

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2.

дисциплина: Архитектура компьютера

Студент: Доронин Никита Максимович

Группа: НКАбд-02-23

Цель работы

Цель работы состоит в изучении идеологии и применения средств контроля версий, а также получении практических навыков по работе с системой git.

Ход работы

Для начала работы с Git требуется регистрация на сайте. Далее я установил GitHub desktop.

Welcome to GitHub Desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Sign in below to get started with your existing projects.

Sign in to GitHub.com

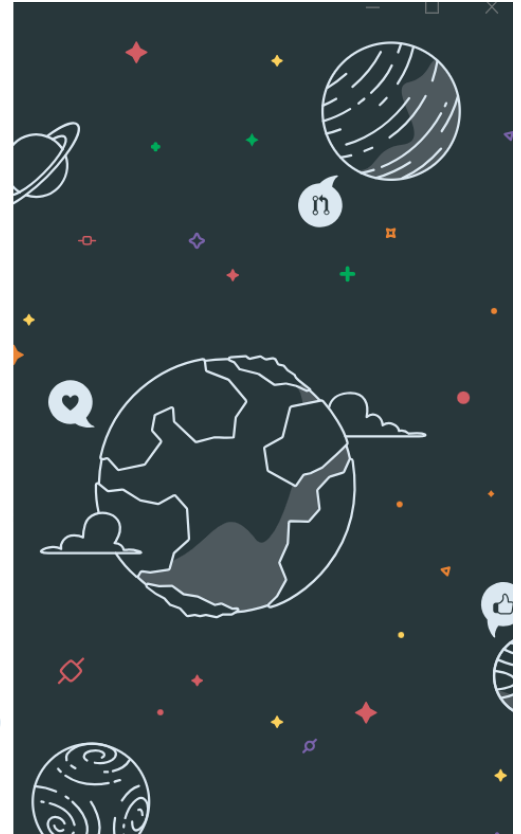
Sign in to GitHub Enterprise

New to GitHub? [Create your free account.](#)

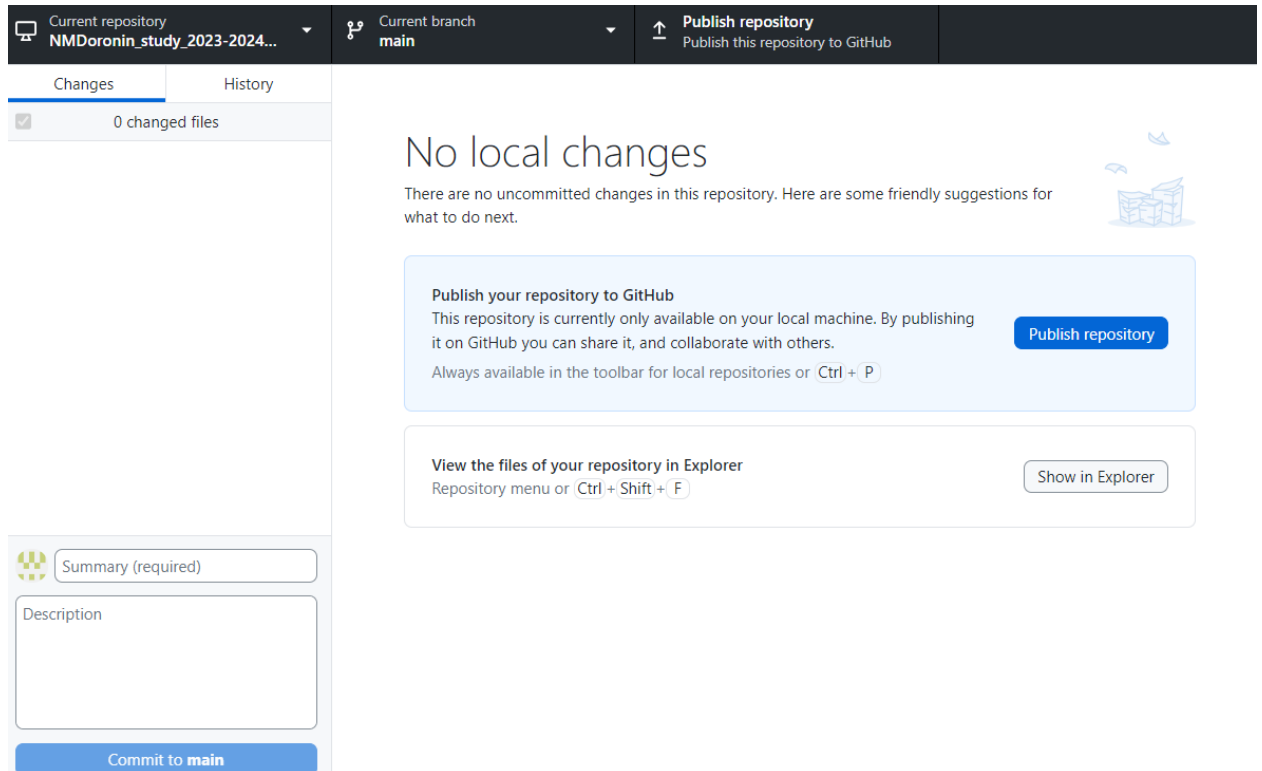
Skip this step

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#).

GitHub Desktop sends usage metrics to improve the product and inform feature decisions. [Learn more about user metrics.](#)



После этого на GitHub нужно найти шаблонный репозиторий и создать свой.



После создания репозитория, предварительно настраиваем git, генерируем пару ключей идентификации и добавляем их на GitHub.

```
Nikita@DESKTOP-DL901K3 MINGW64 ~  
$ git config --global user.name "WasBeJin"  
  
Nikita@DESKTOP-DL901K3 MINGW64 ~  
$ git config --global user.email "nikita.doronin.2005@mail.ru"  
  
Nikita@DESKTOP-DL901K3 MINGW64 ~  
$ git config --global core.quotepath false  
  
Nikita@DESKTOP-DL901K3 MINGW64 ~  
$ git config --global init.defaultBranch master  
  
Nikita@DESKTOP-DL901K3 MINGW64 ~  
$ git config --global core.autocrlf input  
  
Nikita@DESKTOP-DL901K3 MINGW64 ~  
$ git config --global core.safecrlf warn
```

```

Nikita@DESKTOP-DL901K3 MINGW64 ~
$ ssh-keygen -C "Никита Доронин 1132232880@pfur.ru"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Nikita/.ssh/id_ed25519):
Created directory '/c/Users/Nikita/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Nikita/.ssh/id_ed25519
Your public key has been saved in /c/Users/Nikita/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:85EkTBKdudIytZfYDUkwBzcp4tBUraoDqHgfr7M5fzM Никита Доронин 1132232880@pfur.ru
The key's randomart image is:
+--[ED25519 256]--+
|      o++*B=o      |
|    . o+*+=.      |
|  o +oB.+         |
|    = *o+..        |
|  .   o=S.o        |
|.. . ..+ o .      |
|o   ...= .         |
|o . oo o E         |
| . ... .. o        |
+-----[SHA256]-----+

```

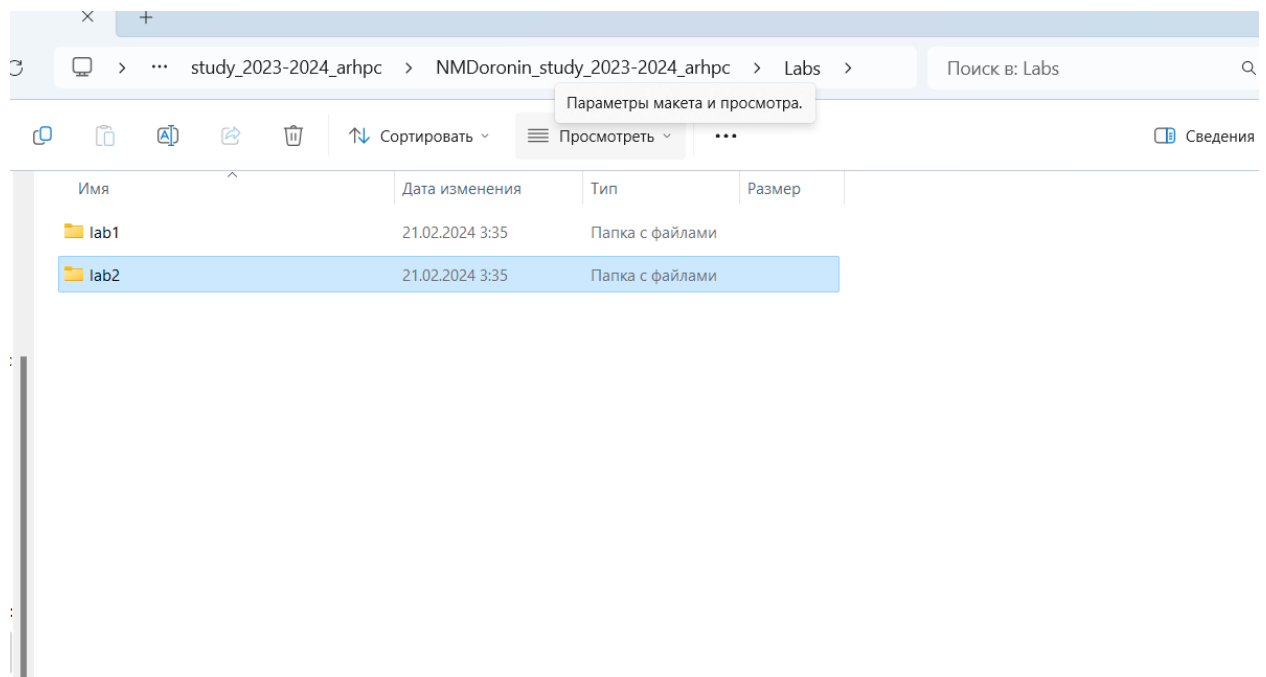
```

Nikita@DESKTOP-DL901K3 MINGW64 ~
$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILQrPsrHGjqvk/5+UyrA3c1NTQBx72TEXrXt+1mn5fJW
Никита Доронин 1132232880@pfur.ru

```

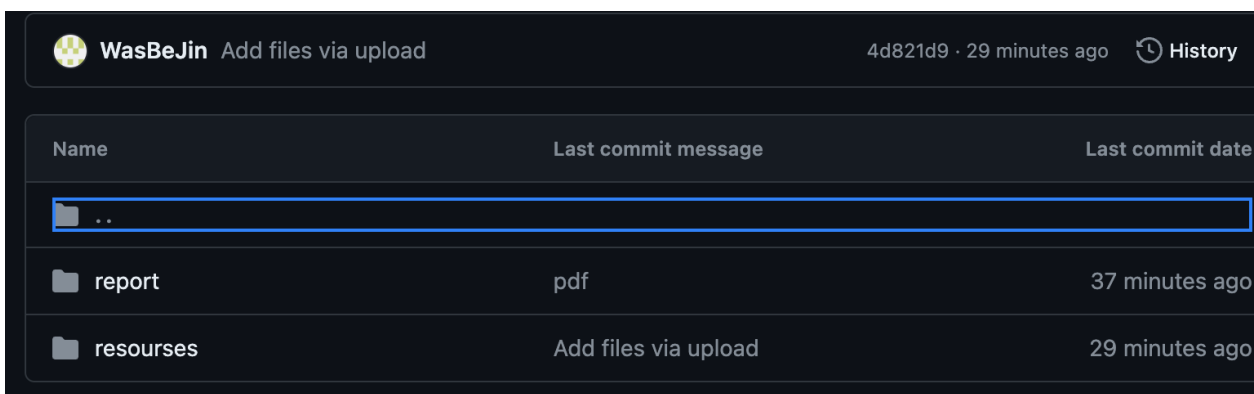
Затем создаем рабочий каталог и добавляем его на GitHub.

Для создания отчета по лабораторной работе в каталоге рабочего пространства (labs>lab02>report) нужно создать каталог "report" внутри "lab02", а затем создать файл отчета внутри этого каталога.



Задание для самостоятельной работы

Для копирования отчетов по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства нужно создать каталоги "lab01" и "lab02", если они еще не существуют, а затем скопировать соответствующие файлы отчетов внутрь этих каталогов.



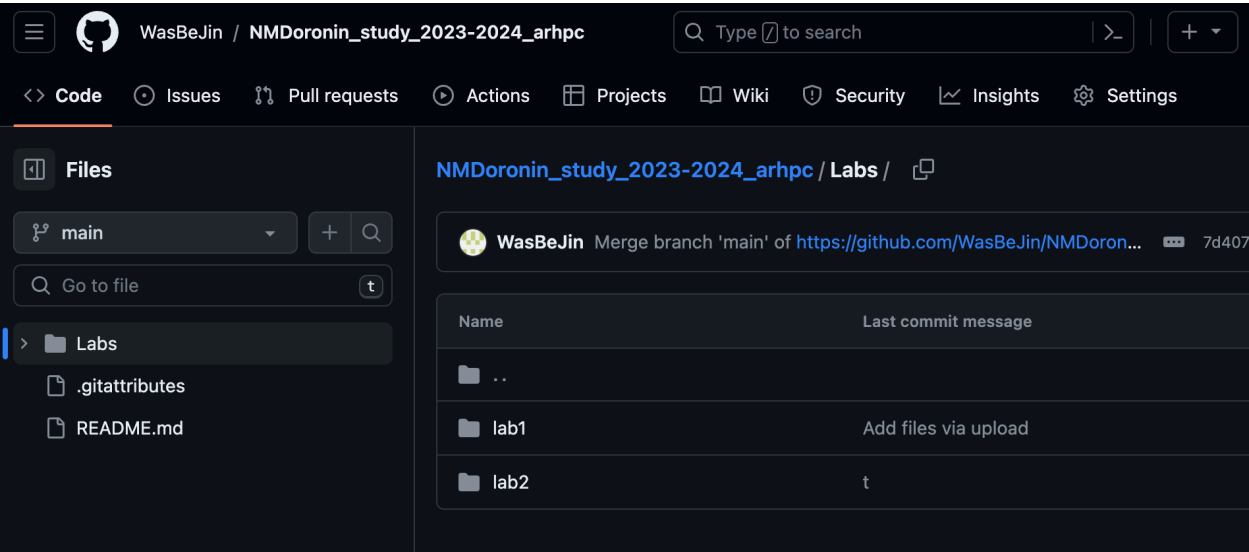
The screenshot shows a GitHub repository interface for a user named 'WasBeJin'. At the top, there is a header bar with the repository name, a button 'Add files via upload', a commit hash '4d821d9', a timestamp '29 minutes ago', and a 'History' link. Below the header is a table with three columns: 'Name', 'Last commit message', and 'Last commit date'. The table contains three rows: a parent directory '..' (highlighted with a blue border), a folder 'report' with a commit message 'pdf' and a timestamp '37 minutes ago', and a folder 'resources' with a commit message 'Add files via upload' and a timestamp '29 minutes ago'.

Name	Last commit message	Last commit date
..		
report	pdf	37 minutes ago
resources	Add files via upload	29 minutes ago

Для загрузки файлов на GitHub необходимо создать новый репозиторий на сайте GitHub, затем связать его с локальным каталогом рабочего пространства и выполнить команды `git add`, `git commit` и `git push`, чтобы добавить, зафиксировать и загрузить файлы на GitHub.

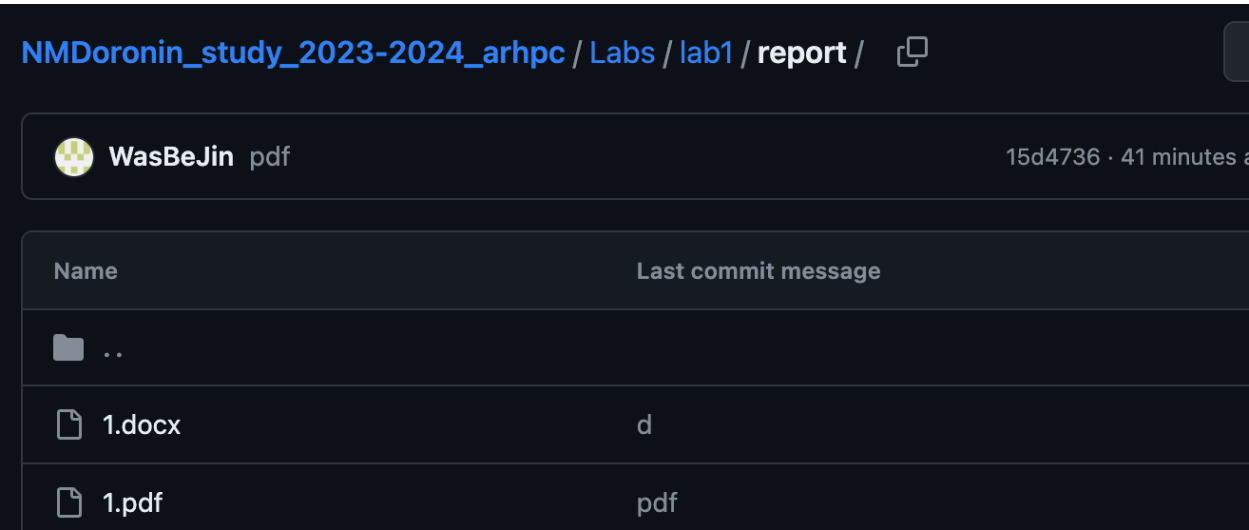
Для выполнения задания по созданию отчета по выполнению лабораторной работы, следует создать отчет исходя из информации, полученной в ходе выполнения лабораторной работы, включая полученные результаты и

ВЫВОДЫ.



Для реализации задачи по копированию отчетов предыдущих лабораторных работ, нужно определить местоположение этих отчетов и скопировать их в соответствующие каталоги нового рабочего пространства.

Для успешной загрузки файлов на GitHub, следует удостовериться, что они были добавлены и зафиксированы в локальном репозитории, а затем загрузить их на сервер GitHub с помощью команды `git push`.



Выводы

В результате выполнения работы были получены навыки работы с системой контроля версий GitHub.

Контрольные вопросы для самопроверки

что такое системы контроля версий (VCS) и для решения каких задач они предназначены?

Системы контроля версий (VCS) — это программные инструменты, предназначенные для управления изменениями в файловой системе и их сохранения, позволяя отслеживать изменения в файлах, возвращаться к предыдущим версиям, объединять изменения из разных веток и т. д. Они предназначены для облегчения совместной работы над одним проектом, а также для обеспечения сохранности истории изменений.

Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Хранилище (репозиторий) - это место, где сохраняются данные истории изменений файлов.

Commit (коммит) - это сохранение изменений в файловой системе в VCS, которое позволяет создавать и сохранять изменения в файлы и работать с ними в будущем.

История (лог) - это список всех предыдущих коммитов, выполненных в рамках проекта.

Рабочая копия - это копия проекта из хранилища, которая отражает текущее состояние проекта на момент последнего коммита.

Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные VCS - это системы, в которых есть одно центральное хранилище, куда все пользователи отправляют свои изменения и откуда получают обновления. Пример: SVN.

Децентрализованные VCS - это системы, в которых каждый пользователь имеет свою собственную копию хранилища, называемую репозиторием, и может работать с ней независимо от других пользователей. Пример: Git.

Опишите действия с VCS при единоличной работе с хранилищем.

Создание локального хранилища (репозитория) с помощью команды `git init`.

Добавление файлов в хранилище с помощью команды `git add`.

Сохранение изменений в хранилище с помощью команды `git commit`.

Просмотр истории изменений с помощью команды `git log`.

Возврат к предыдущим версиям с помощью команды `git checkout`.

Опишите порядок работы с общим хранилищем VCS.

Сначала пользователь загружает (pulls) изменения из общего хранилища на свой компьютер с помощью команды `git pull`.

Затем пользователь вносит изменения в файлы, работая с ними на своем компьютере.

После завершения работы пользователь отправляет (pushes) изменения обратно в общее хранилище с помощью команды `git push`.

Каковы основные задачи, решаемые инструментальным средством `git`?

Создание локальных репозиториев.

Управление изменениями истории репозиториев.

Работа с ветками и объединение изменений из разных веток.

Управление удаленными репозиториями (загрузка и загрузка данных на их).

Назовите и дайте краткую характеристику командам `git`.

`git init` - создает новый локальный репозиторий.

`git add` - добавляет файлы в индекс (подготавливает их к коммиту).

`git commit` - сохраняет изменения, сделанные в индексе, в репозиторий.

`git pull` - загружает изменения из удаленного репозитория в локальный.

`git push` - отправляет изменения из локального репозитория в удаленный.

`git log` - выводит историю изменений в репозитории.

Приведите примеры использования при работе с локальным и удаленным репозиториями.

Локальный репозиторий: `git init` (создание нового репозитория), `git add` (добавление файлов), `git commit` (сохранение изменений).

Удаленный репозиторий: `git pull` (загрузка изменений), `git push` (загрузка изменений).

Список литературы

1. Официальная документация Git: <https://git-scm.com/doc>
2. Про Git на русском языке: <https://git-scm.com/book/ru/v2>
3. Официальный сайт GitHub: <https://github.com>
4. GitHub Guides: <https://guides.github.com>
5. Официальная документация GitHub: <https://docs.github.com>