**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: Ермаков А.В.

Группа: НФПбд-01-21

№ ст. билета: 1032213202

**МОСКВА**

2023 г.

Содержание

[1 Цель работы 2](#_Toc146462559)

[2 Задание 2](#_Toc146462560)

[3 Теоретическое введение 2](#_Toc146462561)

[4 Выполнение лабораторной работы 3](#_Toc146462562)

[5 Выводы 7](#_Toc146462563)

[Список литературы 8](#_Toc146462564)

# 1 Цель работы

Целью работы является изучить применение средств контроля версий - git. Приобрести практические навыки по работе с системой контроля версий git.

# 2 Задание

1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).
2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.
3. Загрузите файлы на github

# 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 табл. [1](#tbl:std-dir) Основные команды git.

Table 1: Описание некоторых команд git

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации |
| git init | создание основного дерева репозитория |
| git pull | получение обновлений (изменений) текущего дерева из центрального репозитория |
| git push | отправка всех произведённых изменений локального дерева в центральный репозиторий |
| git status | просмотр списка изменённых файлов в текущей директории |
| git diff | просмотр текущих изменения |

# 4 Выполнение лабораторной работы

Предварительная конфигурация git проводилась командами:

git config --global user.name

git config --global user.email

Здесь указываются имя и email владельца репозитория

Настройка utf-8 в выводе сообщений git:

git config --global core.quotepath false

Задаётся имя начальной ветки (master):

git config --global init.defaultBranch master

Параметр autocrlf:

git config --global core.autocrlf input

Параметр safecrlf:

git config --global core.safecrlf warn

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. @fig:001).(рис. 4.1).

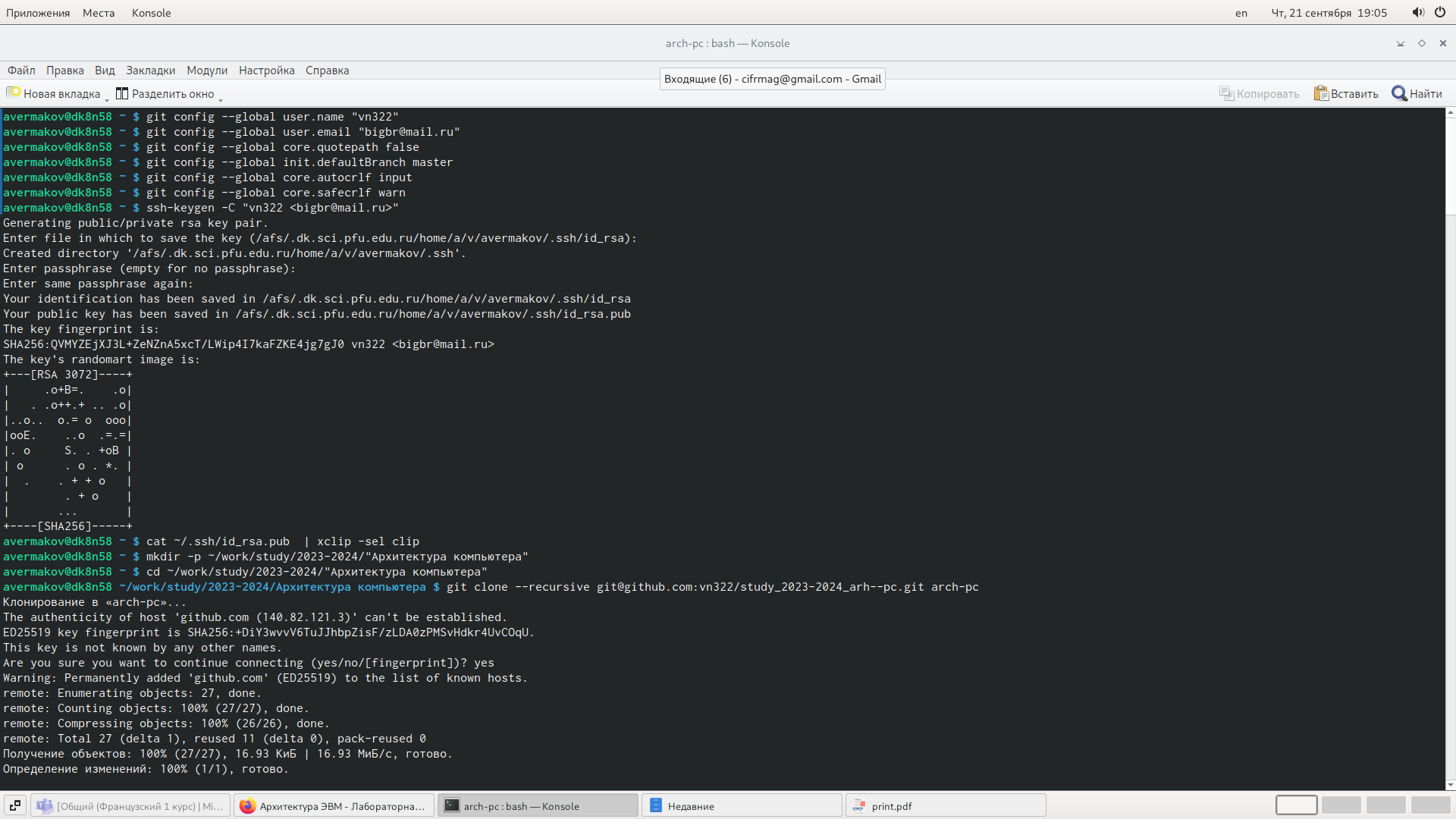


Рис. 4.1. Предварительная конфигурация git

Для последующей идентификации пользователя на сервере репозиториев генерируем пару ключей (приватный и открытый) (рис. @fig:002).(рис. 4.2).

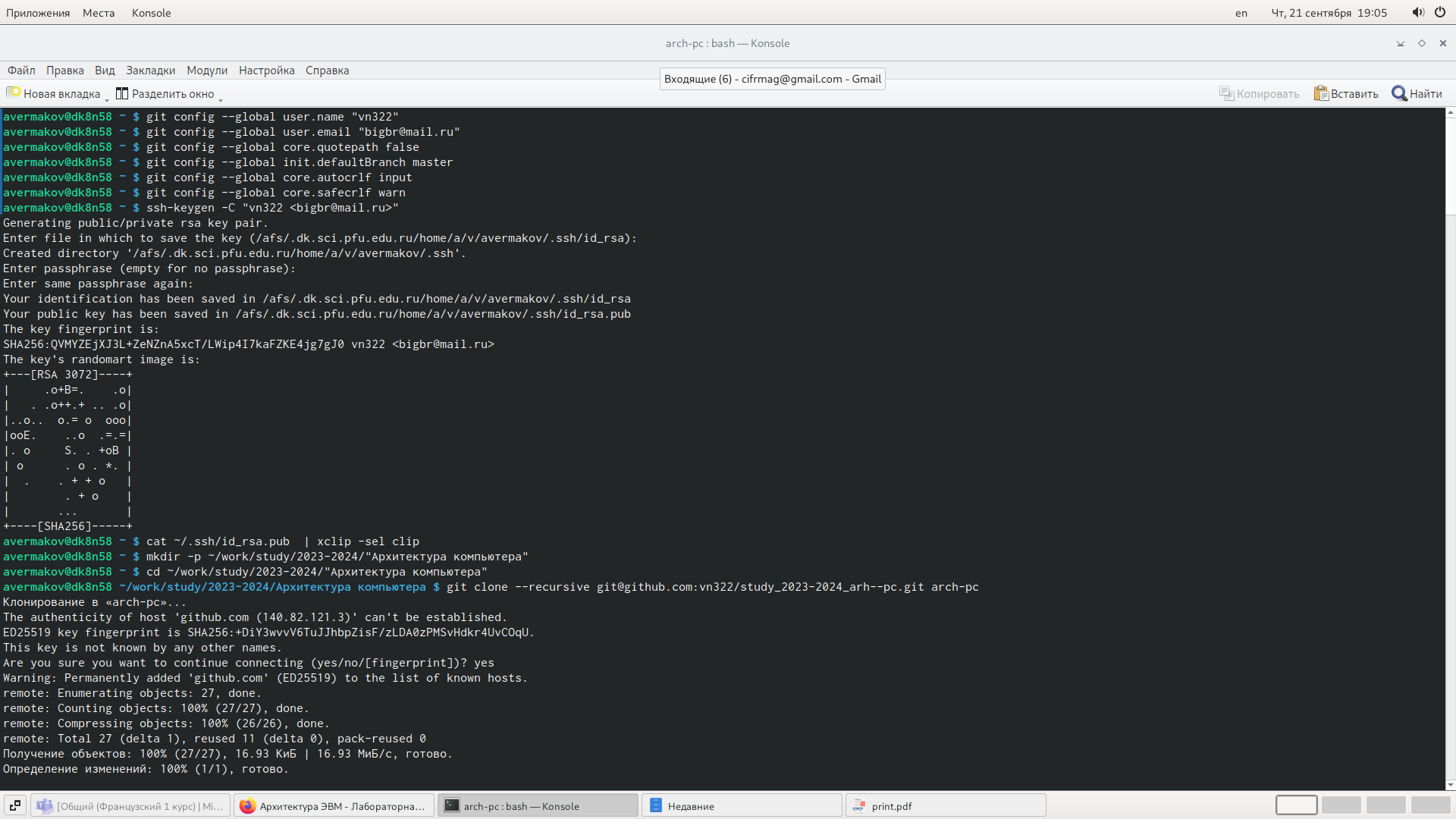


Рис. 4.2. Создание SSH ключаКлючи сохраняться в каталоге ~/.ssh/. Далее загружаем сгенерённый открытый ключ. Для этого на сайте http: //github.org/ под своей учётной записью и в меню Setting выбрать в боковом меню SSH and GPG keys и нажать кнопку New SSH key . Скопировав из локальной консоли ключ в буфер обмена

cat ~/.ssh/id\_rsa.pub | xclip -sel clipСоздаём каталог для предмета «Архитектура компьютера» (рис.4.3.)

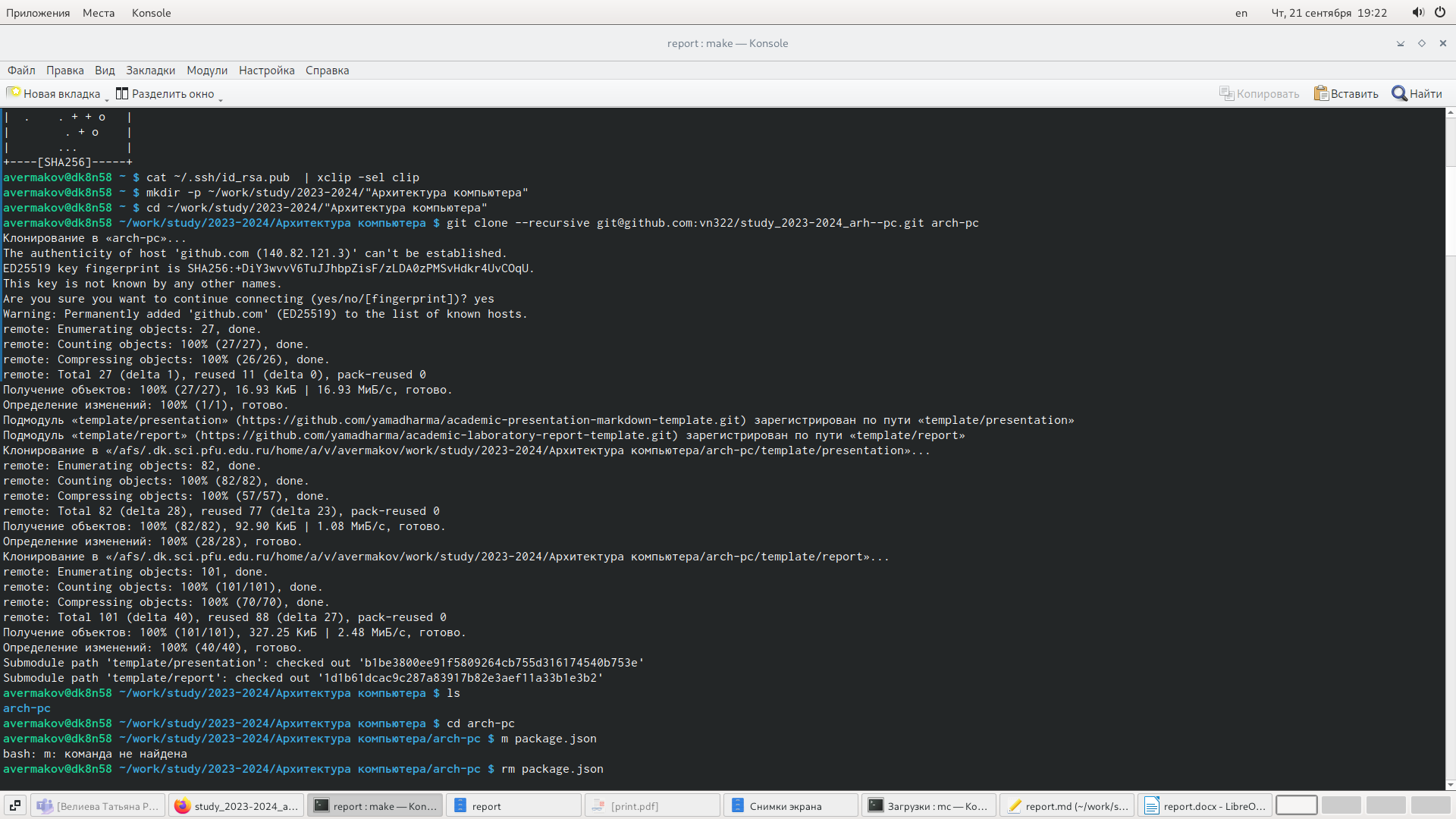


Рис. 4.3. Создание каталог для предмета «Архитектура компьютера»

В каталоге курса:

cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc

Удаляем лишние файлы:

rm package.json

И создаём необходимые каталоги:

echo arch-pc > COURSE

make

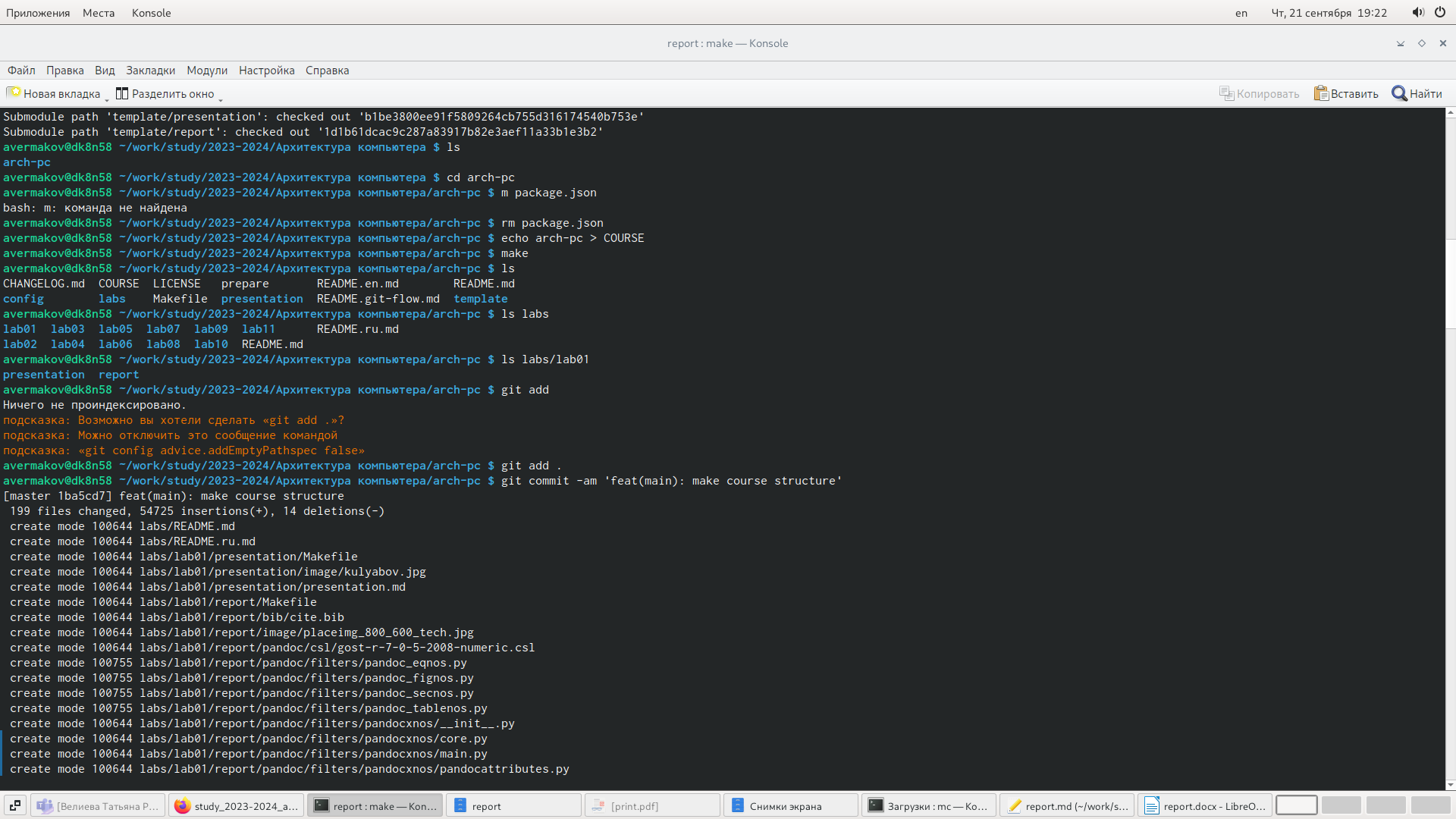
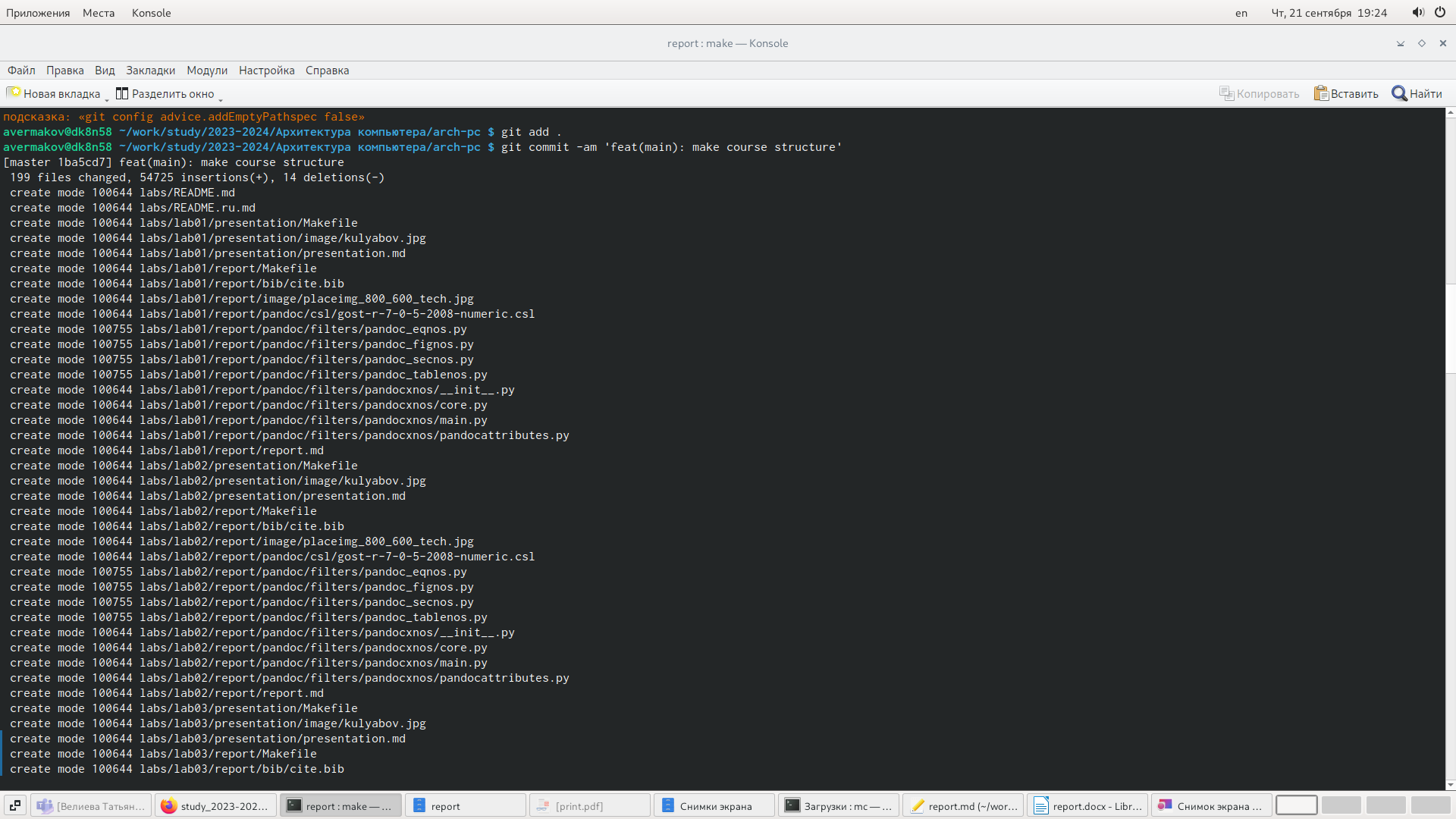


Рис. 4.4. Создание каталог для предмета «Архитектура компьютера»

Добавляем файлы на сервер (коммит):

Командой git add - добавляем все изменённые, созданные файлы или каталоги на сервер

git commit -am 'feat(main): make course structure' сохраняем все добавленные изменения и все изменённые файлы с названием коммита

Рис. 4.5. Добавляем файлы на сервер (коммит)

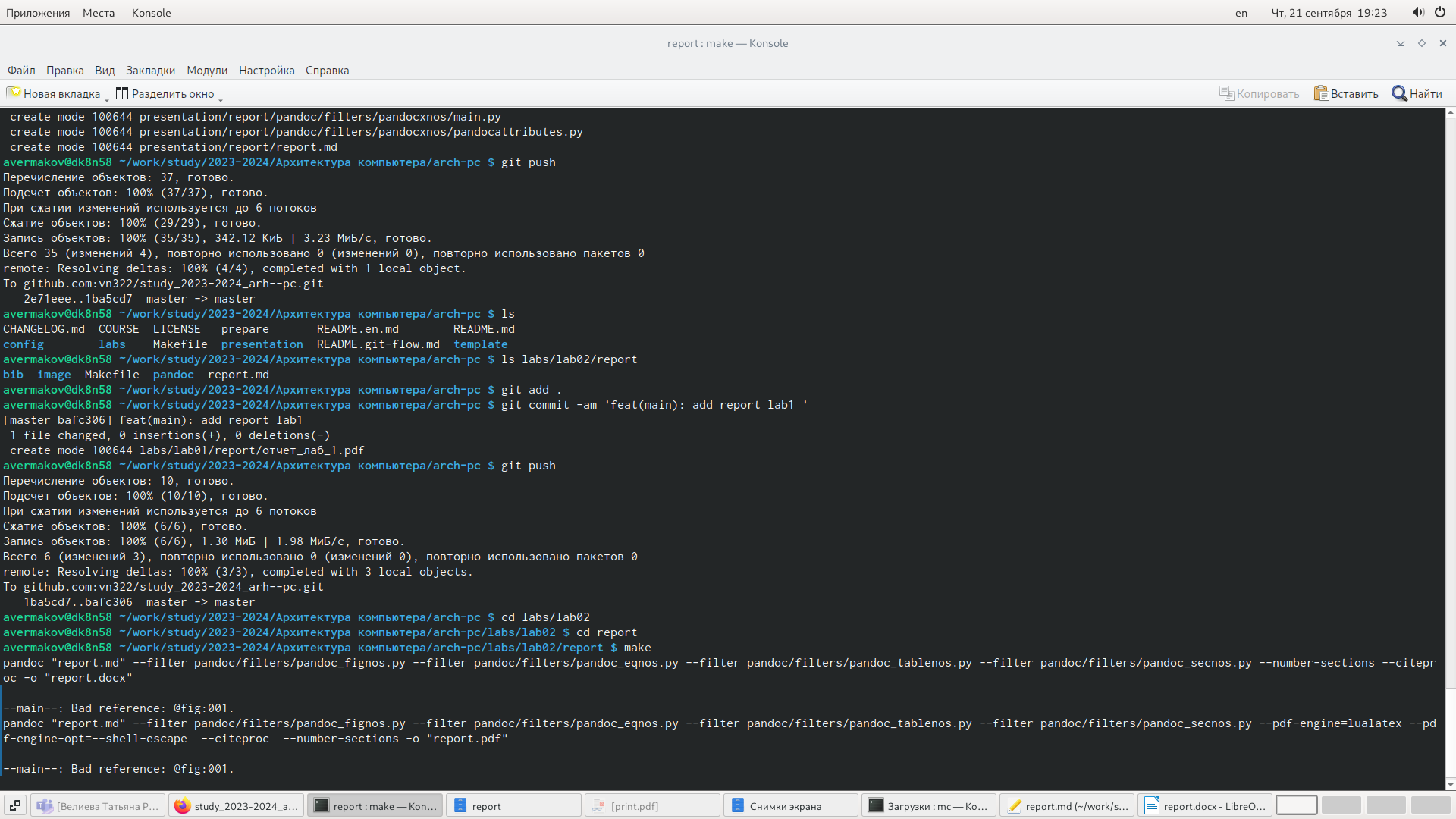
Командой git push производим отправку всех произведённых изменений локального дерева в центральный репозиторий и проверяем результат (рис. 4.6.)

Рис. 4.6. Отправка файлов на сервер

Проверили правильность создания иерархии рабочего пространства в локальном репозитории и на странице github

# 5 Выводы

В данной работе познакомились с системой контроля версий git. Создали клон учебного репозитория создали и отредактировали файлы с последующей отправкой в репозиторий и проверкой их наличия.

Визуальный интерфейс выглядит привлекательнее, но может и в командной строке есть свои плюсы.

# Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.

2. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Robbins A. [Bash Pocket Reference](https://www.ncbi.nlm.nih.gov/pubmed/25246403). O’Reilly Media, 2016. 156 с.

5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.

6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.