**Факультет информационных технологий**

**Кафедра программной инженерии**

**Отчёт**

По дисциплине «Основы программной инженерии»

Выполнил:

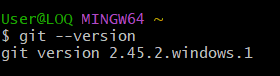
Студент 1 курса 10 группы ПИ

Макаревич Кирилл Витальевич

Проверила: Волчек Дарья Ивановна

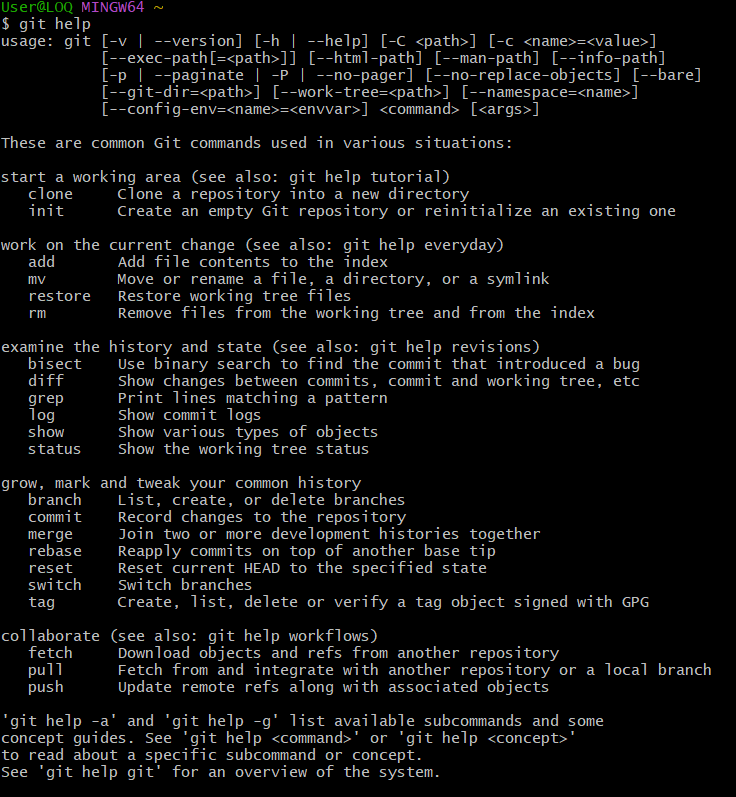
2024, Минск

Задание 4



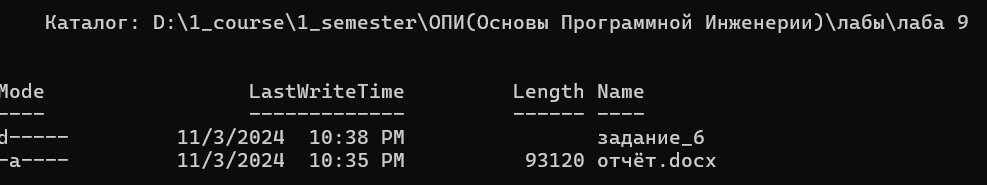
Задание 5

Чтобы получите справку по git используем команду git help. Эта команда предоставляет список основных команд git и краткое описание их функций. Если мы хотим узнать подробно про конкретную команду, то должны ввести git help <команда>.



Задание 6



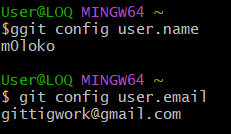


Мы создали локальный репозиторий

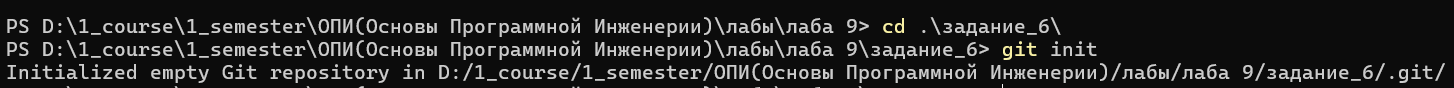
Задание 7

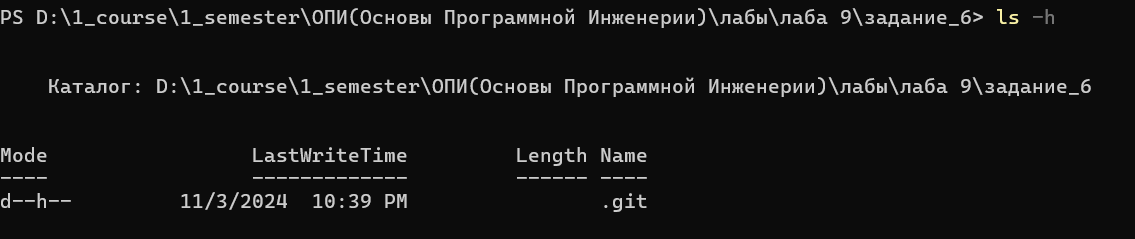
Сделал

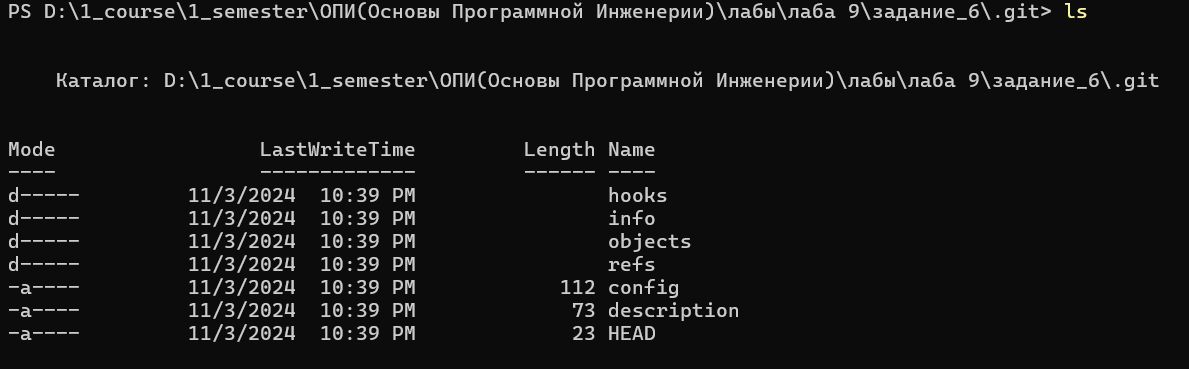
Задание 8



Задание 9

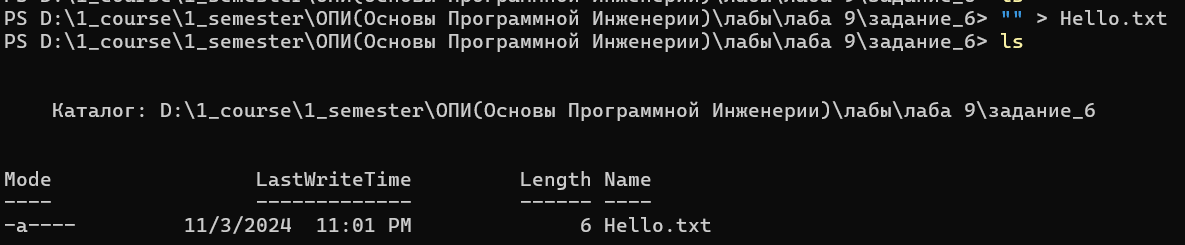




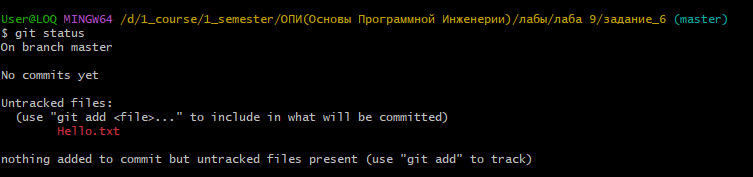


После команды git init внутри директории проекта создалась скрытая папка .git, где git будет хранить все необходимые данные: историю коммитов, информацию об изменениях, и т.д. После инициализации папка стала репозиторием git, и теперь можно использовать команды git для управления версиями файлов в этой папке.

Задание 10

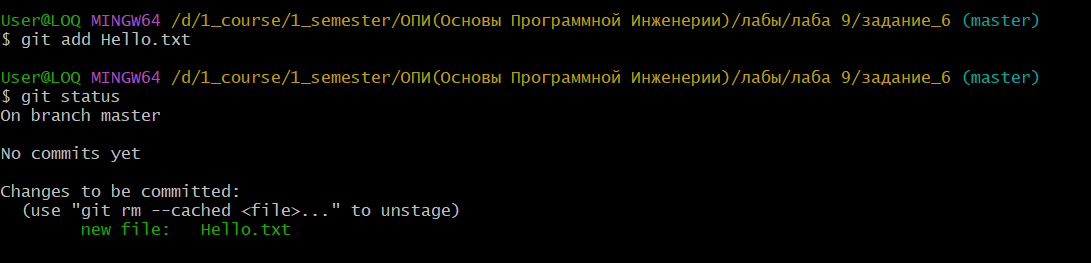


Задание 11



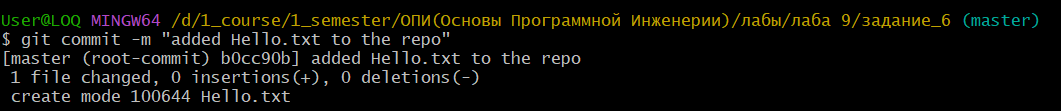
git status красным цветом отмечает новые и модифицированные файлы и папки

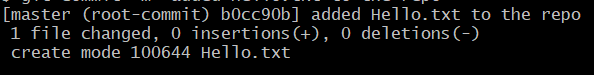
Задание 12



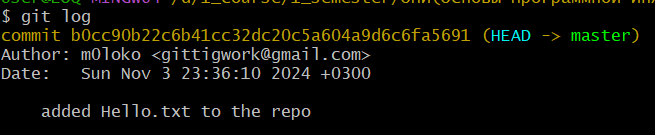
Файл Hello.txt добавлен в область индексирования. Git теперь отслеживает этот файл, но пока мы не выполнили команду git commit то это изменение не сохранено.

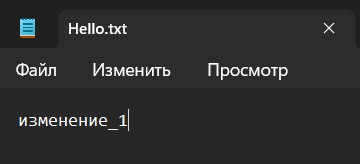
Задание 13

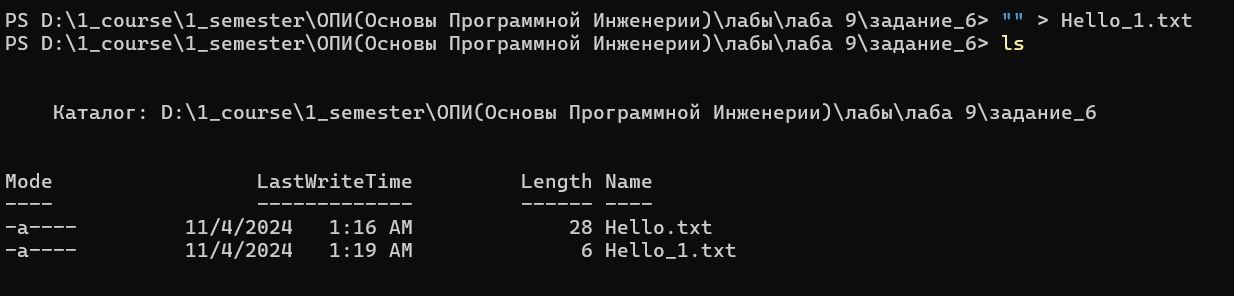


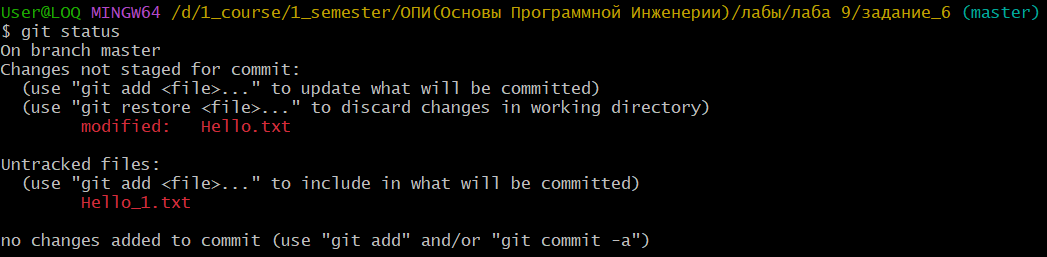
Индексированные изменения теперь сохранены в истории репозитория. Новый коммит содержит состояние файла Hello.txt. Коммит добавлен в мою главную ветку (master).Git отображает такое сообщение , после успешного создания нового коммита. 

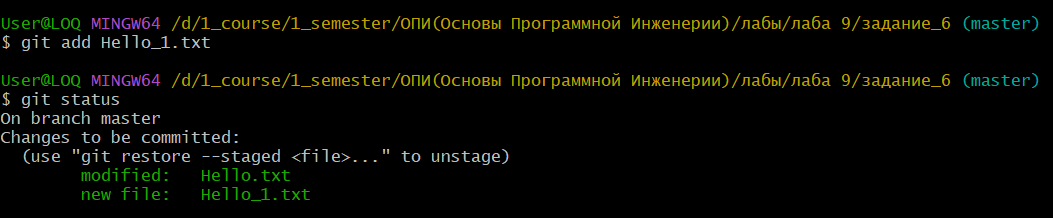
Задание 14

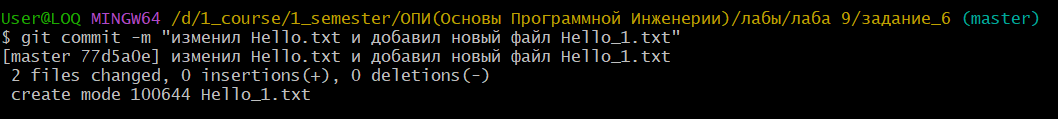
Задание 15

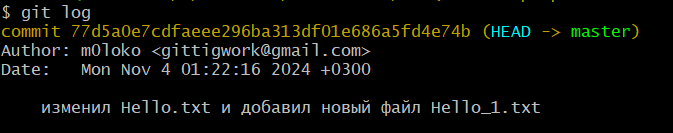












Задание 16

Научился пользоваться

Задание 17

<https://github.com/github/gitignore>

|  |
| --- |
| # Compiled Object files  \*.slo  \*.lo  \*.o  \*.obj  # Executables  \*.exe  \*.out  \*.app |

Вопросы:

1. Система контроля версий — это программное обеспечение, которое отслеживает изменения в файлах и сохраняет версии файлов на разных этапах их изменения. Она используется для управления версиями проектов, облегчает работу в команде, позволяет возвращаться к предыдущим версиям файлов и сравнивать изменения.

2. Принцип хранения файлов в Git основан на снимках (snapshots). Вместо записи изменений файла в каждую версию, Git сохраняет снимок (полное состояние) всех файлов на момент фиксации изменений, что позволяет быстро восстановить любое состояние репозитория.

3. Для повышения продуктивности вместо файлов, которые не претерпели изменений, сохраняется всего лишь ссылка на их ранее сохраненные версии + каждый пользователь имеет полную копию репозитория, включая всю его историю. Это позволяет работать офлайн и уменьшает зависимость от центрального сервера.

4.

1. Модифицированное (modified) - изменения уже внесены в файл, но пока не зафиксированы в базе данных.
2. Индексированное (staged) - вы пометили текущую версию модифицированного файла как предназначенную для следующей фиксации.
3. Зафиксированное (committed) - данные надежно сохранены в локальной базе

5. Индексация файла — это добавление файла в область подготовки (staging area), где он становится готовым для фиксации. Для этого используется команда:

git add <имя\_файла>

6. Фиксация файла — это процесс сохранения текущих изменений в истории версий в репозитории. Команда для фиксации:

git commit -m "Сообщение о фиксации"

7. git config –list

8. git init

9. git add <имя\_файла> или git add . ( за всеми файлами )

10. git commit -m "Сообщение о фиксации"

11. git status

12. Файл .gitignore создается для указания файлов и папок, которые git должен игнорировать и не включать в репозиторий. Этот файл помогает исключить временные или автоматически генерируемые файлы, например, логи или данные сборки. Структура .gitignore: каждый путь или шаблон указывается на отдельной строке, например:

# Игнорировать папку build

build/

# Игнорировать все файлы с расширением .log

\*.log