

Отчёт по лабораторной работе №2

Дисциплина: архитектура компьютера

Кайнова Алина Андреевна

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	10
4.5	Создание репозитория курса на основе шаблона	10
4.6	Настройка каталога курса	12
4.7	Выполнение заданий для самостоятельной работы	13
5	Выводы	16
	Список литературы	17

Список иллюстраций

Список таблиц

1.

1 Цель работы

Изучить идеологию систем контроля версий и научиться применять их, в частности, приобрести навыки по работе с Git.

2.

2 Задание

3. Настройка Github
4. Базовая настройка Git
5. Создание SSH ключа
6. Создание рабочего пространства и репозитория курса на основе шаблона
7. Создание репозитория курса на основе шаблона
8. Настройка каталога курса
9. Задания для самостоятельной работы
- 10.

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 необходимо. 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 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

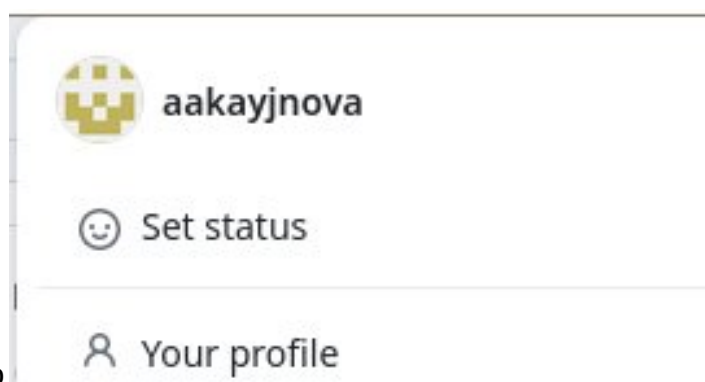
не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, 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.

4.

4 Выполнение лабораторной работы

4.1 Настройка Github

- Создадим учётную запись на сайте GitHub



4.2 Базовая настройка Git

- Сделаем предварительную конфигурацию Git

```
[aakaynova@fedora ~]$ git config --global user.name Al
```

```
[aakaynova@fedora ~]$ git config --global user.email 11
```
- Настроим utf-8

```
[aakaynova@fedora ~]$ git config --global core.quotePath false
```

```
[aakaynova@fedora ~]$
```
- Зададим имя начальной ветви

```
[aakaynova@fedora ~]$ git config --global init.defaultBranch master
```

```
[aakaynova@fedora ~]$
```
- Зададим параметр autocrlf

```
[aakaynova@fedora ~]$ git config --global core.autocrlf input
```
- Зададим параметр safecrlf

```
[aakaynova@fedora ~]$ git config --global core.safecrlf warn
```

```
[aakaynova@fedora ~]$
```

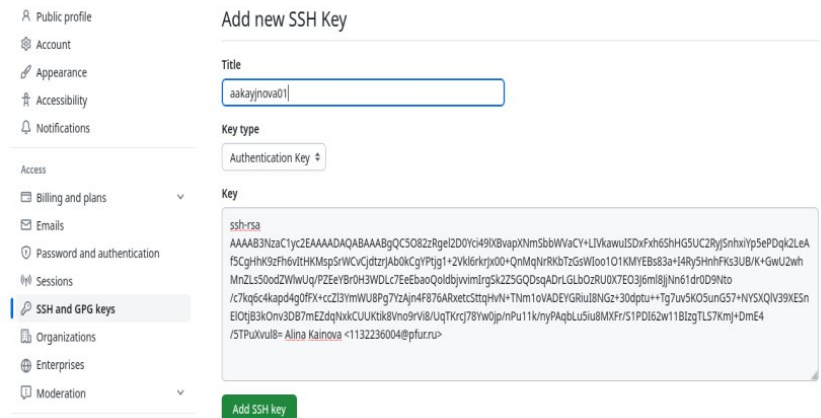
4.3 Создание SSH ключа

- Генерируем пару ключей

```
[aakayjnova@fedora ~]$ ssh-keygen -C "Alina Kainova <1132236004@pfur.ru>"  
Generating public/private rsa key pair.
```

- Копируем ключ из локальной консоли

```
[aakayjnova@fedora ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel c  
[aakayjnova@fedora ~]$
```



- Добавляем новый ключ

4.4 Создание рабочего пространства и репозитория курса на основе шаблона

- Создадим каталог для данного предмета через терминал и проверим пра-

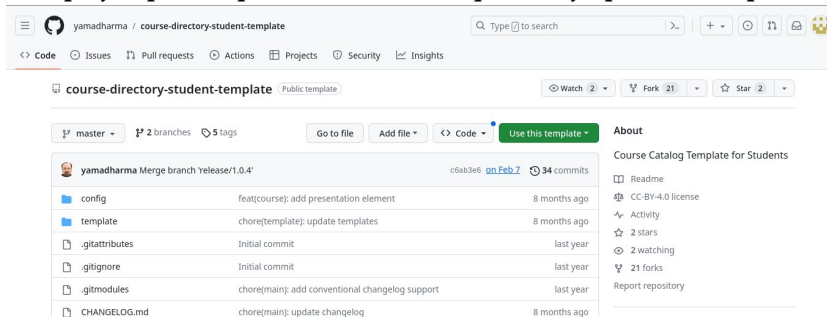
вильность выполнения команды

```
[aakayjnova@fedora ~]$ mkdir -p work/study/2023-2024/"Архитектура компьютера"  
[aakayjnova@fedora ~]$ ls
```

work Видео Документы Загрузки Изображения Музыка Общедоступные 'Рабочий стол' Шаблоны

4.5 Создание репозитория курса на основе шаблона

- В браузере переходим на страницу репозитория с шаблоном курса



Create a new repository

A repository contains all project files, including the revision history. Already have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

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 *

 aakayjnova ▾

Repository name *

study_2023-2024_arh-pc

✓ study_2023-2024_arh-pc is available.

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

Description (optional)

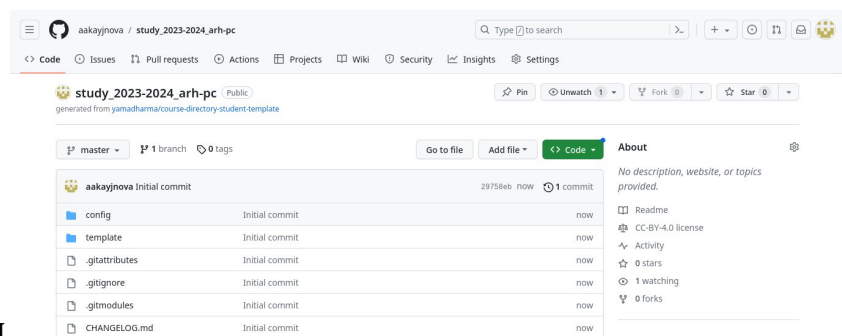
☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

- Зададим имя репозитория и создадим его

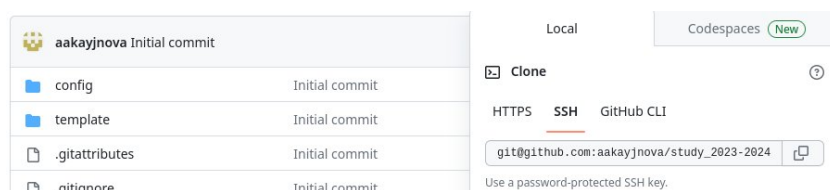


- Репозиторий создан

- Переходим в созданный каталог курса через терминал

```
[aakayjnova@fedora ~]$ cd ~/work/study/2023-2024
[aakayjnova@fedora Архитектура компьютера]$
```

- Копируем ссылку репозитория



```
[aakayjnova@fedora ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"
[aakayjnova@fedora Архитектура компьютера]$ git clone -recursive git@github.com:aakayjnova/arch-pc.git arch-pc
error: имелось в виду '--recursive' (с двумя дефисами)?
[aakayjnova@fedora Архитектура компьютера]$ git clone --recursive git@github.com:aakayjnova/arch-pc.git arch-pc
Клонирование в «arch-pc»...
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
Получение объектов: 100% (27/27), 16.93 КиБ | 5.64 МБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-...) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template) зарегистрирован по пути «template/report»
Клонирование в «/home/aakayjnova/work/study/2023-2024/Архитектура компьютера/arch-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
Получение объектов: 100% (82/82), 92.90 КиБ | 1.04 МБ/с, готово.
Определение изменений: 100% (28/28), готово.
Клонирование в «/home/aakayjnova/work/study/2023-2024/Архитектура компьютера/arch-pc/template/report»...
remote: Enumerating objects: 101, done.
```

- Клонировать репозиторий, используя ссылку

4.6 Настройка каталога курса

- Переходим в каталог arch-pc

```
[aakayjnova@fedora Архитектура компьютера]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
[aakayjnova@fedora arch-pc]$
```

- Удаляем лишние файлы

```
[aakayjnova@fedora arch-pc]$ rm package.json
[aakayjnova@fedora arch-pc]$
```

- Создаём необходимые каталоги

```
[aakayjnova@fedora arch-pc]$ echo arch-pc > COURSE
[aakayjnova@fedora arch-pc]$ make
[aakayjnova@fedora arch-pc]$
```

- Добавляем все созданные в ходе работы каталоги и сохраняем изменения

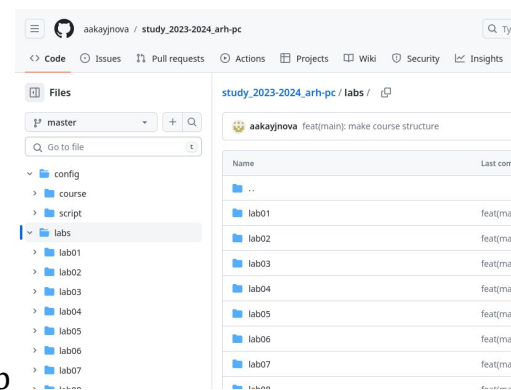
```
[aakayjnova@fedora arch-pc]$ git add .
[aakayjnova@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master 4283168] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
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-7-0-5-2008-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/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
```

на сервере

- Отправляем всё на сервер

```
[aakayjnova@fedora arch-pc]$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 2.80 МиБ/с, готово.
Всего 35 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:aakayjnova/study_2023-2024_arh-pc.git
29758eb..4283168 master -> master
```

- Проверяем правильность выполнения работы на Github



4.7 Выполнение заданий для самостоятельной работы

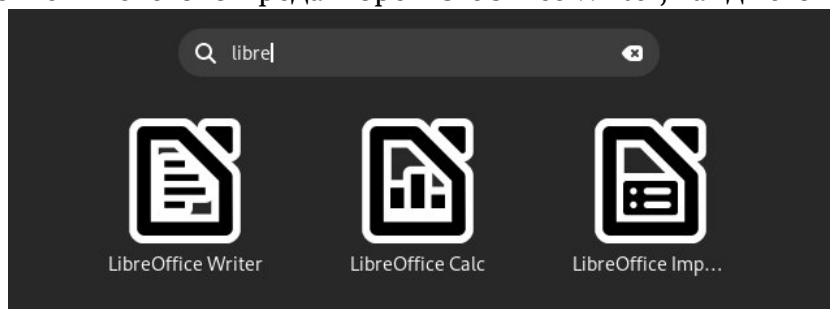
1. Отчёт по выполнению лабораторной работы в соответствующем каталоге рабочего пространства

- Переходим в директорию labs/lab02/report и создаём файл для отчёта по второй лабораторной работе

```
[aakayjnova@fedora arch-pc]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab02/report
[aakayjnova@fedora report]$ touch Л02_Кайнова_отчёт
```

- Оформляем отчёт в текстовом редакторе LibreOffice Writer, найдя его в меню

приложений



2. Копирование отчёта предыдущей лабораторной работы в соответствующие каталоги рабочего пространства


```
[aakayjnova@fedora report]$ cd ..
[aakayjnova@fedora lab02]$ cd ..
[aakayjnova@fedora labs]$ cd lab01/
[aakayjnova@fedora lab01]$ cd ..
[aakayjnova@fedora labs]$ cd lab01/report
[aakayjnova@fedora report]$ |
```

- Переходим в подкаталог lab01/report

- Проверяем местонахождение файла с первой лабораторной работой

```
[aakayjnova@fedora report]$ ls ~/Загрузки
Л01_Кайнова_отчёт.pdf
```

- Копируем первую лабораторную работу и проверяем правильность выполнения команды

```
[aakayjnova@fedora report]$ cp ~/Загрузки/Л01_Кайнова_отчёт.pdf /home/aakayjnova/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
[aakayjnova@fedora report]$ ls
dir image Makefile pandoc report.md Л01_Кайнова_отчёт.pdf
```

- Добавляем созданные файлы на сервер

```
[aakayjnova@fedora report]$ git add Л01_Кайнова_отчёт.pdf
[aakayjnova@fedora report]$ |
```

- Сохраняем изменения на сервере

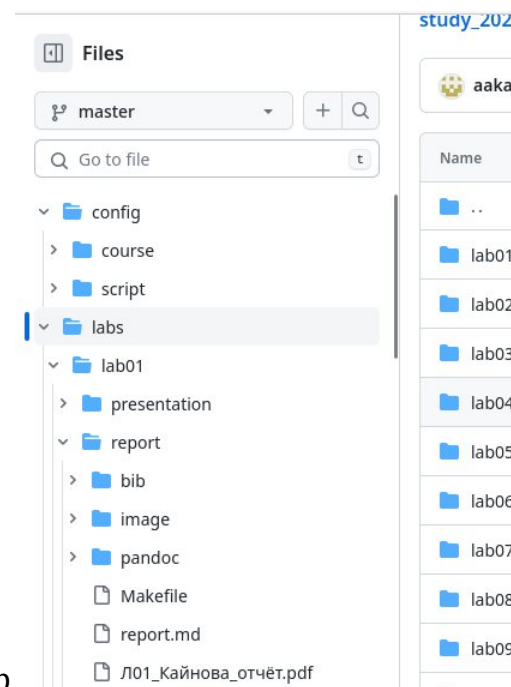
```
[aakayjnova@fedora report]$ git commit -m "Add existing file"
[master 94b224e] Add existing file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Кайнова_отчёт.pdf
```

3. Загрузка файлов на Github

- Отправляем в центральный репозиторий сохранённые изменения

```
[aakayjnova@fedora report]$ git push -f origin master
Перечисление объектов: 10, готово.
Подсчет объектов: 100% (10/10), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 690.38 КиБ | 4.86 МБ/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:aakayjnova/study_2023-2024_arh-pc.git
  4283168..94b224e master -> master
```

- Проверяем правильность выполнения работы на Github



5 Выводы

В ходе данной лабораторной работы мы изучили структуру систем контроля версий и научились применять средства контроля версий по работе с ситемой Git.

Список литературы

1. [Архитектура ЭВМ] (https://esystem.rudn.ru/pluginfile.php/2089082/mod_resource/content/0/)