Лабараторная работа №2

Отчет

Славинский Владислав Вадимович

Содержание

# 1 Цель работы

Изучить Идеологию и применение средств контроля версий. Освоить умения по работе с git.

# 2 Задание

Установка git Установка gh Базовая настройка git Создание ключей ssh и pgp Настройка github Добавление pgp ключа в github Настройка автоматический подписей коммитов git Настройка gh Сознание репозитория курса на основе шаблона Настройка каталога курса Контрольные вопросы

# 3 Выполнение лабораторной работы

В начале установим git (рис. 1)



Рис. 1: Установка git

Установим gh (рис. 2)

Установка gh

Рис. 2: Установка gh

Задал имя и email своего репозитория (рис. 3)



Рис. 3: Данные репозитория

Настройка utf-8, задал имя начальной ветки, ввел параметр autocrlf и safecrlf (рис. 4)



Рис. 4: utf-8,autocrlf,safecrlf

Создал ключи ssh (рис. 5)

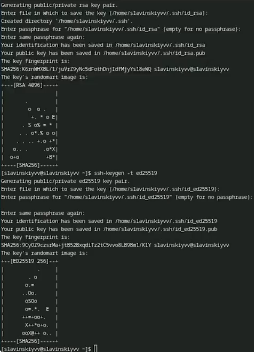


Рис. 5: Создание ключей

Добавление ssh ключа на git (рис. 6)

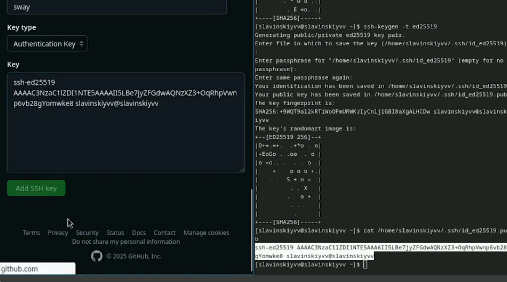


Рис. 6: Добавление ключа

Настройка автоматических подписей коммитов git (рис. 7)

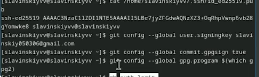


Рис. 7: Подписи

Настройка gh (рис. 8)

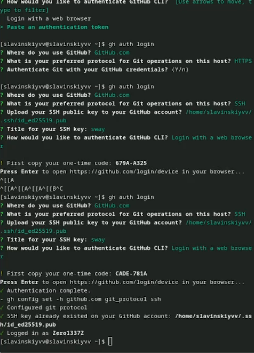


Рис. 8: Настройка gh

Произвел операции над сознанием рабочего пространства (рис. 9)

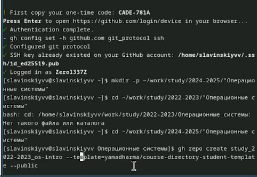


Рис. 9: Сознание рабочего пространства

Настроил каталог курса, удалил лишние файлы, создал необходимые каталоги и отправил их на сервер (рис. 10)

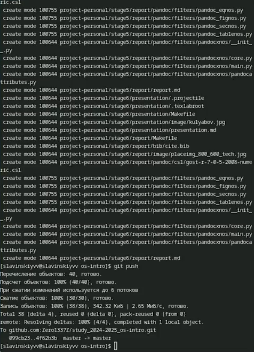


Рис. 10: Настройка каталога курса

# 4 Выводы

В ходе выполнения лабораторной работы был установлен git, его настройка, были созданы ключи для авторизации и подписи. Был создан репозиторий.

# 5 Ответы на контрольные вопросы

1. VCS-это инструменты для отслеживания изменений в файлах и управления проектами, позволяющие сохранять версии и координировать работу.
2. Хранилище-место хранения файлов и их истории. Commit-сохранение изменений в хранилище. История-последовательность всех коммитов, отражающая изменения в проекте. Рабочая копия-локальная версия файлов, с которой работает разработчик.
3. Централизованные: имеют одно центральное хранилище, к которому подключаются все пользователи. Децентрализованные каждый разработчик имеет полную копию хранилища, включая всю историю.
4. Создание хранилища. Внесение изменений в рабочую копию. Выполнение команды commit для сохранения изменений. Просмотр истории изменений.
5. Клонирование удаленного репозитория. Внесение изменений и создание коммитов. Синхронизация с удаленным репозиторием (pull/push). Разрешение конфликтов, если они возникают.
6. Отслеживание изменений в коде. Восстановление предыдущих версий. Совместная работа над проектами. Управление ветвями.
7. git init: инициализация нового репозитория. git clone: клонирование удаленного репозитория. git add: добавление изменений в индекс. git commit: сохранение изменений в хранилище. git push: отправка изменений в удаленный репозиторий. git pull: получение изменений из удаленного репозитория. git branch: управление ветвями. git merge: слияние ветвей.
8. Локальный репозиторий: git init, git add ., git commit -m “Initial commit”. Удаленный репозиторий: git clone , git push origin main.
9. Ветви позволяют создавать параллельные линии разработки, что упрощает работу над новыми функциями или исправлениями, не затрагивая основную ветвь
10. Файлы можно игнорировать с помощью файла .gitignore, чтобы исключить их из коммитов (например, временные файлы, конфигурации среды), что помогает поддерживать чистоту репозитория.