Лабораторная работа №2

Дисциплина: Операционные системы

Жибицкая Евгения Дмитриевна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Для начала установим git и fh (рис. 1).

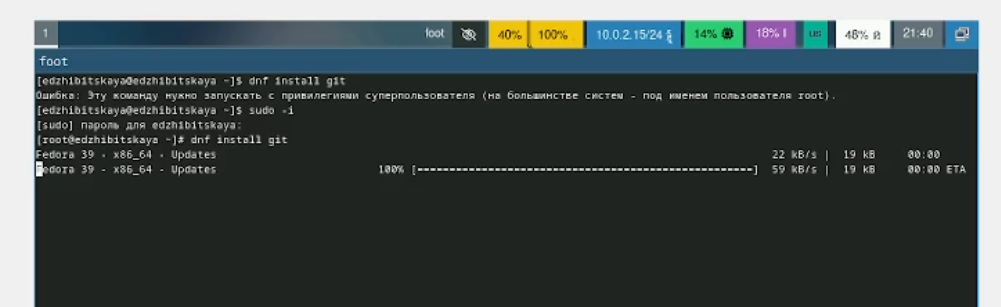


Рис. 1: Установка git

После этого необходимо указать имя и почту (рис. 2), также настроить utf-8 и параметры autocrlf, safecrlf (рис. 3).

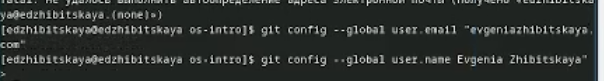


Рис. 2: Имя пользоватя и почта

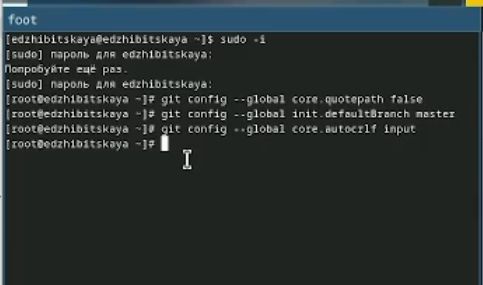


Рис. 3: Настройка параметров

Далее создаем ключ ssh и ключ pgp (рис. 4).

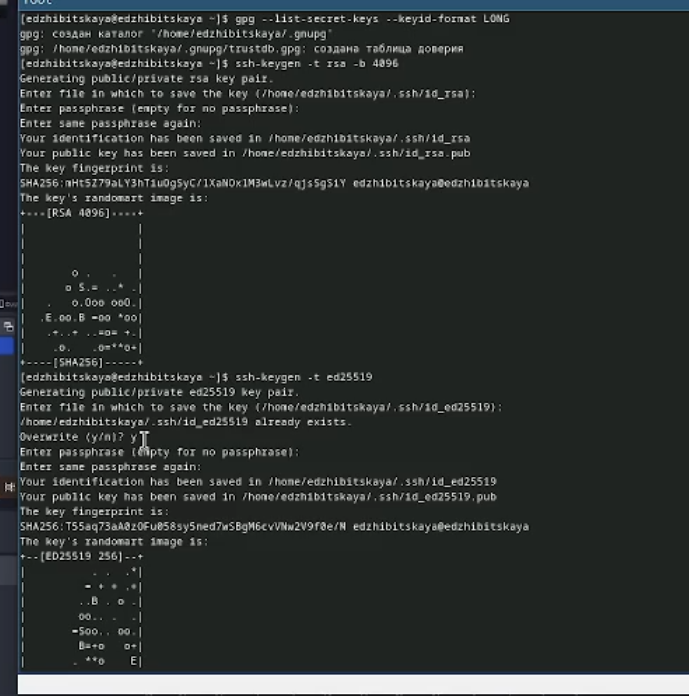


Рис. 4: Создание ключей

На платформе github учетная запись была создана ранее, поэтому просто авторизируемся и добавим созданные ключи туда, скопировав их отпечатки (рис. 5).

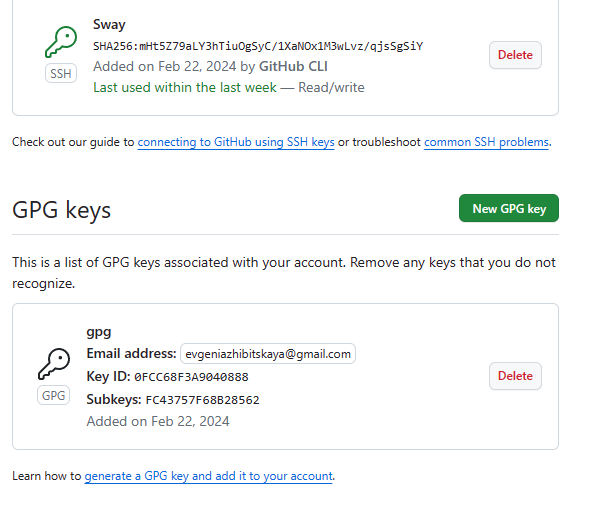


Рис. 5: Добавление ключей

Следующим шагом настроим автоматическую подпись коммитов, введя следующие команды (рис. 6).

Настройка подписи

Рис. 6: Настройка подписи

Наконец авторизирумся с помощью команды gh login auth и, создав и перейдя в нужные каталоги сначала создадим репозиторий на основе шаблона, а затем клонирукем его себе (рис. 7).

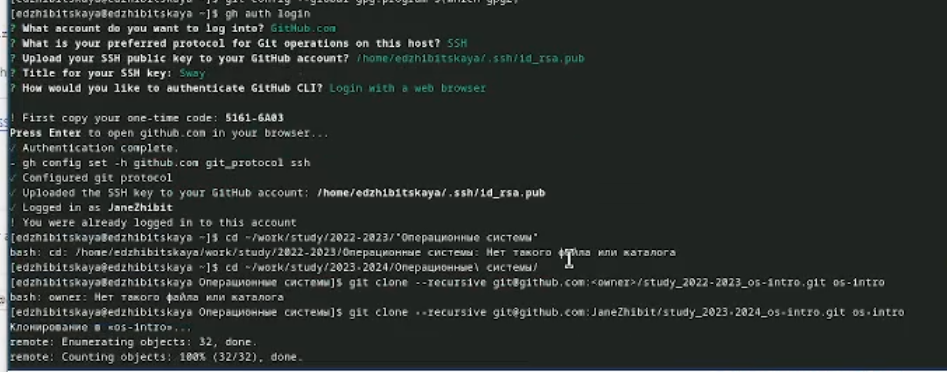


Рис. 7: Авторизация и клонирование

Также, перейдя в нужный каталог, удалим лишние файла и создадим необходимые каталоги (рис. 8).

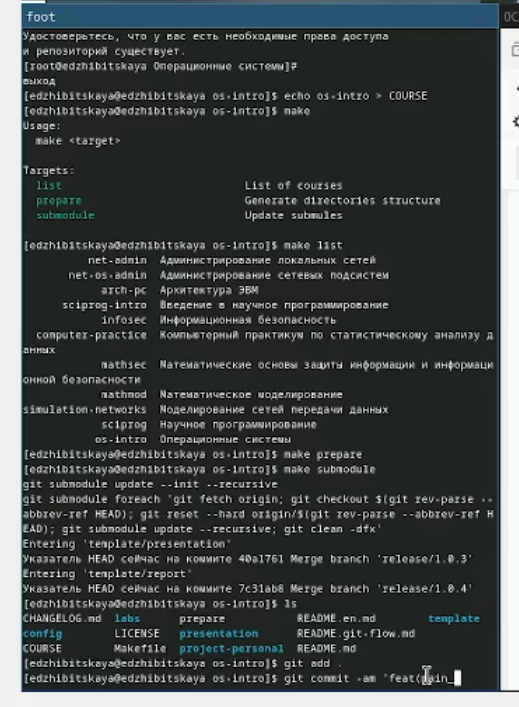


Рис. 8: Создание нужной структуры

Отправим все изменения на githib (рис. **¿fig:009?**).

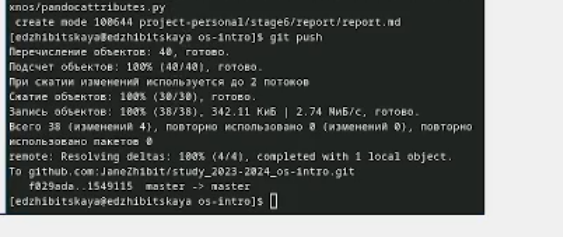


Рис. 9: Отправка изменений на сервер

# 3 Ответы на 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 изменения разных людей

1. 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)
2. 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)

1. Опишите действия с VCS при единоличной работе с хранилищем. Работа происходит на своем компьютере, сначала обновляются данные, в конце они размещаются в центральном репозитории
2. 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
3. Каковы основные задачи, решаемые инструментальным средством git?

* хранение информации о всех изменениях
* обеспечение удобства командной работы

1. Назовите и дайте краткую характеристику командам git. Создание основного дерева репозитория:

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

1. Приведите примеры использования при работе с локальным и удалённым репозиториями. локальный репозиторий - работа со своими файлами удаленный репозиторий - совместная работа, общий проект
2. 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
3. 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.

# 4 Выводы

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