

# **Отчет по лабораторной работе № 2**

**Дисциплина: Архитектура компьютера**

Зайцева Ульяна Владимировна

# Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	2. Базовая настройка Git.	10
6	3. Создание SSH-ключа.	11
7	4. Создание рабочего пространства и репозитория курса на основе шаблона.	14
8	5. Создание репозитория курса на основе шаблона.	15
9	6. Настройка каталога курса.	18
10	7. Выполнение заданий для самостоятельной работы.	20
11	Выводы	21

# Список иллюстраций

4.1	Заполнение данных учетной записи GitHub . . . . .	9
5.1	Предварительная конфигурация git и настройка кодировки . . . .	10
6.1	Генерация SSH-ключа . . . . .	11
6.2	Установка утилиты xclip и копирование содержимого файла . . .	12
6.3	Добавление ключа . . . . .	13
7.1	Создание рабочего пространства . . . . .	14
8.1	Страница шаблона для репозитория . . . . .	15
8.2	Окно создания репозитория . . . . .	16
8.3	Созданный репозиторий . . . . .	16
8.4	Клонирование репозитория . . . . .	17
9.1	Перемещение между директориями, удаление лишних файлов, создание каталогов, сохранение изменений . . . . .	18
9.2	Выгрузка изменений на сервер . . . . .	19
9.3	Страница репозитория . . . . .	19
10.1	Каталог lab01/report . . . . .	20

## Список таблиц

# 1 Цель работы

Изучение идеологии и применения 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 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 (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зави-

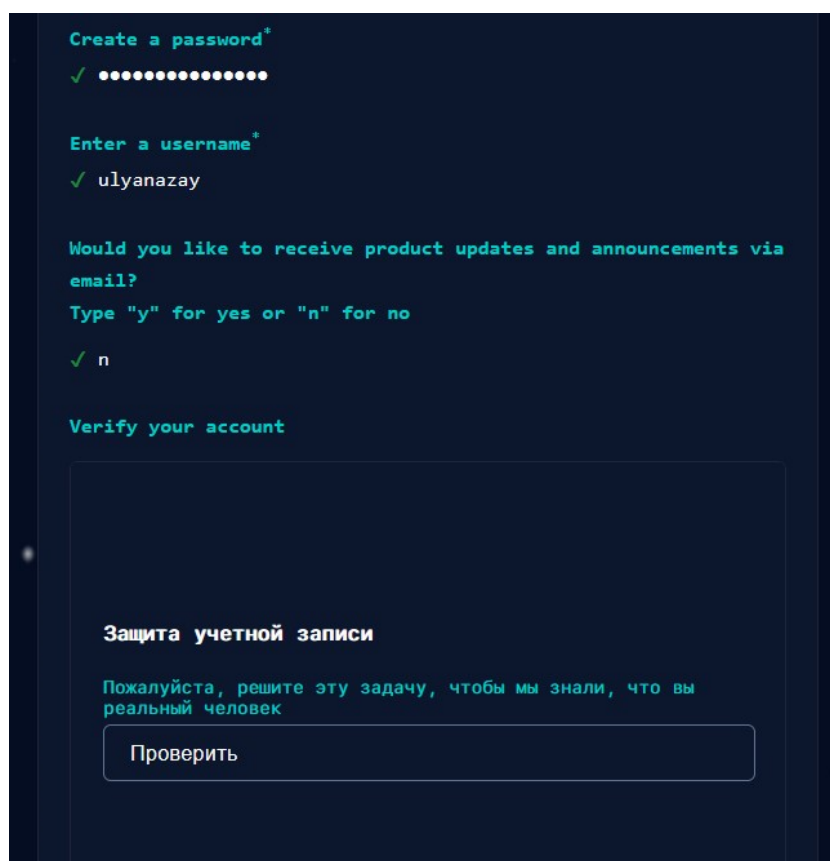
симости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут 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

### #1 . Настройка GitHub

Создаю учётную запись на github и заполняю основные данные(рис. 4.1).

The image shows a dark-themed terminal window with the GitHub account creation process. It includes prompts for creating a password, entering a username (ulyanazay), and a confirmation step titled 'Защита учетной записи' (Account Protection) with a 'Проверить' (Check) button.

Create a password\*

✓ ●●●●●●●●●●

Enter a username\*

✓ ulyanazay

Would you like to receive product updates and announcements via email?

Type "y" for yes or "n" for no

✓ n

Verify your account

**Защита учетной записи**

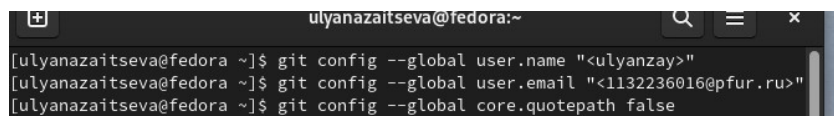
Пожалуйста, решите эту задачу, чтобы мы знали, что вы реальный человек

Проверить

Рис. 4.1: Заполнение данных учетной записи GitHub

## 5 2. Базовая настройка Git.

Открываю терминал и ввожу команды, указывая своё имя и email. Настраиваю utf-8(рис. 5.1)

A screenshot of a terminal window with a dark background. The window title is 'ulyanazaitseva@fedora:~'. It shows three lines of terminal output: the first line is the prompt '[ulyanazaitseva@fedora ~]\$' followed by the command 'git config --global user.name "<ulyanzay>"'; the second line is the prompt followed by 'git config --global user.email "<1132236016@pfur.ru>"'; and the third line is the prompt followed by 'git config --global core.quotepath false'.

```
ulyanazaitseva@fedora:~  
[ulyanazaitseva@fedora ~]$ git config --global user.name "<ulyanzay>"  
[ulyanazaitseva@fedora ~]$ git config --global user.email "<1132236016@pfur.ru>"  
[ulyanazaitseva@fedora ~]$ git config --global core.quotepath false
```

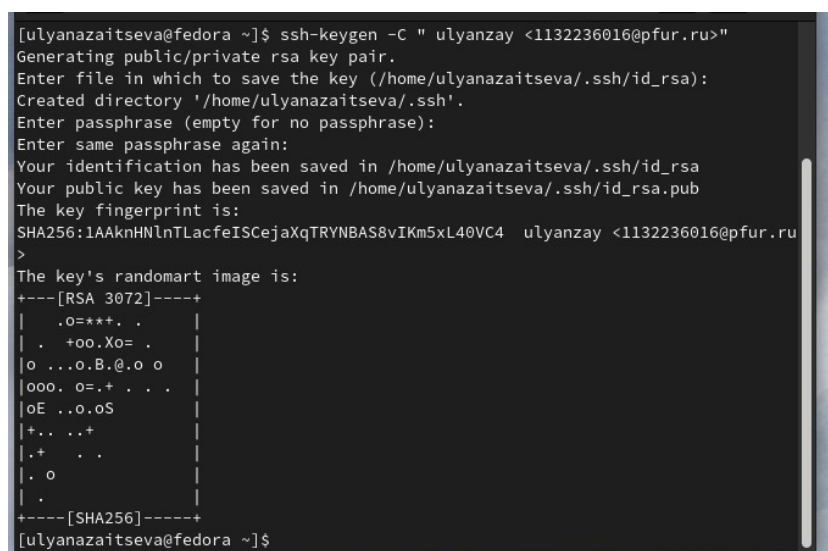
Рис. 5.1: Предварительная конфигурация git и настройка кодировки

Задаю имя начальной ветки(master), использую параметры autocrlf и safecrlf(рис. ??)

![Создание имени для начальной ветки и использование параметров](image/lab33.png{#fig:003 width=70%})

## 6 3. Создание SSH-ключа.

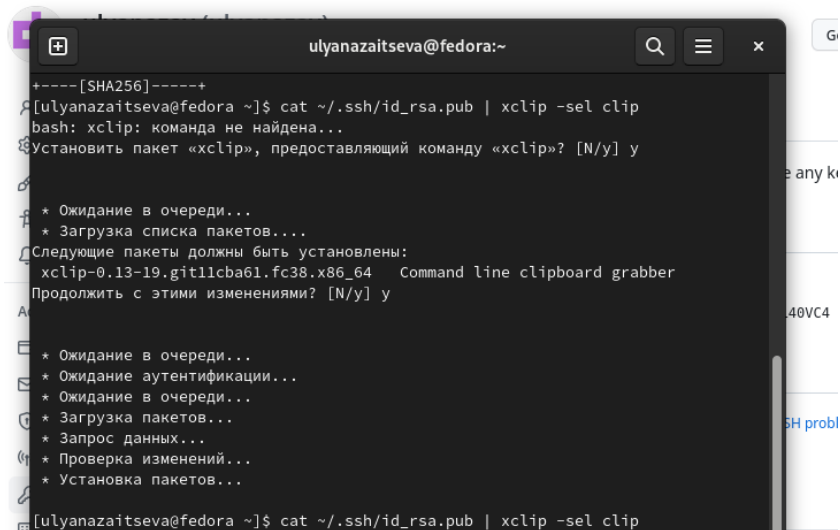
Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду `ssh-keygen -C "Имя Фамилия, work@email"`, указывая имя владельца и электронную почту владельца. (рис. 6.1)



```
[ulyanazaitseva@fedora ~]$ ssh-keygen -C "ulyanzay <1132236016@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ulyanazaitseva/.ssh/id_rsa):
Created directory '/home/ulyanazaitseva/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ulyanazaitseva/.ssh/id_rsa
Your public key has been saved in /home/ulyanazaitseva/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1AAknHnLnTLacfeISCejaXqTRYNBAS8vIKm5xL40VC4 ulyanzay <1132236016@pfur.ru>
>
The key's randomart image is:
+---[RSA 3072]-----+
| .o=***. . |
| . +oo.Xo= . |
|o ...o.B.@.o o |
|ooo. o=.+ . . . |
|oE ..o.oS |
|+.. ..+ |
|. + . . |
|. o |
|. |
+---[SHA256]-----+
[ulyanazaitseva@fedora ~]$
```

Рис. 6.1: Генерация SSH-ключа

Xclip – утилита, которая позволяет скопировать любой текст через терминал. Устанавливаю необходимый пакет для данной утилиты уже при попытке её использовать. После установки заново пишу команду и копирую содержимое файла(сгенерированный ssh-ключ)(рис. 6.2)

A terminal window titled 'ulyanazaitseva@fedora:~' showing the installation of the xclip utility. The user runs 'cat ~/.ssh/id\_rsa.pub | xclip -sel clip'. The terminal shows the command was not found, then the package is installed. It lists the packages to be installed: xclip-0.13-19.git11cba61.fc38.x86\_64, which is a 'Command line clipboard grabber'. The user confirms the installation. The terminal then shows the command being run again: 'cat ~/.ssh/id\_rsa.pub | xclip -sel clip'.

```
ulyanazaitseva@fedora:~  
+----[SHA256]-----+  
[ulyanazaitseva@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip  
bash: xclip: команда не найдена...  
Установить пакет «xclip», предоставляющий команду «xclip»? [N/y] y  
  
* Ожидание в очереди...  
* Загрузка списка пакетов...  
Следующие пакеты должны быть установлены:  
xclip-0.13-19.git11cba61.fc38.x86_64  Command line clipboard grabber  
Продолжить с этими изменениями? [N/y] y  
  
* Ожидание в очереди...  
* Ожидание аутентификации...  
* Ожидание в очереди...  
* Загрузка пакетов...  
* Запрос данных...  
* Проверка изменений...  
* Установка пакетов...  
[ulyanazaitseva@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис. 6.2: Установка утилиты xclip и копирование содержимого файла

Вставляю скопированный ключ в поле Key, задаю имя ключу в поле Title(рис. 6.3)

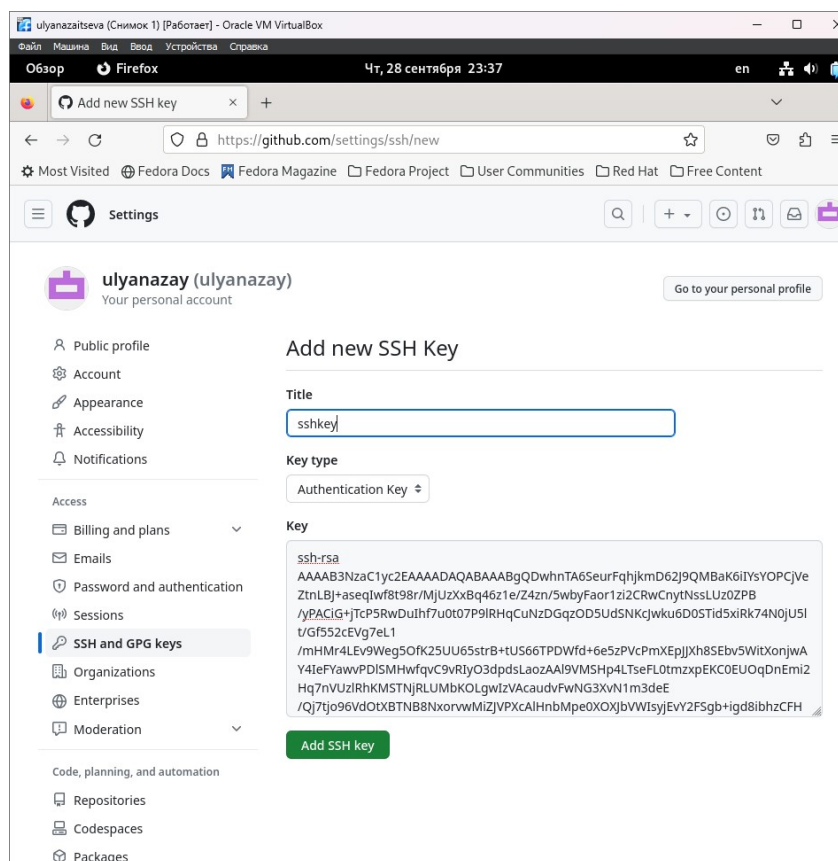


Рис. 6.3: Добавление ключа

## 7 4. Создание рабочего пространства и репозитория курса на основе шаблона.

Возвращаюсь в терминал и создаю рабочее пространство “Архитектура компьютера” с помощью команды `mkdir`. С помощью команды `ls` проверяю выполнение команды(рис. 7.1)

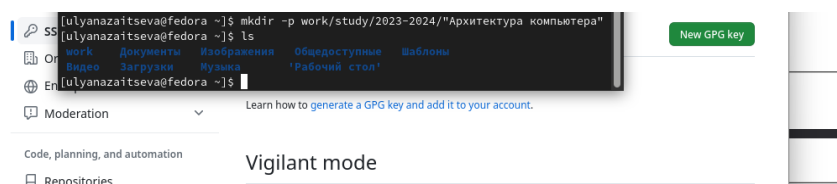


Рис. 7.1: Создание рабочего пространства

## 8 5. Создание репозитория курса на основе шаблона.

В браузере перехожу на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Выбираю «Use this template», чтобы использовать шаблон для своего репозитория(рис. 8.1)

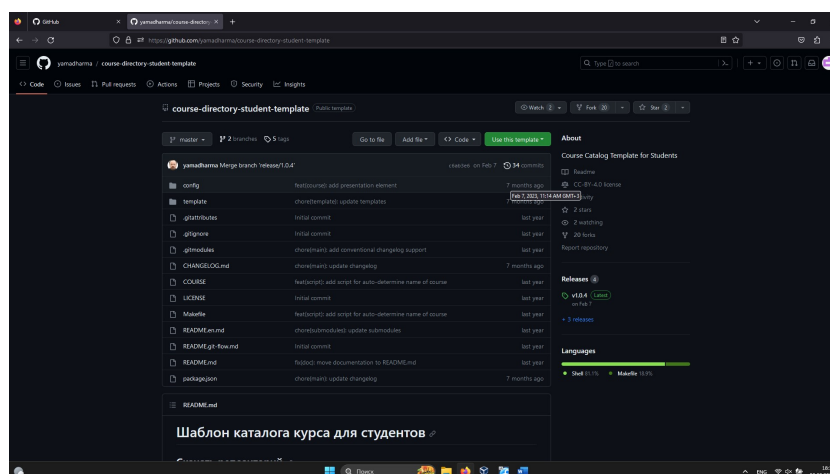


Рис. 8.1: Страница шаблона для репозитория

Задаю имя репозитория study\_2023–2024\_arch-рс и создаю репозиторий (кнопка Create repository)(рис. 8.2)

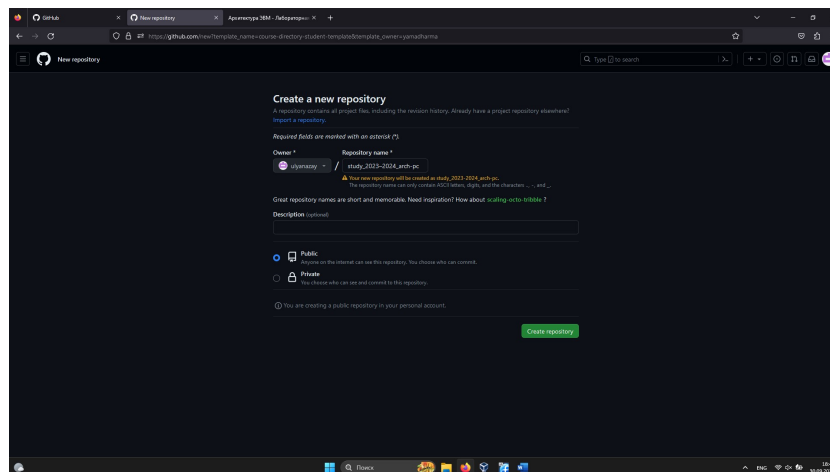


Рис. 8.2: Окно создания репозитория

Репозиторий создан!(рис. 8.3)

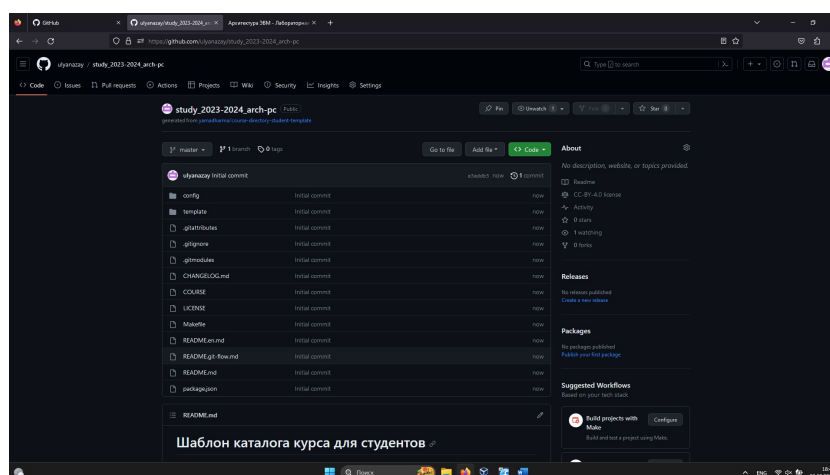
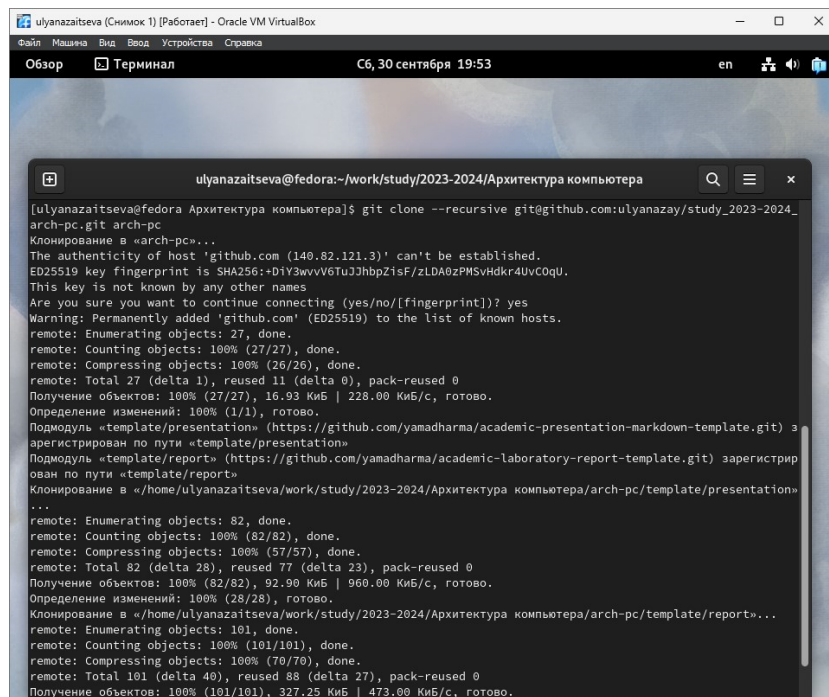


Рис. 8.3: Созданный репозиторий

Открываю терминал и перехожу в каталог курса: `cd ~/work/study/2023-2024/“Архитектура компьютера”`(плохо обрезала скрин и данная команда в него не попала, но по полю перед знаком \$ как раз указан каталог в котором я нахожусь, а именно каталог “Архитектура компьютера”, в который я и перешла до клонирования прописав данную команду!). Клонировую созданный репозиторий.(рис. 8.4)



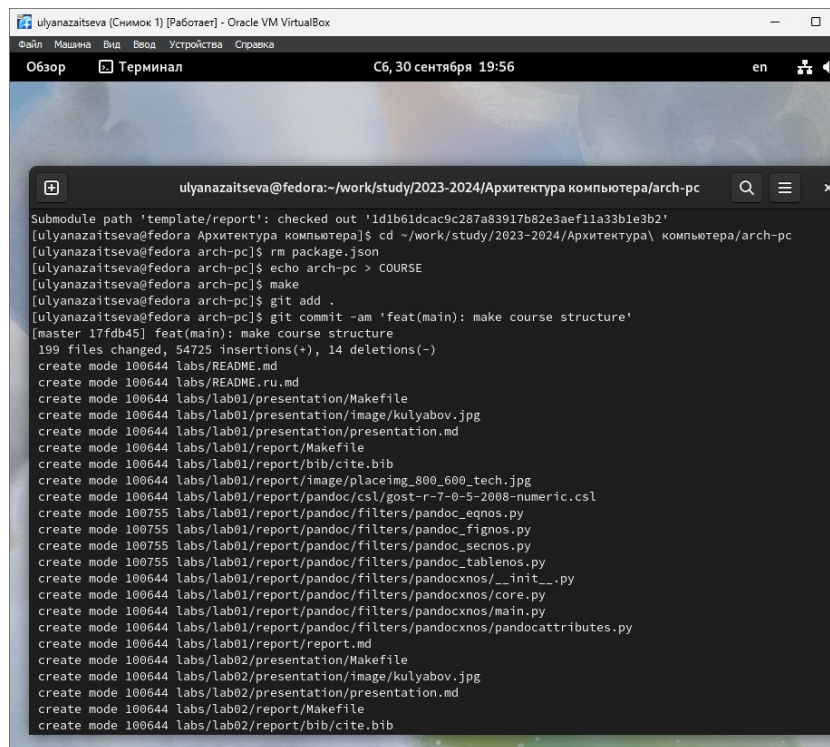


```
ulyanazaitseva@fedora:~/work/study/2023-2024/Архитектура компьютера
[ulyanazaitseva@fedora Архитектура компьютера]$ git clone --recursive git@github.com:ulyanazay/study_2023-2024_
arch-pc.git arch-pc
Клонирование в «arch-pc»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:-D1Y3wvvV6TujJhbpZ1sF/zLDA0zPMSvHdkr4UvC0qU.
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: 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 | 228.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) з
аарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрир
ован по пути «template/report»
Клонирование в «/home/ulyanazaitseva/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 | 960.00 КиБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/ulyanazaitseva/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 | 473.00 КиБ/с, готово.
```

Рис. 8.4: Клонирование репозитория

## 9 6. Настройка каталога курса.

Перехожу в каталог курса с помощью команды `cd`, удаляю лишние файлы с помощью команды `rm`, создаю необходимые каталоги, добавляю их и сохраняю изменения на сервере(рис. 9.1)



```
ulyanazaitseva@fedora:~/work/study/2023-2024/Архитектура компьютера/arch-pc
[ulyanazaitseva@fedora Архитектура компьютера]$ cd ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc
[ulyanazaitseva@fedora arch-pc]$ rm package.json
[ulyanazaitseva@fedora arch-pc]$ echo arch-pc > COURSE
[ulyanazaitseva@fedora arch-pc]$ make
[ulyanazaitseva@fedora arch-pc]$ git add .
[ulyanazaitseva@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master 17fdb45] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.ru.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
```

Рис. 9.1: Перемещение между директориями, удаление лишних файлов, создание каталогов, сохранение изменений

Отправляю созданные каталоги на сервер(рис. 9.2)

```
[ulyanazaitseva@fedora arch-pc]$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.74 МиБ/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:ulyanazay/study_2023-2024_arch-pc.git
 a3addb3..17fdb45 master -> master
[ulyanazaitseva@fedora arch-pc]$ ~
```

Рис. 9.2: Выгрузка изменений на сервер

Проверяю правильность выполнения работы на сайте GitHub (рис. 9.3).

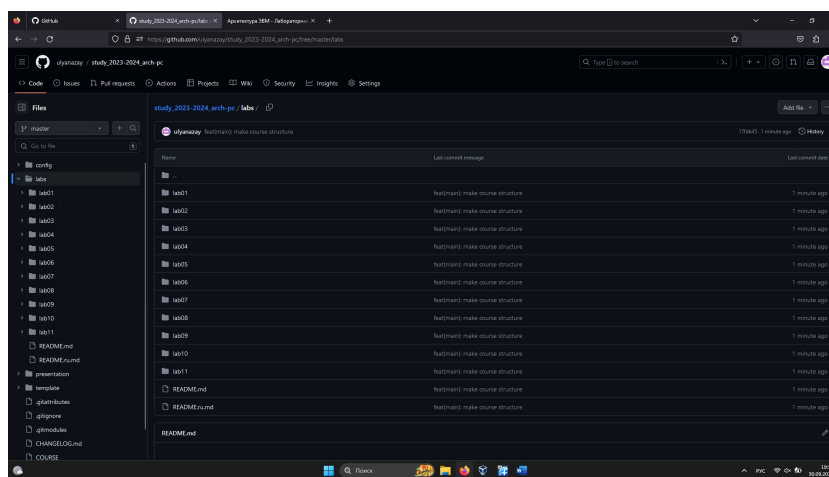


Рис. 9.3: Страница репозитория

## 10 7. Выполнение заданий для самостоятельной работы.

Загружаю отчёт по первой лабораторной работе в нужный каталог, а именно lab01/report, используя сайт(рис. 9.3)

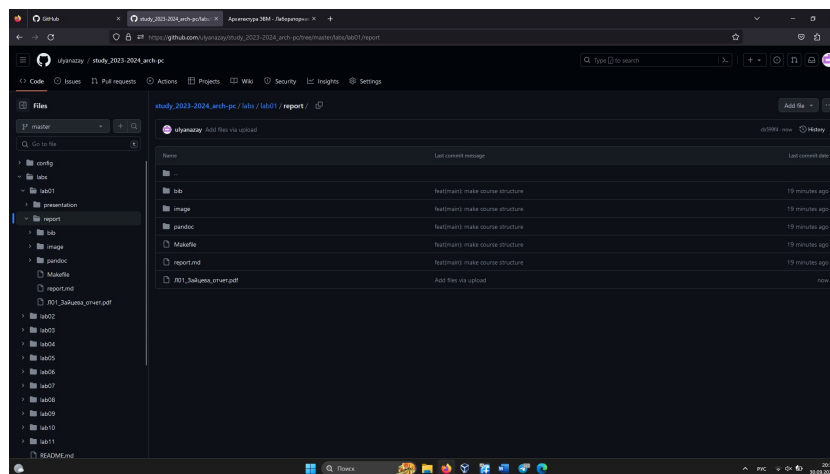


Рис. 10.1: Каталог lab01/report

Когда завершу данный отчёт, загружу его в каталог lab02/report!

## 11 Выводы

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