

Лабораторная работа №2

дисциплина: Архитектура компьютера

Ицков Андрей Станиславович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Техническое обеспечение	9
4.2	Базовая настройка Git	9
4.3	Создание рабочего пространства и репозитория курса на основе шаблона	11
4.4	Создание репозитория курса на основе шаблона	12
4.5	Настройка каталога курса	13
4.6	Задания для самостоятельной работы	15
5	Выводы	16
	Список литературы	17

Список иллюстраций

4.1	Предварительная конфигурация Git	9
4.2	Настройка параметров Git	10
4.3	Создание пары ssh ключей	10
4.4	Установка пары ключей	11
4.5	Создание рабочей директории	11
4.6	Процесс клонирования репозитория через интерфейс GitHub . . .	12
4.7	Копирование репозитория на рабочий компьютер	13
4.8	Настройка каталога курса	14
4.9	Отправка изменений на удаленный репозиторий	14
4.10	Копирование прошлого отчета	15
4.11	Загрузка файлов на github	15

Список таблиц

1 Цель работы

Целью работы является изучить идеологию и применение системы контроля версий. Приобрести практические навыки по работе с системой git.

2 Задание

На основе методических указаний провести работу с базовыми командами системы контроля версий git, выучить применение команд для разных случаев использования, настроить 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 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.

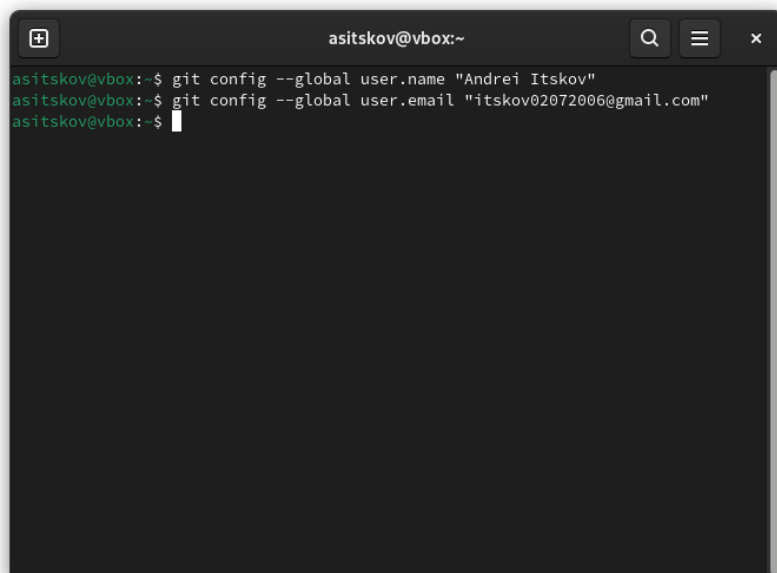
4 Выполнение лабораторной работы

4.1 Техническое обеспечение

Лабораторная работа была выполнена на домашнем компьютере под управлением операционной системы Fedora Workstation 40.

4.2 Базовая настройка Git

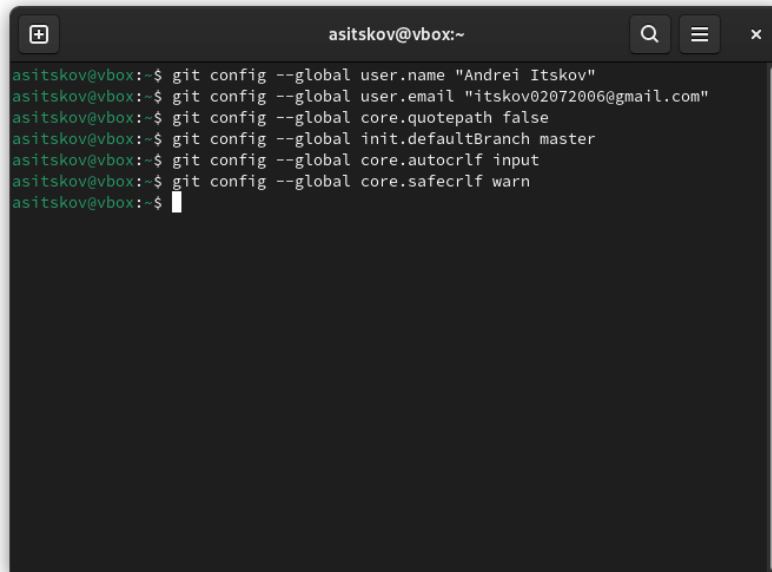
Для начала я проведу предварительную конфигурацию Git, для этого открываю терминал и ввожу команды на (рис. 4.1).

A screenshot of a terminal window titled 'asitskov@vbox:~'. The terminal shows three lines of text: the first line is 'asitskov@vbox:~\$ git config --global user.name "Andrei Itskov"', the second line is 'asitskov@vbox:~\$ git config --global user.email "itskov02072006@gmail.com"', and the third line is 'asitskov@vbox:~\$' followed by a cursor. The terminal has a dark background and a light-colored text. The window has standard Linux window controls (minimize, maximize, close) in the top right corner.

```
asitskov@vbox:~$ git config --global user.name "Andrei Itskov"
asitskov@vbox:~$ git config --global user.email "itskov02072006@gmail.com"
asitskov@vbox:~$
```

Рис. 4.1: Предварительная конфигурация Git

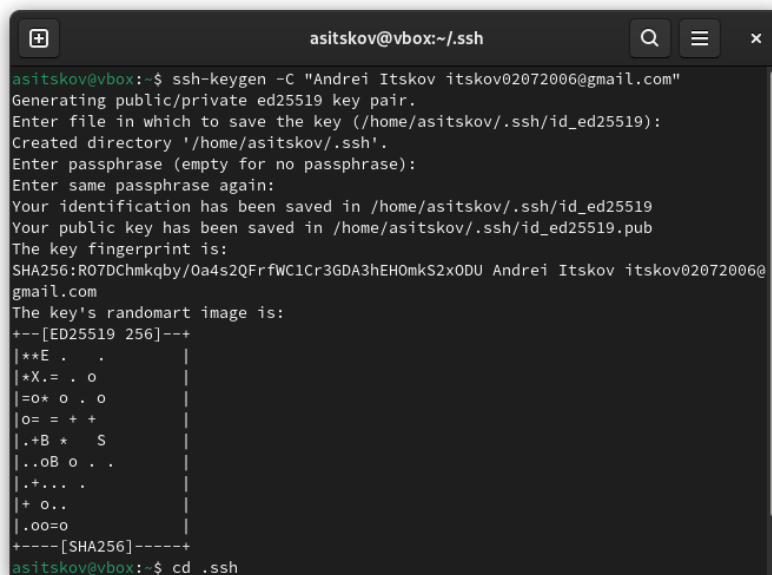
Далее настраиваю параметры utf-8, имя начальной ветки, autocrlf и safecrlf.
(рис. 4.2)

A terminal window titled 'asitskov@vbox:~' with search, menu, and close icons. It shows a series of 'git config' commands being executed to set global settings for user, email, core.quotePath, init.defaultBranch, core.autocrlf, and core.safecrlf.

```
asitskov@vbox:~$ git config --global user.name "Andrei Itskov"
asitskov@vbox:~$ git config --global user.email "itskov02072006@gmail.com"
asitskov@vbox:~$ git config --global core.quotePath false
asitskov@vbox:~$ git config --global init.defaultBranch master
asitskov@vbox:~$ git config --global core.autocrlf input
asitskov@vbox:~$ git config --global core.safecrlf warn
asitskov@vbox:~$
```

Рис. 4.2: Настройка параметров Git

Далее создаю пару ssh ключей для интеграции с платформой GitHub. (рис. 4.3)

A terminal window titled 'asitskov@vbox:~/.ssh' with search, menu, and close icons. It shows the execution of 'ssh-keygen' to create an ed25519 key pair, including prompts for filename, passphrase, and confirmation, followed by the display of the key fingerprint and a randomart image.

```
asitskov@vbox:~$ ssh-keygen -C "Andrei Itskov itskov02072006@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/asitskov/.ssh/id_ed25519):
Created directory '/home/asitskov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/asitskov/.ssh/id_ed25519
Your public key has been saved in /home/asitskov/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:R07DChmkqby/0a4s2QFrFWC1Cr3GDA3hEH0mKS2xODU Andrei Itskov itskov02072006@
gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|**E . . |
|*X.= . o |
|=o* o . o |
|o= = + + |
|.+B * S |
|..OB o . . |
|.+. . . |
|+ o.. |
|.oo=o |
+----[SHA256]-----+
asitskov@vbox:~$ cd ~/.ssh
```

Рис. 4.3: Создание пары ssh ключей

Далее я перехожу на сайт GitHub, авторизуюсь, перехожу в настройки аккаунта, вставляю публичный ключ в предназначенном для этого поле. (рис. 4.4)

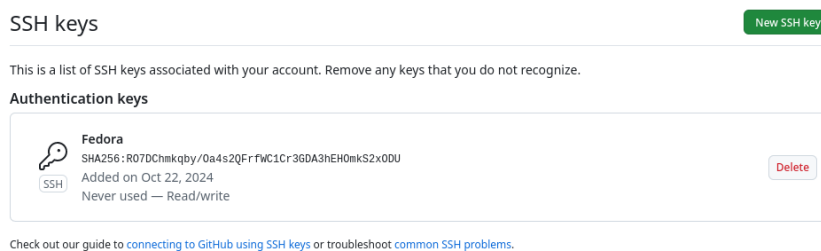


Рис. 4.4: Установка пары ключей

4.3 Создание рабочего пространства и репозитория курса на основе шаблона

Рабочее пространство при выполнении лабораторных работ должно придерживаться определённой структурной иерархии, для этого я создаю директорию на своем рабочем компьютере. (рис. 4.5)

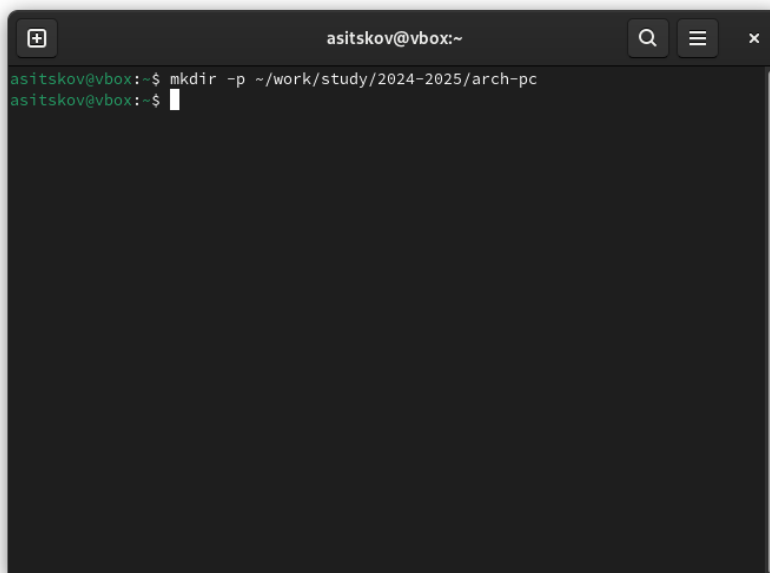


Рис. 4.5: Создание рабочей директории

4.4 Создание репозитория курса на основе шаблона

Создаю репозиторий на основе имеющего шаблона через функционал клонирования интерфейса GitHub. (рис. 4.6)

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

 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 *

 a-its ▾

Repository name *

/ study_2024-2025_arh-pc

✓ Your new repository will be created as study_2024-2025_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 [expert-umbrella](#) ?

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.

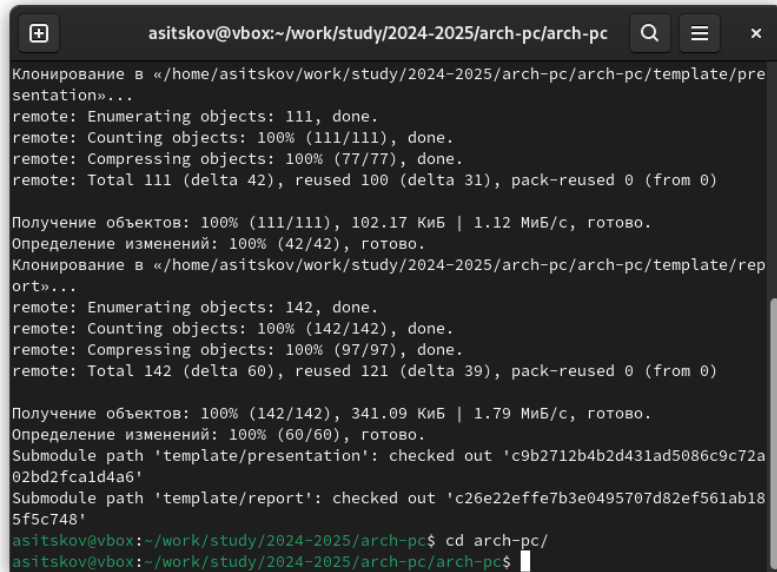


You are creating a public repository in your personal account.

Create repository

Рис. 4.6: Процесс клонирования репозитория через интерфейс GitHub

Сгенерированный репозиторий на основе шаблона клонирую на свой рабочий компьютер, для этого беру ссылку для клонирования через интерфейс GitHub и затем ввожу в терминале `git clone`. (рис. 4.7)



```
asitskov@vbox:~/work/study/2024-2025/arch-pc/arch-pc
Клонирование в «/home/asitskov/work/study/2024-2025/arch-pc/arch-pc/template/pre
sentation»...
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)

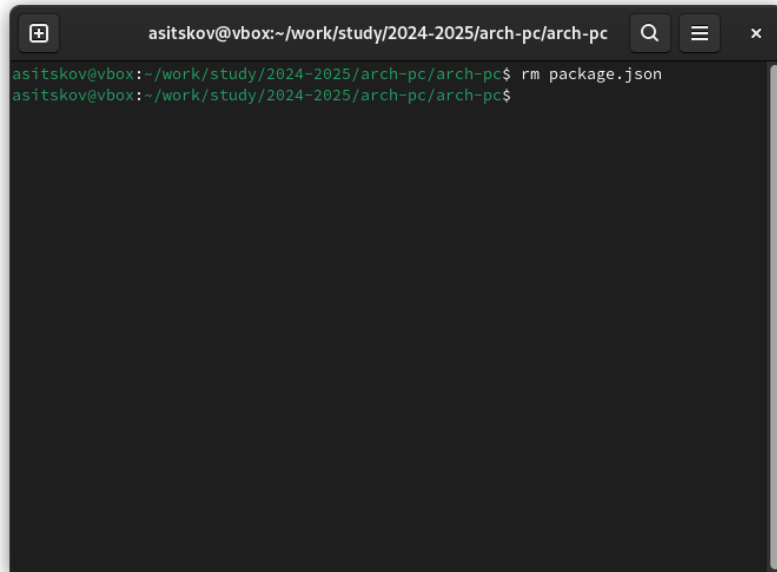
Получение объектов: 100% (111/111), 102.17 КиБ | 1.12 МиБ/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/home/asitskov/work/study/2024-2025/arch-pc/arch-pc/template/rep
ort»...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 142 (delta 60), reused 121 (delta 39), pack-reused 0 (from 0)

Получение объектов: 100% (142/142), 341.09 КиБ | 1.79 МиБ/с, готово.
Определение изменений: 100% (60/60), готово.
Submodule path 'template/presentation': checked out 'c9b2712b4b2d431ad5086c9c72a
02bd2fcald4a6'
Submodule path 'template/report': checked out 'c26e22effe7b3e0495707d82ef561ab18
5f5c748'
asitskov@vbox:~/work/study/2024-2025/arch-pc$ cd arch-pc/
asitskov@vbox:~/work/study/2024-2025/arch-pc/arch-pc$
```

Рис. 4.7: Копирование репозитория на рабочий компьютер

4.5 Настройка каталога курса

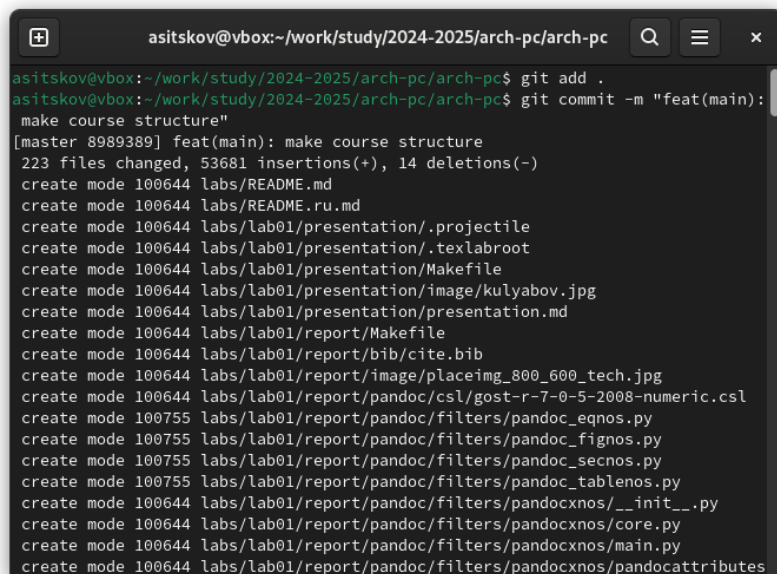
В каталоге курса удаляю лишние файлы и формирую необходимые каталоги.
(рис. -fig. 4.8)

A terminal window with a dark background. The title bar shows the user 'asitskov@vbox' and the current directory '~/work/study/2024-2025/arch-pc/arch-pc'. The terminal contains two lines of text: a command to remove 'package.json' and the resulting prompt.

```
asitskov@vbox:~/work/study/2024-2025/arch-pc/arch-pc$ rm package.json
asitskov@vbox:~/work/study/2024-2025/arch-pc/arch-pc$
```

Рис. 4.8: Настройка каталога курса

Делаю снимок сделанных изменений и push'у их на свой репозиторий в GitHub.
(рис. 4.9)

A terminal window with a dark background. The title bar shows the user 'asitskov@vbox' and the current directory '~/work/study/2024-2025/arch-pc/arch-pc'. The terminal shows a git commit command being executed, followed by a list of files created and their permissions.

```
asitskov@vbox:~/work/study/2024-2025/arch-pc/arch-pc$ git add .
asitskov@vbox:~/work/study/2024-2025/arch-pc/arch-pc$ git commit -m "feat(main):
make course structure"
[master 8989389] feat(main): make course structure
223 files changed, 53681 insertions(+), 14 deletions(-)
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
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
```

Рис. 4.9: Отправка изменений на удаленный репозиторий

4.6 Задания для самостоятельной работы

Копирую отчет по предыдущей лабораторной работе в нужный каталог (рис. 4.10).

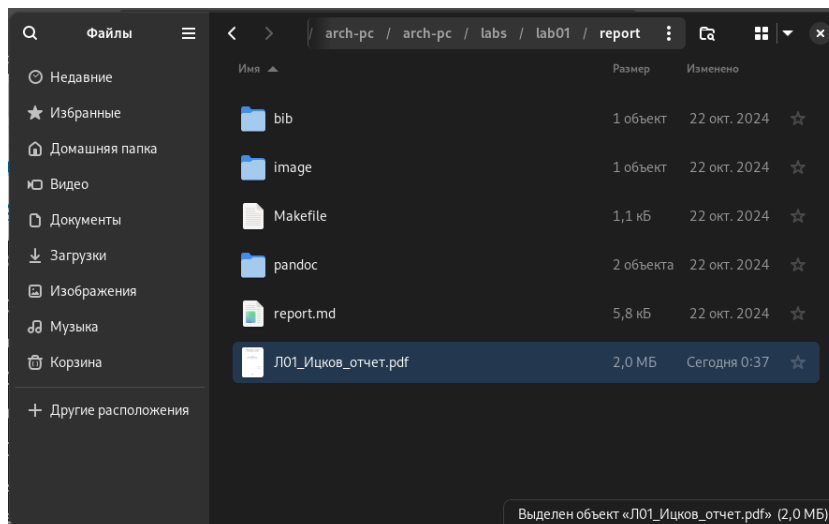


Рис. 4.10: Копирование прошлого отчета

Загружаю файлы на github (рис. 4.11).

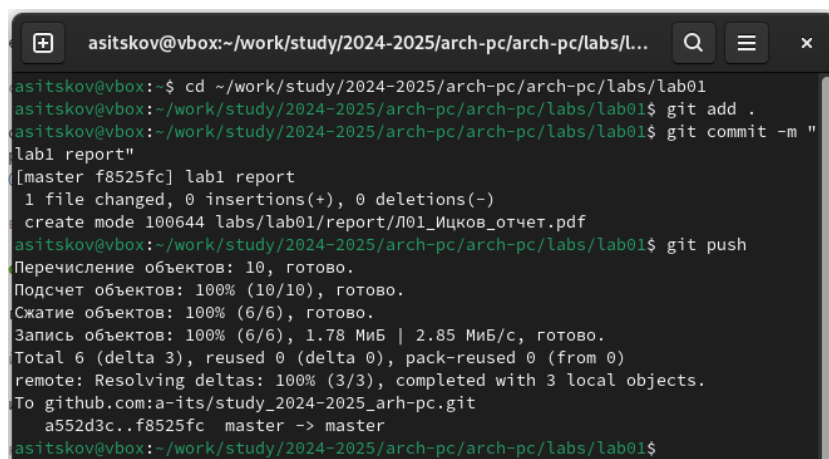


Рис. 4.11: Загрузка файлов на github

5 Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с реализацией VSC git.

Список литературы

1. Лабораторная работа
2. Курс на ТУИС
3. Методические указания
4. Шаблон выполнения лабораторной работы