**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**ФАКУЛЬТЕТ ФИЗИКОМАТЕМАТИЧЕСКИХ И ЕСТЕСТВЕННЫХ НАУК**

**КАФЕДРА МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК**

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

на тему:

“Управление версиями”

ДИСЦИПЛИНА: ОПЕРАЦИОННЫЕ СИСТЕМЫ

Работу выполнила:

Студент 1 курса

Группы НКНбд-02-21

Е.К. Грузинова

Г. Москва,

2022 г.

**Оглавление**

1. Цель работы
2. Задание.
3. Теоретическое введение
4. Выполнение лабораторной работы
5. Выводы
6. Контрольные вопросы
7. Список литературы

**Цель работы**

Изучить идеологию и применение средств контроля версий, а также освоить умения по работе с git.

**Задание**

1. Создайте учётную запись на https://github.com.
2. Заполните основные данные на <https://github.com>.
3. Установка git-flow в Fedora Linux.
4. Установка gh в Fedora Linux.
5. Базовая настройка git.
6. Создайте ключи ssh.
7. Создайте ключи pgp.
8. Добавление PGP ключа в GitHub.
9. Настройка автоматических подписей коммитов git.
10. Настройка gh.
11. Создание репозитория курса на основе шаблона.
12. Настройка каталога курса.

**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 дельтакомпрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

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. Создайте учётную запись на https://github.com.

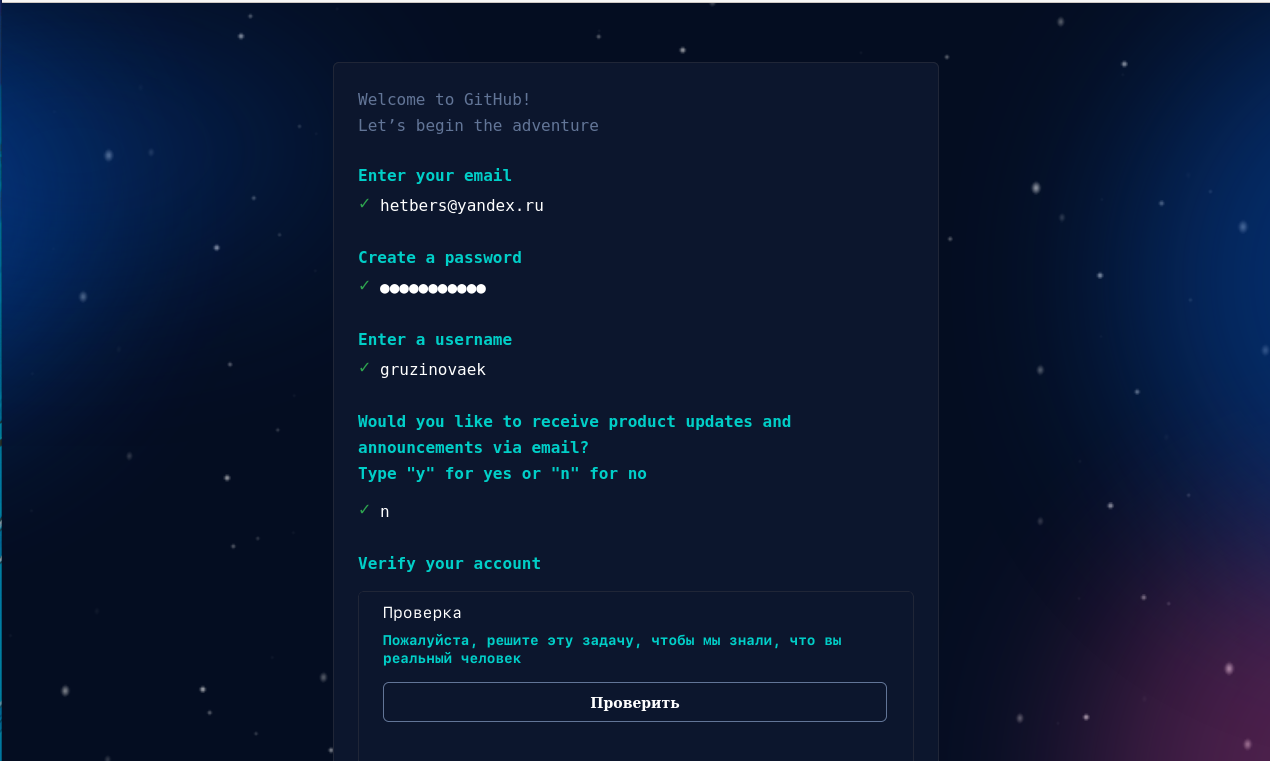


Рис.1. Регистрация учетной записи на github.com.

1. Заполните основные данные на <https://github.com>.

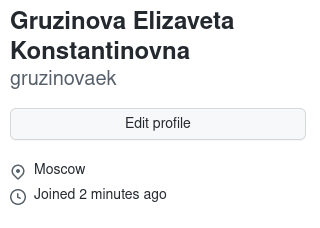


Рис.2. Заполнение основных данных.

1. Установка git-flow в Fedora Linux.

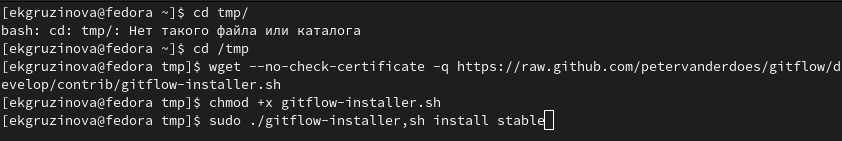


Рис.3. Ручная установка git-flow.

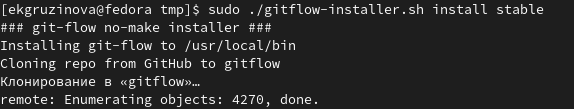


Рис.5. Git-flow успешно установлен.

1. Установка gh в Fedora Linux.

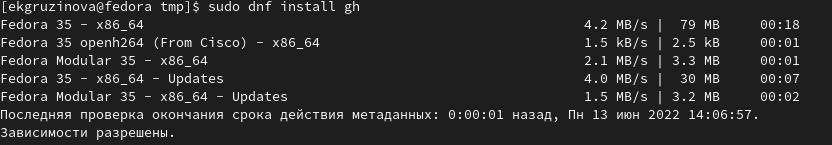


Рис.6. Команда для установки gh в Fedora Linux.

1. Базовая настройка git.

C:\Users\admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана от 2022-06-13 14-09-14.png

Рис.7. Установка имени и email владельца репозитория.

C:\Users\admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана от 2022-06-13 14-09-57.png

Рис.8. Настройка utf-8 в выводе сообщений git.



Рис.9. Установка имени начальной ветки master.

C:\Users\admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана от 2022-06-13 14-17-53.png

Рис.10. Регулирование параметра autocrlf.

C:\Users\admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Снимок экрана от 2022-06-13 14-18-11.png

Рис.11. Регулирование параметра safecrlf.

1. Создайте ключи ssh.

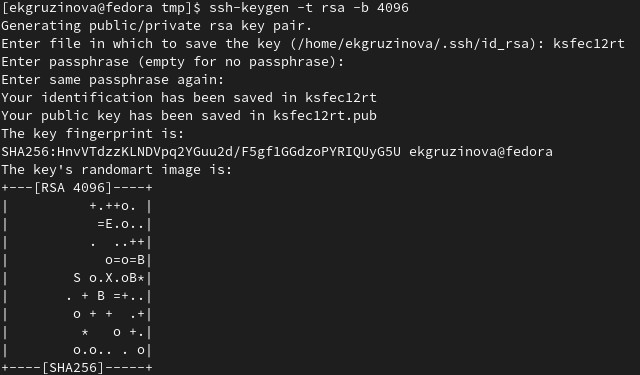


Рис.12. Создание ключа ssh по алгоритму rsa с ключём размером 4096 бит.

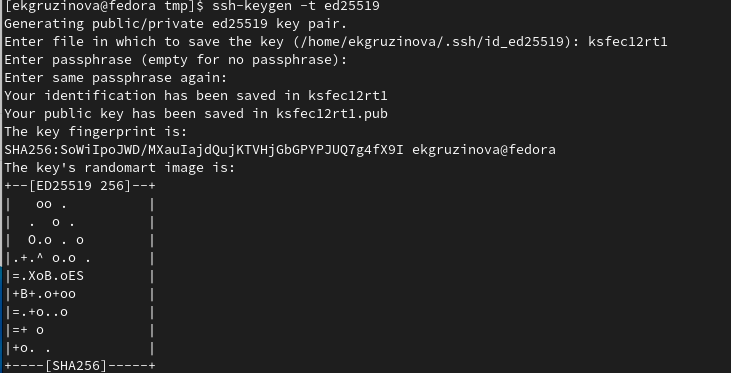


Рис.13. Создание ключа ssh по алгоритму ed25519.

1. Создайте ключи pgp.

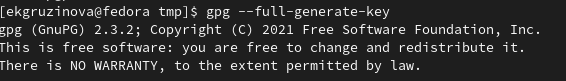


Рис.14. Генерация ключа pgp.

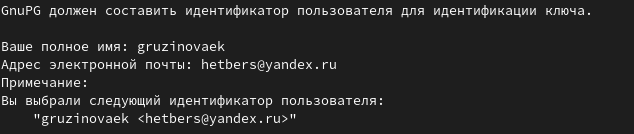


Рис.15. Составление идентификатора пользователя для идентификации ключа.

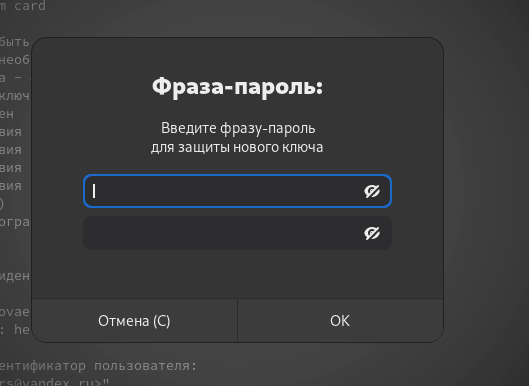


Рис.16. Запрос на создание фразы-пароля.

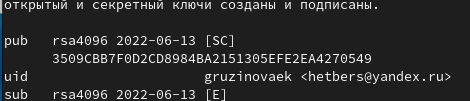


Рис.17. Ключ pgp создан с необходимыми данными.

1. Добавление PGP ключа в GitHub.



Рис.18. Вывод списка ключей, из которого копирует отпечаток приватного ключа.



Рис.19. Копирование сгенерированного PGP ключа в буфер обмена.

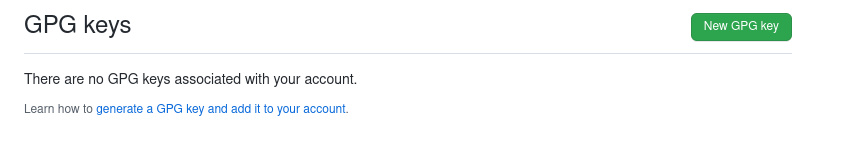


Рис.20. Настройки ключа GPG на github.com

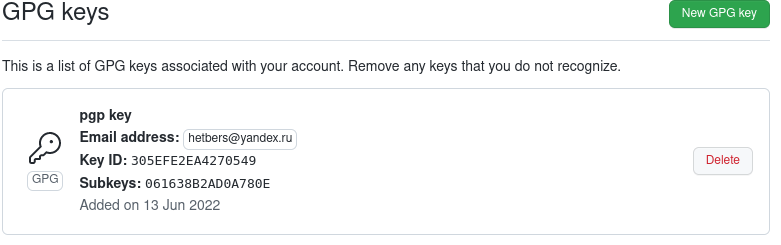


Рис.21. Добавление PGP ключа в github.

1. Настройка автоматических подписей коммитов git.



Рис.22. Через email указываем git применять его при подписи коммитов.

1. Настройка gh.

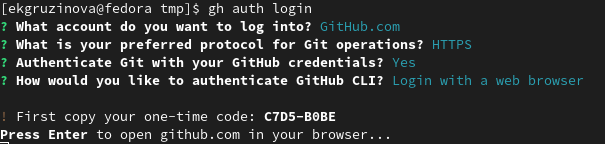


Рис.23. Команда и процесс авторизации gh.

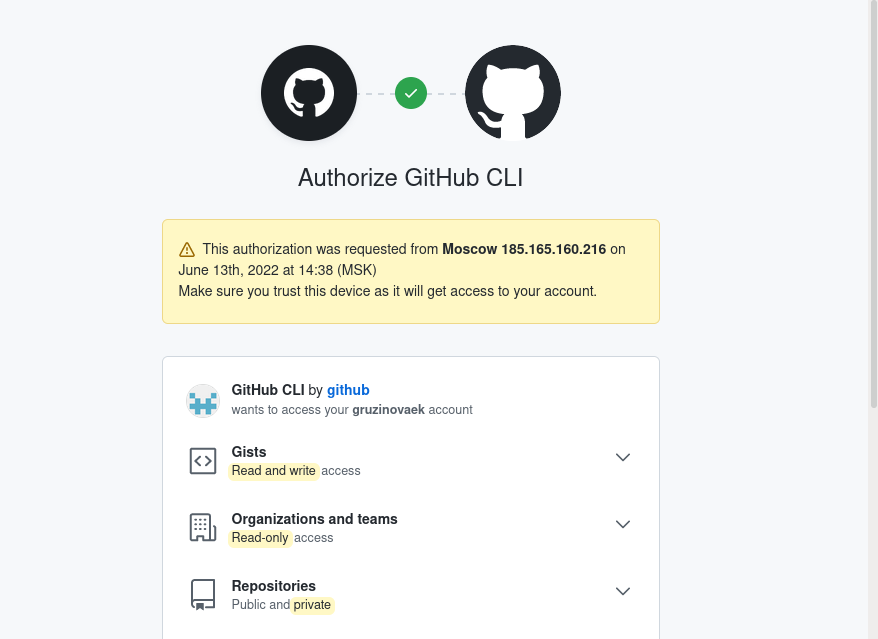


Рис.24. Подтверждение авторизации происходит через браузер.

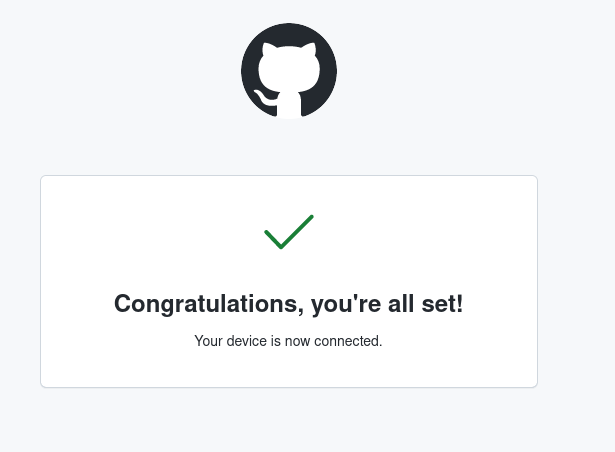


Рис.25. Успешная авторизация gh.

1. Создание репозитория курса на основе шаблона.



Рис.26. Создание папки “Операционные системы” для репозитория.



Рис.27. Создание репозитория на github.com.

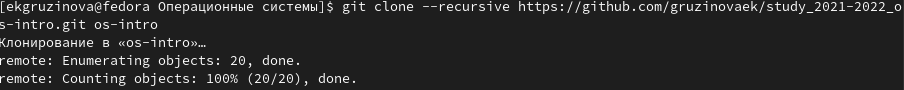


Рис.28. Клонирование репозитория в os-intro.

1. Настройка каталога курса.



Рис.29. Удаление лишних файлов.

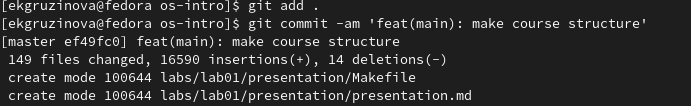


Рис.30. Создание необходимых каталогов.

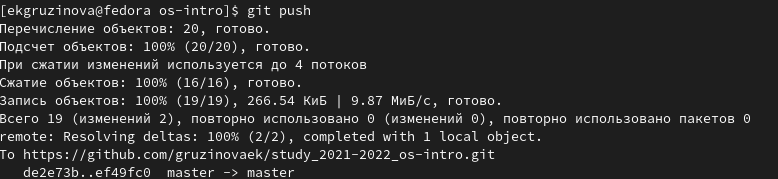


Рис.31. Отправка файлов на сервер.

**Выводы**

В процессе выполенения лабораторной работы я изучила идеологию и применение средств контроля версий, а также освоила умения по работе с git.

**Контрольные вопросы**

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

Система контроля версий — **программное обеспечение, которое обеспечивает командную работу в рамках одного или нескольких проектов**. Команда разработчиков взаимодействует с консольным или браузерным инструментом для выгрузки кода на сервер, скачивания его на рабочий компьютер и изменения структуры. Она хранит все версии проекта и обеспечивает к ним доступ. Любой член команды может взаимодействовать с основной «веткой» проекта или создавать новые.

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

* Хранилище (repository, сокр. repo), или репозитарий, — место хранения всех версий и служебной информации.
* Версия (revision), или ревизия, — состояние всех файлов на определенный момент времени, сохраненное в репозитарии, с дополнительной информацией.
* Коммит (commit; редко переводится как «слепок») —
* 1) синоним версии;
* 2) создание новой версии («сделать коммит», «закоммитить»).
* Рабочая копия (working copy или working tree) — текущее состояние файлов проекта, основанное на версии из хранилища (обычно на последней).

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

Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion.

Децентрализованные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве DVCS нет такого понятия как “выделенный сервер с центральным репозиторием”.

Большое преимущество такого подхода заключается в автономии разработчика при работе над проектом, гибкости общей системы и повышение надежности, благодаря тому, что каждый разработчик имеет локальную копию центрального репозитория. Две наиболее известные DVCS – это Git и Mercurial.

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

Сначала вам нужно создать и подключить удаленный репозиторий. Затем, поскольку никто, кроме вас, не изменяет проект, по мере изменения проекта отправляйте изменения на сервер, и нет необходимости загружать изменения.

5. Опишите порядок работы с общим хранилищем VCS.

Участник проекта (пользователь) получает нужную ему версию файлов перед началом работы с помощью определенных команд. После внесения изменений пользователь помещает новую версию в репозиторий. В то же время предыдущие версии не удаляются из центрального хранилища, и вы можете вернуться к ним в любое время.

6. Каковы основные задачи, решаемые инструментальным средством git?

У **Git** две **основных** **задачи**: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.

7. Назовите и дайте краткую характеристику командам git.

git init – создание основного дерева репозитория.

git pull – получение обновлений (изменений) текущего дерева из центрального репозитория.

git push – отправка всех произведённых изменений локального дерева в центральный репозиторий.

git status – просмотр списка изменённых файлов в текущей директории.

git diff – просмотр текущих изменения.

git add . – добавить все изменённые и/или созданные файлы и/или каталоги.

git add имена\_файлов – добавить конкретные изменённые и/или созданные файлы и/или каталоги.

git rm имена\_файлов – удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории).

git commit -am 'Описание коммита' – сохранить все добавленные изменения и все изменённые файлы.

И т.д.

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

Ветви функций, также иногда называемые ветвями тем, используются для разработки новых функций, которые должны появиться в текущих или будущих выпусках.

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

Существуют временные и системные файлы, которые загромождают проект и не нужны. путь к ним можно добавить в файл .gitignore, тогда они не будут добавлены в проект.

**Список литературы:**

1. Лабораторная работа №2

https://esystem.rudn.ru/pluginfile.php/1383169/mod\_resource/content/4/002-lab\_vcs.pdf