

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Архитектура компьютеров и операционные системы

Студент: Рудань Вячеслав Юрьевич

Группа: НПИбд-03-23

МОСКВА

2023 г.

СОДЕРЖАНИЕ

Теоретическое введение	3
Цель работы	5
Выполнения заданий лабораторной работы.....	5
Выполнение задания для самостоятельной работы	13
Заключение.....	16

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.

Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

Выполнения заданий лабораторной работы

Настройка github

Существует несколько доступных серверов репозитория с возможностью бесплатного размещения данных. Например, <http://bitbucket.org/>, <https://github.com/> и <https://gitflic.ru>. Для выполнения лабораторных работ предлагается использовать Github. Создайте учётную запись на сайте <https://github.com/> и заполните основные данные

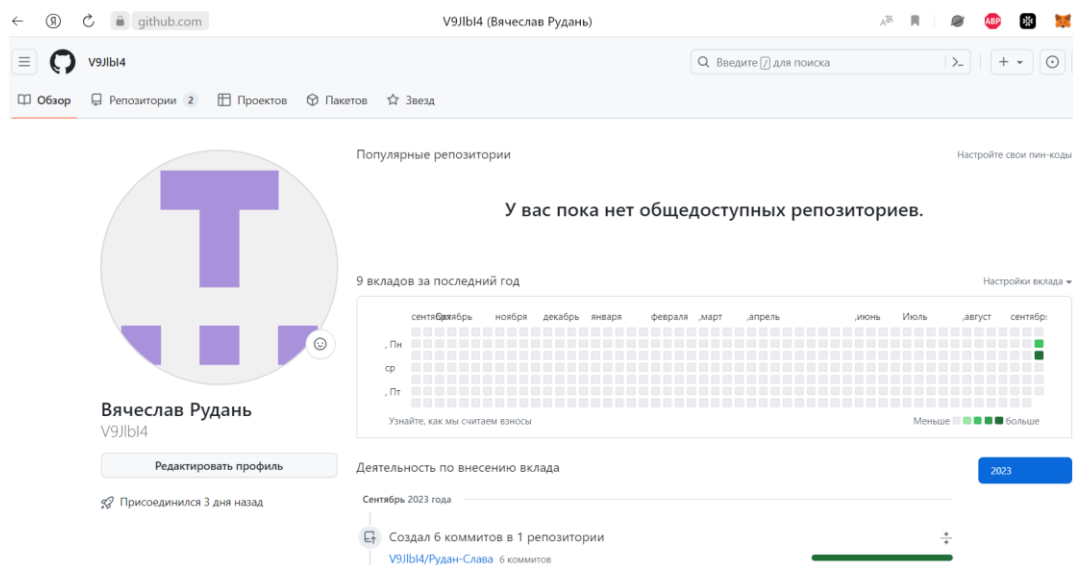


Рисунок 1 (Профиль в github)

Переходим по ссылке и создаем профиль в Github.

Базовая настройка git

Сначала сделаем предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и email владельца репозитория:

```
git config --global user.name "<Name Surname>"
```

```
git config --global user.email "<work@mail>"
```

Сперва нужно настроить git, установив user.name и user.email.

```
slavarudan@slavarudan-VirtualBox:~$ git config --global user.email "slava.rudan00@mail.ru"
slavarudan@slavarudan-VirtualBox:~$ git config --global user.name "V9JlbI4"
slavarudan@slavarudan-VirtualBox:~$
```

Рисунок 2(Настройка git)

Настроим utf-8 в выводе сообщений git:

```
git config --global core.quotepath false
```

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

Рисунок 3(настройка utf-8)

Зададим имя начальной ветки (будем называть её master):

```
git config --global init.defaultBranch master
```

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

Рисунок 4(задаем имя ветке)

Параметр autocrlf:

```
git config --global core.autocrlf input
```

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

Рисунок 5(Параметр autocrlf)

Команда `git config core.autocrlf` используется для изменения того, как Git обрабатывает окончания строк. Она принимает один аргумент.

Параметр safecrlf:

```
git config --global core.safecrlf warn
```

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

Рисунок 6(Параметр safecrlf)

`warn` - это значение, установленное для ключа `core.safecrlf`. Когда это значение установлено, Git будет предупреждать вас, если обнаружит неправильное сочетание символов перевода строки в текстовых файлах.

Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый):

```
ssh-keygen -C "Имя Фамилия <work@mail>"
```

```
slavarudan@slavarudan-VirtualBox:~$ ssh-keygen -C "Вячеслав Рудань slava.rudan00@mail.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/slavarudan/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/slavarudan/.ssh/id_rsa
Your public key has been saved in /home/slavarudan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:3KWfsC1ggLV3K84ukLbSemPcbb86/2CGUyZ3bojI2bo Вячеслав Рудань slava.rudan00@mail.ru
The key's randomart image is:
+---[RSA 3072]-----+
|
|  o .
| . o . . .
|  o o o o
| . = S =
| +. B X + = .
| + ++.B * = o
| . B oo++ +
| .= .E+O=+O.
|
+---[SHA256]-----+
slavarudan@slavarudan-VirtualBox:~$
```

Рисунок 7(Генерация ключей)

Команда **ssh-keygen** используется для создания ключей SSH в системе. Опция **-C** позволяет добавить комментарий к ключу, чтобы помочь вам идентифицировать ключ в будущем.

1. Генерируем пару публичных или приватных ключей
2. Потом создаем парольную фразу и еще раз вводим ее
3. Готово, ваши данные сохранены и ключ тоже

Ключи сохраняются в каталоге `~/.ssh/`. Далее необходимо загрузить сгенерированный открытый ключ. Для этого зайти на сайт <http://github.org/> под своей учётной записью и перейти в меню **Setting**. После этого выбрать в боковом меню **SSH and GPG keys** и нажать кнопку **New SSH key**. Скопировав из локальной консоли ключ в буфер обмена

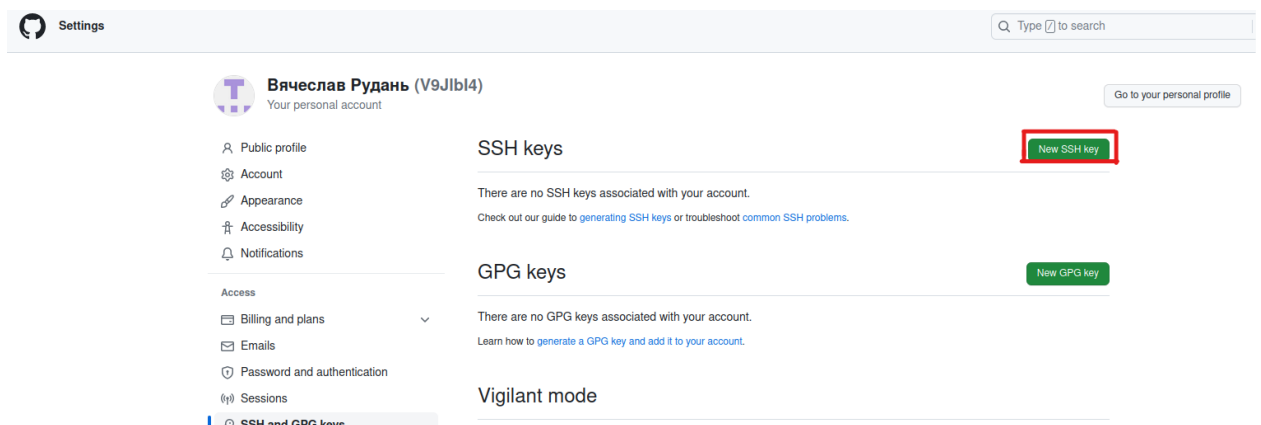


Рисунок 8(Добавляем сгенерированный ключ)

Открываю ключ в терминале чтобы добавить в github.

```
slavarudan@slavarudan-VirtualBox:~$ cat ~/.ssh/id_rsa.pub
cat: /home/slavarudan/.ssh/id_rsa.pub: Нет такого файла или каталога
slavarudan@slavarudan-VirtualBox:~$ cd
slavarudan@slavarudan-VirtualBox:~$ cd .ssh
slavarudan@slavarudan-VirtualBox:~/.ssh$ ls
id_rsa  id_rsa.pub
slavarudan@slavarudan-VirtualBox:~/.ssh$ cat id_rsa.pub | xclip
Команда «xclip» не найдена, но может быть установлена с помощью:
sudo apt install xclip
slavarudan@slavarudan-VirtualBox:~/.ssh$ sudo apt install xclip
[sudo] пароль для slavarudan:
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
  xclip
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 52 пакетов не обновлено.
Необходимо скачать 18,3 kB архивов.
После данной операции объем занятого дискового пространства возрастет на 60,4 kB.
Пол:1 http://nl.archive.ubuntu.com/ubuntu jammy/universe amd64 xclip amd64 0.13-2 [18,3 kB]
Получено 18,3 kB за 0с (36,8 kB/s)
Выбор ранее не выбранного пакета xclip.
(Чтение базы данных ... на данный момент установлено 208597 файлов и каталогов.)
Подготовка к распаковке ./xclip_0.13-2_amd64.deb ...
Распаковывается xclip (0.13-2) ...
Настраивается пакет xclip (0.13-2) ...
Обрабатываются триггеры для man-db (2.10.2-1) ...
^[[A^[[[slavarudan@slavarudan-VirtualBox:~/.ssh$ cat id_rsa.pub | xclip
slavarudan@slavarudan-VirtualBox:~/.ssh$ cd
slavarudan@slavarudan-VirtualBox:~$ cd .ssh
slavarudan@slavarudan-VirtualBox:~/.ssh$ cat id_rsa.pub | xclip
slavarudan@slavarudan-VirtualBox:~/.ssh$ xclip -o
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgOCwSSiWk+mMyb6iLDCOG6iL+Mo3ITz3G+QrLE2t0NwVA88TF/4H4+rabAK
OmtAZS4A5Sn3wBgJmyyiS0VMYOi8JymfLdTr1tfo0I+zDJGEMu2gMNY2l1XhEqy0bYzu3KvYf14R8cmGahrlnsLFA58ajY
Jyk9/zNg/ASZCnMxiozRqunfDmewdg4wP+DRbkkIFBKXclHuCKBz6TEQ1AQ7C5tNmmPpOKZve
/9hsGQeh9gVwWarpadtnRJWDpk6+Y3YmgsIrhOtYSyrH1pWqQUpI65Uui98a8pDKBy44
/ddeDcnG4yXivJUOJ6KnYMR55Ca1UKYkC4OB4IzuHwNN/UjIQT+067nQlhwJ9BUy3T5Brv42b
/RZao5csfUCgT14M1fRJenDmyOHRLXPUMWijjhd15wLzDzIO8oJLA+aVuOj2Hptp8XSf0nE9vpEFSBzVEK5dEsa+qITx
Ak18qaEZOAYuLIYZF7gVbQ+qYCI8DTHRIVkiUhdnAFFCWP7yas= Вячеслав Рудань slava.rudan00@mail.ru
slavarudan@slavarudan-VirtualBox:~/.ssh$
```

Рисунок 9(Открываю ключ)

Копирую и добавляю ключ в git.

Add new SSH Key

Title

Title

Key type

Authentication Key

Key

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQgOCwSSiWk+mMyb6iLDCOG6iL+Mo3ITz3G+QrLE2t0NwVA88TF/4H4+rabAK
OmtAZS4A5Sn3wBgJmyyiS0VMYOi8JymfLdTr1tfo0I+zDJGEMu2gMNY2l1XhEqy0bYzu3KvYf14R8cmGahrlnsLFA58ajY
Jyk9/zNg/ASZCnMxiozRqunfDmewdg4wP+DRbkkIFBKXclHuCKBz6TEQ1AQ7C5tNmmPpOKZve
/9hsGQeh9gVwWarpadtnRJWDpk6+Y3YmgsIrhOtYSyrH1pWqQUpI65Uui98a8pDKBy44
/ddeDcnG4yXivJUOJ6KnYMR55Ca1UKYkC4OB4IzuHwNN/UjIQT+067nQlhwJ9BUy3T5Brv42b
/RZao5csfUCgT14M1fRJenDmyOHRLXPUMWijjhd15wLzDzIO8oJLA+aVuOj2Hptp8XSf0nE9vpEFSBzVEK5dEsa+qITx
Ak18qaEZOAYuLIYZF7gVbQ+qYCI8DTHRIVkiUhdnAFFCWP7yas= Вячеслав Рудань slava.rudan00@mail.ru

Add SSH key


Рисунок 10(Вставляю ключ)

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



SSH

Title

SHA256:6eIj8ME7TFGaH7iIrwfAz0JQ1jd3wK12+EZYJZZnZhPA

Added on Sep 30, 2023

Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

Ключ добавлен.

Создание рабочего пространства и репозитория курса на основе шаблона

Откройте терминал и создайте каталог для предмета «Архитектура компьютера»:

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

Рисунок 12(Создание каталога)

Создание репозитория курса на основе шаблона

Репозиторий на основе шаблона можно создать через web-интерфейс github. Перейдите на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Далее выберите Use this template.

После этого добавляем имя репозитория указанного в ЛР2

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *

V9JlB14 / study_2023-2024_arhpc

Your new repository will be created as study_2023-2024_arhpc.
The repository name can only contain ASCII letters, digits, and the characters -, ., and _.

Great repository names are short and memorable. Need inspiration? How about [symmetrical-waddle](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

You are creating a public repository in your personal account.

[Create repository](#)

Рисунок 13(Создание репозитория)

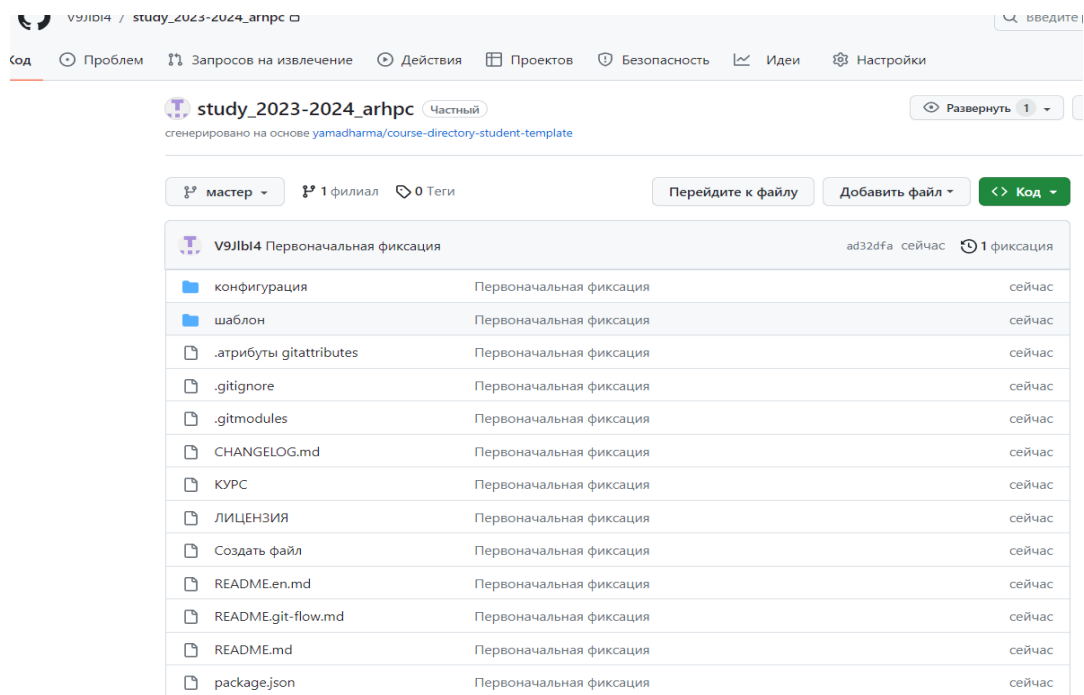


Рисунок 14(Созданный репозиторий)

Откройте терминал и перейдите в каталог курса:

Вводим полный путь каталога и переходим по нему с помощью **cd**

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

Рисунок 15(Переходим в каталог)

Клонируйте созданный репозиторий:

Команда **git clone --recursive** используется для клонирования Git-репозитория вместе со всеми его подмодулями. Подмодули в Git - это репозитории, вложенные внутри другого Git-репозитория. Когда вы клонируете репозиторий с помощью флага **--recursive**, Git не только клонирует основной репозиторий, но и инициализирует и клонирует все подмодули, связанные с ним.

```
slavarudan@slavarudan-VirtualBox: /work/study/2023-2024/Архитектура компьютера$ git clone --recursive git@github.com:V9Jlb14/study_2023-2024_arh-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), 16.97 Киб | 188.00 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 «/home/slavarudan/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 | 1.16 Миб/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/slavarudan/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 | 279.00 Киб/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be380ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
slavarudan@slavarudan-VirtualBox: /work/study/2023-2024/Архитектура компьютера$
```

Рисунок 16(Клонированный репозиторий)

Настройка каталога курса

Перейдите в каталог курса:

Используем команду **cd**

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

Рисунок 17(Переходим в каталог курса)

Удалите лишние файлы:

Удаляем файл package.json с помощью команды **rm**.

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

Рисунок 18(Удаляем файл)

Создайте необходимые каталоги:

Создаем каталог COURSE

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

Рисунок 19(Создаем необходимый каталог)

Отправьте файлы на сервер:

Команда **git add** . используется в системе контроля версий Git для добавления всех измененных, новых и удаленных файлов в текущей директории и её поддиректориях в индекс. Файлы, находящиеся в индексе, готовы к фиксации (commit) в репозитории Git.

Команда **git commit -am** используется для создания коммита в Git, при этом она позволяет добавить изменения в индекс и сразу же выполнить коммит, при этом можно опустить этап выполнения **git add** для файлов, которые уже были отслеживаемыми (уже находятся в индексе).

```
podskaska: ~$ git config advice.addemptypathspec false
slavarudan@slavarudan-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
slavarudan@slavarudan-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master fb880531 feat(main): make course structure]
```


Рисунок 20(Ввод команд)

Команда **git push** используется в системе контроля версий Git для отправки (загрузки) изменений из вашего локального репозитория на удаленный репозиторий. Это позволяет синхронизировать ваши локальные изменения с центральным репозиторием или с удаленным сервером Git

```
create mode 100755 labs/lab11/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab11/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab11/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab11/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab11/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab11/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab11/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab11/report/report.md
delete mode 100644 package.json
create mode 100644 prepare
create mode 100644 presentation/README.md
create mode 100644 presentation/README.ru.md
create mode 100644 presentation/presentation/Makefile
create mode 100644 presentation/presentation/image/kulyabov.jpg
create mode 100644 presentation/presentation/presentation.md
create mode 100644 presentation/report/Makefile
create mode 100644 presentation/report/bib/cite.bib
create mode 100644 presentation/report/image/placeimg_800_600_tech.jpg
create mode 100644 presentation/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 presentation/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_fignos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_secnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/core.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/main.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 presentation/report/report.md
slavarudan@slavarudan-VirtualBox:~/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 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.59 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:V9JlbI4/study_2023-2024_arh-pc.git
 156c4d0..fb88053 master -> master
slavarudan@slavarudan-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```


Рисунок 21(Отправка файлов)

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


study_2023-2024_arh-pc
Private
Unwatch 1

generated from [yamadhama/course-directory-student-template](#)

master
1 branch
0 tags
Go to file
Add file
Code


V9Jlb14 feat(main): make course structure
 fb88053 1 minute ago
2 commits

config	Initial commit	22 minutes ago
labs	feat(main): make course structure	1 minute ago
presentation	feat(main): make course structure	1 minute ago
template	Initial commit	22 minutes ago
.gitattributes	Initial commit	22 minutes ago
.gitignore	Initial commit	22 minutes ago
.gitmodules	Initial commit	22 minutes ago
CHANGELOG.md	Initial commit	22 minutes ago
COURSE	feat(main): make course structure	1 minute ago
LICENSE	Initial commit	22 minutes ago
Makefile	Initial commit	22 minutes ago
README.en.md	Initial commit	22 minutes ago
README.git-flow.md	Initial commit	22 minutes ago
README.md	Initial commit	22 minutes ago
prepare	feat(main): make course structure	1 minute ago


README.md


Рисунок 22(Проверка иерархии)

Выполнение задания для самостоятельной работы

1. **Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).**
2. **Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.**

Команда `mv` в операционной системе Linux предназначена для перемещения или переименования файлов и директорий.

Рисунок 22(Переносим ЛР1 в каталог репорт)



```
slavarudan@slavarudan-VirtualBox:~$ cd work/study/2023-2024/"Архитектура компьютера"/arch-pc
slavarudan@slavarudan-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
slavarudan@slavarudan-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -m 'LR1_Rudan.pdf'
```

Рисунок 22(Подготавливаем файлы)

3. Загрузите файлы на github.

Загружаем ЛР1 и ЛР2.

```
slavarudan@slavarudan-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 10, готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 622.52 Киб | 3.21 Миб/с, готово.
Всего 6 (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:V9JlbI4/study_2023-2024_arh-pc.git
```

Рисунок 22(Загружаем файлы)

Проверяем наши файлы на github в созданном репозитории.

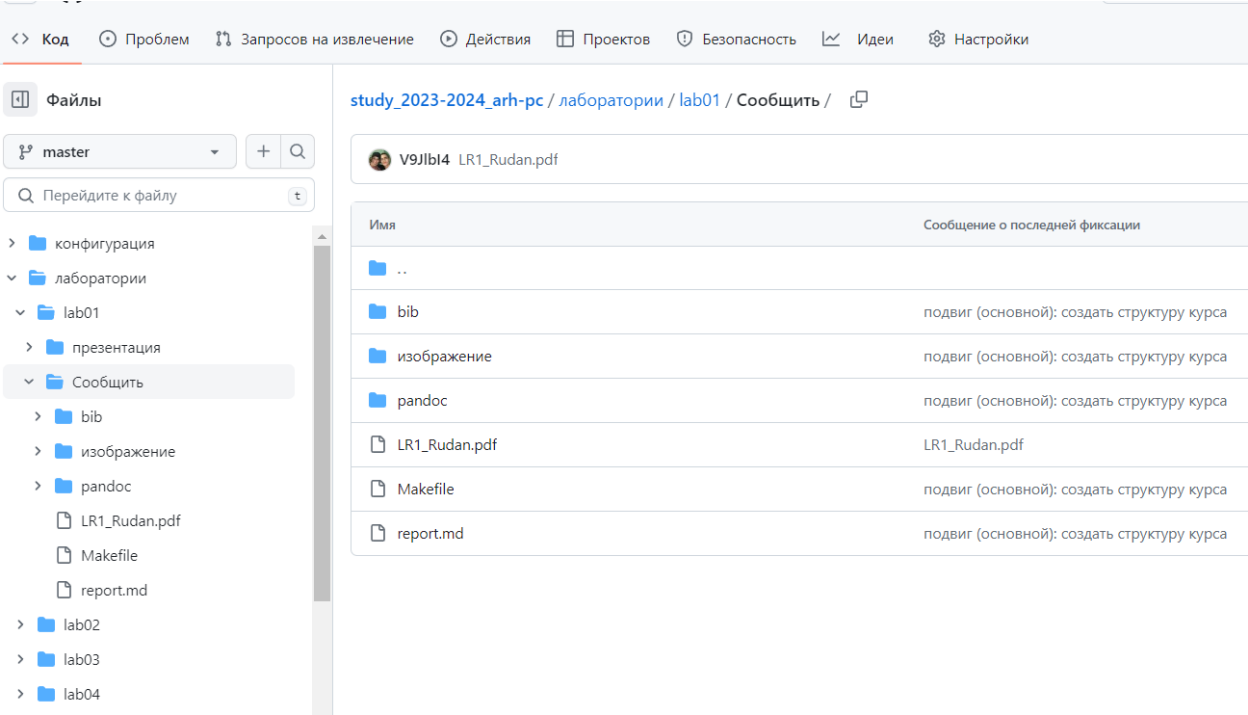


Рисунок 22(ЛР1 загруженная на github lab01/reports)

Заключение

Завершая данное исследование, следует отметить, что работа была выполнена с использованием операционной системы Linux и терминала, что способствовало более глубокому пониманию особенностей системы контроля версий Git. Целью данной работы было изучение идеологии и практическое применение средств контроля версий с акцентом на Git, а также приобретение практических навыков в этой области.

В процессе исследования были рассмотрены основные принципы контроля версий, преимущества его использования, а также основные команды и операции, доступные в системе Git.

Заключительное мнение - данная работа позволила успешно достичь поставленной цели и обогатить профессиональный багаж практическими навыками работы с Git, что, безусловно, будет полезно в будущей работе