

Отчет по лабораторной работе №2

Рогозин Игорь Андреевич

- **Содержание:**

- 1. Цель работы 5
- 2 Задание 6
- 3 Теоретическое введение 7
- 4 Выполнение лабораторной работы 9
- 5 Выводы 28

Список иллюстраций

- 4.1.1: Терминал ОС Линукс 7
- 4.1.2: Домашний каталог ОС Линукс 8
- 4.1.3: Выполнение команды pwd. 10
- 4.1.4: Выполнение команды cd 10
- 4.1.5: Переход в каталог /usr/local. 10
- 4.1.6: Выполнение команды ls. 11
- 4.1.7: Просмотр каталога /usr/local. 11
- 4.2.1: Создание папки parentdir и проверка. 13
- 4.2.2: Создание подкаталога. 13
- 4.2.3: Создание нескольких каталогов 1 командой. 13
- 4.2.4: Создание подкаталога без перехода в каталог. .13
- 4.2.5: Проверка создания папки. 14
- 4.2.6: Создание подкаталогов 1 командой. 14
- 4.2.7: Создание файла и проверка. 14
- 4.3.1.: Удаление каталога с файлами. 15
- 4.3.2: Удаление каталогов с файлами с началом dir. . . . 16
- 4.3.3: Создание файлов. 17
- 4.3.4: Копирование и перенос файлов. 17
- 4.3.5: Проверка выполнения команд. 17
- 4.3.6: Переименование файлов. 18
- 4.3.7: Переименование файлов в каталоге. 18

- 4.4.1: Выполнение команды Cat. . 19
- 4.5.1: Выполнение команд. 21
- 4.5.2: Выполнение команд. 22
- 4.5.3: Выполнение команд. 22
- 4.5.4: Выполнение команд. 23
- 4.5.5: Выполнение команд. 24
- 4.5.6: Выполнение команд. 24
- 4.5.7: Выполнение команд. 25
- 4.5.8: Выполнение команд. 26
- 4.5.9: Выполнение команд удаления . . 27

1 Цель работы

- Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой 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.
- 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.
- Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial.

- Понятия в VCS:
- Хранилище – это система, которая позволяет хранить все версии существовавших файлов.
- Commit – запись изменений.
- История – список изменений ранее.
- Рабочая копия – копия файла, с которой активно ведется работа.
- В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Децентрализованная VCS предназначена для обработки ветвлений и слияний как части своей ДНК, сохраняя указатель на предыдущий коммит в каждом коммите, поэтому любое изменение можно отследить до общего предка.
- 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

- 1.1 Открываем терминал (рис. 4.1):

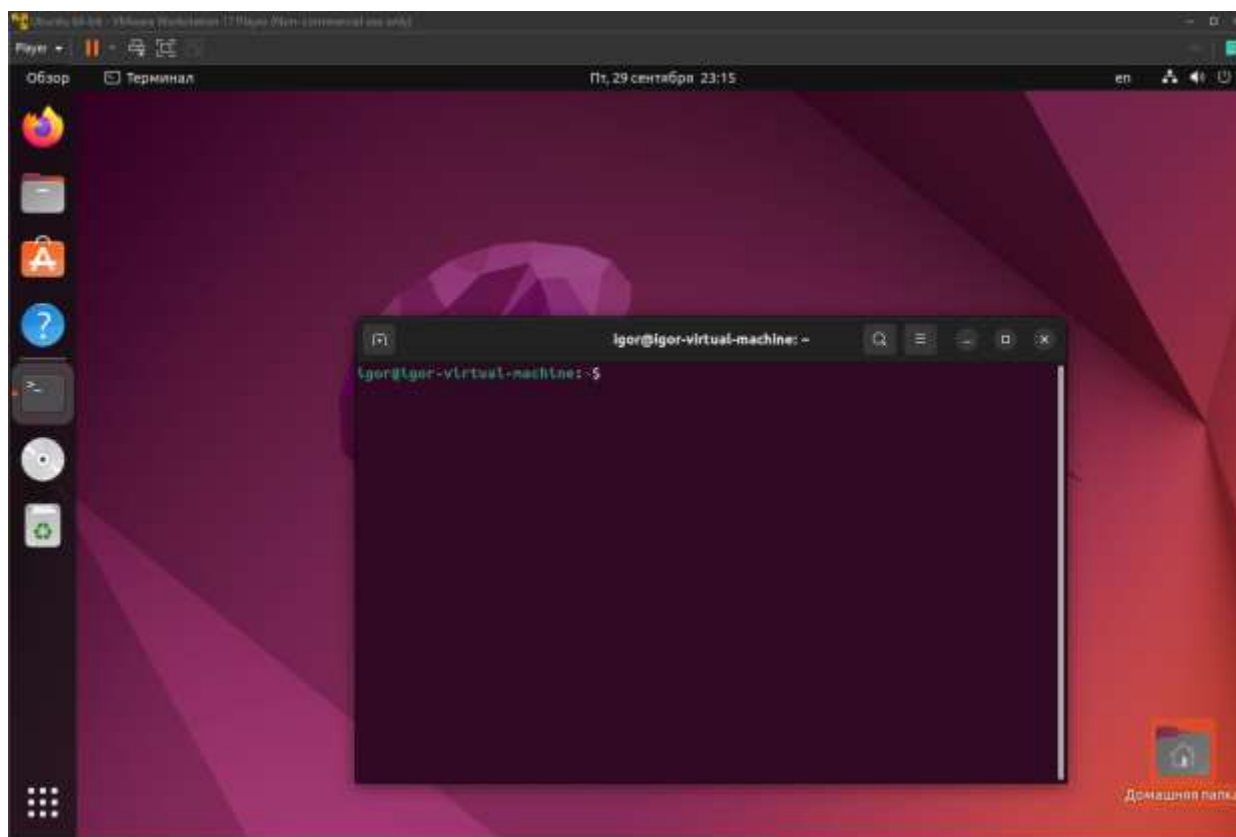


Рис. 4.1.1: Терминал ОС Линукс

- 1.2 После создания аккаунта на сайте <https://github.com/> сделаем предварительную конфигурацию git.

```
igor@igor-virtual-machine:~$ git config --global user.name "Igor Rogozin"  
igor@igor-virtual-machine:~$ git config --global user.email "osnovagt@gmail.com"
```

Рис. 4.1.2 Настройка конфигурации git

Вводим в консоль команды, указав имя и email владельца репозитория:

1.3 Настроим utf-8 в выводе сообщений git:

```
igor@igor-virtual-machine:~$ git config --global core.quotepath false
```

Рис. 4.1.3 Настройка конфигурации git

1.4 Зададим имя начальной ветки (будем называть её master):

```
igor@igor-virtual-machine:~$ git config --global init.defaultBranch master  
igor@igor-virtual-machine:~$
```

Рис. 4.1.4 Настройка конфигурации git

1.5 Зададим параметры:

Параметр autocrlf:

Параметр safecrlf:

```
igor@igor-virtual-machine:~$ git config --global core.autocrlf input  
igor@igor-virtual-machine:~$ git config --global core.safecrlf warn
```

Рис. 4.1.5 Настройка конфигурации git

• 2 Создание SSH ключа

- 2.1 Теперь для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый):

```
igor@igor-virtual-machine:~$ ssh-keygen -C "Igor Rogozin osnovagt@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/igor/.ssh/id_rsa):
Created directory '/home/igor/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/igor/.ssh/id_rsa
Your public key has been saved in /home/igor/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:eEy+rdl8olk/dSHJ1Xnnc3a02B9yb807Eva1/NJyJ1M Igor Rogozin osnovagt@gmail.com
```

Рис. 4.2.1: Создание SSH-ключа в терминале

2.2 Далее загрузим сгенерированный ключ на github.

Для этого зайти на сайт <http://github.com/> под своей учётной записью и перейти в меню Settings. После этого выбрать в боковом меню SSH and GPG keys.

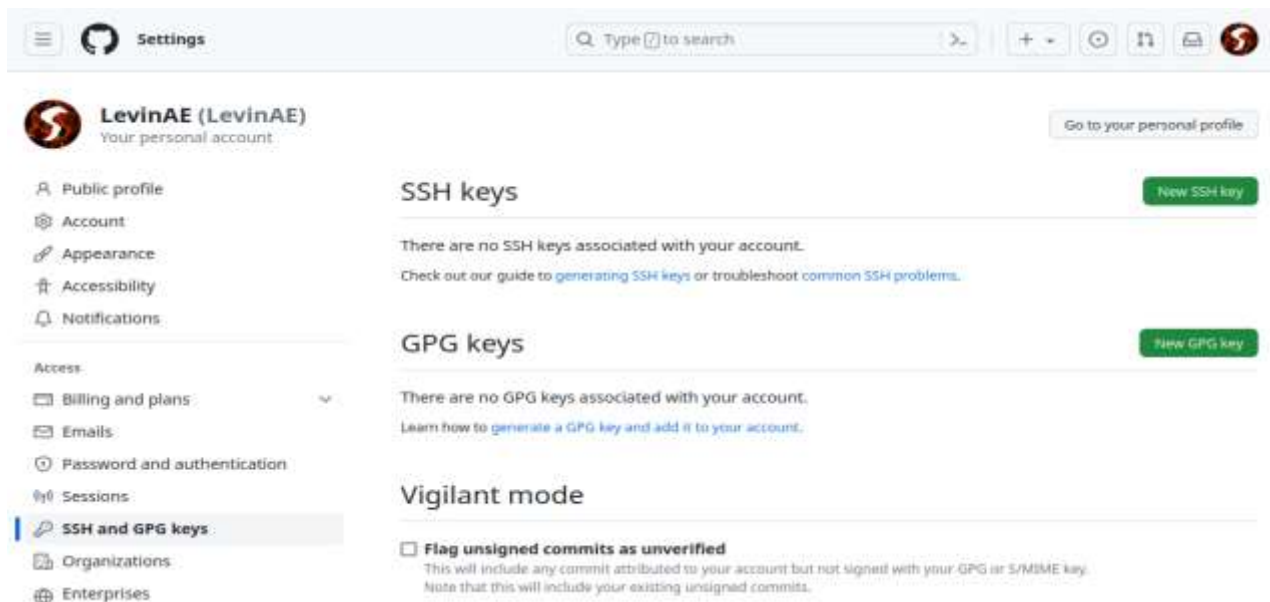


Рис. 4.2.2: Настройки ключей github

- 2.3 Копируем ключ с помощью команды `cat ~/.ssh/id_rsa.pub | xclip -sel clip` и вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title). Нажимаем кнопку Add SSH Key

```
igor@igor-virtual-machine:~$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
```

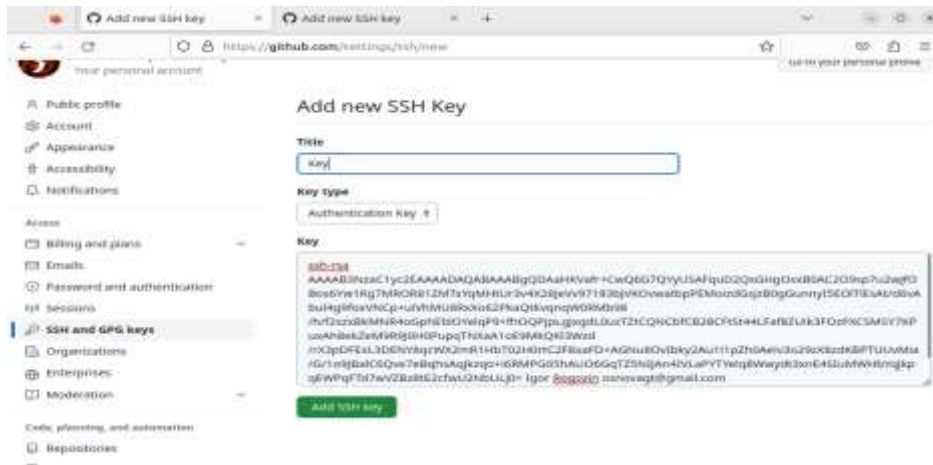


Рис. 4.2.3: Сохранение ключа на github

3 Сознание рабочего пространства и репозитория курса на основе шаблона

3.1 Откроем терминал и создадим каталог для предмета «Архитектура компьютера»:

```
igor@igor-virtual-machine:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"  
igor@igor-virtual-machine:~$
```

Рис. 4.3.1: Создание основной директории

Репозиторий на основе шаблона можно создать через web-интерфейс github.

3.2 Перейдём на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Далее выберите Use this template.

3.3 В открывшемся окне задайте имя репозитория (Repository name) study_2023–2024_arhpc и создайте репозиторий (кнопка Create repository from template).

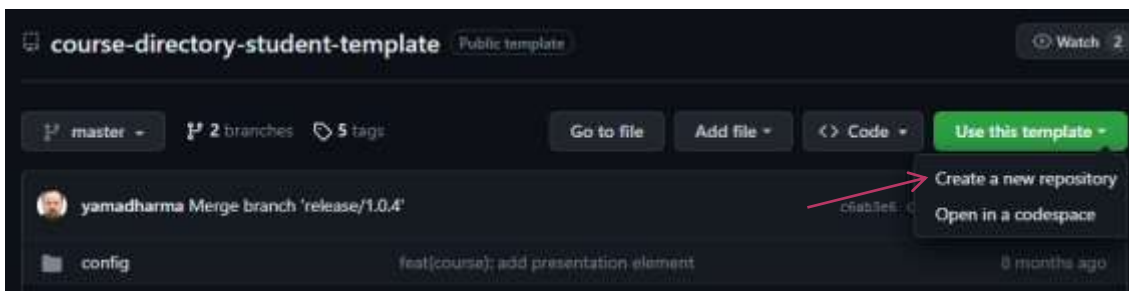


Рис. 4.3.2: Создание основной директории на github

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 (*).

Owner * LevinAE / Repository name * study_2023-2024_arhpc

✓ Your new repository will be created as `study_2023-2024_arhpc`.
The repository name can only contain ASCII letters, digits, and the characters `-`, `.`, and `_`.

Great repository names are short and memorable. Need inspiration? How about `probable-spoon`?

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.

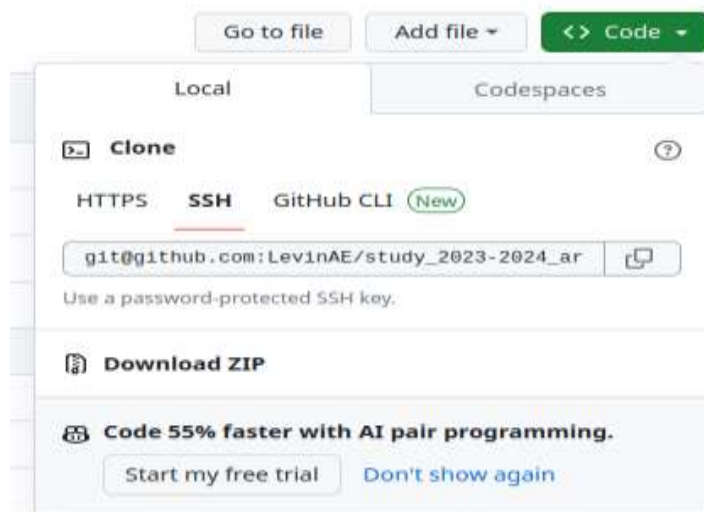
[Create repository](#)

Рис. 4.3.3: Создание основной директории на github

3.4 Теперь зайдём в терминал и перейдём в каталог курса:

```
igor@igor-virtual-machine:~$ cd ~/work/study/2023-2024/"Архитектура компьютера"  
igor@igor-virtual-machine:~/work/study/2023-2024/Архитектура компьютера$
```

И клонируем созданный репозиторий:



```
igor@igor-virtual-machine:~/work/study/2023-2024/Архитектура компьютера$ git clone --recursive git@github.com:LevinAE/study_2023-2024_arhpc.git  
Клонирование в «study_2023-2024_arhpc»...
```

Рис. 4.3.4: Клонирование репозитория из github

4 Настройка каталога курса

4.1 Для начала перейдём в каталог курса:

```
larogozin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера $ cd arch-pc/  
larogozin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ ls  
CHANGELOG.md  COURSE  Makefile  README.en.md  README.md  
config        LICENSE  package.json  README.git-flow.md  template  
larogozin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ rm pack  
age.json
```

Рис. 4.4.1: Удаление файла package.json

4.2 Создадим необходимые каталоги:

```
larogozin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ rm package.json  
larogozin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ echo arch-pc > COURSE  
larogozin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ make
```

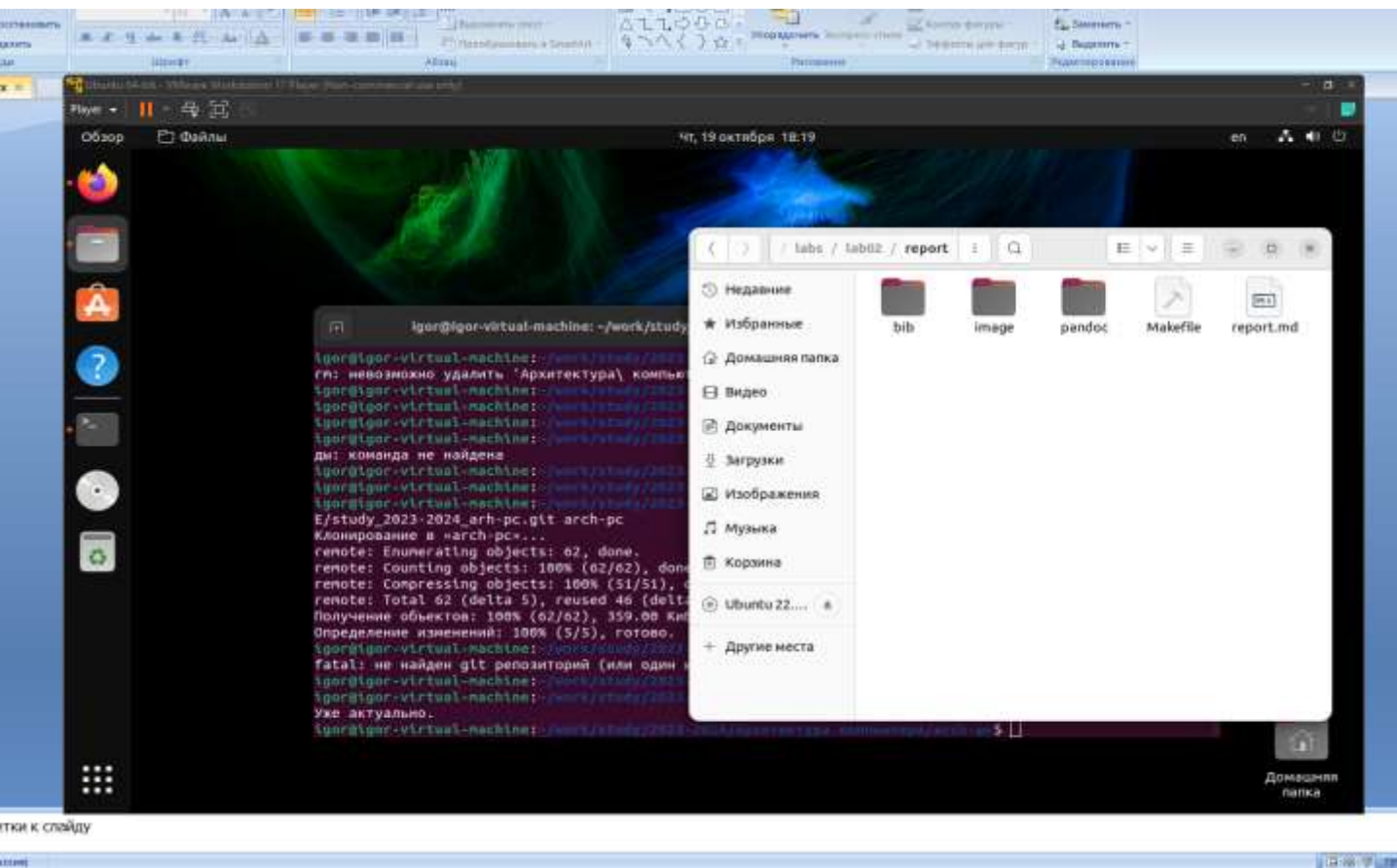
Рис. 4.2.7: Создание каталогов и проверка

4.3 После этого отправим все изменения на сервер

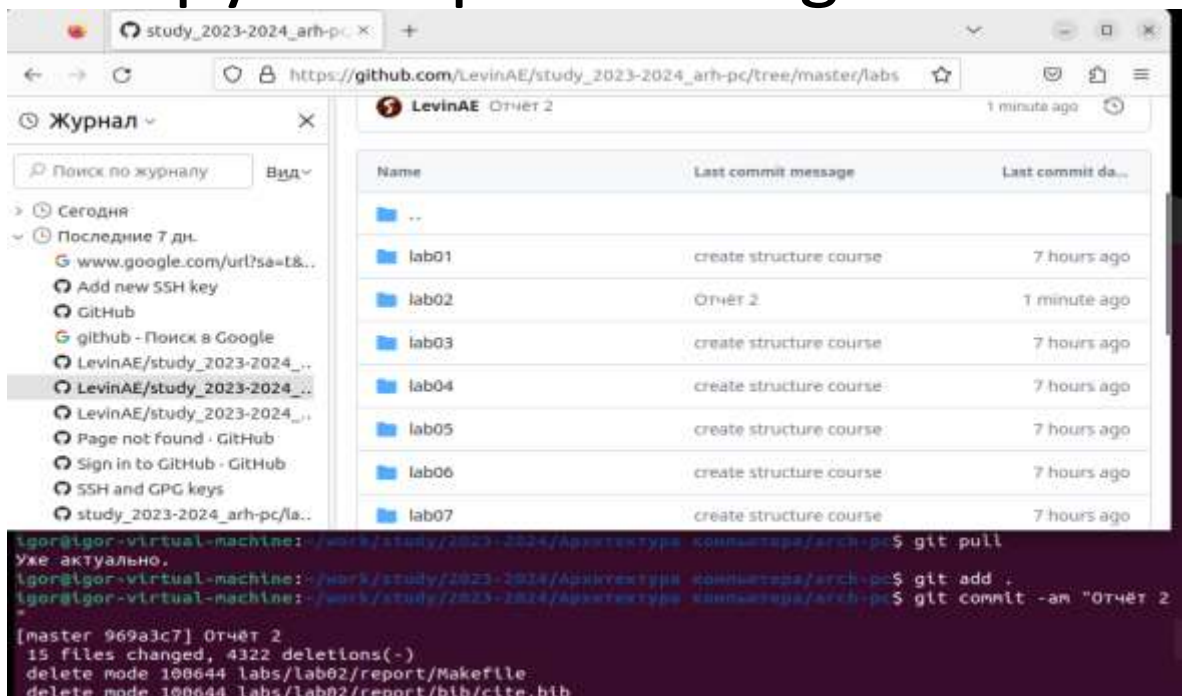
```
larogozin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ make  
larogozin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git add .  
larogozin@dk6n51 ~/work/study/2023-2024/Архитектура компьютера/arch-pc $ git commit -am "create structure course"  
[master 13b386b] create structure course  
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
```

Самостоятельное 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.



```
igor@igor-virtual-machine: ~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 11, готово.
```

5 Выводы

- Сегодня я научился работать с github и сохранять, а затем загружать отчёты в репозиторий.