

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: Хамди Мохаммад Рагид Рибхи

Группа: НПИбд-02-23

МОСКВА 2023г

Содержание

1. Цель работы
2. Теоретическое введение
3. Выполнение лабораторной работы
4. Вывод

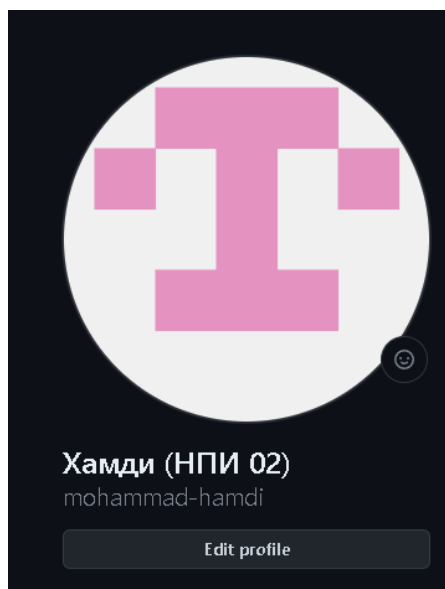
1. ***Цель работы*** Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

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

1. Настройка github

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



Профиль github

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

2. Базовая настройка git.

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

```
hamdi@hamdi-VirtualBox:~$ git config --global user.name "Хамди (НПИ 02)"
hamdi@hamdi-VirtualBox:~$ git config --global user.email "mohammedhamdi500@gmail.com"
```

3. 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

```
hamdi@hamdi-VirtualBox:~$ git config --global core.quotepath false
hamdi@hamdi-VirtualBox:~$
```

4. Настройка utf-8

Зададим имя начальной ветки (будем называть её master)

```
hamdi@hamdi-VirtualBox:~$ git config --global init.defaultbranch master
```

5. Дарим имя начальной ветки

Параметр autocrlf:

```
hamdi@hamdi-VirtualBox:~$ git config --global core.autocrlf input
```

6. autocrlf

Параметр safecrlf:

```
hamdi@hamdi-VirtualBox:~$ git config --global core.safecrlf warn
```

safecrlf

7. Создание SSH ключа

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

```
hamdi@hamdi-VirtualBox:~$ ssh-keygen -C "Хамди (НПИ 02) <mohammedhamdi500@gmail.com> "
```

Генерация ключей

С помощью данной в лабораторной работе команды, я сгенерировал пару ключей

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

```
hamdi@hamdi-VirtualBox:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCfCHjWPV9xR7
QVzRYCpha13NSbWPBd2omeNrgdWVOEc3L84YE8vtYoQ49QqtU
J6oBZQTjVTFLypnP3DiVZtPLpdiskVg/veklHQ3E/yDSXeI+J
wYXi0/4qLnd7ueCz/qzuG+9B6uCj0ieAV410idYWcpE9Iisuq
W40r+GmNm5w1N9bj+xThe28= Хамди (НПИ 02) <mohammed
```

2. Создание рабочего пространства и репозитория курса на основе шаблона

Откройте терминал и создайте каталог для предмета «Архитектура компьютера»

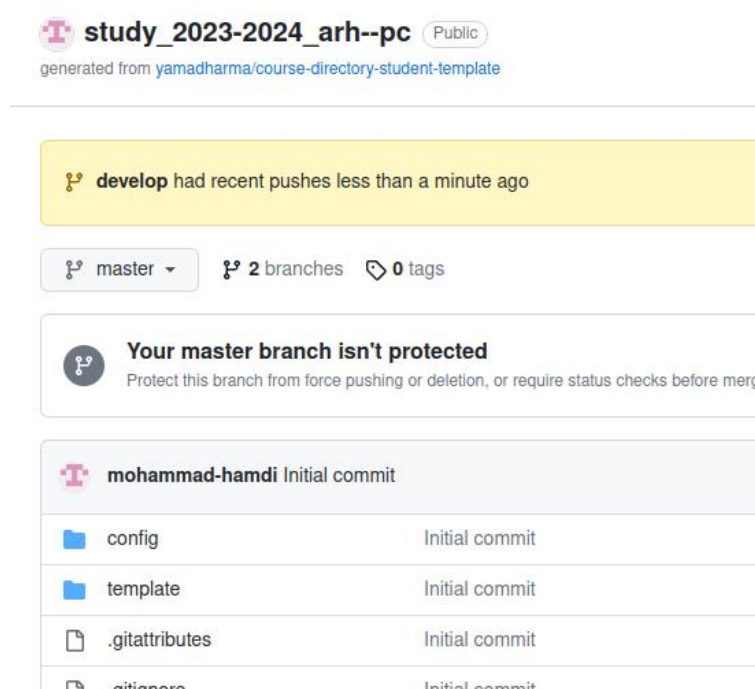
```
hamdi@hamdi-VirtualBox:~$ mkdir -p ~/work/study/2023-2024/"Архитектура Компьютера"
```

Создание ветки каталогов.

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

Создание 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)



Создание репозитория по шаблону.

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

Откройте терминал и перейдите в каталог курса:

```
hamdi@hamdi-VirtualBox:~$ cd work/study/2023-2024/Архитектура\ Компьютера/
```

Переход в каталог курса

Клонируйте созданный репозиторий:

```
hamdi@hamdi-VirtualBox:~/work/study/2023-2024/Архитектура Компьютера$ git clone --recursive  
-2024_arh--pc.git  
Cloning into 'study_2023-2024_arh--pc'...  
remote: Enumerating objects: 28, done.  
remote: Counting objects: 100% (28/28), done.  
remote: Compressing objects: 100% (27/27), done.  
remote: Total 28 (delta 1), reused 11 (delta 0), pack-reused 0  
Receiving objects: 100% (28/28), 17.51 KiB | 17.51 MiB/s, done.  
Resolving deltas: 100% (1/1), done.  
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markd  
emplate/presentation)
```

Настройка каталога курса

Перейдите в каталог курса:

```
hamdi@hamdi-VirtualBox:~/work/study/2023-2024/Архитектура Компьютера$ cd arch-pc/
```

Переход в каталог курса

Удалите лишние файлы:

```
rm package.json
```

```
hamdi@hamdi-VirtualBox:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ rm package.json
```

Удаление лишнего файла

Создайте необходимые каталоги:

```
echo arch-pc > COURSE
```

Make

```
hamdi@hamdi-VirtualBox:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ echo arch-pc > COURSE  
hamdi@hamdi-VirtualBox:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ make
```

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
```

```

hamdi@hamdi-VirtualBox:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ git add .
hamdi@hamdi-VirtualBox:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master 015fec4] feat(main): make course structure
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






hamdi@hamdi-VirtualBox:~/work/study/2023-2024/Архитектура Компьютера/arch-pc$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 8 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.16 KiB | 2.83 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0

```

Отправка файлов в github

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

Проверьте правильность создания иерархии рабочего пространства в локальном репозитории и на странице github

	mohammad-hamdi feat(main): make course structure	015fec4 2 minutes ago	 2 commits
	config	Initial commit	43 minutes ago
	labs	feat(main): make course structure	2 minutes ago
	presentation	feat(main): make course structure	2 minutes ago