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** Дибиров Курбанали Арсенович:

**Группа:** НПМбв 01-18

**МОСКВА**

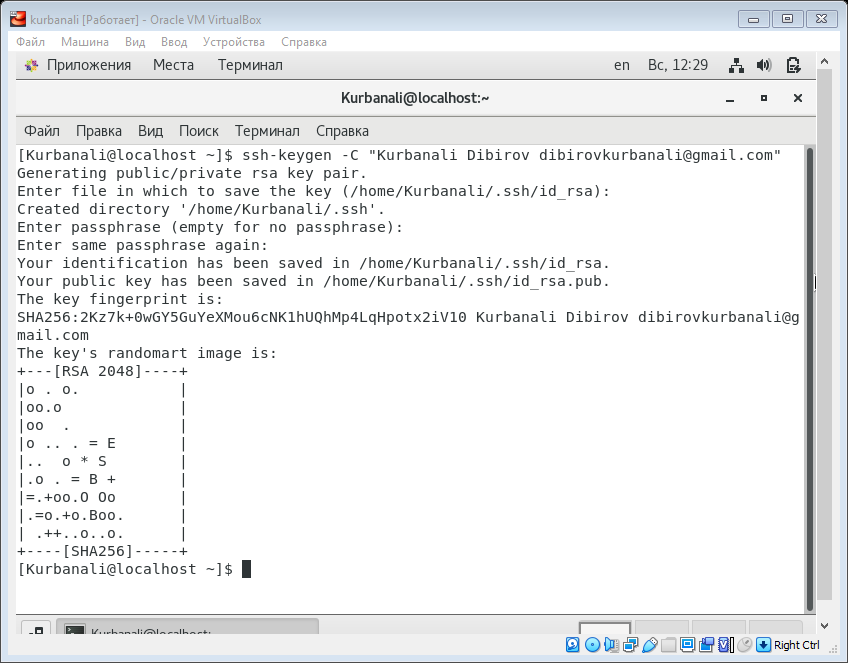
2022 г.

**Цель работы:** Изучить идеологию и применение средств контроля версий.

**Ход работы:** Настройка git

- Учетная запись уже была создана. https://github.com/DibirovKurbanali

- Генерация ключа:



- Создание структуры каталога лабораторных работ согласно пункту М.2.:

Изображение выглядит как текст

Автоматически созданное описание

Подключение репозитория к github.

- Создайте репозиторий на GitHub. Для примера назовём его sciproc-intro: https://github.com/DibirovKurbanali/sciproc-intro

– Рабочий каталог будем обозначать как laboratory. Вначале нужно перейти в этот каталог: cd laboratory

Изображение выглядит как стол

Автоматически созданное описание

Создаём README файл и добавляем его в коммит



Фиксируем коммит и добавляем комментарий

Изображение выглядит как текст

Автоматически созданное описание

Добавим файл лицензии

Изображение выглядит как текст

Автоматически созданное описание

Смотрим список gitignore

Изображение выглядит как текст

Автоматически созданное описание

Добавляем gitignore для языка С



Добавим новые файлы



Выполним коммит

Изображение выглядит как текст

Автоматически созданное описание

Отправим на github

Изображение выглядит как текст

Автоматически созданное описание

# Вывод

В ходе выполнения лабораторной работы №2 были приобретены навыки по администрированию и взаимодействию с децентрализованной системой контроля версий и программой git для параллельной поддержки программного кода.

# Контрольные вопросы

1. Системы контроля версий (VCS) применяются при работе нескольких человек над одним проектом, совместная работа путём изменения файлов в одном репозитории.
2. Понятия:
   * Хранилище - общее пространство для хранения файлов.
   * Commit - команда для записи индексированных изменений в репозиторий.
   * История - в истории сохраняются все коммиты, по которым можно отследить автора.
   * Рабочая копия - все файлы, кроме .git называются рабочей копией и принадлежат пользователям.
3. Централизованные системы контроля версий - сохраняют проект и его файлы на один общий сервер, децентрализованные системы контроля версий - при каждом копировании данных удалённого репозитория происходит полное копирование данных в локальный репозиторий. Пример ЦСКВ - SVN, MS TFS, ClearCase. Пример ДСКВ - Git, Mercurical, Bazaar.
4. Действия:
   * Создаём репозиторий и именуем его
   * Добавляем файлы, которые нужны зафиксировать
   * Фиксируем коммитом файлы
   * Фиксиурем изменения на сервер
5. Действия:
   * Создаём репозиторий или подключаемся к нему
   * Создаём и добавляем файлы для коммита
   * Фиксируем коммитом
   * Отправляем коммит на сервер
   * Ждём проверки на внесения изменений от других пользователей
6. Систематизация, параллельность разработки программного обеспечения, единое место для хранения файлов проекта.
7. git init - создание репозитория, git clone - клонирование репозитория, git add - добавление изменений в индекс, git reset - удаление изменений из индекса, git commit - коммит, git rm - удаление файла.
8. Для написания черновых работ по лабраторным работам я использую локальные репозитории, для их распространения или для оценивания преподавателем я использую удалённый репозиторий git.
9. Ветви служат для параллельной разработки программного обеспечения, тестирования, отладки и улучшения.
10. Игнорирование можно установить для проекта, компьютера и репозитория, цель игнорирования заключается в том, чтобы не отслеживать файлы служебного типа, например временные файлы сборных утилит для проектов или только те файлы, которые полезны при взаимодействии только с очень ограниченным программным обеспечением.