**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 2**

*дисциплина: Архитектура компьютера*

Студент: Ягодин Максим Сергеевич

Группа: НПИбд-03-24

**МОСКВА**

2025г.

**Содержание**

1. [Цель работы](#_bookmark0)
2. [Выполнение лабораторной работы](#_bookmark2)
3. Контрольные вопросы
4. [Выводы](#_bookmark28)

# Цель работы

# Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умений работать с git

# **Выполнения лабораторной работы**

# Для начала мы установим git и gh

# Снимок экрана 2025-06-25 120233

# Снимок экрана 2025-06-25 120247

# После мы настроим git, для этого мы задажим имя и email владельца репозитория

# Снимок экрана 2025-06-25 120337

# Также настроим usf-8 в выводе сообщений git

# Снимок экрана 2025-06-25 120411

# И зададим имя начальной ветке как master, параметр autocrlf и safecrtf

# Снимок экрана 2025-06-25 120425

# Снимок экрана 2025-06-25 120445

# Снимок экрана 2025-06-25 120508

# Теперь мы можем создать ключи ssh по алгоритму rsa, рамер ключа будет 4096 бит

# Снимок экрана 2025-06-25 120550

# И по алгоритму ed25519

# Снимок экрана 2025-06-25 120615

# Также создадим ключ pgp, выбираем варианты из предложенных опций который нам даны в инструкции для лабораторной работы

# Снимок экрана 2025-06-25 120823

# Дальше нам предлагают создать учётную запись github, но этот шаг мы пропустим по сколько у нас уже имеется учётная запись

# Переходим к следующему шагу, добавление PGP ключа в GitHub

# Для этого выводим список ключей и копируем отпечаток приватного ключа

# Снимок экрана 2025-06-25 121041

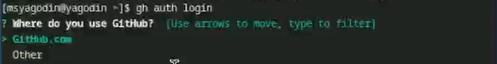
# И вводим полученный ключ в Github

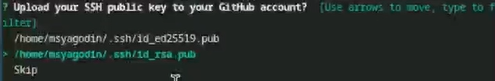
# Снимок экрана 2025-06-25 121113

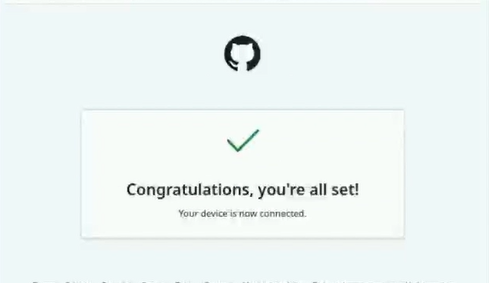
Настроим автоматические подписи коммитов git, используя введёный email, укажим Git применим его при подписи коммитов

Снимок экрана 2025-06-25 121157

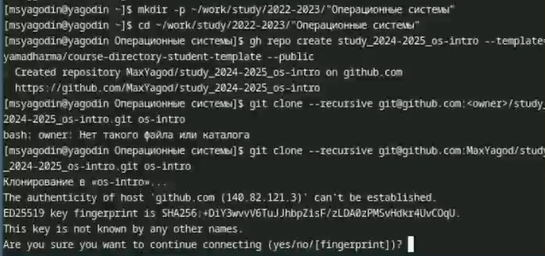
Теперь настроим Gh, для начала нам необходимо авторизоваться



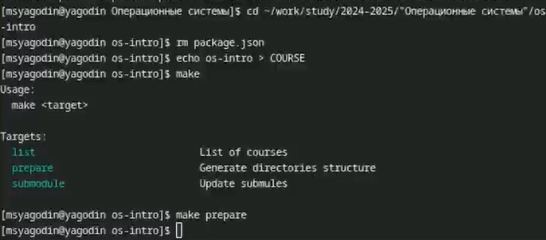
Снимок экрана 2025-06-25 121238



Далее мы создадим репозиторий на основе шаблона который нам дан



Переходи в каталог курса удаляем лишние файлы и создаём необходимые каталоги



И теперь мы отправляем файлы на сервер

Снимок экрана 2025-06-25 121719Снимок экрана 2025-06-25 121737

**Вывод**

Мы приобрели практические навыки работы с сервисом github.

**Контрольные вопросы**

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

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

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

1 хранилище - пространство на накопителе где расположен репозиторий

2 commit - сохранение состояния хранилища

3 история - список изменений хранилища (коммитов)

4 рабочая копия - локальная копия сетевого репозитория, в которой работает программист. Текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней)

1. 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. Распределенные 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”.

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

Один пользователь работает над проектом и по мере необходимости делает коммиты, сохраняя определенные этапы.

1. Опишите порядок работы с общим хранилищем VCS.

Несколько пользователей работают каждый над своей частью проекта. При этом каждый должен работать в своей ветки. При завершении работы ветка пользователя сливается с основной веткой проекта.

1. Каковы основные задачи, решаемые инструментальным средством git?

1 Ведение истории версий проекта: журнал (log), метки (tags), ветвления (branches).

2 Работа с изменениями: выявление (diff ), слияние (patch, merge).

3 Обеспечение совместной работы: получение версии с сервера, загрузка обновлений на сервер.

1. Назовите и дайте краткую характеристику командам git.

1 git config - установка параметров • git status - полный список изменений файлов, ожидающих коммита

2 git add . - сделать все измененные файлы готовыми для коммита.

3 git commit -m “[descriptive message]” - записать изменения с заданным сообщением.

4 git branch - список всех локальных веток в текущей директории.

5 git checkout [branch-name] - переключиться на указанную ветку и обновить рабочую директорию.

6 git merge [branch] — соединить изменения в текущей ветке с изменениями из заданной.

7 git push - запушить текущую ветку в удаленную ветку.

8 git pull - загрузить историю и изменения удаленной ветки и произвести слияние с текущей веткой.