Лабораторная работа №2

Дисциплина: Операционные системы

Газизянов Владислав Альбертович

Содержание

# 1 Цель работы

Изучение работы и назначение системы контроля версий git приобретение навыков по работе с ней

# 2 Выполнение лабораторной работы

Устанавливаем git и fh

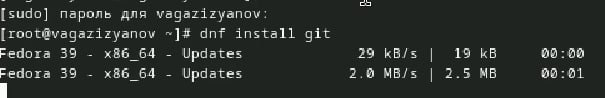


Рис. 1: Git

Задаём имя и email владельца репозитория, настраиваем utf-8 и параметры autocrlf and safeclrf

Имя

Рис. 2: Имя

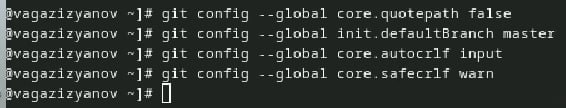


Рис. 3: Параметры

Создаём ключ ssh и pgp

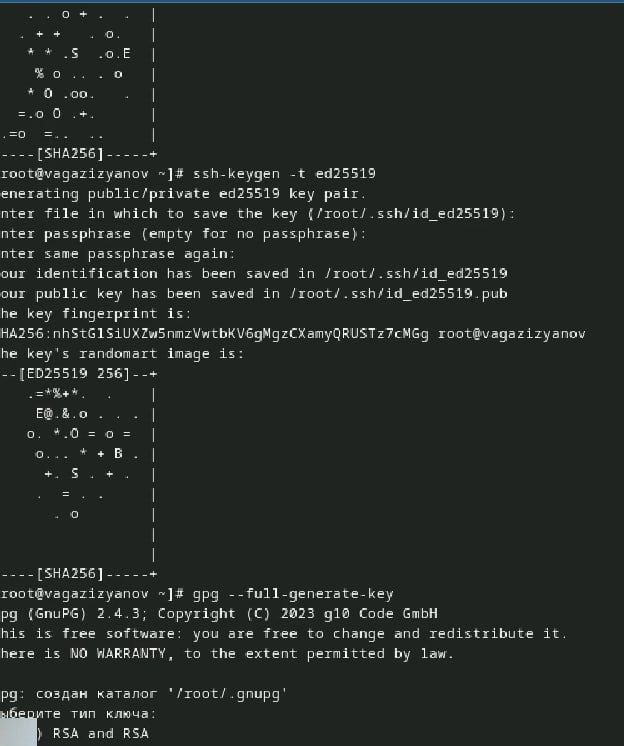


Рис. 4: Ключи

Авторизирруемся на GitHub и добавим созаднные ключи, скопировав их отпечатки

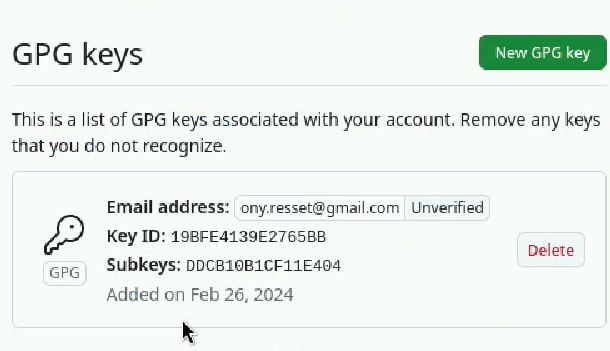


Рис. 5: Добавление ключей

Настройка автоматической подписи коммитов

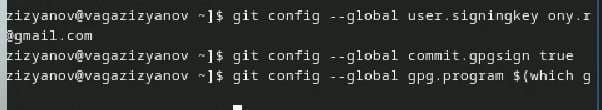


Рис. 6: Автоматические подписи

Авторизируемся с помощью команды gh login auth и, создав и перейдяв нужные каталоги создадим репозиторий на сонове шаблона, а затем клонируем его себе

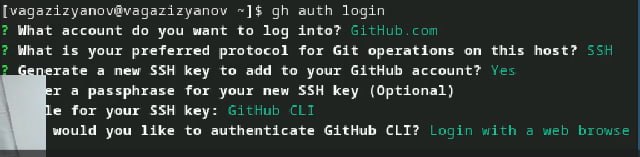


Рис. 7: Авторизация

Далее удаляем все лишние файлы и создаём необходимые каталоги

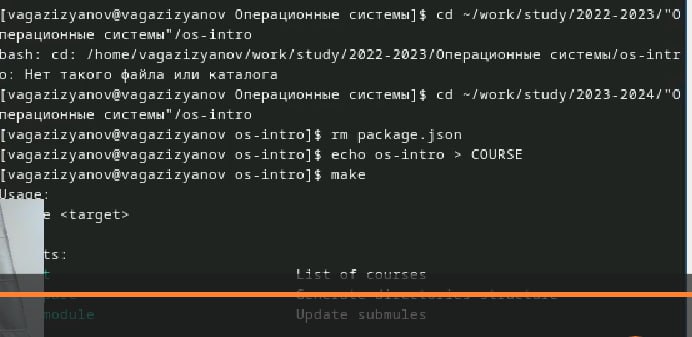


Рис. 8: Удаление файлов и создание каталогов

Отправляем все изменения на github

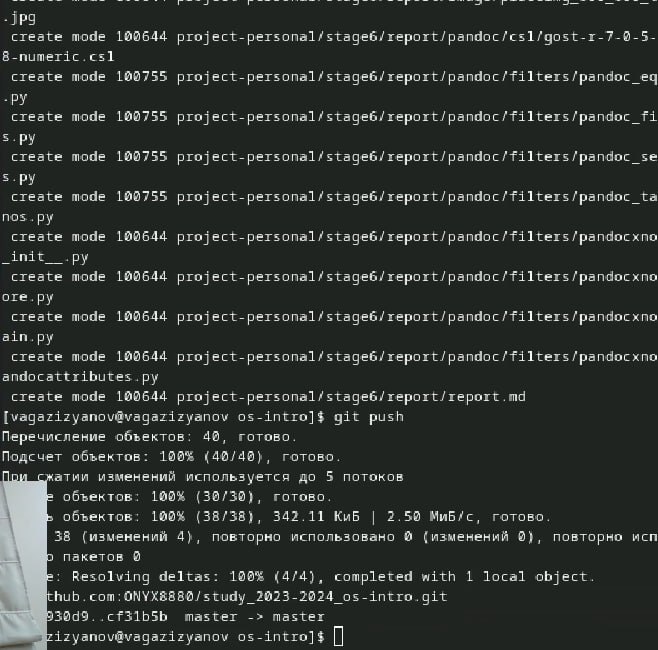


Рис. 9: Отправка

## 2.1 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Они применяются при работе нескольких человек с одним проектом. При внесении изменений позволяют фиксировать, совмещать и возвращать из- менения разных людей
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище - место, где находятся данные(файлы, коды и тд) Commit - команда, для сохранения изменений История - информация о предыдущих изменениях Рабочая копия - одна из версий проекта, с которй ведется работа(= текущая/основная)
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. Централизованные системы предполагают наличие единого репозитория для хранения данных(CVC, Subversion) В децентрализованных системах центральный репозитроий не обязателен(Git, Bazaar)
4. Опишите действия с VCS при единоличной работе с хранилищем. абота происходит на своем компьютере, сначала обновляются данные, в конце они размещаются в центральном репозитории
5. Опишите порядок работы с общим хранилищем VCS. Для идентификации на сервере необходмы ключи и затем создание репозитория, только затем можно работать на локальной машиной. Также в конце изменения добавляются на сервер
6. Каковы основные задачи, решаемые инструментальным средством git? хранение информации о всех изменениях обеспечение удобства командной работы
7. Назовите и дайте краткую характеристику командам git. git init Получение обновлений (изменений) текущего дерева из центрального репозитория: git pull Отправка всех произведённых изменений локального дерева в центральный репозиторий: git push Просмотр списка изменённых файлов в текущей директории: git status Просмотр текущих изменений: git diff Сохранение текущих изменений: добавить все изменённые и/или созданные файлы и/или каталоги: git add . добавить конкретные изменённые и/или созданные файлы и/или каталоги: git rm имена\_файлов Сохранение добавленных изменений: сохранить все добавленные изменения и все изменённые файлы: git commit -am ‘Описание коммита’ сохранить добавленные изменения с внесением комментария через встроенный редактор: git checkout -b имя\_ветки переключение на некоторую ветку: git checkout имя\_ветки (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) отправка изменений конкретной ветки в центральный репозиторий: git push origin имя\_ветки слияние ветки с текущим деревом: git merge –no-ff имя\_ветки Удаление ветки: удаление локальной уже слитой с основным деревом ветки: git branch -d имя\_ветки принудительное удаление локальной ветки: git branch -D имя\_ветки удаление ветки с центрального репозитория: git push origin :имя\_ветки
8. Приведите примеры использования при работе с локальным и удалённым репозиториями. локальный репозиторий - работа со своими файлами удаленный репозиторий - совместная работа, общий проект
9. Что такое и зачем могут быть нужны ветви (branches)? Ветви это пути к отдельным состовдющим (отделы проекта),они дают возможность вносить изменения только в часть проекта и не трогать все вышестоящее
10. Как и зачем можно игнорировать некоторые файлы при commit? Игнорировать файлы можно с помощью .gitignore. Нужно это, например, при наличии ненужных(лишних или созданных автоматически) файлов.

# 3 Выводы

В ходе работы была освоена работа с системой контроля версий, был установлен git, проведена авторизация, заданы базовые настройки, создаись ключи, клонировался репозиторий и так даллее.

# Список литературы