

Отчёт по лабораторной работе №2

**Дисциплина: Архитектура компьютеров и операционные
системы**

Гусев Степан Андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Настройка GitHub	9
4.2	Базовая настройка Git	9
4.3	Создание SSH-ключа	10
4.4	Создание рабочего пространства и репозитория курса на основе шаблона	11
4.5	Создание репозитория курса на основе шаблона	11
4.6	Настройка каталога курса	12
4.7	Выполнение заданий для самостоятельной работы.	14
5	Выводы	18
6	Список литературы	19

Список иллюстраций

4.1	Аккаунт GitHub	9
4.2	Предварительную конфигурация git	10
4.3	Генерация ключей	10
4.4	Копирование ключа	10
4.5	Добавление ключа	11
4.6	Создание каталога	11
4.7	Шаблон курса	11
4.8	Создание каталога	12
4.9	Перемещение между директориями	12
4.10	Клонирование репозитория	12
4.11	Перемещение между директориями	12
4.12	Создание директорий	13
4.13	Загрузка файлов на сервер	13
4.14	Загрузка файлов на сервер	13
4.15	Удаление предыдущих файлов	13
4.16	Компиляция файлов	14
4.17	Создание файлов	14
4.18	Поиск приложения	15
4.19	Открытие файла	15
4.20	Перемещение файла	15
4.21	git add	15
4.22	git push	16
4.23	Проверка	16
4.24	Проверка	16
4.25	Проверка	17
4.26	Проверка	17

Список таблиц

1 Цель работы

Изучить идеологию и применение средств контроля версий и научиться работать с системой контроля версий git.

2 Задание

1. Настройка 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.
7. Выполнение заданий для самостоятельной работы.

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 изменения, сделанные разными участниками, 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. 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. 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. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией. Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений). Затем можно вносить изменения в локальное дерево и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.

4 Выполнение лабораторной работы

4.1 Настройка GitHub

Создал учётную запись на сайте GitHub (рис. 4.1).

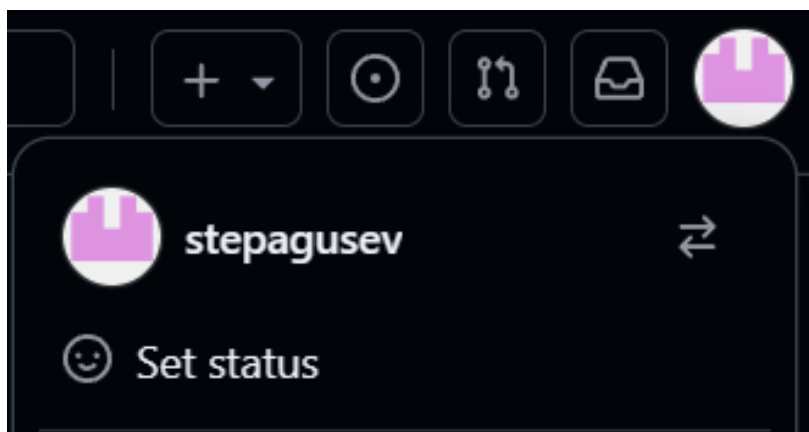


Рисунок 4.1: Аккаунт GitHub

4.2 Базовая настройка Git

Сделал предварительную конфигурацию git. В терминале ввёл следующие команды, указав свои имя и email. Сделал предварительную конфигурацию git. В терминале ввёл следующие команды, указав свои имя и email. Задал имя master для начальной ветки. Задал параметры autocrlf и safecrlf (рис. 4.2).

```
gusev-s-a@GusevSA:~$ git config --global user.name "<Stepan Gusev>"
gusev-s-a@GusevSA:~$ git config --global user.email "<1032242444@pfur.ru>"
gusev-s-a@GusevSA:~$ git config --global core.quotepath false
gusev-s-a@GusevSA:~$ git config --global init.defaultBranch master
gusev-s-a@GusevSA:~$ git config --global core.autocrlf input
gusev-s-a@GusevSA:~$ git config --global core.safecrlf warn
gusev-s-a@GusevSA:~$
```

Рисунок 4.2: Предварительную конфигурация git

4.3 Создание SSH-ключа

Сгенерировал пару ключей: приватный и открытый (рис. 4.3).

```
gusev-s-a@GusevSA:~$ ssh-keygen -C "Stepan Gusev <1032242444@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/gusev-s-a/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gusev-s-a/.ssh/id_ed25519
Your public key has been saved in /home/gusev-s-a/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:vTZWlv9518/Q7+wL75K9vbwxR+MhXaqqHR0j0Pg/404 Stepan Gusev <1032242444@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|
|   o   |
|  o .  |
| + . . . |
| S = * . . |
| . 0 =ooo|
| B = +* = |
| = + .+B& |
| .oE  @^  |
+----[SHA256]-----+
```

Рисунок 4.3: Генерация ключей

Скопировал ключ в буфер обмена (рис. 4.4).

```
gusev-s-a@GusevSA:~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
```

Рисунок 4.4: Копирование ключа

На сайте GitHub в меню settings в разделе SSH and GPG keys добавил только что созданный SSH-ключ (рис. 4.5).

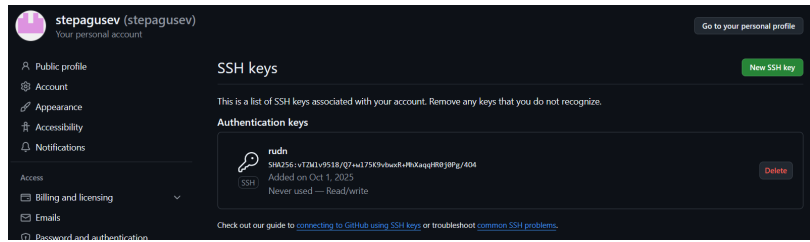


Рисунок 4.5: Добавление ключа

4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Создал каталог для предмета для предмета «Архитектура компьютера» (рис. 4.6).

```
gusev-s-a@GusevSA:~$ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"
```

Рисунок 4.6: Создание каталога

4.5 Создание репозитория курса на основе шаблона

В браузере перешёл на страницу репозитория с шаблоном курса. Далее нажал «Use this template», чтобы использовать этот шаблон для своего репозитория (рис. 4.7).

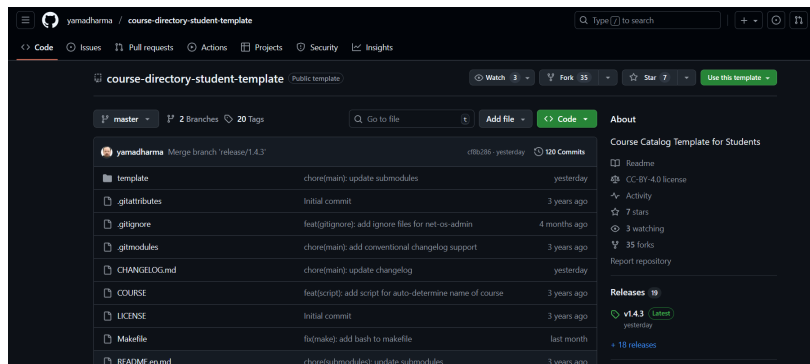


Рисунок 4.7: Шаблон курса

Задал имя репозитория и создал его (рис. 4.8).

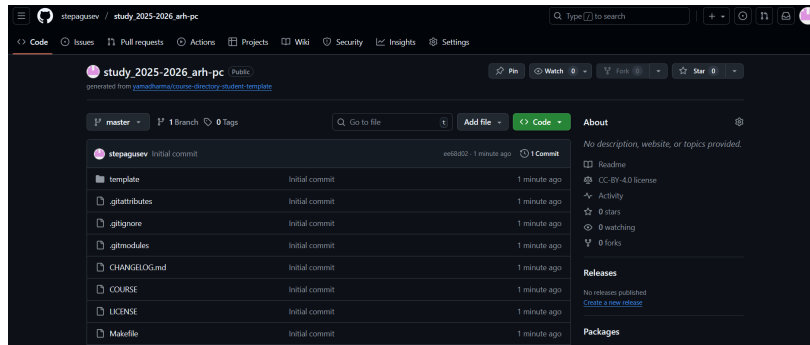


Рисунок 4.8: Создание каталога

В терминале перешёл в каталог курса (рис. 4.9).

```
gusev-s-a@GusevSA:~$ cd ~/work/study/2025-2026/Архитектура\ компьютера/
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура\ компьютера$
```

Рисунок 4.9: Перемещение между директориями

Клонировал созданный репозиторий (рис. 4.10).

```
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура\ компьютера$ git clone --recursive git@github.com:stepagusev/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 28 (delta 1), pack-reused 0 (from 0)
Получение объектов: 100% (38/38), 23.53 KiB | 3.36 MiB/c, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yanadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yanadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/gusev-s-a/work/study/2025-2026/Архитектура\ компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 182, done.
remote: Counting objects: 100% (182/182), done.
remote: Compressing objects: 100% (122/122), done.
remote: Total 182 (delta 78), reused 162 (delta 58), pack-reused 0 (from 0)
Получение объектов: 100% (182/182), 2.65 MiB | 4.76 MiB/c, готово.
Определение изменений: 100% (78/78), готово.
Клонирование в «/home/gusev-s-a/work/study/2025-2026/Архитектура\ компьютера/arch-pc/template/report»...
remote: Enumerating objects: 239, done.
remote: Counting objects: 100% (239/239), done.
remote: Compressing objects: 100% (163/163), done.
remote: Total 239 (delta 105), reused 196 (delta 62), pack-reused 0 (from 0)
Получение объектов: 100% (239/239), 771.21 KiB | 2.89 MiB/c, готово.
Определение изменений: 100% (185/185), готово.
Submodule path 'template/presentation': checked out 'a1bc497b7b15dd240c7c105508f478f8bcc345e7'
Submodule path 'template/report': checked out '192c8bcd638d72c8ad0f3b98a67d4e7eddc5db16'
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура\ компьютера$
```

Рисунок 4.10: Клонирование репозитория

4.6 Настройка каталога курса

Перешёл в каталог курса (рис. 4.11).

```
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура\ компьютера$ cd arch-pc/
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура\ компьютера/arch-pc$
```

Рисунок 4.11: Перемещение между директориями

Создал необходимые каталоги (рис. 4.12).

```
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ make prepare
```

Рисунок 4.12: Создание директорий

Отправил файлы на сервер (рис. 4.13)(рис. 4.14).

```
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git add .
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git commit -an 'feat(main): make course structure'
[master dc5f52f] feat(main): make course structure
260 files changed, 8746 insertions(+), 216 deletions(-)
delete mode 100644 CHANGELOG.md
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.gitignore
create mode 100644 labs/lab01/presentation/.marksman.toml
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/_assets/auto/beamer.el
create mode 100644 labs/lab01/presentation/_assets/auto/preamble.el
```

Рисунок 4.13: Загрузка файлов на сервер

```
create mode 100644 presentation/report/bib/cite.bib
create mode 100644 presentation/report/image/solvay.jpg
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 74, готово.
Подсчет объектов: 100% (74/74), готово.
При схватки изменений используется до 2 потоков
Сжатие объектов: 100% (58/58), готово.
Запись объектов: 100% (71/71), 700.98 КиБ | 7.46 МиБ/с, готово.
сего 71 (изменений 25), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (25/25), completed with 1 local object.
to github.com:stepagusev/study_2025-2026_arh-pc.git
ee68d02..dc5f52f master -> master
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура компьютера/arch-pc$
```

Рисунок 4.14: Загрузка файлов на сервер

Проверил правильность создания иерархии рабочего пространства в локальном репозитории (рис. 4.15).

```
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ ls
COURSE labs LICENSE Makefile package.json prepare presentation README.en.md README.git-flow.md README.md template
gusev-s-a@GusevSA:~/work/study/2025-2026/Архитектура компьютера/arch-pc$
```

Рисунок 4.15: Удаление предыдущих файлов

Проверил правильность создания иерархии рабочего пространства на странице GitHub (рис. 4.16).

File/Folder	Commit Message	Time Ago
labs	feat(main): make course structure	1 minute ago
presentation	feat(main): make course structure	1 minute ago
template	Initial commit	9 minutes ago
.gitattributes	Initial commit	9 minutes ago
.gitignore	Initial commit	9 minutes ago
.gitmodules	Initial commit	9 minutes ago
COURSE	feat(main): make course structure	1 minute ago
LICENSE	Initial commit	9 minutes ago
Makefile	Initial commit	9 minutes ago
README.en.md	Initial commit	9 minutes ago
README.git-flow.md	Initial commit	9 minutes ago
README.md	Initial commit	9 minutes ago
package.json	feat(main): make course structure	1 minute ago
prepare	feat(main): make course structure	1 minute ago

Рисунок 4.16: Компиляция файлов

4.7 Выполнение заданий для самостоятельной работы.

Перешёл с помощью команды `cd` в каталог `labs/lab02/report` и создал в нём файл для отчёта по второй лабораторной работе с помощью команды `touch`. Проверил, что файл создан, с помощью команды `ls` (рис. 4.17).

```
gusev-s-a@gusevsa:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ cd labs/lab02/report
gusev-s-a@gusevsa:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab02/report$ touch l02_fysev_отчет
gusev-s-a@gusevsa:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab02/report$ ls
arch-pc-lab02-report.qmd  _assets  bib  image  Makefile  _quarto.yml  _resources  l02_fysev_отчет
gusev-s-a@gusevsa:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab02/report$
```

Рисунок 4.17: Создание файлов

Оформить отчёт я смогу в текстовом редакторе LibreOffice Writer, найдя его в меню приложений (рис. 4.18).

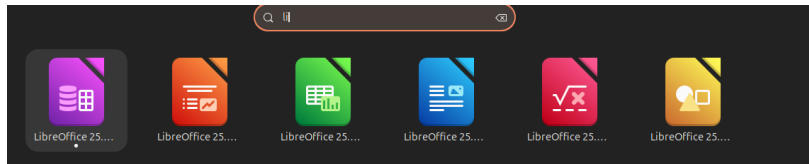


Рисунок 4.18: Поиск приложения

После открытия редактора, открыл в нем созданный файл и начал работу над отчётом (рис. 4.19).

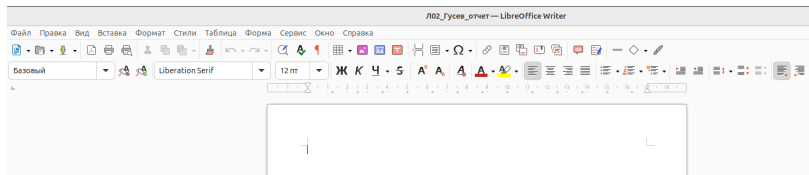


Рисунок 4.19: Открытие файла

С помощью команды `cd` перешёл в каталог `lab01/report` и переместил в него отчёт по первой лабораторной работе с помощью команды `mv`. Проверил, что файл успешно перемещён, с помощью `ls` (рис. 4.20).

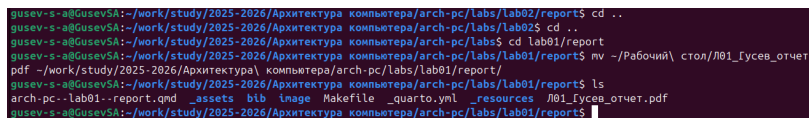


Рисунок 4.20: Перемещение файла

С помощью команды `git add` добавил в коммит отчёты по двум лабораторным работам. Сохранил изменения на сервере командой `git commit -m «Add existing file»`, пояснив добавление файлов (рис. 4.21)(рис. 4.22).

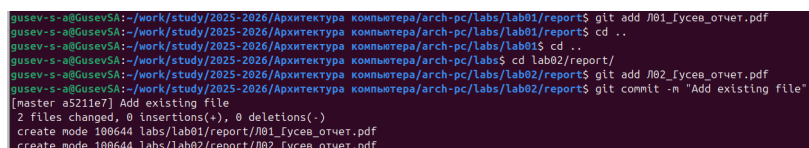


Рисунок 4.21: git add

Отправил в центральный репозиторий изменения сохранённые изменения командой `git push -f origin master` (рис. 4.22).

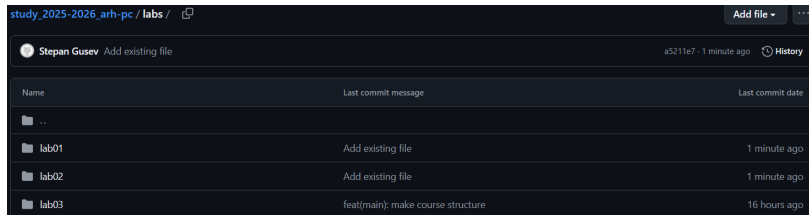
```

stepan@stepanov:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab02/report$ git push -f origin master
Перечисление объектов: 14, готово.
Подсчет объектов: 100% (14/14), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (8/8), готово.
Запись объектов: 100% (8/8), 1023.41 Киб | 7.87 Миб/с, готово.
Всего 8 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:stepagusev/study_2025-2026_arch-pc.git
dc5f52f...a5211e7 master -> master

```

Рисунок 4.22: git push

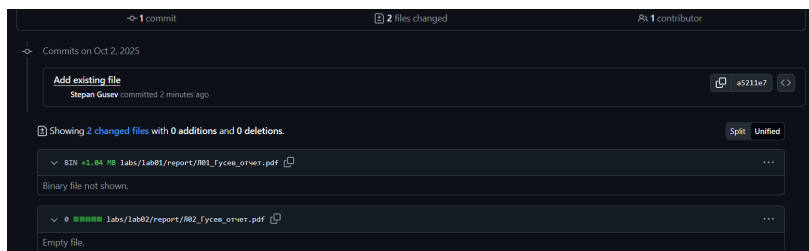
Проверил на сайте GitHub правильность выполнения заданий. Вижу, что пояснение к совершённым действиям отображается (рис. 4.23).



Name	Last commit message	Last commit date
..		
lab01	Add existing file	1 minute ago
lab02	Add existing file	1 minute ago
lab03	feat(main): make course structure	16 hours ago

Рисунок 4.23: Проверка

При просмотре изменений увидел, что были добавлены файлы с отчётами по лабораторным работам (рис. 4.24).



File	Changes
labs/lab01/report/001_fycew_orwer.pdf	Binary file not shown.
labs/lab02/report/002_fycew_orwer.pdf	Empty file.

Рисунок 4.24: Проверка

Проверил, что отчёты по лабораторным работам находятся в соответствующих каталогах репозитория: отчёт по первой лабораторной в lab01/report (рис. 4.25), по второй – в lab02/report (рис. 4.26).

study_2025-2026_arh-pc / labs / lab01 / report / Add file ...

Stepan Gusev Add existing file a5211e7 · 3 minutes ago History

Name	Last commit message	Last commit date
..		
assets	feat(main): make course structure	16 hours ago
resources/csl	feat(main): make course structure	16 hours ago
bib	feat(main): make course structure	16 hours ago
image	feat(main): make course structure	16 hours ago
.gitignore	feat(main): make course structure	16 hours ago
.marksmat.toml	feat(main): make course structure	16 hours ago
.projectile	feat(main): make course structure	16 hours ago
Makefile	feat(main): make course structure	16 hours ago
.quarto.yml	feat(main): make course structure	16 hours ago
arch-pc-lab01-report.qmd	feat(main): make course structure	16 hours ago
/01_fyceo_orver.pdf	Add existing file	3 minutes ago

Рисунок 4.25: Проверка

study_2025-2026_arh-pc / labs / lab02 / report / Add file ...

Stepan Gusev Add existing file a5211e7 · 4 minutes ago History

Name	Last commit message	Last commit date
..		
assets	feat(main): make course structure	16 hours ago
resources/csl	feat(main): make course structure	16 hours ago
bib	feat(main): make course structure	16 hours ago
image	feat(main): make course structure	16 hours ago
.gitignore	feat(main): make course structure	16 hours ago
.marksmat.toml	feat(main): make course structure	16 hours ago
.projectile	feat(main): make course structure	16 hours ago
Makefile	feat(main): make course structure	16 hours ago
.quarto.yml	feat(main): make course structure	16 hours ago
arch-pc-lab02-report.qmd	feat(main): make course structure	16 hours ago
/02_fyceo_orver.pdf	Add existing file	4 minutes ago

Рисунок 4.26: Проверка

5 Выводы

В ходе работы я приобрел практические навыки работы с системой контроля версий GitHub, разобрав следующие команды: clone, add, commit, push. Также изучил идеологию и применение подобных систем.

6 Список литературы

1. Архитектура ЭВМ
2. Git - gitattributes Документация