

# **Отчет по лабораторной работе №2**

**Дисциплина: архитектура компьютера**

Стрижов Дмитрий Павлович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>3</b>
<b>2</b>	<b>Задание</b>	<b>4</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>5</b>
3.1	Базовая настройка git . . . . .	5
3.2	Создание SSH ключа . . . . .	6
3.3	Создание рабочего пространства и репозитория курса на основе шаблона . . . . .	7
3.4	Создание репозитория курса на основе шаблона . . . . .	7
3.5	Настройка каталога курса . . . . .	8
3.6	Выполнение заданий для самостоятельной работы . . . . .	11
<b>4</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

# 1 Цель работы

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

## 2 Задание

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

## 3 Выполнение лабораторной работы

### 3.1 Базовая настройка git

Создаю аккаунт на github и начинаю работу.

Для начала задаю мои имя и email (рис. 3.1). Для этого используется команда `git config --global`.

```
[dpstrizhov@localhost-live ~]$ git config --global user.name "<Strizhov Dmitriy>"
[dpstrizhov@localhost-live ~]$ git config --global user.email "<1132236954@pfur.ru>"
[dpstrizhov@localhost-live ~]$
```

Рис. 3.1: Указание имени и адреса почты владельца

Настраиваем кодировку utf-8 в выводе сообщений, написав `core.quotePath false` (рис. 3.2).

```
[dpstrizhov@localhost-live ~]$ git config --global core.quotePath false
[dpstrizhov@localhost-live ~]$
```

Рис. 3.2: Настройка кодировки вывода сообщений

Задаем имя начальной ветки с помощью конфигурации `init.defaultBranch` и назовем её `master` (рис. 3.3).

```
[dpstrizhov@localhost-live ~]$ git config --global init.defaultBranch master
[dpstrizhov@localhost-live ~]$
```

Рис. 3.3: Указание имени начальной ветки

Задаем имя начальной ветки с помощью конфигурации `init.defaultBranch` и назовем её `master` (рис. 3.4).

```
[dpstrizhov@localhost-live ~]$ git config --global core.autocrlf input
[dpstrizhov@localhost-live ~]$
```

Рис. 3.4: Параметр `core.autocrlf`

```
[dpstrizhov@localhost-live ~]$ git config --global core.safecrlf warn
[dpstrizhov@localhost-live ~]$
```

Рис. 3.5: Параметр `core.safecrlf`

На этом и заканчивается базовая настройка `git`.

## 3.2 Создание SSH ключа

Создаем ключ для дальнейшей идентификации пользователя на сервере репозитория. Делаем это с помощью команды `ssh-keygen -C`, указывая имя пользователя и его почту, которые были заданы выше. В итоге ключ сохраняется в каталоге `~/.ssh/` (рис. 3.6).

```
[dpstrizhov@localhost-live ~]$ ssh-keygen -C "Strizhov Dmitry <1132236054@pfur.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dpstrizhov/.ssh/id_rsa):
Created directory '/home/dpstrizhov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

Рис. 3.6: Создание SSH-ключа

Копируем данный ключ (рис. 3.7) для дальнейшей его загрузки на github. Загружаем на github (рис. 3.8).

```
[dpstrizhov@localhost-live ~]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
[dpstrizhov@localhost-live ~]$
```

Рис. 3.7: Копирование ключа

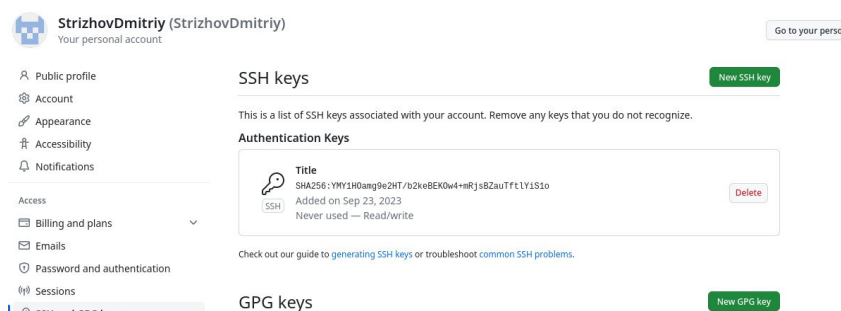


Рис. 3.8: Созданный ключ на github

### 3.3 Сознание рабочего пространства и репозитория курса на основе шаблона

Создаем каталог «Архитектура компьютеров» согласно указанной иерархии, делаю это с помощью команды mkdir и опции -p (рис. 3.9).

```
[dpstrizhov@localhost-live ~]$ mkdir -p ~/work/study/2023-2024/«Архитектура компьютера»
[dpstrizhov@localhost-live ~]$
```

Рис. 3.9: Создание каталога «Архитектура компьютера»

### 3.4 Сознание репозитория курса на основе шаблона

Перехожу на страницу репозитория с шаблоном курса. Выбираю Use this template (рис. 3.10).

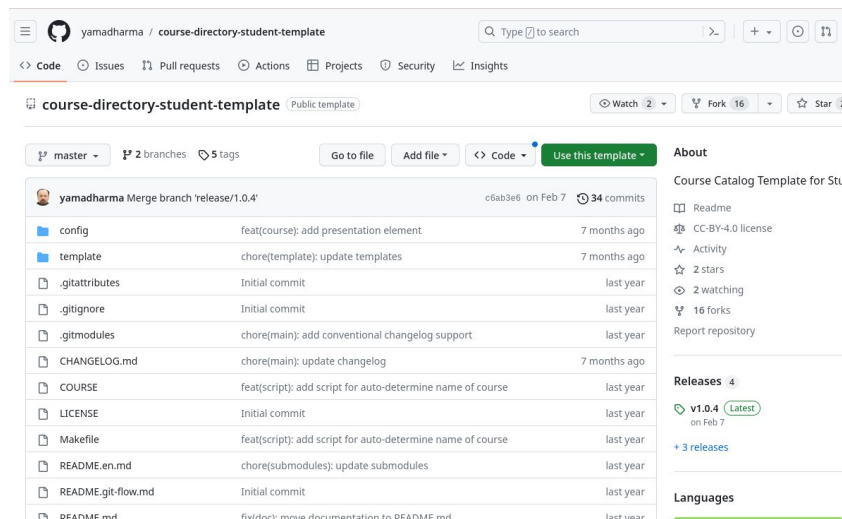


Рис. 3.10: Шаблон курса

Открываю терминал и перехожу в каталог курса с помощью команды `cd` (рис. 3.11).

```
[dpstrizhov@localhost-live ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"
[dpstrizhov@localhost-live Архитектура компьютера]$
```

Рис. 3.11: Каталог курса

Копирую репозиторий с github, используя команду `git clone --recursive` (рис. 3.12).

```
[dpstrizhov@localhost-live Архитектура компьютера]$ git clone --recursive git@github.com:StrizhovDmitriy/study_2023-2024_arch-pc.git arch-pc
Клонирование в «arch-pc»...
```

Рис. 3.12: Клонирование репозитория

## 3.5 Настройка каталога курса

Перехожу в каталог курса (рис. 3.13).

```
[dpstrizhov@localhost-live Архитектура компьютера]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
[dpstrizhov@localhost-live arch-pc]$
```

Рис. 3.13: Переход в каталог arch-pc



Удаляем лишние файлы (рис. 3.14).

```
[dpstrizhov@localhost-live arch-pc]$ rm package.json
[dpstrizhov@localhost-live arch-pc]$
```

Рис. 3.14: Удаление лишних файлов

Создаем необходимые каталоги (рис. 3.15).

```
[dpstrizhov@localhost-live arch-pc]$ echo arch-pc > COURSE
[dpstrizhov@localhost-live arch-pc]$ make
[dpstrizhov@localhost-live arch-pc]$
```

Рис. 3.15: Создание необходимых каталогов

Добавляем файлы на сервер. Сочетание `git add .` и `git commit -am` сохраняет все изменения в каталоге, `git push` отправляет изменения на сервер.(рис. 3.16, рис. 3.17).

```
[dpstrizhov@localhost-live arch-pc]$ git add
[dpstrizhov@localhost-live arch-pc]$ git commit -am 'feat(main): make course structure'
[master b67bd48] feat(main): make course structure
109 files changed, 54726 insertions(+), 1 deletion(-)
create mode 100644 COURSE
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/placement_600_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_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.py
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
```

Рис. 3.16: Добавление файлов на сервер

```
create mode 100644 presentation/README.ru.md
create mode 100644 presentation/presentation/Makefile
create mode 100644 presentation/presentation/image/kulyabov.jpg
create mode 100644 presentation/presentation/presentation.md
create mode 100644 presentation/report/Makefile
create mode 100644 presentation/report/bib/cite.bib
create mode 100644 presentation/report/image/placeimg_800_600_tech.jpg
create mode 100644 presentation/report/pandoc/csl/gost-r-7-8-5-2008-numeric.csl
create mode 100755 presentation/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_fignos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_secnos.py
create mode 100755 presentation/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/_init_.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/cors.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/main.py
create mode 100644 presentation/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 presentation/report/report.md
[dpstrizhov@localhost-live arch-pc]$ git push
Перечисление объектов: 100% (38/38), готово.
Подсчет объектов: 100% (38/38), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (29/29), готово.
Запись объектов: 100% (36/36), 342.18 Киб | 2.43 Миб/с, готово.
Всего 36 (изменений 4), повторно использовано 1 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:StrizhovDmitriy/study_2023-2024_arh-pc.git
 b8c397b..b67bd48 master -> master
[dpstrizhov@localhost-live arch-pc]$
```

Рис. 3.17: Отправка изменений на сервер

Проверяем иерархию файлов на github (рис. 3.18, рис. 3.19).

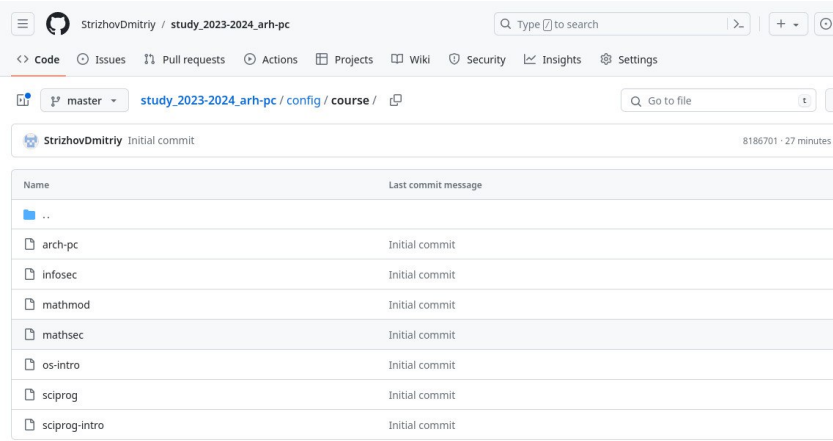


Рис. 3.18: Репозиторий на github

StrizhovDmitriy feat(main): make course structure		b67bd48 · 2 minute
Name	Last commit message	
..		
lab01	feat(main): make course structure	
lab02	feat(main): make course structure	
lab03	feat(main): make course structure	
lab04	feat(main): make course structure	
lab05	feat(main): make course structure	
lab06	feat(main): make course structure	
lab07	feat(main): make course structure	
lab08	feat(main): make course structure	
lab09	feat(main): make course structure	
lab10	feat(main): make course structure	
lab11	feat(main): make course structure	
README.md	feat(main): make course structure	

Рис. 3.19: Каталог labs

Убеждаемся, что всё сделано правильно.

## 3.6 Выполнение заданий для самостоятельной работы

№1 Создаем отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report). Делаем это с помощью команды touch, предварительно перейдя в подкаталог report каталога lab02 (с помощью команды cd). В итоге сначала переходим в каталог arch-pc, используя cd .., а затем cd labs/lab02/report. Далее используем touch.

№2 Копируем первую лабораторную работу в подкаталог report каталога lab01 (рис. 3.20), предварительно узнав путь к файлу с отчетом по выполнению первой лабораторной работы с помощью команды ls(ряда её использования).

```
[dpstrizhov@localhost-live lab01]$ cp ~/Загрузки/001_Стриков_отчет.pdf /home/dpstrizhov/work/study/2023-2024/'Архитектура компьютера'/arch-pc
[dpstrizhov@localhost-live lab01]$
```

Рис. 3.20: Копирование отчета по первой лабораторной работы в каталог lab01/report

№3 Загружаем файлы на github (рис. 3.21).

```
[dpstrizhov@localhost-live arch-pc]$ git add .
[dpstrizhov@localhost-live arch-pc]$ git commit -am 'feat(main): make course structure'
[master b8c397b] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
[dpstrizhov@localhost-live arch-pc]$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 298 байтов | 298.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:StrizhovOmitriy/study_2023-2024_arh-pc.git
8186701..b8c397b master -> master
[dpstrizhov@localhost-live arch-pc]$
```

Рис. 3.21: Отправка файлов

## 4 Выводы

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

## Список литературы

Git за полчаса: руководство для начинающих. Получено из proglib:  
<https://proglib.io/p/git-for-half-an-hour?ysclid=lmz0qy2u6q207353842>