**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 2**

*дисциплина: Операционные системы*

Студент:

Батов Дмитрий Сергеевич

Группа:

НПМБВ-02-21

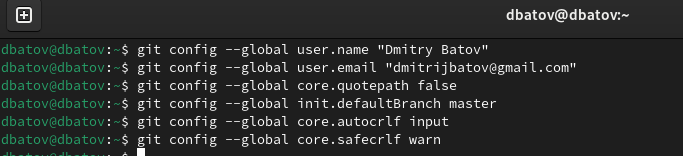
**МОСКВА**

2025 г.

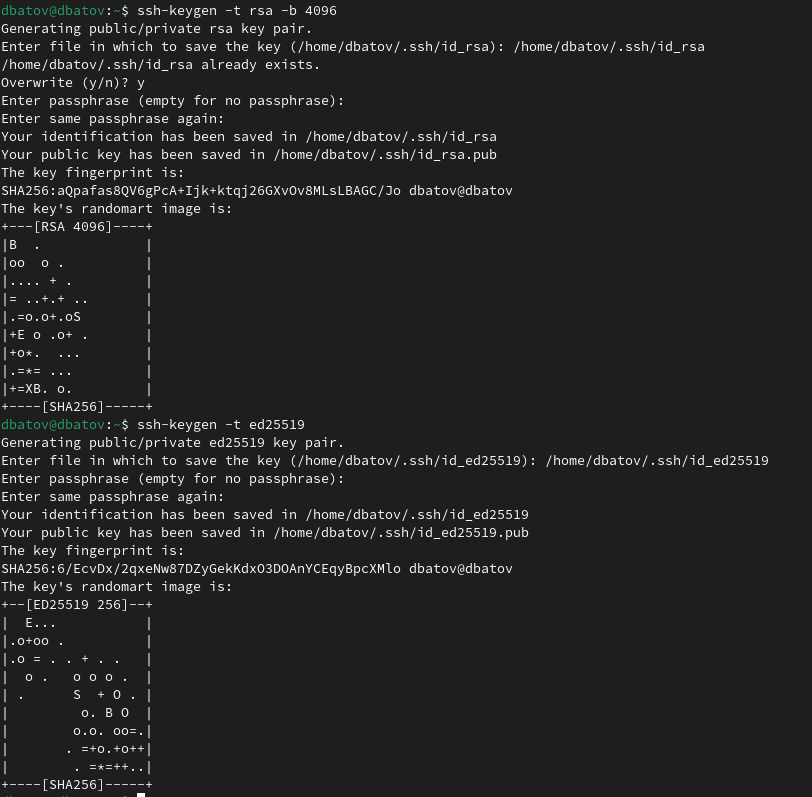
Цель работы – изучить идеологию и применение средств контроля версий, а также освоить умения по работе с git.

Git и gh уже были установлены на машину, поэтому я начну с базовой настройки.

1. Базовая настройка git

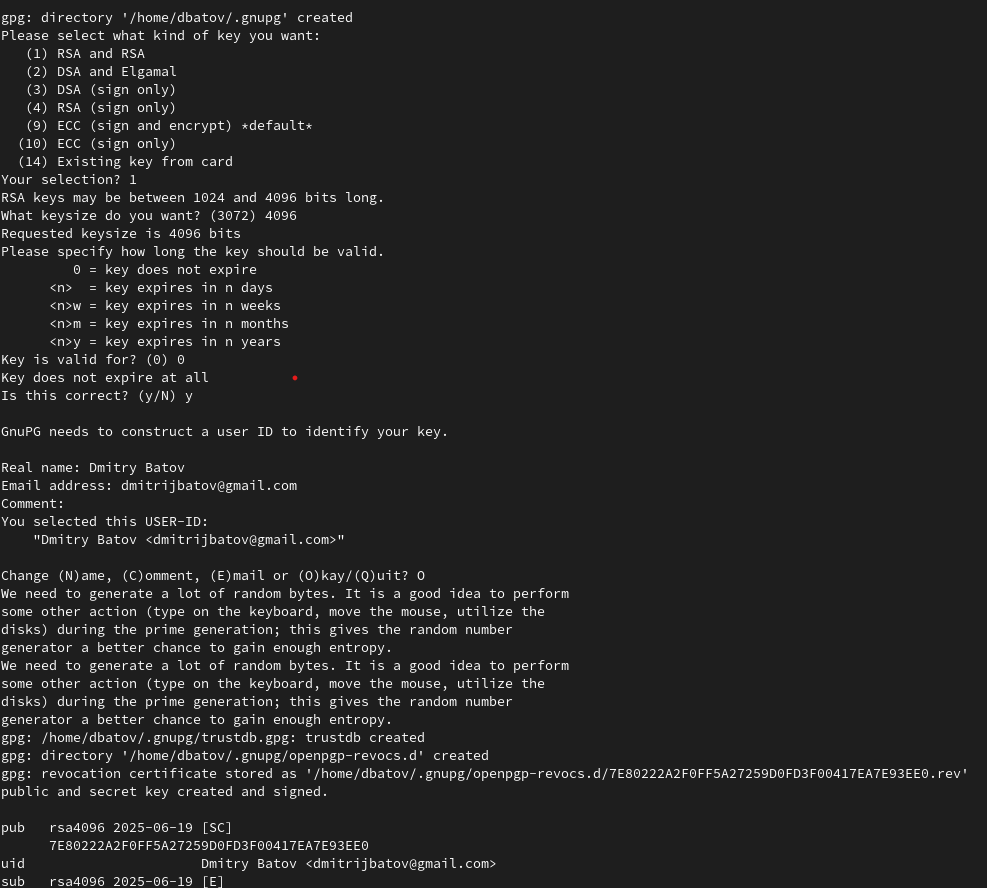


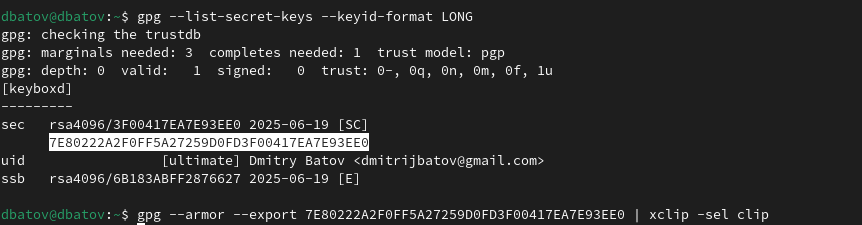
1. Создание ssh-ключа



После этого ssh-ключ, созданный по алгоритму ed25519, был загружен на гитхаб.

1. Создание и копирование gpg-ключа



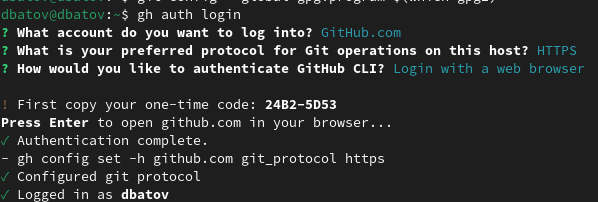


GPG-ключ также был внесен на гитхаб.

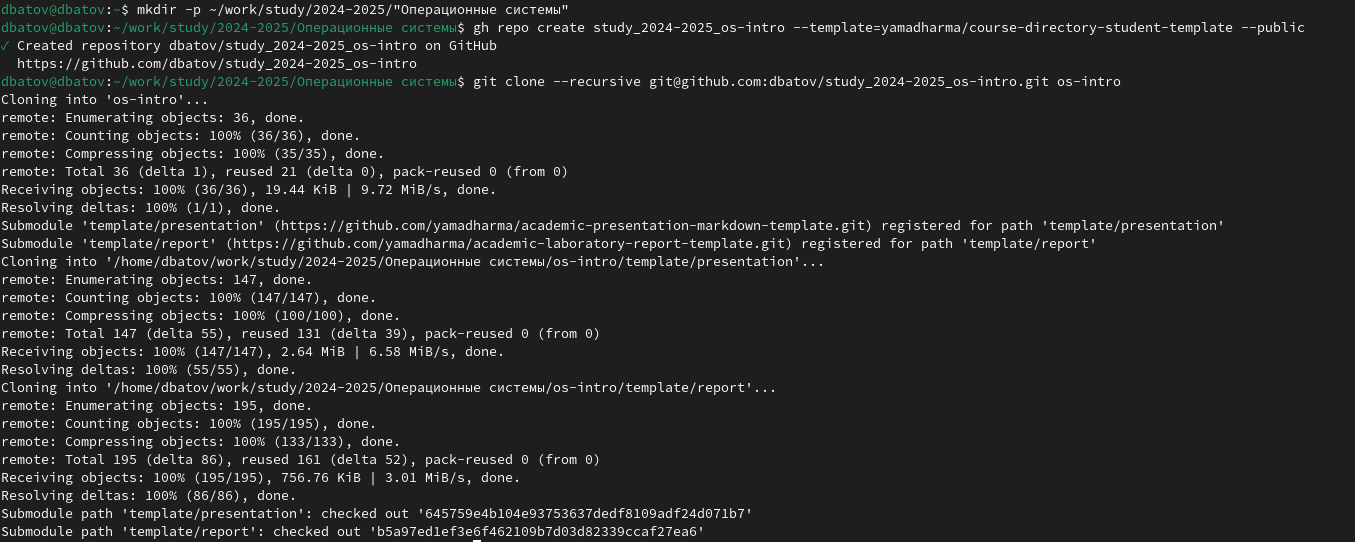
1. Настройка автоматических подписей коммитов git



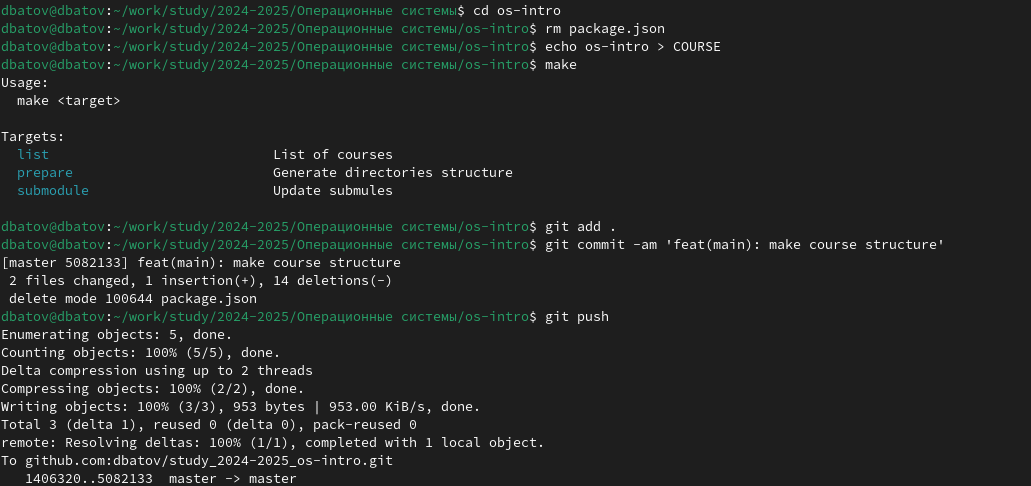
1. Настройка gh



1. Создание репозитория



1. Настройка каталога курса



Выводы

В ходе данной лабораторной работы мной были изучены идеология и применение средств контроля версий, а также были освоены умения по работе с git.

Ответы на контрольные вопросы

1. VCS – система, принимающая и регистрирующая изменения в файлах. Применяется при совместной работе над проектом группы людей, когда каждый должен видеть, кто и когда вносил какие изменения.
2. Хранилище – место, где хранятся файлы. Commit – команда, с помощью которой сохраняются изменения в хранилище. История – записи об изменениях в файлах в хранилище, внесенных коммитом. Рабочая копия – та, которую участник проекта извлекает из хранилища для работы.
3. Централизованная VCS – такая, где все изменения проходят строго через центральный сервер; чаще всего используются для хранения бэкап-версии файлов и отслеживания. Пример такой системы – Subversion или Perforce. Децентрализованная VCS – такая, в которой у каждого пользователя есть свой репозиторий. Такие системы используются для обмена изменениями, самый яркий пример – Git.
4. Создать локальный репозиторий, настроить его, работать с коммитами последовательностью команд add – commit – push
5. Для работы с общим хранилищем используется ssh-ключ, с помощью которого мы подтверждаем право доступа к хранилищу.
6. Основные задачи:
   1. Фиксирование изменений
   2. Совмещение изменений
   3. Хранение информации об изменениях
   4. Хранение ранних версий файла
7. Команды:
   1. init – создание основного дерева
   2. pull – получение обновлений с сервера
   3. add – добавление файлов к коммиту
   4. commit – сохранение изменений
   5. push – отправка изменений на сервер
   6. rm – удаление файлов
   7. diff – просмотр изменений
   8. merge – вливание ветки в основное дерево
8. Сначала файлы создаются локально (например, в папке репозитория мы можем создать файл file.doc), потом с помощью команды git add file.doc он добавляется в коммит, командой git commit –am ‘comment’ создается новый коммит и командой git push он отправляется на сервер.
9. Ветка создается при новом коммите для того, чтобы разные участники могли работать, не мешая друг другу, а после завершения задачи можно влить ветку в базовое дерево.
10. Файлы можно вносить в gitignore, если они в данный момент не нуждаются в загрузке на сервер или если они изначально для этого не предназначены и созданы разработчиком в личных целях.