

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: Кудякова С. А.

Группа: НКАбд-03-23

МОСКВА

2023 г.

Содержание

1 Цель работы.....	3
2 Задание.....	4
3 Теоретическое введение.....	5
4 Выполнение лабораторной работы.....	7
5 Выводы.....	17
6 Список литературы.....	18

1 Цель работы

Цель данной работы - изучить идеологию и применение средств контроля версий, а также научиться работать 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 (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. 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.

4 Выполнение лабораторной работы

4.1. Настройка GitHub

Создаю учётную запись на сайте <https://github.com/> и заполняю основные данные (рис.1).

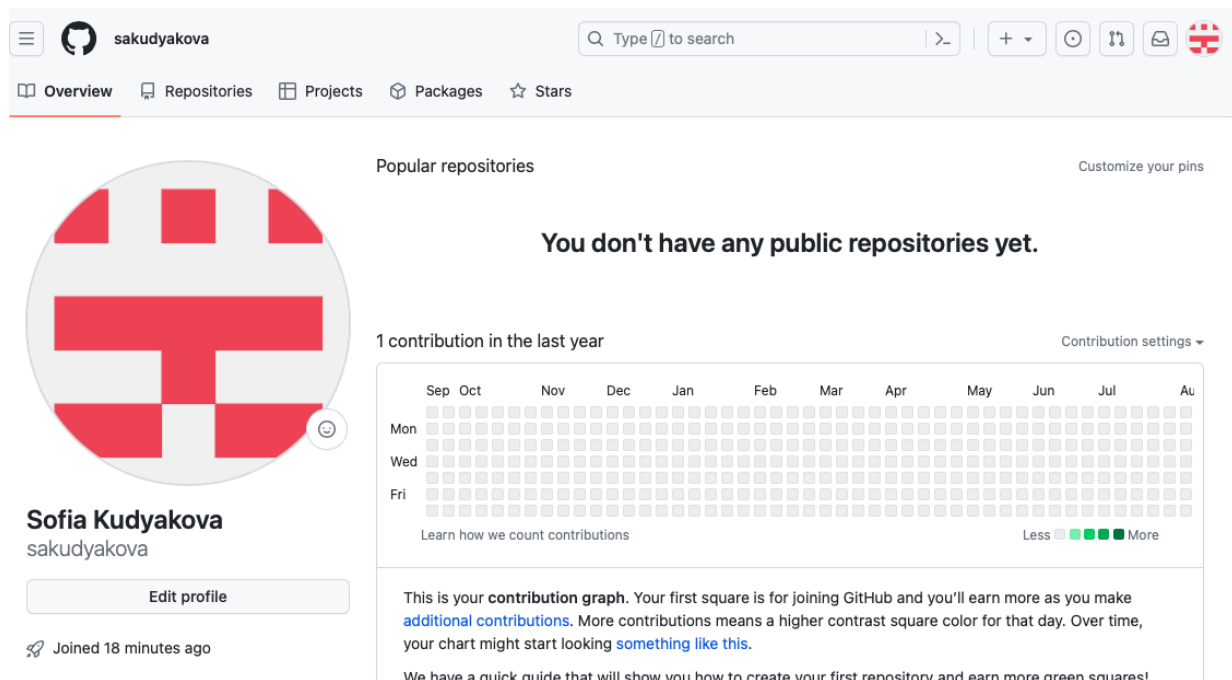


Рис. 1. Создание учетной записи GitHub

4.2. Базовая настройка Git

Делаю предварительную конфигурацию git. Открываю терминал и ввожу следующие команды, указав имя и email своего репозитория. (рис. 2)

```
sakudyakova@camuna59:~  
[sakudyakova@camuna59 ~]$ git config --global user.name "<Sofia Kudyakova>"  
[sakudyakova@camuna59 ~]$ git config --global user.email "<1132236033@pfur.ru>"  
[sakudyakova@camuna59 ~]$
```

Рис. 2. Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git (рис. 3).

```
sakudyakova@camuna59 ~]$ git config --global core.quotePath false
sakudyakova@camuna59 ~]$
```

Рис. 3. Настройка utf-8 в выводе сообщений git

Задаю имя начальной ветки (будем называть её master). Далее устанавливаю параметры autocrlf и safecrlf (рис. 4).

```
[sakudyakova@camuna59 ~]$ git config --global init.defaultBranch master
[sakudyakova@camuna59 ~]$ git config --global core.autocrlf input
[sakudyakova@camuna59 ~]$ git config --global core.safecrlf warn
[sakudyakova@camuna59 ~]$
```

Рис. 4. Создание имени для начальной ветки. Параметры autocrlf и safecrlf.

4.3. Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый) (рис. 5). Ключи сохранятся в каталоге ~/.ssh/

```
[sakudyakova@camuna59 ~]$ ssh-keygen -C"Sofia Kudyakova, 1132236033@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sakudyakova/.ssh/id_rsa):
Created directory '/home/sakudyakova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sakudyakova/.ssh/id_rsa
Your public key has been saved in /home/sakudyakova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:sC+3c+GMP1Aycaez23PcQ7qsS13BBM45CkaVLZW0nNk Sofia Kudyakova, 1132236033@pfur.ru
The key's randomart image is:
+---[RSA 3072]---+
|      ...oo+o. |
|      o ..+.+. |
|      . = o.=*o |
|      o+ = .=.E. |
|      . S+ +   . |
|      .. o . . . |
|      . o= = oo. |
|      oo.B +.o.. |
|      .+.+o=. . |
+-----[SHA256]-----+
```

Рис. 5. Создание SSH-ключа

Устанавливаю xclip (рис.6) и с помощью него копирую сгенерированный ключ (рис.7).

```
[sakudyakova@camuna59 ~]$ sudo dnf -y install xclip
[sudo] password for sakudyakova:
Last metadata expiration check: 0:05:28 ago on Sun 24 Sep 2023 12:13:37 PM EDT.
Dependencies resolved.
=====
Package Architecture Version Repository
Size
=====
Installing:
xclip x86_64 0.13-19.git11cba61.fc38 fedora
37 k
Transaction Summary
=====
Install 1 Package
```

Рис. 6. Установка xclip

```
[sakudyakova@camuna59 ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис. 7. Копирование с помощью xclip

Далее загружаю сгенерированный открытый ключ. Захожу на GitHub под своей учётной записью и перехожу в меню, выбираю в боковом меню “SSH and GPG keys”. Копирую ключ из локальной консоли в буфер обмена. Вставляю ключ в появившееся на сайте поле и указываю для ключа имя (Title) (рис. 8).

Sofia Kudryakova (sakudyakova)
Your personal account

Public profile
Account
Appearance
Accessibility
Notifications

Access
Billing and plans
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation

Add new SSH Key

Title
sakudyakova1

Key type
Authentication Key

Key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGCbmNaNv5qgPJalbvNk63txbcs4Yp2V7rXIE7fw3UGQBzPwSiKRFCTQML
72GuyStOHWh0H3+7w4IZ9gMXUXe70dRw4R4RGi3uvso0Iox9HnZkVYlaa2p9E8hFXjhVvcxxdnj0eLJCkw4Il6+v8k0
0hXcl1PaUWplol1b8qPqY40zMuHfQtBTuTDjpWbaWbAjszuZlpZHs/4UIhCeqM06GSptTD5i2+1Xm3h6tegy7B7NW
MvR0iTB8Yps+IP6gEEeSp8B 06mmJ2EZWLTAuqzCNxyRL+cv+RVqQkU/JvjllGA0oP5AHD7WwS9tGo+a
0DFB5HxoJ1OMc5+vV47teV8IbPHdGoKXb8G4m0LU/lzIW6bvsUVK+ENZJdVElyU6WIWSiVPaWA+8hTBuTGHKNpG
pFIT1ESlaYL3wW6X0Xgx15Qeim1blMqn1FPSFGxVli5uvV/AAF2VKM6r8RCXM/6FatJmp//Bp1W7nl3blJ3WYDb1ZXSjg
QPu7hA35g1Rptk= Sofia Kudryakova, 1132236033@pfur.ru

Add SSH key

Рис. 8. Добавление SSH key

4.4. Создание рабочего пространства и репозитория курса на основе шаблона

Открываю терминал и создаю каталог для предмета «Архитектура компьютера», используя команду `mkdir` с ее опцией `-p`, которая, в свою очередь, позволяет рекурсивно создать все директории. С помощью команды `ls` проверяю корректность совершенных действий (рис.9).

```
[sakudyakova@camuna59 ~]$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
[sakudyakova@camuna59 ~]$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  work
[sakudyakova@camuna59 ~]$
```

Рис. 9. Создание рабочего пространства

4.5. Создание репозитория курса на основе шаблона

Создаю репозиторий на основе шаблона через web-интерфейс GitHub.

Перехожу на страницу репозитория с шаблоном курса

<https://github.com/yamadharma/course-directory-student-template> и выбираю “Use this template” (рис. 10).

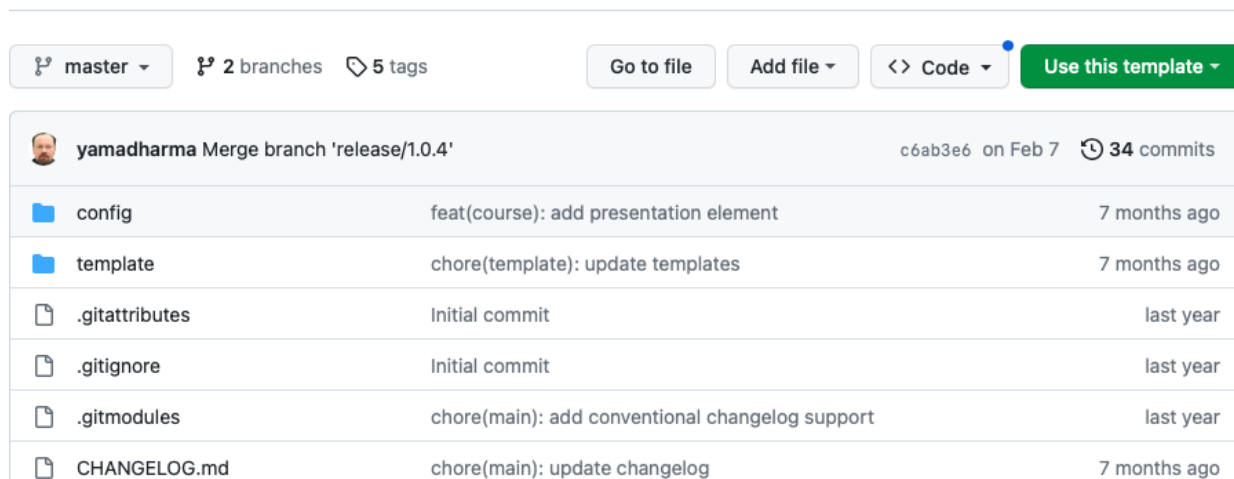


Рис. 10. Выбор шаблона


В открывшемся окне задаю имя репозитория (Repository name) study_2023–2024_arh-рс и создаю репозиторий, нажав на кнопку “Create repository” (рис. 11).

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 *


 sakudyakova / study_2023-2024_arh-pc

✔ study_2023-2024_arh-pc is available.

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

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.

① You are creating a public repository in your personal account.




[Create repository](#)

Рис. 11. Создание репозитория

Убеждаюсь в том, что репозиторий был создан (рис. 12).

 **study_2023-2024_arh-pc** Public Pin Unwatch 1

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

 master  1 branch  0 tags Go to file Add file Code









 sakudyakova	Initial commit	95dd6a7 now	 1 commit
 config	Initial commit		now
 template	Initial commit		now
 .gitattributes	Initial commit		now
 .gitignore	Initial commit		now
 .gitmodules	Initial commit		now
 CHANGELOG.md	Initial commit		now

Рис. 12. Созданный репозиторий

Далее открываю терминал и перехожу в каталог курса, используя команду `cd` (рис.13).

```
[sakudyakova@camuna59 ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"
[sakudyakova@camuna59 Архитектура компьютера]$
```

Рис. 13. Переход в каталог курса

Клонирую созданный репозиторий (рис. 15), предварительно скопировав ссылку на странице созданного репозитория Code -- SSH (рис. 14)

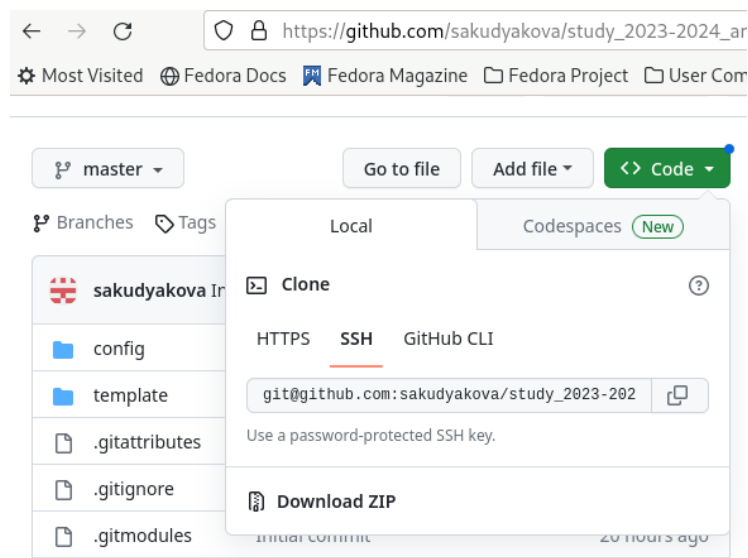


Рис. 14. Копирование ссылки

```
[sakudyakova@camuna59 Архитектура компьютера]$ git clone --recursive git@github.com:sakudyakova/study_2023-2024_arh-pc.git arch-pc
Cloning into '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
Receiving objects: 100% (27/27), 16.94 KiB | 4.23 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/sakudyakova/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
Receiving objects: 100% (82/82), 92.90 KiB | 1.13 MiB/s, done.
Resolving deltas: 100% (28/28), done.
Cloning into '/home/sakudyakova/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
Receiving objects: 100% (101/101), 327.25 KiB | 1.21 MiB/s, done.
Resolving deltas: 100% (40/40), done.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
```

Рис. 15. Клонирование репозитория

4.6. Настройка каталога курса

Перехожу в каталог курса (рис. 16).

```
[sakudyakova@camuna59 ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
[sakudyakova@camuna59 arch-pc]$
```

Рис. 16. Переход в каталог курса

С помощью команды `rm` удаляю лишние файлы, а именно файл “package.json” (рис. 17).

```
[sakudyakova@camuna59 arch-pc]$ rm package.json
[sakudyakova@camuna59 arch-pc]$
```

Рис. 17. Удаление файлов

Создаю необходимые каталоги (рис. 18).

```
[sakudyakova@camuna59 arch-pc]$ echo arch-pc > COURSE
[sakudyakova@camuna59 arch-pc]$ make
[sakudyakova@camuna59 arch-pc]$
```

Рис. 18. Создание необходимых каталогов

Отправляю файлы на сервер (рис. 19).

```
[sakudyakova@camuna59 arch-pc]$ git add .
[sakudyakova@camuna59 arch-pc]$ git commit -am 'feat(main): make course structure'
[master 9c68e2d] 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
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/placeimg_800_600_tech.jpg
```

```
[sakudyakova@camuna59 arch-pc]$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.13 KiB | 3.46 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:sakudyakova/study_2023-2024_arh-pc.git
 95dd6a7..9c68e2d master -> master
[sakudyakova@camuna59 arch-pc]$ 3
```

Рис. 19. Отправка файлов на сервер

Проверяю корректность создания иерархии рабочего пространства в локальном репозитории и на странице GitHub (рис. 20).


 sakudyakova feat(main): make course structure 9c68e2d · 17 minutes ago History		
Name	Last commit message	Last commit date
..		
lab01	feat(main): make course structure	17 minutes ago
lab02	feat(main): make course structure	17 minutes ago
lab03	feat(main): make course structure	17 minutes ago
lab04	feat(main): make course structure	17 minutes ago
lab05	feat(main): make course structure	17 minutes ago
lab06	feat(main): make course structure	17 minutes ago
lab07	feat(main): make course structure	17 minutes ago
lab08	feat(main): make course structure	17 minutes ago
lab09	feat(main): make course structure	17 minutes ago
lab10	feat(main): make course structure	17 minutes ago
lab11	feat(main): make course structure	17 minutes ago

Рис. 20. Проверка иерархии создания рабочего пространства на GitHub

4.7. Задания для самостоятельной работы

1. Создание отчета по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).

С помощью команды `cd` перехожу в каталог `labs/lab02/report`. Далее с помощью утилиты `touch` создаю файл для отчета по выполнению второй лабораторной работы (рис. 21).

```
[sakudyakova@camuna59 arch-pc]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab02/report
[sakudyakova@camuna59 report]$ touch Л02_Кудякова_отчет
```

Рис. 21. Создание файла

2. Копирование отчетов по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.

С помощью команды `cd` перехожу в каталог `lab01/report` (рис. 22).

```
[sakudyakova@camuna59 report]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
```

Рис. 22. Переход в каталог

С помощью команды `ls` проверяю, есть ли в подкаталоге “Downloads” файл с отчетом по первой лабораторной работы (рис. 23).

```
[sakudyakova@camuna59 ~]$ ls Downloads
Л01_Кудякова_отчет.pdf
```

Рис. 23. Содержимое “Downloads”

Убеждаюсь в том, что файл есть.

Далее копирую отчет в каталог `lab01/report` с помощью `cd` и проверяю корректность выполненных действий с помощью `ls` (рис. 24)

```
[sakudyakova@camuna59 ~]$ cp ~/Downloads/Л01_Кудякова_отчет.pdf ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
[sakudyakova@camuna59 ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
[sakudyakova@camuna59 report]$ ls
bib image Makefile pandoc report.md Л01_Кудякова_отчет.pdf
[sakudyakova@camuna59 report]$
```

Рис. 24. Копирование файла в каталог

3. Загрузка файлов на GitHub.

С помощью команды `git add` добавляю файлы. Далее сохраняю изменения, используя команду `git commit -m` и добавляя пояснение “Add existing file”. После чего использую команду `git push`, чтобы отправить файлы в репозиторий (рис. 25)


```



[sakudyakova@camuna59 report]$ git add Л01_Кудякова_отчет.pdf
[sakudyakova@camuna59 report]$ git commit -m "Add existing file"
[master 3d53f10] Add existing file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Кудякова_отчет.pdf
[sakudyakova@camuna59 report]$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 931.73 KiB | 5.68 MiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:sakudyakova/study_2023-2024_arh-pc.git
9c68e2d..3d53f10 master -> master

```

Рис. 25. Загрузка файлов на GitHub

Захожу на GitHub, чтобы проверить действительно ли загрузились файлы (рис. 26).

study_2023-2024_arh-pc / labs / lab01 / report /  Add file ▾ ...

 sakudyakova Add existing file 3d53f10 · 7 minutes ago  History








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_Кудякова_отчет.pdf	Add existing file	7 minutes ago

Рис. 26. Добавление файла на GitHub

Такие же действия выполняю, чтобы добавить отчет по второй лабораторной работе на GitHub.

5 Выводы

В ходе данной лабораторной работы я изучила идеологию и применение средств контроля версий, а также научилась работать с системой git.

6 Список литературы

1. [Архитектура ЭВМ](#)