Отчёт по лабораторной работе №2

Дисциплина : операционная система

Замула Егор Сергеевич

Содержание

# Цель работы

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

# Задание

- Создать базовую конфигурацию для работы с git.  
- Создать ключ SSH.  
- Создать ключ PGP.  
- Настроить подписи git.  
- Зарегистрироваться на Github.  
- Создать локальный каталог для выполнения заданий по предмету.

# 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 хранилища 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 конфликтов, 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.

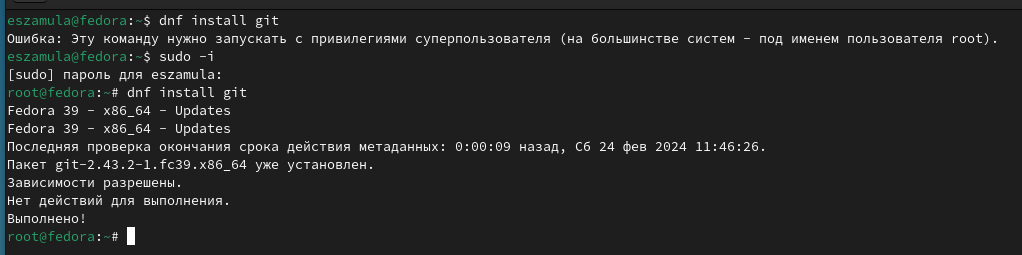
Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

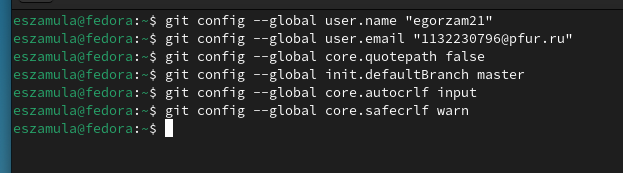
# Выполнение лабораторной работы

1. Установка системы git в нашу ОС



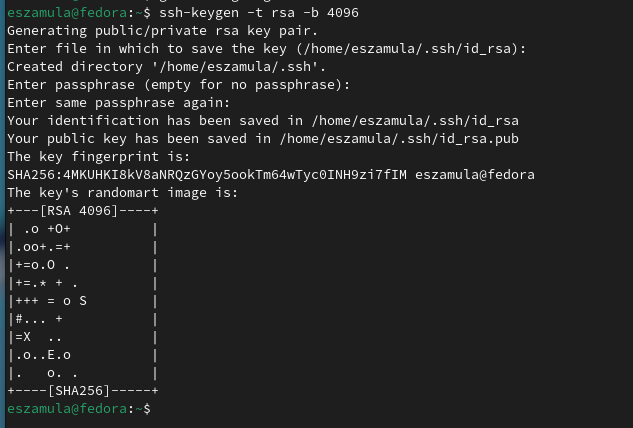
Установка git

1. Проводим базоваю настройку git



Настройка git

1. Создаем SSH ключа по алгоритму rsa с ключём размером 4096 бит



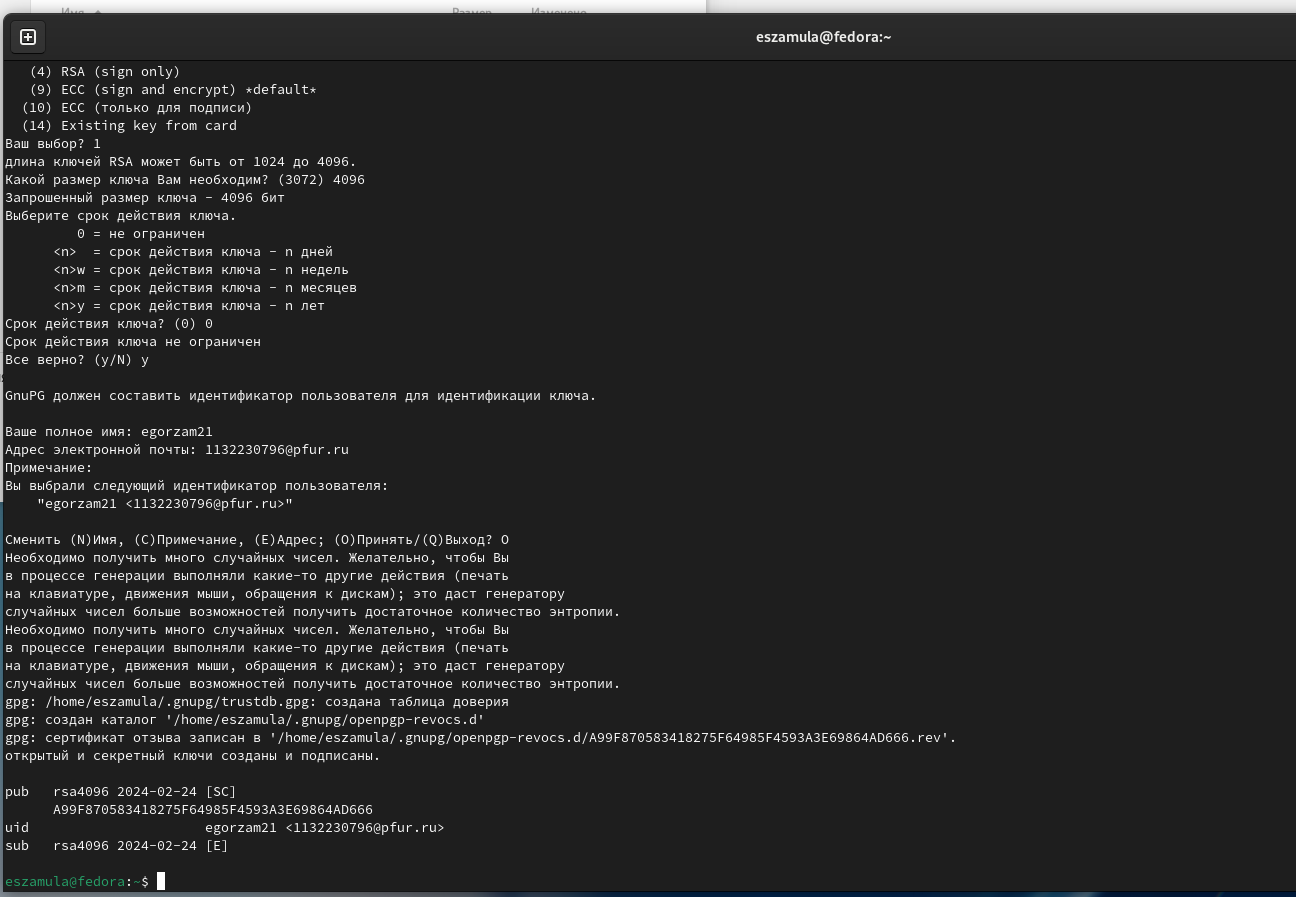
Создание ssh ключа

1. Добавляем SSH ключ на гитхаб в разеделе settings



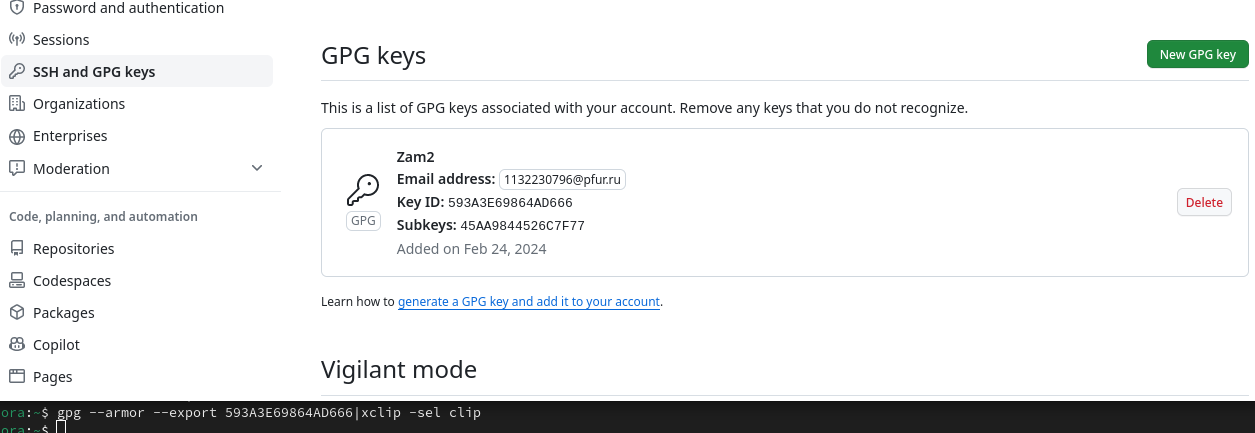
Добавление SSH ключа

1. Создание pgp ключа



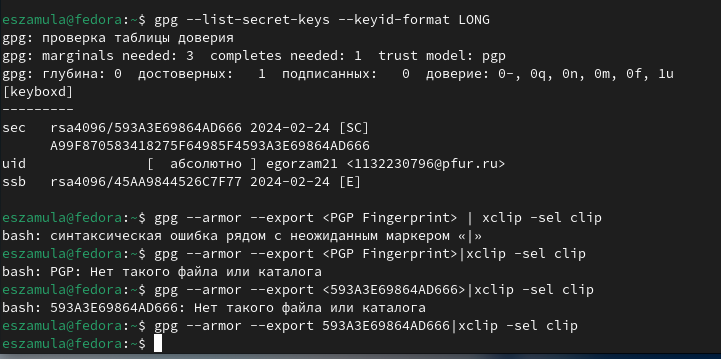
pgp ключ

1. Добавляем ключ на гитхаб (точно так же как и SSH)



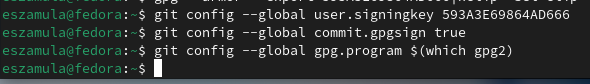
Добавляем pgp ключ

1. Получение sec параметра



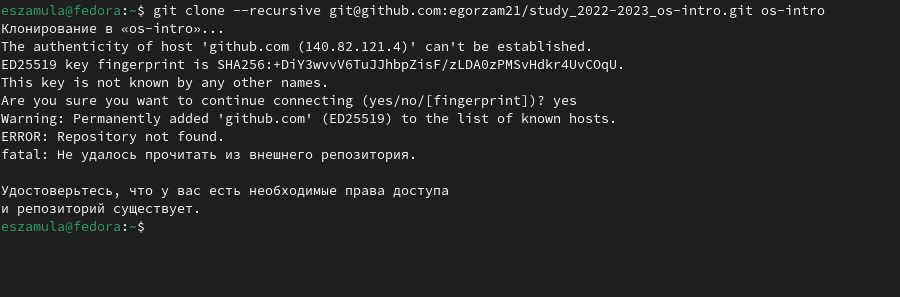
sec

1. Настройка автоматических подписей коммитов git



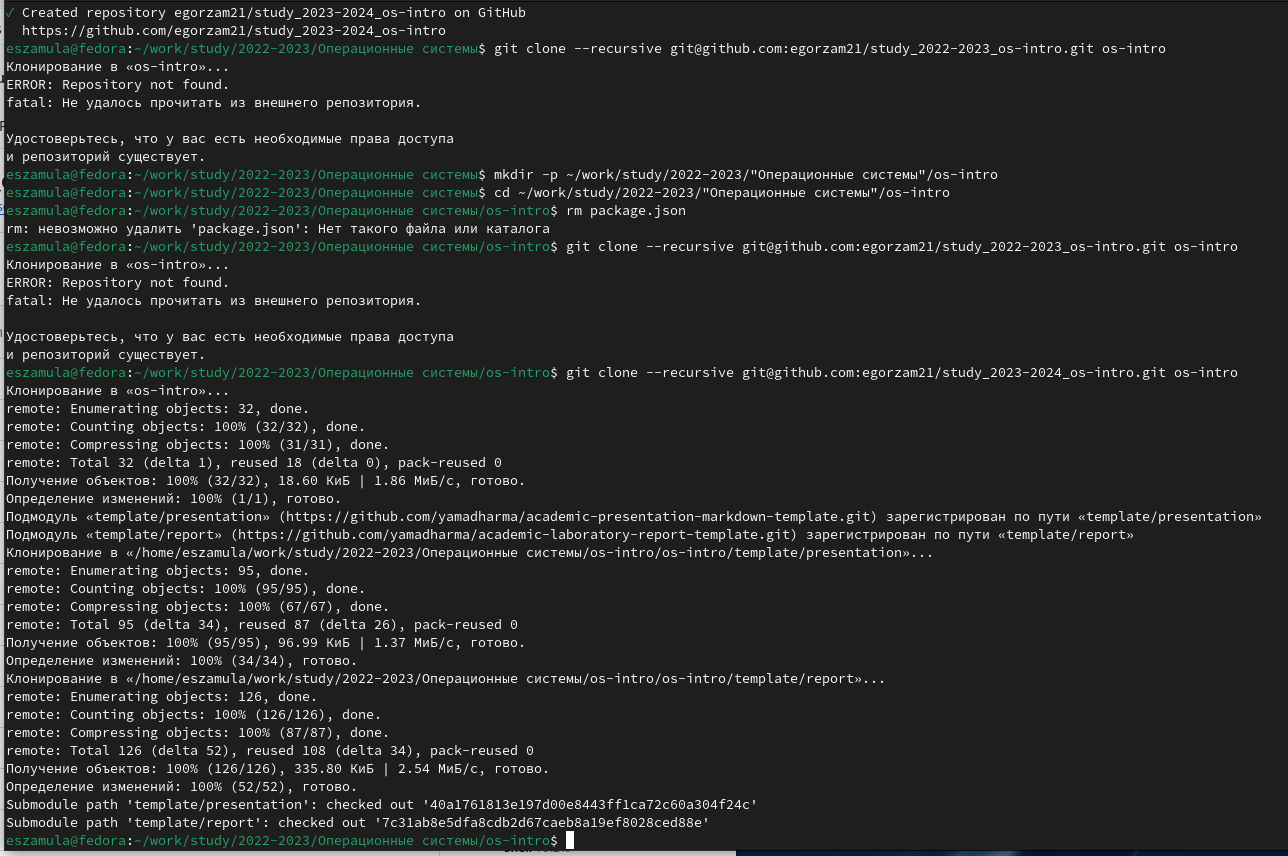
Подписи

1. Клонируем репозиторий

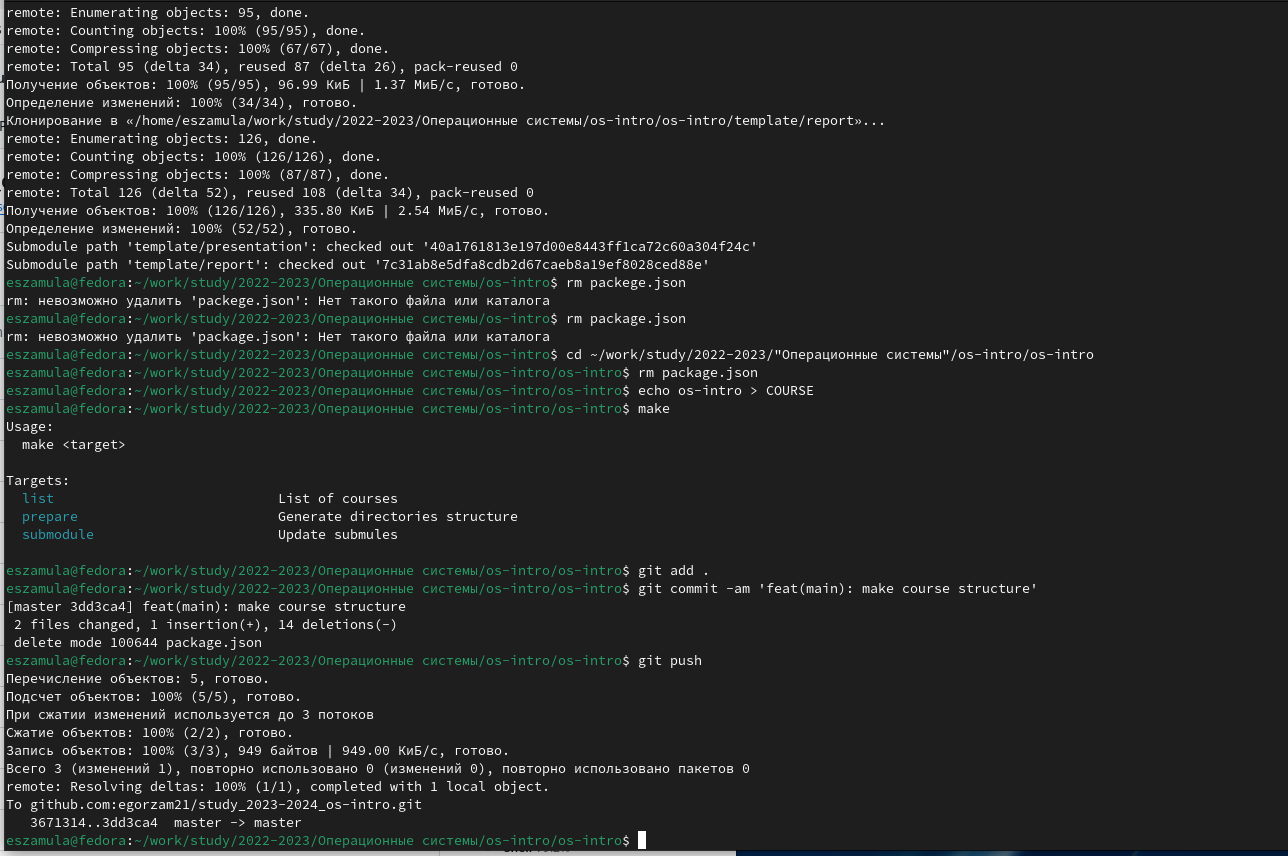


Клонируем

1. Настройка каталога курса



Настройка каталога

1. Отправка на сервер нашего отредактированного репозитория 

# Контрольные вопросы

Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?  
Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.  
Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.  
Опишите действия с VCS при единоличной работе с хранилищем.  
Опишите порядок работы с общим хранилищем VCS.  
Каковы основные задачи, решаемые инструментальным средством git?  
Назовите и дайте краткую характеристику командам git.  
Приведите примеры использования при работе с локальным и удалённым репозиториями.  
Что такое и зачем могут быть нужны ветви (branches)?  
Как и зачем можно игнорировать некоторые файлы при commit?

# Выводы

В ходе выполнения работы изучил git

# Список литературы