

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2.

дисциплина: Архитектура компьютера

Студент: Кузьмин Е. В.

Группа:НКАбд-03-23

МОСКВА

2023 г.

Содержание

1. Цель работы	3
2. Задание	4
3. Теоретическое введение	5
4. Выполнение лабораторной работы.....	7
5. Выводы	18
6. Источники	19

1. Цель работы

Целью данной работы является приобретение практического опыта работы с системой git, изучение принципов и применения контроля версий.

2. Задание

0. Общее ознакомление с git.
1. Настройка GitHub.
2. Базовая настройка Git.
3. Создание SSH-ключа.
4. Создание рабочего пространства.
5. Создание репозитория курса на основе шаблона.
6. Настройка каталога курса.
7. Выполнение задания 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 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.

4. Выполнение лабораторной работы

1. Настройка GitHub

Создаю учетную запись на сайте GitHub, ввожу нужные данные учетной записи (Рис. 1)

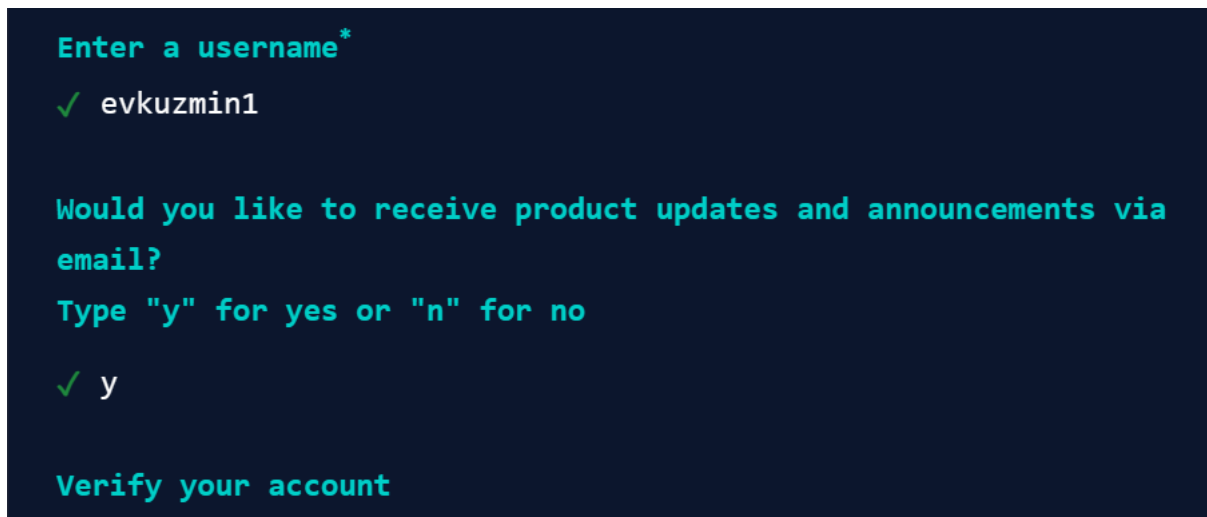


Рис. 1. Создание учетной записи GitHub

Можно персонализировать свой аккаунт (Рис. 2)

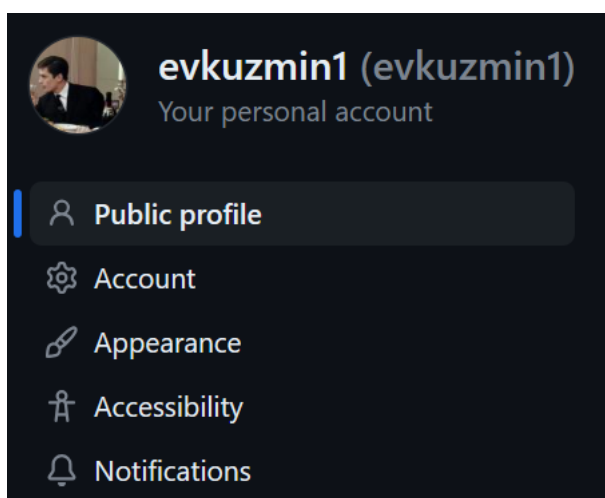
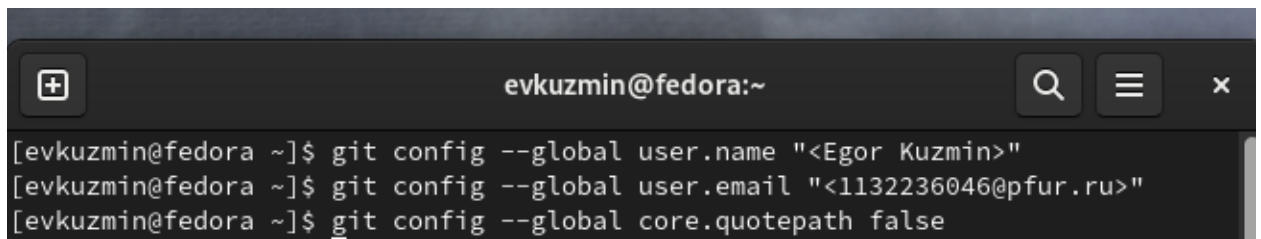


Рис. 2. Персонализация учетной записи GitHub

2. Базовая настройка Git

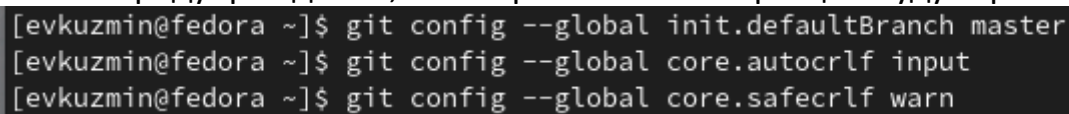
Открываю виртуальную машину, затем запускаю терминал, делаю предварительную конфигурацию git. Ввожу команду `git config --global user.name`, указывая свое имя и команду `git config --global user.email «work@mail»`, указывая в этой команде свою электронную почту, настраиваю utf-8 в выводе сообщений git для корректного отображения символов (Рис. 3)

A terminal window titled 'evkuzmin@fedora:~' with search, menu, and close icons. It shows three commands being executed: 'git config --global user.name "<Egor Kuzmin>"', 'git config --global user.email "<1132236046@pfur.ru>"', and 'git config --global core.quotepath false'.

```
[evkuzmin@fedora ~]$ git config --global user.name "<Egor Kuzmin>"
[evkuzmin@fedora ~]$ git config --global user.email "<1132236046@pfur.ru>"
[evkuzmin@fedora ~]$ git config --global core.quotepath false
```

Рис. 3. Предварительная конфигурация git

Задаю имя начальной ветке, а также параметры autocrlf и safecrlf, причем параметр autocrlf дополняем значением input, для конвертации символов разрыва строки в текстовых файлах (CRLF и LF) только при коммитах. Параметру safecrlf задаю значение warn, так Git будет проверять преобразование на обратимость, и при данном значении будет выведено только предупреждение, а необратимые конвертации будут приняты. (Рис. 4).

A terminal window titled 'evkuzmin@fedora ~' showing three commands: 'git config --global init.defaultBranch master', 'git config --global core.autocrlf input', and 'git config --global core.safecrlf warn'.

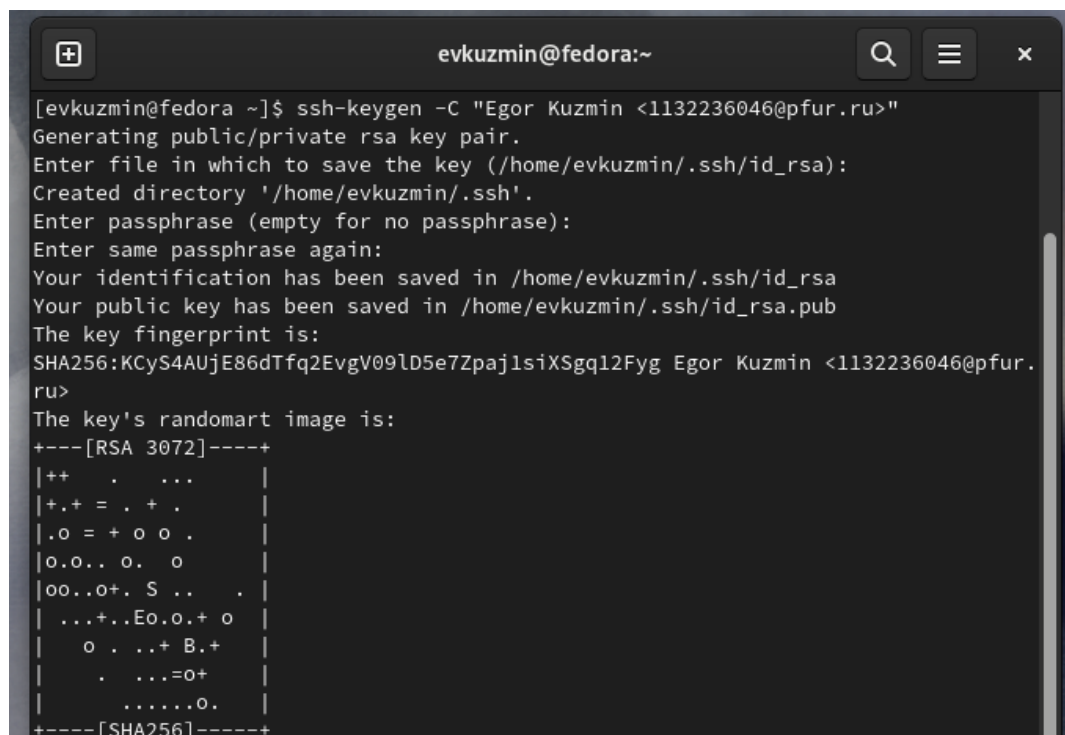
```
[evkuzmin@fedora ~]$ git config --global init.defaultBranch master
[evkuzmin@fedora ~]$ git config --global core.autocrlf input
[evkuzmin@fedora ~]$ git config --global core.safecrlf warn
```

Рис. 4. Создание имени для начальной ветки, присваивание параметров

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 своё имя и электронную почту. Ключ автоматически сохранится в каталоге `~/.ssh/`.

(Рис. 5)

A terminal window titled 'evkuzmin@fedora:~' showing the execution of 'ssh-keygen -C "Egor Kuzmin <1132236046@pfur.ru>"'. It displays the generation of an RSA key pair, prompts for a passphrase (which is left empty), and shows the key fingerprint and a randomart image.

```
[evkuzmin@fedora ~]$ ssh-keygen -C "Egor Kuzmin <1132236046@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/evkuzmin/.ssh/id_rsa):
Created directory '/home/evkuzmin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/evkuzmin/.ssh/id_rsa
Your public key has been saved in /home/evkuzmin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:KCYS4AUjE86dTfq2EvvgV09lD5e7ZpajlsiXSgq12Fyg Egor Kuzmin <1132236046@pfur.ru>
The key's randomart image is:
+---[RSA 3072]-----+
|++ . ...|
|+.+ = . + .|
|.o = + o o .|
|o.o.. o. o|
|oo..o+. S .. .|
|...+..Eo.o.+ o|
| o . ..+ B.+|
| . ...=o+|
| .....o.|
+---[SHA256]-----+
```

Рис. 5. Генерация SSH-ключа

Устанавливаю утилиту xclip, позволяющую копировать любой текст через терминал. Использую команду «dnf install» с ключом -y от имени суперпользователя, ввожу в начале «sudo» (Рис. 6)

```
[evkuzmin@fedora ~]$ sudo dnf install -y xclip
Последняя проверка окончания срока действия метаданных: 0:17:14 назад, Пн 25 сен 2023 22:12:16.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
xclip      x86_64       0.13-19.git11cba61.fc38  fedora       37 k

Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 37 k
Объем изменений: 63 k
Загрузка пакетов:
xclip-0.13-19.git11cba61.fc38.x86_64.rpm 322 kB/s | 37 kB 00:00
-----
Общий размер 43 kB/s | 37 kB 00:00
Fedora 38 - x86_64 1.6 MB/s | 1.6 kB 00:00
Импорт GPG-ключа 0xEB10B464:
Идентификатор пользователя: "Fedora (38) <fedora-38-primary@fedoraproject.org>"
Отпечаток: 6A51 BBAB BA3D 5467 B617 1221 809A 8D7C EB10 B464
Источник: /etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-38-x86_64
Импорт ключа успешно завершен
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка : 1/1
Установка : xclip-0.13-19.git11cba61.fc38.x86_64 1/1
Запуск скрипглета: xclip-0.13-19.git11cba61.fc38.x86_64 1/1
Проверка : xclip-0.13-19.git11cba61.fc38.x86_64 1/1
Установлен:
```

Рис. 6. Установка утилиты xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip (Рис. 7)

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

Рис. 7. Копирование ключа

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key». Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (Рис. 8)

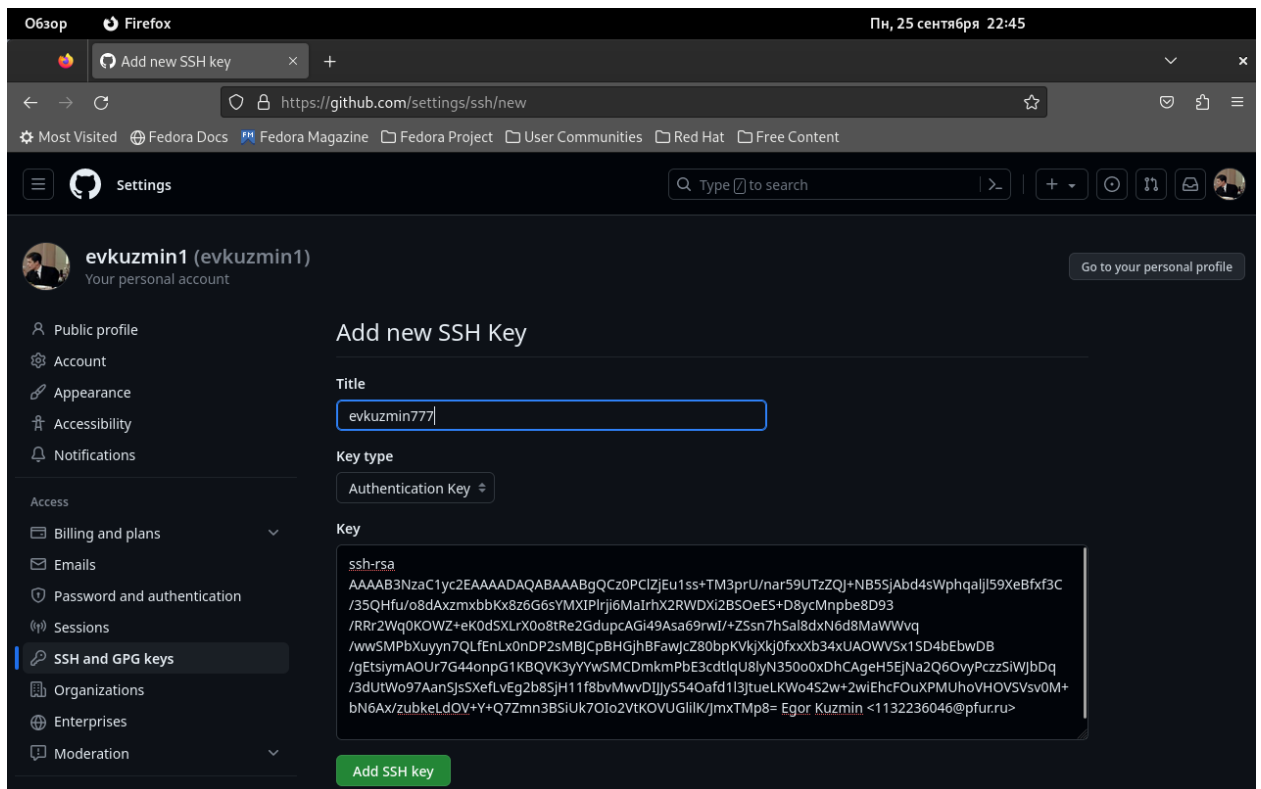


Рис. 8. Добавление ключа

4. Создание рабочего пространства

Закрываю браузер, открываю терминал. Создаю директорию, рабочее пространство с помощью утилиты `mkdir`. С помощью ключа `-p` создаю рекурсивно все директории после домашней `~/work/study/2023-2024/«Архитектура Компьютера»`. Далее проверяю с помощью `ls`, действительно ли были созданы необходимые мне каталоги (Рис. 9)

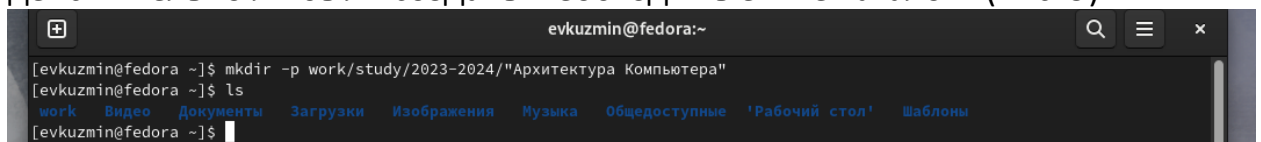


Рис. 9. Создание рабочего пространства

5. Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитория с шаблоном курса по адресу: <https://github.com/yamadharma/course-directory-student-template>. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория (Рис. 10)

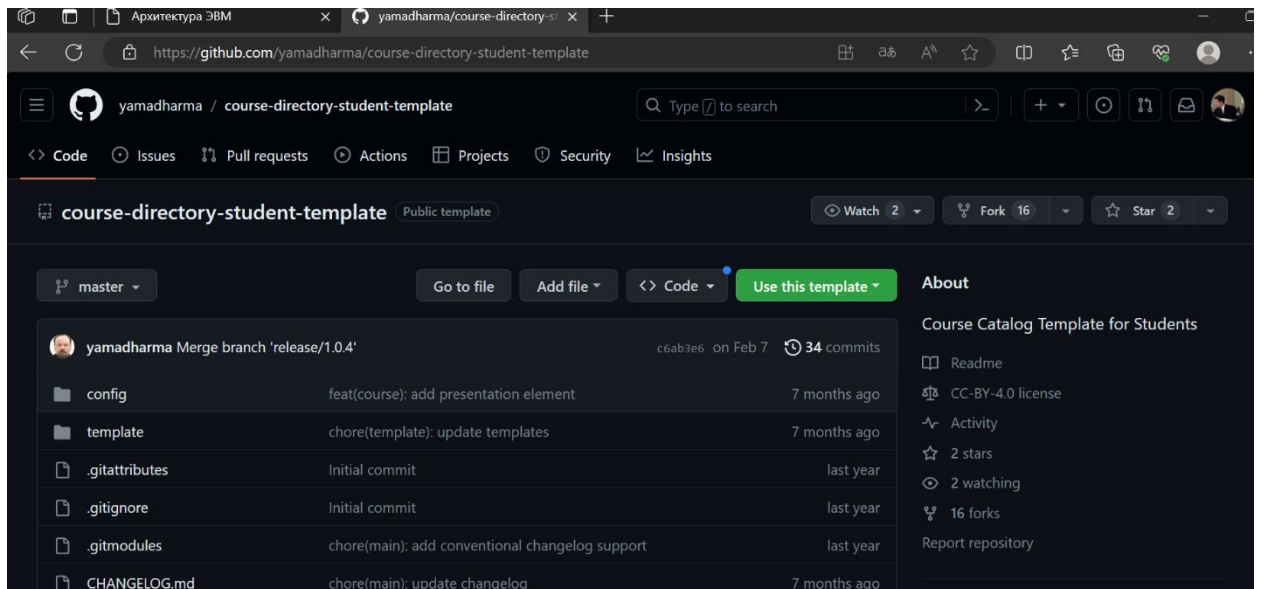


Рис. 10. Страница репозитория с шаблоном

В открывшемся окне задаю имя репозитория: `study_2023–2024_arh-pc`. Создаю репозиторий, нажимаю на кнопку «Create repository from template» (Рис. 11)

Рис. 11. Страница создания репозитория

Ждем создания репозитория, затем он откроется (Рис. 12)



Рис. 12. Созданный репозиторий

Далее через терминал перехожу в созданный каталог курса с помощью утилиты `cd`. Клонировать созданный репозиторий с помощью команды:

«`git clone --recursive git@github.com:/study_2023–2024_arh-pc.git arch-pc`» (Рис. 13)

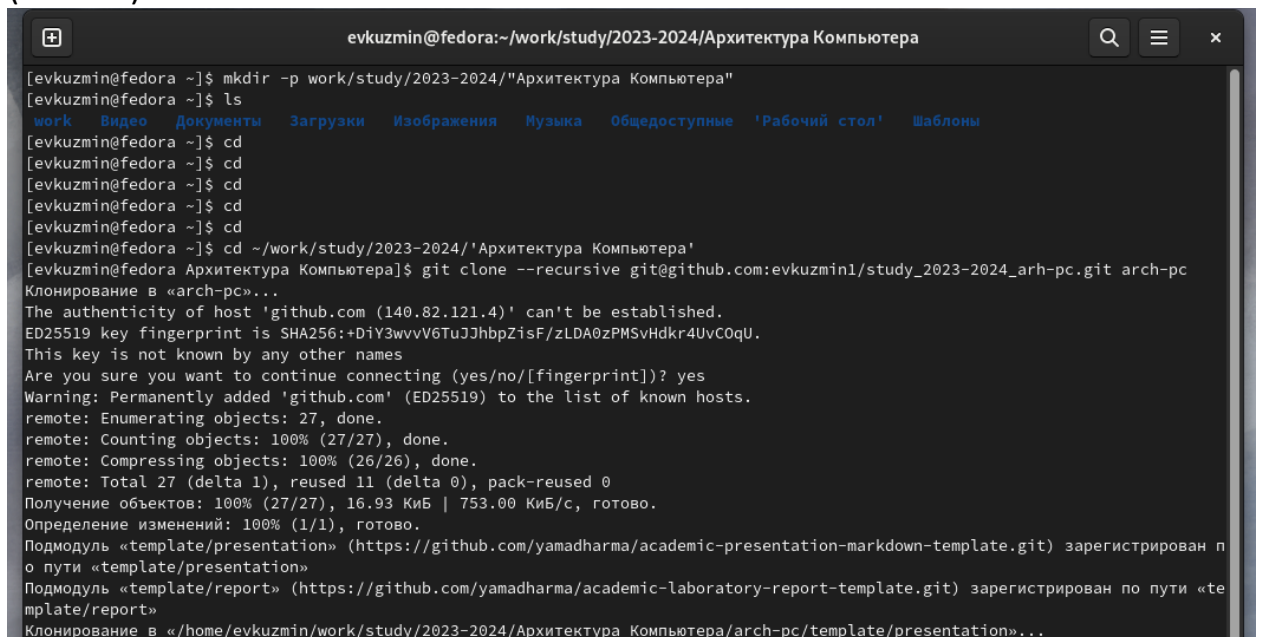


Рис. 13. Клонирование репозитория

Ссылку для копирования можно взять на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (Рис. 14)

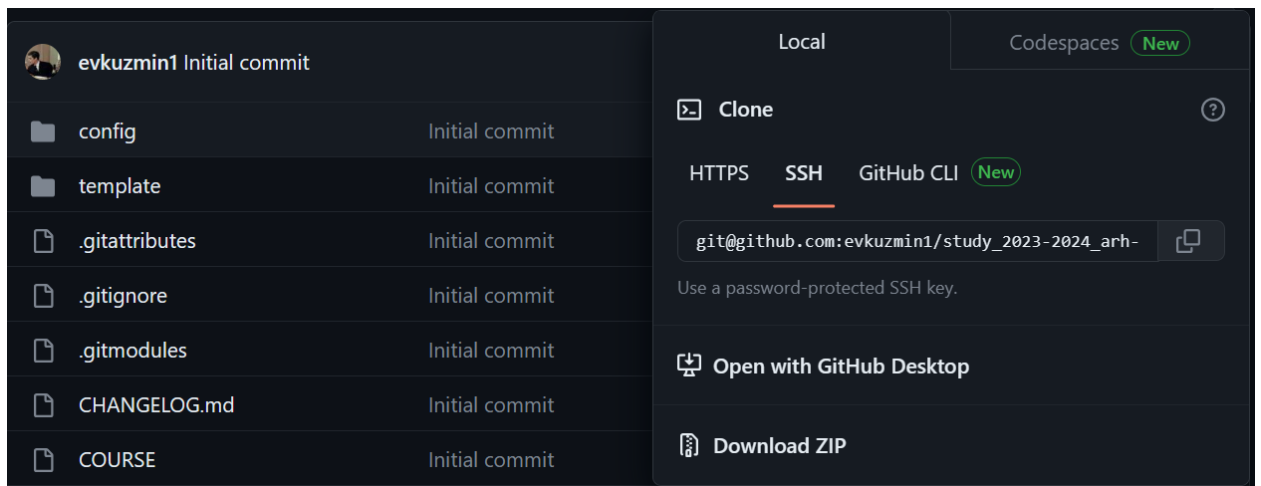


Рис. 14. Окно с ссылкой на копирование репозитория

6. Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты `cd`. Затем удаляю лишние файлы с помощью утилиты `rm` и создаю необходимые каталоги (Рис. 15)

```
[evkuzmin@fedora Архитектура Компьютера]$ cd ~/work/study/2023-2024/Архитектура\ Компьютера/arch-pc
[evkuzmin@fedora arch-pc]$ rm package.json
[evkuzmin@fedora arch-pc]$ echo arch-pc > COURSE
[evkuzmin@fedora arch-pc]$ make
```

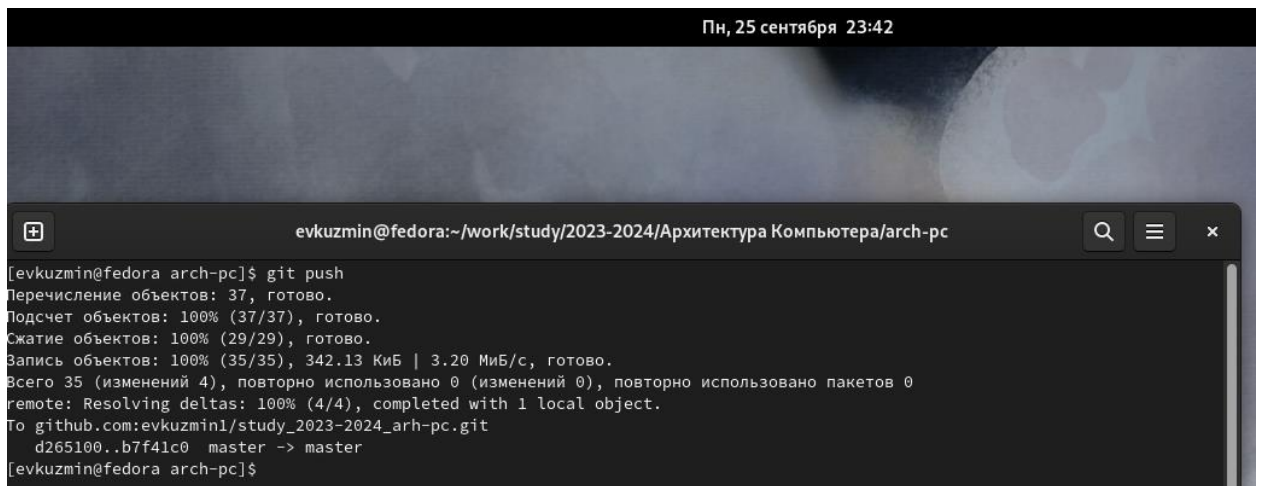
Рис. 15. Перемещение между директориями, удаление файлов, создание каталогов

Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью «`git add`», комментирую и сохраняю изменения на сервере с помощью «`git commit`» (Рис. 16)

```
[evkuzmin@fedora arch-pc]$ git add .
[evkuzmin@fedora arch-pc]$ git commit -am 'feat(main): make course structure'
[master b7f41c0] 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
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/core.py
```

Рис. 16. Добавление и сохранение изменений на сервере

Отправляю все на сервер с помощью «`push`» (Рис. 17)



```
Пн, 25 сентября 23:42

evkuzmin@fedora:~/work/study/2023-2024/Архитектура Компьютера/arch-pc

[evkuzmin@fedora arch-pc]$ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 КиБ | 3.20 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:evkuzmin1/study_2023-2024_arh-pc.git
   d265100..b7f41c0  master -> master
[evkuzmin@fedora arch-pc]$
```

Рис. 17. Выгрузка изменений на сервер

Проверяю правильность выполнения работы на самом сайте GitHub (Рис.18)

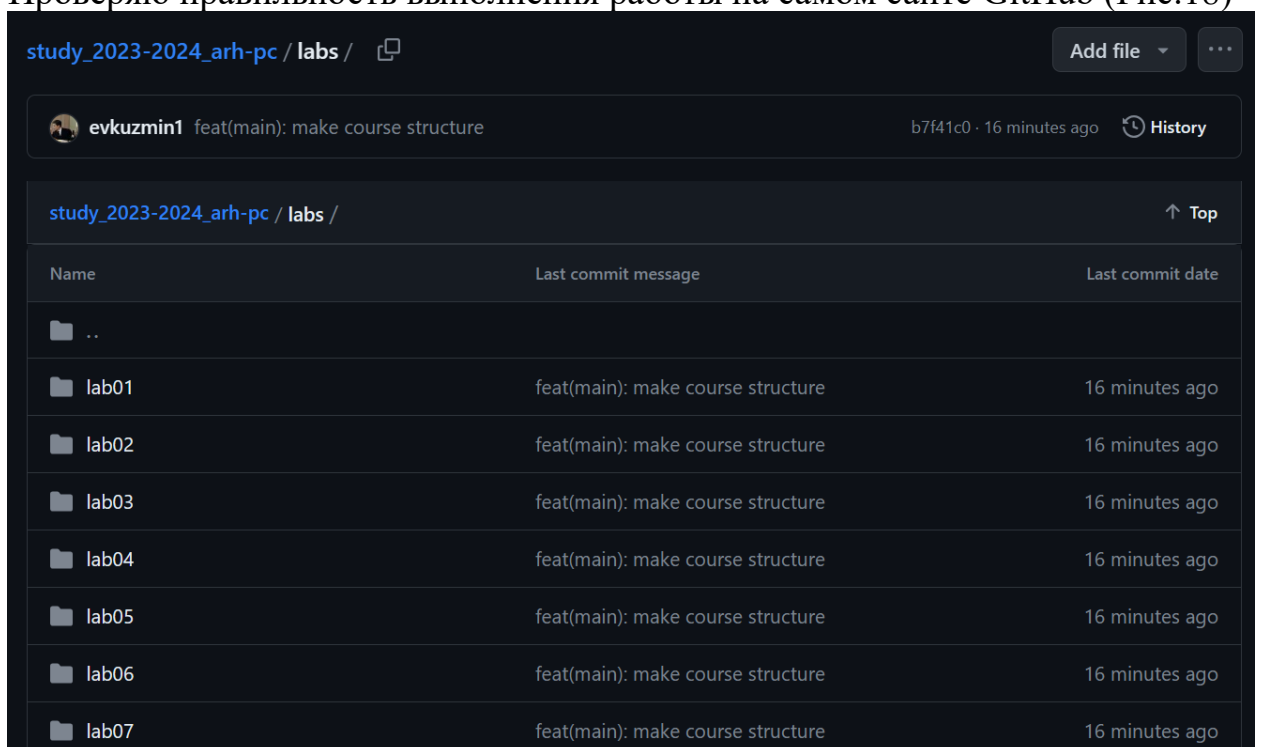
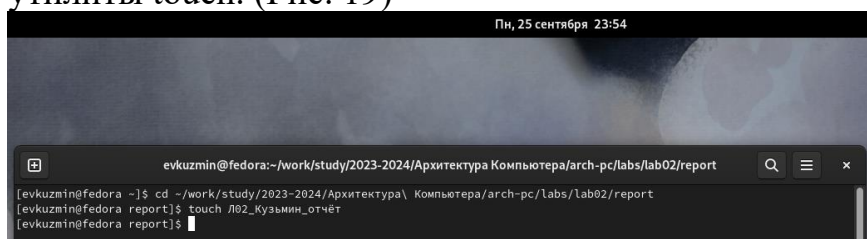


Рис. 18. Проверка результата на странице репозитория

7. Настройка каталога курса

Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по третьей лабораторной работе с помощью утилиты touch. (Рис. 19)



```
Пн, 25 сентября 23:54

evkuzmin@fedora:~/work/study/2023-2024/Архитектура Компьютера/arch-pc/labs/lab02/report

[evkuzmin@fedora ~]$ cd ~/work/study/2023-2024/Архитектура\ Компьютера/arch-pc/labs/lab02/report
[evkuzmin@fedora report]$ touch lab02_Кузьмин_отчет
[evkuzmin@fedora report]$
```


Рис. 19. Перемещение по директориям, создание файла

1) Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений. После открытия текстового процессора открываю в нем созданный файл. Теперь можно создать в нем отчет (Рис. 20)

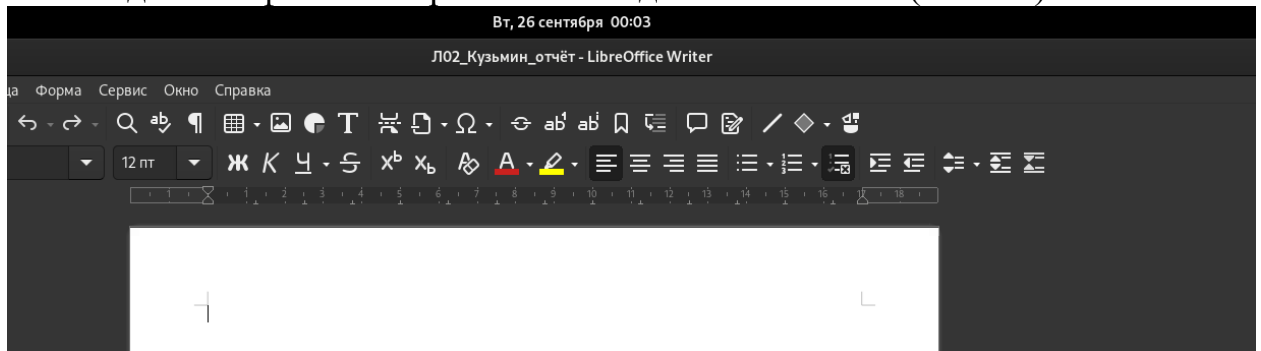


Рис. 20. Работа с отчётом в текстовом редакторе

2) Перехожу из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd. Проверяю местонахождение файла с отчетом по первой лабораторной работе, используя команду ls (Рис. 21)

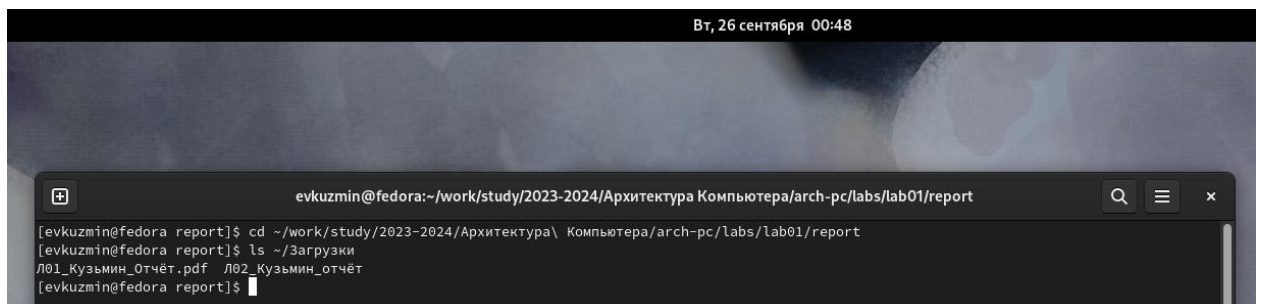


Рис. 21. Перемещение между директориями, проверка местонахождения файла

Перехожу из подкаталога lab01/report в подкаталог lab02/report с помощью утилиты cd. Копирую вторую лабораторную с помощью утилиты cp. (Рис. 22)

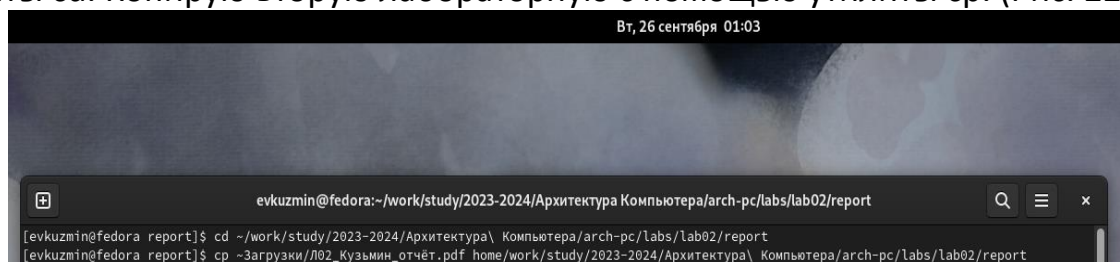


Рис. 22. Перемещение между директориями, копирование файла

3) Добавляю с помощью команды git add в коммит созданные файлы: Л01_Кузьмин_отчет. Перехожу в директорию, в которой находится отчет по второй лабораторной. Добавляю файл Л01_Кузьмин_отчет (Рис. 23, 24)

```
[evkuzmin@fedora report]$ git add Л01_Кузьмин_отчёт.pdf
```

```
evkuzmin@fedora report]$ git add Л02_Кузьмин_отчёт
evkuzmin@fedora report]$ git commit -m "Add existing file"
master 48772ad] Add existing file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab02/report/Л02_Кузьмин_отчёт
```

Рис. 23, 24. Добавление файла на сервер

Отправляю в центральный репозиторий сохраненные изменения командой `git push -f origin master` (Рис. 25)

```
[evkuzmin@fedora report]$ git push -f origin master
Перечисление объектов: 67, готово.
Подсчет объектов: 100% (67/67), готово.
Сжатие объектов: 100% (59/59), готово.
Запись объектов: 100% (67/67), 359.39 КиБ | 2.22 МиБ/с, готово.
Всего 67 (изменений 8), повторно использовано 26 (изменений 1), повторно использовано пакетов 0
remote: Resolving deltas: 100% (8/8), done.
To github.com:evkuzmin1/study_2023-2024_arh-pc.git
 + efd6bfb...48772ad master -> master (forced update)
[evkuzmin@fedora report]$
```

Рис. 25. Отправка в центральный репозиторий сохраненных изменений

Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что пояснение к совершенным действиям отображается (Рис. 26)


<div>  evkuzmin1 </div> Add existing file <div>48772ad · 27 minutes ago</div> <div>History</div>		
study_2023-2024_arh-pc / labs / <div>↑ Top</div>		
Name	Last commit message	Last commit date
..		
lab01	feat(main): make course structure	2 hours ago
lab02	Add existing file	27 minutes ago

Рис. 26. Отправка в центральный репозиторий сохраненных изменений

При просмотре изменений так же вижу, что были добавлены файлы с отчетами по лабораторным работам (Рис. 27)



Рис. 27. Страница последних изменений в репозитории

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория: отчет по первой - в `lab01/report`, по второй – в `lab02/report` (Рис. 28, 29)

study_2023-2024_arh-pc / labs / lab01 / report /

evkuzmin1

Add files via upload

Name
..
bib
image
pandoc
Makefile
report.md
ЛО1_Кузьмин_Отчёт.pdf

study_2023-2024_arh-pc / labs / lab02 / report /

evkuzmin1

Add existing file

48772ad · 42 minutes ago

History

Name	Last commit message	Last commit date
..		
bib	feat(main): make course structure	2 hours ago
image	feat(main): make course structure	2 hours ago
pandoc	feat(main): make course structure	2 hours ago
Makefile	feat(main): make course structure	2 hours ago
report.md	feat(main): make course structure	2 hours ago
ЛО2_Кузьмин_отчёт	Add existing file	42 minutes ago

Рис. 28, 29. Проверка каталогов

5. Выводы

Я приобрел практический опыт работы с системой git, изучил принципы и применение контроля версий.

6. Список литературы

1. [Архитектура ЭВМ \(rudn.ru\)](#)
2. [Инструкция по использованию Git](#)