Отчёт по лабораторной работе №4

Операционные системы

Закиров Нурислам Дамирович

Содержание

# 1 Цель работы

Целью данной лаболаторной работы является изучение идеологии и применение средств контроля версий и приобретение знаний по работе с git.

# 2 Задание

1. Заргеистрироваться на GitHub
2. Создать базовую конфигурацию для работы с git
3. Создать ключ SSH
4. Создать ключ GPG
5. Настроить подписи Git
6. Создать локальный каталог для выполнения заданий по предмету.

# 3 Выполнение лабораторной работы

## 3.1 Установка программного обеспечения

Заранее регистрация в git уже была успешно выполнена, поэтому сразу переходим к следующим этапам. Устанавливаем git при помощи dnf install git и устанавливаем gh при помощи dnf install gh(рис. fig. 1).

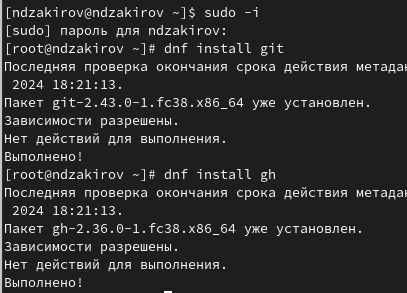


Рис. 1: Установка git и gh

## 3.2 Базовая настройка git

Далее настраиваем верификацию и подписание коммитов git, зададаем имя начальной ветки master, параметр autocrlf и safecrlf(рис. fig. 2).

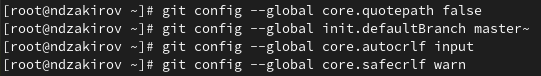


Рис. 2: Базовая настройка git

## 3.3 Создание ключа SSH

Создаем ключ ssh по алгоритму rsa с ключём размером 4096 бит и по алгоритму ed25519 (рис. fig. 3).

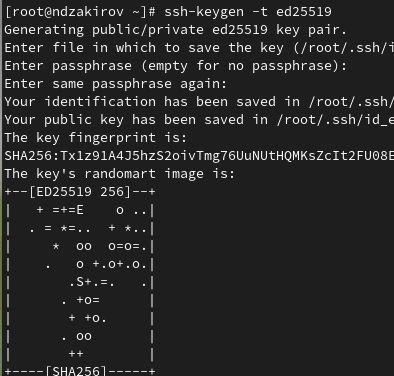


Рис. 3: Создание ключа SSH

Следующим этапом мы генерируем ключ pgp выбирая слудеющие опции:тип RSA and RSA, размер 4096, срок действия неограниченный (рис. fig. 4).

## 3.4 Создание ключа GPG

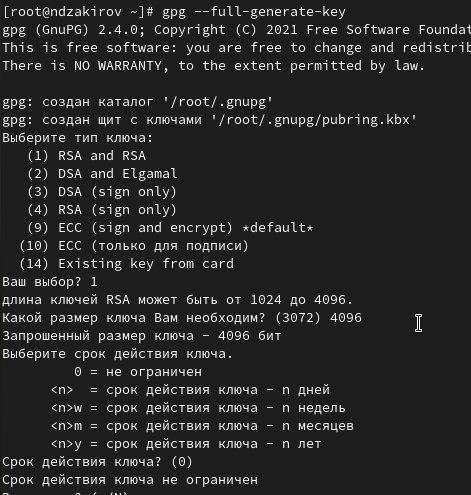


Рис. 4: Создание ключа pgp

После чего, мы копируем данный нам ключ командой gpg –armor –export | xclip -sel clip в буфер обмена, где вместо “PGP Fingerprint” вставляем данные занчения.(рис. fig. 5).

Копирование ключа в буфер обмена

Рис. 5: Копирование ключа в буфер обмена

Теперь вставляем данный ключ при помощи комбинации Ctrl+V на официальном сайте GitHub в настройках GitHub, нажав на кнопку New GPG key(рис. fig. 6).

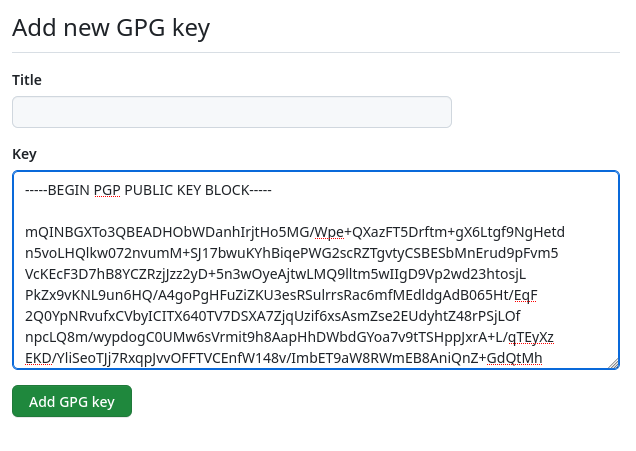


Рис. 6: Окно New GPG key в GitHub

Ключ успешно привязался(рис. fig. 7).

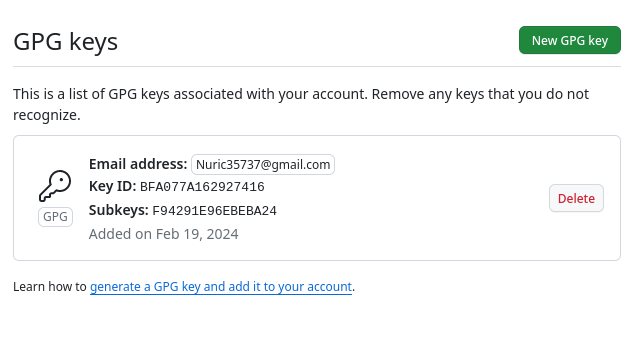


Рис. 7: Окно GitHub

## 3.5 Настроить подписи Git

Cледующим этапом мы используем введёный email и указываем Git применять его при подписи коммитов (рис. fig. 8).

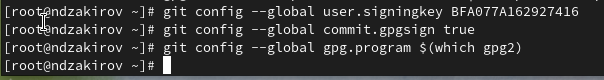


Рис. 8: Настройка подписей коммитов git

## 3.6 Настройка gh

Даллее мы авторизируемся в терминале через команду gh auth login. Отвечаем на все наводящие вопросы(рис. fig. 9).

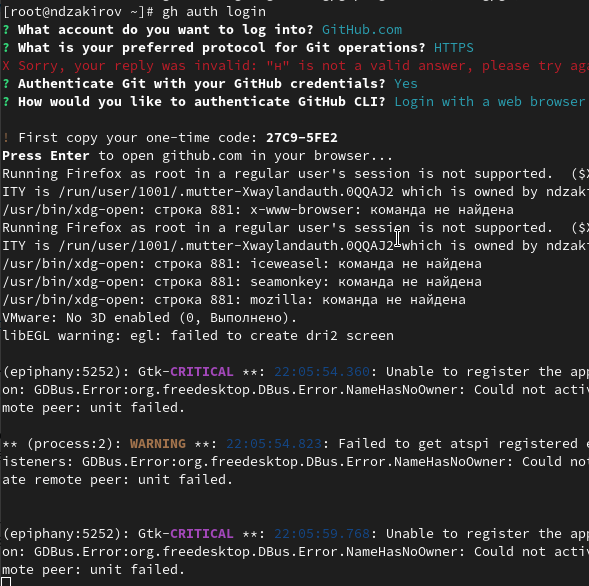


Рис. 9: Авторизация gh

После входим в свой аккаунт GitHub через браузер(рис. fig. 10).

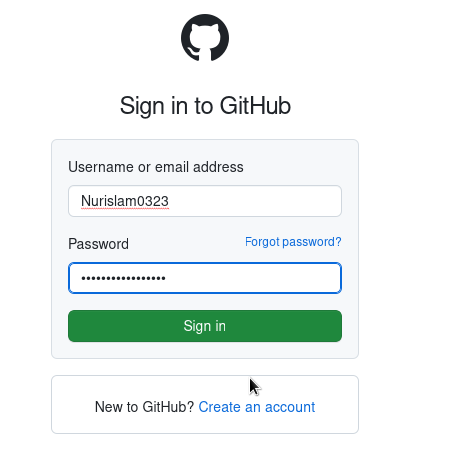


Рис. 10: Авторизация через браузер

После авторизации, Git просит ввести код, который мы заранее скопировали с терминала и вставили бразуре, после чего мы успешно авторизовались(рис. fig. 11).

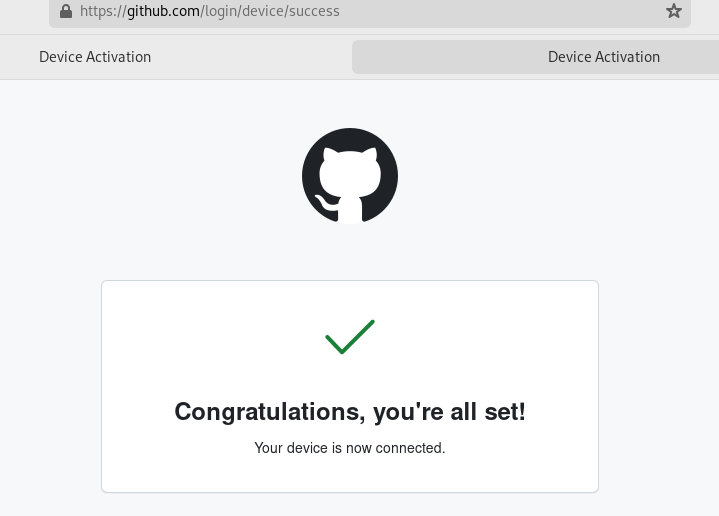


Рис. 11: Успешная авторизация

В терминале также подтвержается наша авторизация(рис. fig. 12).

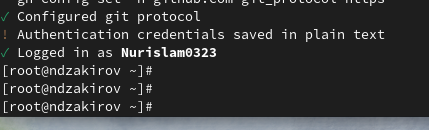


Рис. 12: Успешная авторизация

## 3.7 Создание шаблона для рабочего пространства

Далее нам необходимо создать шаблон рабочего пространства. При промощи mkdir -p ~/work/study/2023-2024/“Операционные системы” мы создаем каталог, после чего переходим в нее. Создаем шаблон при помощи gh repo create(рис. fig. 13).

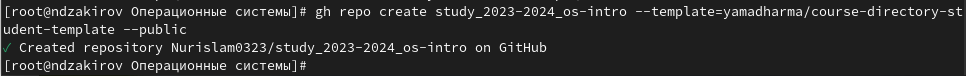


Рис. 13: Создание репозитории курса

После чего мы клониуем репозиторий курса на нашу систему при помощи git clone(рис. fig. 14).

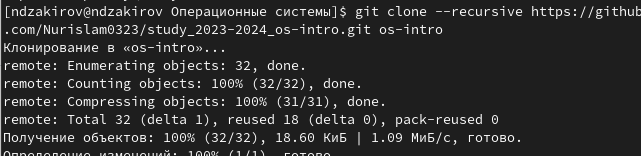


Рис. 14: Создание репозитории курса

## 3.8 Настройка каталога курса

Переходим в созданную папку os-intro при помощи cd(рис. fig. 15).

Перемещение по каталогам

Рис. 15: Перемещение по каталогам

Удаляем лишние файлы при помощи rm (рис. fig. 16).

Удаление файлов

Рис. 16: Удаление файлов

Создаем необходимые каталоги при помощи echo и make(рис. fig. 17).

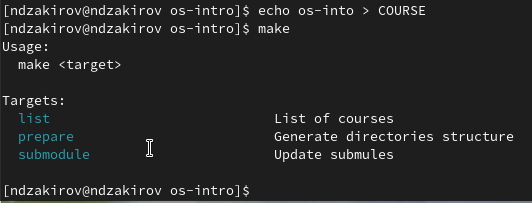


Рис. 17: Создание каталогов

Отправляем файлы на сервер GitHub при помощи git add, git commit и git push(рис. fig. 18).

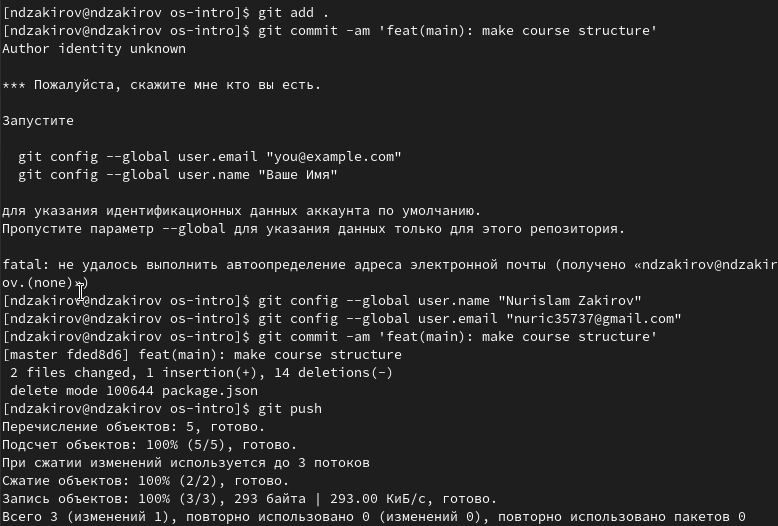


Рис. 18: Отправка файлов на сервер

# 4 Выводы

В ходе данной лабораторной работы я изучил идеологии и применения средств контроля версий, а также приобрел знания по работе с git.

# 5 Ответы на контрольные вопросы.

1. 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. Хранилище – репозиторий, хранилище версий, в нем хранятся все документы, включая историю их изменения и прочей служебной информацией. commit – отслеживание изменений, сохраняет разницу в изменениях. История – хранит все изменения в проекте и позволяет при необходимости вернуться/обратиться к нужным данным. Рабочая копия – копия проекта, основанная на версии из хранилища, чаще всего последней версии.
3. Централизованные VCS (например: CVS, TFS, AccuRev) – одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет, затем добавляет изменения обратно в хранилище. Децентрализованные VCS (например: Git, Bazaar) – у каждого пользователя свой вариант репозитория (возможно несколько вариантов), есть возможность добавлять и забирать изменения из любого репозитория. В отличие от классических, в распределенных (децентралиованных) системах контроля версий центральный репозиторий не является обязательным.
4. Сначала создается и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер.
5. Участник проекта перед началом работы получает нужную ему версию проекта в хранилище, с помощью определенных команд, после внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются. К ним можно вернуться в любой момент.
6. Хранение информации о всех изменениях в вашем коде, обеспечение удобства командной работы над кодом.
7. 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 удаление локальной ветки: git branch -D имя\_ветки

удаление ветки с центрального репозитория: git push origin :имя\_ветки

1. git push -all отправляем из локального репозитория все сохраненные изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию.
2. Ветвление - один из параллельных участков в одном хранилище, исходящих из одной версии, обычно есть главная ветка. Между ветками, т. е. их концами возможно их слияние. Используются для разработки новых функций.
3. Во время работы над проектом могут создаваться файлы, которые не следуют добавлять в репозиторий. Например, временные файлы. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл .gitignore с помощью сервисов.

# Список литературы

1. Лабораторная работа № 2 [Электронный ресурс] URL: https://esystem.rudn.ru/mod/page/view.php?id=970819