

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: Павлушина Виктория Александровна

Группа: НКАбд-05-25

МОСКВА

2025 г.

Оглавление

Цель работы	3
1. Теоретическое введение.....	4
1.2. Системы контроля версий. Общие понятия	4
1.3. Система контроля версий Git	5
1.4. Стандартные процедуры работы при наличии центрального репозитория	6
2. Выполнение лабораторной работы.....	8
2.1. Настройка github.....	8
2.2. Базовая настройка git	8
2.3 Создание SSH-ключа.....	9
2.4 Создание рабочего пространства и репозитория курса на основе шаблона	10
2.5. Создание репозитория курса на основе шаблона.....	11
2.6 Настройка каталога курса.....	12
3. Задание для самостоятельной работы	14
Вывод.....	15
Список литературы:	16

Цель работы

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

1. Теоретическое введение

1.2. Системы контроля версий. Общие понятия

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. 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.

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.

1.3. Система контроля версий Git

Система контроля версий Git представляет собой набор программ командной строки.

Основные команды git :

Таблица 1 Описание основных команд git

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

Команда	Описание
<code>git add</code> имена_файлов	добавить конкретные изменённые и/или созданные файлы и/или каталоги
<code>git rm</code> имена_файлов	удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)
<code>git commit -am</code> 'Описание коммита'	сохранить все добавленные изменения и все изменённые файлы
<code>git checkout -b</code> имя_ветки	создание новой ветки, базирующейся на текущей
<code>git checkout</code> имя_ветки	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> имя_ветки	отправка изменений конкретной ветки в центральный репозиторий
<code>git merge</code> --no-ff имя_ветки	слияние ветки с текущим деревом
<code>git branch -d</code> имя_ветки	удаление локальной уже слитой с основным деревом ветки
<code>git branch -D</code> имя_ветки	принудительное удаление локальной ветки
<code>git push origin</code> :имя_ветки	удаление ветки с центрального репозитория

1.4. Стандартные процедуры работы при наличии центрального репозитория

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):

git checkout master

git pull

git checkout -b имя_ветки

Затем можно вносить изменения в локальном дереве и/или ветке.

После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории. Для этого необходимо проверить, какие файлы изменились к текущему моменту:

git status

При необходимости удаляем лишние файлы, которые не хотим отправлять в центральный репозиторий. Затем полезно просмотреть текст изменений на предмет соответствия правилам ведения чистых коммитов:

git diff

Если какие-либо файлы не должны попасть в коммит, то помечаем только те файлы, изменения которых нужно сохранить. Для этого используем команды добавления и/или удаления с нужными опциями:

git add имена_файлов

git rm имена_файлов

Если нужно сохранить все изменения в текущем каталоге, то используем:

git add .

Затем сохраняем изменения, поясняя, что было сделано:

git commit -am "Some commit message"

и отправляем в центральный репозиторий:

git push origin имя_ветки

или

git push

2. Выполнение лабораторной работы

2.1. Настройка github

Создадим учётную запись на сайте <https://github.com/> и заполним основные данные:

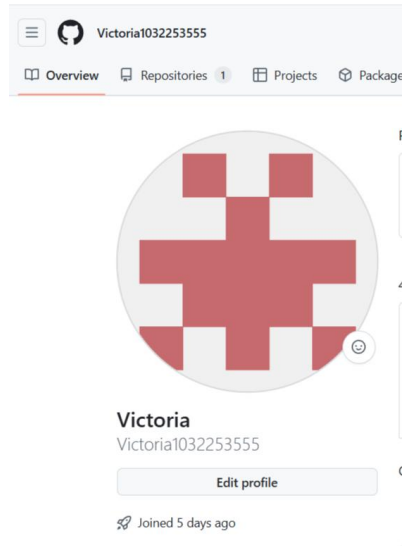


Рисунок 1 Созданный аккаунт в github

2.2. Базовая настройка git

Сначала сделаем предварительную конфигурацию git. Откроем терминал и введём следующие команды, указав имя и e-mail владельца репозитория:

```
vapavlushina@dk8n60 ~ $ git config --global user.name "<Victoria>"
vapavlushina@dk8n60 ~ $ git config --global user.email "<1032253555@pfur.ru>"
```

Рисунок 2 Вводим данные

Настроим utf-8 в выводе сообщений **git**:

```
vapavlushina@dk8n60 ~ $ git config --global core.quotepath false
```

Рисунок 3 Настройка utf-8

Зададим имя начальной ветки (будем называть её **master**):

```
vapavlushina@dk8n60 ~ $ git config --global init.defaultBranch master
```

Рисунок 4 Задаём название начальной ветки

Параметр **autocrlf**:

```
vapavlushina@dk8n60 ~ $ git config --global core.autocrlf input
```

Рисунок 5 Используем autocrlf

Параметр **safecrlf**:

```
vapavlushina@dk8n60 ~ $ git config --global core.safecrlf warn
```

Рисунок 6 Используем satecrlf

2.3 Создание SSH-ключа

Создадим ключи с помощью команды :

ssh-keygen -C “Имя Фамилия <work@mail>”

У нас получилось :

```
vapavlushina@dk8n63 ~ $ ssh-keygen -C "Victoria Pavlushina <1032253555@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/v/a/vapavlushina/.ssh/id_ed25519): /afs/.
dk.sci.pfu.edu.ru/home/v/a/vapavlushina/Victoria pavlushina.odt
/afs/.dk.sci.pfu.edu.ru/home/v/a/vapavlushina/Victoria pavlushina.odt already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/a/vapavlushina/Victoria pavlushina.odt
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/v/a/vapavlushina/Victoria pavlushina.odt.pub
The key fingerprint is:
SHA256:W+Kd/rcG/55wpeJRh8UbmTnFMnOaQg+/V53FVz1XHE Victoria Pavlushina <1032253555@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|                ooE|
|                .@+|
|                .o.+oX|
|                o.oob+|
|                S . .o=.+|
|                . = ... +=|
|                o o o+o.+|
|                . . o*..|
|                ..oo.=o|
+-----[SHA256]-----+
```

Рисунок 7 Создание ключа

Теперь, с помощью команды

cat ~/.ssh/id_rsa.pub | xclip -sel clip

```
vapavlushina@dk8n63 ~ $ cd .ssh/  
vapavlushina@dk8n63 ~/.ssh $ ls  
id_ed25519 id_ed25519.pub known_hosts  
vapavlushina@dk8n63 ~/.ssh $ cat id_ed25519.pub  
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBguCGUcLFJjZHpCq1TIungyegk1cUgJMh6k+yU5hYgk Victoria Pavolushina <1032253555@pfur.ru>
```

Рисунок 8 cat

Копируем из локальной консоли ключ в буфер обмена,
Вставляем его в появившееся на сайте поле и указываем для
ключа имя (Title).

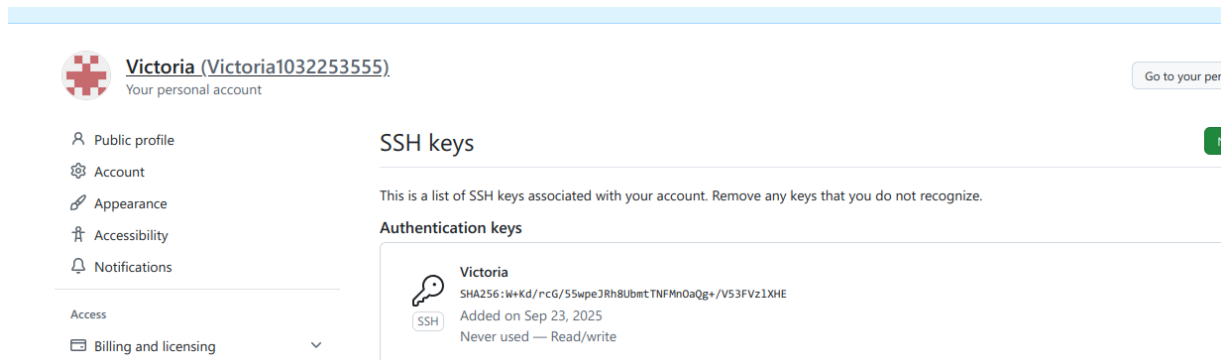


Рисунок 9 Ключ

2.4 Создание рабочего пространства и репозитория курса на основе шаблона

Откроем терминал и создадим каталог для предмета
«Архитектура компьютера»:

mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"

```
+ vapavlushina@dk8n63 - vapavlushina  
vapavlushina@dk8n63 ~ $ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"  
vapavlushina@dk8n63 ~ $
```

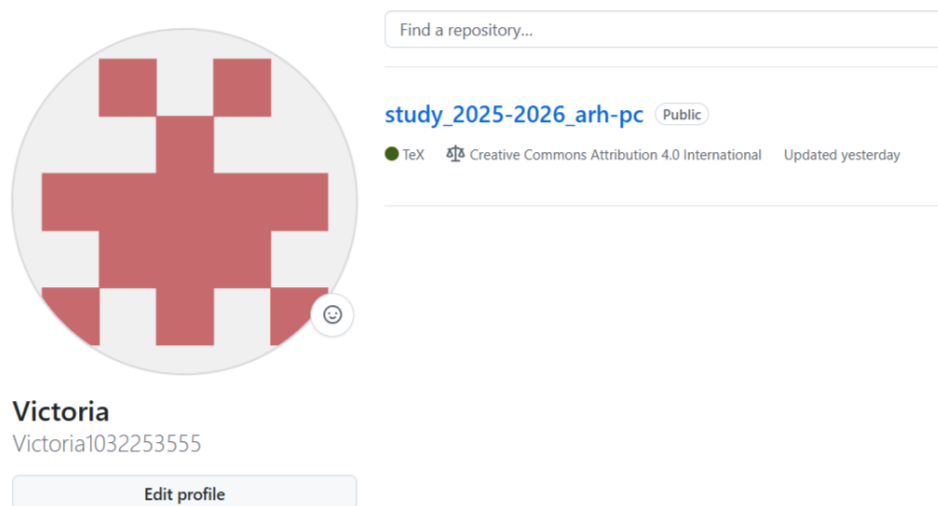
Рисунок 10 Создание рабочего пространства

2.5. Создание репозитория курса на основе шаблона

Репозиторий можно создать через web-интерфейс **github**.

Перейдём на страницу репозитория с шаблоном курса:
<https://github.com/yamadharma/course-directory-student-template>.

Далее выберем : **Use this template** .



В открывшемся окне зададим имя репозитория (Repository name) **study_2025–2026_arh-pc** и создадим репозиторий :

Рисунок 11 Создание репозитория

Теперь откроем терминал и перейдем в каталог курса с помощью команды **cd** и скопируем созданный репозиторий с помощью команды

```
git clone --recursive git@github.com:/study_2025–2026_arh-pc.git arch-pc:
```

```

vapavlushina@dk8n63 ~/.ssh $ cd
vapavlushina@dk8n63 ~ $ cd work/study/2025-2026/Архитектура\ компьютера/
vapavlushina@dk8n63 ~/work/study/2025-2026/Архитектура компьютера $ git clone --recursive git@github.com:Victoria1032253555/study_2025-2026_arh-pc.git
Клонирование в «study_2025-2026_arh-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 27 (delta 1), pack-reused 0 (from 0)
Получение объектов: 100% (38/38), 23.45 КиБ | 600.00 КиБ/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/v/a/vapavlushina/work/study/2025-2026/ Архитектура компьютера/study_2025-2026_arh-pc/template/presentation»...
remote: Enumerating objects: 161, done.
remote: Counting objects: 100% (161/161), done.
remote: Compressing objects: 100% (111/111), done.
remote: Total 161 (delta 60), reused 142 (delta 41), pack-reused 0 (from 0)
Получение объектов: 100% (161/161), 2.65 МиБ | 7.86 МиБ/с, готово.
Определение изменений: 100% (60/60), готово.
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/v/a/vapavlushina/work/study/2025-2026/ Архитектура компьютера/study_2025-2026_arh-pc/template/report»...
remote: Enumerating objects: 221, done.
remote: Counting objects: 100% (221/221), done.
remote: Compressing objects: 100% (152/152), done.
remote: Total 221 (delta 98), reused 180 (delta 57), pack-reused 0 (from 0)
Получение объектов: 100% (221/221), 765.46 КиБ | 3.87 МиБ/с, готово.
Определение изменений: 100% (98/98), готово.
Submodule path 'template/presentation': checked out '6efd5c4ee78e4456caff3dc7062cfcad26058ca6'
Submodule path 'template/report': checked out '89a9622199b4df88227b9b3fa3d4714c85f68dd2'

```

Рисунок 12 Копирование репозитория

2.6 Настройка каталога курса

Перейдем в каталог курса с помощью команды **cd**,
создадим необходимые каталоги(

echo arch-pc > COURSE

make prepare

)

и отправим файлы на сервер (

git add .

git commit -am 'feat(main): make course structure'

git push

)

```

make prepare
vavavlushina@dk8n63 ~/work/study/2025-2026/Архитектура компьютера $ cd study_2025-2026_arh-pc
vavavlushina@dk8n63 ~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc $ echo arch-pc > COURSE
make prepare
vavavlushina@dk8n63 ~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc $ git add .
git commit -am 'feat(main): make course structure'
git push
[master 0615f70] feat(main): make course structure
212 files changed, 8074 insertions(+), 207 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/_quarto.yml
create mode 100644 labs/lab01/presentation/_resources/image/logo_rudn.png
create mode 100644 labs/lab01/presentation/arch-pc--lab01--presentation.qmd
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/report/.gitignore
create mode 100644 labs/lab01/report/.marksman.toml
create mode 100644 labs/lab01/report/.projectile
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/_quarto.yml
create mode 100644 labs/lab01/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab01/report/arch-pc--lab01--report.qmd
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/solvay.jpg
create mode 100644 labs/lab02/presentation/.gitignore
create mode 100644 labs/lab02/presentation/.marksman.toml
create mode 100644 labs/lab02/presentation/.projectile
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/_quarto.yml

```

Рисунок 13 Настройка каталога курса

3. Задание для самостоятельной работы

Загружаем отчёт на 1 лабораторную работу на github в **labs/lab01/report**

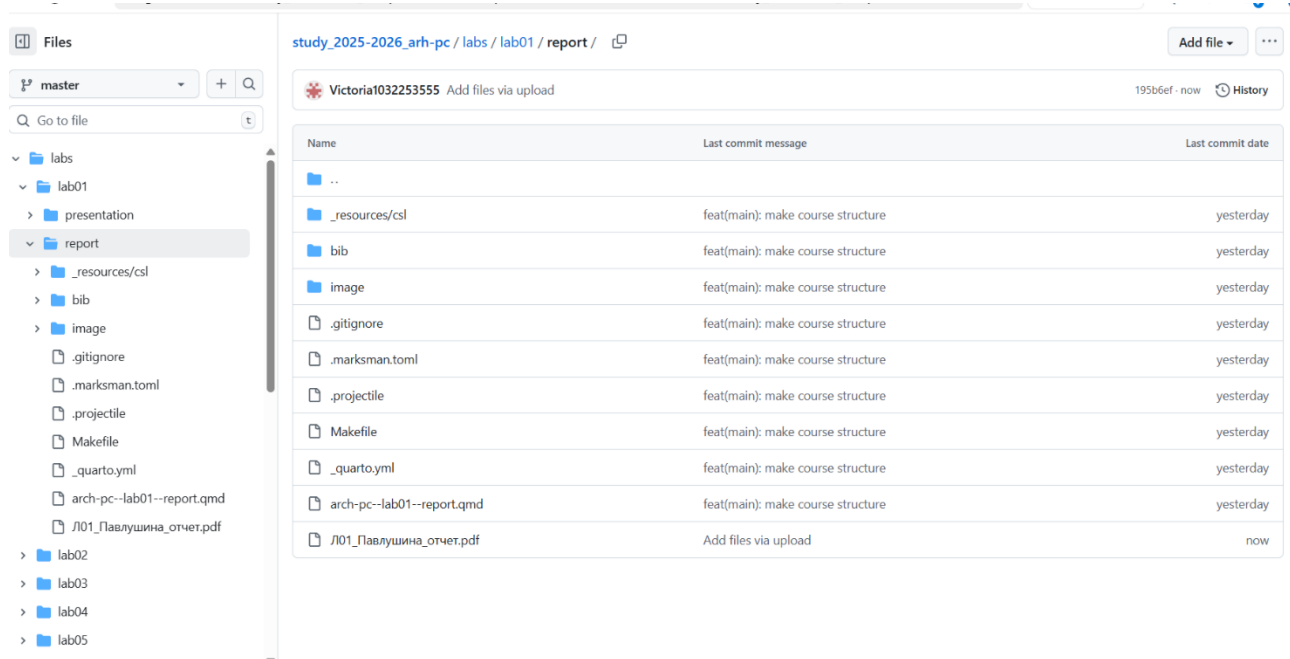


Рисунок 14 Загрузка отчёта по 1 лаб. работе на github

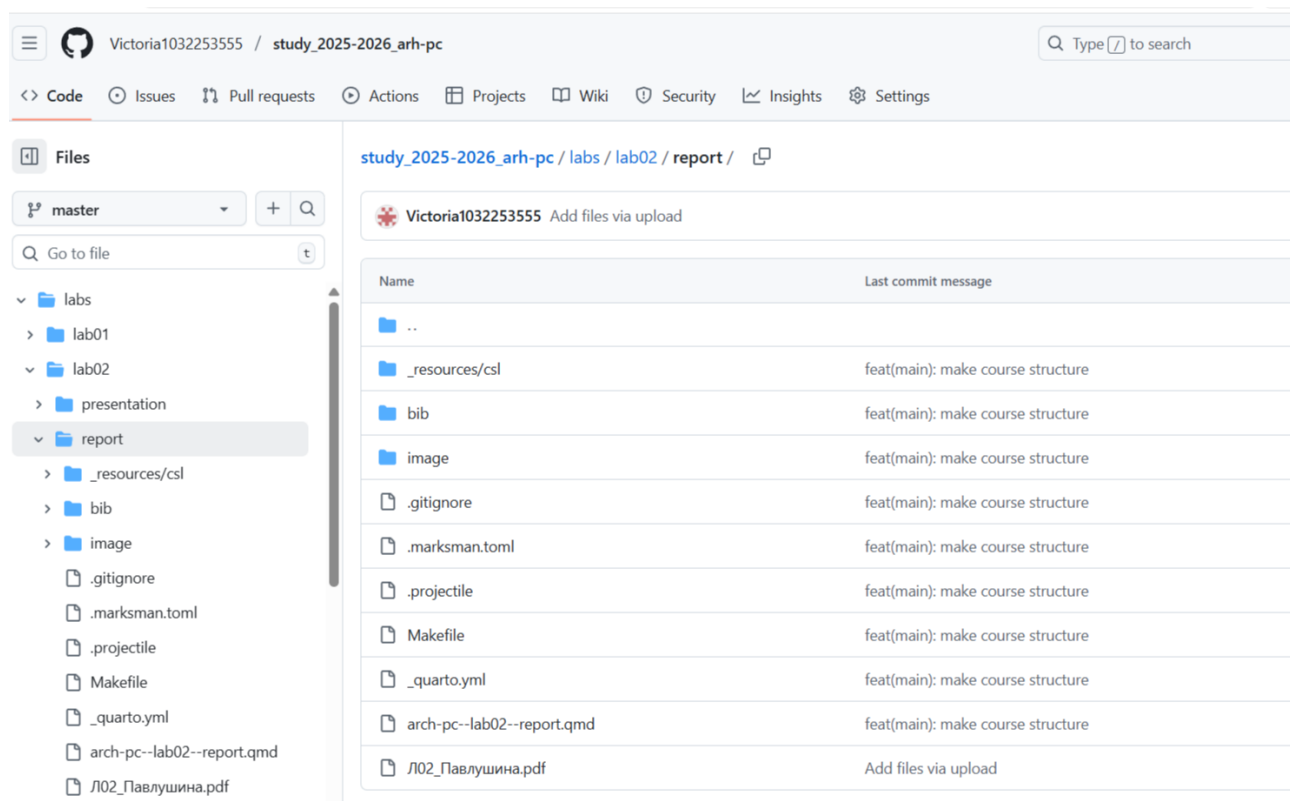


Рисунок 15 Загрузка отчёта по 2 лаб. работе на github

Вывод

В ходе выполнения лабораторной работы были успешно достигнуты поставленные цели по изучению идеологии и практическому освоению системы контроля версий Git. Были приобретены следующие ключевые навыки:

- Освоена работа с базовыми командами Git: **git add, git commit, git status;**
- Приобретен опыт работы с удаленными репозиториями: **git clone, git push, git pull;**
- Приобретен навык работы с хостингом **Git-репозитория** (GitHub).

Список литературы:

- 1) Основные команды git : <https://git-scm.com/book/ru/v2/Приложение-C%3A-Команды-Git-Основные-команды>
- 2) Настройка Git и создание SSH ключа : <https://htmlacademy.ru/blog/git/git-console>
- 3) Создание репозитория : <https://docs.github.com/ru/repositories/creating-and-managing-repositories/creating-a-new-repository>