

# Отчет по лабораторной работе №2

## Дисциплина: Архитектура компьютера

Карпачев Ярослав Олегович

### Содержание

1	Цель работы .....	1
2	Задание .....	1
3	Теоретическое введение.....	2
4	Выполнение лабораторной работы.....	4
4.1	Настройка GitHub.....	4
4.2	Базовая настройка Git.....	4
4.3	Создание SSH-ключа.....	5
4.4	Создание репозитория курса на основе шаблона.....	5
4.5	Настройка каталога курса.....	6
4.6	Выполнение заданий для самостоятельной работы .....	7
5	Выводы.....	8

### 1 Цель работы

Целью работы является изучить Git. 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. Выполнение заданий 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. 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. 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.

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.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

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.

#### *Основные команды `git`{#tbl:std-dir}*

Команда	Описание
<code>git init</code>	создание основного дерева репозитория
<code>git pull</code>	получение обновлений (изменений) текущего дерева из центрального репозитория
<code>git push</code>	отправка всех произведённых изменений локального дерева в центральный репозиторий
<code>git status</code>	просмотр списка изменённых файлов в текущей директории
<code>git diff</code>	просмотр текущих изменений

Команда	Описание
<code>git add</code>	добавить все изменённые и/или созданные файлы и/или каталоги
<code>git add имена_файлов</code>	добавить конкретные изменённые и/или созданные файлы и/или каталоги
<code>git rm имена_файлов</code>	удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)
<code>git commit -am 'Описание коммита'</code>	сохранить все добавленные изменения и все изменённые файлы
<code>git checkout -b имя_ветки</code>	создание новой ветки, базирующейся на текущей
<code>git checkout имя_ветки</code>	переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)
<code>git push origin имя_ветки</code>	отправка изменений конкретной ветки в центральный репозиторий
<code>git merge --no-ff имя_ветки</code>	слияние ветки с текущим деревом
<code>git branch -d имя_ветки</code>	удаление локальной уже слитой с основным деревом ветки
<code>git branch -D</code>	принудительное удаление локальной ветки

Команда	Описание
<code>git push origin :имя_ветки</code>	удаление ветки с центрального репозитория

## 4 Выполнение лабораторной работы

### 4.1 Настройка GitHub

Создал учетную запись на сайте <https://github.com/> (рис. 1).

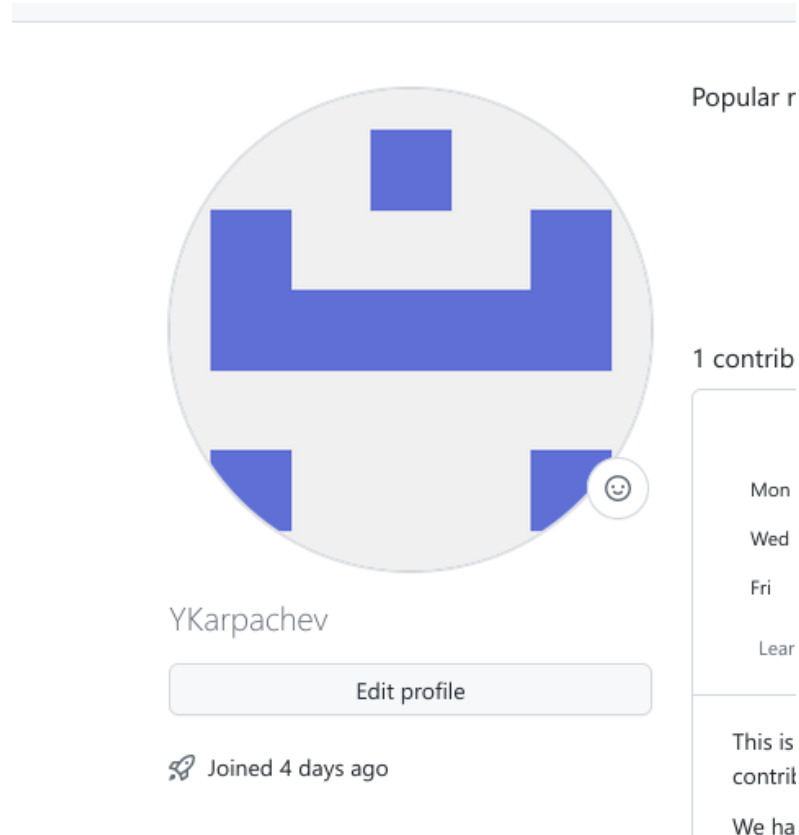


Рис. 1: Мой профиль в github

### 4.2 Базовая настройка Git

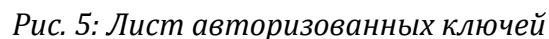
Произвел базовые настройки Git, email, name и тд (рис. 2).

Рис. 2: Настройка git

Сгенерировал ключи приватный и открытый (рис. 3). Просмотрел с помощью cat, скопировал его, вставил в Github (рис. 4). Авторизовала ключ (рис. 5).

Рис. 3: Создание SHH ключа

Рис. 4: Открытый ключ SHH



На основе шаблона создал свой репозиторий (рис 6.)

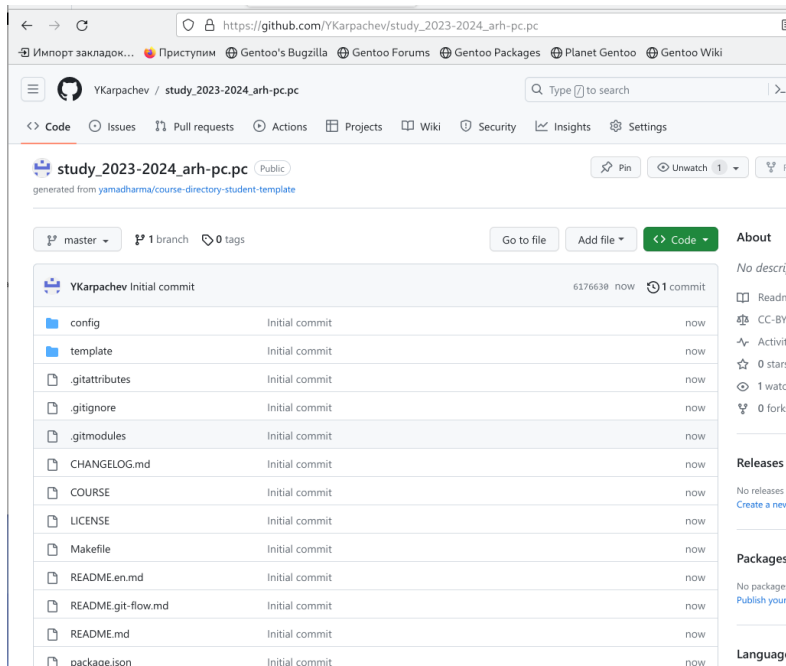


Рис. 6: Страница созданного репозитория

Открыл терминал и перешел в каталог. Клонировала созданный репозиторий, скопировав ссылку для клонирования на странице созданного репозитория Code -> SSH (рис. 7).

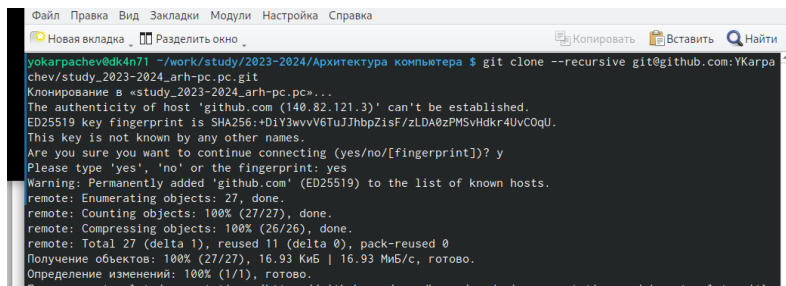


Рис. 7: Клонирование репозитория

## 4.5 Настройка каталога курса

Удалил лишние файлы с помощью команды `rm` (рис. 8).

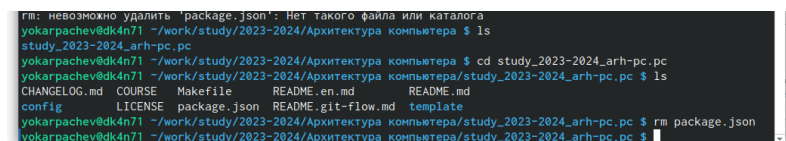


Рис. 8: Удаление лишних файлов

создал необходимые каталоги с помощью `git add`, прокомментировал и сохранил изменения на сервере как добавление с помощью `git commit` (рис. 9), и отправил изменения на сервер (рис. 10) и проверил на гитхаб правильность действий (рис. 11)

```

yokarpachev@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc $ echo arch-pc > COURSE
yokarpachev@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc $ make
make: 'ls' > 'cd' не существует: команда.
yokarpachev@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc $ git add
Ничего не проиндексировано.
yokarpachev@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc $ git commit -m "feat(main): make course structure"
190 files changed, 5173 insertions(+), 14 deletions(-)
create mode 100644 labs/README.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/fib/cite.bib
create mode 100644 labs/lab01/report/image/pilcing_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-8-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_epros.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_figmap.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secmap.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tableofcontents.py

```

Рис. 9: Создание необходимых каталогов и сохранение изменений с `git commit`

```

create mode 100644 presentation/report/report.md
yokarpachev@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc $ git push
Перечисление объектов: 37, готово.
Подсчет объектов: 100% (37/37), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (35/35), 342.13 Киб | 2.41 Мб/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:YKarpachev/study_2023-2024_arh-pc-pc.git
6176630..5a98bfe master -> master
yokarpachev@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc $

```

Рис. 10: Отправка файлов на сервер

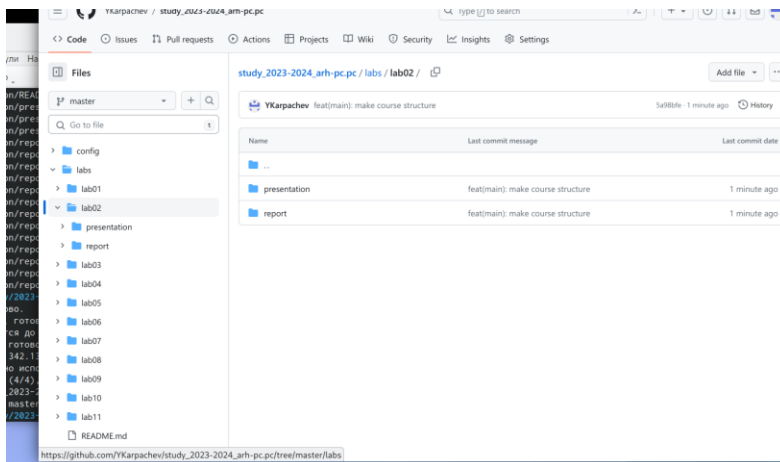


Рис. 11: Страница репозитория

## 4.6 Выполнение заданий для самостоятельной работы

Перехожу в директорию `labs/lab02/report` с помощью утилиты `cd`. Создаю в каталоге файл для отчета по второй лабораторной работе с помощью утилиты `touch`, оформлю его и отправил на сервер. (рис. 12).

```

yokarpachev@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab02/report $ touch R02_Карнаев_отчет.doc
yokarpachev@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab02/report $ ls
bib image Makefile pandoc_report.md R02_Карнаев_отчет.doc
yokarpachev@dk4n71: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab02/report $

```

Рис. 12: Создание текстового документа для оформления отчета и загрузка его на сервер

Перенес (копировал) и загрузил предыдущий отчет на сервер в каталог `labs/lab01/report`. И проверил правильность действий на Github.

```
yokarpachev@6471 ~/Документы $ cd ~/Документы/ОтчетФинал.docx ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab01
yokarpachev@6471 ~/Документы $ cd ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab01
yokarpachev@6471 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab01 $ ls
presentation report ОтчетФинал.docx
yokarpachev@6471 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab01 $ git add
yokarpachev@6471 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab01 $ git commit -m "Add existing files"
git: <commit> не является командой git. Смотрите <git --help>.

Самые похожие команды:
commit
yokarpachev@6471 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab01 $ git commit -m "Add existing files"
[master 49a861b] Add existing files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 180544 labs/lab01/ОтчетФинал.docx
yokarpachev@6471 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab01 $ git push
Пересечение объектов: 8, готово.
Подсчет объектов: 100% (8/8), готово.
При сжатии изменений используется по 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 1.82 Мб | 8.26 Мб/с, готово.
Всего 5 (изменений: 2), повторно использовано 0 (изменений: 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:Yokarpachev/study_2023-2024_arh-pc-pc.git
5d8b0fe..49a861b master -> master
yokarpachev@6471 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc-pc/labs/lab01 $
```

Рис. 13: добавление и отправка файлов на сервер

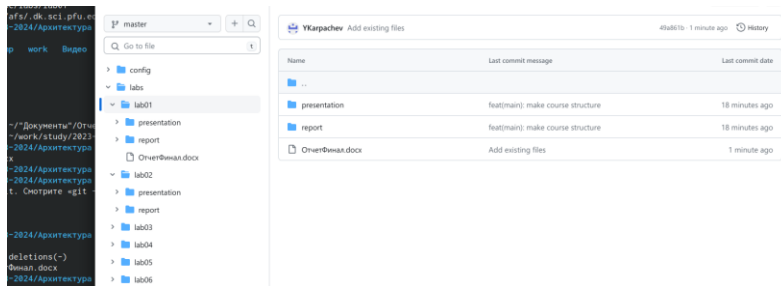


Рис. 14: проверка своих действий

## 5 Выводы

В ходе лабораторной работы я получил практически навыки работы с системой git, а также изучил Git посредством выполнения практических заданий.