

# 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: Байрамов Керим Сапарович

Группа: НКАбд-02-24

МОСКВА

2024г.

# Содержание

1. Цель работы.....	4
2. Задание .....	5
3. Теоретическое введение .....	6
4. Выполнение лабораторной работы .....	8
4.1. Настройка GitHub. ....	8
4.2. Базовая настройка git.....	8
4.3. Создание SSH ключа .....	9
4.4. Создание рабочего пространства, репозитория курса на основе шаблона. ....	11
4.5. Создание репозитория курса на основе шаблона.....	11
4.6. Настройка каталога курса. ....	13
4.7. Выполнение заданий для самостоятельной работы.....	14
5. Заключение.....	17
6. Список используемой литературы .....	18

## Список иллюстраций

Рис.3.1 . Перечень основных команд git .....	7
рис.4.1.1. Учетная запись на GitHub .....	8
рис.4.2.1. Предварительная конфигурация в git .....	8
рис.4.2.2. Настройка utf-8. ....	9
рис.4.2.3. Имя начальной ветки.....	9
рис.4.2.4. Ввод команд autocrlf и safecrlt.....	9
рис.4.3.1. Генерация ключей. ....	9
рис.4.3.2. Скачивание команды xclip. ....	10
рис.4.3.3. Копирование ключа. ....	10
рис.4.3.4. Вставка ключа.....	11
рис.4.4.1. Создание терминала для предмета «Архитектура компьютера».....	11
рис.4.5.1. Выбор шаблона.....	12
рис.4.5.2. Создание репозитория.....	12
рис.4.5.2. Переход в каталог курса. ....	12
рис.4.5.4. Клонирование репозитория.....	13
рис.4.5.5. Удаление лишних файлов в каталоге курса.....	13
рис.4.5.6. Создание каталогов и их отправка на сервер. ....	13
рис.4.5.7. Проверка выполненной работы. ....	14
рис.4.7.1. Создание отчета о выполнении работы. ....	14
рис.4.7.2. Местонахождение лабораторных работ .....	14
рис.4.7.3. Копирование отчета по лабораторной работе в нужный каталог .....	15
рис.4.7.4. Добавление файлов с помощью команды git add .....	15
рис.4.7.5. Поручаю консоли совершить изменения.....	15
рис.4.7.6. Команды git status и git push для завершения копирования.....	15
рис.4.7.7. Проверка проделанных операций .....	16

## **1. Цель работы**

Целью работы является применение средств контроля версий. А также очень важно 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

Система контроля версий ( Version Control System, VCS ) — это инструмент, используемый разработчиками программного обеспечения для управления изменениями в исходном коде и других файловых ресурсах.

Системы контроля версий разработаны специально для того, чтобы максимально упростить и упорядочить работу над проектом (вне зависимости от того, сколько человек в этом участвуют). СКВ дает возможность видеть, кто, когда и какие изменения вносил; позволяет формировать новые ветви проекта, объединять уже имеющиеся; настраивать контроль доступа к проекту ; осуществлять откат до предыдущих версий.

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.

Демидова А. В. 14 Архитектура ЭВМ 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.

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.

Ниже показан перечень основных команд `git` (рис.3.1)

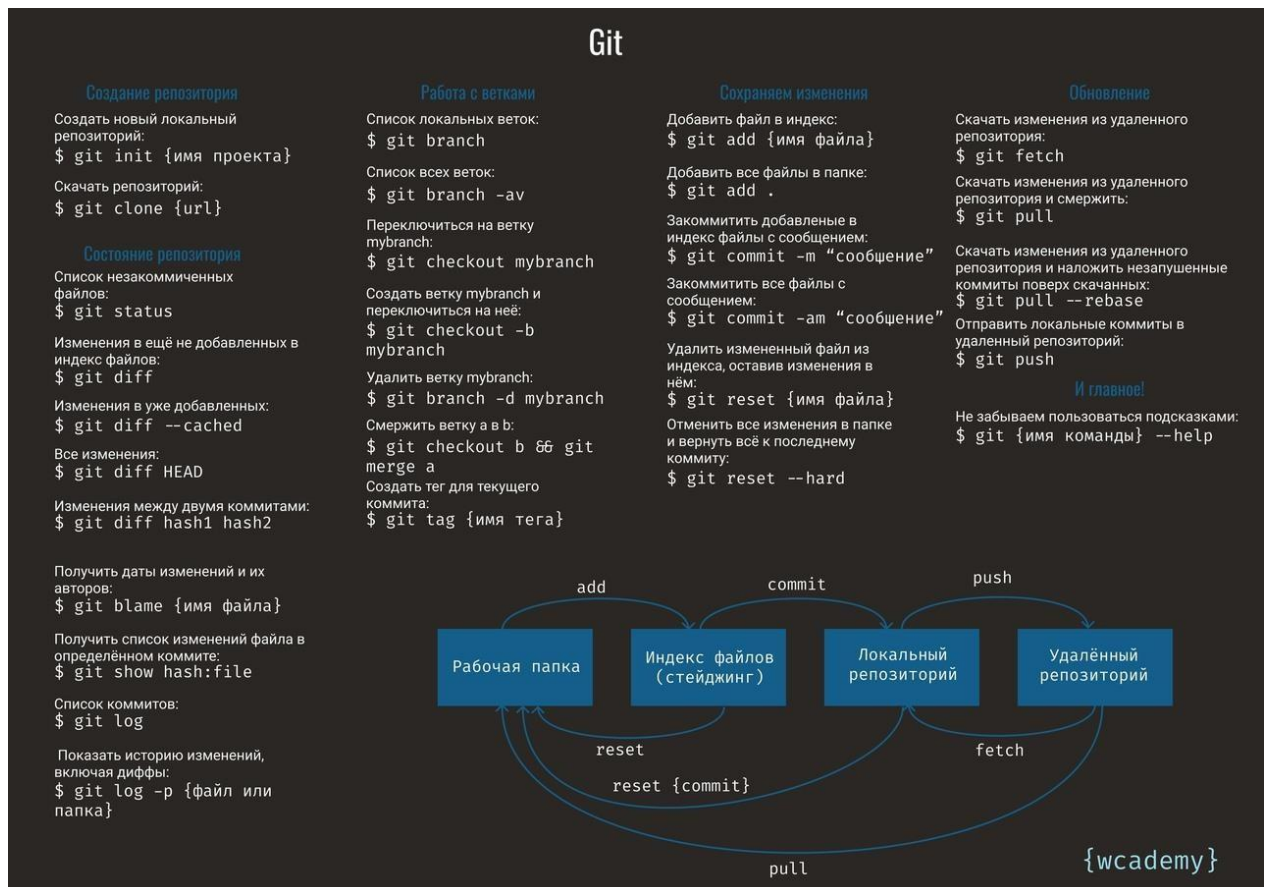


Рис.3.1 . Перечень основных команд `git`.

## 4. Выполнение лабораторной работы

### 4.1. Настройка GitHub.

Для выполнения лабораторной работы создаю учетную запись на <https://github.com/> (рис. 4.1.1)

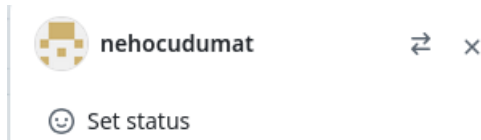


рис.4.1.1. Учетная запись на GitHub.

### 4.2. Базовая настройка `git`.

Делаю предварительную конфигурацию `git`. Захожу в терминал и ввожу команды, указывая свое имя и email (рис.4.2.1)

```
nehocudumat@nehocudumat-VirtualBox:~$ git config --global user.name "<Kerim Bayramow>"
nehocudumat@nehocudumat-VirtualBox:~$ git config --global user.email "<icanthearyoubaby@gmail.com>"
```

рис.4.2.1. Предварительная конфигурация в `git`.



Настраиваю utf-8 в выходе сообщений [git](#). (рис.4.2.2)

```
nehocudumat@nehocudumat-VirtualBox:~$ git config --global core.quotePath false
```

рис.4.2.2. Настройка utf-8.

Задаю имя начальной ветки, которую буду называть [master](#) (рис. 4.2.3)

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

рис.4.2.3. Имя начальной ветки

А также ввожу [autocrlf](#) и [safecrlf](#) (рис. 4.2.4)

```
nehocudumat@nehocudumat-VirtualBox:~$ git config --global core.autocrlf input
nehocudumat@nehocudumat-VirtualBox:~$ git config --global core.safecrlf warn
```

рис.4.2.4. Ввод команд `autocrlf` и `safecrlf`.

### 4.3. Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория генерирую пару ключей (приватный и открытый). (рис. 4.3.1)

```
nehocudumat@nehocudumat-VirtualBox:~$ ssh-keygen -C "Kerim Bayramow <icanthearyoubaby@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nehocudumat/.ssh/id_rsa):
/home/nehocudumat/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nehocudumat/.ssh/id_rsa
Your public key has been saved in /home/nehocudumat/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5tQuHZ8RU2ZrHCRMP6Ca5PSH2s1900PiiXdcnyuHWM Kerim Bayramow <icanthearyoubaby@gmail.com>
The key's randomart image is:
+---[RSA 3072]-----+
|      o o=0.*      |
|      + = .+* +    |
|      + o +.=..    |
|      + + E...     |
|      S + +.0.0    |
|      + o oo=+.+    |
|      o o.=+0+.    |
|      . ..0...     |
|      o.           |
+---[SHA256]-----+
nehocudumat@nehocudumat-VirtualBox:~$
```

рис.4.3.1. Генерация ключей.

Чтобы скопировать из локальной консоли ключ в буфер обмена, устанавливаю команду `xclip` (рис. 4.3.2).

```
nehocudumat@nehocudumat-VirtualBox:~$ sudo apt-get install -y xclip
[sudo] password for nehocudumat:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
xclip is already the newest version (0.13-2).
0 upgraded, 0 newly installed, 0 to remove and 81 not upgraded.
nehocudumat@nehocudumat-VirtualBox:~$
```

*рис.4.3.2. Скачивание команды xclip.*

Теперь воспользуюсь командой `xclip` (рис 4.3.3)

```
nehocudumat@nehocudumat-VirtualBox:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

*рис.4.3.3. Копирование ключа.*


Вставляю ключ в появившееся на сайте поле, указываю его имя. (рис. 4.3.4)

## 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



**Title**  
SHA256:5tQuHZ8RU2ZrHCRMPGCa5PSH2s1900PiiXdcnyuHwBM  
Added on Sep 28, 2024  
Never used — Read/write

Delete

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

## GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

en 📄 🔊 🗑️ сен 28 19:19

рис.4.3.4. Вставка ключа.

## 4.4. Создание рабочего пространства, репозитория курса на основе шаблона.

Открываю терминал и создаю репозиторий для предмета «Архитектура компьютера». (рис. 4.4.1.)

```
nehocudumat@nehocudumat-VirtualBox:~$ mkdir -p work/study/2024-2025/"Архитектура Компьютера"
nehocudumat@nehocudumat-VirtualBox:~$ ls
Desktop  Downloads  Music      parentdir1  parentdir3  Public  Templates  text2.txt  tmp      work
Documents  labs       parentdir  parentdir2  Pictures     temp    text1.txt  text3.txt  Videos
```

рис.4.4.1. Создание терминала для предмета «Архитектура компьютера».

## 4.5. Создание репозитория курса на основе шаблона.

Захожу на страницу репозитория с шаблоном курса, выбираю его в качестве своего нового. (рис. 4.5.1.)

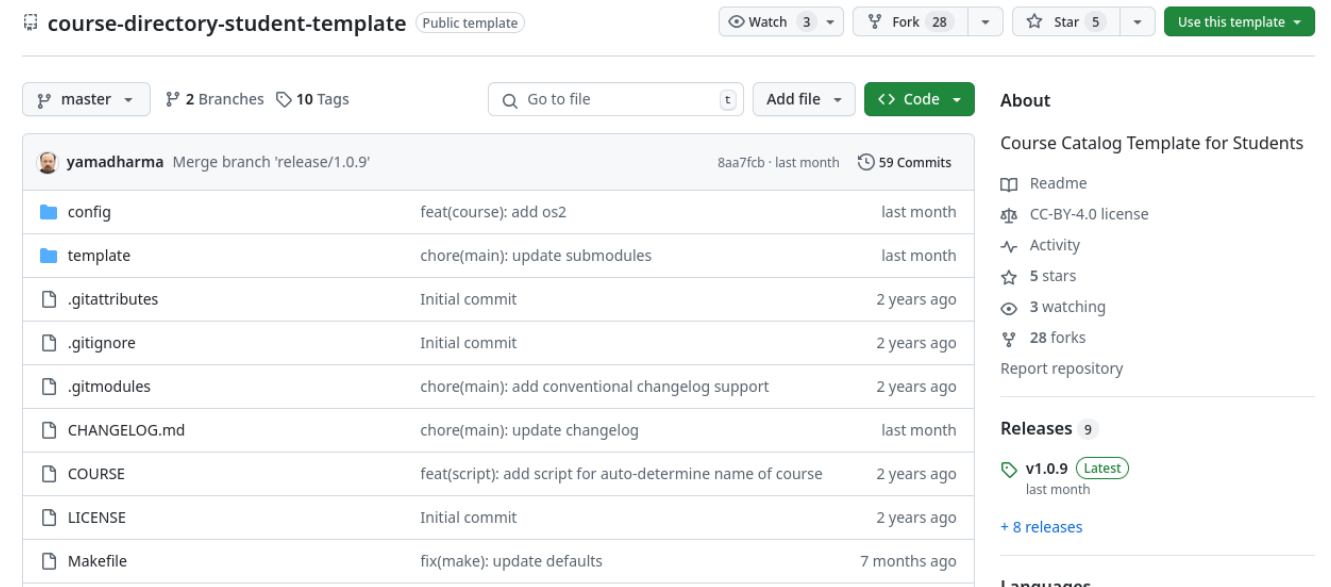


рис.4.5.1. Выбор шаблона

## 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 (\*).

### Repository template

yamadharm/course-directory-student-template

Start your repository with a template repository's contents.

☐ Include all branches

Copy all branches from yamadharm/course-directory-student-template and not just the default branch.

### Owner \*

nehocudumat

### Repository name \*

study\_2024-2025\_arch-pc

study\_2024-2025\_arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about [upgraded-octo-waffle](#) ?

### 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.

Далее создаю его, задав ему имя. (рис. 4.5.2.)

рис.4.5.2. Создание репозитория.

Открываю терминал и перехожу в каталог курса. (рис. 4.5.3).

```
nehocudumat@nehocudumat-VirtualBox:~$ cd ~/work/study/2024-2025/"Архитектура компьютера"
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера$
```

рис.4.5.2. Переход в каталог курса.

## Клонирую созданный репозиторий (рис. 4.5.4.)

```
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:nehocudumat/study_2024-2025_arch-pc.git
arch-pc
Cloning into 'arch-pc'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 33 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (33/33), 18.81 KiB | 3.76 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into '/home/nehocudumat/work/study/2024-2025/Архитектура компьютера/arch-pc/template/presentation'...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (111/111), 102.17 KiB | 977.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/nehocudumat/work/study/2024-2025/Архитектура компьютера/arch-pc/template/report'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)
Receiving objects: 100% (142/142), 341.09 KiB | 1.94 MiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
```

рис.4.5.4. Клонирование репозитория.

## 4.6. Настройка каталога курса.

### Перехожу в каталог курса и удаляю лишний файлы. (рис. 4.5.5.)

```
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера$ cd ~/work/study/2024-2025/Архитектура\ компьютера/arch-pc
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
```

рис.4.5.5. Удаление лишних файлов в каталоге курса.

### Создаю необходимые каталоги, отправляю файлы на сервер. (рис. 4.5.6.)

```
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am "Feat(main): make course structure"
[master e9bbcb9] Feat(main): make course structure
223 files changed, 53681 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/.projectile
create mode 100644 labs/lab01/presentation/.texlabroot
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
```

рис.4.5.6. Создание каталогов и их отправка на сервер.

В локальном репозитории проверяю результат выполненной работы (рис.

4.5.7)

..		
lab01	Add lab01/report	24 minutes ago
lab02	Feat(main): make course structure	1 hour ago
lab03	Feat(main): make course structure	1 hour ago
lab04	Feat(main): make course structure	1 hour ago
lab05	Feat(main): make course structure	1 hour ago
lab06	Feat(main): make course structure	1 hour ago
lab07	Feat(main): make course structure	1 hour ago
lab08	Feat(main): make course structure	1 hour ago
lab09	Feat(main): make course structure	1 hour ago
lab10	Feat(main): make course structure	1 hour ago
lab11	Feat(main): make course structure	1 hour ago

рис.4.5.7. Проверка выполненной работы.

## 4.7. Выполнение заданий для самостоятельной работы.

Создаю отчет по выполнению второй лабораторной работы в соответствующем каталоге. (рис. 4.7.1) С помощью команды `ls` проверяю, создан ли файл.

```
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd labs/lab02/report
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ touch Л02_Керим_отчет
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ ls
bib image Makefile pandoc report.md Л02_Керим_отчет
```

рис.4.7.1. Создание отчета о выполнении работы.

Для выполнения второго задания проверяю местонахождение своих лабораторных работ. (рис. 4.7.2)

```
nehocudumat@nehocudumat-VirtualBox:~$ ls Downloads
'Байрамов Керим НКАбд 02-24.Первая лабораторная работа по Архитектуре данных (1).docx'  'Лабораторная работа №2. Система контроля версий Git.pdf'
'Байрамов Керим НКАбд 02-24.Первая лабораторная работа по Архитектуре данных (1).pdf'
```

рис.4.7.2. Местонахождение лабораторных работ.

Копирую лабораторную работу с помощью утилиты `cp`. (рис.4.7.3)

`ls`

```
nehocudumat@nehocudumat-VirtualBox:~/Downloads$ cp Л01_Керим_отчет.pdf /home/nehocudumat/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab01/report
nehocudumat@nehocudumat-VirtualBox:~/Downloads$ cp Л01_Керим_отчет.docx /home/nehocudumat/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab01/report
```

рис.4.7.3. Копирование отчета по лабораторной работе в нужный каталог.

Для того чтобы загрузить эти файлы на GitHub, в первую очередь я использую команду `git add`. Так добавленные мной файлы станут отслеживаемые (рис.4.7.4)

```
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git add Л01_Керим_отчет.pdf
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git add Л01_Керим_отчет.docx
```

рис.4.7.4. Добавление файлов с помощью команды `git add`

Теперь осуществляю полноценный перенос файлов с помощью команды `git commit -m "..."` (рис. 4.7.5.)

```
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git commit -m "Add lab01/report"
[master 60a819f] Add lab01/report
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Керим_отчет.docx
create mode 100644 labs/lab01/report/Л01_Керим_отчет.pdf
```

рис.4.7.5. Поручаю консоли совершить изменения.

Использую команды: `git status` и `git push`, чтобы опубликовать свои локальные коммиты. (рис. 4.7.6.)

```
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  ../../lab02/report/Л02_Керим_отчет

nothing added to commit but untracked files present (use "git add" to track)
nehocudumat@nehocudumat-VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 10 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 3.24 MiB | 622.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:nehocudumat/study_2024-2025_arch-pc.git
e9bbcb9..60a819f master -> master
```

рис.4.7.6. Команды `git status` и `git push` для завершения копирования.

Перехожу в каталоги на GitHub, чтобы убедиться в том, что файлы находятся в нужных репозиториях. (рис. 4.7.7)

Bayramow Kerim Add lab01/report60a819f · 41 minutes agoHistory

Name	Last commit message	Last commit date
..		
bib	Feat(main): make course structure	1 hour ago
image	Feat(main): make course structure	1 hour ago
pandoc	Feat(main): make course structure	1 hour ago
Makefile	Feat(main): make course structure	1 hour ago
report.md	Feat(main): make course structure	1 hour ago
Л01_Керим_отчет.docx	Add lab01/report	41 minutes ago
Л01_Керим_отчет.pdf	Add lab01/report	41 minutes ago

рис.4.7.7. Проверка проделанных операций.



## **5. Заключение**

В заключение хочется отметить, что данная лабораторная работа позволила мне научиться работать с системой Git. Я практиковал свои навыки в работе с командной строкой, теперь уже связывая выполнимое с директориями GitHub.

## **6. Список используемой литературы**

1. Архитектура ЭВМ

[https://esystem.rudn.ru/pluginfile.php/2089082/mod\\_resource/content/0/Лабораторная%20работа%20№2.%20Система%20контроля%20версий%20Git.pdf](https://esystem.rudn.ru/pluginfile.php/2089082/mod_resource/content/0/Лабораторная%20работа%20№2.%20Система%20контроля%20версий%20Git.pdf)

2. 30 команд Git, необходимых для освоения интерфейса командной строки Git / Хабр <https://habr.com/ru/companies/ruvds/articles/599929/>

3. Система контроля версий: определение, функции, популярные решения <https://gb.ru/blog/sistema-kontrolya-versij/>