

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Архитектура компьютера

Студент: Мальянц В. К.

Группа: НКАбд-03-24

МОСКВА

2024 г.

Содержание

1. Цель работы.....	5
2. Задание.....	6
3. Теоретическое введение.....	7
4. Выполнение лабораторной работы.....	9
5. Вывод.....	19
Список литературы.....	20

Список иллюстраций

4.1 Создание учетной записи на github.....	8
4.2.1 Предварительная конфигурация git (имя владельца репозитория).....	9
4.2.2 Предварительная конфигурация git (почта владельца репозитория).....	9
4.2.3 Настройка кодировки.....	9
4.2.4 Создание имени начальной ветки.....	9
4.2.5. Параметр autocrlf.....	9
4.2.5. Параметр safecrlf.....	9
4.3.1 Генерация SSH-ключа.....	10
4.3.2 Вывод содержимого файла с открытым ключом.....	10
4.3.3 Добавление ключа.....	10
4.3.4 Созданный SSH-ключ.....	11
4.4 Создание рабочего пространства.....	11
4.5.1 Создание репозитория.....	12
4.5.2 Созданный репозиторий.....	13
4.5.3 Переход в каталог курса.....	13
4.5.4 Клонирование репозитория.....	14
4.5.5 Ссылка для клонирования репозитория.....	14
4.6.1 Перемещение в каталог курса.....	14
4.6.2 Удаление файлов.....	15
4.6.3 Создание каталогов.....	15
4.6.4 Добавление и сохранение изменений на сервере.....	15
4.6.5 Выгрузка изменений на сервер.....	15

4.6.6 Репозиторий.....	16
4.7.1 Добавление отчета по Лабораторной работе № 1.....	17
4.7.2 Добавление отчета по Лабораторной работе № 2.....	18

1 Цель работы

Изучение идеологии применения средств контроля версий.

Приобретение практических навыков о работе с системой git.

2 Задание

4.1 Настройка github.

4.2 Базовая настройка git.

4.3 Создание SSH-ключа.

4.4 Создание рабочего пространства и репозитория курса на основе шаблона.

4.5 Создание репозитория курса на основе шаблона.

4.6 Настройка каталога курса.

4.7 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 (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.

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.

4 Выполнение лабораторной работы

4.1 Настройка github

Создала учетную запись на сайте <https://github.com/> и заполнила свои данные (рис. 4.1).



Рис. 4.1 Создание учетной записи на github

4.2 Базовая настройка git

Открываю терминал и делаю предварительную конфигурацию git. Ввожу команды: `git config --global user.name ""` и указываю в ней имя владельца репозитория (свое имя), `git config --global user.email ""` и указываю в ней почту владельца репозитория (свою почту) (рис. 4.2.1, рис. 4.2.2).

```
vkmaljanc@localhost-live:~$ git config --global user.name "<Victoria Malyants>"
```

Рис. 4.2.1 Предварительная конфигурация git (имя владельца репозитория)

```
vkmaljanc@localhost-live:~$ git config --global user.email "<1132246740@pfur.ru>"
```

Рис. 4.2.2 Предварительная конфигурация git (почта владельца репозитория)

Настраиваю utf-8 в выводе сообщений git (рис. 4.2.3).

```
vkmaljanc@localhost-live:~$ git config --global core.quotePath false
```

Рис. 4.2.3 Настройка кодировки

Задаю имя начальной ветки, называю ее master (рис. 4.2.4).

```
vkmaljanc@localhost-live:~$ git config --global init.defaultBranch master
```

Рис. 4.2.4 Создание имени начальной ветки

Параметр autocrlf (рис. 4.2.5).

```
vkmaljanc@localhost-live:~$ git config --global core.autocrlf input
```

Рис. 4.2.5 Параметр autocrlf

Параметр safecrlf (рис. 4.2.6).

```
vkmaljanc@localhost-live:~$ git config --global core.safecrlf warn
```

Рис. 4.2.6 safecrlf

4.3 Создание SSH ключа

Генерирую пару ключей для идентификации пользователя на сервере репозитория (рис. 4.3.1).

```

vkmaljyanc@localhost-live:~$ ssh-keygen -C "Victoria Malyants <1132246740@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/vkmaljyanc/.ssh/id_ed25519):
Created directory '/home/vkmaljyanc/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vkmaljyanc/.ssh/id_ed25519
Your public key has been saved in /home/vkmaljyanc/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:hAtDTNaJsIbyZQowxi9mnR3rX/vtRxjChUBdCqraUA Victoria Malyants <1132246740@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|=.++o . . .o.o |
|=.+o o. . . o . |
| = o.=E. o. . . |
|. =.+. + .o . |
|.o+ ..o S. . |
|o ..o . |
|      +. . = |
|      . . . = + |
|      . o+ o |
+-----[SHA256]-----+

```

Рис. 4.3.1 Генерация SSH-ключа

Вывожу содержимое файла с открытым ключом с помощью команды `cat` (рис. 4.3.2).

```

vkmaljyanc@localhost-live:~$ cat /home/vkmaljyanc/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKRmTiZPTlbV4S5H3yI23++bmc11TnSHFung7Xd8PpyX Victoria Malyants <1132246740@pfur.ru>

```

Рис. 4.3.2 Вывод содержимого файла с открытым ключом

Вставляю ключ в окно New SSH key, указывая имя ключа (рис. 4.3.3).

Рис. 4.3.3 Добавление ключа

Убеждаюсь в том, что SSH key создан (рис. 4.3.4).

victoriamalyants (victoriamalyants)
Your personal account

Go to your personal profile

Public profile
Account
Appearance
Accessibility
Notifications

Access
Billing and plans
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

Key name	Fingerprint	Added on	Used	Actions
victoriamalyants	SHA256:hAtDTNaJsIbyZQowx119mnR3rX/vtRxjChUBdCqraUA	Sep 23, 2024	Never used	Read/write Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

New SSH key
New GPG key

Рис. 4.3.4 Созданный SSH-ключ

4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Создаю каталог для предмета «Архитектура компьютера» с помощью `mkdir -p` (рис. 4.4).

```
vkmalijanc@localhost-live:~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"
```

Рис. 4.4 Создание рабочего пространства

4.5 Создание репозитория курса на основе шаблона

Перехожу на страницу репозитория с шаблоном курса

<https://github.com/yamadharma/course-directory-student-template>. Задаю имя репозитория и создаю его (рис. 4.5.1, рис. 4.5.2).

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 (*).

Repository template

 yamadharma/course-directory-student-template ▾

Start your repository with a template repository's contents.

☐ **Include all branches**

Copy all branches from yamadharma/course-directory-student-template and not just the default branch.

Owner *

 victoriamalyants ▾

Repository name *

/ study_2024-2025_arh-pc

✔ Your new repository will be created as **study_2024-2025_arh-pc**.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about **refactored-octo-system** ?

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

Рис. 4.5.1 Создание репозитория

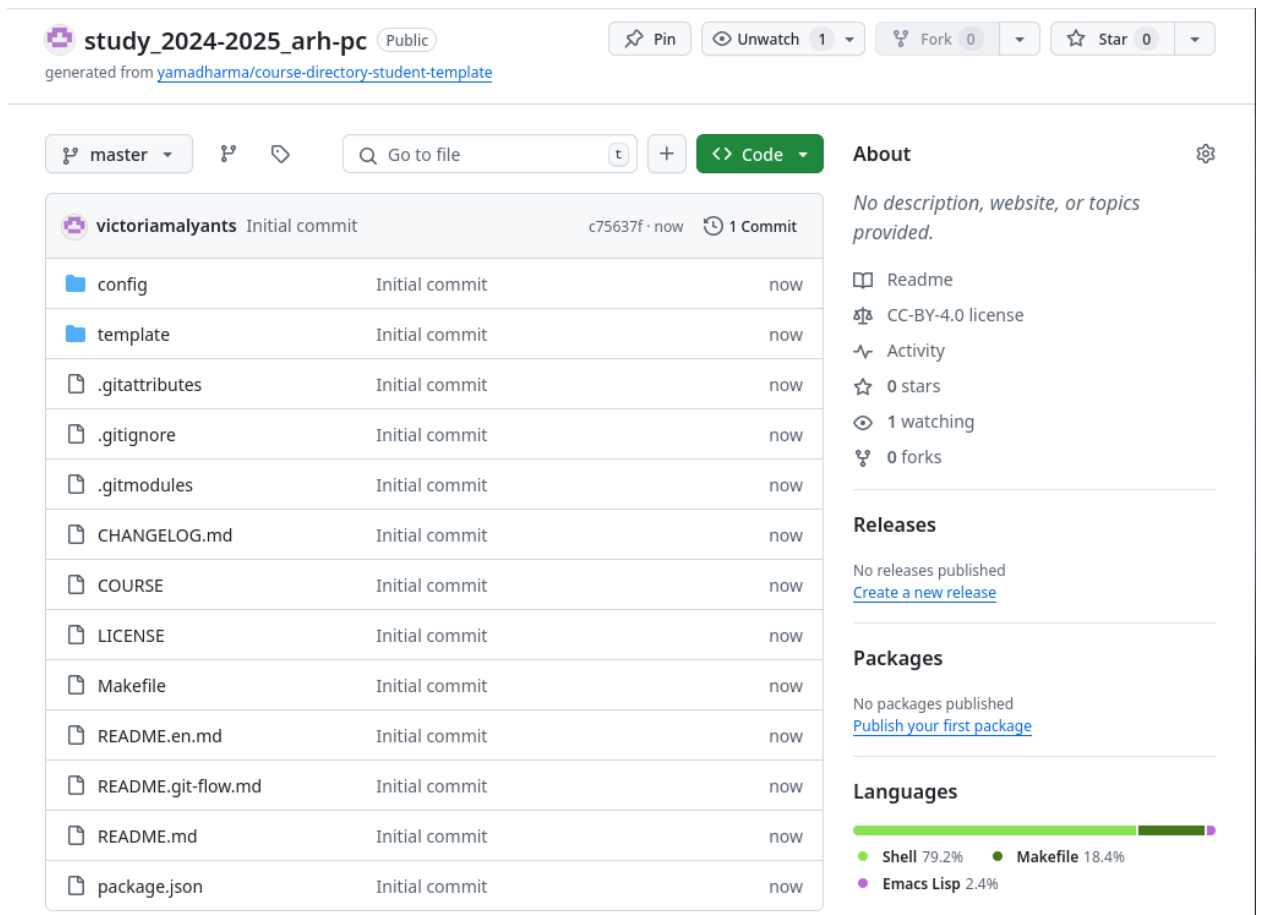


Рис. 4.5.2 Созданный репозиторий

Перехожу в каталог курса с помощью команды `cd` (рис. 4.5.3).

```
vkmaljyanc@localhost-live:~$ cd ~/work/study/2024-2025/"Архитектура компьютера"
```

Рис. 4.5.3Переход в каталог курса

Клонирую созданный репозиторий (рис. 4.5.4).

```

vkmaljyanc@localhost-live:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive g
it@github.com:victoriamaalyants/study_2024-2025_arh-pc.git arch-pc
Cloning into 'arch-pc'...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdK4UvC0Qu.
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: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 33 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (33/33), 18.82 KiB | 4.70 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-
template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.g
it) registered for path 'template/report'
Cloning into '/home/vkmaljyanc/work/study/2024-2025/Архитектура компьютера/arch-pc/template/pres
entation'...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (111/111), 102.17 KiB | 486.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/vkmaljyanc/work/study/2024-2025/Архитектура компьютера/arch-pc/template/repo
rt'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Receiving objects: 100% (142/142), 341.09 KiB | 255.00 KiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'

```

Рис. 4.5.4 Клонирование репозитория

Ссылку для клонирования копирую на странице созданного репозитория (рис. 4.5.5).

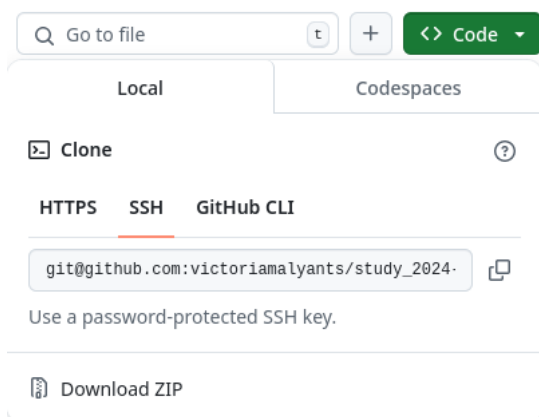


Рис. 4.5.5 Ссылка для копирования репозитория

4.6. Настройка каталога курса

Перехожу в каталог курса (рис. 4.6.1).

```

vkmaljyanc@localhost-live:~/work/study/2024-2025/Архитектура компьютера$ cd ~/work/study/2024-20
25/"Архитектура компьютера"/arch-pc

```

Рис. 4.6.1 Перемещение в каталог курса

Удаляю лишние файлы (рис. 4.6.2).

```
vkmaljyanc@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
```

Рис. 4.6.2 Удаление файлов

Создаю необходимые каталоги (рис. 4.6.3).

```
vkmaljyanc@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
vkmaljyanc@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
```

Рис. 4.6.3 Создание каталогов

Отправляю файлы на сервер (рис. 4.6.4, рис. 4.6.5).

```
vkmaljyanc@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
vkmaljyanc@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master 069a662] feat(main): make course structure
221 files changed, 53680 insertions(+)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/.texlabroot
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/.projectile
create mode 100644 labs/lab02/presentation/.texlabroot
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
```

Рис. 4.6.4 Добавление и сохранение изменений на сервере

```
vkmaljyanc@localhost-live:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 36, done.
Counting objects: 100% (36/36), done.
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 341.39 KiB | 464.00 KiB/s, done.
Total 35 (delta 4), reused 1 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:victoriamaalyants/study_2024-2025_arh-pc.git
0303fda..069a662 master -> master
```

Рис. 4.6.5 Выгрузка изменений на сервер

Убеждаюсь в правильности создания иерархии рабочего пространства (рис. 4.6.6).

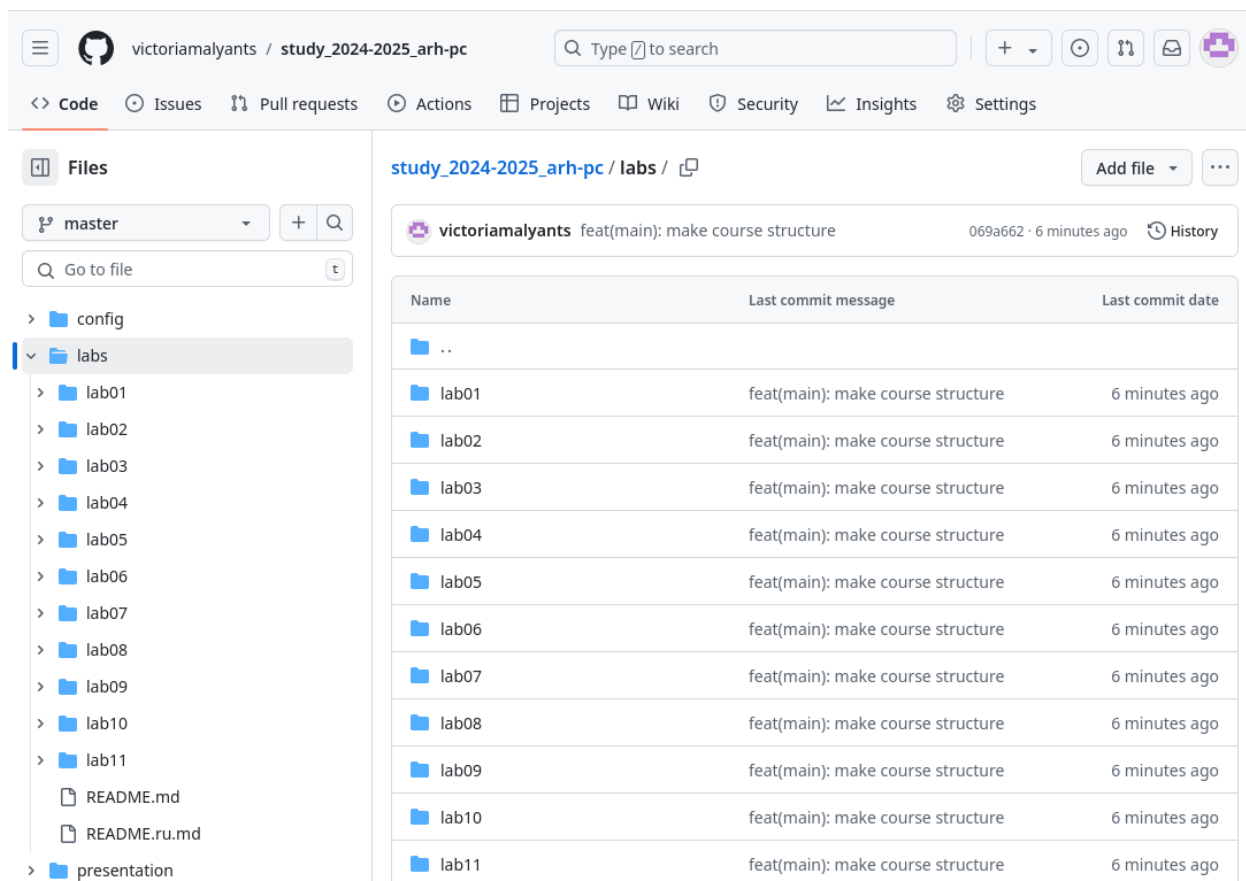


Рис. 4.6.6 Репозиторий

4.7 Выполнение заданий для самостоятельной работы (рис. 4.7)

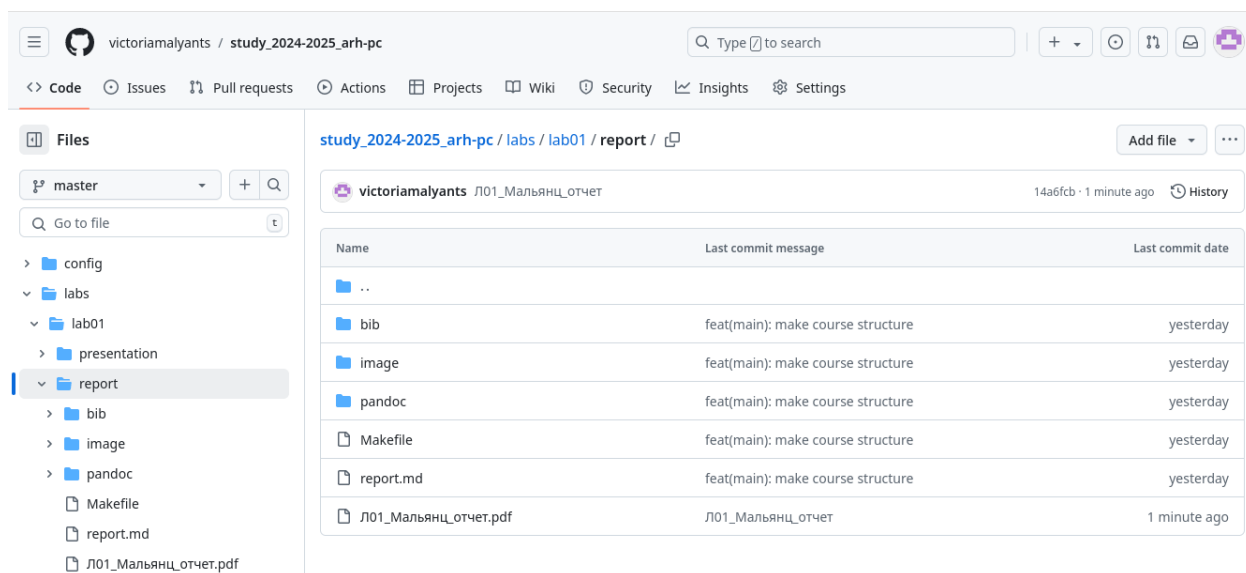


Рис. 4.7.1 Добавление отчета по Лабораторной работе № 1

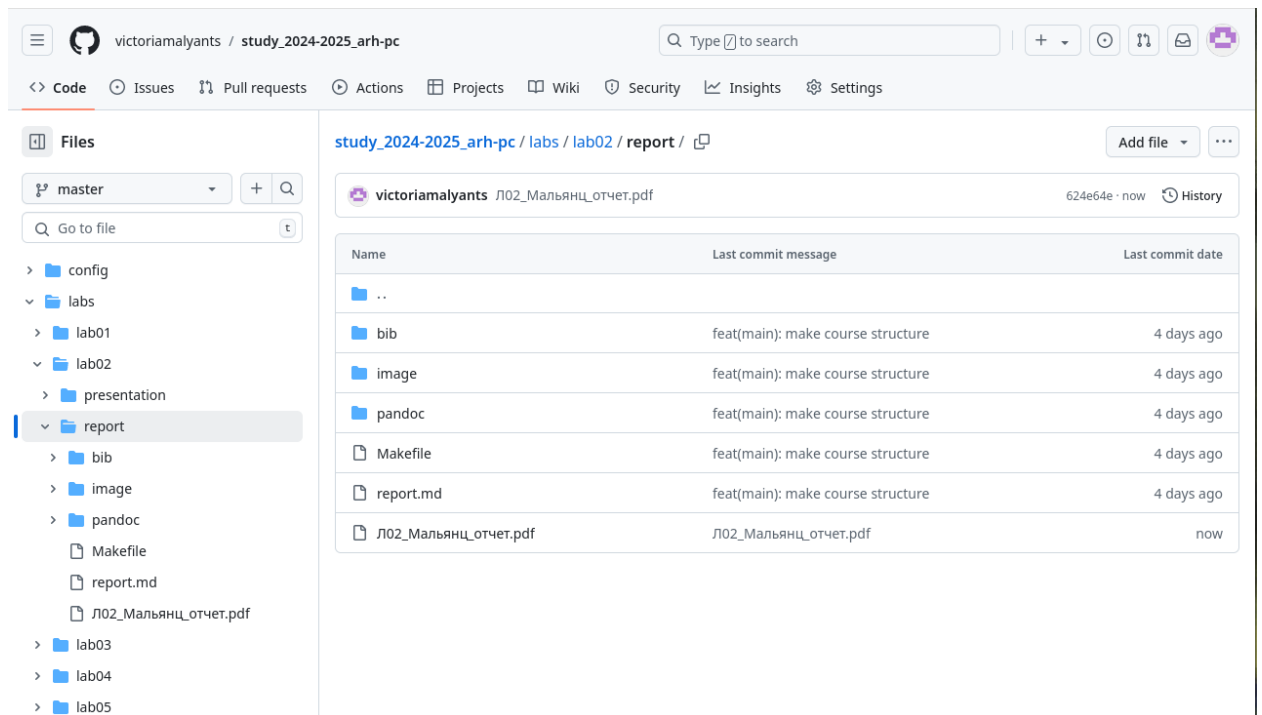


Рис. 4.7.2 Добавление отчета по лабораторной работе № 2

5 Вывод

Я изучила идеологию и применение средств контроля версий и приобрела практические навыки по работе с системой git.

Список литературы

1. GDB: The GNU Project Debugger. — URL:
<https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL:
<https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL:
<https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL:
<https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL:
<https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL:
<https://www.opennet.ru/docs/RUS/nasm/>.

13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.

14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.

15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).

16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).