

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных
наук Кафедра прикладной информатики и теории
вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2
дисциплина: Архитектура компьютера

Студент: Метвалли Ахмед Фарг Набеев

Группа: НПИбд-02-23

МОСКВА 2023г

Содержание

1. Цель работы

2. Теоретическое Введение

3. Выполнение лабораторной работы

4. Вывод

1. Цель работы

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

2. 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. 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. Создайте учётную запись на сайте <https://github.com/> и заполните основные данные.



Рис 3.1.1. Профиль github.

До выполнения данной лабораторной работы я уже имел профиль в github и просто зашел в него.

3.2 Базовая настройка git.

Сначала сделаем предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и email владельца репозитория

```
ahmad-farg@ahmadfarg-VirtualBox:~$ git config --global user.name "AhmadFarg96"  
ahmad-farg@ahmadfarg-VirtualBox:~$ git config --global user.email "a.farg22@icloud.com"
```

Рис 3.2.1. Предварительная конфигурация git.

Используя данные в лабораторной работе команды, я сделал предварительную конфигурацию на основе моего имени и моей почты Настроим utf-8 в выводе сообщений git.

```
ahmad-farg@ahmadfarg-VirtualBox:~$ git config --global core.quotepath false
```

Рис 3.2.2. Настройка utf-8.

Зададим имя начальной ветки (будем называть её master)

```
ahmad-farg@ahmadfarg-VirtualBox:~$ git config --global init.defaultBranch master
```

Рис 3.2.3. Дарим имя начальной ветки.

Параметр autocrlf:

```
ahmad-farg@ahmadfarg-VirtualBox:~$ git config --global core.autocrlf input
```

Рис 3.2.4. autocrlf.

Параметр safecrlf:

```
ahmad-farg@ahmadfarg-VirtualBox:~$ git config --global core.safecrlf warn
```

Рис 3.2.5. safecrlf.

3.3 Создание SSH ключа

Для последующей идентификации пользователя на сервере репозитория необходимо генерировать пару ключей (приватный и открытый): `ssh-keygen -C "Имя Фамилия <work@mail>"`

Рис 3.3.1. Генерация ключей.

С помощью данной в лабораторной работе команды я сгенерировал пару ключей

Далее необходимо загрузить сгенерированный открытый ключ. Скопировав из локальной консоли ключ в буфер обмена.

```
ahmad-farg@ahmadfarg-VirtualBox:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCKPoTj6UDgWvQMCHPMLAJAVo6oj9J7dE8Ve51T2g+vbQCzcZsQ4Qct7aWL
DESfhMSTccYja0eU0pyY3YTVm7VfV599KF0RZku44jaatjX0seLZpiy00KxBBzR7tobebVwu5fLmZCehiXCgp+ww0jnkHEFC
SVQU6rl61hreZ5FtpoT1BsqWF9RlmeDIBU+sOuUFGMKUNtuZH7QJenKwBNhJ6D+FhrX9oUmmhnTmKYYSgoF7L843bvdEJ7K
LliyNGK5Cgoau6S5mfjHohu2EphW2V10YxY2eY4rDK8WUuo8wLuoSxCNaLAKnEXlgK3J8+3XfxoquAlbqt2Ub34r7ihu0/M+
g8h5BG+smo+cnE841ug9Z7W5jPDGxcJR0vhNT69qw5Dd22f7nLDLVKbxbg0o5fywhrCuEYUJUxt5CrYR/NT1hdD9OdTFuV
G0r0m0sv9GpaALSM+vFLUCox0htTzhztwGngDU470vTJ1k6Z+61bId5GUX10+qmyteR4L50= AhmadFarg96 <a.farg22@i
cloud.com>
```

Рис 3.3.2. Вывод ключа.

Используя команду cat, я вывел в консоль сгенерированный мною ключ

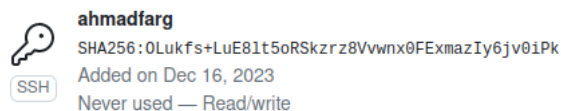


Рис 3.3.3. Загрузка ключа

в github. Следуя инструкциям в лабораторной работе, я загрузил ключ на github

3.4 Сознание рабочего пространства и репозитория курса на основе шаблона

Откройте терминал и создайте каталог для предмета «Архитектура компьютера»

```
ahmad-farg@ahmadfarg-VirtualBox:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
```

Рис 3.4.1. Создание ветки каталогов.

Используя команду mkdir с опцией -p, я создал цепочку каталогов.

3.5. Сознание репозитория курса на основе шаблона В открывшемся окне задайте имя репозитория (Repository name)

study_2023– 2024_arh-pc и создайте репозиторий (кнопка Create repository from template)

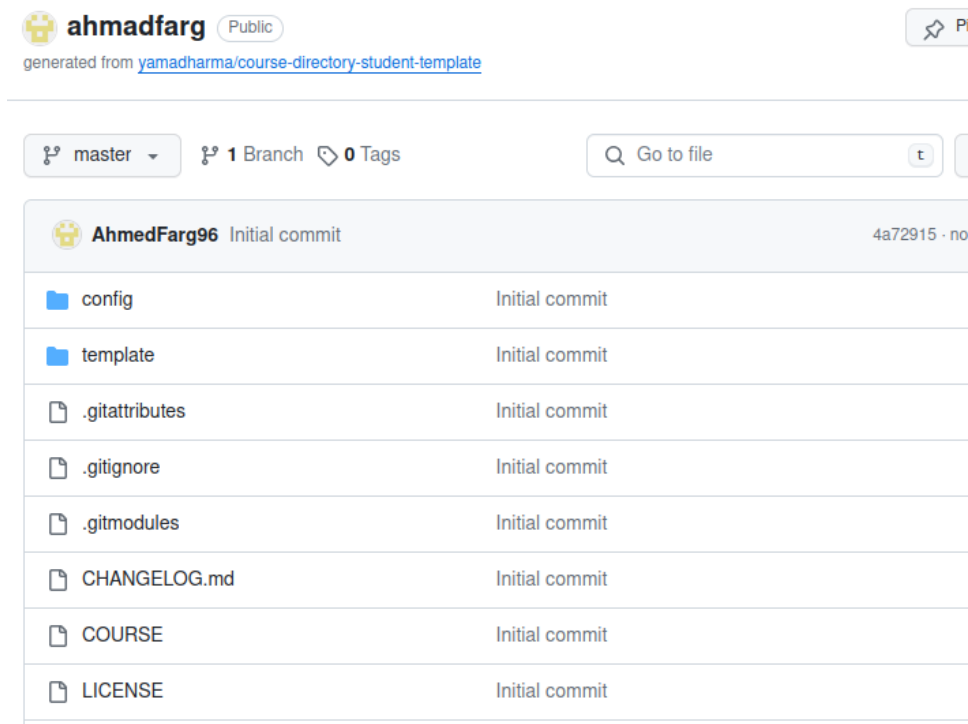


Рис 3.5.1. Создание репозитория по шаблону.

Следуя инструкциям в лабораторной работе, я создал репозиторий по данному мне шаблону

Откройте терминал и перейдите в каталог курса:

```
ahmad-farg@ahmadfarg-VirtualBox:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
```

Рис 3.5.2. Переход в каталог курса.

Клонируйте созданный репозиторий:

```

ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера$ git clone git@github.com:AhmedFarg96/ahmadfarg.git
Cloning into 'ahmadfarg'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 30 (delta 1), reused 17 (delta 0), pack-reused 0
Receiving objects: 100% (30/30), 17.76 KiB | 8.88 MiB/s, done.
Resolving deltas: 100% (1/1), done.

```

Рис 3.5.3. Клонирование репозитория.

Благодаря команде `git clone`, я клонировал репозиторий из github в мою систему

3.6 Настройка каталога курса

Перейдите в каталог курса:

```

ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера$ cd arch-pc/
ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$

```

Рис 3.6.1. Переход в каталог курса.

Удалите лишние файлы:

`rm package.json`

```

ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ make

```

Рис 3.6.2. Удаление лишнего файла.

Создайте необходимые каталоги:

`echo arch-pc > COURSE`

`Make`

```

ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc > COURSE
ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ make

```

Рис 3.6.3. Создание необходимых каталогов.

Благодаря команде `echo` я создал файл в моем древе

каталогов Отправьте файлы на сервер: `git add . git commit -am`

'feat(main): make course structure' git push.

```
ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master 371b81a] feat(main): make course structure
 7 files changed, 5 insertions(+), 14 deletions(-)
 create mode 100644 labs/README.md
 create mode 100644 labs/README.ru.md
 delete mode 100644 package.json
 create mode 100644 prepare
 create mode 100644 presentation/README.md
 create mode 100644 presentation/README.ru.md

ahmad-farg@ahmadfarg-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 3 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 710 bytes | 355.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:AhmedFarg96/ahmadfarg.git
 4a72915..371b81a master -> master
```

Рис 3.6.4. Отправка файлов в github.

Благодаря данным командам из лабораторной работы я синхронизировал файлы со средой github.

4. Вывод: В ходе выполнения лабораторной работы были изучены идеологии и применение средств контроля версий. Также были приобретены практические навыки по работе с системой git.