

# **Лабораторная работа № 3**

**дисциплина: Архитектура компьютера**

Клюкин Михаил Александрович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
3.1	Системы контроля версий. Общие понятия . . . . .	7
3.2	Система контроля версий Git . . . . .	8
3.3	Основные команды git . . . . .	9
3.4	Стандартные процедуры при наличии центрального репозитория	9
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>11</b>
4.1	Настройка github . . . . .	11
4.2	Базовая настройка git . . . . .	11
4.3	Создание SSH ключа . . . . .	12
4.4	Создание рабочего пространства и репозитория курса на основе шаблона . . . . .	13
4.5	Создание репозитория курса на основе шаблона . . . . .	14
4.6	Настройка каталога курса . . . . .	15
4.7	Задание для самостоятельной работы . . . . .	16
<b>5</b>	<b>Выводы</b>	<b>18</b>
	<b>Список литературы</b>	<b>19</b>

## Список иллюстраций

4.1	Базовая настройка git . . . . .	12
4.2	Создание ключа SSH . . . . .	12
4.3	Копирование ключа в буфер обмена . . . . .	12
4.4	Загрузка сгенерированного SSH ключа . . . . .	13
4.5	Создание каталога для предмета “Архитектура компьютера” . . .	13
4.6	Создание репозитория на основе шаблона . . . . .	14
4.7	Изменение текущей директории . . . . .	14
4.8	Клонирование созданного репозитория . . . . .	15
4.9	Настройка каталога курса . . . . .	15
4.10	Выполнение команд git add и git commit . . . . .	16
4.11	Выполнение команды git push . . . . .	16
4.12	Создание отчета о выполнении работы . . . . .	16
4.13	Копирование предыдущих отчетов в соответствующие каталоги .	17
4.14	Загрузка файлов на github . . . . .	17

## Список таблиц

# 1 Цель работы

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

## 2 Задание

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

## **3 Теоретическое введение**

### **3.1 Системы контроля версий. Общие понятия**

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 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.

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.

## **3.2 Система контроля версий Git**

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.



### 3.3 Основные команды git

git init – создание основного дерева репозитория git pull – получение обновлений 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 git add . – добавить все изменённые и/или созданные файлы и/или каталоги git add имена\_файлов – добавить конкретные изменённые и/или созданные файлы и/или каталоги git rm имена\_файлов – удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории) git commit -am 'Описание коммита' – сохранить все добавленные изменения и все изменённые файлы git checkout -b имя\_ветки – создание новой ветки, базирующейся на текущей git checkout имя\_ветки – переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) git push origin имя\_ветки – отправка изменений конкретной ветки в центральный репозиторий git merge --no-ff имя\_ветки – слияние ветки с текущим деревом git branch -d имя\_ветки – удаление локальной уже слитой с основным деревом ветки git branch -D имя\_ветки – принудительное удаление локальной ветки git push origin :имя\_ветки – удаление ветки с центрального репозитория

### 3.4 Стандартные процедуры при наличии центрального репозитория

Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений): git checkout master git pull git checkout -b имя\_ветки

Затем можно вносить изменения в локальном дереве и/или ветке.

После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральной репозитории. Для этого необходимо проверить, какие файлы изменились к текущему моменту: `git status` и при необходимости удаляем лишние файлы, которые не хотим отправлять в центральный репозиторий.

Затем полезно просмотреть текст изменений на предмет соответствия правилам ведения чистых коммитов: `git diff` Если какие-либо файлы не должны попасть в коммит, то помечаем только те файлы, изменения которых нужно сохранить. Для этого используем команды добавления и/или удаления с нужными опциями: `git add имена_файлов` `git rm имена_файлов`

Если нужно сохранить все изменения в текущем каталоге, то используем: `git add` .

Затем сохраняем изменения, поясняя, что было сделано: `git commit -am "Some commit message"` и отправляем в центральный репозиторий: `git push origin имя_ветки` или `git push`

## 4 Выполнение лабораторной работы

### 4.1 Настройка github

Создали учетную запись на сайте <https://github.com/> и заполнили основные данные.

### 4.2 Базовая настройка git

Сделали предварительную конфигурацию git: открыли терминал и ввели следующие команды, указав имя и email владельца репозитория (Рис. 1):

```
git config - -global user.name ""
```

```
git config - -global user.email "work@email"
```

Настроили utf-8 в выводе сообщений git (Рис. 4.1):

```
git config - -global core.quotepath false
```

Задали имя начальной ветки master (Рис. 4.1):

```
git config -global init.defaultBranch master
```

Задали параметр autocrlf (Рис. 4.1):

```
git config -global core.autocrlf input
```

Задали параметр safecrlf (Рис. 4.1):

```
git config -global core.safecrlf warn
```

```
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$ git config --global user.name "MaKYaro"
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$ git config --global user.email "klykin-2@yandex.ru"
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$ git config --global core.quotepath false
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$ git config --global init.defaultBranch master
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$ git config --global core.autocrlf input
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$ git config --global core.safecrlf warn
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$
```

Рис. 4.1: Базовая настройка git

## 4.3 Создание SSH ключа

Сгенерировали пару ключей – приватный и открытый (Рис. 4.2):  
ssh-keygen -C "Имя Фамилия work@mail"

```
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$ ssh-keygen -C "Михаил Ключин klykin-2@yandex.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/maklyukin/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/maklyukin/.ssh/id_rsa
Your public key has been saved in /home/maklyukin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:S/waydE3GbKwtuK1dU4TChStypJemQCDWBb1z5KTU Михаил Ключин klykin-2@yandex.ru
The key's randomart image is:
+----[RSA 3072]-----+
| . = . . . 0 . . . |
| . + .. E. + * . . . |
| = 0. = + 0. 0 |
| . + .. 0 = 0 . = . |
| . 0. S + 0 + 0 . . |
| . 0. . + + 0 |
| . 0 0 0 |
| . . . . . |
| . 0. |
+----[SHA256]-----+
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$
```

Рис. 4.2: Создание ключа SSH

Скопировали из локальной консоли ключ в буфер обмена (Рис. 4.3):  
cat ~/.ssh/id\_rsa.pub | xclip -sel clip

```
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
maklyukin@makyarо-HP-Laptop-15-da0xxx:~$
```

Рис. 4.3: Копирование ключа в буфер обмена

Загрузили сгенерированный открытый ключ: зашли на сайт <https://github.org/> под своей учетной записью и перешли в меню Settings, в боковом меню выбрали SSH and GPG keys и нажали кнопку New SSH key, вставили ключ в появившемся на сайте поле и указали имя для ключа – Title (Рис. 4.4).

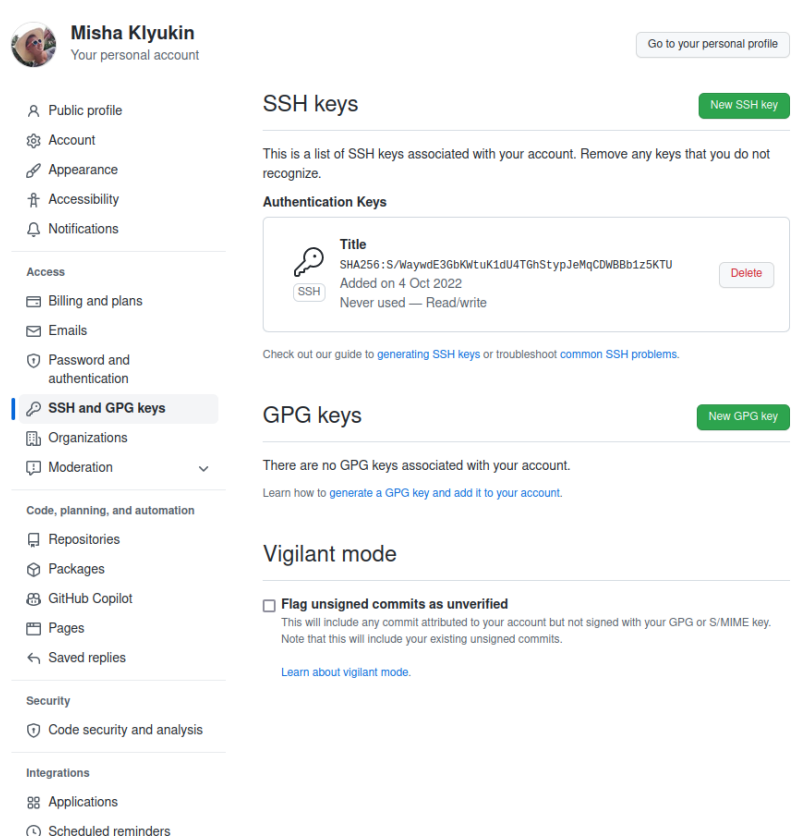


Рис. 4.4: Загрузка сгенерированного SSH ключа

## 4.4 Создание рабочего пространства и репозитория курса на основе шаблона

С помощью терминала создали каталог для предмета “Архитектура компьютера” (Рис. 4.5):

```
mkdir -p ~/work/study/2022-2023/“Архитектура компьютера”
```

```
maklyukin@nakyaro-HP-Laptop-15-da0xxx: $ mkdir -p ~/work/study/2022-2023/“Архитектура компьютера”
maklyukin@nakyaro-HP-Laptop-15-da0xxx: $
```

Рис. 4.5: Создание каталога для предмета “Архитектура компьютера”

## 4.5 Создание репозитория курса на основе шаблона

Перешли на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>, выбрали Use this template, в открывшемся окне задали имя репозитория study\_2022–2023\_arh-pc и создали репозиторий (Рис. 4.6).

Create a new repository from course-directory-student-template

The new repository will start with the same files and folders as yamadharma/course-directory-student-template.

Owner \* Repository name \*

MaKYaro / study\_2022-2023\_arh-pc ✓

Great repository names < Your new repository will be created as study\_2022-2023\_arh-pc. ndly-robot?

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.

☐ Include all branches  
Copy all branches from yamadharma/course-directory-student-template and not just master.

*i* You are creating a public repository in your personal account.

Create repository from template

Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

© 2022 GitHub, Inc.

Рис. 4.6: Создание репозитория на основе шаблона

В терминале перешли в каталог курса (Рис. 4.7):

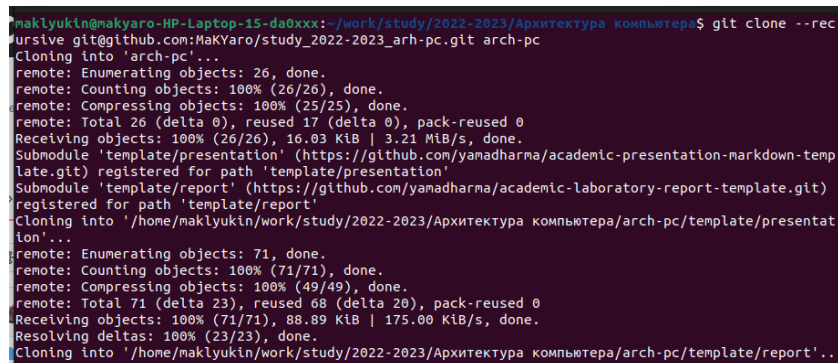
`cd ~/work/study/2022-2023/“Архитектура компьютера”`

```
maklyukin@nakyaro-HP-Laptop-15-da0xxx:~$ cd ~/work/study/2022-2023/“Архитектура компьютера”
maklyukin@nakyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера$
```

Рис. 4.7: Изменение текущей директории

Склонировали созданный репозиторий (Рис. 4.8):

```
git clone --recursive git@github.com:<user_name>/study_2022-2023_arh-  
pc.git arch-pc
```



```
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера$ git clone --rec  
ursive git@github.com:MaKYaro/study_2022-2023_arh-pc.git arch-pc  
Cloning into 'arch-pc'...  
remote: Enumerating objects: 26, done.  
remote: Counting objects: 100% (26/26), done.  
remote: Compressing objects: 100% (25/25), done.  
remote: Total 26 (delta 0), reused 17 (delta 0), pack-reused 0  
Receiving objects: 100% (26/26), 16.03 KiB | 3.21 MiB/s, done.  
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-temp  
late.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/maklyukin/work/study/2022-2023/Архитектура компьютера/arch-pc/template/presentat  
ion'...  
remote: Enumerating objects: 71, done.  
remote: Counting objects: 100% (71/71), done.  
remote: Compressing objects: 100% (49/49), done.  
remote: Total 71 (delta 23), reused 68 (delta 20), pack-reused 0  
Receiving objects: 100% (71/71), 88.89 KiB | 175.00 KiB/s, done.  
Resolving deltas: 100% (23/23), done.  
Cloning into '/home/maklyukin/work/study/2022-2023/Архитектура компьютера/arch-pc/template/report'..
```

Рис. 4.8: Клонирование созданного репозитория

## 4.6 Настройка каталога курса

Перешли в каталог курса (Рис. 4.9):

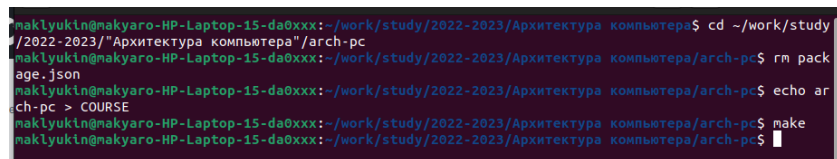
```
cd ~/work/study/2022-2023/“Архитектура компьютера”/arch-pc
```

Удалили лишние файлы (Рис. 4.9):

```
rm package.json
```

Создали необходимые каталоги (Рис. 4.9):

```
echo arch-pc > COURSE make
```



```
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера$ cd ~/work/study  
/2022-2023/“Архитектура компьютера”/arch-pc  
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ rm pack  
age.json  
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ echo ar  
ch-pc > COURSE  
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ make  
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$
```

Рис. 4.9: Настройка каталога курса

Отправили файлы на сервер (Рис. 4.10, 4.11):

```
git add . git commit -am 'feat(main): make course structure' git push
```

```

maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ git add .
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ git commit -an 'feat(main): make course structure'
[master 116102e] feat(main): make course structure
91 files changed, 8229 insertions(+), 14 deletions(-)
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 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 100644 labs/lab02/report/report.md
create mode 100644 labs/lab03/presentation/Makefile
create mode 100644 labs/lab03/presentation/image/kulyabov.jpg
create mode 100644 labs/lab03/presentation/presentation.md

```

Рис. 4.10: Выполнение команд git add и git commit

```

maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 8 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (20/20), 310.95 KiB | 1.00 MiB/s, done.
Total 20 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:MaKVaro/study_2022-2023_arh-pc.git
   d08c299..116102e  master -> master
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$

```

Рис. 4.11: Выполнение команды git push

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

## 4.7 Задание для самостоятельной работы

Создали отчет о выполнении лабораторной работы в соответствующем каталоге рабочего пространства – labs/lab03/report (Рис. 4.12).

```

maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ touch ./labs/lab03/report/л03_Клюкин_отчет.docx
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ ls ./labs/lab03/report
bib image Makefile pandoc report.md л03_Клюкин_отчет.docx
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютера/arch-pc$

```

Рис. 4.12: Создание отчета о выполнении работы

Скопировали отчеты по выполнению предыдущих лабораторных в соответствующие каталоги созданного рабочего пространства (Рис. 4.13).



```

maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютер
epa/arch-pc$ mv ~/Загрузки/Л01_Клюкин_отчет.pdf ./labs/lab01/report
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютер
epa/arch-pc$ mv ~/Загрузки/Л02_Клюкин_отчет.pdf ./labs/lab02/report
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютер
epa/arch-pc$ ls ./labs/lab01/report
bib image Makefile pandoc report.md Л01_Клюкин_отчет.pdf
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютер
epa/arch-pc$ ls ./labs/lab02/report
bib image Makefile pandoc report.md Л02_Клюкин_отчет.pdf
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютер
epa/arch-pc$

```

Рис. 4.13: Копирование предыдущих отчетов в соответствующие каталоги

Загрузили файлы на github (Рис. 4.14).

```

maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютер
epa/arch-pc$ git add .
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютер
epa/arch-pc$ git commit -am 'Добавили предыдущие отчеты'
[master ed53057] Добавили предыдущие отчеты
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Клюкин_отчет.pdf
create mode 100644 labs/lab02/report/Л02_Клюкин_отчет.pdf
create mode 100644 labs/lab03/report/Л03_Клюкин_отчет.docx
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютер
epa/arch-pc$ git push
Enumerating objects: 19, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 3.66 MiB | 1.63 MiB/s, done.
Total 11 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To github.com:MaKYaro/study_2022-2023_arh-pc.git
116102e..ed53057 master -> master
maklyukin@makyaro-HP-Laptop-15-da0xxx:~/work/study/2022-2023/Архитектура компьютер
epa/arch-pc$

```

Рис. 4.14: Загрузка файлов на github

## **5 Выводы**

Изучили идеологию и применение средств контроля версий. Приобрели практические навыки по работе с системой git.

# Список литературы

::: Демидова А. В. Лабораторная работа №3. Система контроля версий Git –  
Методическое пособие{#refs} :::