

Отчёт по лабораторной работе 2

Архитектура компьютеров

Сулани Али НКАбд-06-24

Содержание

1	Цель работы	5
2	Теоретическое введение	6
2.1	Системы контроля версий. Общие понятия	6
3	Выполнение лабораторной работы	8
3.1	Настройка github	8
3.2	Создание репозитория курса на основе шаблона	9
3.3	Настройка каталога курса	10
3.4	Задание для самостоятельной работы	12
4	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	настраиваю Git	8
3.2	Генерирую SSH-ключи	9
3.3	Добавляю ключ на GitHub	9
3.4	Репозиторий-шаблон	10
3.5	Копирование шаблона	10
3.6	Клонирование репозитория	11
3.7	Подготовка папок	11
3.8	Отправка изменений в гитхаб	12
3.9	Отправка изменений в гитхаб	12
3.10	Обзор репозитория	13

Список таблиц

1 Цель работы

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

2 Теоретическое введение

2.1 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.

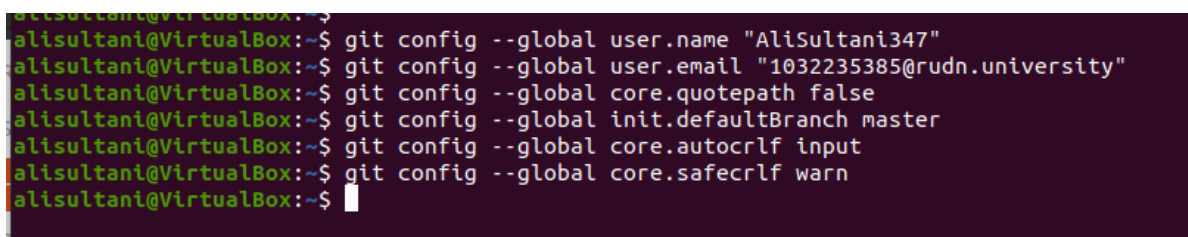
Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких

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.

3 Выполнение лабораторной работы

3.1 Настройка github

Сначала настраиваю Git на своем компьютере. Создаю пользователя в системе Git и указываю параметры — имя и email, чтобы мои действия были правильно подписаны. (рис. 3.1)

A screenshot of a terminal window with a dark purple background. The prompt is 'alisultani@VirtualBox:~\$'. The user enters a series of 'git config' commands to set global Git settings. The commands are: 'git config --global user.name "AliSultani347"', 'git config --global user.email "1032235385@rudn.university"', 'git config --global core.quotepath false', 'git config --global init.defaultBranch master', 'git config --global core.autocrlf input', and 'git config --global core.safecrlf warn'. The prompt returns after each command.

```
alisultani@VirtualBox:~$ git config --global user.name "AliSultani347"
alisultani@VirtualBox:~$ git config --global user.email "1032235385@rudn.university"
alisultani@VirtualBox:~$ git config --global core.quotepath false
alisultani@VirtualBox:~$ git config --global init.defaultBranch master
alisultani@VirtualBox:~$ git config --global core.autocrlf input
alisultani@VirtualBox:~$ git config --global core.safecrlf warn
alisultani@VirtualBox:~$
```

Рис. 3.1: настраиваю Git

Генерирую SSH-ключи, которые нужны для идентификации на GitHub. (рис. 3.2) Сохраняю их на компьютере и добавляю публичный ключ в свой профиль на GitHub для установления связи. (рис. 3.3)


```
alisultani@VirtualBox:~$
alisultani@VirtualBox:~$ ssh-keygen -C "AliSultani347 1032235385@rudn.university"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alisultani/.ssh/id_rsa): Created directory '/home/ali
sultani/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alisultani/.ssh/id_rsa
Your public key has been saved in /home/alisultani/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:L3wYj/sem3FiyiAudBicUAoqB33dUeIib6HP9eLH/EQ AliSultani347 1032235385@rudn.university
The key's randomart image is:
+----[RSA 3072]-----+
|++ . . .oo. |
|ooo . .... |
|+..o. o . |
|..o + o |
| .. o S E |
| o .+ o B . |
| . o. + *oX o |
| .. . + B+O |
| .. =+ =.. |
+----[SHA256]-----+
alisultani@VirtualBox:~$
```

Рис. 3.2: Генерирую SSH-ключи

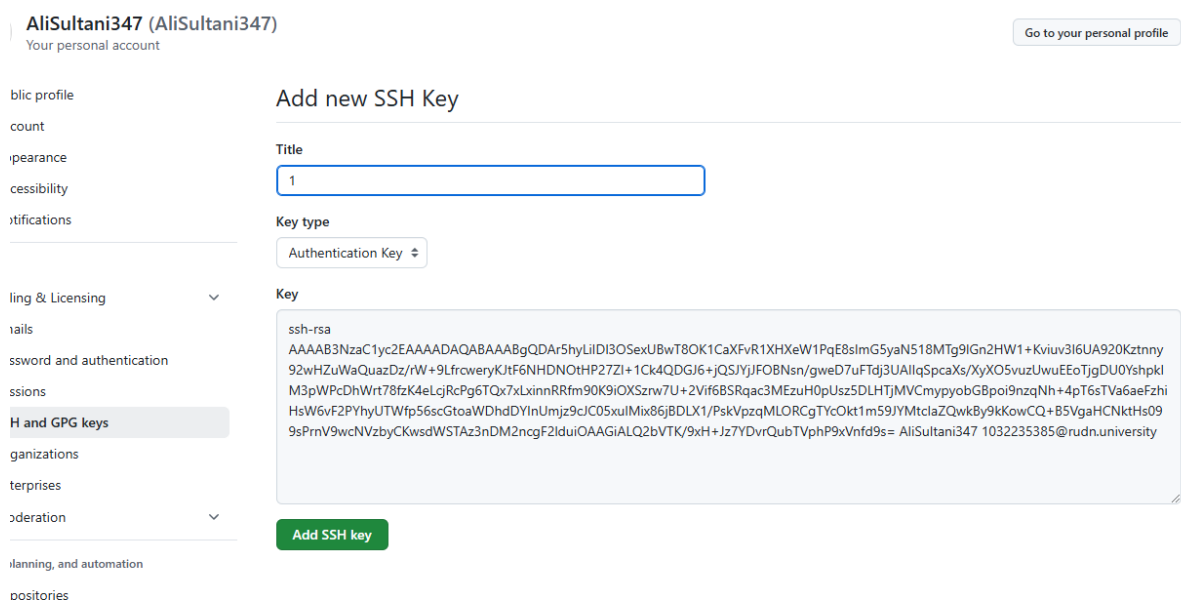


Рис. 3.3: Добавляю ключ на GitHub

3.2 Создание репозитория курса на основе шаблона

Нахожу нужный репозиторий-шаблон (рис. 3.4) и делаю из него копию для работы (рис. 3.5).

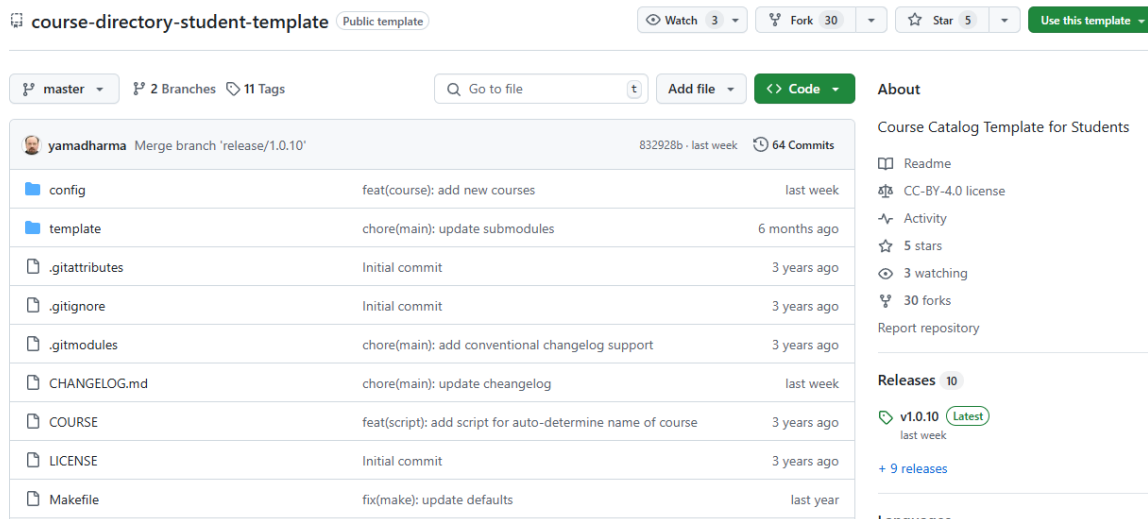


Рис. 3.4: Репозиторий-шаблон

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

yamadharm/course-directory-student-template

Start your repository with a template repository's contents.

☐ Include all branches

Copy all branches from yamadharm/course-directory-student-template and not just the default branch.

Owner *

AliSultani347

Repository name *

arch-pc

arch-pc is available.

Great repository names are short and memorable. Need inspiration? How about [supreme-funicular](#) ?

Description (optional)

Рис. 3.5: Копирование шаблона

3.3 Настройка каталога курса

Создаю рабочую директорию на своем компьютере для хранения файлов проекта. Клонировать репозиторий с GitHub в эту директорию, чтобы работать с

файлами локально. (рис. 3.6) (рис. 3.7)

```
and the repository exists.
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера$ git clone --recursive git@github.com:AliSultani347/arch-pc.git
Cloning into 'arch-pc'...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 36 (delta 1), reused 21 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (36/36), 19.38 KiB | 9.69 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.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/alisultani/work/study/2024-2025/Архитектура компьютера/arch-pc/template/presentation'...
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)
Receiving objects: 100% (111/111), 102.17 KiB | 879.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
Cloning into '/home/alisultani/work/study/2024-2025/Архитектура компьютера/arch-pc/template/report'...
remote: Enumerating objects: 142, done.
```

Рис. 3.6: Клонирование репозитория

```
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера$
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера$ cd ~/work/study/2024-2025/Архитектура компьютера/arch-pc
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ rm package.json
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ make prepare
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ ls
CHANGELOG.md  COURSE  LICENSE  prepare  README.en.md  README.md
config        labs   Makefile  presentation  README.git-flow.md  template
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 3.7: Подготовка папок

После подготовки структуры добавляю изменения в локальный репозиторий и отправляю их на GitHub командой push. (рис. 3.8)

```

create mode 100044 presentation/report/pandoc/est/gost-17403-2008-numerical-est
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/core.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
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 4 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 341.28 KiB | 2.25 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:AliSultani347/arch-pc.git
   ea6b067..279f14d  master -> master
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$

```

Рис. 3.8: Отправка изменений в гитхаб

3.4 Задание для самостоятельной работы


Загружаю отчёты по выполненным работам в соответствующие папки на GitHub, обновляя репозиторий по мере необходимости. (рис. 3.9) (рис. 3.10)



```

alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): add lab01'
[master 650a9b0] feat(main): add lab01
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 labs/lab01/report/Л01_Султани Аии_отчет.pdf.pdf
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 557.92 KiB | 3.62 MiB/s, done.
Total 6 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:AliSultani347/arch-pc.git
   279f14d..650a9b0  master -> master
alisultani@VirtualBox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$

```

Рис. 3.9: Отправка изменений в гитхаб

arch-pc / labs / lab01 / report / 

 AliSultani347 feat(main): add lab01 650a9b0 · now  History








Name	Last commit message	Last commit date
 ..		
 bib	feat(main): make course structure	1 minute ago
 image	feat(main): make course structure	1 minute ago
 pandoc	feat(main): make course structure	1 minute ago
 Makefile	feat(main): make course structure	1 minute ago
 report.md	feat(main): make course structure	1 minute ago
 Л01_Султани Аии_отчет.pdf.pdf	feat(main): add lab01	now

Рис. 3.10: Обзор репозитория

4 Выводы

В ходе выполнения работы изучили работу с GitHub.

Список литературы

1. Архитектура ЭВМ - Материалы курса
2. Markdown Документация
3. GitHub Документация