

Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютера

Панина Жанна Валерьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	11
4.1	Настройка github	11
4.2	Базовая настройка git	11
4.3	Создание SSH ключа	12
4.4	Создание рабочего пространства и репозитория курса на основе шаблона	15
4.5	Создание репозитория курса на основе шаблона	15
4.6	Настройка каталога курса	17
4.7	Выполнение заданий для самостоятельной работы	19
5	Выводы	20
	Список литературы	21

Список иллюстраций

4.1	Создание учётной записи в Github	11
4.2	Предварительная конфигурация git	12
4.3	Настройка соответствующих параметров	12
4.4	Создание SSH ключа	12
4.5	Меню SSH and GPG keys	13
4.6	Предложение установки xclip	13
4.7	Установка xclip	13
4.8	Ввод команды от имени админа	14
4.9	Ключ создан	14
4.10	Создание каталога	15
4.11	Страница репозитория с шаблоном курса	15
4.12	Создание репозитория	16
4.13	Клонирование созданного репозитория	16
4.14	Удаление лишних файлов	17
4.15	Создание каталогов	17
4.16	Отправка файлов на сервер	17
4.17	Отправка файлов на сервер	18
4.18	Иерархия рабочего пространства создана	18
4.19	Отчет в markdown	19

Список таблиц

1 Цель работы

Целью данной работы является изучить идеологию и применение средств контроля версий, а также 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

3.1. Системы контроля версий. Общие понятия 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 (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.

3.2. Система контроля версий Git

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.

3.3. Основные команды git

Наиболее часто используемые команды git представлены в таблице 2.1.

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 изменений
<code>git add .</code>	добавить все изменённые и/или созданные файлы и/или каталоги
<code>git add имена_файлов</code>	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

3.4. Стандартные процедуры работы при наличии центрального репозитория

Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений): `git checkout master` `git pull` `git checkout -b имя_ветки` Затем можно вносить изменения в локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории. Для этого необходимо проверить, какие файлы изменились к текущему моменту: `git status` и при необходимости удаляем лишние файлы, которые не хотим отправлять в центральный репозиторий. Затем полезно просмотреть текст изменений на предмет соответствия правилам ведения чистых коммитов: `git diff` Если какие-либо файлы не должны попасть в коммит, то помечаем только те файлы, изменения которых нужно сохранить. Для этого используем команды добавления и/или удаления с нужными опциями: `git add имена_файлов` `git rm имена_файлов` Если нужно сохранить все изменения в текущем каталоге, то используем: `git add .` Затем сохраняем изменения, поясняя, что было сделано: `git commit -am "Some commit message"` и отправляем в центральный репозиторий:

`git push origin имя_ветки` или `git push`

4 Выполнение лабораторной работы

4.1 Настройка github

Я создала учётную запись на сайте <https://github.com/> и заполнила основные данные. (рис. 4.1)

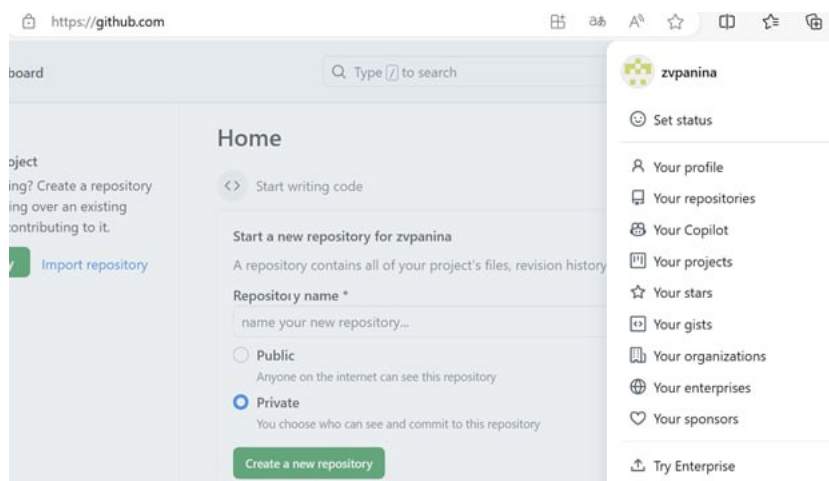


Рис. 4.1: Создание учётной записи в Github

4.2 Базовая настройка git

Сначала сделаем предварительную конфигурацию git. Открываю терминал и ввожу следующие команды, указав своё имя и email (рис. 4.2)

```

zvpanina@fedora:~$ git config --global user.name "Zhanna Panina"
zvpanina@fedora:~$ git config --global user.email "<1132246710@pfur.ru>"
zvpanina@fedora:~$

```

Рис. 4.2: Предварительная конфигурация git

Настроим utf-8 в выводе сообщений git, зададим имя начальной ветки (будем называть её master); параметр autocrlf; параметр safecrlf (рис. 4.3)

```

zvpanina@fedora:~$ git config --global core.quotepath false
zvpanina@fedora:~$ git config --global init.defaultBranch master
zvpanina@fedora:~$ git config --global core.autocrlf input
zvpanina@fedora:~$ git config --global core.safecrlf warn
zvpanina@fedora:~$

```

Рис. 4.3: Настройка соответствующих параметров

4.3 Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория генерирую пару ключей (приватный и открытый)(рис. 4.4)

```

zvpanina@fedora:~$ ssh-keygen -C "Zhanna Panina <1132246710@pfur.ru>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/zvpanina/.ssh/id_ed25519):
Created directory '/home/zvpanina/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/zvpanina/.ssh/id_ed25519
Your public key has been saved in /home/zvpanina/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:j0n0dZH0RbqrvpAA40amLPjzYaYAZWLzLvFENajJWus Zhanna Panina <1132246710@pfur.ru>
The key's randomart image is:
+--[ED25519 256]--+
|      .      .o|
|      .      .o o|
|.oo o .      o o|
|.oo ..      . .|
|. .o. oS . . .|
|.oo+ .+. * o .|
|+.oB+o = + . .|
| +++.      . .|
|. +E..      . +|
+----[SHA256]-----+
zvpanina@fedora:~$

```

Рис. 4.4: Создание SSH ключа

Далее мне необходимо загрузить сгенерированный открытый ключ. Для этого захожу на сайт <http://github.org/> под своей учётной записью и перехожу в меню Settings . После этого выбираю в боковом меню SSH and GPG keys и нажимаю кнопку New SSH key (рис. 4.5)

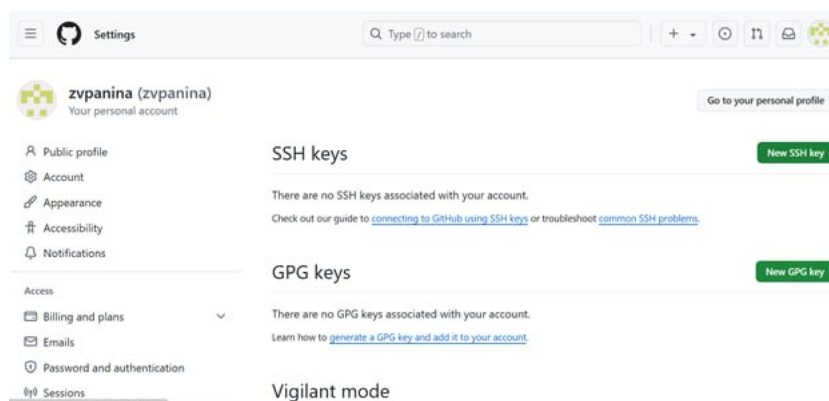


Рис. 4.5: Меню SSH and GPG keys

Скопировав из локальной консоли ключ в буфер обмена `cat ~/.ssh/id_rsa.pub | xclip -sel clip` нужно вставить ключ в появившееся на сайте поле и указать для ключа имя (Title) (рис. 4.6)

```
zvpanina@fedora:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
cat: /home/zvpanina/.ssh/id_rsa.pub: No such file or directory
bash: xclip: command not found...
Install package 'xclip' to provide command 'xclip'? [N/y]
```

Рис. 4.6: Предложение установки xclip

Как я поняла, введя эту команду, мне сначала нужно установить утилиту xclip. (рис. 4.7)

```
Install package 'xclip' to provide command 'xclip'? [N/y] y

* Waiting in queue...
* Loading list of packages...
The following packages have to be installed:
xclip-0.13-21.git11c6a61.fc40.x86_64  Command line clipboard grabber
Proceed with changes? [N/y] y

* Waiting in queue...
* Waiting for authentication...
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...

zvpanina@fedora:~$
```

Рис. 4.7: Установка xclip

Ввожу команду повторно и вижу, что нужная мне папка не находится, значит,

она скрыта, как видно через команду `ls -la`, т.е. мне нужно использовать права админа (команда `sudo`) (рис. 4.8)

```
zvpanina@fedora:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
cat: /home/zvpanina/.ssh/id_rsa.pub: No such file or directory
zvpanina@fedora:~$ ls -la
total 44
drwx----- 1 zvpanina zvpanina 804 Sep 21 10:52 .
drwxr-xr-x 1 root root 16 Sep 21 04:48 ..
-rw-r--r-- 1 zvpanina zvpanina 18 Feb 8 2024 .bash_logout
-rw-r--r-- 1 zvpanina zvpanina 144 Feb 8 2024 .bash_profile
-rw-r--r-- 1 zvpanina zvpanina 522 Feb 8 2024 .bashrc
drwx----- 1 zvpanina zvpanina 308 Sep 21 11:32 .cache
drwxr-xr-x 1 zvpanina zvpanina 348 Sep 21 11:32 .config
drwxr-xr-x 1 zvpanina zvpanina 0 Sep 21 04:48 Desktop
drwxr-xr-x 1 zvpanina zvpanina 0 Sep 21 04:48 Documents
drwxr-xr-x 1 zvpanina zvpanina 0 Sep 21 04:48 Downloads
-rw-r--r-- 1 zvpanina zvpanina 153 Sep 21 10:38 .gitconfig
drwx----- 1 zvpanina zvpanina 20 Sep 21 04:48 .local
drwxr-xr-x 1 zvpanina zvpanina 34 Apr 14 18:57 .mozilla
drwxr-xr-x 1 zvpanina zvpanina 0 Sep 21 04:48 Music
drwxr-xr-x 1 zvpanina zvpanina 0 Sep 21 04:48 Pictures
drwxr-xr-x 1 zvpanina zvpanina 0 Sep 21 04:48 Public
drwx----- 1 zvpanina zvpanina 48 Sep 21 10:52 .ssh
drwxr-xr-x 1 zvpanina zvpanina 0 Sep 21 04:48 Templates
-rw-r----- 1 zvpanina zvpanina 5 Sep 21 04:48 .vboxclient-clipboard-tty2-control.pid
-rw-r----- 1 zvpanina zvpanina 5 Sep 21 04:48 .vboxclient-clipboard-tty2-service.pid
-rw-r----- 1 zvpanina zvpanina 5 Sep 21 04:48 .vboxclient-draganddrop-tty2-control.pid
-rw-r----- 1 zvpanina zvpanina 5 Sep 21 04:48 .vboxclient-draganddrop-tty2-service.pid
-rw-r----- 1 zvpanina zvpanina 5 Sep 21 04:48 .vboxclient-hostversion-tty2-control.pid
-rw-r----- 1 zvpanina zvpanina 5 Sep 21 04:48 .vboxclient-seamless-tty2-control.pid
-rw-r----- 1 zvpanina zvpanina 5 Sep 21 04:48 .vboxclient-seamless-tty2-service.pid
drwxr-xr-x 1 zvpanina zvpanina 0 Sep 21 04:48 Videos
zvpanina@fedora:~$ sudo cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис. 4.8: Ввод команды от имени админа

Далее вставляю ключ в появившееся на сайте поле и указываю для ключа имя (Title) (рис. 4.9)

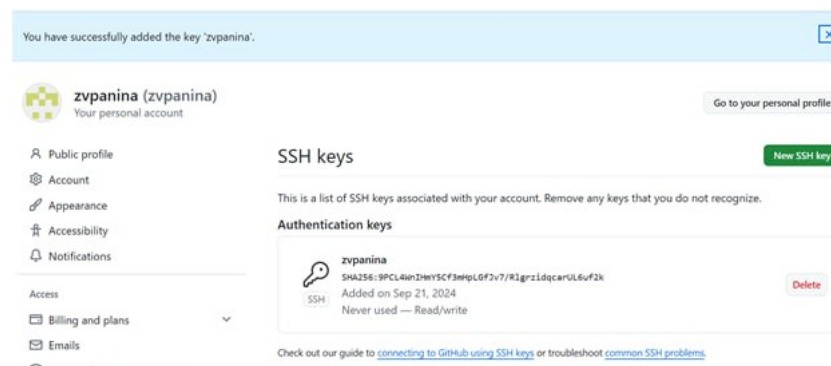


Рис. 4.9: Ключ создан

4.4 Сознание рабочего пространства и репозитория курса на основе шаблона

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

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

Рис. 4.10: Создание каталога

4.5 Сознание репозитория курса на основе шаблона

Репозиторий на основе шаблона можно создать через web-интерфейс github. Перехожу на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю Use this template.(рис. 4.11)

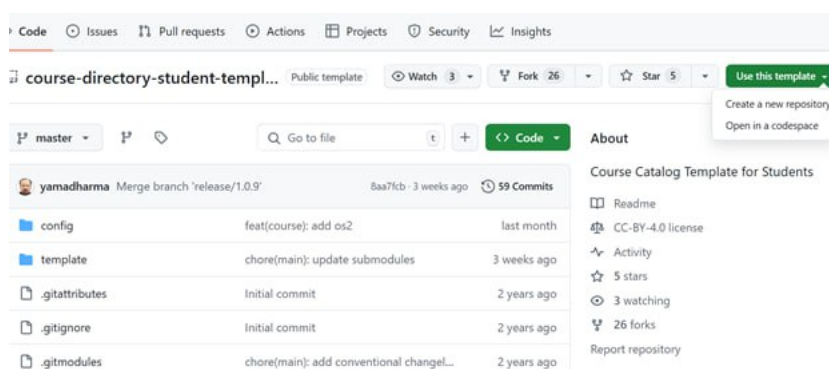



Рис. 4.11: Страница репозитория с шаблоном курса

В открывшемся окне задаю имя репозитория (Repository name) study_2023–2024_arh-
ps и создаю репозиторий (рис. 4.12)


Repository template

 yamadharma/course-directory-student-template ▾

Start your repository with a template repository's contents.

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


Owner * Repository name *


 zvpanina ▾ / study_2024-2025_arh-pc

✔ study_2024-2025_arh-pc is available.

Great repository names are short and memorable. Need inspiration? How about [jubilant-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.


 You are creating a public repository in your personal account.

Рис. 4.12: Создание репозитория

Открываю терминал и перехожу в каталог курса. `cd ~/work/study/2023–2024/“Архитектура компьютера”` Клонировую созданный репозиторий (рис. 4.13)

```
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:zvpanina/study_2024-2025_arh-pc.git
Cloning into 'study_2024-2025_arh-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 | 18.81 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/zvpanina/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-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 | 917.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/zvpanina/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arh-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 | 2.01 MiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера$
```

Рис. 4.13: Клонирование созданного репозитория

4.6 Настройка каталога курса

Перехожу в каталог курса и удаляю лишние файлы (рис. 4.14)

```
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_ar  
h-pc$ rm package.json  
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_ar  
h-pc$
```

Рис. 4.14: Удаление лишних файлов

Создаю необходимые каталоги (рис. 4.15)

```
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE  
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make  
Usage:  
  make <target>  
  
Targets:  
  list          List of courses  
  prepare       Generate directories structure  
  submodule     Update submules  
  
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE  
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare  
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.15: Создание каталогов

Отправляю файлы на сервер (рис. 4.16)

```
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE  
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make  
Usage:  
  make <target>  
  
Targets:  
  list          List of courses  
  prepare       Generate directories structure  
  submodule     Update submules  
  
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE  
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare  
zvpanina@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.16: Отправка файлов на сервер

(рис. 4.17)

```

create mode 100644 labs/lab11/report/Makefile
create mode 100644 labs/lab11/report/bib/cite.bib
create mode 100644 labs/lab11/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab11/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab11/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab11/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab11/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab11/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab11/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab11/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab11/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab11/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab11/report/report.md
create mode 100644 prepare
create mode 100644 presentation/README.md
create mode 100644 presentation/README.ru.md
create mode 100644 presentation/presentation/.projectile
create mode 100644 presentation/presentation/.texlabroot
create mode 100644 presentation/presentation/Makefile
create mode 100644 presentation/presentation/image/kulyabov.jpg
create mode 100644 presentation/presentation/presentation.md
create mode 100644 presentation/report/Makefile
create mode 100644 presentation/report/bib/cite.bib
create mode 100644 presentation/report/image/placeimg_800_600_tech.jpg
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/pandocattributes.py
create mode 100644 presentation/report/report.md
vpanina@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 36, done.
Counting objects: 100% (36/36), done.
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 341.39 KiB | 1.31 MiB/s, done.
Total 35 (delta 4), reused 1 (delta 0), pack-reused 0 (from 0)
Remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:zvpanina/study_2024-2025_arh-pc.git
  9e56998..20ce028 master -> master
vpanina@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc$

```

Рис. 4.17: Отправка файлов на сервер

Проверяю правильность создания иерархии рабочего пространства в локальном репозитории и на странице github (рис. 4.18)

The screenshot shows the GitHub interface for the repository 'study_2024-2025_arh-pc'. The repository is public and was generated from 'yamadharma/course-directory-student-template'. The file browser shows the 'master' branch with a search bar and a 'Code' button. Below the navigation bar, a commit by Zhanna Panina is shown, titled 'feat(main): make course structure', committed 2 minutes ago. The commit message is expanded to show the file hierarchy created:

File	Commit Message	Time
config	Initial commit	17 minutes ago
labs	feat(main): make course structure	2 minutes ago
presentation	feat(main): make course structure	2 minutes ago
template	Initial commit	17 minutes ago
.gitattributes	Initial commit	17 minutes ago

Рис. 4.18: Иерархия рабочего пространства создана

4.7 Выполнение заданий для самостоятельной работы

1. Создаю отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report). (рис. 4.19)



Рис. 4.19: Отчет в markdown

2. Копирую отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.
3. Загружаю файлы на github.

5 Выводы

В ходе выполнения работы я изучила идеологию и применение средств контроля версий, а также приобрела практические навыки по работе с системой git.

Список литературы

Архитектура ЭВМ