

# **Отчёт по лабораторной работе №2**

**Дисциплина: архитектура компьютера**

Шибаета Алесандра Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Настройка GitHub . . . . .	8
4.2	Базовая настройка Git . . . . .	9
4.3	Создание SSH-ключа . . . . .	10
4.4	Создание рабочего пространства и репозитория курса на основе шаблона . . . . .	13
4.5	Создание репозитория курса на основе шаблона . . . . .	13
4.6	Настройка каталога курса . . . . .	15
4.7	Выполнение заданий для самостоятельной работы . . . . .	17
<b>5</b>	<b>Выводы</b>	<b>19</b>
<b>6</b>	<b>Список литературы</b>	<b>20</b>

# Список иллюстраций

4.1	Заполнение данных учетной записи GitHub . . . . .	8
4.2	Аккаунт GitHub . . . . .	8
4.3	Предварительная конфигурация git . . . . .	9
4.4	Настройка кодировки . . . . .	9
4.5	Создание имени для начальной ветки . . . . .	9
4.6	Параметр autocrlf . . . . .	10
4.7	Параметр safecrlf . . . . .	10
4.8	Генерация SSH-ключа . . . . .	11
4.9	Установка утилиты xclip . . . . .	11
4.10	Копирование содержимого файла . . . . .	12
4.11	Окно SSH and GPG keys . . . . .	12
4.12	Добавление ключа . . . . .	13
4.13	Создание рабочего пространства . . . . .	13
4.14	Созданный репозиторий . . . . .	14
4.15	Перемещение между директориями . . . . .	14
4.16	Клонирование репозитория . . . . .	15
4.17	Окно с ссылкой для копирования репозитория . . . . .	15
4.18	Создание каталогов . . . . .	16
4.19	Добавление и сохранение изменений на сервере . . . . .	16
4.20	Выгрузка изменений на сервер . . . . .	17
4.21	Страница репозитория . . . . .	17
4.22	Создание файла . . . . .	17
4.23	Работа с отчетом в текстовом процессоре . . . . .	18
4.24	Перемещение между директориями . . . . .	18
4.25	Проверка местонахождения файлов . . . . .	18

# 1 Цель работы

Целью данной работы является изучить идеологию и применение средств контроля версий, а также 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 изменения, сделанные разными участниками, 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. 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. 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). Затем можно вносить изменения в локальное дерево и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.

## 4 Выполнение лабораторной работы

### 4.1 Настройка GitHub

Создаю учетную запись на сайте GitHub (рис. [4.1]). Далее я заполнила основные данные учетной записи.

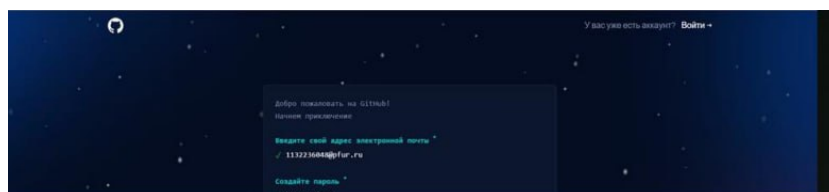


Рис. 4.1: Заполнение данных учетной записи GitHub

Аккаунт создан (рис. [4.2]).

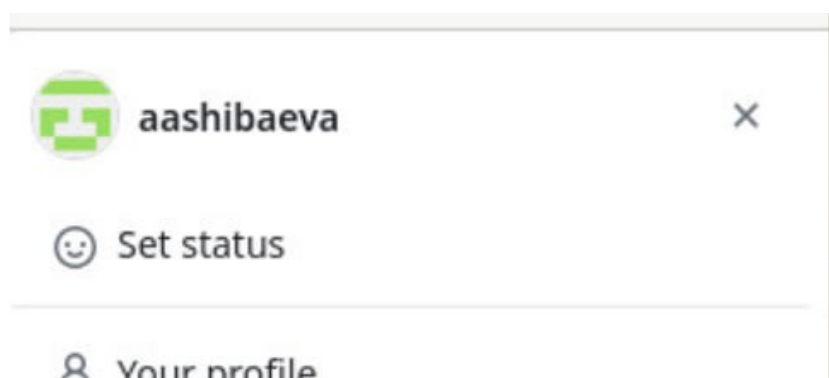
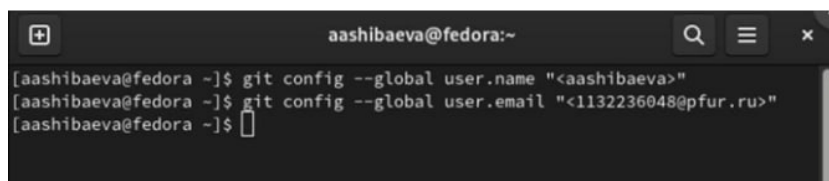


Рис. 4.2: Аккаунт GitHub



## 4.2 Базовая настройка Git

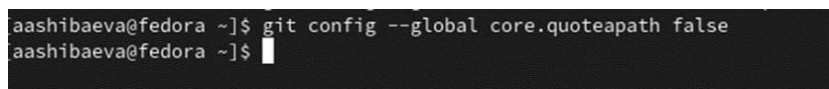
Открываю виртуальную машину, затем открываю терминал и делаю предварительную конфигурацию git. Ввожу команду `git config --global user.name "aashibaeva"`, указывая свое имя и команду `git config --global user.email "work@mail"`, указывая в ней электронную почту владельца, то есть мою (рис. [4.3]).

A screenshot of a terminal window titled 'aashibaeva@fedora:~'. It shows three lines of command execution: the first sets the global user name to 'aashibaeva', the second sets the global user email to '<1132236048@pfur.ru>', and the third shows the prompt character. The terminal has a dark background with light-colored text.

```
[aashibaeva@fedora ~]$ git config --global user.name "aashibaeva"
[aashibaeva@fedora ~]$ git config --global user.email "<1132236048@pfur.ru>"
[aashibaeva@fedora ~]$
```

Рис. 4.3: Предварительная конфигурация git

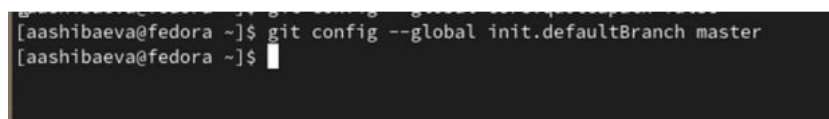
Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. [4.4]).

A screenshot of a terminal window showing a single command to set the global core.quoteapath to false. The terminal has a dark background with light-colored text.

```
[aashibaeva@fedora ~]$ git config --global core.quoteapath false
[aashibaeva@fedora ~]$
```

Рис. 4.4: Настройка кодировки

Задаю имя «master» для начальной ветки (рис. [4.5]).

A screenshot of a terminal window showing a single command to set the global init.defaultBranch to master. The terminal has a dark background with light-colored text.

```
[aashibaeva@fedora ~]$ git config --global init.defaultBranch master
[aashibaeva@fedora ~]$
```

Рис. 4.5: Создание имени для начальной ветки

Задаю параметр `autocrlf` со значением `input`, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. [4.6]). CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

```
[aashibaeva@fedora ~]$ git config --global core.autocrlf input
[aashibaeva@fedora ~]$
```

Рис. 4.6: Параметр autocrlf

Задаю параметр safecrlf со значением warn, так Git будет проверять преобразование на обратимость (рис. [4.7]). При значении warn Git только выведет предупреждение, но будет принимать необратимые конвертации.

```
[aashibaeva@fedora ~]$ git config --global core.safecrlf warn
[aashibaeva@fedora ~]$
```

Рис. 4.7: Параметр safecrlf

## 4.3 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. [4.8]). Ключ автоматически сохранится в каталоге `~/.ssh/`.

```

aashibaeva@fedora ~]$ ssh-keygen -C "aashibaeva <1132236048@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aashibaeva/.ssh/id_rsa):
Created directory '/home/aashibaeva/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aashibaeva/.ssh/id_rsa
Your public key has been saved in /home/aashibaeva/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:AYUESRmVU3VynX+0FaPfFzboM3/LTkHceQWJzKdVJ8E aashibaeva <1132236048@pfur.ru>
The key's randomart image is:
---[RSA 3072]-----+
  +=+=0.000++0=|
  ++. += E.*|
  o .. X 0o|
  . o *.*|
  S + o+|
    + o|
    o.|
    o o|
    .+ |
-----[SHA256]-----+
aashibaeva@fedora ~]$

```

Рис. 4.8: Генерация SSH-ключа

Xclip – утилита, позволяющая скопировать любой текст через терминал. Оказывается, в дистрибутиве Linux Kali ее сначала надо установить. Устанавливаю xclip с помощью команды apt-get install с ключом -у от имени суперпользователя, введя в начале команды sudo (рис. [4.9]).

```

[!%+. |
[0Bo. |
[+E+ |
+----[SHA256]-----+
[aashibaeva@fedora ~]$ sudo dnf install -y xclip
[sudo] пароль для aashibaeva:
Последняя проверка окончания срока действия метаданных: 2:27:05 назад, C6 30 сен 2023 07:48:39.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
xclip      x86_64       0.13-19.git11cba61.fc38      fedora       37 k

Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 37 k
Объем изменений: 63 k
Загрузка пакетов:
xclip-0.13-19.git11cba61.fc38.x86_64.rpm                4.6 kB/s | 37 kB    00:08
-----
Общий размер                                           3.9 kB/s | 37 kB    00:09
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      :
Установка       : xclip-0.13-19.git11cba61.fc38.x86_64      1/1
Запуск скрипта  : xclip-0.13-19.git11cba61.fc38.x86_64      1/1
Проверка        : xclip-0.13-19.git11cba61.fc38.x86_64      1/1

Установлен:
xclip-0.13-19.git11cba61.fc38.x86_64

Выполнено!
[aashibaeva@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
[aashibaeva@fedora ~]$

```

Рис. 4.9: Установка утилиты xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip (рис. [4.10]).

```
[aashibaeva@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
[aashibaeva@fedora ~]$ cat ~/.ssh/id_rsa.pub |
> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCV2F8KeRyT4sFcATq2TvUjeuUvd3I
```

Рис. 4.10: Копирование содержимого файла

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key» (рис. [4.11]).

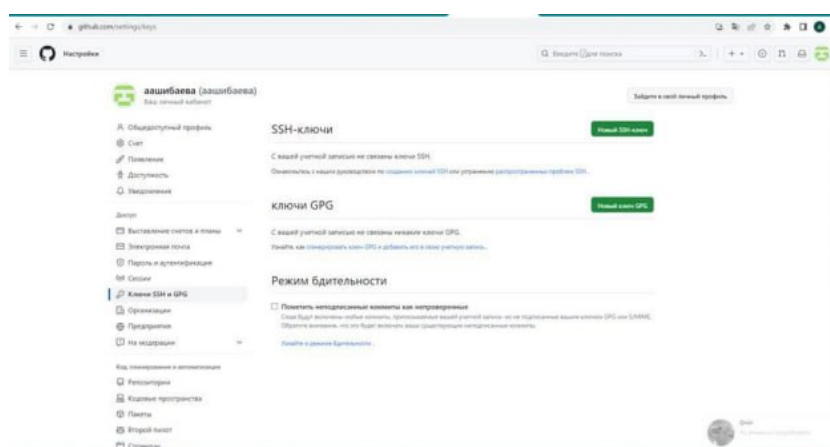


Рис. 4.11: Окно SSH and GPG keys

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. [4.12]).

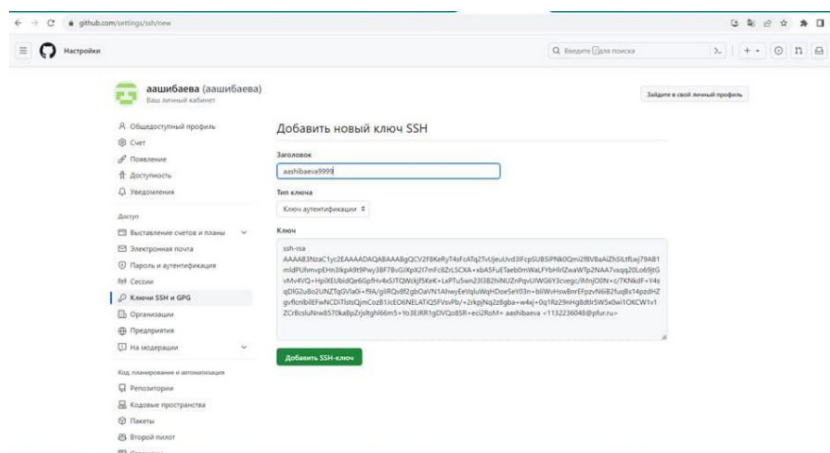


Рис. 4.12: Добавление ключа

## 4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Закрываю браузер, открываю терминал. Создаю директорию, рабочее пространство, с помощью утилиты `mkdir`, благодаря ключу `-p` создаю все директории после домашней `~/work/study/2022-2023/“Архитектура компьютера”` рекурсивно. Далее проверяю с помощью `ls`, действительно ли были созданы необходимые мне каталоги (рис. [4.13]).

```
[aashibaeva@fedora ~]$ mkdir -p work/2023-2024/"Архитектура компьютера и операционных систем"
[aashibaeva@fedora ~]$ ls
work  Видео  Документы  Загрузки  Избранное  Музыка  Общедоступные  "Рабочий стол"  Шаблоны
```

Рис. 4.13: Создание рабочего пространства

## 4.5 Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория.

В открывшемся окне задаю имя репозитория (Repository name): study\_2022–2023\_arh-  
pc и создаю репозиторий, нажимаю на кнопку «Create repository from template».

Репозиторий создан (рис. [4.14]).

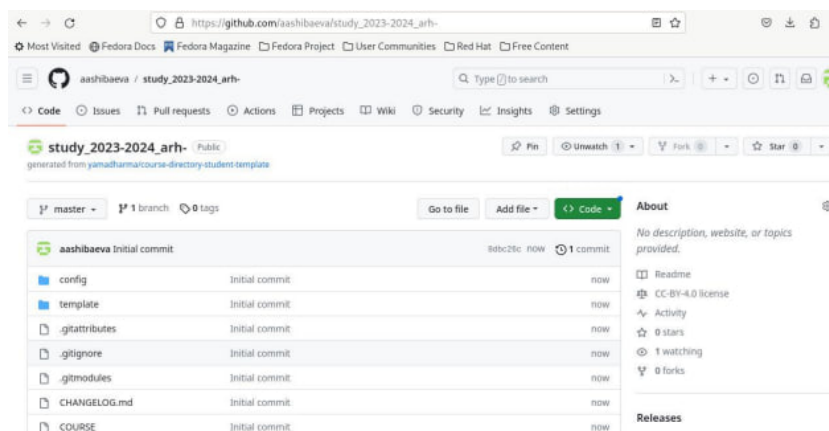


Рис. 4.14: Созданный репозиторий

Через терминал перехожу в созданный каталог курса с помощью утилиты cd  
(рис. [4.15]).

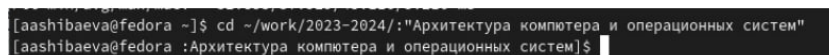


Рис. 4.15: Перемещение между директориями

Клонирую созданный репозиторий с помощью команды git clone –recursive  
git@github.com:/study\_2023–2024\_arh-pc.git arch-pc (рис. [4.16]).

```

[aashibaeva@fedora :Архитектура компьютера и операционных систем]$ mkdir -p ~/work/study/2023-2024/Архитектура компьютера
[aashibaeva@fedora :Архитектура компьютера и операционных систем]$ cd ~/work/study/2023-2024/Архитектура компьютера
[aashibaeva@fedora :Архитектура компьютера]$ git clone --recursive git@github.com:aashibaeva/study_2023-2024_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 KiB | 8.47 MiB/c, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Клонирование в «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/aashibaeva/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 25), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.90 KiB | 319.00 KiB/c, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/aashibaeva/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 KiB | 943.00 KiB/c, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3880ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a839175b2e3ae5f1a33b1e3b2'
[aashibaeva@fedora :Архитектура компьютера]$

```

Рис. 4.16: Клонирование репозитория

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. [4.17]).

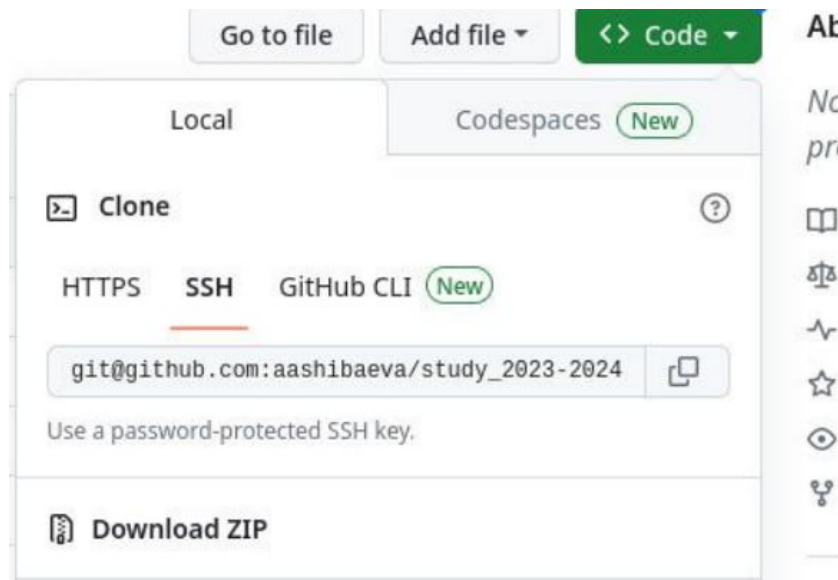


Рис. 4.17: Окно с ссылкой для копирования репозитория

## 4.6 Настройка каталога курса

Удаляю лишние файлы с помощью утилиты `rm`.

Создаю необходимые каталоги (рис. [4.18]).

```

[aashibaeva@fedora arch-pc]$ rm package.json
[aashibaeva@fedora arch-pc]$ echo arch-pc > COURSE
bash: echo: команда не найдена...
[aashibaeva@fedora arch-pc]$ echo arch-pc > COURSE
[aashibaeva@fedora arch-pc]$ make
bash: make: команда не найдена...
Установить пакет «make», предоставляющий команду «make»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов...
Следующие пакеты должны быть установлены:
gc-8.2.2-3.fc38.x86_64 Garbage collector for C and C++
guile22-2.2.7-7.fc38.x86_64 A GNU implementation of Scheme for application extensibility
make-1:4.4.1-1.fc38.x86_64 A GNU tool which simplifies the build process for users
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...

```

Рис. 4.18: Создание каталогов

Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью `git add`, комментирую и сохраняю изменения на сервере как добавление курса с помощью `git commit` (рис. [4.19]).

```

[aashibaeva@fedora arch-pc]$ git add .
[aashibaeva@fedora arch-pc]$ git commit -m 'feat(main): make course structure'
master 7d99531 feat(main): make course structure
130 files changed, 5475 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/placement_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-8-5-2008-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/placement_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-8-5-2008-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/_init_.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab02/report/report.md
create mode 100644 labs/lab03/presentation/Makefile
create mode 100644 labs/lab03/presentation/image/kulyabov.jpg
create mode 100644 labs/lab03/presentation/presentation.md
create mode 100644 labs/lab03/report/Makefile
create mode 100644 labs/lab03/report/bib/cite.bib

```

Рис. 4.19: Добавление и сохранение изменений на сервере

Отправляю все на сервер с помощью `push` (рис. [4.20]).



```
[aashibaeva@fedora arch-pc]$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 1.95 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:aashibaeva/study_2023-2024_arch-pc.git
44afa54..7db9634 master -> master
[aashibaeva@fedora arch-pc]$
```

Рис. 4.20: Выгрузка изменений на сервер

Проверяю правильность выполнения работы сначала на самом сайте GitHub (рис. [4.21]).

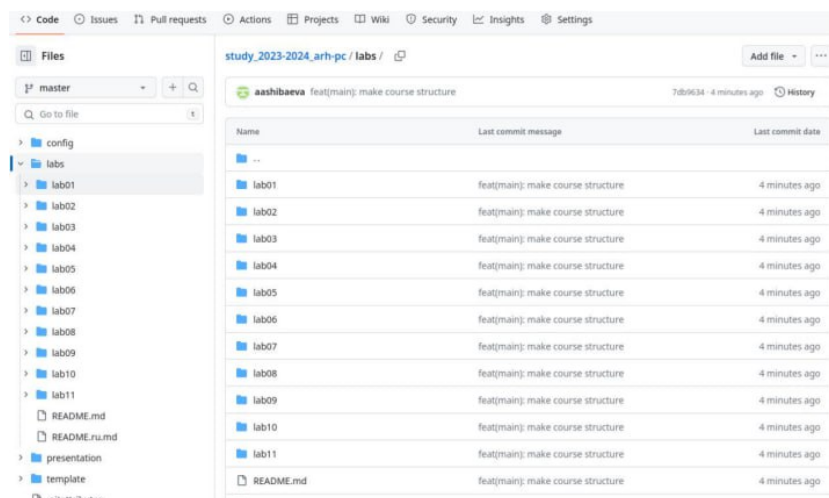


Рис. 4.21: Страница репозитория

## 4.7 Выполнение заданий для самостоятельной работы

1. Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch (рис. [4.22]).

```
[aashibaeva@fedora arch-pc]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab02/report
[aashibaeva@fedora report]$ touch l02_ашбаева_отчет
[aashibaeva@fedora report]$
```

Рис. 4.22: Создание файла

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений.

После открытия текстового процессора открываю в нем созданный файл и могу начать в нем работу над отчетом (рис. [4.23]).

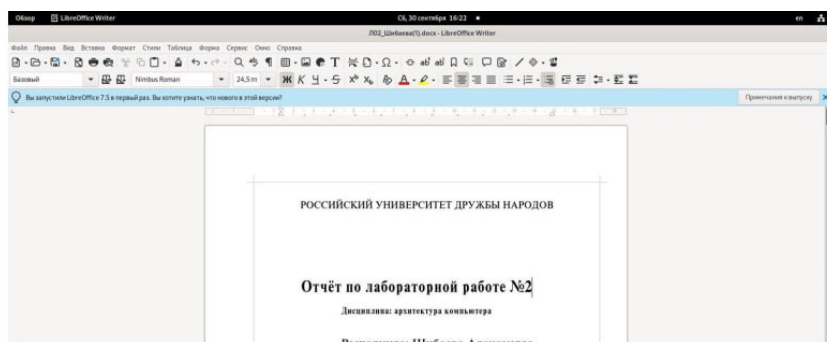


Рис. 4.23: Работа с отчетом в текстовом процессоре

2. Перехожу из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd (рис. [4.24]).

```
серо 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:aashibaeva/study_2023-2024_arh-pc.git
  44afa54..7db9634 master -> master
aashibaeva@fedora arch-pc$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab2/report
aashibaeva@fedora arch-pc$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab2/report
aashibaeva@fedora report$ touch l02_шибеева_отчет
aashibaeva@fedora report$ cd ..
aashibaeva@fedora lab02$ cd ..
aashibaeva@fedora labs$ cd lab01/
aashibaeva@fedora lab01$ cd ..
aashibaeva@fedora labs$ cd lab01/report/
aashibaeva@fedora report$
```

Рис. 4.24: Перемещение между директориями

Проверяю местонахождение файлов с отчетами по первой и второй лабораторным работам. Они должны быть в подкаталоге домашней директории «Загрузки», для проверки использую команду ls (рис. [4.25]).

```
[aashibaeva@fedora report]$ ls ~/Загрузки
l01_шибеева_отчет.pdf l02_шибеева_отчет.pdf
[aashibaeva@fedora report]$
```

Рис. 4.25: Проверка местонахождения файлов

## 5 Выводы

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

## 6 Список литературы

1. Архитектура ЭВМ
2. Git - gitattributes Документация