Отчёт по лабораторной работе №2

Простейший вариант

Югай Александр Витальевич

Содержание

# 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 хранили- ща 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 кон- фликтов, 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

**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.

**Базовая настройка git.**

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

Предварительная конфигурация git

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

Настройка utf-8

Настройка utf-8

Зададим имя начальной ветки (будем называть её master)

Дарим имя начальной ветки

Дарим имя начальной ветки

Параметр autocrlf:

autocrlf

autocrlf

Параметр safecrlf:

safecrlf

safecrlf

**Создание SSH ключа**

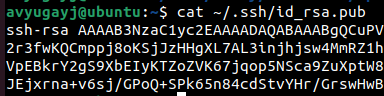
Для последующей идентификации пользователя на сервере репозиториев необходимо генерировать пару ключей (приватный и открытый): ssh-keygen -C “Имя Фамилия [work@mail](mailto:work@mail)”

Генерация ключей

Генерация ключей

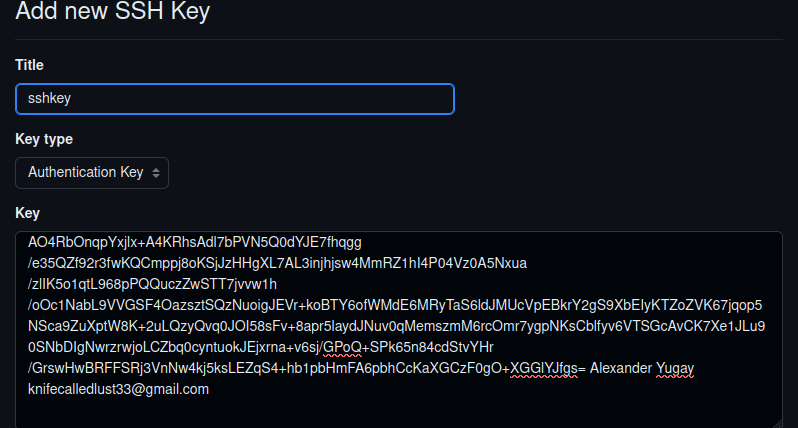
С помощью данной в лабораторной работе команды, я сгенерировал пару ключей

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



Вывод ключа

Используя команду cat, я вывел в консоль сгенерированый мною ключ



Загрузка ключа в github

Следуя инструкциям в лабораторной работе, я загрузил ключ на github

**Сознание рабочего пространства и репозитория курса на основе шаблона**

Откройте терминал и создайте каталог для предмета «Архитектура компьютера»

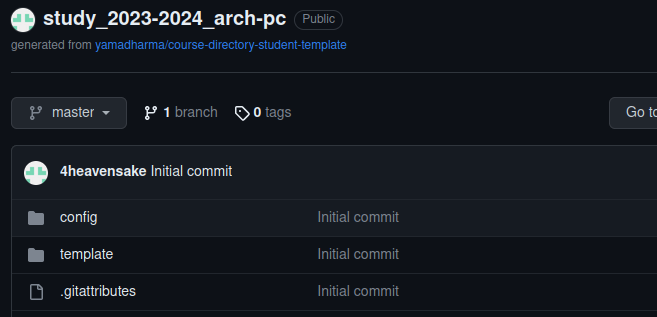
Создание ветки каталогов

Создание ветки каталогов

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

**Сознание репозитория курса на основе шаблон**

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.

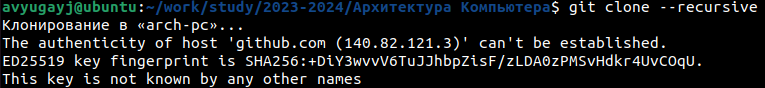
Следуя инструкциям в лабораторной работе, я создал репозиторий по данному мне шаблону

Откройте терминал и перейдите в каталог курса:

Переход в каталог курса

Переход в каталог курса

Клонируйте созданный репозиторий:



Клонирование репозитория

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

**Настройка каталога курса**

Перейдите в каталог курса:

Переход в каталог курса

Переход в каталог курса

Удалите лишние файлы:

rm package.json

Удаление лишнего файла

Удаление лишнего файла

Создайте необходимые каталоги:

echo arch-pc > COURSE

make

Создание необходимых каталогов

Создание необходимых каталогов

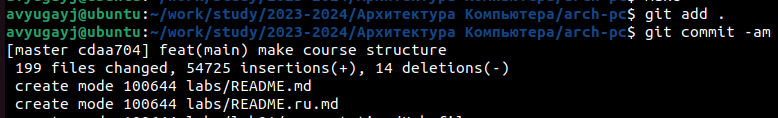
Благодаря команде echo я создал файл в моем древе каталогов

Отправьте файлы на сервер:

git add .

git commit -am ‘feat(main): make course structure’

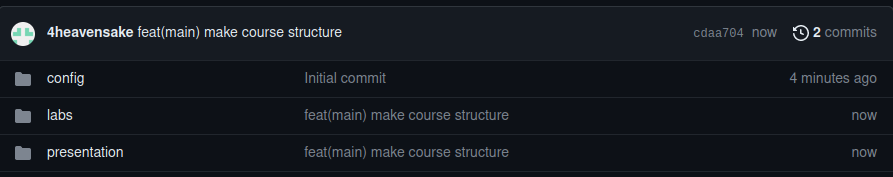
git push



Отправка файлов в github

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

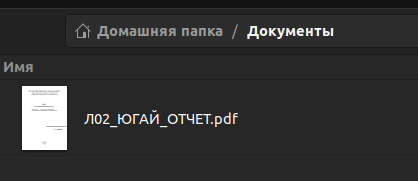
Проверьте правильность создания иерархии рабочего пространства в локальном репозитории и на странице github



Проверка правильности задания на github

**Задания для самостоятельной работы.**

Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report)



Создание отчета в папке “Документы”

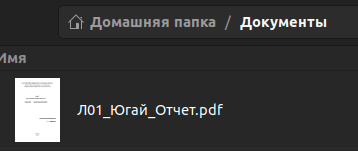
Копирование отчета из “Документы” в нужный каталог

Копирование отчета из “Документы” в нужный каталог

Проверка выполненных действий.

Проверка выполненных действий.

Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.



Перенос первого отчета в каталог “Документы”

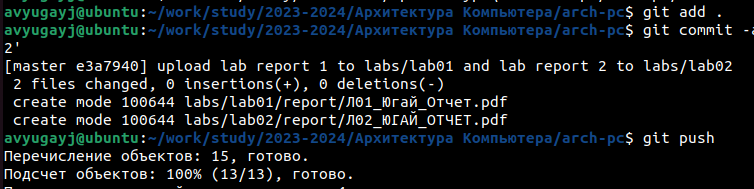
Копирование отчета в нужный каталог

Копирование отчета в нужный каталог

Проверка выполненных действий.

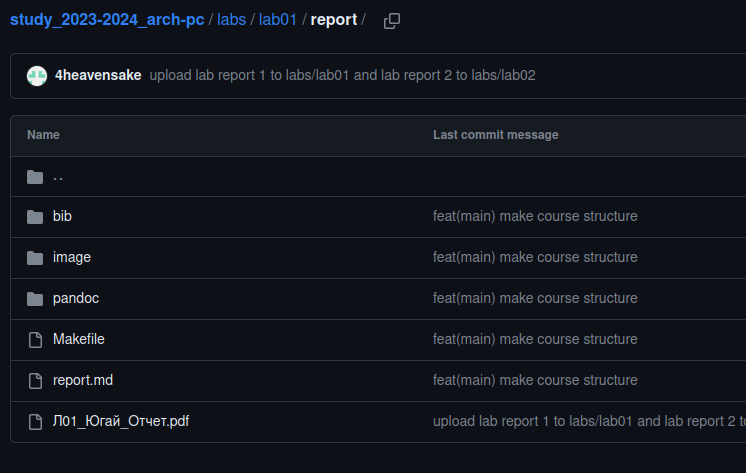
Проверка выполненных действий.

Загрузите файлы на github



Загрузка файлов в github

Благодаря уже известным мне командам, я синхронизировал все изменения в системе со средой github



Проверка правильности загрузки.

# 4 Выводы

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