

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
ФАКУЛЬТЕТ ФИЗИКО-МАТЕМАТИЧЕСКИХ И
ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ТЕОРИИ
ВЕРОЯТНОСТЕЙ**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2**

дисциплина: Архитектура компьютера и операционные
системы

Студент: Приспешкин А.А
Группа: НКАбд-05-23

МОСКВА
2023 г.

Содержание

1. Цель работы.....	3
2. Задание.....	4
3. Выполнение лабораторной работы.....	5
4. Выводы.....	17

1) Цель работы

Целью работы является изучить применение средств для контроля версий, а также приобрести практические навыки по работе с системой контроля версий git.

2) Задания

- 1) Настройка GitHub.
- 2) Базовая настройка git.
- 3) Создание SSH-ключа.
- 4) Создание рабочего пространства.
- 5) Создание репозитория на основе шаблона.
- 6) Настройка каталога курса.
- 7) Задания для самостоятельной работы.

3) Выполнение лабораторной работы

1) Создадим учётную запись на GitHub (рис.1).

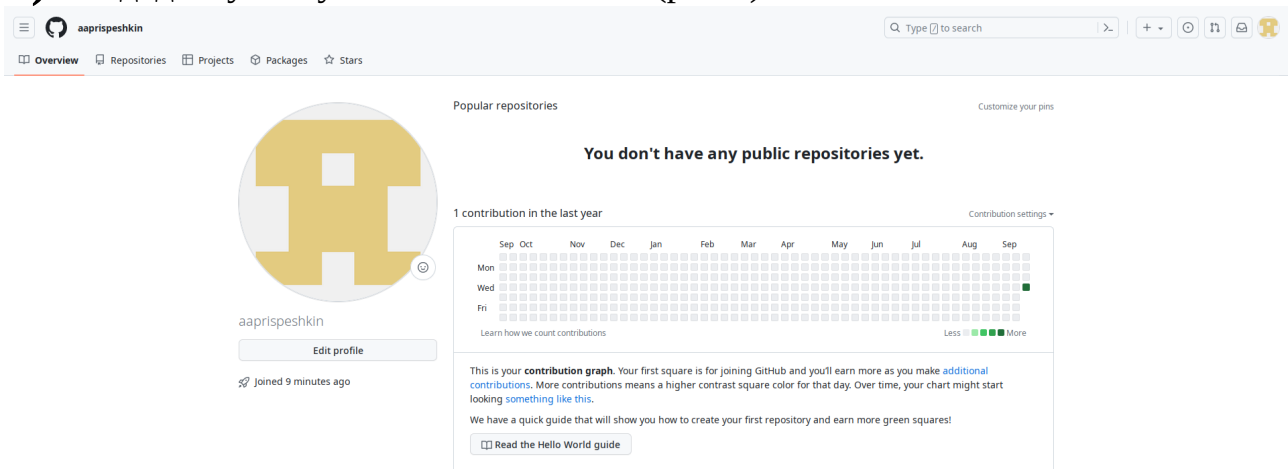


Рис.1 Созданный аккаунт на Github

2) Откроем терминал и введём команду git config, указав имя фамилию и электронную почту пользователя (рис.2).

```
aapripeshkin:[aapripeshkin]:~$ git config --global user.name "Andrey Prispeshkin"
aapripeshkin:[aapripeshkin]:~$ git config --global user.email "prspandrey@gmail.com"
```

Рис. 2 Настройка git с помощью команды git config

Продолжая использовать команду git config, настраиваем кодировку utf-8 для вывода сообщений (рис.3) и создаём начальную ветку “master” (рис.4).

```
aapripeshkin:[aapripeshkin]:~$ git config --global core.quotepath false
aapripeshkin:[aapripeshkin]:~$
```

Рис.3 Настройка вывода сообщений в кодировке utf-8

```
aapripeshkin:[aapripeshkin]:~$ git config --global init.defaultBranch master
```

Рис.4 Создание начальной ветки

Зададим параметр autocrlf со значением input (рис.5).

```
aapripeshkin:[aapripeshkin]:~$ git config --global core.autocrlf input
```

Рис. 5 Задание параметра autocrlf

Чтобы git мог определять преобразование на обратимость, зададим параметр autocrlf со значением warn(рис.6)

```
aapripeshkin:[aapripeshkin]:~$ git config --global core.autocrlf warn
```

Рис.6 Задание параметра autocrlf со значением warn

3) Для последующей работы с git и github, нам потребуется сгенерировать уникальный ssh ключ. Сделать это можно командой ssh-keygen(рис.7).

```
aapripeshkin:[aapripeshkin]:~$ ssh-keygen -C "Andrey Prispeshkin <prspandrey@gmail.com>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aapripeshkin/.ssh/id_rsa):
Created directory '/home/aapripeshkin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aapripeshkin/.ssh/id_rsa
Your public key has been saved in /home/aapripeshkin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:MYFcqWSNurm0FDRDy0BBrqjVYgsfTvybCySCfEfdS2U Andrey Prispeshkin <prspandrey@gmail.com>
The key's randomart image is:
+---[RSA 3072]-----+
|o= o.. =o. E      |
|o + + B +.o      |
| o . B ooo       |
|=. .+ . .o.      |
|B.O..= S.        |
|oX.=*            |
|. =o.o           |
| .oo            |
| +.             |
+---[SHA256]-----+
aapripeshkin:[aapripeshkin]:~$
```

Рис. 7 Генерация ssh ключа с помощью команды ssh-keygen

Скопируем ключ из директории в которой он был сохранён, используя команды cat и xclip(рис.8)

```
aapripeshkin:[aapripeshkin]:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

Рис. 8 Использование команды xclip для копирования ключа из терминала в буфер обмена

Затем, загружаем скопированный ключ в GitHub. В настройках выберем пункт “SSH and GPG keys”(рис.9), там выберем “Add new SSH key” и вставим скопированный ранее ключ(рис.10).

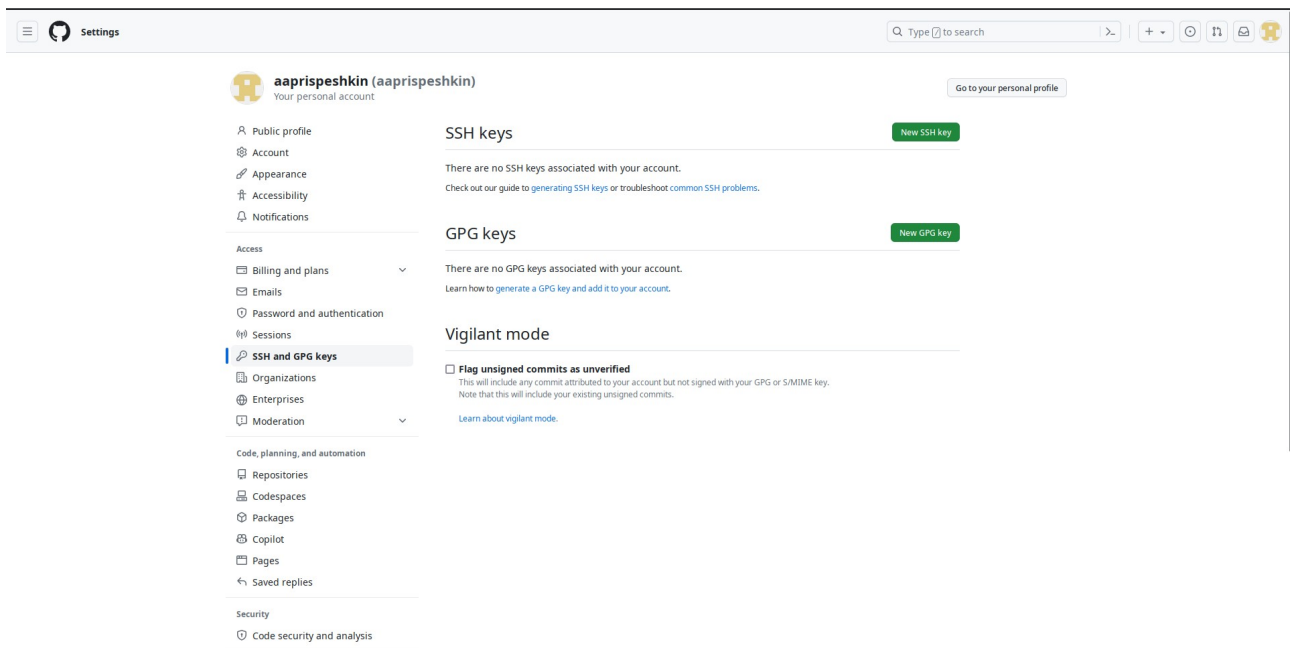


Рис. 9 Меню SSH and GPG keys в настройках

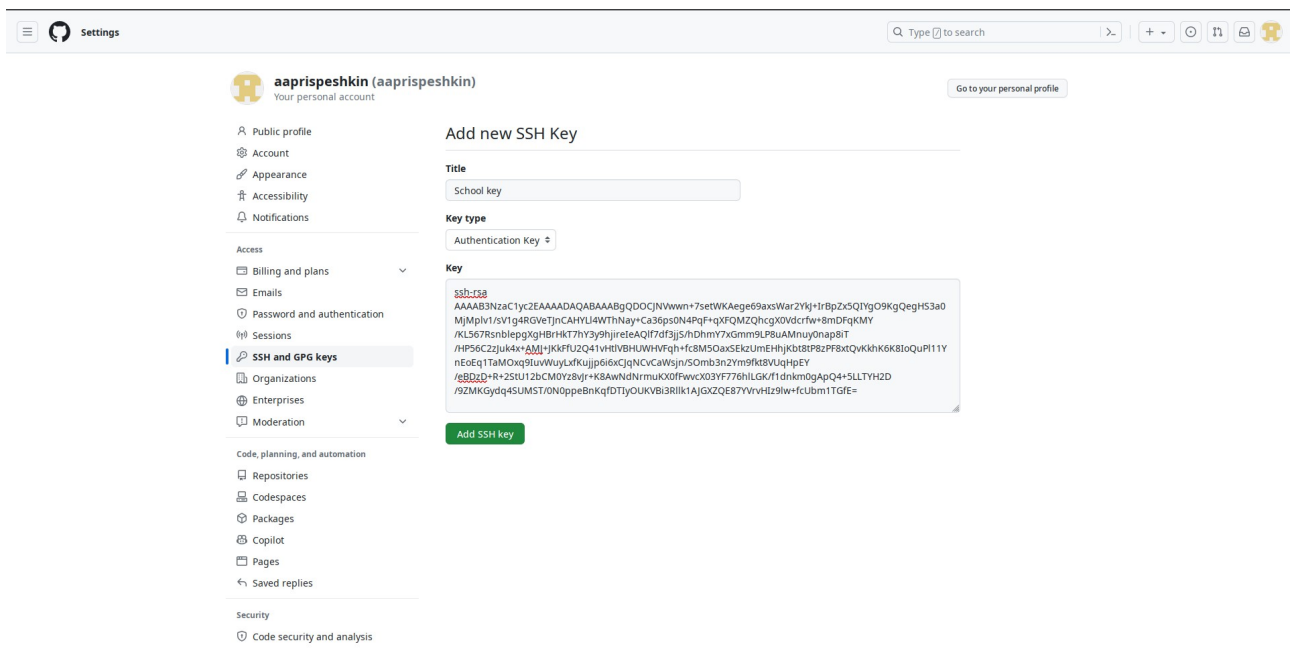


Рис.10 Вставленный SSH ключ сгенерированный ранее

4) С помощью утилиты `mkdir` с ключом `-p` создадим рабочую директорию(рис.11).

```
aapripeshkin: aapripeshkin] :~$ mkdir -p ~/work/study/2023-2024/"Архитектура Компьютера"
aapripeshkin: aapripeshkin] :~$
```

Рис. 11 Создание рабочей директории

5) Перейдём на страницу с шаблоном курса (<https://github.com/yamadharma/course-directory-student-template>) и скопируем шаблон (рис.12, 13 и 14)

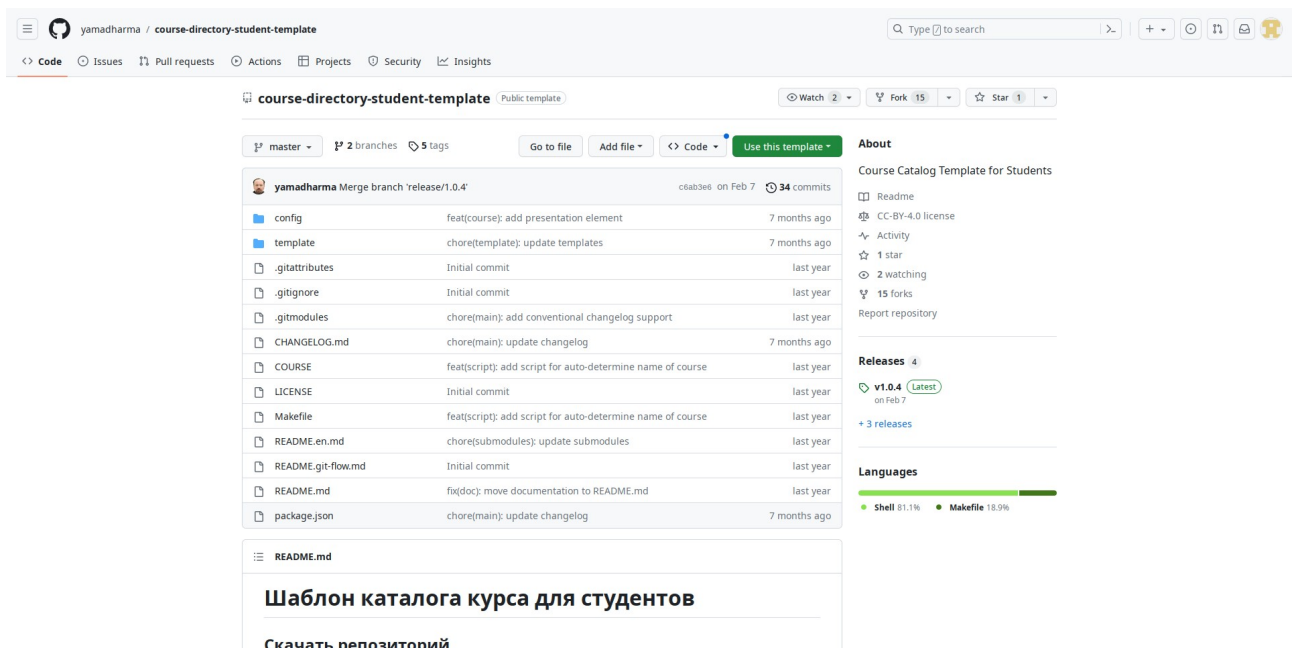


Рис. 12 Шаблон каталога на Github

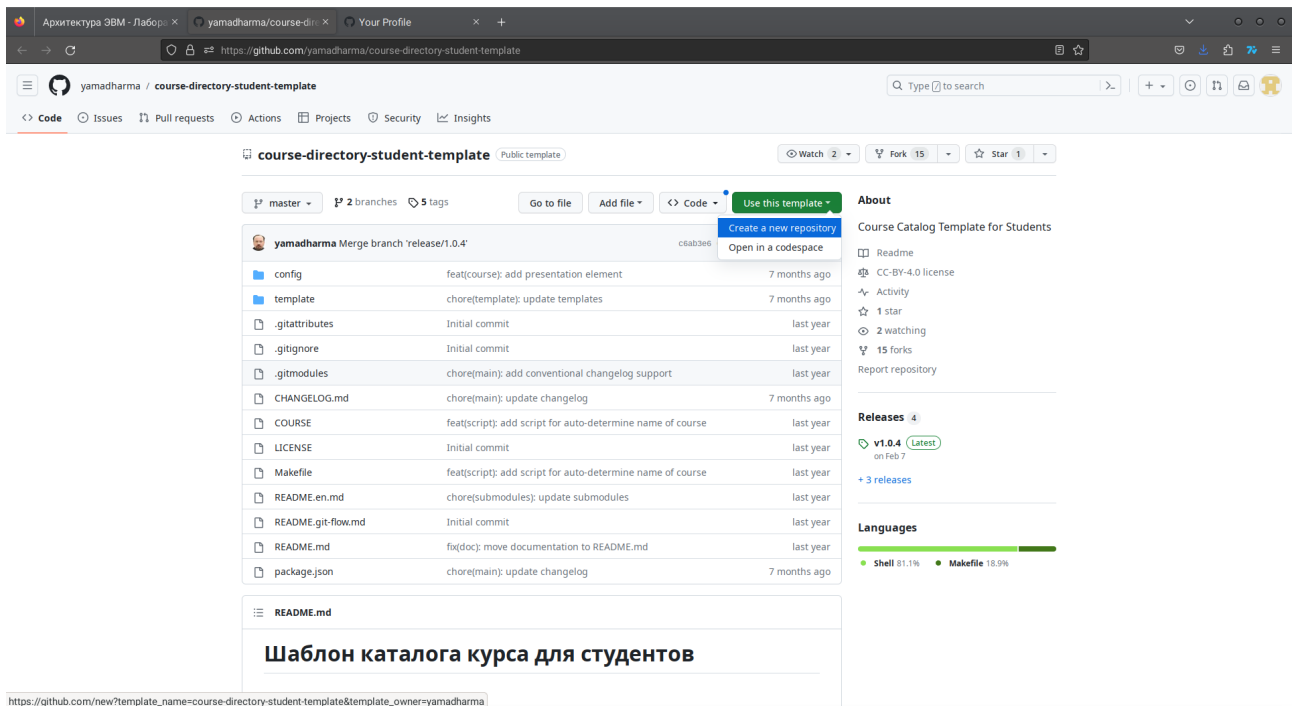


Рис.13 Создание нового репозитория на основе шаблона

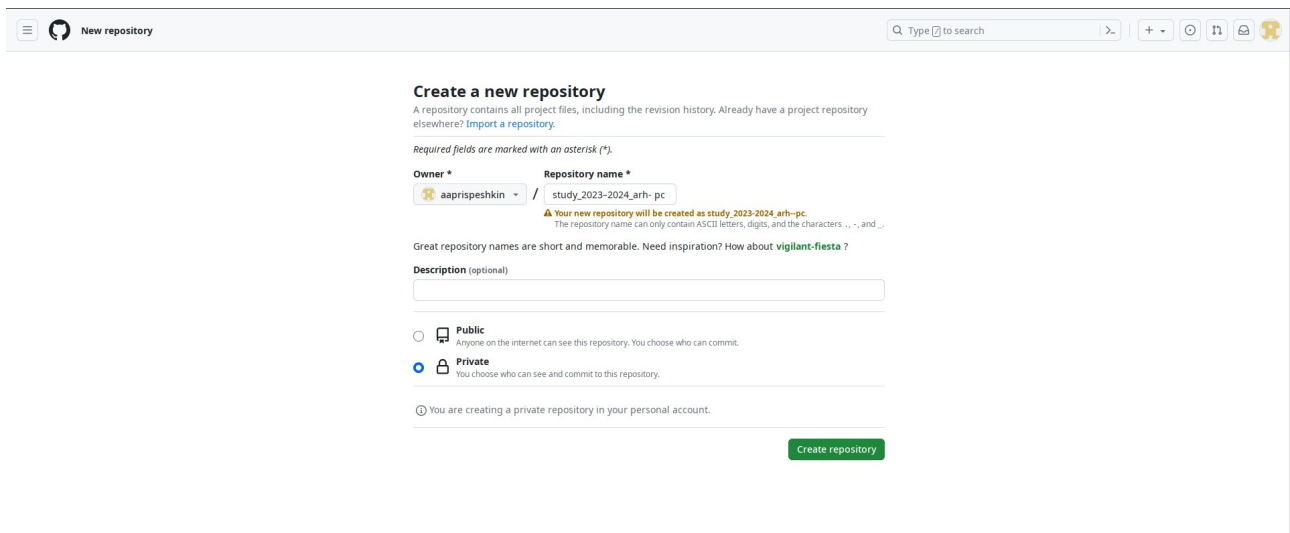


Рис.14 Окно создания нового репозитория

Пользуясь утилитой `cd` перейдём в каталог курса(рис.15).

```
aapripeshkin: [aapripeshkin] :~$ cd ~/work/study/2023-2024/Архитектура\ Компьютера/
```

Рис.15 Переход в созданный каталог курса

Командой `git clone` с ключом `--recursive` и пользуясь ссылкой на github клонируем созданный репозиторий(рис.16).

```
aapripeshkin: [aapripeshkin] :~/work/study/2023-2024/Архитектура Компьютера$ git clone --recursive git@github.com:aapripeshkin/study_2023-2024_arh--pc.git
Cloning into 'study_2023-2024_arh--pc' ...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:4DiY3wV6TujJhbpZisF/zLDA8zPMSvHdkR4UvCOqJ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
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
Receiving objects: 100% (27/27), 16.93 KiB | 963.00 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/aapripeshkin/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--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
Receiving objects: 100% (82/82), 92.99 KiB | 559.00 KiB/s, done.
Resolving deltas: 100% (28/28), done.
Cloning into '/home/aapripeshkin/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--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 48), reused 88 (delta 27), pack-reused 0
Receiving objects: 100% (101/101), 327.25 KiB | 761.00 KiB/s, done.
Resolving deltas: 100% (40/40), done.
Submodule path 'template/presentation': checked out 'b1be3808ee91f5889264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dca9c287a83917b82e3aef11a33b1e3b2'
aapripeshkin: [aapripeshkin] :~/work/study/2023-2024/Архитектура Компьютера$
```

Рис.16 Клонирование репозитория

6) Утилитой `rm` удалим лишние файлы из каталога курса(рис.17).

```
aapripeshkin: [aapripeshkin] :~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc$ rm package.json
```

Рис.17 Удаление файла package.json из каталога курса

Командами `echo` и `make` создадим необходимые каталоги(рис.18).

```
aapripeshkin: [aapripeshkin] :~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc$ echo arch-pc > COURSE
aapripeshkin: [aapripeshkin] :~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc$ make
```

Рис.18 Создание нужных каталогов командами `echo` и `make`

Командой `git add` добавим созданные каталоги на сервер, а командой `git commit` сохраним сделанные изменения(рис.19).

```
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc$ git add .
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc$ git commit -am 'feat(main): make course structure'
[master eab3bf8] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
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/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-9-5-2008-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
```

Рис.19 Использование команд `git add` и `git commit`

Командой `git push` отправим все сделанные изменения на сервер(рис.20).

```
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 12 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.14 KiB | 2.88 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:aapripeshkin/study_2023-2024_arh--pc.git
 3ce8af9..eab3bf8 master -> master
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc$
```

Рис.20 Отправка сделанных изменений на сервер

Проверим правильность сделанных изменений на github(рис.21).

Commits

History for [study_2023-2024_arh--pc / labs](#)

Commits on Sep 20, 2023

feat(main): make course structure

aapripeshkin committed last week

eab3bf8

<>

End of commit history for this file

Рис.21 Проверка истории изменений на github

7) Перейдём в подкаталог `lab02` и с помощью утилиты `touch` создадим файл отчёта по лабораторной работе(рис.22).

```
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs$ cd lab02
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab02$ touch Лабораторная2_Приспешкин
```

Рис.22 Создание отчёта по лабораторной работе (составлен отчёт будет в текстовом редакторе LibreOffice Writer)

Скопируем отчёт по первой лабораторной работе из каталога `Documents` в каталог `lab01/report` и проверим правильность копирования утилитой `ls`(рис.23).

```
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab03/report$ cd ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab01/report/
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab01/report$ cp ~/Documents/Лабораторная1_Приспешкин.pdf ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab01/report
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab01/report$ ls
bib image Makefile pandoc report.md Лабораторная1_Приспешкин.pdf
aapripeshkin: aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab01/report$
```

Рис.23 Копирование отчёта в рабочий репозиторий

Командой `git add` добавим отчёт по первой лабораторной работе на сервер, повторим эти действия с отчётом по второй лабораторной работе(рис.24, 25 и 26)

```
aapripeshkin@aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab01/report$ git add Лабораторная1_Приспешкин.pdf
aapripeshkin@aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab01/report$ git commit -m "Added lab report 1"
[master 838af9f] Added lab report 1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Лабораторная1_Приспешкин.pdf
```

Рис. 24 Добавление отчёта по первой лабораторной работе

```
aapripeshkin@aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab02$ git add Лабораторная2_Приспешкин
```

Рис.25 Добавление отчёта по второй лабораторной работе

```
aapripeshkin@aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab02$ git commit -m "Added lab report 2"
[master 6a19315] added lab2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab02/Лабораторная2_Приспешкин
aapripeshkin@aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab02$
```


Рис.26 Использование команды `git commit` для сохранения изменений по отчёту второй лабораторной работы



Командой `git push` отправим все созданные файлы в центральный репозиторий(рис.27).

```
aapripeshkin@aapripeshkin: ~/work/study/2023-2024/Архитектура Компьютера/study_2023-2024_arh--pc/labs/lab03/report$ git push -f origin master
Enumerating objects: 17, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 1.08 MiB | 7.55 MiB/s, done.
Total 11 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
To github.com:aapripeshkin/study_2023-2024_arh--pc.git
 eab3bf8..3ffec3 master -> master
```

Рис.27 Отправка добавленных нами файлов в центральный репозиторий командой `git push`

Проверим правильность введенных нами команд в GitHub(рис.28).

study_2023-2024_arh--pc / labs / 

 aapripeshkin added lab2 6a19315 · 9 minutes ago  History

Name	Last commit message	Last commit date
..		
lab01	Added lab report 1	44 minutes ago
lab02	added lab2	9 minutes ago

Рис.28 Проверка правильности добавления отчётов по лабораторной работе

4) Вывод

В результате выполнения данной лабораторной работы я научился пользоваться системой контроля версий git в терминале Linux, а также создавать и обновлять репозитории в GitHub.

5) Список литературы

1. Лабораторная работа №2. Система контроля версий Git