Лабораторная работа №2

Первоначальна настройка git.

Хохлачёва Полина Дмитриевна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Установка программного обеспечения(рис. 1).

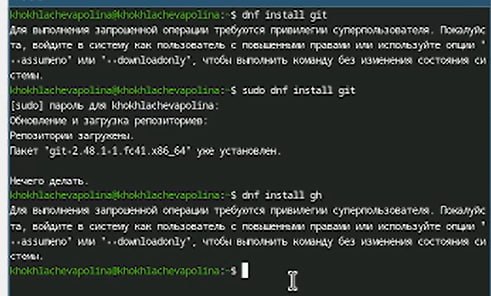


Рис. 1: Установка

Задаём имя и email репозитория(рис. 2).

Задаём имя

Рис. 2: Задаём имя

Параметры(рис. 3).



Рис. 3: Параметры

Создание ключей(рис. 4).



Рис. 4: Создание

создание ключей pgp(рис. 5).

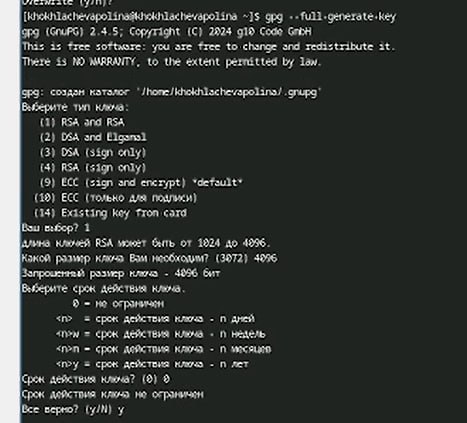


Рис. 5: Создание

добавляем ключ(рис. 6).

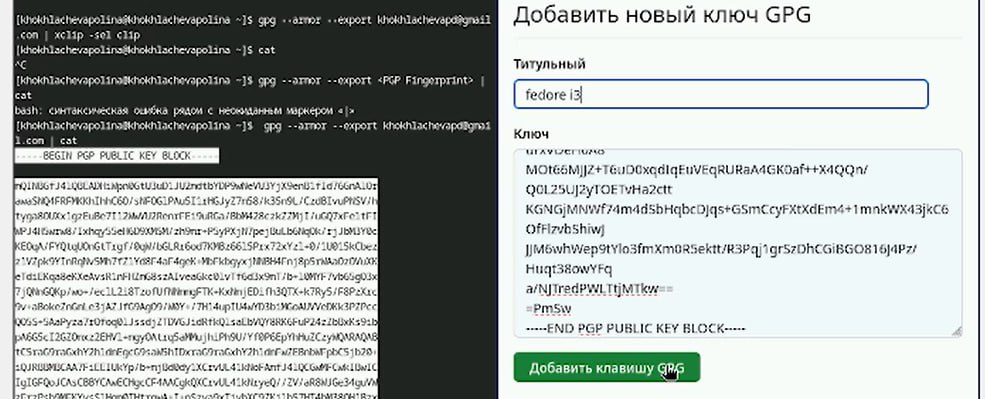


Рис. 6: Создание

настройка подписей(рис. 7).

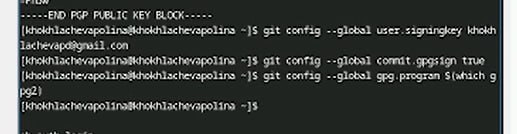


Рис. 7: Настройка

создание репозитория на основе шаблона(рис. 8).



Рис. 8: Создание

настройка каталога курса(рис. 9).

Настройка

Рис. 9: Настройка

удаление лишних файлов и создание необходимых каталогов(рис. **¿fig:010?**).

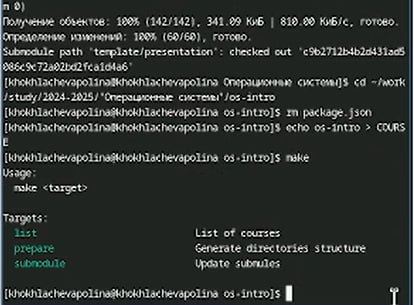


Рис. 10: Удаление и создание

отправляем файл(рис. **¿fig:011?**).

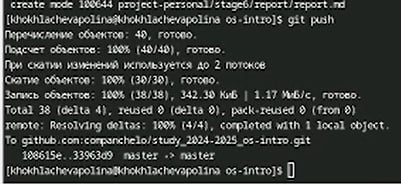


Рис. 11: Отправление

Ответы на вопросы

▎1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Системы контроля версий (Version Control Systems, VCS) — это инструменты, которые позволяют отслеживать изменения в файлах и координировать работу нескольких людей над одним проектом. Основные задачи, которые решают VCS, включают:

• Отслеживание изменений в коде и других файлах.

• Сохранение историй изменений, что позволяет откатываться к предыдущим версиям.

• Совместная работа над проектом несколькими разработчиками.

• Управление конфликтами при одновременной работе нескольких пользователей над одним файлом.

▎2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

• Хранилище (repository) — это место, где хранятся все версии проекта, включая файлы и их изменения.

• Commit — это операция, которая сохраняет текущее состояние файлов в хранилище, фиксируя изменения с комментарием, описывающим эти изменения.

• История — это последовательность всех коммитов в репозитории, позволяющая отслеживать, какие изменения были внесены и когда.

• Рабочая копия (working copy) — это локальная версия файлов проекта, с которой разработчик работает на своем компьютере. Она может содержать изменения, которые еще не были зафиксированы (commit).

▎3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

• Централизованные VCS: Все изменения хранятся на одном центральном сервере. Пользователи получают доступ к этому серверу для получения последних изменений и отправки своих. Пример: Subversion (SVN), CVS.

• Децентрализованные VCS: Каждый разработчик имеет полную копию репозитория, включая всю историю изменений. Это позволяет работать офлайн и синхронизировать изменения с другими разработчиками по мере необходимости. Пример: Git, Mercurial.

▎4. Опишите действия с VCS при единоличной работе с хранилищем.

При единоличной работе с VCS разработчик обычно выполняет следующие действия:

1. Создание репозитория: Инициализация нового репозитория с помощью команды git init.
2. Работа с файлами: Создание или редактирование файлов в рабочей копии.
3. Добавление изменений: Использование команды git add для добавления измененных файлов в индекс.
4. Коммит изменений: Фиксация изменений с помощью команды git commit.
5. Просмотр истории: Использование команды git log для просмотра истории коммитов.
6. Откат изменений: При необходимости можно откатить изменения к предыдущему коммиту.

▎5. Опишите порядок работы с общим хранилищем VCS.

При работе с общим хранилищем VCS порядок действий может быть следующим:

1. Клонирование репозитория: Получение копии удаленного репозитория с помощью git clone.
2. Создание новой ветки: Создание новой ветки для работы над новой функцией или исправлением.
3. Внесение изменений: Работа с файлами в рабочей копии.
4. Добавление и коммит изменений: Использование git add и git commit для сохранения изменений.
5. Синхронизация с удаленным репозиторием: Использование git pull для получения последних изменений из удаленного репозитория.
6. Отправка изменений: Использование git push для отправки своих коммитов в удаленный репозиторий.

▎6. Каковы основные задачи, решаемые инструментальным средством git?

Основные задачи Git включают:

• Отслеживание изменений в коде.

• Ведение истории всех изменений.

• Управление ветвями и слияниями.

• Поддержка совместной работы нескольких разработчиков.

• Упрощение отката изменений и исправления ошибок.

▎7. Назовите и дайте краткую характеристику командам git.

• git init: Инициализация нового репозитория.

• git clone: Клонирование удаленного репозитория.

• git add: Добавление изменений в индекс для последующего коммита.

• git commit: Фиксация изменений в локальном репозитории.

• git status: Просмотр статуса рабочей копии (изменения, которые еще не закоммичены).

• git log: Просмотр истории коммитов. # Выводы

Мы изучили идеологию и применение средств контроля версий.Освоили умения по работе с git.