

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: Лаптев Тимофей Сергеевич

Группа: НКАбд-01-24

МОСКВА

2024 г.

Цель работы

Изучить идеологию и применение средств контроля версий, а также приобрести практические навыки по работе с системой git.

1. Выполнение лабораторной работы

1). Настройка GitHub

1. Создаю учетную запись на сайте GitHub. Далее я заполнил основные данные учетной записи.

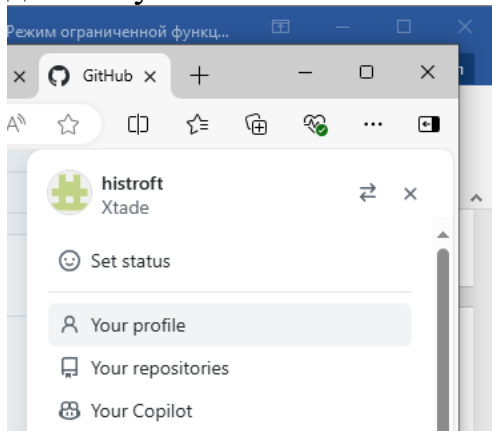


Рис.1: создан аккаунт Github

2). Базовая настройка Git

1. Отрыл терминал и сделал предварительную конфигурацию git. Ввел команду `git config --global user.name ""`, указал имя и команду `git config --global user.email "work@mail"`, указал в ней электронную почту.

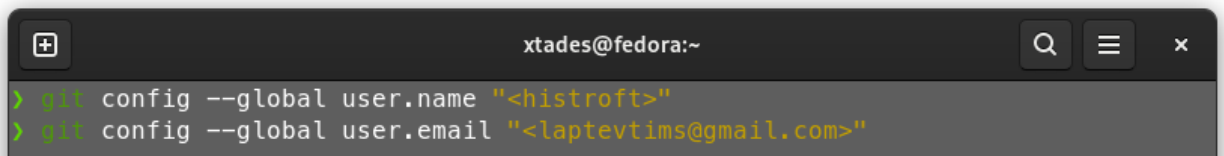


Рис.2

2. Настраиваю utf-8 в выводе сообщений git для корректного отображения СИМВОЛОВ

```
> git config --global core.quoterpath false
```

Рис.3

3. Задаю имя «master» для начальной ветки

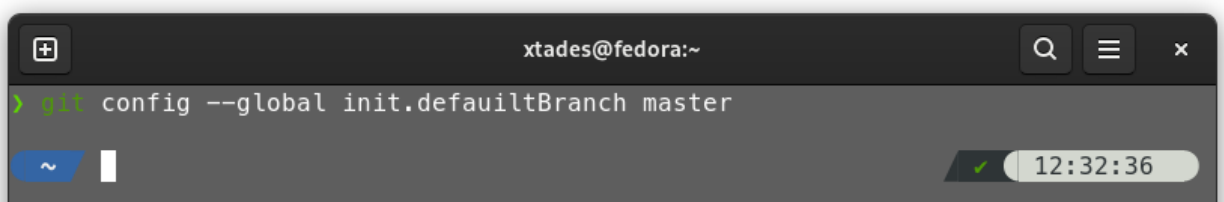
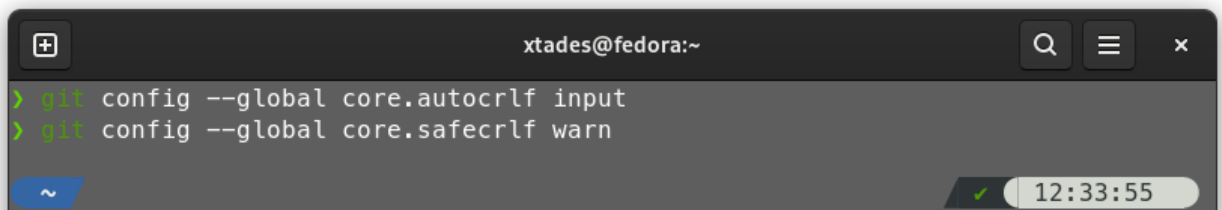


Рис.4

4. Задаю параметр `autocrlf` со значением `input`, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах. CR и LF –это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах. А также параметр `safecrlf` со значением `warn`, так Git будет проверять преобразование на обратимость. При значении `warn` Git только выведет предупреждение, но будет принимать необратимые конвертации.

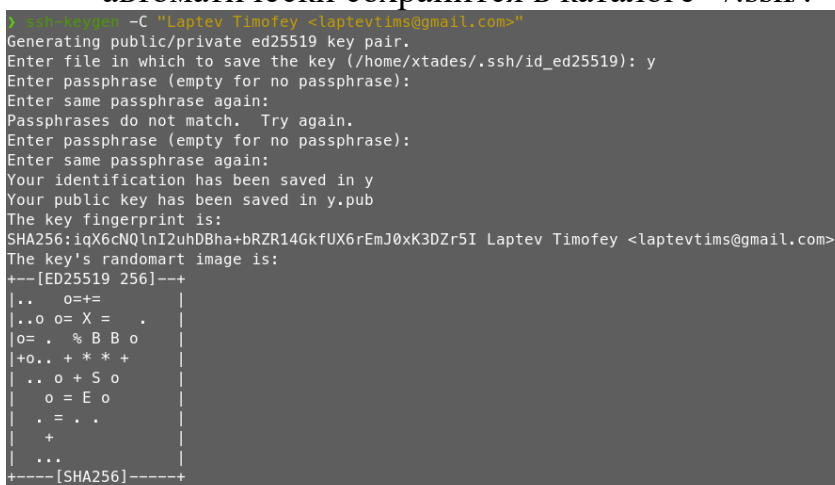


```
xtades@fedora:~  
> git config --global core.autocrlf input  
> git config --global core.safecrlf warn
```

Рис.5

3). Создание SSH-ключа

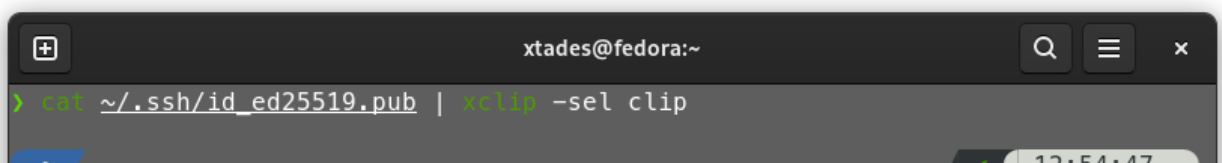
1. Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду `ssh-keygen -C "Имя Фамилия, work@email"`, указывая имя владельца и электронную почту владельца. Ключ автоматически сохранится в каталоге `~/.ssh/`.



```
$ ssh-keygen -C "Laptev Timofey <laptevtims@gmail.com>"  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/xtades/.ssh/id_ed25519): y  
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 y  
Your public key has been saved in y.pub  
The key fingerprint is:  
SHA256:iqX6cN0lnI2uhDBha+bRZR14GkfUX6rEmJ0xK3DZr5I Laptev Timofey <laptevtims@gmail.com>  
The key's randomart image is:  
+--[ED25519 256]--+  
|.. o+=+ |  
|..o o= X = |  
|o= . % B B o |  
|+O.. + * * + |  
|.. o + S o |  
| o = E o |  
| . = . . |  
| + |  
| ... |  
+-----[SHA256]-----+
```

Рис.6

2. Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты `xclip`



```
xtades@fedora:~  
> cat ~/.ssh/id_ed25519.pub | xclip -sel clip
```

Рис.7

3. Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key»

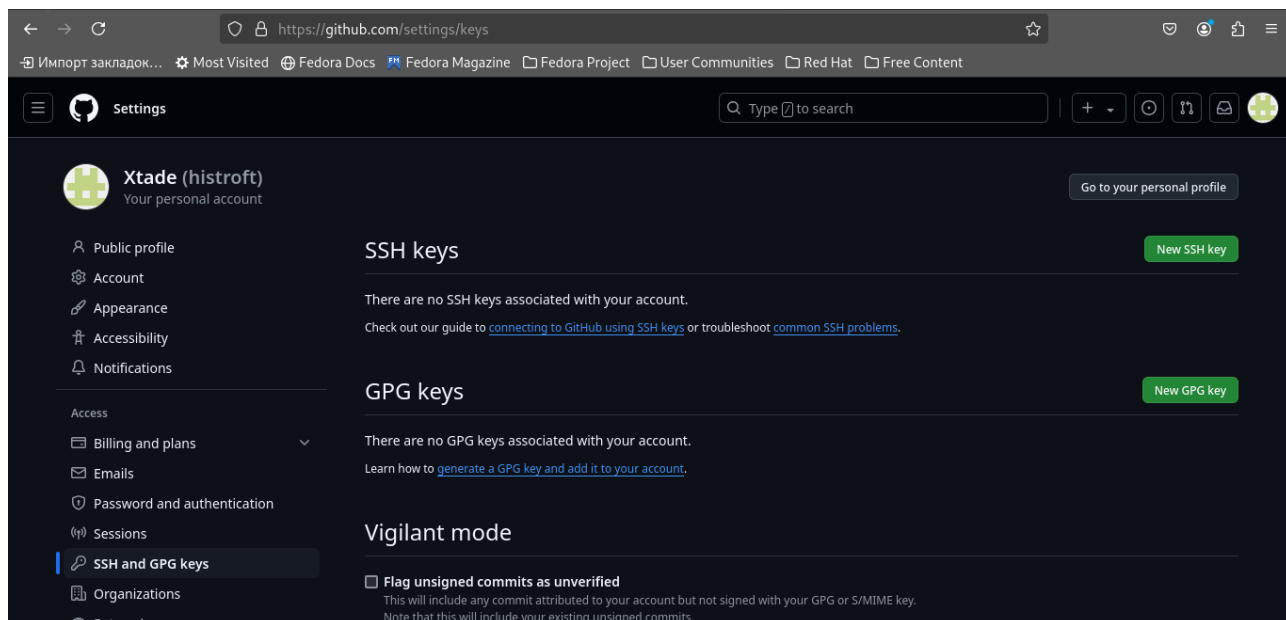


Рис.8

4. Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа

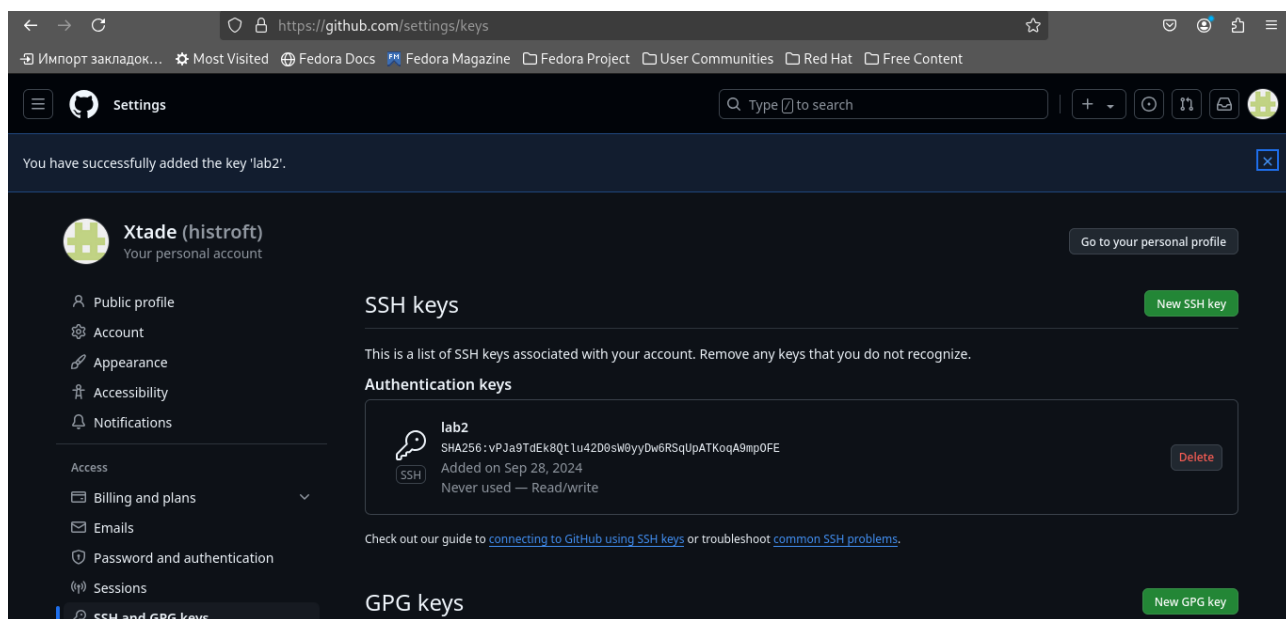


Рис.9

4). Создание рабочего пространства и репозитория курса на основе шаблона

1. Создаю директорию, рабочее пространство, с помощью утилиты `mkdir`, благодаря ключу `-p` создаю все директории после домашней `~/work/study/2022-2023/“Архитектура компьютера”` рекурсивно. Далее проверяю с помощью `ls`, действительно ли были созданы необходимые мне каталоги

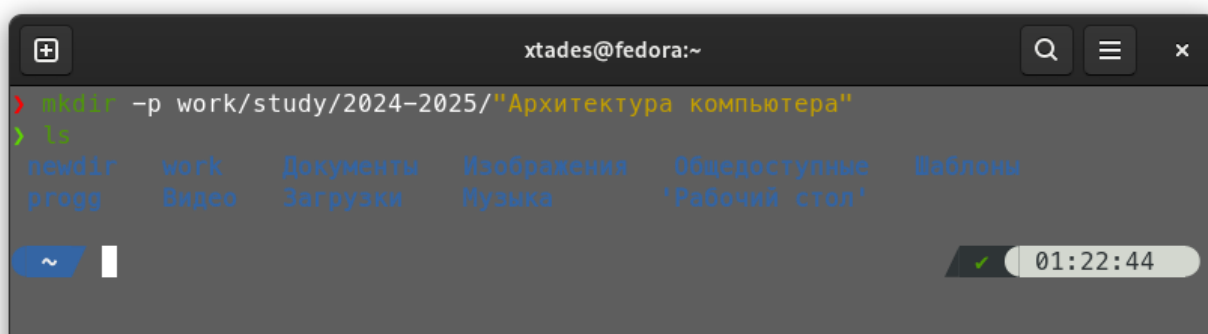


Рис.10

3). Создание репозитория курса на основе шаблона

1. В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория

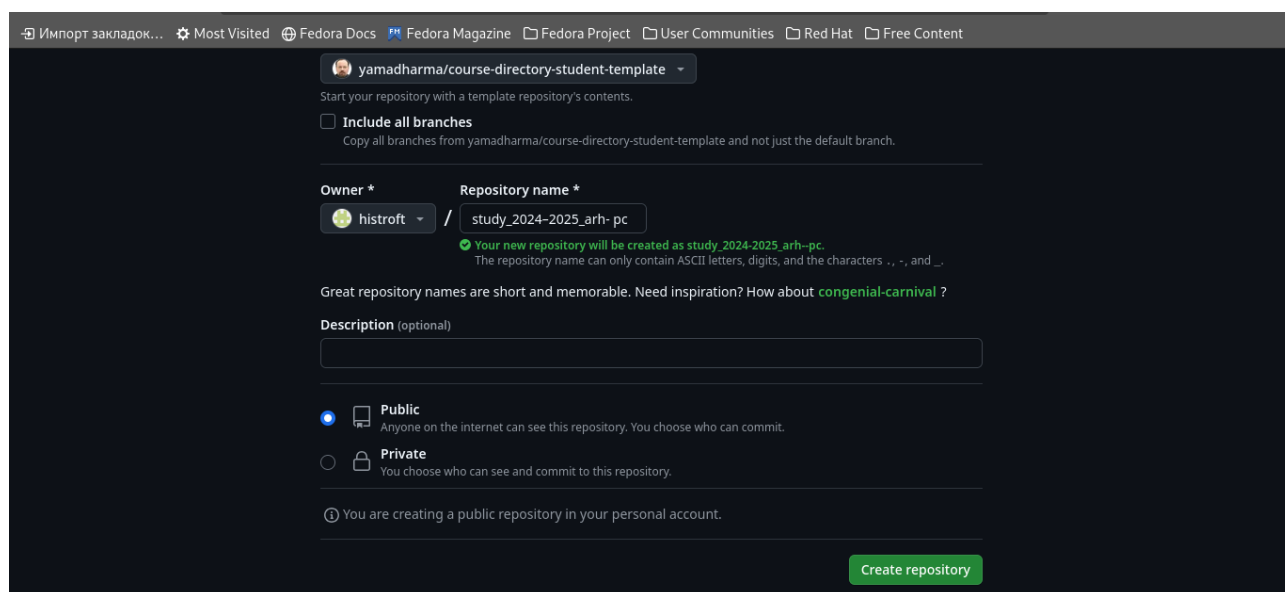
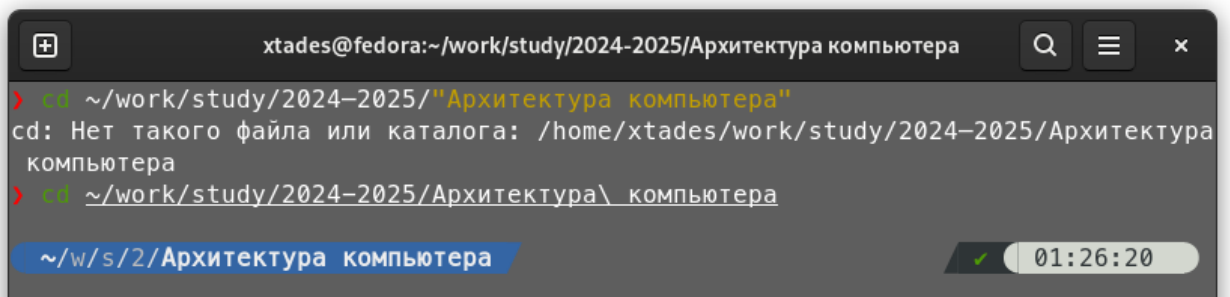


Рис.11

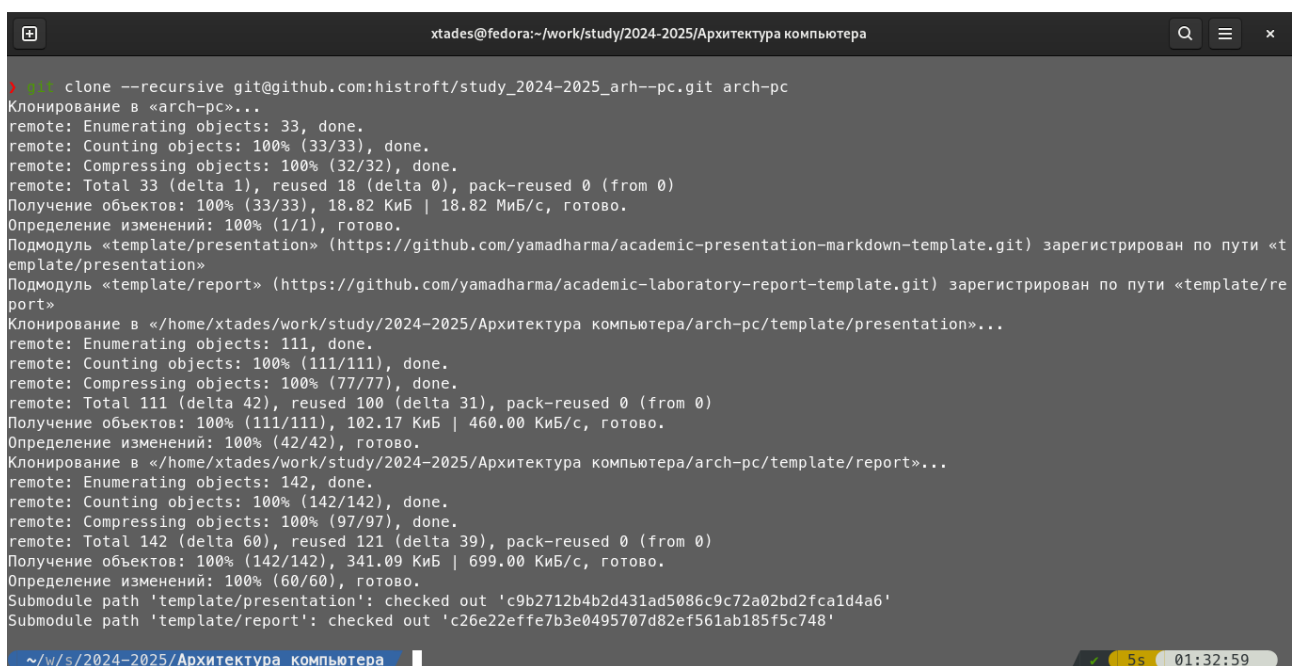
2. Через терминал перехожу в созданный каталог курса с помощью утилиты `cd`



```
xtades@fedora:~/work/study/2024-2025/Архитектура компьютера
> cd ~/work/study/2024-2025/"Архитектура компьютера"
cd: Нет такого файла или каталога: /home/xtades/work/study/2024-2025/Архитектура
компьютера
> cd ~/work/study/2024-2025/Архитектура\ компьютера
~/w/s/2/Архитектура компьютера
```

Рис.12

3. Клонировать созданный репозиторий с помощью команды `git clone` – `recursive` и ссылка, которую я копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH»

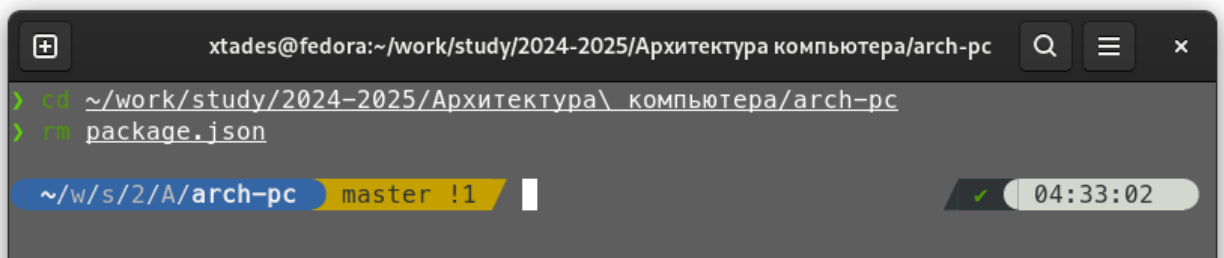


```
xtades@fedora:~/work/study/2024-2025/Архитектура компьютера
> git clone --recursive git@github.com:histroft/study_2024-2025_arh--pc.git arch-pc
Клонирование в «arch-pc»...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 33 (delta 1), reused 18 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (33/33), 18.82 КиБ | 18.82 МБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «t
emplate/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/re
port»
Клонирование в «/home/xtades/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)
Получение объектов: 100% (111/111), 102.17 КиБ | 460.00 КиБ/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/home/xtades/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)
Получение объектов: 100% (142/142), 341.09 КиБ | 699.00 КиБ/с, готово.
Определение изменений: 100% (60/60), готово.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
~/w/s/2024-2025/Архитектура компьютера
```

Рис.13

5). Настройка каталога курса

1. Удаляю лишние файлы с помощью утилиты `rm`

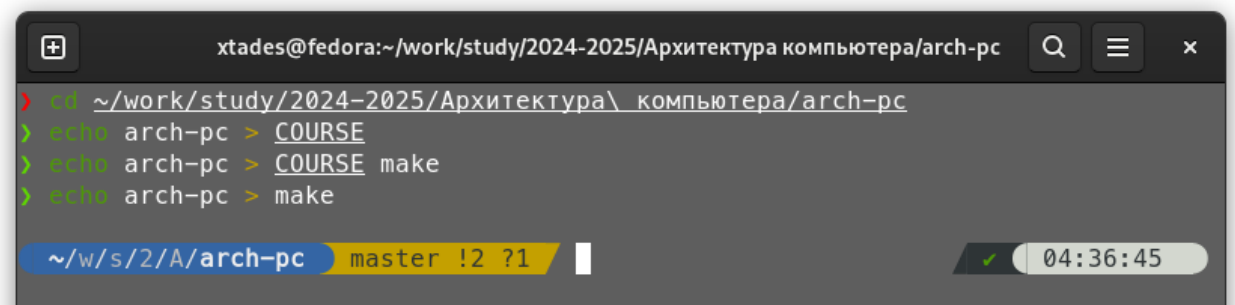


```
xtades@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc
> cd ~/work/study/2024-2025/Архитектура\ компьютера/arch-pc
> rm package.json

~/w/s/2/A/arch-pc master !1 04:33:02
```

Рис. 14

2. Создаю необходимые каталоги



```
xtades@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc
> cd ~/work/study/2024-2025/Архитектура\ компьютера/arch-pc
> echo arch-pc > COURSE
> echo arch-pc > COURSE make
> echo arch-pc > make

~/w/s/2/A/arch-pc master !2 ?1 04:36:45
```

Рис. 15

3. Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью git add, комментирую и сохраняю изменения на сервере как добавление курса с помощью git commit



```
> git commit -am 'feat(main): make course structure'
[master 4c4a025] feat(main): make course structure
3 files changed, 2 insertions(+), 14 deletions(-)
create mode 100644 make
delete mode 100644 package.json
> git push
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (4/4), 332 байта | 332.00 КиБ/с, готово.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:histroft/study_2024-2025_arh--pc.git
94b0a86..4c4a025 master -> master

~/w/s/2/A/arch-pc master 04:39:52
```

Рис. 16

4. Понял, что совершил ошибку в пункте 2 и у меня не создались директории, далее боролся с проблемой, установил пакет “make”, но директории все еще не создавались попытался найти решение в интернете, но ничего не нашел


```
xtades@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc
> git commit -am 'feat(main): make course structure'
[master 16c1be8] feat(main): make course structure
1 file changed, 1 insertion(+), 1 deletion(-)
> cd arch-pc > COURSE
> make
zsh: make: команда не найдена...
Установить пакет «make», предоставляющий команду «make»? [N/y] y

* Ожидание в очереди...
* Загрузка списка пакетов....
Следующие пакеты должны быть установлены:
guile30-3.0.7-12.fc40.x86_64  A GNU implementation of Scheme for application extensibility
make-1:4.4.1-6.fc40.x86_64  A GNU tool which simplifies the build process for users
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
* Ожидание в очереди...
* Загрузка пакетов...
* Запрос данных...
* Проверка изменений...
* Установка пакетов...
Usage:
  make <target>

Targets:
  list              List of courses
  prepare           Generate directories structure
  submodule         Update submules
```

Рис. 17

5. Внезапно понял, что функция make работает с файлом “makefile” и после этого повторил шаг 2, но уже находясь в нужной директории при помощи команды cd, в директории arch-pc произошли изменения, теперь, повторяя шаг 3 получен результат

```
xtades@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc
> git add .
> git commit -am 'feat(main): make course structure'
[master bb76ed3] feat(main): make course structure
223 files changed, 53687 insertions(+), 6 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
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/.projectile
create mode 100644 labs/lab02/presentation/.texlabroot
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
```

Рис. 18

```
xtades@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc
create mode 100644 presentation/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 presentation/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_fignos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_secnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/core.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/main.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/pandocattribut
es.py
create mode 100644 presentation/report/report.md

> git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (32/32), готово.
Запись объектов: 100% (38/38), 341.77 КиБ | 2.51 МиБ/с, готово.
Total 38 (delta 6), reused 1 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To github.com:histroft/study_2024-2025_arh--pc.git
4c4a025..bb76ed3 master -> master

~/w/s/2/A/arch-pc master 05:48:35
```

Рис. 19

6. Осталось лишь проверить правильность выполнения работы на сайте GitHub

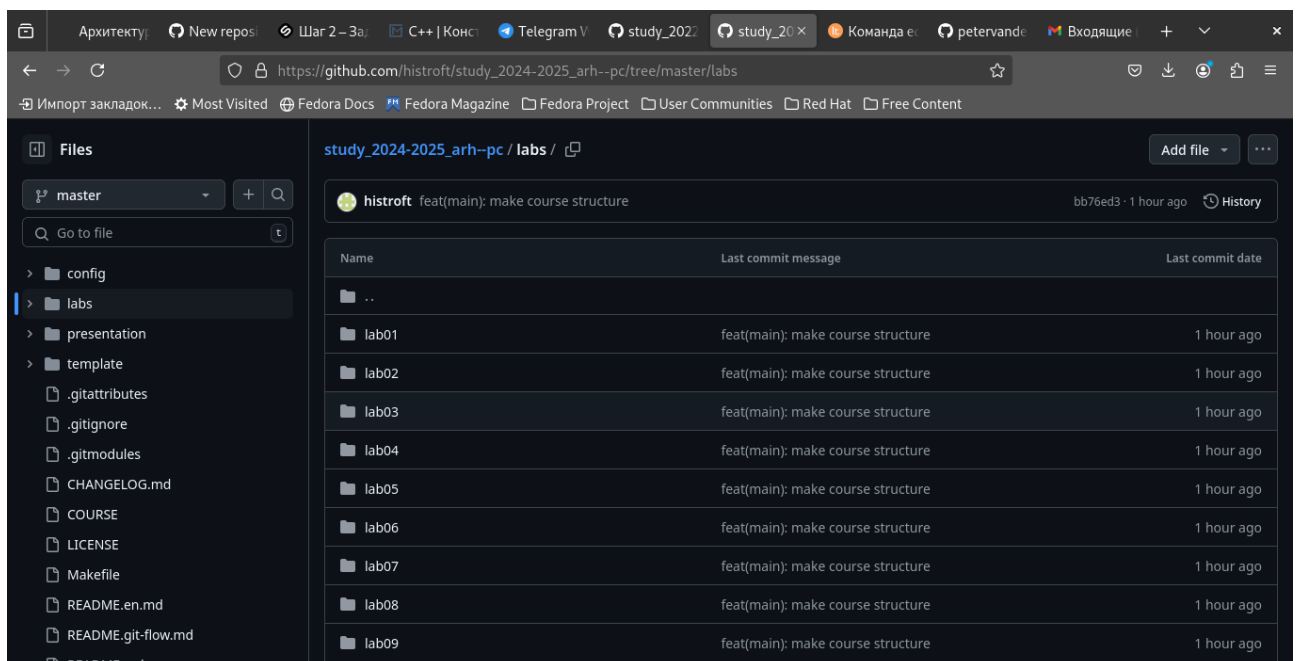
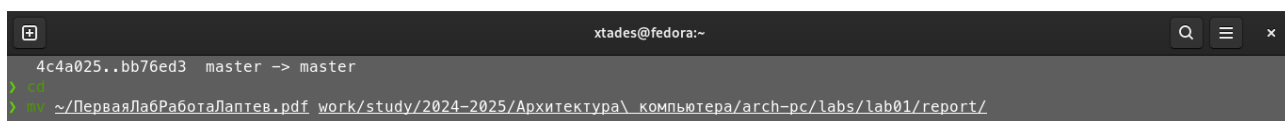


Рис. 20

6). Задание для самостоятельной работы.

1. Переносу первую лабораторную работу в нужную директорию



```
4c4a025..bb76ed3 master -> master
> git checkout ~/ПерваяЛабРаботаЛаптев.pdf work/study/2024-2025/Архитектура\_компьютера/arch-pc/labs/lab01/report/
```

Рис. 21

(повторил то же действие и со второй лабораторной работой, а также перенес их на платформу GitHub по аналогии с 3 и 5 пунктами настройки каталога курса)

7). Вывод

При выполнении данной лабораторной работы я приобрел практические навыки работы с Git, GitHub, а также изучил идеологию и применение средств контроля версий, научился решать непредвиденные проблемы при работе с ресурсами.

8). Ответы на вопросы

- Системы контроля версий (VCS) — это инструменты, позволяющие отслеживать изменения в файлах и управлять их версиями. Они предназначены для решения задач, связанных с совместной работой над проектами, восстановлением предыдущих версий и упрощением управления изменениями.
- Хранилище: место, где сохраняются все версии проекта.
Commit: действие, которое фиксирует изменения в хранилище, создавая новую версию.
История: последовательность всех коммитов, отражающая изменения проекта.
Рабочая копия: локальная версия проекта, с которой работает разработчик.
- Централизованные VCS имеют центральное хранилище, к которому подключаются пользователи (например, Subversion). Децентрализованные VCS позволяют каждому пользователю иметь полную копию репозитория (например, Git).
- При единоличной работе с хранилищем пользователь выполняет коммиты, создает ветки и управляет историей изменений, работая только с локальной копией.
- При работе с общим хранилищем пользователи синхронизируют свои изменения с центральным репозиторием, используя такие команды, как pull и push для получения или отправки изменений.
- Основные задачи Git: отслеживание изменений, управление версиями, слияние изменений из разных веток, восстановление предыдущих состояний.
- git init создание основного дерева репозитория git pull получение обновлений (изменений) текущего дерева из центрального репозитория git push отправка всех произведённых изменений локального дерева в центральный репозиторий git status просмотр списка изменённых файлов в текущей директории git diff просмотр текущих изменений git add . добавить все изменённые и/или созданные файлы и/или каталоги git add имена_файлов добавить конкретные изменённые и/или созданные файлы и/или каталоги git rm имена_файлов удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории) git commit -am 'Описание коммита' сохранить все добавленные изменения и все изменённые файлы git checkout -b имя_ветки создание новой ветки, базирующейся на текущей git checkout имя_ветки переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) git push origin имя_ветки отправка изменений конкретной ветки в центральный репозиторий git merge --no-ff имя_ветки слияние ветки с текущим деревом

`git branch -d имя_ветки` удаление локальной уже слитой с основным деревом ветки `git branch -D имя_ветки` принудительное удаление локальной ветки `git push origin :имя_ветки` удаление ветки с центрального репозитория

- Локальный репозиторий: 1. Создание нового репозитория: `git init` 2. Добавление файлов в индекс: `git add .` 3. Сохранение изменений: `git commit -m "Первый коммит"`
Удалённый репозиторий: 1. Клонирование удалённого репозитория: `git clone` и ссылка на репозиторий на ресурсе GitHub 2. Отправка изменений в удалённый репозиторий: `git push origin main` 3. Получение изменений из удалённого репозитория: `git pull origin main`