

# **Отчёт по лабораторной работе №2**

**Дисциплина: Архитектура компьютера**

Бызова Мария Олеговна

# Содержание

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Цель работы</b>   | <b>5</b>  |
| <b>2</b> | <b>Теоретическое введение</b>  | <b>6</b>  |
| <b>3</b> | <b>Выполнение лабораторной работы</b>  | <b>8</b>  |
| 3.1      | 1. Настройка Github. . . . .   | 8         |
| 3.2      | 2. Базовая настройка Git. . . . .  | 10        |
| 3.3      | 3. Создание SSH ключа. . . . .   | 11        |
| 3.4      | 4. Создание рабочего пространства и репозитория курса на основе шаблона. . . . . | 13        |
| 3.5      | 5. Создание репозитория курса на основе шаблона. . . . .                         | 14        |
| 3.6      | 6. Настройка каталога курса. . . . .   | 17        |
| <b>4</b> | <b>Описание результатов выполнения заданий для самостоятельной работы.</b>       | <b>20</b> |
| <b>5</b> | <b>Выводы</b>  | <b>23</b> |

## Список иллюстраций

|      |   |    |
|------|---|----|
| 3.1  | Главная страница сайта Github. . . . .                                | 8  |
| 3.2  | Вход в ранее созданный аккаунт. . . . .                               | 9  |
| 3.3  | Аккаунт Github. . . . .   | 9  |
| 3.4  | Предварительная конфигурация git. . . . .                             | 10 |
| 3.5  | Настройка кодировки . . . . .   | 10 |
| 3.6  | Создание имени для начальной ветки. . . . .                           | 10 |
| 3.7  | Параметр autocrlf. . . . .  | 11 |
| 3.8  | Параметр safecrlf. . . . .  | 11 |
| 3.9  | Создание SSHключа. . . . .  | 12 |
| 3.10 | Добавление ключа. . . . .   | 13 |
| 3.11 | Создание рабочего пространства. . . . .                               | 13 |
| 3.12 | Проверка работы команды. . . . .                                      | 14 |
| 3.13 | Страница шаблона для репозитория. . . . .                             | 14 |
| 3.14 | Окно создания репозитория. . . . .                                    | 15 |
| 3.15 | Созданный репозиторий. . . . .  | 15 |
| 3.16 | Перемещение между директориями. . . . .                               | 15 |
| 3.17 | Окно с ссылкой для копирования репозитория. . . . .                   | 16 |
| 3.18 | Клонирование репозитория. . . . .                                     | 16 |
| 3.19 | Перемещение между директориями. . . . .                               | 17 |
| 3.20 | Удаление файлов. . . . .  | 17 |
| 3.21 | Создание каталогов. . . . .   | 17 |
| 3.22 | Добавление и сохранение изменений на сервере. . . . .                 | 17 |
| 3.23 | Выгрузка изменений на сервер. . . . .                                 | 18 |
| 3.24 | Страница репозитория (1). . . . .                                     | 18 |
| 3.25 | Страница репозитория (2). . . . .                                     | 19 |
| 4.1  | Перемещение между директориями. . . . .                               | 20 |
| 4.2  | Создание файла. . . . .   | 20 |
| 4.3  | Проверка местонахождения файла. . . . .                               | 21 |
| 4.4  | Копирование файл. . . . .   | 21 |
| 4.5  | Проверка выполнения команды. . . . .                                  | 21 |
| 4.6  | Загрузка файла с отчетом по лабораторной работе №1 на GitHub. . . . . | 21 |
| 4.7  | Проверка правильности работы команд. . . . .                          | 21 |
| 4.8  | Загрузка файла с отчетом по лабораторной работе №2 на GitHub. . . . . | 22 |
| 4.9  | Проверка правильности работы команд. . . . .                          | 22 |

## **Список таблиц**

# 1 Цель работы

Целью данной лабораторной работы является изучение идеологии и применение средств контроля версий, приобретение практических навыков по работе с системой git.

## 2 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 дельтакомпрессию — 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 7

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 локальном дереве и/или ветке. После завершения внесения какого-либо изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории.

## 3 Выполнение лабораторной работы

### 3.1 1. Настройка Github.

Чтобы приступить к настройке, зайдя через браузер на сайт GitHub, создаем учетную запись или входим в ранее уже созданную. При необходимости заполняем основные данные. Для выполнения лабораторной работы я воспользуюсь ранее созданной учетной записью. (рис. [3.1], [3.2], [3.3])

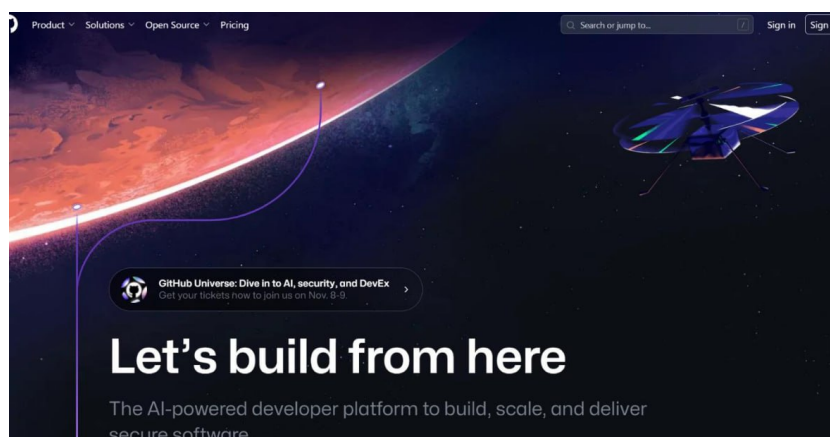


Рис. 3.1: Главная страница сайта Github.



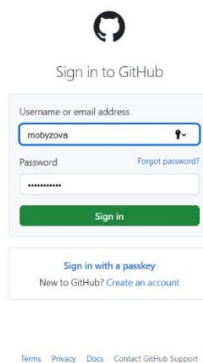


Рис. 3.2: Вход в ранее созданный аккаунт.

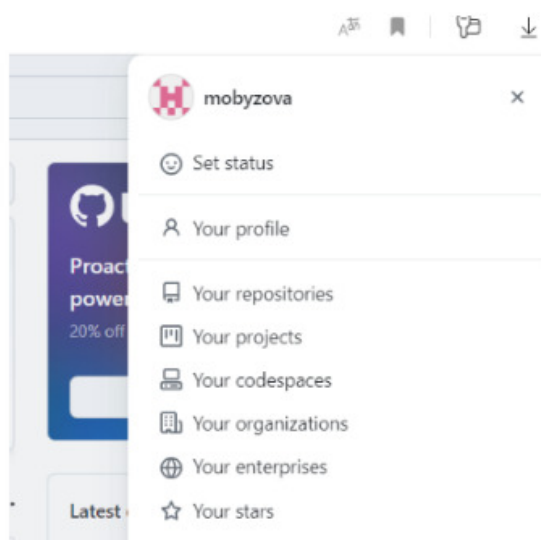


Рис. 3.3: Аккаунт Github.

## 3.2 2. Базовая настройка Git.

Для дальнейшей настройки Git открываем терминал и вводим следующие команды: команду `git config --global user.name ""`, указывая свое имя, и команду `git config --global user.email "work@mail"`, указывая в ней электронную почту владельца репозитория (рис. [3.4]).

```
mobihzova@dk8n73 ~ $ git config --global user.name "<mobyzova>"
mobihzova@dk8n73 ~ $ git config --global user.email "<1132236129@pfur.ru>"
mobihzova@dk8n73 ~ $
```

Рис. 3.4: Предварительная конфигурация git.

Настраиваем utf8 в выводе сообщений git для корректного отображения символов (рис. [3.5]).

```
mobihzova@dk8n73 ~ $ git config --global core.core.quotepath false

```

Рис. 3.5: Настройка кодировки

Далее зададим имя начальной ветки. Назовём её «master» (рис. [3.6])

```
mobihzova@dk8n73 ~ $ git config --global init.defaultBranch master

```

Рис. 3.6: Создание имени для начальной ветки.

Зададим параметр `autocrlf` со значением `input` для того, чтобы конвертировать CRLF в LF только при коммитах. Уточним, что CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах (рис. [3.7])

```
mobihzova@dk8n73 ~ $ git config --global core.autocrlf input
```

Рис. 3.7: Параметр autocrlf.

Далее зададим параметр safecrlf со значением warn. Таким образом, Git будет проверять преобразование на обратимость. В дальнейшей работе при значении warn Git только выведет предупреждение, но будет принимать необратимые конвертации (рис. [3.8]).

```
mobihzova@dk8n73 ~ $ git config --global core.safecrlf warn
```

Рис. 3.8: Параметр safecrlf.

### 3.3 3. Создание SSH ключа.

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого вводим команду ssh-keygen с “Имя Фамилия, work@email”, указывая имя владельца и электронную почту владельца. Ключ автоматически сохранится в каталоге ~/.ssh/ (рис. [3.9]).

```

mobihzova@dk8n73 ~ $ ssh-keygen -C "mobyzova <1132236129@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/.ssh/id_rsa):
Created directory '/afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/m/o/mobihzova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1j09zAQ1SI8j9goPoo2Mh7sfc1bNOy8dlkZtSRE+R60 mobyzova <1132236129@pfur.ru>
The key's randomart image is:
+---[RSA 3072]-----+
|      .oo=0..  |
|      .=.o   |
|      o oo=... |
|      + =.++oE |
|      . + S.+o= |
|      + + o = 0=. |
|o B +  =+ .   |
| o =   .o.    |
|oo.         |
+---[SHA256]-----+
mobihzova@dk8n73 ~ $

```

Рис. 3.9: Создание SSHключа.

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

```

mobihzova@dk8n73 ~ $ cat ~/.ssh/id_rsa.pub | xclip -sel clip
mobihzova@dk8n73 ~ $

```

{#fig:010

width=70%}

Далее необходимо загрузить сгенерённый открытый ключ. Для этого заходим на сайт <http://github.org/> под своей учётной записью и переходим в меню Setting. После этого выбираем в боковом меню SSH and GPG keys и нажимаем кнопку New SSH key. Вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title). Нажимаем «Add SSHkey», чтобы завершить добавление ключа (рис. [3.10]).

Рис. 3.10: Добавление ключа.

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

После проделанных действий возвращаемся в терминал. Создаём каталог для предмета «Архитектура компьютера» с помощью утилиты `mkdir` с ключом `p`. Далее проверяем работу команды (рис. [3.11], [3.12]) любым удобным способом.

```
mobihzova@dk8n73 ~ $ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
mobihzova@dk8n73 ~ $
```

Рис. 3.11: Создание рабочего пространства.

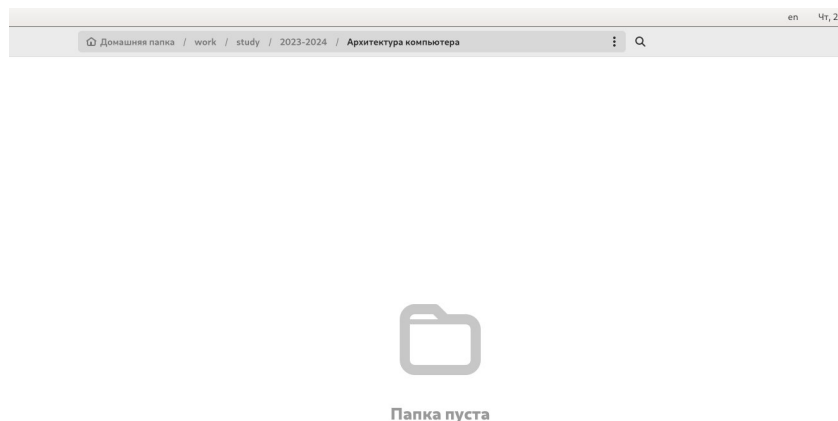


Рис. 3.12: Проверка работы команды.

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

Далее в браузере переходим на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharmacoursedirectorystudenttemplate>. Теперь выбираем «Use this template», чтобы использовать этот шаблон для своего репозитория (рис. [3.13]).

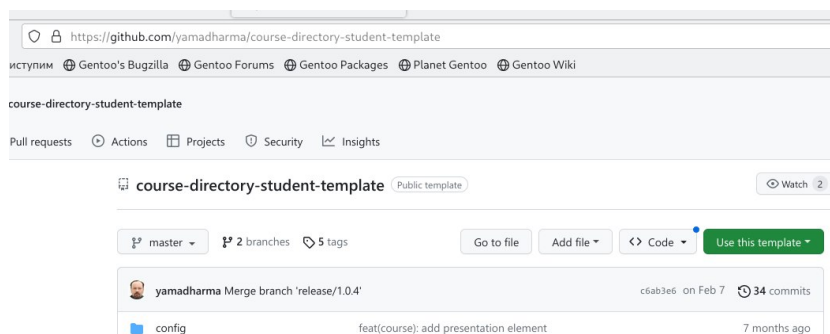


Рис. 3.13: Страница шаблона для репозитория.

В открывшемся окне задаем имя репозитория (Repository name) «study\_2023–2024\_arhpc» и создаем репозиторий (кнопка Create repository from template) (рис. [3.14]).

Рис. 3.14: Окно создания репозитория.

Таким образом, репозиторий успешно создан (рис. [3.15]).

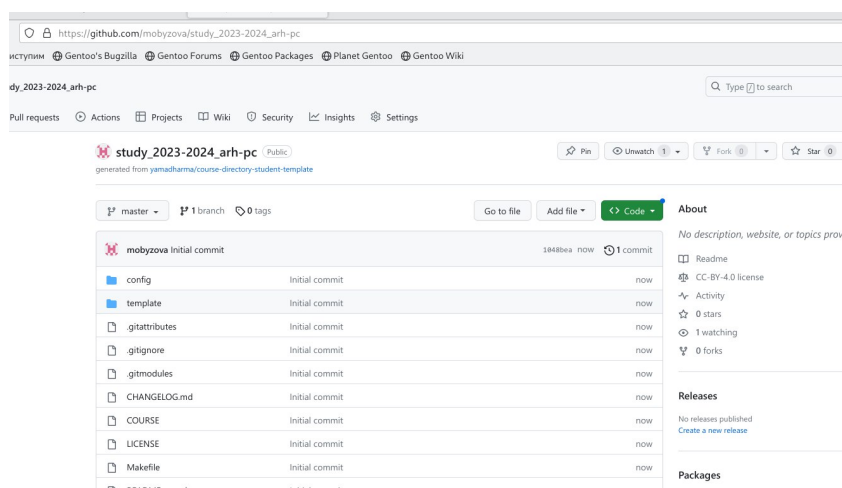


Рис. 3.15: Созданный репозиторий.

Для дальнейшей работы через терминал переходим в созданный каталог курса с помощью уже известной нам утилиты cd (рис. [3.16]).

```
mobihzova@dk8n73 ~ $ cd ~/work/study/2023-2024/"Архитектура компьютера"
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера $
```

Рис. 3.16: Перемещение между директориями.

Теперь клонируем созданный репозиторий с помощью команды `git clone --recursive git@github.com:/study_2023-2024_arhpc.git archpc`, предварительно скопировав ссылку для клонирования на странице созданного репозитория: сначала переходим в окно «code», далее выбираем в окне вкладку «SSH» (рис. [3.17], [3.18]).

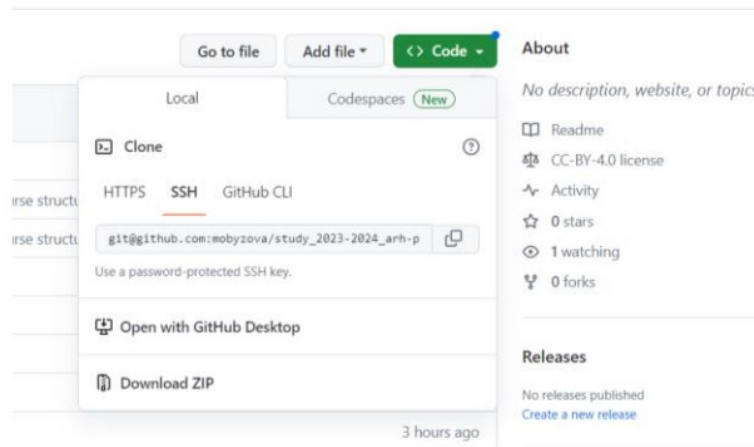


Рис. 3.17: Окно с ссылкой для копирования репозитория.

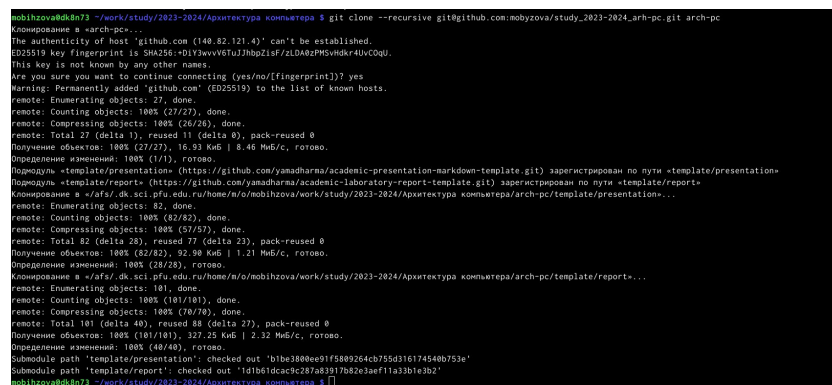


Рис. 3.18: Клонирование репозитория.

Таким образом, репозиторий успешно скопирован.



## 3.6 6. Настройка каталога курса.

Для дальнейшей работы переходим в каталог курса с помощью утилиты `cd` (рис. [3.19]).

```
nobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера $ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc
```

Рис. 3.19: Перемещение между директориями.

Теперь удалим лишние файлы с помощью утилиты `rm` (рис. [3.20]).

```
nobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ rm package.json
nobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $
```

Рис. 3.20: Удаление файлов.

Создадим все необходимые каталоги при помощи ряда команд (рис. [3.21]).

```
nobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ echo arch-pc > COURSE
nobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ make
nobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $
```

Рис. 3.21: Создание каталогов.

Отправим созданные нами каталоги на наш сервер с локального репозитория: для этого добавляем все созданные ранее каталоги при помощи `git add`, комментируем и сохраняем изменения на сервере как добавление курса с помощью `git commit` (рис. [3.22]).

```
nobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git add .
nobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git commit -am "feat(main): make course structure"
[master 2942e9a] 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
```

Рис. 3.22: Добавление и сохранение изменений на сервере.

При помощи push отправляем всё на сервер (рис. [3.23]).

```
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При скатки изменений используется до 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.61 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:mobyzova/study_2023-2024_arh-pc.git
 1048bea..2942e9a master -> master
mobihzova@dk8n73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $
```

Рис. 3.23: Выгрузка изменений на сервер.

В конце концов проверим правильность создания иерархии рабочего пространства в локальном репозитории и на странице github (рис. [3.24], [3.25]).

|  |                                   |                     |           |
|--|-----------------------------------|---------------------|-----------|
| mobyzova feat(main): make course structure |                                   | 2942e9a 3 hours ago | 2 commits |
| config                                     | Initial commit                    | 3 hours ago         |           |
| labs                                       | feat(main): make course structure | 3 hours ago         |           |
| presentation                               | feat(main): make course structure | 3 hours ago         |           |
| template                                   | Initial commit                    | 3 hours ago         |           |
| .gitattributes                             | Initial commit                    | 3 hours ago         |           |
| .gitignore                                 | Initial commit                    | 3 hours ago         |           |
| .gitmodules                                | Initial commit                    | 3 hours ago         |           |
| CHANGELOG.md                               | Initial commit                    | 3 hours ago         |           |
| COURSE                                     | feat(main): make course structure | 3 hours ago         |           |
| LICENSE                                    | Initial commit                    | 3 hours ago         |           |
| Makefile                                   | Initial commit                    | 3 hours ago         |           |
| README.en.md                               | Initial commit                    | 3 hours ago         |           |
| README.git-flow.md                         | Initial commit                    | 3 hours ago         |           |
| README.md                                  | Initial commit                    | 3 hours ago         |           |
| prepare                                    | feat(main): make course structure | 3 hours ago         |           |

Рис. 3.24: Страница репозитория (1).

mybzone feat(main): make course structure 2464 files · 3 hours ago History

| Name        | Last commit message               | Last commit date |
|-------------|-----------------------------------|------------------|
| ...         |                                   |                  |
| lab01       | feat(main): make course structure | 3 hours ago      |
| lab02       | feat(main): make course structure | 3 hours ago      |
| lab03       | feat(main): make course structure | 3 hours ago      |
| lab04       | feat(main): make course structure | 3 hours ago      |
| lab05       | feat(main): make course structure | 3 hours ago      |
| lab06       | feat(main): make course structure | 3 hours ago      |
| lab07       | feat(main): make course structure | 3 hours ago      |
| lab08       | feat(main): make course structure | 3 hours ago      |
| lab09       | feat(main): make course structure | 3 hours ago      |
| lab10       | feat(main): make course structure | 3 hours ago      |
| lab11       | feat(main): make course structure | 3 hours ago      |
| README.md   | feat(main): make course structure | 3 hours ago      |
| README.rund | feat(main): make course structure | 3 hours ago      |

Рис. 3.25: Страница репозитория (2).

## 4 Описание результатов выполнения заданий для самостоятельной работы.

1. Для создания отчета по выполнению лабораторной работы в соответствующем каталоге рабочего пространства переходим в директорию `labs/lab02/report` с помощью утилиты `cd`. Создаем в каталоге файл для отчета по второй лабораторной работе с помощью утилиты `touch` (рис. [4.1], [4.2]).

```
mobihzova@dkn73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report
mobihzova@dkn73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $
```

Рис. 4.1: Перемещение между директориями.

```
mobihzova@dkn73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $ touch lab02_Базова_Отчет
mobihzova@dkn73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $
```

Рис. 4.2: Создание файла.

2. Для копирования отчетов по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства проверим местонахождение файла с отчетом по первой лабораторной работе он должен находиться в подкаталоге домашней директории «Загрузки», для проверки используем команду `ls` (рис. [4.3]). Затем копируем необходимый файл с помощью утилиты `cp` (рис. [4.4]). Проверяем правильность выполнения команды `cp` с помощью `ls` (рис. [4.5]).

```
mobihzova@dkk73 ~$ ls ~/Зарпудки
'Telegram Web_files'  'Telegram Web.html'  Отчёт_Лабораторная_работа_№1_Бухзова_М.О..pdf
mobihzova@dkk73 ~$
```

Рис. 4.3: Проверка местонахождения файла.

```

m0b1hz0v@dk8n73 ~ % cp ~/Заргузки/Отчет_лабораторная_работа_№1_Бывова_М_0_.pdf ~/work/study/2023-2024/Архитектура_компьютера/arch-pc/labs/lab01/report
m0b1hz0v@dk8n73 ~ %

```

Рис. 4.4: Копирование файл.

```
mobihzova@dkn73 ~ % cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report
mobihzova@dkn73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report % ls
lib image Makefile pandoc report.md Ойчёр_Лабораторная_работа_№1_Булдана_М.О..pdf
mobihzova@dkn73 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report %
```

Рис. 4.5: Проверка выполнения команды.

3. Далее с помощью ранее изученных команд загружаем файлы на GitHub: добавляем с помощью `git add` нужный файл, сохраняем изменения с помощью `git commit`, отправляем в центральный репозиторий сохраненные изменения командой `git push`. Проверяем правильность работы (рис. [4.6], [4.7], [4.8], [4.9]).

[illegible]

Рис. 4.6: Загрузка файла с отчетом по лабораторной работе №1 на GitHub.

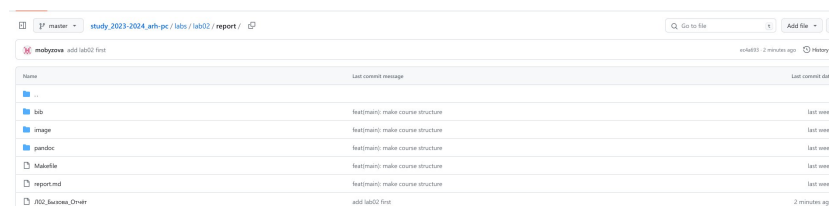
Рис. 4.7: Проверка правильности работы команд.

```

mobihzova@dk6n52 ~$ cd work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $ git add .
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $ git commit -am "add lab02 first"
[master ec4a693] add lab02 first
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "labs/lab02/report/\320\23302\320\221\321\213\320\267\320\276\320\262\320\266\320\236\321\202\321\207\321\221\321\202"
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При скатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 486 байтов | 486.00 Кб/с, готово.
Всего 5 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:mobyzoza/study_2023-2024_arh-pc.git
 faebf2c...ec4a693 master -> master
mobihzova@dk6n52 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report $

```

Рис. 4.8: Загрузка файла с отчетом по лабораторной работе №2 на GitHub.



| Name                | Last commit message               | Last commit date |
|---------------------|-----------------------------------|------------------|
| ..                  |                                   |                  |
| lab                 | feat(main): make course structure | last week        |
| image               | feat(main): make course structure | last week        |
| paradoc             | feat(main): make course structure | last week        |
| Makefile            | feat(main): make course structure | last week        |
| report.md           | feat(main): make course structure | last week        |
| ./DSC_Sources/Onair | add lab02 first                   | 2 minutes ago    |

Рис. 4.9: Проверка правильности работы команд.

Мы видим, что пояснение к совершенным действиям отображается, а отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой в lab01/report, по второй – в lab02/report. Убеждаемся, что всё выполнено верно.

## 5 Выводы

В ходе выполнения данной лабораторной работы были получены практические навыки работы с системой git, а также была изучена идеология и применение средств контроля версий посредством выполнения различных заданий, а именно: 1. Настройки GitHub. 2. Базовой настройки Git. 3. Создания SSHключа. 4. Создания рабочего пространства и репозитория курса на основе шаблона. 5. Создания репозитория курса на основе шаблона. 6. Настройки каталога курса. 7. Выполнения заданий для самостоятельной работы для подведения итогов и закрепления полученных знаний.