

Отчёт по лабораторной работе №2

Дисциплина: архитектура компьютера

Неустроева Ирина

Содержание

1	Цель работы	5
2	2 Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	10
4.0.1	Настройка GitHub	10
4.0.2	Создание SSH ключа.	12
4.0.3	Создание рабочего пространства и репозитория курса на основе шаблона	13
4.0.4	Создание репозитория курса на основе шаблона	14
4.0.5	Настройка каталога курса	16
4.0.6	Выполнение заданий для самостоятельной работы	17
5	Выводы	21

Список иллюстраций

Список таблиц

1 Цель работы

Получить практические навыки по работе с системой git, изучить идеологию и применение средств контроля версий.

2 2 Задание

1. Цель работы
2. Теоретическое введение
3. Выполнение лабораторной работы
4. Вывод

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. 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 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.

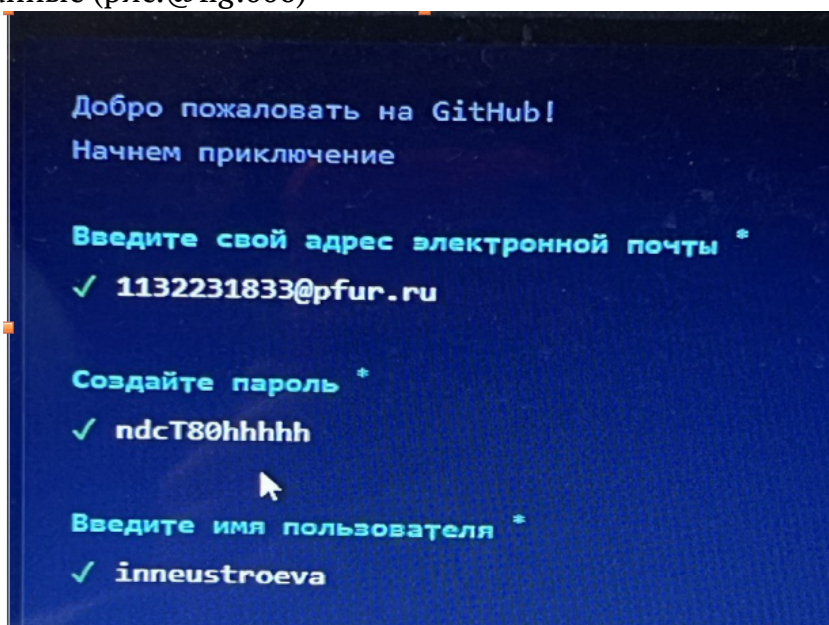
Команда	Описание
<code>git init</code>	Создание основного дерева репозитория
<code>git pull</code>	Получение обновлений (изменений) текущего дерева из центрального репозитория
<code>git push</code>	Отправка всех произведённых изменений локального дерева в центральный репозиторий
<code>git status</code>	Просмотр списка изменённых файлов в текущей директории
<code>git diff</code>	Просмотр текущих изменения
<code>git add .</code>	Добавить все изменённые и/или созданные файлы и/или каталоги
<code>git add файл</code>	Добавить конкретные изменённые и/или созданные файлы и/или каталоги

Команда	Описание
<code>git rm файл</code>	Удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)
<code>git commit -am</code>	Сохранить все добавленные изменения и все изменённые файлы
<code>git checkout -b</code>	Создание новой ветки, базирующейся на текущей
<code>git checkout</code>	Переключение на некоторую ветку
<code>git push origin</code>	Отправка изменений конкретной ветки в центральный репозиторий
<code>git merge --no-ff</code>	Слияние ветки с текущим деревом
<code>git branch -d</code>	Удаление локальной уже слитой с основным деревом ветки
<code>git branch -D</code>	Принудительное удаление локальной ветки
<code>git push origin</code>	Удаление ветки с центрального репозитория

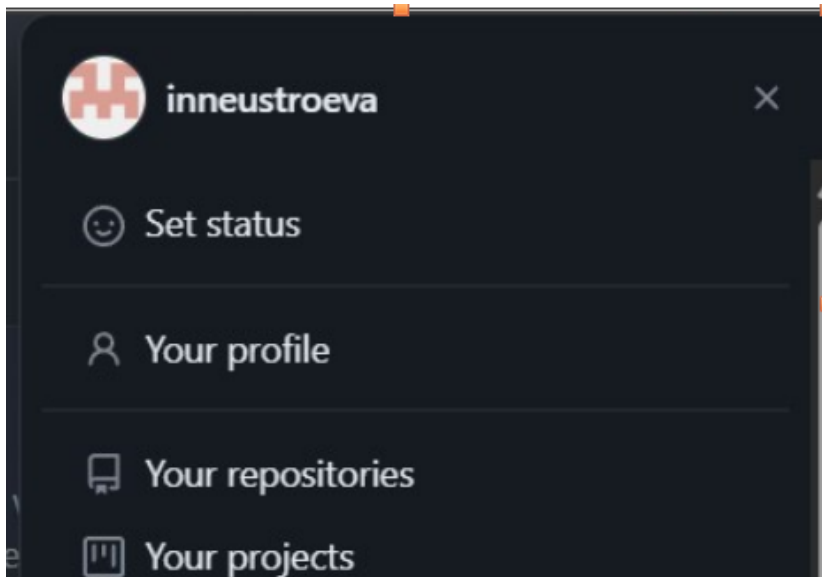
4 Выполнение лабораторной работы

4.0.1 Настройка GitHub

1 Переходим на сайт GitHub и создаю учетную запись, заполняя основные данные (рис.@fig:006)



3 Открываю терминал на виртуальной машине и задаю предварительную конфигурацию git, введя команду `git config --global user.name ""`, указывая свое имя и команду `git config --global user.email "work@mail"`, указывая свою электронную почту (рис.@fig:007)



4 Настроим utf-8 в выводе сообщений git следующей командой: `git config --global core.quotePath false` (рис.@fig:009)

```
inneustroeva@inneustroevaLinux:~$ git config --global core.quotePath false
inneustroeva@inneustroevaLinux:~$
```

5 Далее задаю имя “master” для начальной ветки, введя команду: `git config --global init.defaultBranch master` (рис.@fig:010)

```
inneustroeva@inneustroevaLinux:~$ git config --global init.defaultBranch master
inneustroeva@inneustroevaLinux:~$
```

6 Следующим шагом задаю параметр `autocrlf` со значением `input` следующей командой: `git config --global core.autocrlf input`, чтобы все переводы строк текстовых файлов в главном репозитории одинаковыми (конвертирую CRLF в LF только при коммитах. CR и LF – это символы, которые используют для обозначения разрыва строки в текстовых файлах (рис.@fig:011)

```
inneustroeva@inneustroevaLinux:~$ git config --global core.autocrlf input
inneustroeva@inneustroevaLinux:~$
```

7 Задаю параметр `safecrlf` со значением `warn`, таким образом Git будет проверять является ли окончаний строк обратимым для текущей настройки `core.autocrlf`, будет печатать предупреждение, но коммиты с необратимым переходом принимаются (рис.@fig:012)

```
inneustroeva@inneustroevaLinux:~$ git config --global core.safecrlf warn
inneustroeva@inneustroevaLinux:~$
```

4.0.2 Создание SSH ключа.

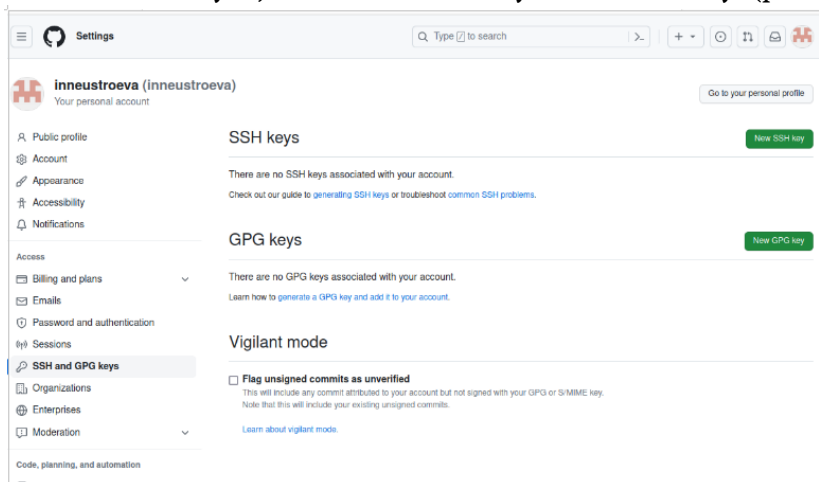
1 Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать приватный и открытый ключи, следующей командой: `ssh-keygen -C "Имя Фамилия, почта"`, указывая свои данные, ключ автоматически сохранится в каталоге `~/.ssh/` (рис.@fig:013)

```
lnneustroeva@lnneustroevaLinux:~$ ssh-keygen -C "Irina Neustroeva <1132231833@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/lnneustroeva/.ssh/id_rsa):
Created directory '/home/lnneustroeva/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/lnneustroeva/.ssh/id_rsa
Your public key has been saved in /home/lnneustroeva/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:HjNMg3w9lsNVTPVrULePFIA6X50b0njNBW4ivYdjymI Irina Neustroeva <1132231833@pfur.ru>
The key's randomart image is:
+---[RSA 3072]-----+
|
| .o====
| . .o.o .ooB
| o+.B =+*+
| +oo o+o@o*
| So o =.*o
| .+E o o
| ..
|
+---[SHA256]-----+
lnneustroeva@lnneustroevaLinux:~$
```

2 Копирую открытый ключ из директории с помощью утилиты `xclip`(рис.@fig:014)

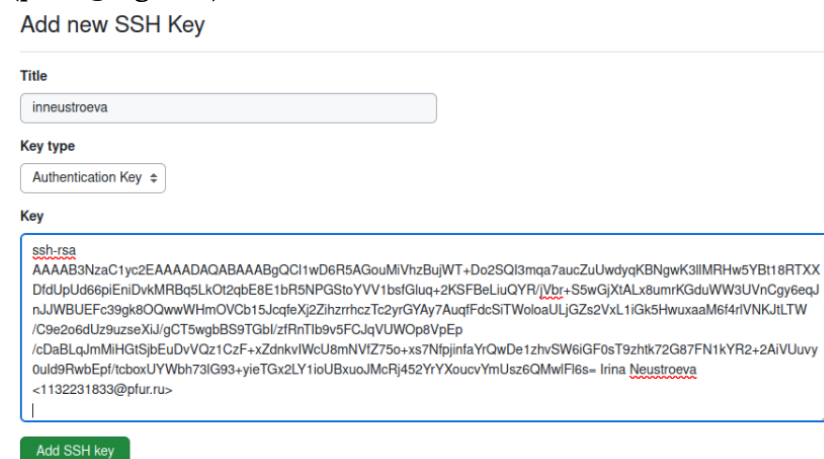
```
lnneustroeva@lnneustroevaLinux:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
lnneustroeva@lnneustroevaLinux:~$
```

3 Затем перехожу на сайт GitHub В своем профиле выбираю в боковом меню «SSH and GPG keys», нажимаю кнопку «New SSH key»(рис.@fig:015)

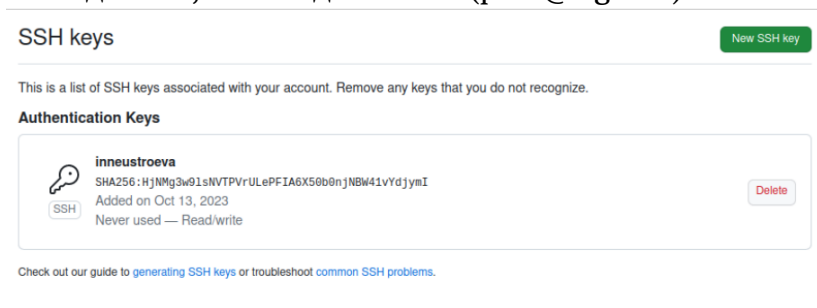


4 Далее в поле Title указываю свое имя для ключа. Вставляю скопированный

ключ в поле «Key». Нажимаю «Add SSH-key», чтобы завершить добавление ключа(рис.@fig:016)

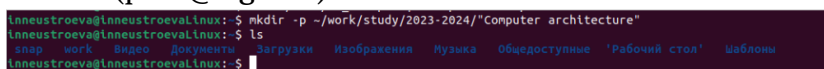


5 Убедились, что создали ключ(рис.@fig:017)



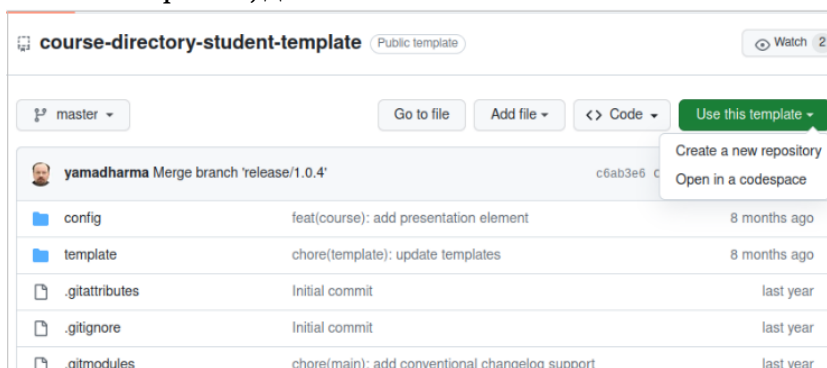
4.0.3 Создание рабочего пространства и репозитория курса на основе шаблона

1 открываю терминал, создаю рабочее пространство, с помощью команды `mkdir` и ключа `-p` создаю каталог для предмета Архитектура компьютера, по средством ввода следующей команды: `~/work/study/2023-2024/“Архитектура компьютера”` Далее проверяю с помощью команды `ls` на создание необходимого мне каталога.(рис.@fig:018)

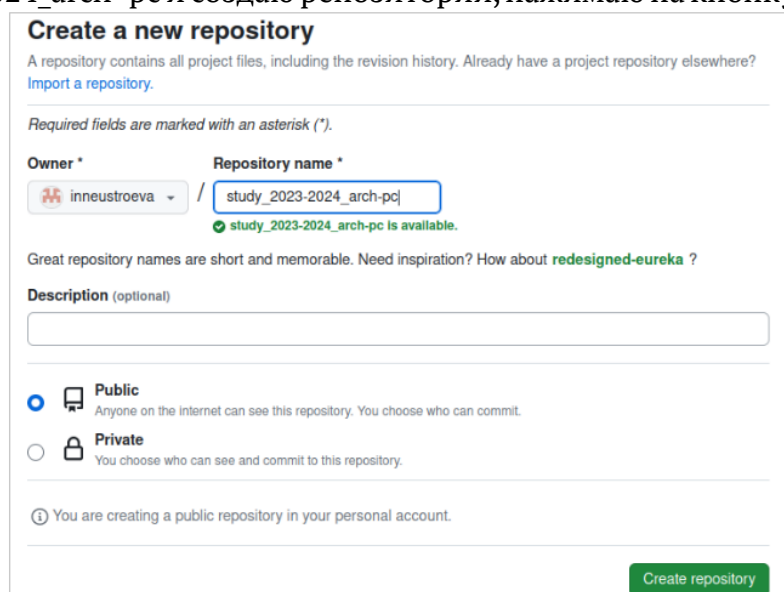


4.0.4 Создание репозитория курса на основе шаблона

1 Перешла на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharm/course-directory-student-template>. Далее выбираю «Use this template», для того чтобы использовать этот шаблон (рис.@fig:019)



2 В открывшемся окне задаю имя репозитория (Repository name): study_2023-2024_arch-рс и создаю репозиторий, нажимаю на кнопку «Create repository»(рис.@fig:020)



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 *

inneustroeva / study_2023-2024_arch-pc

study_2023-2024_arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about redesigned-eureka ?

Description (optional)

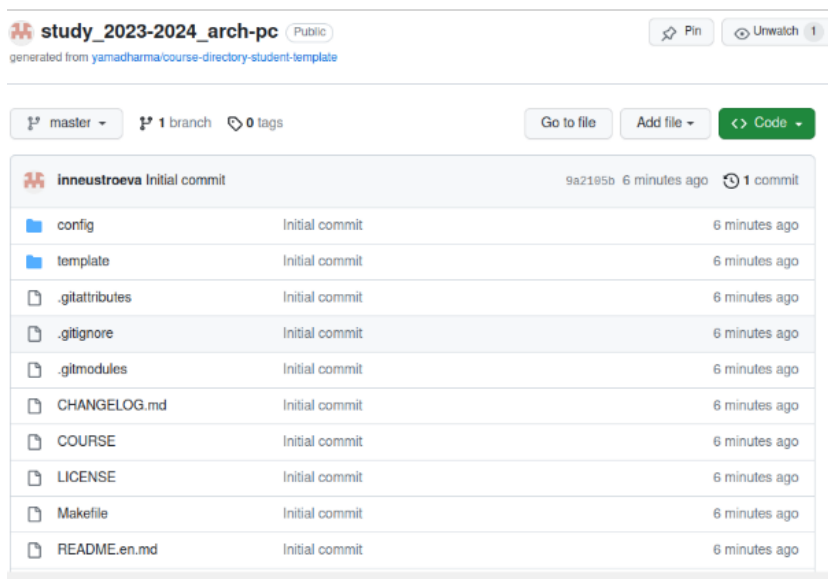
☒ Public
Anyone on the internet can see this repository. You choose who can commit.

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

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

Create repository

3 Проверяем, что репозиторий study_2023–2024_archрс успешно создан (рис.@fig:021)



4 Открыла терминал и с помощью команды `cd` перешла в созданный каталог курса (рис.@fig:022)

```
inneustroeva@inneustroevaLinux: ~$ cd ~/work/study/2023-2024/"Computer architecture"
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture$
```

5 Следующим шагом, клонирую созданный репозиторий с помощью команды `git clone --recursive git@github.com: inneustroeva/study_2023-2024_arh-pc.git arch-pc` (рис.@fig:023)

```
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture$ git clone --recursive git@github.com:inneustroeva/study_2023-2024_arch-pc.git
Клонирование в «study_2023-2024_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), 16.93 Киб | 2.82 Миб/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharna/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharna/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/inneustroeva/work/study/2023-2024/Computer architecture/study_2023-2024_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), 92.98 Киб | 951.00 Миб/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/inneustroeva/work/study/2023-2024/Computer architecture/study_2023-2024_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 | 2.14 Миб/с, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dca9c287a83917b82e3aef11a33b1e3b2'
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture$
```

6 Затем на странице созданного репозитория копирую ссылку, перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис.@fig:024)

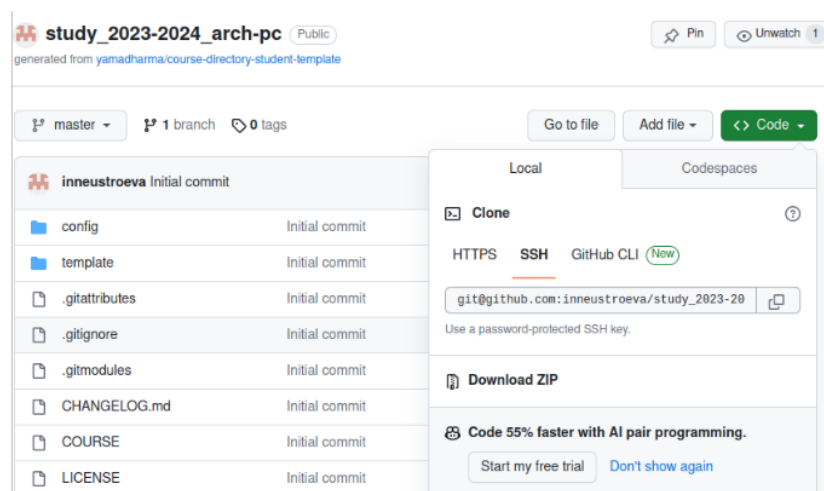


Рис.4.18 Окно с ссылкой для копирования.

4.0.5 Настройка каталога курса

1 Перехожу в каталог arch-pc с помощью команды cd (рис.@fig:025)

```
inneustroeva@inneustroeva:linux:~$ cd ~/work/study/2023-2024/Computer architecture/arch-pc
inneustroeva@inneustroeva:linux:~/work/study/2023-2024/Computer architecture/arch-pc$
```

2 Далее создаю необходимые каталоги (рис.@fig:026)

```
inneustroeva@inneustroeva:linux:~/work/study/2023-2024/Computer architecture/arch-pc$ echo arch-pc > COURSE
inneustroeva@inneustroeva:linux:~/work/study/2023-2024/Computer architecture/arch-pc$ make
inneustroeva@inneustroeva:linux:~/work/study/2023-2024/Computer architecture/arch-pc$
```

3 Отправляю созданные файлы с локального репозитория на сервер: добавляю все созданные каталоги с помощью git add, сохраняю изменения на сервере с помощью git commit (рис.@fig:027)

```
inneustroeva@inneustroeva:linux:~/work/study/2023-2024/Computer architecture/arch-pc$ git add .
inneustroeva@inneustroeva:linux:~/work/study/2023-2024/Computer architecture/arch-pc$ git commit -am 'feat(main): make course structure'
[main 5a97126] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
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/placimg_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/pandocxenos/_init_.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxenos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxenos/main.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxenos/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/placimg_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxenos/_init_.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxenos/core.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxenos/main.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxenos/pandocattributes.py
create mode 100644 labs/lab02/report/report.md
create mode 100644 labs/lab03/presentation/Makefile
```

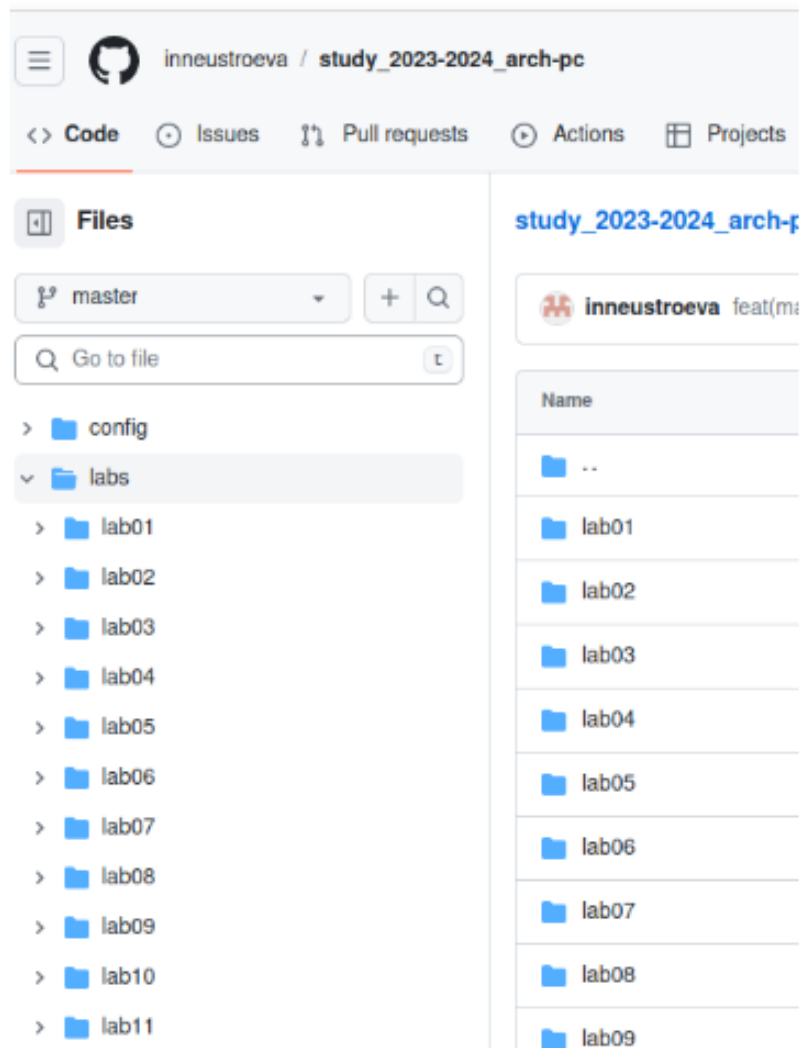
4 Отправляю все на сервер с помощью команды git push (рис.@fig:028)


```

inneustroeva@inneustroevaLinux:~/work/study/2023-2024/Computer architecture/arch-pc$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 Киб | 2.34 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:inneustroeva/study_2023-2024_arch-pc.git
 9a2105b..5a97126 master -> master
inneustroeva@inneustroevaLinux:~/work/study/2023-2024/Computer architecture/arch-pc$

```

5 Проверяю правильность создания иерархии рабочего пространства в локальном репозитории на сайте github (рис.@fig:029)



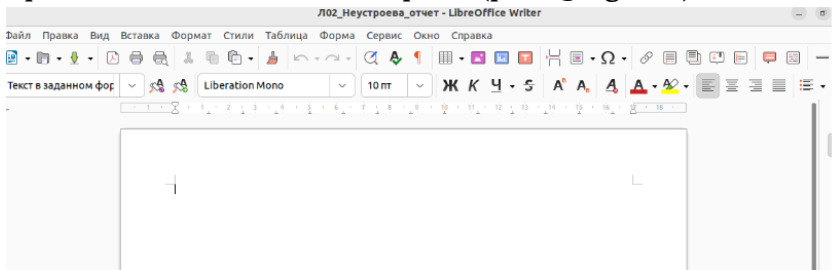
4.0.6 Выполнение заданий для самостоятельной работы

1 Перехожу в директорию labs/lab02/report с помощью команды cd. Создаю в каталоге файл для отчета по лабораторной работе с помощью утилиты touch

(рис.@fig:030)

```
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc$ cd ~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab02/report
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab02/report$ touch Л02_Неустроева_отчет
```

2 Открываю текстовый редактор LibreOffice Writer, через меню приложений. И открываю в нём созданный файл (рис.@fig:031)



3 Далее перехожу в каталог lab01/report (рис.@fig:032)

```
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab02/report$ cd ~/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$
```

4 Следующим шагом проверяю нахождение файла первой лабораторной работы, файл находится в Загрузках (рис.@fig:033)

```
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$ ls ~/Загрузки
Л01_Неустроева_отчет.pdf
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$
```

5 Копирую первую лабораторную работу, с помощью команды cp, и перемещаю её в нужный нам каталог (рис.@fig:034)

```
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$ cp ~/Загрузки/Л01_Неустроева_отчет.pdf /home/lnneustroeva/work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$
```

6 Проверяем правильность выполнения с помощью команды "ls" (рис.@fig:035)

```
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$ ls
bib image Makefile pandoc report.md Л01_Неустроева_отчет.pdf
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$
```

7 Добавляю файл Л01_Неустроева_отчет.pdf на сервер, с помощью команды git add и сохраняю изменения на сервере с помощью git commit (рис.@fig:036)

```
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$ git add Л01_Неустроева_отчет.pdf
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$ git commit -m "Add existing files"
[master daedc6a] Add existing files
1 file changed, 0 insertions(+), 0 deletions(-)
```

8 Отправляю в центральный репозиторий сохраненные изменения командой git push -f origin master (рис.@fig:037)

```
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$ git push -f origin master
Перечисление объектов: 10, готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 1.32 Миб | 1.85 Миб/с, готово.
Всего 6 (изменений 3), повторно использовано 6 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:lnneustroeva/study_2023-2024_arch-pc.git
5a97126..daedc6a master -> master
lnneustroeva@lnneustroevaLinux: /work/study/2023-2024/Computer_architecture/arch-pc/labs/lab01/report$
```

9 То же самое делаю для второй лабораторной работы: перехожу в директо-

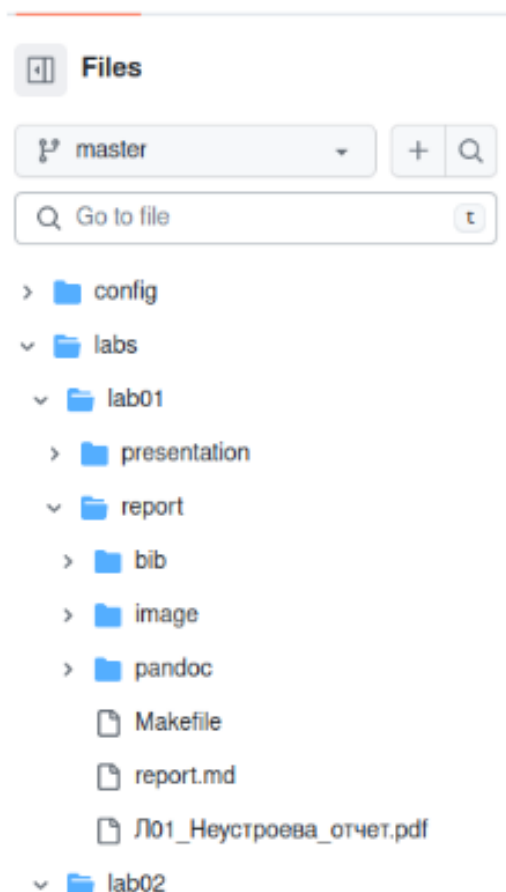
рию labs/lab02/report с помощью cd, добавляю с помощью git add нужный файл Л02_Неустроева_отчет, сохраняю изменения с помощью git commit (рис.@fig:038)

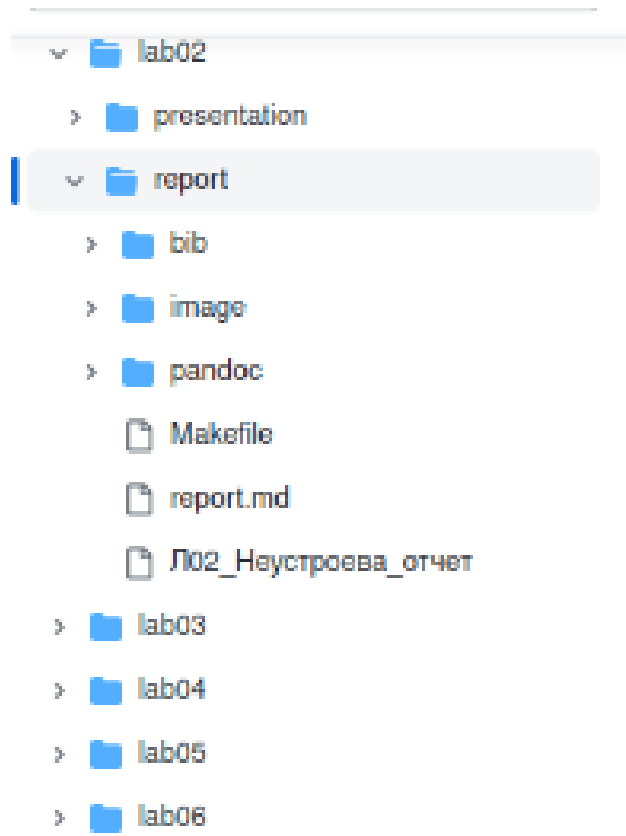
```
5897126..daedc6a master -> master
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01/report$ cd ..
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab01$ cd ..
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture/arch-pc/labs$ cd lab02/report
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report$ git add Л02_Неустроева_отчет
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report$ git commit -m "Add existing files"
[master 7385b91] Add existing files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab02/report/Л02_Неустроева_отчет
```

10 Отправляю в центральный репозиторий сохраненные изменения командой git push -f origin master (рис.@fig:039)

```
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report$ git push -f origin master
Перечисление объектов: 9, готово.
Порсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 497 байтов | 38.00 Кб/с, готово.
Всего 5 (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:inneustroeva/study_2023-2024_arch-pc.git
daedc6a..7385b91 master -> master
inneustroeva@inneustroevaLinux: ~/work/study/2023-2024/Computer architecture/arch-pc/labs/lab02/report$
```

11 Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что отчет по лабораторной работе номер 1 находится в соответствующем каталоге (рис.@fig:040) и отчет по лабораторной работе номер 2 тоже находится в соответствующем каталоге (рис.@fig:041)





5 Выводы

Приобрела практические навыки по работе с системой git. Изучила идеологию и применение средств контроля версий