### **3 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 средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

<code>git commit -am</code>	сохранить все добавленные изменения и все изменённые файлы 'Описание коммита'
<code>git checkout -b</code>	создание новой ветки, базирующейся на текущей имя_ветки
<code>git checkout</code>	переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)
<code>git push origin</code>	отправка изменений конкретной ветки в центральный репозиторий имя_ветки
<code>git merge</code>	слияние ветки с текущим деревом
<code>--no-ff</code>	
<code>имя_ветки</code>	
<code>git branch -d</code>	удаление локальной уже слитой с основным деревом ветки имя_ветки
<code>git branch -D</code>	принудительное удаление локальной ветки имя_ветки
<code>git push origin</code>	удаление ветки с центрального репозитория :имя_ветки

Таблица 3.1 Описание некоторых команд системы контроля версий Git.

## 4 Выполнение лабораторной работы

### 4.1 Техническое обеспечение

Лабораторная работа была выполнена на домашнем компьютере под управлением операционной системы Fedora Workstation 40.

### 4.2 Базовая настройка Git

Для начала я проведу предварительную конфигурацию Git, для этого открываю терминал и ввожу команды, настраиваю параметры utf-8, имя начальной ветки, autocrlf и safecrlf (рис. 4.2.1)

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~$ git config --global user.email "<1032251952@pfur.ru>"  
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~$ git config --global core.quotePath false  
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~$ git config --global init.defaultBranch master  
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~$ git config --global core.autocrlf input  
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~$ git config --global core.safecrlf warn
```

Рис. 4.2.1 Предварительная конфигурация Git и Настройка параметров Git.

Далее создаю пару ssh ключей для интеграции с платформой GitHub (рис. 4.2.2)

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~$ ssh-keygen -C "Скворцова Дарья <1032251952@pfur.ru">"
```

Рис. 4.2.2 Создание пары ssh ключей.

Далее я перехожу на сайт GitHub, авторизуюсь, перехожу в настройки аккаунта, вставляю публичный ключ в предназначенном для этого поле. (рис. 4.2.3)

The screenshot shows the 'SSH keys' section of a GitHub account settings page. At the top right is a green button labeled 'New SSH key'. Below it, a message reads: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' Underneath, a heading 'Authentication keys' is followed by a table listing one key:

My Work Laptop	SHA256:0021gQ1NCWmd494C9htuLbUJlokfv4agYp+xxKUnZA	Delete
<small>Added on Dec 09, 2005</small>		

(Рис. 4.2.3)

## **4.3 Создание рабочего пространства и репозитория курса на основе шаблона.**

Рабочее пространство при выполнении лабораторных работ должно придерживаться определённой структурной иерархии, для этого я создаю директорию на своем рабочем компьютере (рис. 4.3.1)

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера$ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"
```

рис. 4.3.1

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера$ cd ~/work/study/2025-2026/Архитектура компьютера
```

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера$ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"
```

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера$ cd ~/work/study/2025-2026/"Архитектура компьютера"
```

увеличенный рис. 4.3.1

## **4.4 Создание репозитория курса на основе шаблона.**

Создаю репозиторий на основе имеющего шаблона (рис. 4.4.1) через функционал клонирования интерфейса GitHub. (рис 4.4.2)

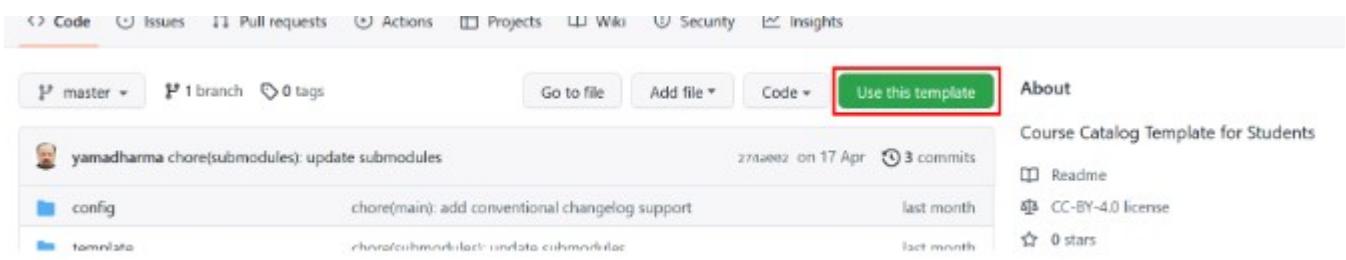


Рис. 4.4.1 Шаблон для клонирования на платформе GitHub.

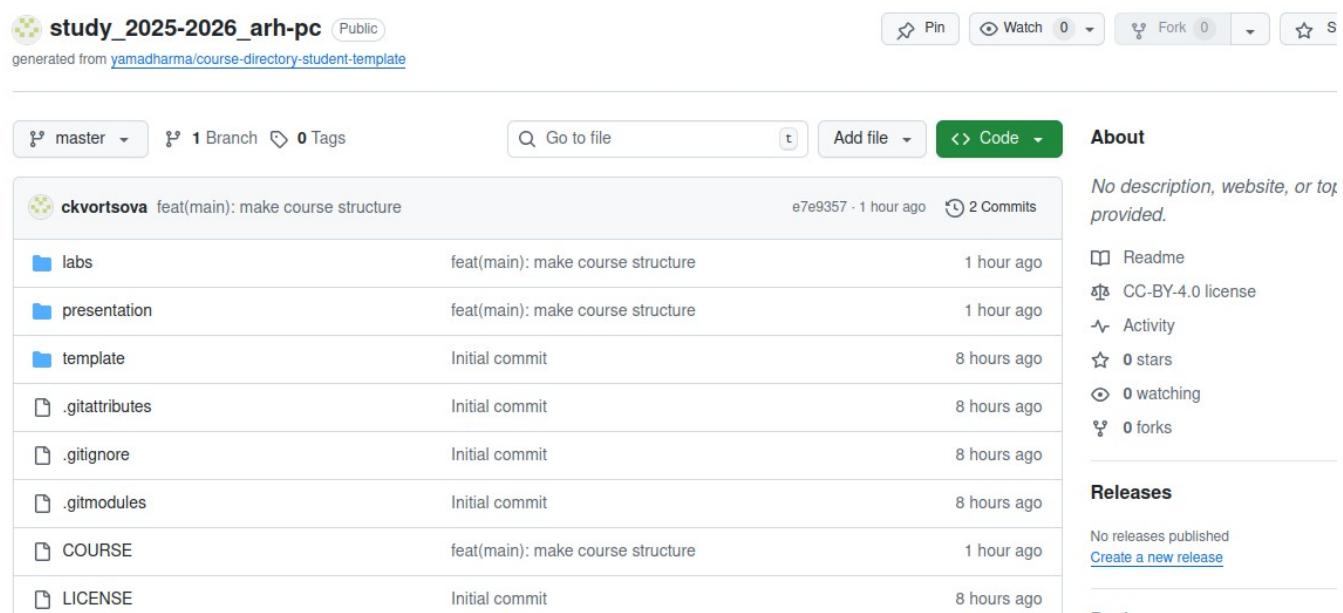


Рис. 4.4.2 Процесс клонирования репозитория через интерфейс GitHub.

Сгенерированный репозиторий на основе шаблона клонирую на свой рабочий компьютер, для этого беру ссылку для клонирования через интерфейс GitHub (рис. 4.4.3) и затем ввожу в терминале git clone. (рис 4.4.4)

Рис. 4.4.3 Копирование ссылки для последующей вставки в терминал.

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:/work/study/2025-2026/Архитектура компьютера$ cd ~/work/study/2025-2026/Архитектура компьютера"
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:/work/study/2025-2026/Архитектура компьютера$ git clone --recursive git@github.com:ckvortsova/study_2025-2026_arh-pc.git arch-pc
Клонирование в «arch-pc»...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 38 (delta 1), reused 26 (delta 1), pack-reused 0 (from 0)
Получение объектов: 100% (38/38), 23.57 Кб | 7.86 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 «/home/ckvortsova/work/study/2025-2026/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 219, done.
remote: Counting objects: 100% (219/219), done.
remote: Compressing objects: 100% (151/151), done.
remote: Total 219 (delta 86), reused 189 (delta 56), pack-reused 0 (from 0)
Получение объектов: 100% (219/219), 2.66 Мб | 5.30 Мб/с, готово.
Определение изменений: 100% (86/86), готово.
Клонирование в «/home/ckvortsova/work/study/2025-2026/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 251, done.
remote: Counting objects: 100% (251/251), done.
remote: Compressing objects: 100% (172/172), done.
remote: Total 251 (delta 111), reused 204 (delta 64), pack-reused 0 (from 0)
Получение объектов: 100% (251/251), 775.12 Кб | 1.46 Мб/с, готово.
Определение изменений: 100% (111/111), готово.
Submodule path 'template/presentation': checked out 'ic93acf9e731bf186384c85de4aff70037314240'
Submodule path 'template/report': checked out '8ee157c58b3362947b1c71492a65d4dc6882d5ad'
```

Рис. 4.4.4 Копирование репозитория на рабочий компьютер.

## 4.5 Настройка каталога курса.

В каталоге курса удаляю лишние файлы и формирую необходимые каталоги и делаю снимок сделанных изменений и push'у их на свой репозиторий в GitHub. (рис. 4.3.1)

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ make prepare
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git add .
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
```

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 73, готово.
Подсчет объектов: 100% (73/73), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (56/56), готово.
Запись объектов: 100% (70/70), 700.94 Киб | 4.83 МиБ/с, готово.
Всего 70 (изменений 24), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (24/24), completed with 1 local object.
To github.com:ckvortsova/study_2025-2026_arh-pc.git
  997ea34..e7e9357  master -> master
```

Рис. 4.3.1 Настройка каталога курса, отправка изменений на удаленный репозиторий. .

## 5 Задания для самостоятельной работы.

Через терминал отправляю предыдущий отчет по лабораторной работе на свой удаленный репозиторий в GitHub (рис. 5.1), затем проверяю изменения на самом GitHub.

```
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ make prepare
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git add .
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'

create mode 100644 presentation/report/images/course.jpg
ckvortsova@ckvortsova-HP-Pavilion-Laptop-14-ce3xxx:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 73, готово.
Подсчет объектов: 100% (73/73), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (56/56), готово.
Запись объектов: 100% (70/70), 700.94 Киб | 4.83 Миб/с, готово.
Всего 70 (изменений 24), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (24/24), completed with 1 local object.
To github.com:ckvortsova/study_2025-2026_arh-pc.git
  997ea34..e7e9357 master -> master
```

Рис 5.1 Отправка изменений на удаленный репозиторий.

## **6 Выводы**

При выполнении данной лабораторной работы я изучила идеологию и применение средств контроля версий, а также приобрела практические навыки по работе с реализацией VSC git.

## **Список литературы**

- 1.Пример выполнения лабораторной работы
- 2.Курс на ТУИС
- 3.Лабораторная работа №2  
[https://esystem.rudn.ru/pluginfile.php/2945866/mod\\_resource/content/0/lab2.pdf](https://esystem.rudn.ru/pluginfile.php/2945866/mod_resource/content/0/lab2.pdf)
- 4.Программирование на языке ассемблера NASM Столяров А. В.  
[https://esystem.rudn.ru/pluginfile.php/2088953/mod\\_resource/content/](https://esystem.rudn.ru/pluginfile.php/2088953/mod_resource/content/)  
2/%D0%A1%D1%82%D0%BE%D0%BB%D1%8F  
%D1%80%D0%BE  
%D0%B2%20%D0%90.%20%D0%92.%20-%20%D0%9F  
%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC  
%D0%BC  
%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD  
%D0%B8%D0%B5%20%D0%B0%D1%81%D1%81%D0%B5%D0%BC  
%D0%B1%D0%BB%D0%B5%D1%80%D0%B0%20NASM  
%20%D0%B4%D0%BB%D1%8F%20%D0%9E  
%D0%A1%20Unix.pdf