Отчёта по лабораторной работе №2

Еремина Оксана Андреевна

Содержание

Список таблиц

Список иллюстраций

1 Цель работы

Целью данной лабораторной работы является освоение практических навыков умений по работе с git

2 Задание

1. Создать базовую конфигурацию для работы с git.
2. Создать ключ SSH.
3. Создать ключ PGP.
4. Настроить подписи git.
5. Зарегистрироваться на Github.
6. Создать локальный каталог для выполнения заданий по предмету.

3 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.

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

4 Выполнение лабораторной работы

1. Установка программного обеспечения

В терминале ввожу команду, которая устанавливает git и gh(рис.1)

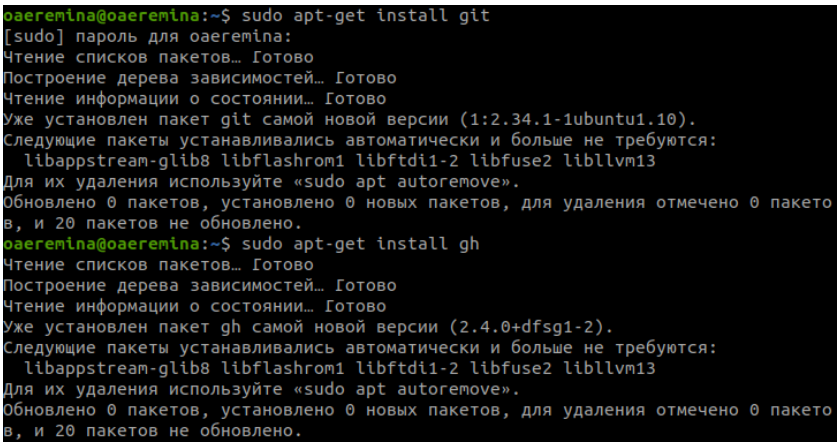

Рис. 4.1: Установка git и gh

2. Базовая настройка git

В терминале задаю имя и email моего репозитория. Далее настраиваю его вводя команды. (рис.2)

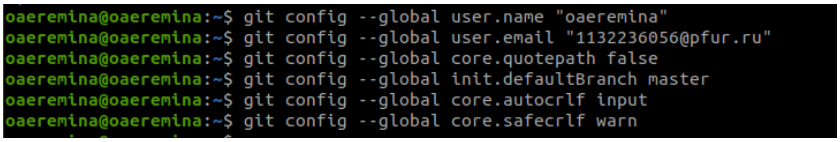

Рис. 4.2: Настройка git

3. Создание ключа ssh

Генерирую ключ и выбираю нужные опции (рис. 3)

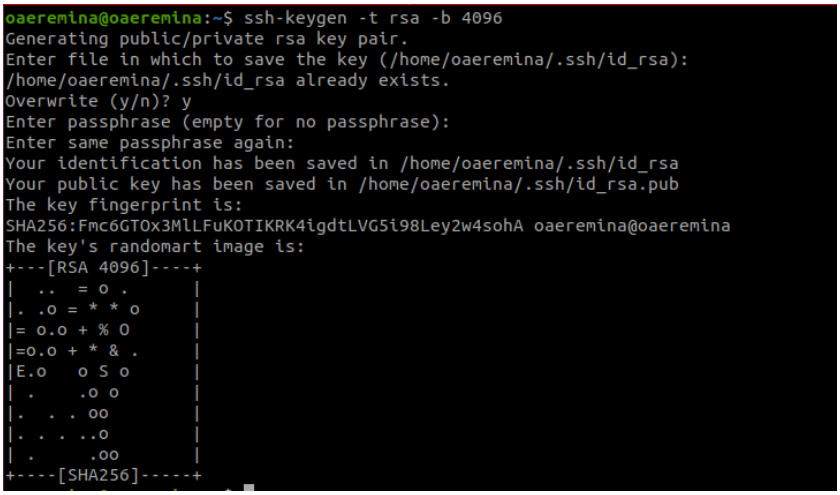

Рис. 4.3: Создание ключа ssh

4. Создание ключа pgp

Вывожу список ключей и копирую сгенерированый ключ в буфер обмена (рис.4)

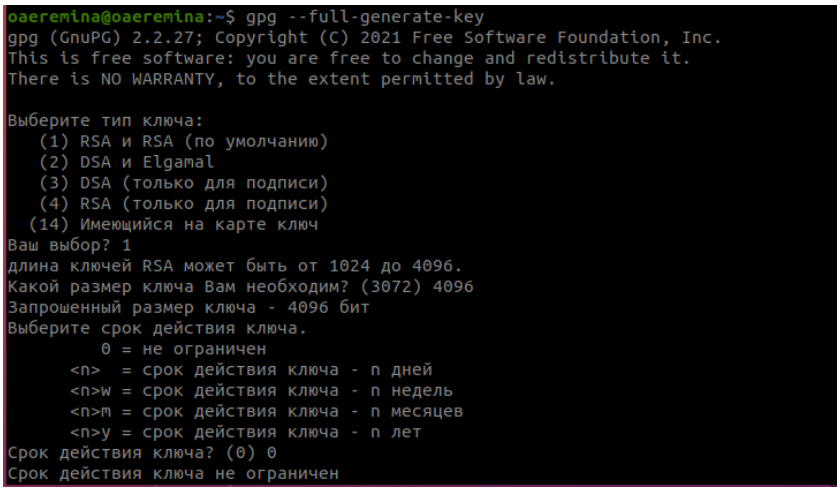

Рис. 4.4: Создание ключа pgp

Перехожу в свой репозиторий, переходу в New GPG key и вставляю код из буфера обмена (рис.5)

Рис. 4.5: Добавление ключа pgp

5. Настройка автоматических подписей коммитов git

Ввожу команды, чтобы git применял мой email при подписи коммитов (рис.6)

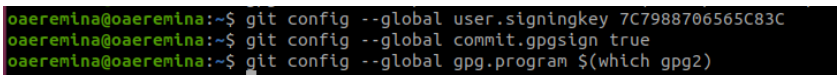

Рис. 4.6: Настройка подписей коммитов

6. Настройка gh

Авторизуюсь, вводя команду, далее отвечаю на вопросы (рис.7) После ответов на вопросы меня переносит в браузер и я ввожу код, который написан в терминале.

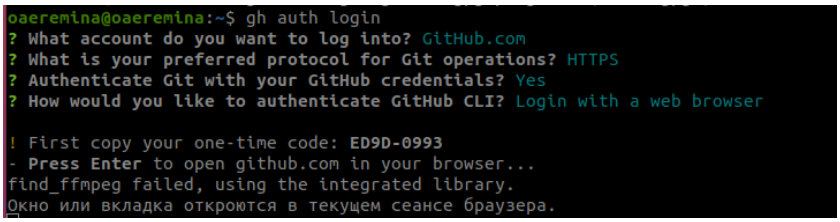

Рис. 4.7: Настройка gh

7. Сознание репозитория курса на основе шаблона

Создаю шаблон рабочего пространства, воодя нужные команды (рис.8)

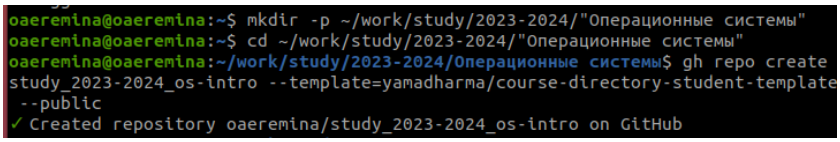

Рис. 4.8: Создание шаблона курса

Перехожу в каталог курса, удаляю лишние файлы, затем создаю необходимые каталоги (рис.9) И отправляю эти файлы на сервер

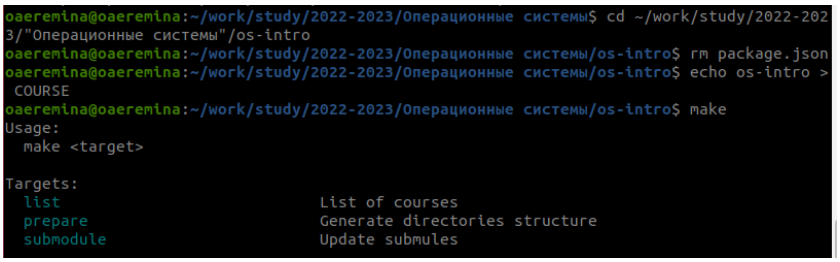

Рис. 4.9: Настройка каталога курса

5 Выводы

При выполнение данной лабораторной работы я приобрела практические навыки работы с git

6 Контрольные вопросы

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

Системы контроля версий (VCS) - программное обеспечение для облегчения работы с изменяющейся информацией. Они позволяют хранить несколько версий изменяющейся информации, одного и того же документа, может предоставить доступ к более ранним версиям документа. Используется для работы нескольких человек над проектом, позволяет посмотреть, кто и когда внес какое-либо изменение и т. д. VCS ррименяются для: Хранения понлой истории изменений, сохранения причин всех изменений, поиска причин изменений и совершивших изменение, совместной работы над проектами.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, ра

Хранилище - репозиторий, хранилище версий, в нем хранятся все документы, включая

3. Что представляют собой и чем отличаются централизованные и децентрализованные

Централизованные VCS (например: CVS, TFS, AccuRev) - одно основное хранилище всег

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

Сначала создается и подключается удаленный репозиторий, затем по мере изменения г

5. Опишите порядок работы с общим хранилищем VCS.

Участник проекта перед началом работы получает нужную ему версию проекта в хранил

6. Каковы основные задачи, решаемые инструментальным средством git?

Хранение информации о всех изменениях в вашем коде, обеспечение удобства командно

7. 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 add имена_файлов удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории): 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 commit создание новой ветки, базирующейся на текущей: 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 уда-

ление 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 репозитория

git push -all отправляем из локального репозитория все сохраненные изменения в це

9. Что такое и зачем могут быть нужны ветви (branches)?

Ветвление - один из параллельных участков в одном хранилище, исходящих из одной в

10. Как и зачем можно игнорировать некоторые файлы при commit?

Во время работы над проектом могут создаваться файлы, которые не следуют добавлят

Список литературы

1. Лабораторная работа № 2 URL: https://esystem.rudn.ru/mod/page/view.php?id=1098790