

Отчёт по лабораторной работе №2

Дисциплина: архитектура компьютера

Аннагулыев Арслан Мухаммедович

Содержание

1	Цель работы	1
2	Задание	1
3	Теоретическое введение.....	2
4	Выполнение лабораторной работы.....	3
4.1	Настройка GitHub.....	3
4.2	Базовая настройка Git.....	3
4.3	Создание SSH-ключа.....	4
4.4	Создание рабочего пространства и репозитория курса на основе шаблона.....	6
4.5	Создание репозитория курса на основе шаблона.....	7
4.6	Настройка каталога курса.....	9
4.7	Выполнение заданий для самостоятельной работы	11
5	Выводы	15
6	Список литературы	15

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

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, 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. 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. 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). 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.

4 Выполнение лабораторной работы

4.1 Настройка GitHub

Вхожу в давно созданную учетную запись на сайте GitHub (рис. 1).

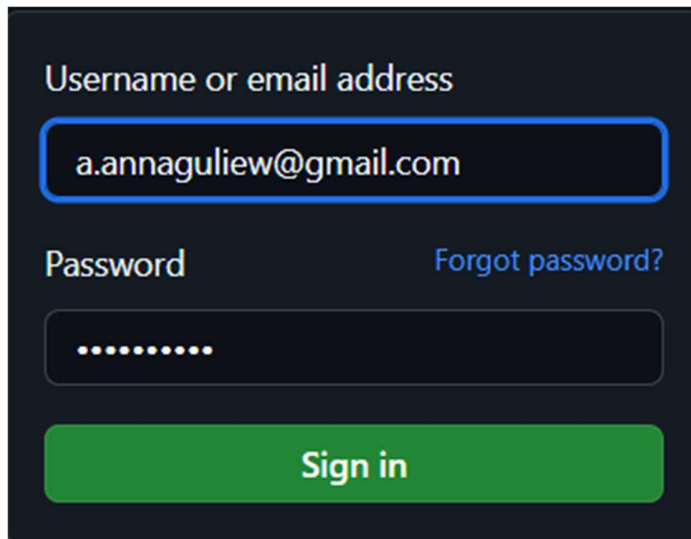
The image shows the GitHub login interface. It has a dark background. At the top, the text 'Username or email address' is in white. Below it is a text input field with a blue border containing the email 'a.annaguliew@gmail.com'. Underneath the email field is the label 'Password' in white, followed by a password input field with dots. To the right of the password field is a link 'Forgot password?' in blue. At the bottom is a large green button with the text 'Sign in' in white.

Рис. 1: Заполнение данных учетной записи GitHub

Вошел в аккаунт (рис. 2).

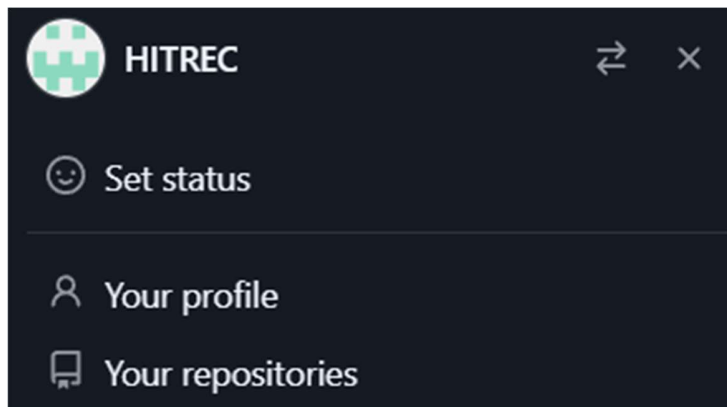


Рис. 2: Аккаунт GitHub

4.2 Базовая настройка Git

Открываю виртуальную машину, затем открываю терминал и делаю предварительную конфигурацию git. Ввожу команду `git config --global user.name ""`, указывая свое имя и команду `git config --global user.email "work@mail"`, указывая в ней электронную почту владельца, то есть мою (рис. 3).

```
vboxuser@rabot:~$ git config --global user.name "<Arslan Annagulyev>"
vboxuser@rabot:~$ git config --global user.email "<a.annagulyew@gmail.com>"
vboxuser@rabot:~$
```

Рис. 3: Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. 4).

```
vboxuser@rabot:~$ git config --global core.quotePath false
vboxuser@rabot:~$
```

Рис. 4: Настройка кодировки

Задаю имя «master» для начальной ветки (рис. 5).

```
vboxuser@rabot:~$ git config --global init.defaultBranch master
vboxuser@rabot:~$
```

Рис. 5: Создание имени для начальной ветки

Задаю параметр autocrlf со значением input, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. 6). CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

```
vboxuser@rabot:~$ git config --global core.autocrlf input
vboxuser@rabot:~$
```

Рис. 6: Параметр autocrlf

Задаю параметр safecrlf со значением warn, так Git будет проверять преобразование на обратимость (рис. 7). При значении warn Git только выведет предупреждение, но будет принимать необратимые конвертации.

```
vboxuser@rabot:~$ git config --global core.safecrlf warn
vboxuser@rabot:~$
```

Рис. 7: Параметр safecrlf

4.3 Создание SSH-ключа

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 (рис. 8). Ключ автоматически сохранится в каталоге `~/.ssh/`.

```

Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/vboxuser/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vboxuser/.ssh/id_ed25519
Your public key has been saved in /home/vboxuser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:ersq75v1hnhxpd3vSQGY0u48zH03lBM+GnVnetIGLL8 Arslan Annagulyev <a.annagulyew@gmail.com>
The key's randomart image is:
+--[ED25519 256]--+
|                .. |
|                . o..|
|                . + ..++|
|                o . +=*|
|                S  = ooB=|
|                .. B o =E+|
|                ..O+ * o =.|
|                . +oo. . o =|
|                +*+oo.   o.|
+-----[SHA256]-----+

```

Рис. 8: Генерация SSH-ключа

Xclip – утилита, позволяющая скопировать любой текст через терминал. Оказывается, в дистрибутиве Linux Kali ее сначала надо установить. Устанавливаю xclip с помощью команды apt-get install с ключом -у от имени суперпользователя, введя в начале команды sudo (рис. 9).

```

Command 'xclip' not found, but can be installed with:
sudo apt install xclip
vboxuser@rabot:~$ sudo apt-get install -y xclip
[sudo] password for vboxuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  xclip
0 upgraded, 1 newly installed, 0 to remove and 50 not upgraded.
Need to get 17.6 kB of archives.
After this operation, 54.3 kB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu noble/universe amd64 xclip amd64 0.13-3 [17.6 kB]
Fetched 17.6 kB in 0s (43.5 kB/s)
Selecting previously unselected package xclip.
(Reading database ... 152079 files and directories currently installed.)
Preparing to unpack .../xclip_0.13-3_amd64.deb ...
Unpacking xclip (0.13-3) ...
Setting up xclip (0.13-3) ...
Processing triggers for man-db (2.12.0-4build2) ...

```

Рис. 9: Установка утилиты xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip (рис. 10).

```

vboxuser@rabot:~$ cat ~/.ssh/id_ed25519.pub | xclip -sel clip

```

Рис. 10: Копирование содержимого файла

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key» (рис. 11).

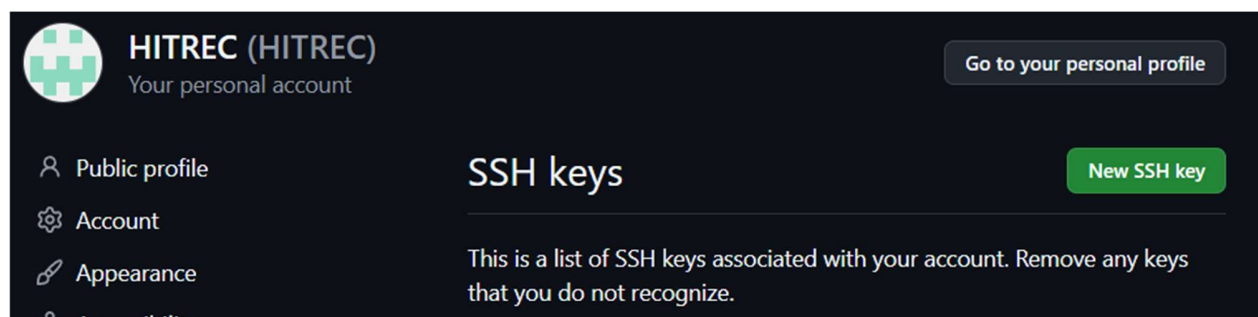


Рис. 11: Окно SSH and GPG keys

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. 12).

Title

Key type

Authentication Key ↕

Key

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMtejAFskXTLGUFPIvN12QC6ob+QmzAn2A2zm6d6xBM4 Arslan Annagulyev <a.annagulyew@gmail.com>
```

Рис. 12: Добавление ключа

4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Закрываю браузер, открываю терминал. 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. 13).

```
vboxuser@rabot:~$ mkdir -p work/study/2024-2025/"Архитектура компьютера"
vboxuser@rabot:~$ ls
Desktop  Downloads  old-work  parentdir1  parentdir3  Public  Templates  Videos
Documents  Music      parentdir  parentdir2  Pictures    snap     tmp         work
```

Рис. 13: Создание рабочего пространства

4.5 Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharm/course-directory-student-template>. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория (рис. 14).

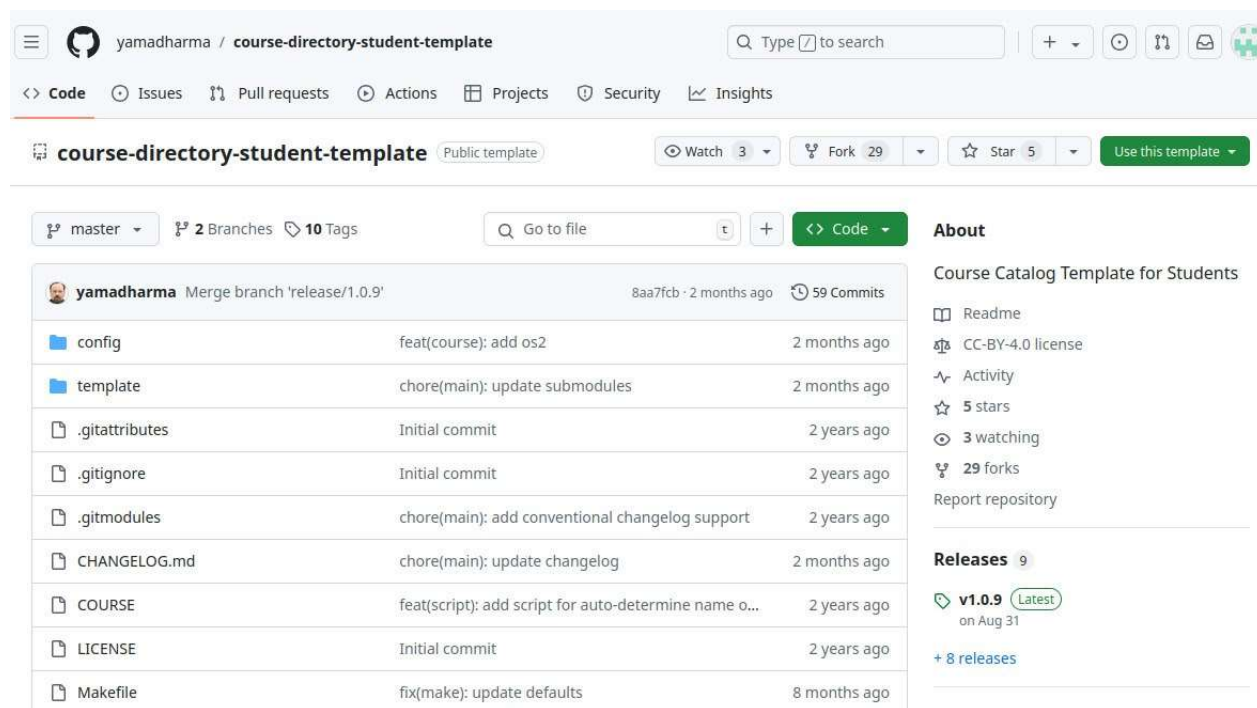


Рис. 14: Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study_2022–2023_arh-рс и создаю репозиторий, нажимаю на кнопку «Create repository from template» (рис. 15).

Owner * HITREC / **Repository name *** study_2024-2025_arh-pc
Checking availability...

Great repository names are short and memorable. Need inspiration? How about **laughing-parakeet** ?

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.

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

Рис. 15: Окно создания репозитория

Репозиторий создан (рис. 16).

 **study_2024-2025_arh-pc** Public Pin Unwatch 1
generated from [yamadharm/course-directory-student-template](#)

master 1 Branch Tags


 HITREC Initial commit	702883c · 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
📄 COURSE	Initial commit	now
📄 LICENSE	Initial commit	now

Рис. 16: Созданный репозиторий

Через терминал перехожу в созданный каталог курса с помощью утилиты cd (рис. 17).


```
vboxuser@rabot:~$ cd work/study/2024-2025/"Архитектура компьютера"
```

Рис. 17: Перемещение между директориями

Клонирую созданный репозиторий с помощью команды `git clone -recursive git@github.com:/study_2024-2025_arh-pc.git arch-pc` (рис. 18).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:HITREC/study_2024-2025_arh-pc.git arch-pc
Cloning into 'arch-pc'...
remote: Enumerating objects: 33, done.
remote: Total 10000 (delta 10000), reused 0 (delta 0), pack-reused 10000
```

Рис. 18: Клонирование репозитория

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 19).

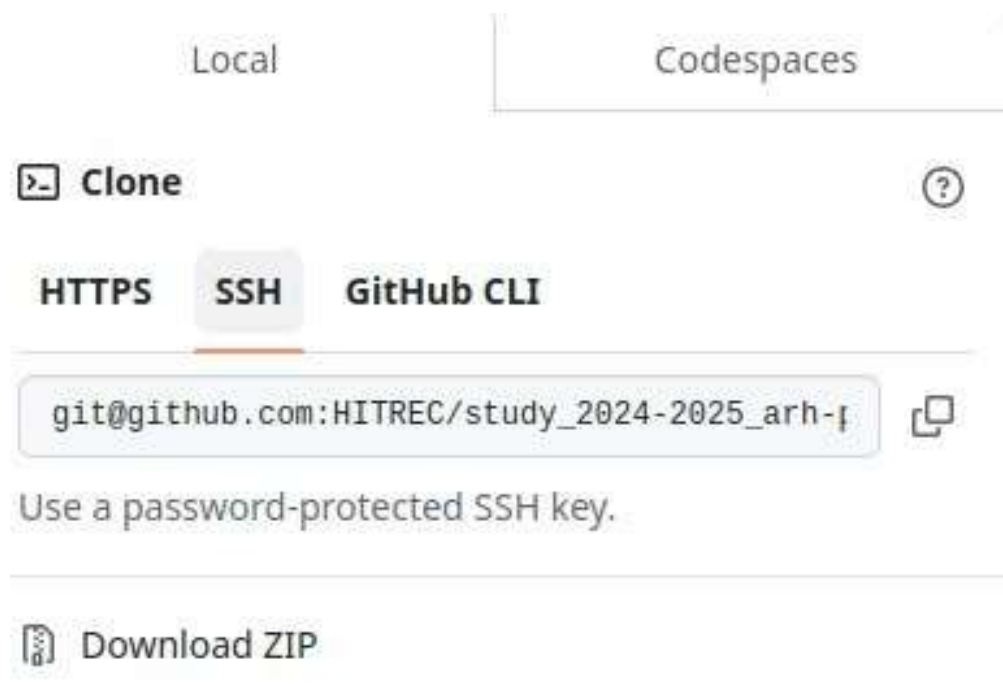


Рис. 19: Окно с ссылкой для копирования репозитория

4.6 Настройка каталога курса

Перехожу в каталог `arch-pc` с помощью утилиты `cd` (рис. 20).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера$ cd arch-pc
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 20: Перемещение между директориями

Удаляю лишние файлы с помощью утилиты `rm` (рис. 21).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 21: Удаление файлов

Создаю необходимые каталоги (рис. 22).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make
```

Рис. 22: Создание каталогов

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 (рис. 23).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master ac2c574] 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/.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
```

Рис. 23: Добавление и сохранение изменений на сервере

Отправляю все на сервер с помощью push (рис. 24).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 291.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:HITREC/study_2024-2025_arh-pc.git
702883c..2f63016 master -> master
```

Рис. 24: Выгрузка изменений на сервер

Проверяю правильность выполнения работы сначала на самом сайте GitHub (рис. 25).



 Arslan Annagulyev feat(main): make course structure		ac2c574 · 2 minutes ago	 History
Name	Last commit message	Last commit da...	
..	Last commit message		
lab01		feat(main): make course structure	
lab02	feat(main): make course structure	2 minutes ago	
lab03	feat(main): make course structure	2 minutes ago	
lab04	feat(main): make course structure	2 minutes ago	
lab05	feat(main): make course structure	2 minutes ago	
lab06	feat(main): make course structure	2 minutes ago	
lab07	feat(main): make course structure	2 minutes ago	
lab08	feat(main): make course structure	2 minutes ago	
lab09	feat(main): make course structure	2 minutes ago	
lab10	feat(main): make course structure	2 minutes ago	
lab11	feat(main): make course structure	2 minutes ago	
README.md	feat(main): make course structure	2 minutes ago	
README.ru.md	feat(main): make course structure	2 minutes ago	

Рис. 25: Страница репозитория

4.7 Выполнение заданий для самостоятельной работы

1. Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch (рис. 26).

```
vboxuser@rabort:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ touch Л02_Аннагулыев_отчёт
```

Рис. 26: Создание файла

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений (рис. 27).

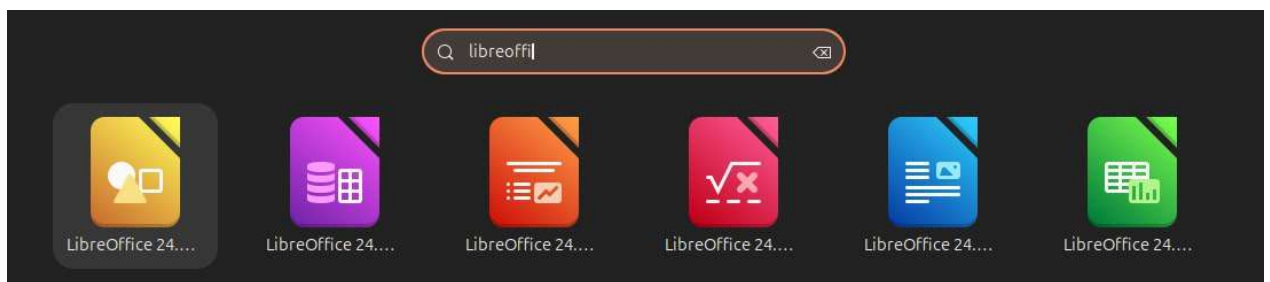


Рис. 27: Меню приложений

После открытия текстового процессора открываю в нем созданный файл и могу начать в нем работу над отчетом (рис. 28).

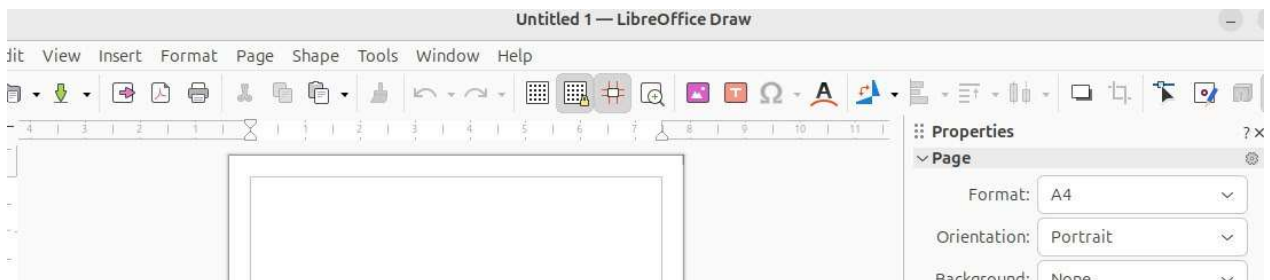


Рис. 28: Работа с отчетом в текстовом процессоре

2. Перехожу из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd (рис. 29).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ cd ..
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02$ cd ..
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs$ cd lab01
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01$ cd ..
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs$ cd lab01/report
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис. 29: Перемещение между директориями

Проверяю местонахождение файла с отчетом первой лабораторной работой. Они должны быть в подкаталоге домашней директории «Загрузки», для проверки использую команду ls (рис. 30).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ ls ~/Downloads
install-tl-20241020  install-tl-unx.tar.gz  lab01_Аннагульев_отчет-1.pdf
```

Рис. 30: Проверка местонахождения файлов

Копирую первую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls (рис. 31).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ cp ~/Downloads/lab01_Аннагульев_отчет-1.pdf ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ ls
bib image lab01_Аннагульев_отчет-1.pdf Makefile pandoc report.md
```

Рис. 31: Копирование файла

Перехожу из подкаталога lab01/report в подкаталог lab02/report с помощью утилиты cd (рис. 32).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ cd ..
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01$ cd ..
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs$ cd lab02/report
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$
```

Рис. 32: Перемещение между директориями

Копирую вторую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды (рис. 33).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ cp ~/Downloads/Л02_Аннагулыев_отчет.pdf /work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report
```

Рис. 33: Копирование файла

3. Добавляю с помощью команды git add в коммит созданные файлы: Л02_Аннагулыев_отчет (рис. 34). и

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ git add Л02_Аннагулыев_отчет.pdf
```

Рис. 34: Добавление файла на сервер

Перехожу в директорию, в которой находится отчет по первой лабораторной работе с помощью cd (рис. 35).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ cd ..
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02$ cd ..
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs$ cd lab01
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01$ cd ..
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs$ cd lab01/report
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рис. 35: Перемещение между директориями

Добавляю файл Lab01_Аннагулыев_отчет (рис. 36).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab01/report$ git add lab01_Аннагулыев_отчет-1.pdf
```

Рис. 36: Добавление файла на сервер

Сохраняю изменения на сервере командой git commit -m "...", поясняя, что добавила файлы.

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master 2f63016] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
```

Рис. 37: Подкаталоги и файлы в репозитории

Отправляю в центральный репозиторий сохраненные изменения командой git push -f origin master (рис. 38).

```
vboxuser@rabot:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab02/report$ git push -f origin master
```

Рис. 38: Отправка в центральный репозиторий сохраненных изменений

Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что пояснение к совершенным действиям отображается (рис. 39).

Arslan Annagulyev Add existing file aa01b16 · 4 minutes ago History		
Name	Last commit message	Last commit da...
..		
lab01	Add existing file	4 minutes ago
lab02	Add existing file	4 minutes ago

Рис. 39: Страница каталога в репозитории

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в lab01/report (рис. 40), по второй – в lab02/report (рис. 41).

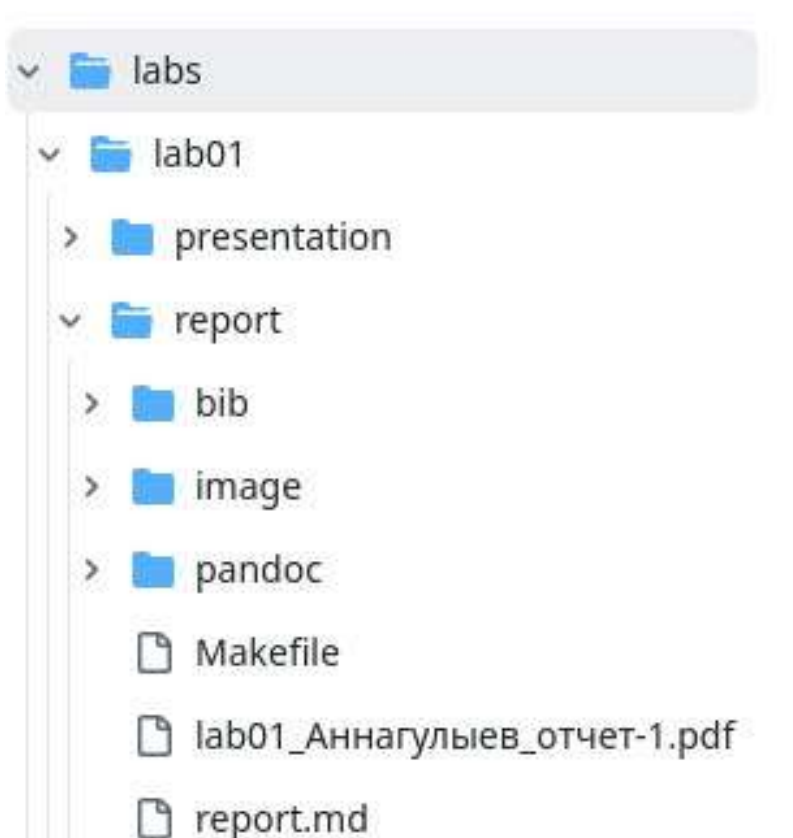


Рис. 40: Каталог lab01/report

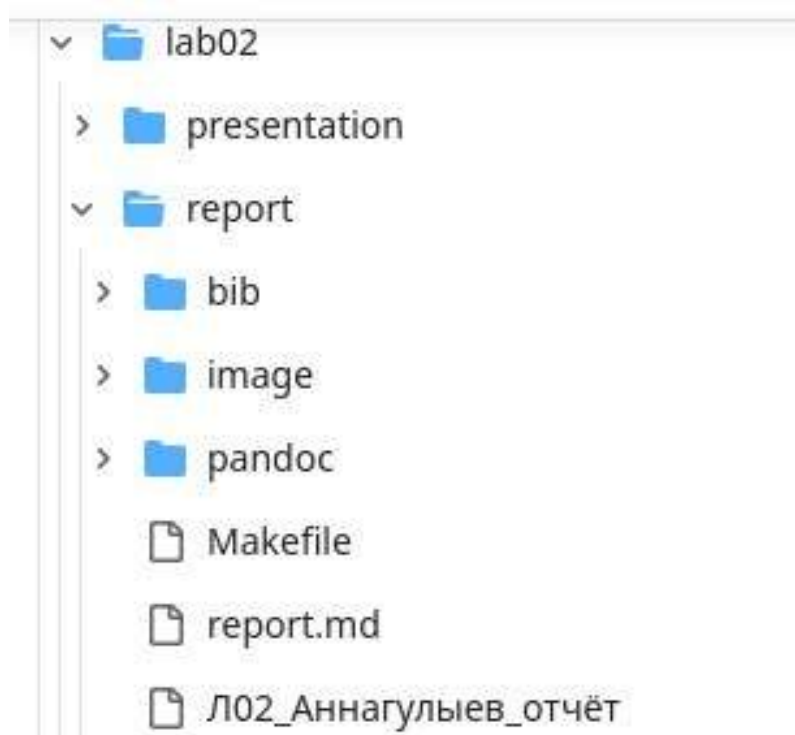


Рис. 41: Каталог *lab02/report*

5 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрела практические навыки по работе с системой git.

6 Список литературы

1. [Архитектура ЭВМ](#)
2. [Git - gitattributes Документация](#)