

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: Венис Дмитрий Сергеевич

Группа: НКАбд-01-25

МОСКВА

20_25_____ г.

Содержание

- 1.Цель работы
- 2.Теоретическое введение
- 3.Выполнение лабораторной работы
- 4.Выполнение самостоятельной работы
- 5.Выводы

Список иллюстраций

Рис 3.1.1.....	6
Рис.3.2.1.....	6
Рис 3.2.2.....	7
Рис 3.2.3.....	7
Рис 3.3.1.....	8
Рис 3.3.2.....	8
Рис 3.4.1.....	9
Рис 3.4.2.....	9
Рис 3.4.3.....	10
Рис 4.1.1.....	11
Рис 4.2.1.....	11

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

2.1. Системы контроля версий.

Общие понятия 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.

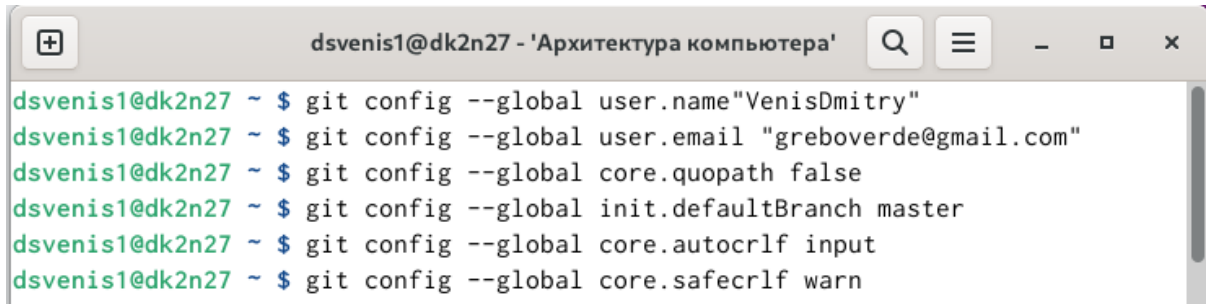
2.2. Система контроля версий Git

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.

3.Выполнение лабораторной работы

3.1

Сначала сделаем предварительную конфигурацию git. Откроем терминал и введём следующие команды, указав имя и e-mail владельца репозитория, настроим utf-8 в выводе сообщений git, зададим имя начальной ветки (будем называть её master), параметр autocrlf, параметр safecrlf:



```
dsvenis1@dk2n27 - 'Архитектура компьютера'
dsvenis1@dk2n27 ~ $ git config --global user.name "VenisDmitry"
dsvenis1@dk2n27 ~ $ git config --global user.email "greboverde@gmail.com"
dsvenis1@dk2n27 ~ $ git config --global core.quopath false
dsvenis1@dk2n27 ~ $ git config --global init.defaultBranch master
dsvenis1@dk2n27 ~ $ git config --global core.autocrlf input
dsvenis1@dk2n27 ~ $ git config --global core.safecrlf warn
```

Рис 3.1.1

3.2.

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый), сделаем это с помощью команды:

`ssh-keygen -C "VenisDmitry<greboverde@gmail.com>"`

```
dsvenis1@dk2n27 ~ $ ssh-keygen -C "VenisDmitry<greboverde@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/d/s/dsvenis1/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/d/s/dsvenis1/.ssh/id_ed25519
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/d/s/dsvenis1/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:y0ngwD83nNpimW15x2zks1YwdIS+KuCBXQfGTSfww1s VenisDmitry<greboverde@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
|      ..+o .o.    |
|      +o.o. .     |
|      . .+oE.     |
|      . .o.+o     |
|    oo..+.S  o    |
|  .+++  .. +     |
|  .+B*= +        |
|    X=o.X        |
|    o.o.=.o      |
+----[SHA256]-----+
```

Рис.3.2.1

Далее ввожу команду:

`cat /afs/.dk.sci.pfu.edu.ru/home/d/s/dsvenis1/.ssh/id_ed25519.pub`

для того, чтобы терминал показал мне мой открытый ключ для идентификации пользователя на сервере репозитория, также необходимо загрузить сгенерированный открытый ключ. Для этого зайдя на сайт <http://github.org/> под своей учётной записью и перейду в меню Setting . После этого выберу в боковом меню SSH and GPG keys и нажму кнопку New SSH key . Копируем из локальной консоли ключ в буфер обмена. Вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title).

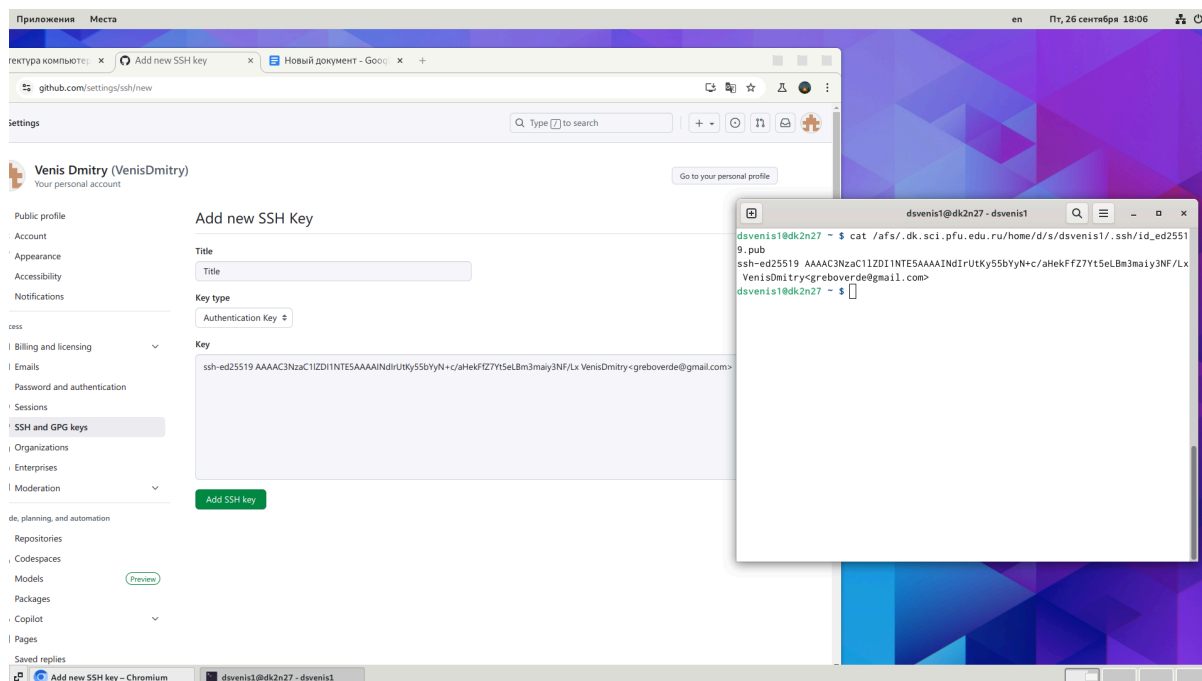


Рис 3.2.2

На рисунке 3.2.3 представлен снимок экрана с ключом который появился вследствие действий перечисленных выше

SSH keys

New SSH key

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

Authentication keys


	Title	
	SHA256:yOngwD83nNpimW15x2zks1YhdIS+KuCBXQfGTSfw1s Added on Sep 26, 2025 Never used — Read/write	<div>Delete</div>

Рис 3.2.3

3.3

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

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

и перейдем в только что созданный каталог курса

```
cd ~/work/study/2025-2026/"Архитектура компьютера"
```

```
dsvenis1@dk2n27 ~ $ mkdir -p ~/work/study/2025-2026/"Архитектура компьютера"
dsvenis1@dk2n27 ~ $ cd ~/work/study/2025-2026/"Архитектура компьютера"
```

Рис 3.3.1

Клонируем созданный репозиторий

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

```
dsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера $ git clone --recursive git@github.com:VenisDmitry/-study_2025-2026_arh-pc.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3vVV6TuJJhbzF/zLDA0zPMSvHdkr4lvC0qU.
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: 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 | 558.00 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/d/s/dsvenis1/work/study/2025-2026/Архитектура компьютера/arch-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 | 7.49 МиБ/с, готово.
Определение изменений: 100% (60/60), готово.
Клонирование в «/afs/.dk.sci.pfu.edu.ru/home/d/s/dsvenis1/work/study/2025-2026/Архитектура компьютера/arch-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 | 2.09 МиБ/с, готово.
Определение изменений: 100% (98/98), готово.
Submodule path 'template/presentation': checked out '6efd5c4ee78e4456caff3dc7062cfcad26058ca6'
Submodule path 'template/report': checked out '89a9622199b4df88227b9b3fa3d4714c85f68dd2'
dsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера $
```

Рис 3.3.2

3.4

Перейдём в каталог курса:

```
cd ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc
```

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
```

```

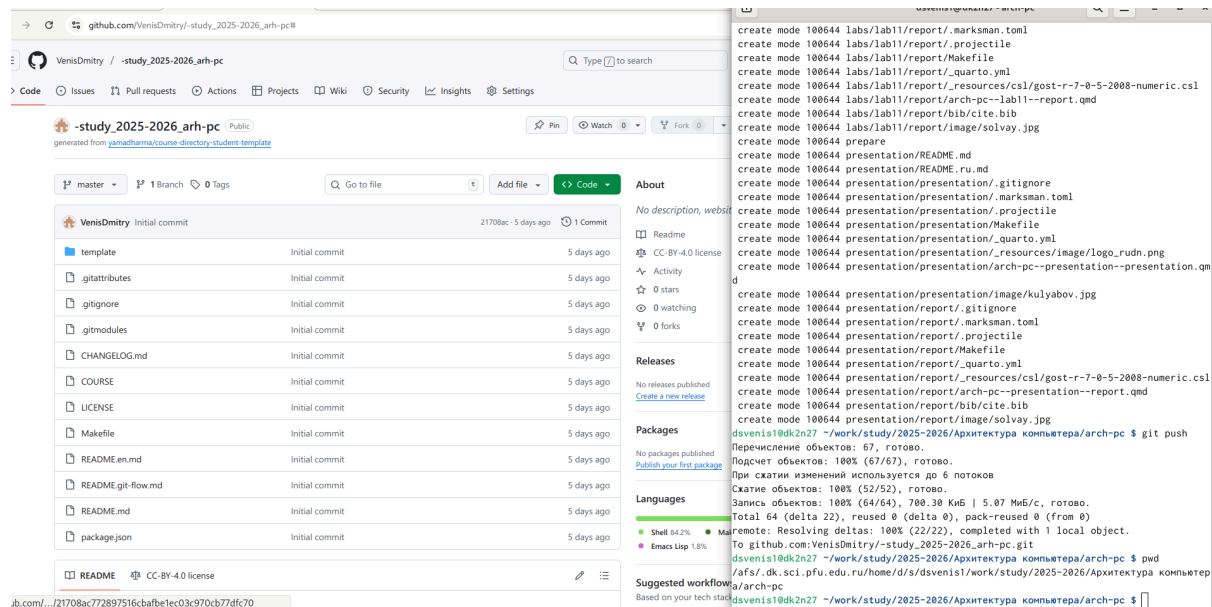
create mode 100644 labs/lab11/presentation/_resources/image/logo_rudn.png
create mode 100644 labs/lab11/presentation/arch-pc--lab11--presentation.qmd
create mode 100644 labs/lab11/presentation/image/kulyabov.jpg
create mode 100644 labs/lab11/report/.gitignore
create mode 100644 labs/lab11/report/.marksman.toml
create mode 100644 labs/lab11/report/.projectile
create mode 100644 labs/lab11/report/Makefile
create mode 100644 labs/lab11/report/_quarto.yml
create mode 100644 labs/lab11/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab11/report/arch-pc--lab11--report.qmd
create mode 100644 labs/lab11/report/bib/cite.bib
create mode 100644 labs/lab11/report/image/solvay.jpg
create mode 100644 prepare
create mode 100644 presentation/README.md
create mode 100644 presentation/README.ru.md
create mode 100644 presentation/presentation/.gitignore
create mode 100644 presentation/presentation/.marksman.toml
create mode 100644 presentation/presentation/.projectile
create mode 100644 presentation/presentation/Makefile
create mode 100644 presentation/presentation/_quarto.yml
create mode 100644 presentation/presentation/_resources/image/logo_rudn.png
create mode 100644 presentation/presentation/arch-pc--presentation--presentation.qmd
create mode 100644 presentation/presentation/image/kulyabov.jpg
create mode 100644 presentation/report/.gitignore
create mode 100644 presentation/report/.marksman.toml
create mode 100644 presentation/report/.projectile
create mode 100644 presentation/report/Makefile
create mode 100644 presentation/report/_quarto.yml
create mode 100644 presentation/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 presentation/report/arch-pc--presentation--report.qmd
create mode 100644 presentation/report/bib/cite.bib
create mode 100644 presentation/report/image/solvay.jpg
dsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 67, готово.
Подсчет объектов: 100% (67/67), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (52/52), готово.
Запись объектов: 100% (64/64), 700.30 КиБ | 5.07 МБ/с, готово.
Total 64 (delta 22), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (22/22), completed with 1 local object.
To github.com:VenisDmitry/-study-2025-2026_arh-pc.git
21708ac..cb8b316 master -> master
dsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ █

dsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера $ cd ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc
dsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ ecdsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ echo arch-pc >
dsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ make prepare
git add dsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ git add .
dsvenis1@dk2n27 ~/work/study/2025-2026/Архитектура компьютера/arch-pc $ git commit -am 'feat(main): make course structure'
[master cb8b316] 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
create mode 100644 labs/lab02/presentation/_resources/image/logo_rudn.png
create mode 100644 labs/lab02/presentation/arch-pc--lab02--presentation.qmd
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
create mode 100644 labs/lab02/report/.gitignore
create mode 100644 labs/lab02/report/.marksman.toml
create mode 100644 labs/lab02/report/.projectile
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/_quarto.yml
create mode 100644 labs/lab02/report/_resources/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab02/report/arch-pc--lab02--report.qmd
create mode 100644 labs/lab02/report/bib/cite.bib

```

Рис 3.4.1-3.4.2

Проверим правильность создания иерархии рабочего пространства в локальном репозитории и на странице github



Ответ:Всё совпадает

Рис 3.4.3

Вывод:Выполнили задания лабораторной работы и научились работать с новыми командами и VCS Git

4.Выполнение самостоятельной работы

1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs/lab02/report).

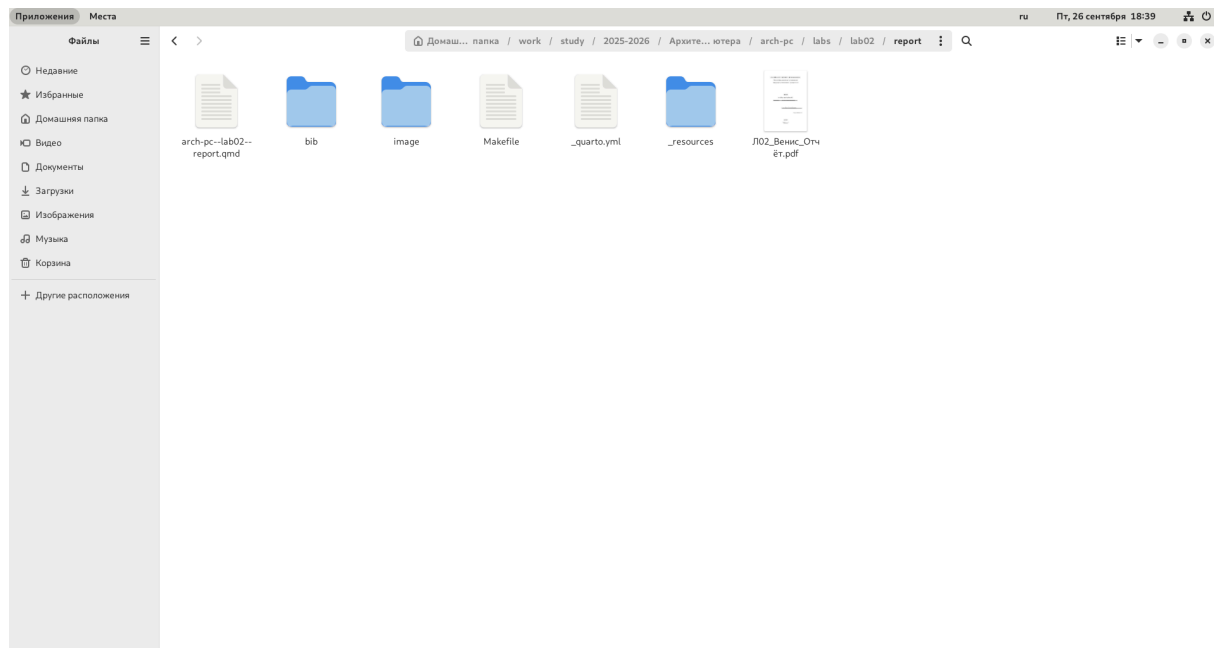


Рис 4.1.1

Ответ:4.1.1

2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.

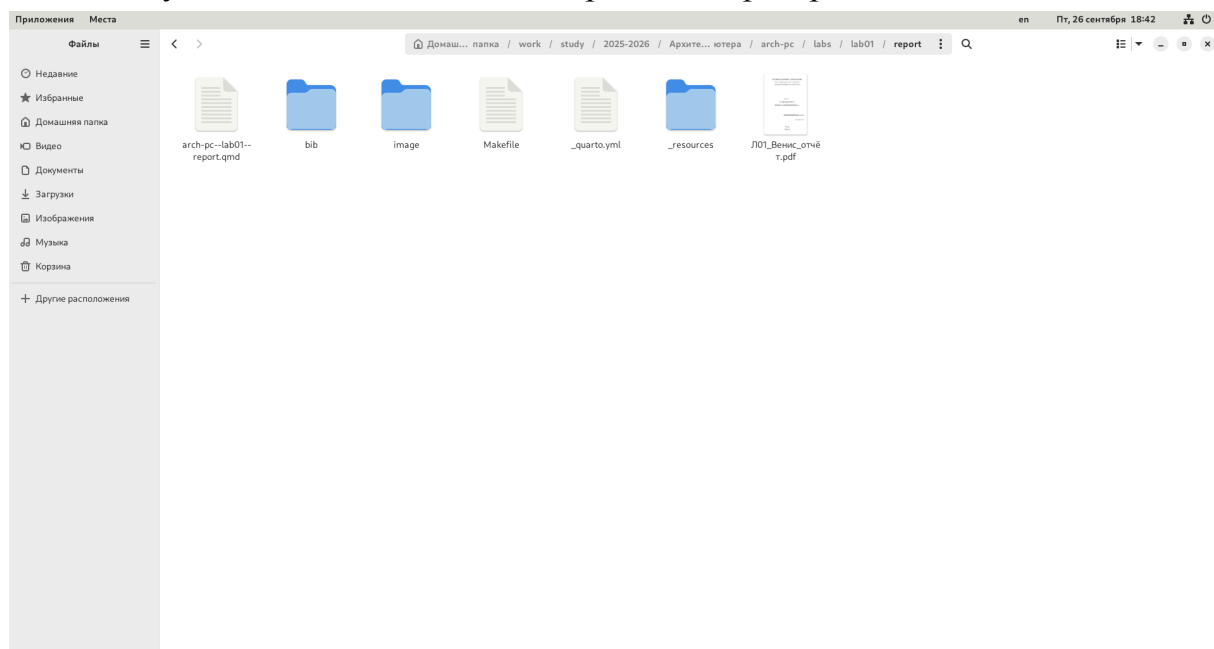


Рис 4.2.1

3.Выполнил

5.Выводы

Я изучил идеологию и применение средств контроля версий, приобрёл практические навыки по работе с системой контроля версий git.