

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

**Кафедра фундаментальная информатика и
информационные технологии**

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Архитектура компьютеров

Студент: Волобуев И. А.

Группа: НБИбд-01-23

МОСКВА

2023 г.

Содержание

1 Цель работы.....	3
2 Задание.....	4
3 Теоретическое введение.....	5
4 Выполнение лабораторной работы	6
5 Выводы	18
6 Источники	19

1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой 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 курс на основе шаблона.
6. Настройка каталога курса.
7. Задания 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. 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.

4 Выполнение лабораторной работы

1. Настройка Github

Создаю учётную запись на сайте Github и заполняю основные данные (рис. 1)

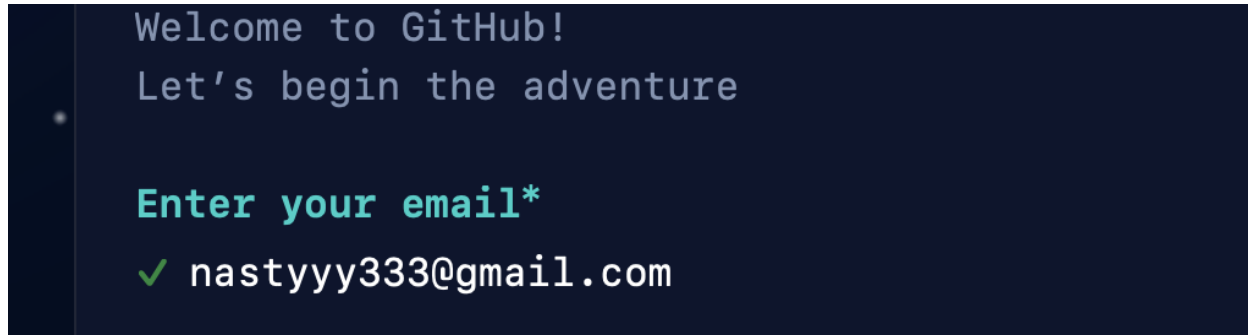


Рис. 1. Заполнение данных учётной записи Github

2. Базовая настройка Git

Сначала делаю предварительную конфигурацию Git. Открываю терминал и ввожу команды указав имя и email. (рис. 2).

```
iavolobuev@iavolobuev:~$ git config --global user.name "<iavolobuev>"
iavolobuev@iavolobuev:~$ git config --global user.email "<nastyyy333@gmail.com>"
iavolobuev@iavolobuev:~$
```

Рис. 2. Предварительная конфигурация Git

Настраиваю utf-8 в выводе сообщений Git (рис.13).

```
iavolobuev@iavolobuev:~$ git config --global core.quotepath false
```

Рис. 3. Настройка кодировки

Задаю имя 'master' для начальной ветки(рис. 4).

```
iavolobuev@iavolobuev:~$ git config --global init.defaultBranch master
```

Рис. 4. Создание имени для начальной ветки

Задаю параметр autocrlf (рис. 5).

```
iavolobuev@iavolobuev:~$ git config --global core.autocrlf input
```

Рис. 5. Параметр autocrlf

Задаю параметр safecrlf (рис. 6).

```
iavolobuev@iavolobuev:~$ git config --global core.safecrlf warn
```

Рис. 6. Параметр safecrlf

3. Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого использую команду `ssh-keygen -C 'Имя пользователя , <work@mail>'`. Ключ автоматически сохранится в каталоге `~/.ssh/` (рис. 7).

```
iavolobuev@iavolobuev:~$ ssh-keygen -C "iavolobuev <nastyyy333@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/iavolobuev/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/iavolobuev/.ssh/id_rsa
Your public key has been saved in /home/iavolobuev/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:0YSwwb743f2Tky/fnySyuw8i4eucfJ+Ythjy+7lFles iavolobuev <nastyyy333@gmail.com>
The key's randomart image is:
+---[RSA 3072]-----+
|      .o.  ..      |
|      .o.o   .     |
|      .. . . o     |
|      . . . . .    |
|      .oS  . .     |
|      . o . . .     |
|      o = o = E +   |
|      * BoB * O o   |
|      .X=B+*+o.B*   |
+-----[SHA256]-----+
```

Рис. 7. Генерация SSH ключа

Читаю содержимое файла при помощи команды cat и копирую(рис. 8).

```
iavolobuev@iavolobuev:~$ cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDnZ3Pj3+bS2dUXzRg6v7T5SrUHzaK9T1UXwX4GQo0FgY6+OxNkUnS7sa2nVWeB0tpnXy  
p236k/PLXAFSfHzrz9KPY+3TmuzbpQ4FAryGfgZBGZmWAIMqakDwF1Y9DKdlGI4suB/11eSaBwnW9jA0gSh1LVkb80aR0rNEUoJHvQvEPZ  
tr7v5vfioT3xxA38i1wzxN/70N+E04ofk0up+ugxJ/TvJkb4Cv0JS+9rvkiTbGkS3WuK0GxWuWGl6l5g1/0SjE0IYRzbaTzABntwI+TabZ  
fEMrQXtZ1TvhzgTTFgTTvGN2x9rxaFkax0V+9BNHScqMl00LL1NR2Y0z+N9rskjCLooryEr4DGyPkJV7tdvTBAw9hsiogDoYP/s6Cvtg/L  
Yrn6y23+mP/LLV7Y8o9Y4X5e02WskTaiExeuKeGQjiDQuA0yG8koe3Ixn7TAGCNpu+xxBDs2f0BsLn1Kbm1smCyohCtXiu99K/7BK4MCIG  
tyuBaDWCiikygyoyVxgsc= iavolobuev <nasty333@gmail.com>
```

Рис. 8. Чтение файла

Открываю браузер, захожу на сайт Github. Открываю свой профиль и выбираю страницу “SSH and GPG keys”. Нажимаю кнопку «New SSH key». Вставляю скопированный ключ в поле «Key». В поле «Title» указываю имя для ключа. Нажимаю «Add SSH key», чтобы завершить добавление ключа (рис. 9).

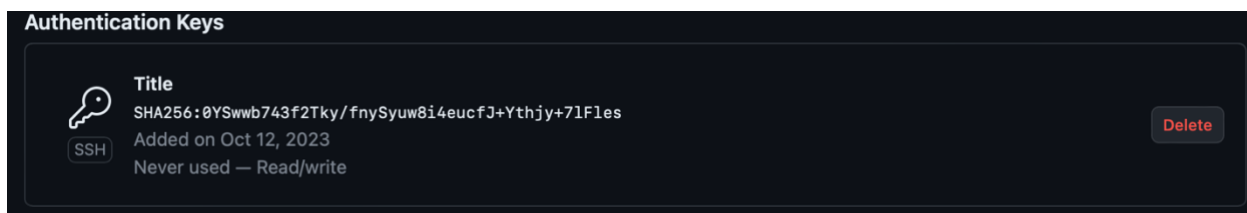


Рис. 9. Добавление ключа

4. Создание рабочего пространства и репозитория курса на основе шаблона

Перехожу в терминал. Создаю директорию, рабочее пространство, с помощью утилиты mkdir, далее при помощи ключа -p создаю все директории после домашней. (рис. 10).


```
iavolobuev@iavolobuev:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
```

Рис. 10. Создание рабочего пространства

5. Создание репозитория курса на основе шаблона

Открываю браузер и перехожу на страницу репозитория с шаблоном курса. Далее выбираю «Use this template», чтобы использовать этот шаблон (рис. 11).

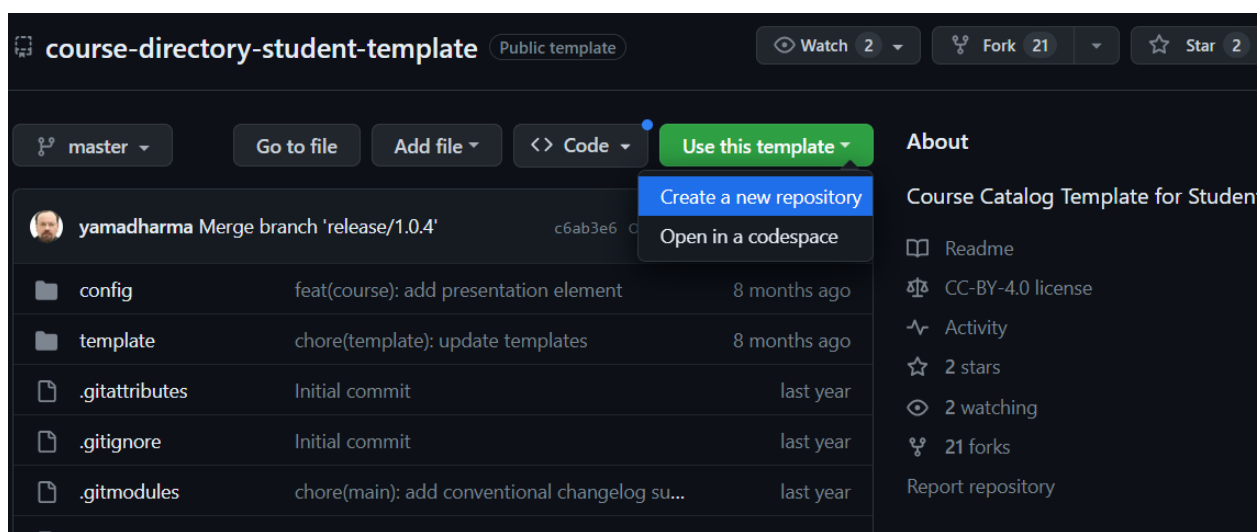


Рис. 11. Страница шаблона для репозитория

Потом задаю имена репозитория (study_2023-2024_arch-rc) и создаю репозиторий (рис. 12)

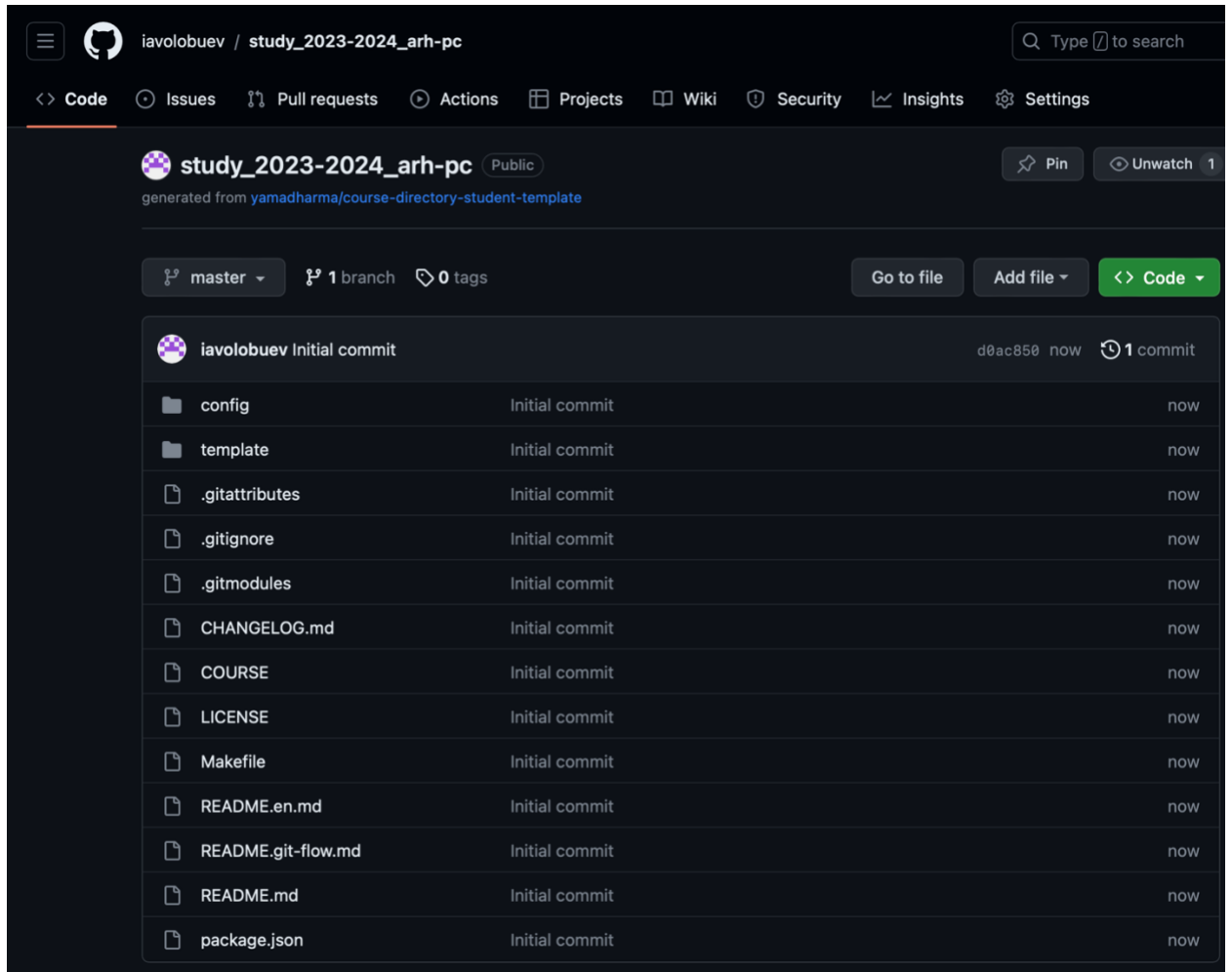


Рис. 12. Созданный репозиторий

Через терминал перехожу в созданный каталог курса с помощью утилиты cd(рис. 13).

```
iavolobuev@iavolobuev:~$ cd ~/work/study/2023-2024/"Архитектура компьютера"
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера$
```

Рис. 13. Перемещение между директориями

Копирую ссылку для клонирования на странице созданного репозитория, нажав на «Code», потом на «SSH»(рис. 14).

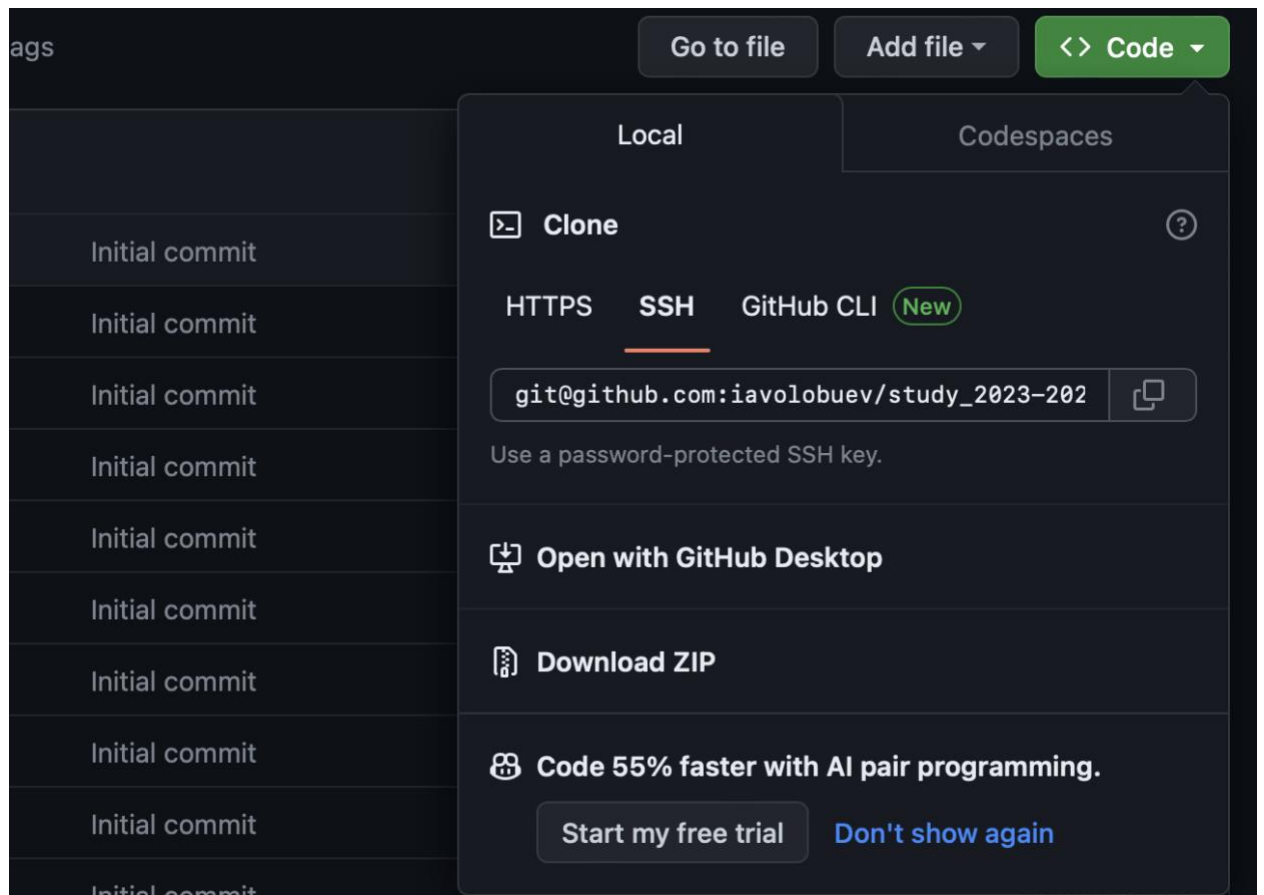


Рис. 14. Окно с ссылкой для копирования репозитория

Клонирую созданный репозиторий с помощью команды `git clone --recursive`(рис.15).

```
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера$ git clone --recursive git@github.com:iavolobuev/study_2023-2024_arch-pc.git arch-pc
Клонирование в «arch-pc»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.93 КиБ | 8.47 МиБ/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/iavolobuev/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 КиБ | 1.13 МиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/iavolobuev/work/study/2023-2024/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 КиБ | 692.00 КиБ/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
```

Рис. 15. Клонирование репозитория

6. Настройка каталога курса

Перехожу в каталог arch-pc при помощи утилиты cd(рис. 16).

```
iavolobuev@iavolobuev:~$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 16. Перемещение между директориями

Удаляю лишние файлы при помощи утилиты rm(рис. 17).

```
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ rm package.json
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 17. Удаление файлов

Создаю необходимые каталоги(рис. 18).

```
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ make
```

Рис. 18. Создание каталогов

Отправляю файлы на сервер. При помощи команды `git add` добавляю все созданные каталоги и комментирую при помощи команды `git commit`(рис. 19).

```
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master b39d269] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
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/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
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_fignos.py
```

Рис. 19. Добавление и сохранение изменений на сервере

Отправляю на сервер с помощью команды `git push`(рис. 20).

```
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.63 МБ/с, готово.
Всего 35 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:iavolobuev/study_2023-2024_arh-pc.git
 d0ac850..b39d269 master -> master
```

Рис. 20. Выгрузка изменений на сервер

Проверяю правильность выполнения на сайте Github(рис. 21).

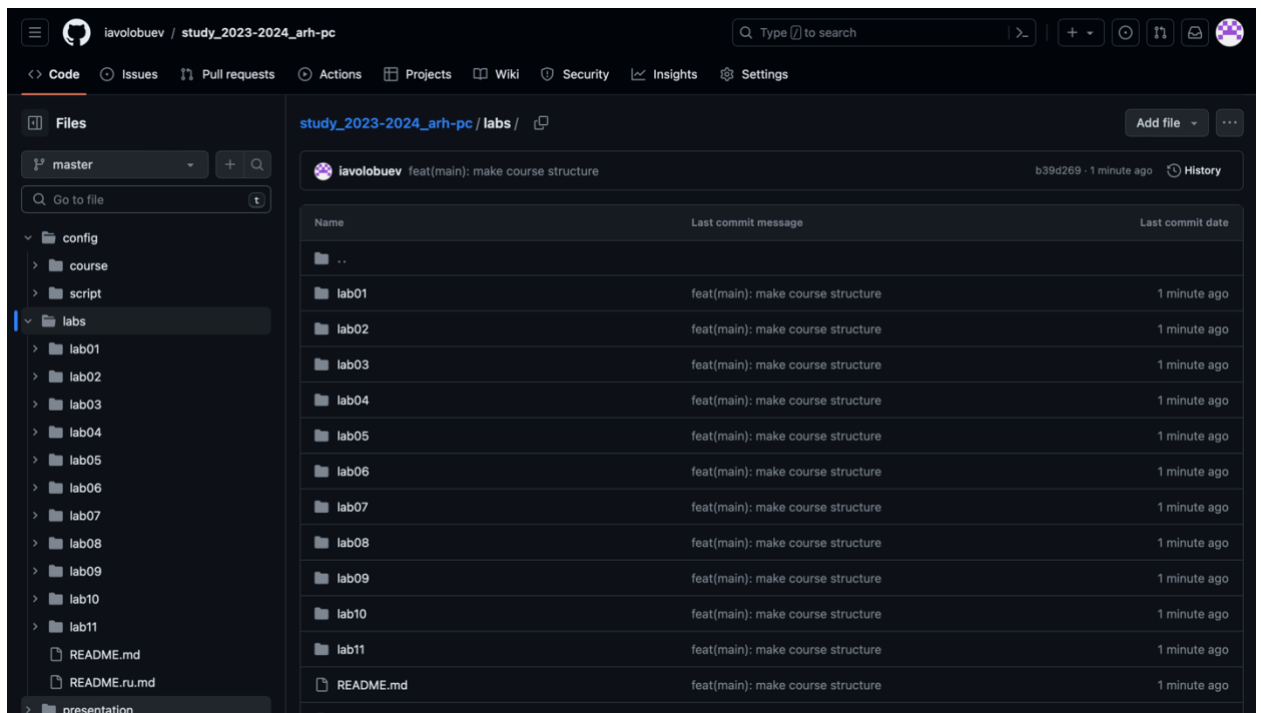


Рис. 21. Страница репозитория

7.Выполнение заданий для самостоятельной работы

Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчёта по второй лабораторной работе с помощью утилиты touch (рис.22).

```
iavolobuev@iavolobuev: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ touch L02_iavolobuev_Report
```

Рис.22. Создание файла

Оформить отчёт я могу в текстовом редакторе LibreOffice Writer(рис.23).

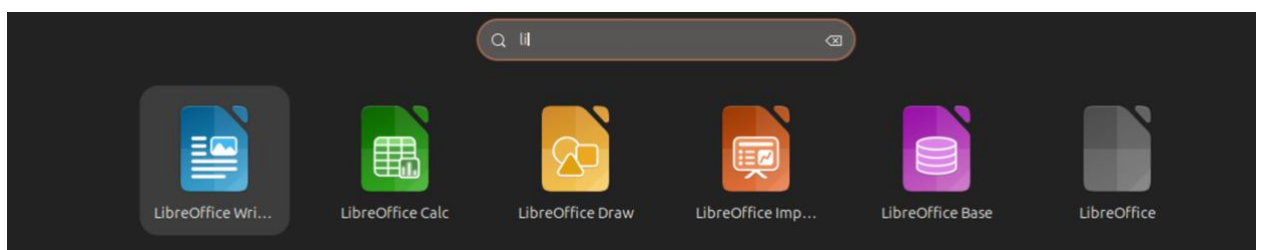


Рис. 23. Меню приложений

Открываю в текстовом редакторе файл , где буду делать отчёт(рис.24).

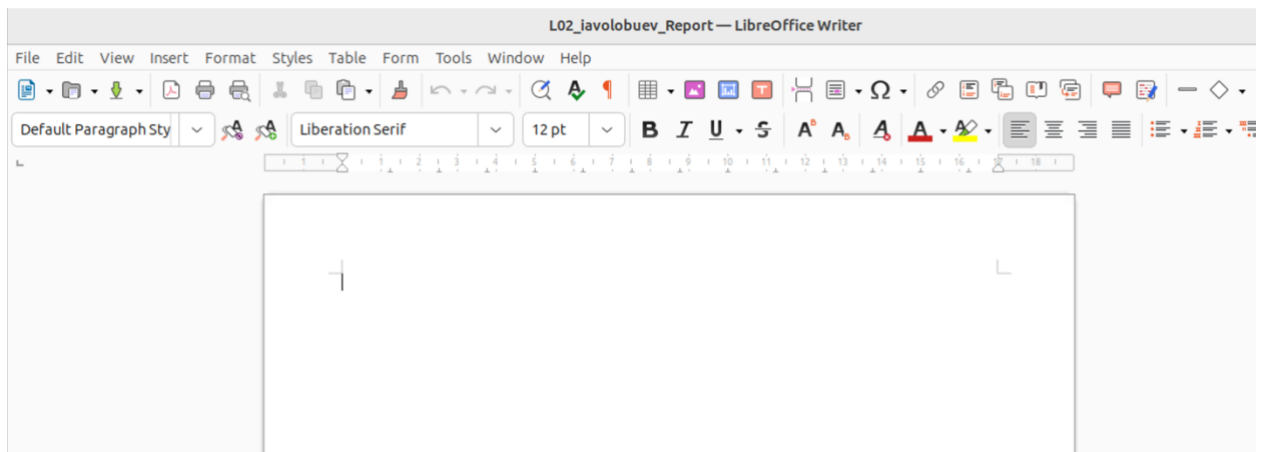


Рис. 24. Текстовый редактор

Перехожу в подкаталог lab01/report с помощью утилиты cd(рис.25).

```
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис. 25. Перемещение между директориями

Проверяю где находится файл с отчётом по первой лабораторной работе(рис.26).

```
iavolobuev@iavolobuev:~/Загрузки$ ls
'Telegram Desktop' 'Волобуев НБИбд-01-23.pdf'
iavolobuev@iavolobuev:~/Загрузки$
```

Рис.26. Проверка нахождения файла

Копирую первую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды(рис.27.).

```
iavolobuev@iavolobuev:~/Загрузки$ cp ~/Загрузки/'Волобуев НБИбд-01-23.pdf' /home/iavolobuev/work/study/2023-2024/'Архитектура комп
ьютера'/arch-pc/labs/lab01/report
iavolobuev@iavolobuev:~/Загрузки$ cd
iavolobuev@iavolobuev:~$ cd ~/work/study/2023-2024/'Архитектура компьютера'/arch-pc/labs/lab01/report
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ ls
bib image Makefile pandoc report.md 'Волобуев НБИбд-01-23.pdf'
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис.27. Копирование файла

Добавляю файл «Волобуев НБИбд-01-23» при помощи команды git add и сохраняю изменения при помощи git commit (рис.28,29).

```
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ git add 'Волобуев НБИбд-01-23.pdf'
```

```
iavolobuev@iavolobuev:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ git commit -m "Add existing file s"
[master 83f6ed1] Add existing files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Волобуев НБИБд-01-23.pdf
```

Рис.28,29. Добавление и сохранения файла

Проверяю правильность выполнения задания(рис.30)

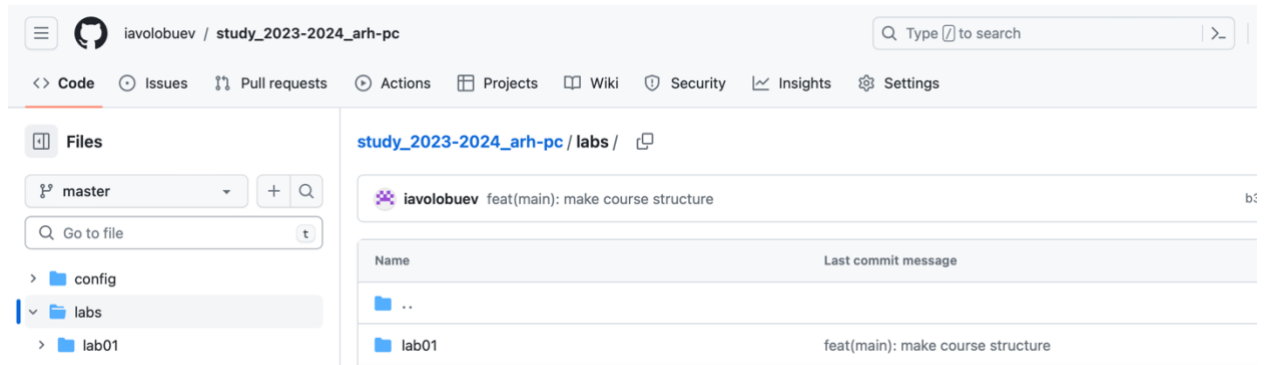


Рис.30.Страница каталога в репозитории.

5 Выводы

При выполнении лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с системой git.

6 Источники

1. [Архитектура ЭВМ \(rudn.ru\)](http://rudn.ru)