

Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютера

Ермакова Анастасия Алексеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Настройка github	9
4.2	Базовая настройка git	9
4.3	Создание SSH ключа	10
4.4	Создание рабочего пространства и репозитория курса на основе шаблона	12
4.5	Создание репозитория курса на основе шаблона	13
4.6	Настройка каталога курса	16
5	Выполнение самостоятельной работы	19
6	Выводы	22
7	Список литературы	23

Список иллюстраций

4.1	Учетная запись на сайте GitHub	9
4.2	Предварительная конфигурация git	9
4.3	Настройки git	10
4.4	Создание ключей	10
4.5	Окно SSH and GPG keys на сайте	11
4.6	Установка утилиты xclip	11
4.7	Копирование ключа	12
4.8	Ключ на сайте GitHub	12
4.9	Создание каталога “Архитектура компьютера”	12
4.10	Страница репозитория курса	13
4.11	Создание репозитория на основе шаблона	14
4.12	Созданный репозиторий	15
4.13	Перемещение в директорию курса	15
4.14	Клонирование репозитория	16
4.15	Удаление ненужных файлов	16
4.16	Создание каталогов	17
4.17	Установка команды make	17
4.18	Создание и отправка на сервер файлов и каталогов	17
4.19	Создание и отправка на сервер файлов и каталогов	17
4.20	Иерархия рабочего пространства на странице github	18
5.1	Расположение файла с отчетом	19
5.2	Копирование файла	19
5.3	Расположение файла в нужном каталоге	19
5.4	Загрузка файла на github	20
5.5	Расположение файла отчета №2. Копирование файла	20
5.6	Расположение файла в нужном каталоге	20
5.7	Загрузка файла на сайт	20
5.8	Репозиторий на сайте	21

Список таблиц

1 Цель работы

Цель данной работы – изучение идеологии и применение средств контроля версий, а также приобретение практических навыков 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. Настройки каталога курса
7. Выполнение 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 (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зави-

симости 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.

4 Выполнение лабораторной работы

4.1 Настройка github

Для начала создам учетную запись на сайте <https://github.com/> и заполню основные данные. (рис. 4.1)

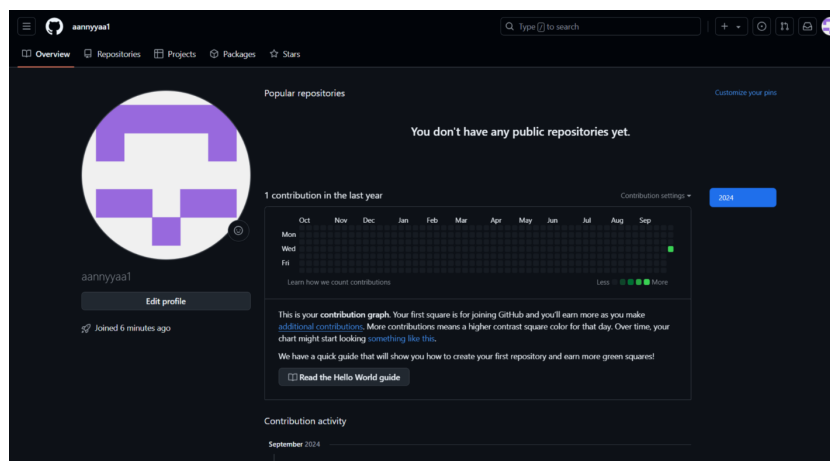


Рис. 4.1: Учетная запись на сайте GitHub

4.2 Базовая настройка git

Делаю предварительную конфигурацию git. Открываю терминал и ввожу следующие команды, указав свое имя и email. (рис. 4.2)

```
liveuser@localhost-live:~$ git config --global user.name "<Nastya Yermakova>"
liveuser@localhost-live:~$ git config --global user.email "<ermakova6363@gmail.com>"
```

Рис. 4.2: Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов. Задаю имя master для начальной ветки. Ввожу параметры autocrlf и safecrlf. (рис. 4.3)

```
liveuser@localhost-live:~$ git config --global core.quotePath false
liveuser@localhost-live:~$ git config --global init.defaultBranch master
liveuser@localhost-live:~$ git config --global core.autocrlf input
liveuser@localhost-live:~$ git config --global core.safecrlf warn
```

Рис. 4.3: Настройки git

4.3 Создание SSH ключа

Генерирую пару ключей (приватный и открытый) для последующей идентификации на сервере. Для этого ввожу ssh-keygen -C с указанием своего имени и почты. Ключи автоматически сохраняются в каталоге ~/.ssh/. (рис. 4.4)

```
liveuser@localhost-live:~$ ssh-keygen -C "Nastya Yermakova <ermakova6363@gmail.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_ed25519):
Created directory '/home/liveuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liveuser/.ssh/id_ed25519
Your public key has been saved in /home/liveuser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:yu2hpPiqaY2SKgwX2FgYeA2t2PhNaedkGTCT9cv4DPY Nastya Yermakova <ermakova6363@gmail.com>
>
The key's randomart image is:
+--[ED25519 256]--+
|oo.+++          |
| o o +o..       |
| X . . o.       |
|= = + =o .      |
| . = += S       |
| . o .o.B       |
|ooo  + E        |
|+= o o o .      |
|O.ooo . .       |
+----[SHA256]-----+
```

Рис. 4.4: Создание ключей

Затем захожу на сайт [//github.org/](https://github.org/) в свою учетную запись, перехожу в меню Settings, в меню SSH and GPG keys нажимаю New SSH key. (рис. 4.5)

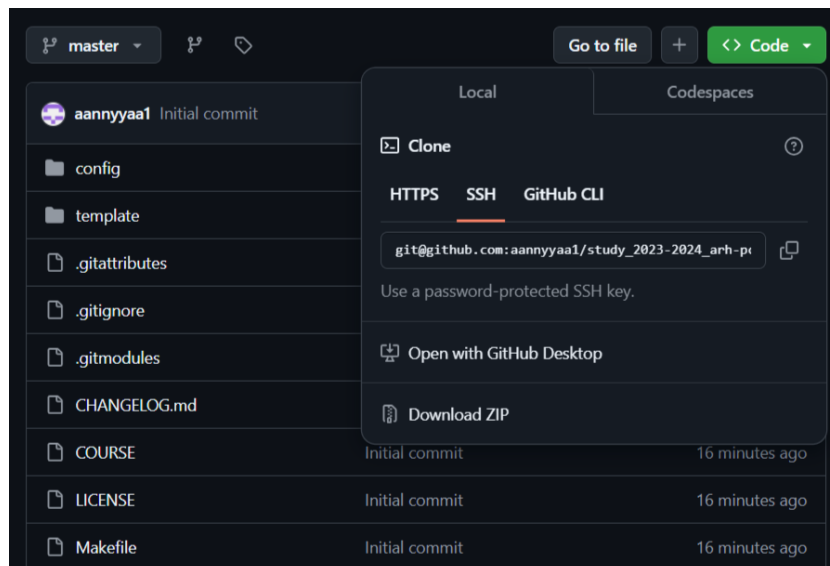


Рис. 4.5: Окно SSH and GPG keys на сайте

Копирую из локальной консоли ключ в буфер обмена с помощью утилиты xclip. Для этого ее сначала потребуется установить следующим образом: (рис. 4.6)

```
liveuser@localhost-live:~$ sudo dnf install -y xclip
Fedora 40 - x86_64                                3.3 MB/s | 20 MB    00:06
Fedora 40 openh264 (From Cisco) - x86_64         2.1 kB/s | 1.4 kB   00:00
Fedora 40 - x86_64 - Updates                      3.3 MB/s | 10 MB    00:03
Last metadata expiration check: 0:00:01 ago on Wed 25 Sep 2024 03:09:12 PM EDT.
Dependencies resolved.
=====
Package            Architecture Version                                Repository      Size
=====
Installing:
xclip              x86_64      0.13-21.git11cba61.fc40              fedora          37 k
Transaction Summary
=====
Install 1 Package

Total download size: 37 k
Installed size: 62 k
Downloading Packages:
xclip-0.13-21.git11cba61.fc40.x86_64.rpm          175 kB/s | 37 kB    00:00
-----
Total                                           34 kB/s | 37 kB    00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : xclip-0.13-21.git11cba61.fc40.x86_64 1/1
  Running scriptlet: xclip-0.13-21.git11cba61.fc40.x86_64 1/1
Installed:
xclip-0.13-21.git11cba61.fc40.x86_64
Complete!
```

Рис. 4.6: Установка утилиты xclip

После установки ввожу код, чтобы скопировать ключ, с указанием директории.

(рис. 4.7)

```
liveuser@localhost-live:~$ cat /home/liveuser/.ssh/id_ed25519.pub | xclip -sel clip
```

Рис. 4.7: Копирование ключа

Скопированный ключ вставляю его в поле на сайте и указываю название ключа «Ключ». (рис. 4.8)

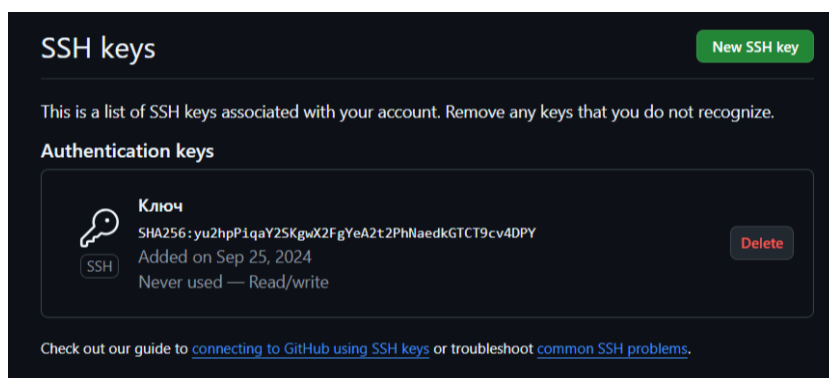


Рис. 4.8: Ключ на сайте GitHub

4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Для того, чтобы при выполнении лабораторных работ придерживаться структуры рабочего пространства, создаю каталог для предмета «Архитектура компьютера». Ввожу следующий код и с помощью команды `ls` убеждаюсь, что каталог создан. (рис. 4.9)

```
liveuser@localhost-live:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
liveuser@localhost-live:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  work
```

Рис. 4.9: Создание каталога “Архитектура компьютера”

4.5 Создание репозитория курса на основе шаблона

Для создания репозитория буду использовать готовый шаблон курса (<https://github.com/yamadharma/course-directory-student-template>) на сайте GitHub. Захожу на страницу репозитория и нажимаю Use this template. (рис. 4.10)

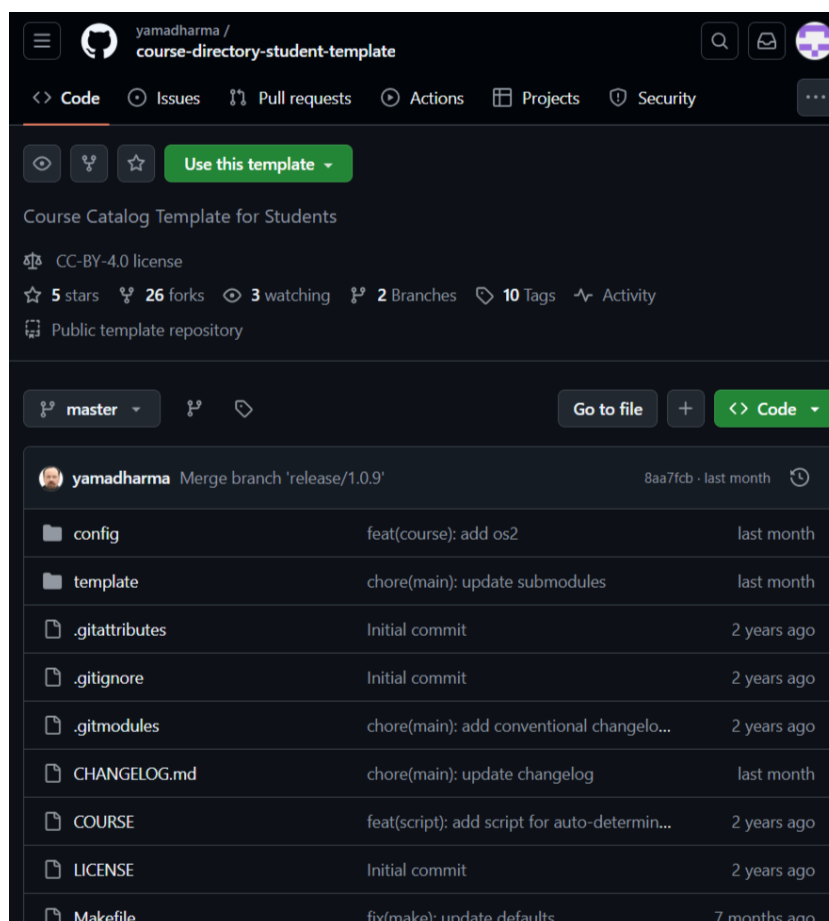


Рис. 4.10: Страница репозитория курса

В открывшемся окне задаю имя репозитория (study_2023-2024_arh-rc) и нажимаю Create repository from template. (рис. 4.11)


New repository

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 ().*


Repository template

 yamadharm/course-directory-student-template ▾

Start your repository with a template repository's contents.

☐ **Include all branches**
Copy all branches from yamadharm/course-directory-student-template and not just the default branch.


Owner * **Repository name ***


 aannyaa1 ▾ / study_2023-2024_arh-pc


✓ Your new repository will be created as study_2023-2024_arh-pc-.
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about [sturdy-memory](#) ?

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

Рис. 4.11: Создание репозитория на основе шаблона

Репозиторий создан. (рис. 4.12)

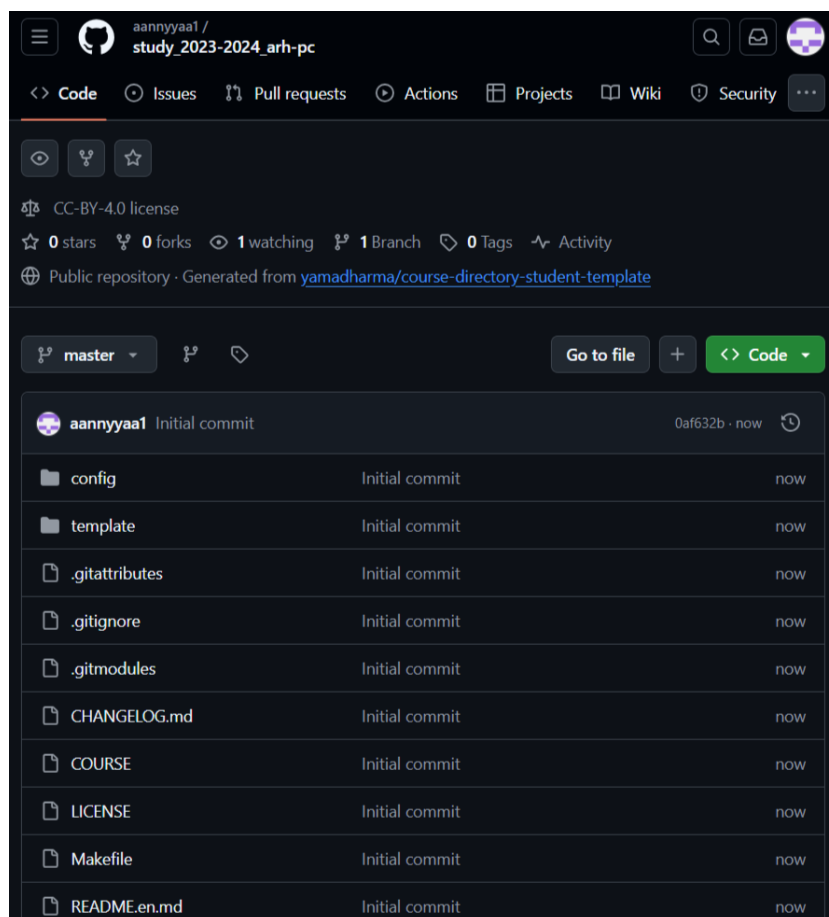


Рис. 4.12: Созданный репозиторий

Открываю терминал и перехожу в каталог курса с помощью утилиты cd. (рис. 4.13)

```
liveuser@localhost-live:~$ cd ~/work/study/2023-2024/"Архитектура компьютера"
```

Рис. 4.13: Перемещение в директорию курса

Клонирую созданный репозиторий, указав ссылку для клонирования из окна SSH. (рис. 4.14)

```
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера$ git clone --recursive git@github.com:aannyaa1/study_2023-2024_arh-pc.git arch-pc
Cloning into '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)
Receiving objects: 100% (33/33), 18.81 KiB | 688.00 KiB/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/liveuser/work/study/2023-2024/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)
Receiving objects: 100% (111/111), 102.17 KiB | 702.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/liveuser/work/study/2023-2024/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)
Receiving objects: 100% (142/142), 341.09 KiB | 1.19 MiB/s, done.
Resolving deltas: 100% (60/60), done.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a02bd2fca1d4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab185f5c748'
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера$
```

Рис. 4.14: Клонирование репозитория

4.6 Настройка каталога курса

С помощью утилиты `cd` перехожу в каталог курса. С помощью утилиты `rm` удаляю ненужные файлы. (рис. 4.15)

```
liveuser@localhost-live:~$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ rm package.json
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ make
bash: make: command not found...
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рис. 4.15: Удаление ненужных файлов

Создаю необходимые каталоги, ввожу `echo arch-pc > Course` (рис. 4.16), затем утилиту `make`. Данная команда не находится, об этом говорит сообщение «`bash: make: command not found...`», скачиваю ее. Обновляю базу данных `yum` с помощью `dnf`, ввожу команду `sudo dnf makecache --refresh`. После обновления базы данных скачиваю `make` с помощью команды `sudo dnf -y install make`. Команда установлена. (рис. 4.17)


```
liveuser@localhost-live:~$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
```

Рис. 4.16: Создание каталогов

```
liveuser@localhost-live:~$ make
bash: make: command not found...
liveuser@localhost-live:~$ sudo dnf makecache --refresh
Fedora 40 - x86_64                                31 kB/s | 28 kB    00:00
Fedora 40 openh264 (From Cisco) - x86_64          4.5 kB/s | 989 B  00:00
Fedora 40 - x86_64 - Updates                       49 kB/s | 9.9 kB  00:00
Fedora 40 - x86_64 - Updates                       3.3 MB/s | 3.8 MB  00:01
Metadata cache created.
liveuser@localhost-live:~$ sudo dnf -y install make
Last metadata expiration check: 0:00:34 ago on Sat 28 Sep 2024 06:41:50 AM EDT.
Dependencies resolved.
=====
Package                Architecture      Version           Repository        Size
=====
Installing:
make                   x86_64            1:4.4.1-6.fc40    fedora            588 k
Installing dependencies:
gc                      x86_64            8.2.2-6.fc40     fedora            110 k
guile30                x86_64            3.0.7-12.fc40    fedora            8.1 M
=====
Transaction Summary
=====
Install 3 Packages

Total download size: 8.8 M
Installed size: 54 M
Downloading Packages:
(1/3): gc-8.2.2-6.fc40.x86_64.rpm                  159 kB/s | 110 kB    00:00
(2/3): make-4.4.1-6.fc40.x86_64.rpm                 593 kB/s | 588 kB    00:00
(3/3): guile30-3.0.7-12.fc40.x86_64.rpm             2.2 MB/s | 8.1 MB    00:03
-----
Total                                           1.9 MB/s | 8.8 MB    00:04
Running transaction check
  Installing      : make-1:4.4.1-6.fc40.x86_64                3/3
  Running scriptlet: make-1:4.4.1-6.fc40.x86_64                3/3
Installed:
gc-8.2.2-6.fc40.x86_64      guile30-3.0.7-12.fc40.x86_64      make-1:4.4.1-6.fc40.x86_64
Complete!
```

Рис. 4.17: Установка команды make

После успешной установки использую команду еще раз, но пишу make с таргетом prepare, затем отправляю файлы на сервер. (рис. 4.18,4.19)

```
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ make prepare
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master f0d8ae3] 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
```

Рис. 4.18: Создание и отправка на сервер файлов и каталогов

```
liveuser@localhost-live:~/work/study/2023-2024/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 | 2.25 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:aannyaa1/study_2023-2024_arh-pc.git
 e09c31d..f0d8ae3 master -> master
```

Рис. 4.19: Создание и отправка на сервер файлов и каталогов

Проверяю правильность создания иерархии рабочего пространства на странице github. (рис. 4.20)

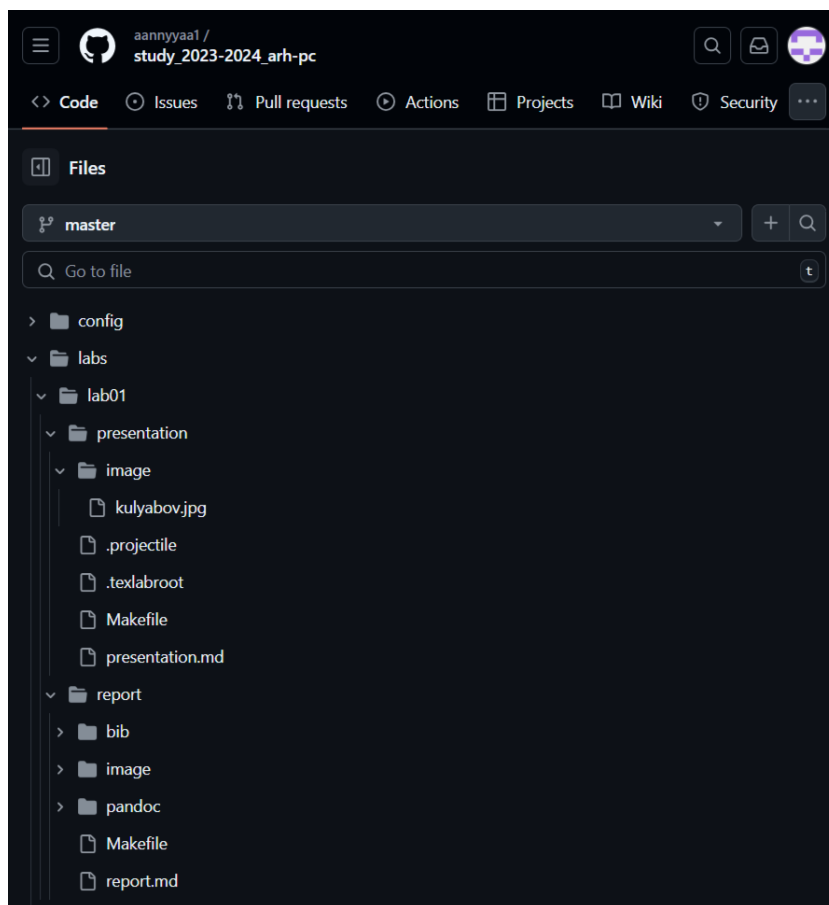


Рис. 4.20: Иерархия рабочего пространства на странице github

5 Выполнение самостоятельной работы

Скопирую отчет по выполнению первой лабораторной работе в каталог labs/lab01/report. Для начала проверю лежит ли сохраненный файл первой лабораторной работы в папке Downloads с помощью команды ls. (рис. 5.1)

```
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab03/report$ ls ~/Downloads
Л01_Ермакова_отчет.pdf
```

Рис. 5.1: Расположение файла с отчетом

Затем с помощью утилиты cp копирую файл в нужный мне каталог. (рис. 5.2)

```
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ cp ~/Downloads/
Л01_Ермакова_отчет.pdf /home/liveuser/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
```

Рис. 5.2: Копирование файла

Проверяю правильность выполнения команды с помощью команды ls. (рис. 5.3)

```
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ ls
bib image Makefile pandoc report.md Л01_Ермакова_отчет.pdf
```

Рис. 5.3: Расположение файла в нужном каталоге

Загружаю файл на сайт github. (рис. 5.4)

```
liveuser@localhost-live:~$ cd /home/liveuser/work/study/2023-2024/"Архитектура компьютера"/arch-pc
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ cd labs/lab01/report
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ ls
bib image Makefile pandoc report.md Л01_Ермакова_отчет.pdf
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ git add Л01_Ермакова_отчет.pdf
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ git commit -m "
Add existing file"
[master e2fe4f9] Add existing file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Ермакова_отчет.pdf
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ git push -f ori
gin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 625.90 KiB | 4.41 MiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:aannyaa1/study_2023-2024_arh-pc.git
f0d8ae3..e2fe4f9 master -> master
```

Рис. 5.4: Загрузка файла на github

Те же самые действия выполняю с файлом данной лабораторной работы. Проверяю содержание файла в каталоге Downloads. Копирую его в нужный каталог. (рис. 5.5)

```
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ ls ~/Downloads
Л01_Ермакова_отчет.pdf Л02_Ермакова_отчет.pdf
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ cp ~/Downloads/
Л02_Ермакова_отчет.pdf /home/liveuser/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab02/report
```

Рис. 5.5: Расположение файла отчета №2. Копирование файла

Проверяю, находится ли теперь файл в каталоге labs/lab02/report с помощью команды ls. (рис. 5.6)

```
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ cd labs/lab02/report
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ ls
bib image Makefile pandoc report.md Л02_Ермакова_отчет.pdf
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$
```

Рис. 5.6: Расположение файла в нужном каталоге

И наконец загружаю файл на сайт github. (рис. 5.7)

```
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ git add Л02_Ермакова_отчет.pdf
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ git commit -m "
Add existing file"
[master 3a8179d] Add existing file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab02/report/Л02_Ермакова_отчет.pdf
liveuser@localhost-live:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ git push -f ori
gin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.10 MiB | 7.58 MiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:aannyaa1/study_2023-2024_arh-pc.git
e2fe4f9..3a8179d master -> master
```

Рис. 5.7: Загрузка файла на сайт

Непосредственно на сайте проверяю, все ли программы были выполнены верно. Да, файлы с отчетами загрузились в нужные каталоги. (рис. 5.8)

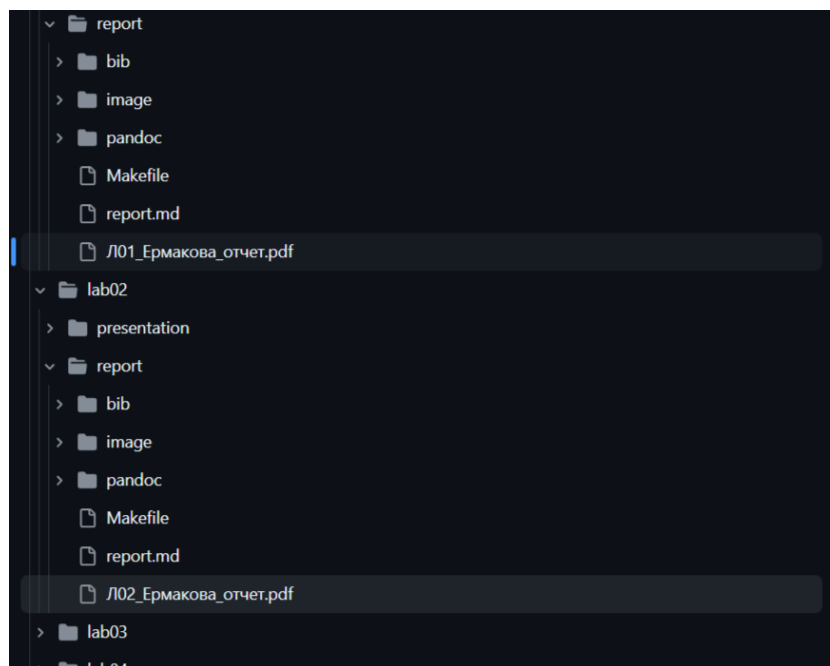


Рис. 5.8: Репозиторий на сайте

6 Выводы

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

7 Список литературы

1. Архитектура ЭВМ