

# **Отчет по выполнению лабораторной работы №2**

**Дисциплина: Архитектура компьютера**

Ефремова Полина Александровна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>10</b>
4.1	1.Настройка GitHub . . . . .	10
4.2	2.Базовая настройка git. . . . .	10
4.3	3.Создание SSH ключа . . . . .	11
4.4	4.Создание рабочего пространства, репозитория курса на основе шаблона. . . . .	13
4.5	5.Создание репозитория курса на основе шаблона. . . . .	14
4.6	6.Настройка каталога курса. . . . .	16
<b>5</b>	<b>7. Выполнение заданий для самостоятельной работы.</b>	<b>17</b>
<b>6</b>	<b>Выводы</b>	<b>20</b>
<b>7</b>	<b>Список литературы</b>	<b>21</b>

# Список иллюстраций

3.1	Перечень основных команд git . . . . .	9
4.1	Учетная запись . . . . .	10
4.2	Ввод команд . . . . .	11
4.3	Настройка . . . . .	11
4.4	Имя начальной ветки . . . . .	11
4.5	Ввод команд . . . . .	11
4.6	Генерация ключей . . . . .	12
4.7	Установка xclip . . . . .	12
4.8	Использование xclip . . . . .	13
4.9	Вставка ключа . . . . .	13
4.10	Создание репозитория . . . . .	14
4.11	Создание репозитория . . . . .	14
4.12	Даю имя . . . . .	15
4.13	Переход в каталог курса . . . . .	15
4.14	Клонирование . . . . .	15
4.15	Переход в каталог курса . . . . .	16
4.16	Добавление файлов на сервер . . . . .	16
4.17	Проверка выполненной работы . . . . .	16
5.1	Создание отчета по лабораторной работе . . . . .	17
5.2	Проверка местонахождения лабораторных работ . . . . .	17
5.3	Копирование . . . . .	17
5.4	Использование git add . . . . .	18
5.5	Перенос файлов . . . . .	18
5.6	Ввод команд . . . . .	18
5.7	Переход в каталоги . . . . .	19
5.8	Переход в каталоги . . . . .	19

## **Список таблиц**

# 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 внёс. 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)



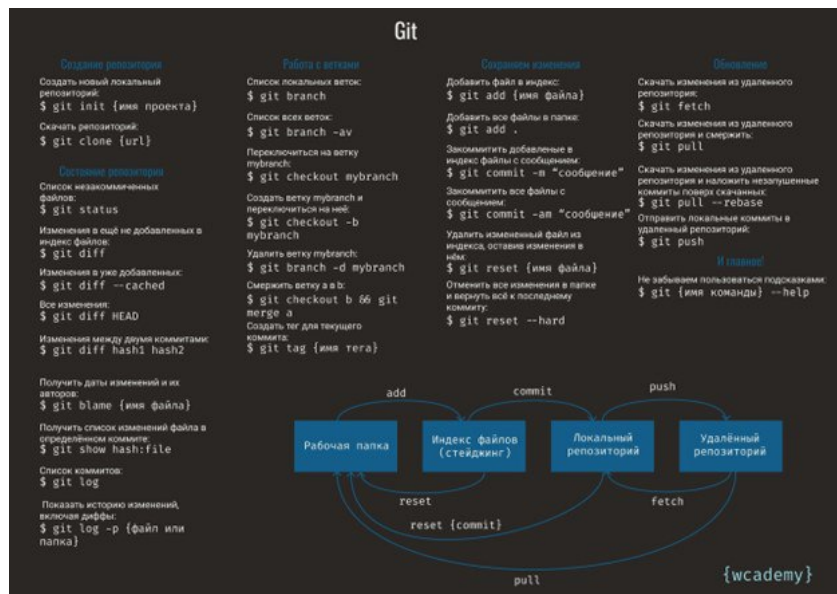


Рис. 3.1: Перечень основных команд git

## 4 Выполнение лабораторной работы

### 4.1 1.Настройка GitHub

Для выполнения лабораторной работы создаю учетную запись на <https://github.com/> (рис. 4.1)

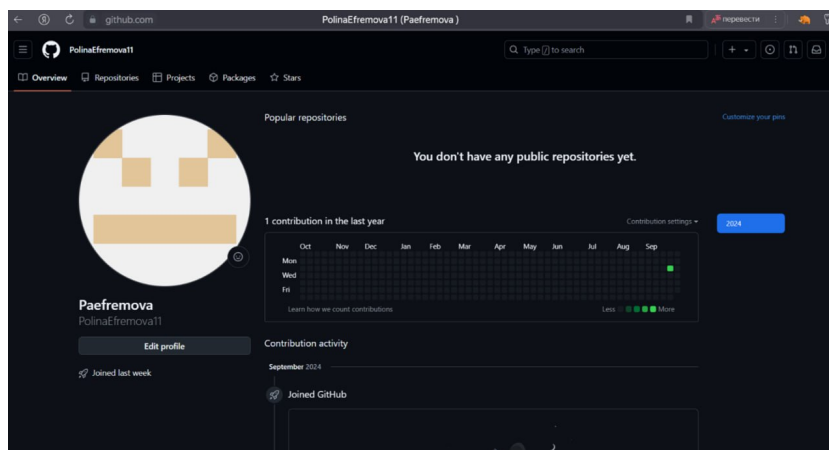
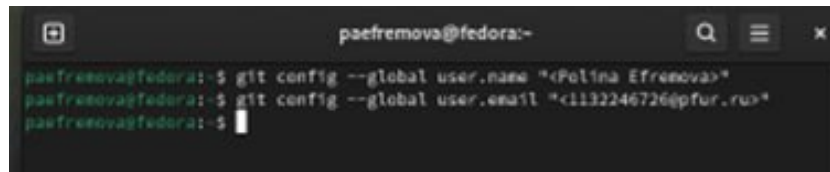


Рис. 4.1: Учетная запись

### 4.2 2.Базовая настройка git.

Делаю предварительную конфигурацию git. Захожу в терминал и ввожу команды, указывая свое имя и email (рис. 4.2)

A terminal window with a dark background. The title bar shows 'paeefremova@fedora:-'. The terminal contains three lines of text: 'paeefremova@fedora:~\$ git config --global user.name "Polina Efremova"', 'paeefremova@fedora:~\$ git config --global user.email "1132246726@pfur.ru"', and 'paeefremova@fedora:~\$' followed by a cursor.

```
paeefremova@fedora:~$ git config --global user.name "Polina Efremova"
paeefremova@fedora:~$ git config --global user.email "1132246726@pfur.ru"
paeefremova@fedora:~$
```

Рис. 4.2: Ввод команд

Настраиваю utf-8 в выходе сообщений git. (рис. 4.3)

A terminal window with a dark background. The terminal contains two lines of text: 'paeefremova@fedora:~\$ git config --global core.quotePath false' and 'paeefremova@fedora:~\$' followed by a cursor.

```
paeefremova@fedora:~$ git config --global core.quotePath false
paeefremova@fedora:~$
```

Рис. 4.3: Настройка

Задаю имя начальной ветки, которую буду называть master (рис. 4.4)

A terminal window with a dark background. The terminal contains one line of text: 'paeefremova@fedora:~\$ git config --global init.defaultBranch master'.

```
paeefremova@fedora:~$ git config --global init.defaultBranch master
```

Рис. 4.4: Имя начальной ветки

А также ввожу autocrlf и safecrlf (рис. 4.5)

A terminal window with a dark background. The terminal contains two lines of text: 'paeefremova@fedora:~\$ git config --global core.autocrlf input' and 'paeefremova@fedora:~\$ git config --global core.safecrlf warn'.

```
paeefremova@fedora:~$ git config --global core.autocrlf input
paeefremova@fedora:~$ git config --global core.safecrlf warn
```

Рис. 4.5: Ввод команд

## 4.3 3.Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория генерирую пару ключей (приватный и открытый). рис. 4.6)

```
paefremova@fedora:~$ ssh-keygen -C "Polina Efremova <1132246726@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/paefremova/.ssh/id_ed25519):
Created directory '/home/paefremova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/paefremova/.ssh/id_ed25519
Your public key has been saved in /home/paefremova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:6C7TTZivJmcRyisoXYvf3wA+a4lJr1Jm3KQMtNxnLEg Polina Efremova <1132246726@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
| .E. . |
| . . o |
| o o . |
| + . =. |
| = BooS |
| @o=.. |
| o * =+B. |
| o = Xo00ooo |
| | . =o%=o. . |
|-----[SHA256]-----+
paefremova@fedora:~$
```

Рис. 4.6: Генерация ключей

Чтобы скопировать из локальной консоли ключ в буфер обмена, устанавливаю команду xclip (рис. 4.7)

```
bash: xclip: команда не найдена...
Установить пакет «xclip», предоставляющий команду «xclip»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов....
Следующие пакеты должны быть установлены:
xclip-0.13-21.git11c6a61.fc40.x86_64 Command line clipboard grabber
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...
```

Рис. 4.7: Установка xclip

Теперь воспользуюсь командой xclip. (рис. 4.8)

```
paefremova@fedora:~$ ssh-keygen -C "Polina Efremova <1132246726@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/paefremova/.ssh/id_ed25519):
/home/paefremova/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/paefremova/.ssh/id_ed25519
Your public key has been saved in /home/paefremova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:bfgARGNAA4HIB4w38ajFx3Ij1MnFLXmw+ZotoLmYPE Polina Efremova <1132246726@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|+==0=..oo.      |
|O+==..o.o.+     |
|.+.+.+.+.o      |
|O.+ .+.o+       |
|. O $+O .        |
| O E ..         |
|. = +           |
|. . . + .        |
|. . . + .        |
+-----[SHA256]-----+
paefremova@fedora:~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip
paefremova@fedora:~$
```

Рис. 4.8: Использование xclip

Вставляю ключ в появившееся на сайте поле, указываю его имя. (рис. 4.9)

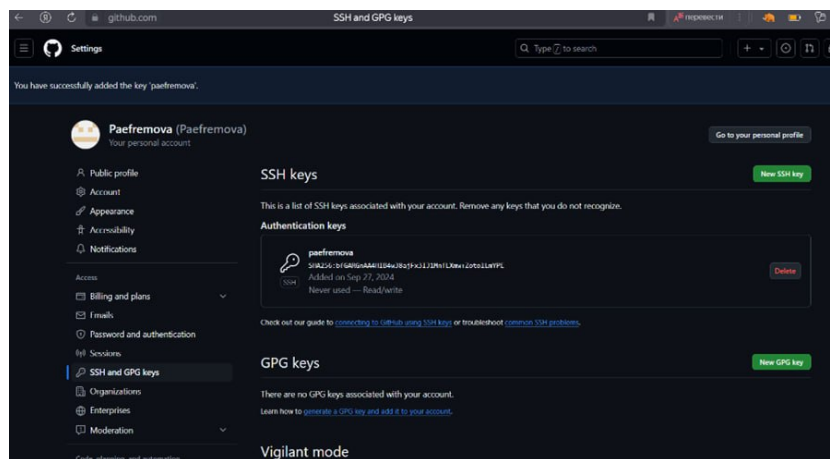


Рис. 4.9: Вставка ключа

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

Открываю терминал и создаю репозиторий для предмета «Архитектура компьютера». (рис. 4.10)

```
paefremova@fedora:~$ mkdir -p ~/work/study/2024-2025/"Архитектура компьютера"
paefremova@fedora:~$
```

Рис. 4.10: Создание репозитория

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

Захожу на страницу репозитория с шаблоном курса, выбираю его в качестве своего нового. (рис. 4.11)

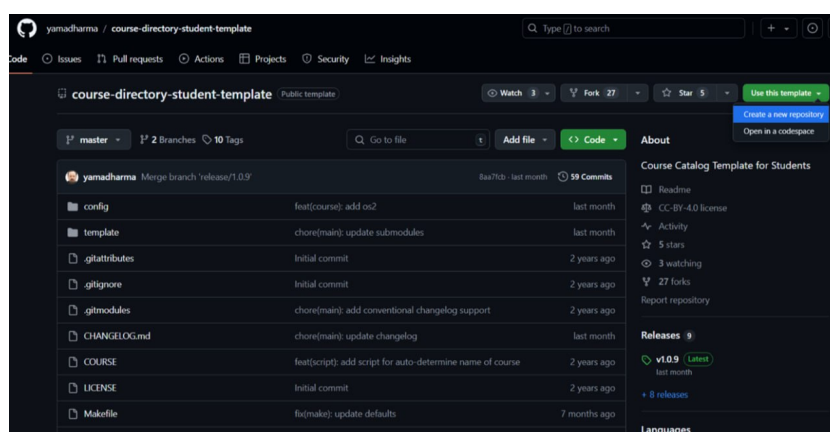


Рис. 4.11: Создание репозитория

Далее создаю его, задав ему имя. (рис. 4.12)

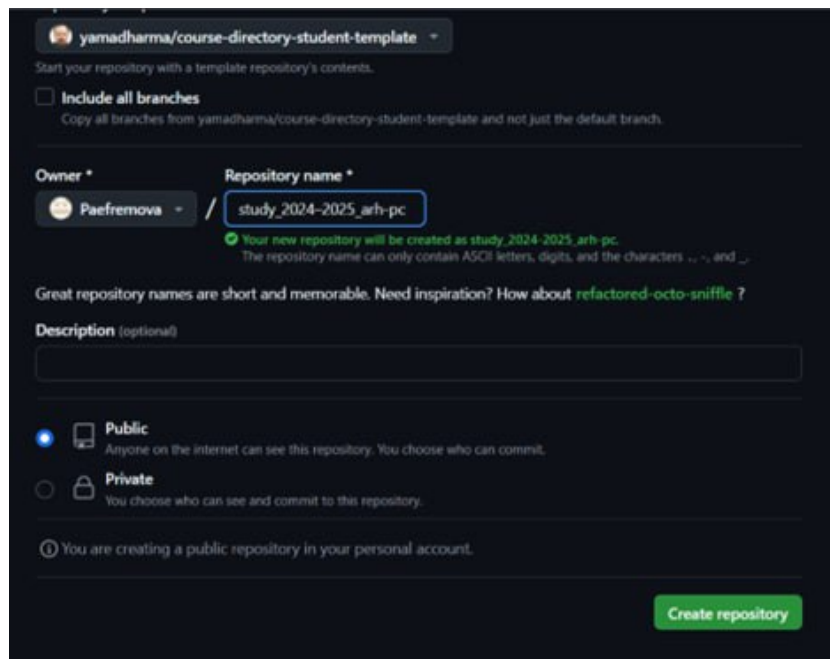


Рис. 4.12: Даю имя

Открываю терминал и перехожу в каталог курса. (рис. 4.13)



Рис. 4.13: Переход в каталог курса

Клонирую созданный репозиторий (рис. 4.14)

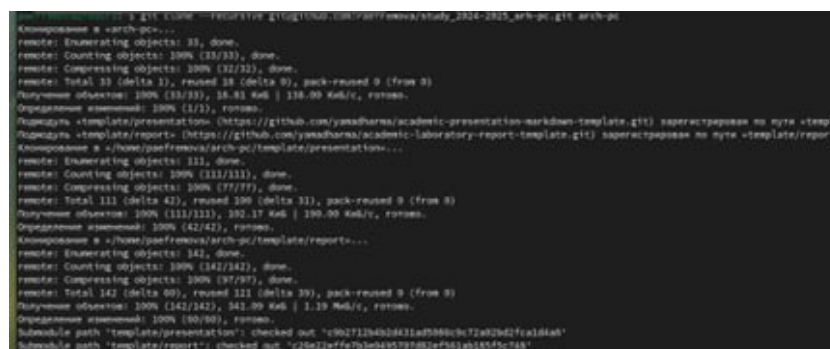


Рис. 4.14: Клонирование

# 4.6 6.Настройка каталога курса.

Перехожу в каталог курса и удаляю лишний файлы. (рис. 4.15)

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

Рис. 4.15: Переход в каталог курса

Создаю необходимые каталоги, отправляю файлы на сервер (рис. 4.16)

```
paefremova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
paefremova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
paefremova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
paefremova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -m "feat(main): make course structure"
[master 5314153] feat(main): make course structure
221 files changed, 53680 insertions(+)
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/_testlab01
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulxdov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/0th/ctte.bib
create mode 100644 labs/lab01/report/image/planning_800_800_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-8-5-2008-numer1.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_agnos.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/pandocnos/_..._text..._py
create mode 100644 labs/lab01/report/pandoc/filters/pandocnos/cert.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocnos/mark.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocnos/pandocattributes.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocnos/pandoc...
```

Рис. 4.16: Добавление файлов на сервер

В локальном репозитории проверяю результат выполненной работы (рис. 4.17)

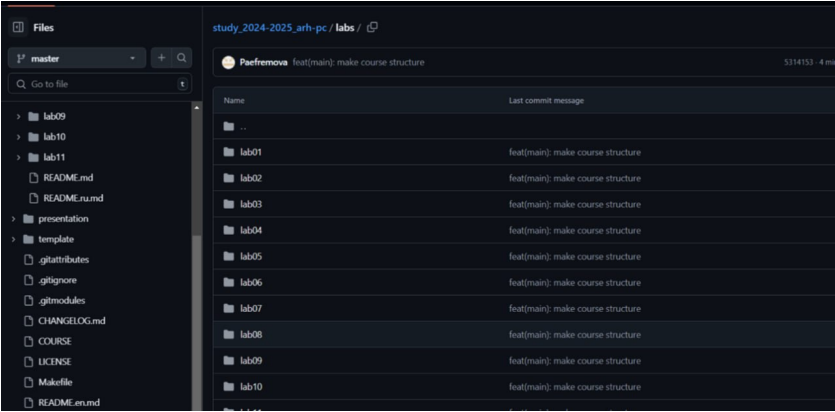


Рис. 4.17: Проверка выполненной работы



## 5 7. Выполнение заданий для самостоятельной работы.

Создаю отчет по выполнению второй лабораторной работы в соответствующем каталоге. С помощью команды `ls` проверяю, создан ли файл. (рис. 5.1)

```
paefremova@fedora: ~$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab02/report
paefremova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ touch Л02_Ефремова_П_отчет
paefremova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ ls
lib image Makefile pandoc report.md Л02_Ефремова_П_отчет
paefremova@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$
```

Рис. 5.1: Создание отчета по лабораторной работе

Для выполнения второго задания проверяю местонахождение своих лабораторных работ (рис. 5.2)

```
paefremova@fedora: ~$ ls Загрузки
Л01_Ефремова_П_отчет.docx Л01_Ефремова_П_отчет.pdf Л02_Ефремова_П_отчет.docx
paefremova@fedora: ~$
```

Рис. 5.2: Проверка местонахождения лабораторных работ

Копирую лабораторную работу с помощью утилиты `cp`, проверяю местонахождение файлов с помощью команды `ls` (рис. 5.3)

```
paefremova@fedora: ~$ ls /home/paefremova/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab01/report
lib image Makefile pandoc report.md Л01_Ефремова_П_отчет.pdf
paefremova@fedora: ~$ cp ~/Загрузки/Л01_Ефремова_П_отчет.docx /home/paefremova/work/study/2024-2025/"Архитектура
компьютера"/arch-pc/labs/lab01/report
paefremova@fedora: ~$ ls /home/paefremova/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab01/report
lib Makefile report.md Л01_Ефремова_П_отчет.pdf
image pandoc Л01_Ефремова_П_отчет.docx
paefremova@fedora: ~$
```

Рис. 5.3: Копирование

Для того чтобы загрузить эти файлы на GitHub, в первую очередь я использую команду `git add`. Так добавленные мной файлы станут отслеживаемыми. (рис. 5.4)

```

paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git add /01_Ефремова_
П_отчет.docx
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git add /01_Ефремова_
П_отчет.pdf
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ cd
paefremova@fedora:~$ cd /home/paefremova/work/study/2024-2025/"Архитектура компьютера"/arch-pc/labs/lab02/report

```

Рис. 5.4: Использование git add

Теперь осуществляю полноценный перенос файлов с помощью команды git commit -m “...” (рис. 5.5)

```

paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ git commit -m "Add la
b01/report"
[master f77491e] Add lab01/report
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/01_Ефремова_П_отчет.docx
create mode 100644 labs/lab01/report/01_Ефремова_П_отчет.pdf
create mode 100644 labs/lab02/report/02_Ефремова_П_отчет

```

Рис. 5.5: Перенос файлов

Использую команды: git status и git push, чтобы опубликовать свои локальные коммиты. (рис. 5.6)

```

paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git status
Текущая ветка: master
Ваша ветка опережает «origin/master» на 1 коммит.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)

ничего коммитить, нет изменений в рабочем каталоге
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ cd
paefremova@fedora:~$ cd /home/paefremova/work/study/2024-2025/"Архитектура компьютера"/arch-pc
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 15, готово.
Подсчет объектов: 100% (13/13), готово.
Сжатие объектов: 100% (9/9), готово.
Запись объектов: 100% (9/9), 4.31 МБ | 529.00 КБ/с, готово.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:Paefremova/study_2024-2025_arh-pc.git
5314153..f77491e master -> master
paefremova@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$

```

Рис. 5.6: Ввод команд

Перехожу в каталоги на GitHub, чтобы убедиться в том, что файлы находятся в нужных репозиториях. (рис. 5.7)

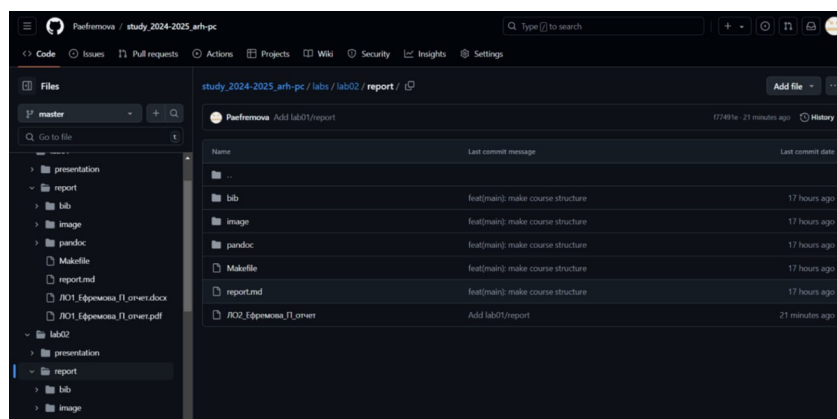


Рис. 5.7: Переход в каталоги

(рис. 5.8)

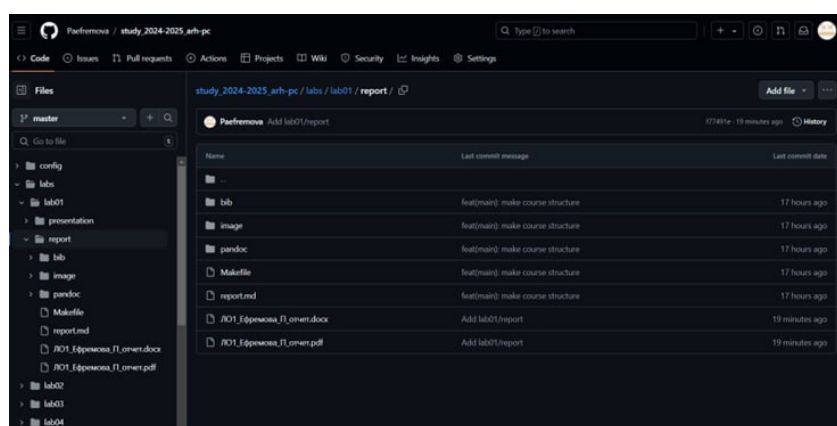


Рис. 5.8: Переход в каталоги

## 6 Выводы

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

## 7 Список 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/>