

# 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**
- 2 Выполнение лабораторной работы**
- 3 Контрольные вопросы**
- 4 Выводы**

## 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

```
[msyagodin@yagodin ~]$ sudo dnf install git
[sudo] пароль для msyagodin:
Обновление и загрузка репозитория:
Репозитории загружены.

[msyagodin@yagodin ~]$ sudo dnf install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет
Установка:
gh
Арх.
x86_64
Версия
2.74.0-1.fc42
Репозиторий
updates
Размер
38.9 MiB
```

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

```
[msyagodin@yagodin ~]$ git config --global user.name "MaxYagod"
[msyagodin@yagodin ~]$ git config --global user.email "maxagodin@gmail.com"
```

Также настроим usf-8 в выводе сообщений git

```
[msyagodin@yagodin ~]$ git config --global core.quotepath false
[msyagodin@yagodin ~]$
```

И зададим имя начальной ветке как master, параметр autocrlf и safecrlf

```
[msyagodin@yagodin ~]$ git config --global init.defaultBranch master
[msyagodin@yagodin ~]$
```

```
[msyagodin@yagodin ~]$ git config --global core.autocrlf input
[msyagodin@yagodin ~]$
```

```
[msyagodin@yagodin ~]$ git config --global core.safecrlf warn
[msyagodin@yagodin ~]$
```

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

```

[msyagodin@yagodin ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/msyagodin/.ssh/id_rsa):
Created directory '/home/msyagodin/.ssh'.
Enter passphrase for "/home/msyagodin/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/msyagodin/.ssh/id_rsa
Your public key has been saved in /home/msyagodin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:U69sBAOG/YwW0xLzPPDq8XNI70n1KGNvBkkR30hriAQ msyagodin@yagodin
The key's randomart image is:
+---[RSA 4096]---+
|      oo..+      |
|    E =.o.+ .    |
|      B Bo..     |
|      o X =+ .    |
|      . = =5oo .  |
|      + o =* +    |
|      . + +=,= .  |
|      . +oo=,     |
|      +ooo        |
+-----[SHA256]-----+
[msyagodin@yagodin ~]$

```

И по алгоритму ed25519

```

[msyagodin@yagodin ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/msyagodin/.ssh/id_ed25519):
Enter passphrase for "/home/msyagodin/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/msyagodin/.ssh/id_ed25519
Your public key has been saved in /home/msyagodin/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:2MOGXThZmBkm2+8ztRbWliezUUUktqnOCZn2FDztbE08 msyagodin@yagodin
The key's randomart image is:
+--[ED25519 256]--+
|      +* . . .+  |
|    .+* o o.E|   |
|      = o . = *   |
|    * o . * +. +  |
|    o S. o.B++ .  |
|    . = . =o o    |
|    + . ++.o      |
|      +           |
+-----[SHA256]-----+
[msyagodin@yagodin ~]$

```

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

```
[msyagodin@yagodin ~]$ gpg --full-generate-key
gpg (GnuPG) 2.4.7; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: создан каталог '/home/msyagodin/.gnupg'
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (только для подписи)
 (14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
  <n> = срок действия ключа - n дней
  <n>w = срок действия ключа - n недель
  <n>m = срок действия ключа - n месяцев
  <n>y = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: MaxYagod
Адрес электронной почты: maxagodin@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "MaxYagod <maxagodin@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? █
```

Далее нам предлагают создать учётную запись `github`, но этот шаг мы пропустим поскольку 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

```
[msyagodin@yagodin ~]$ gpg --armor --export maxagodin@gmail.com | xclip -sel clip
```

И вводим полученный ключ в Github



<> Developer settings

### Add new GPG key

Title

Key

```
QCRQkmbOkYscenMqu7Ujfb9ASw+bqrT6pn8hMIUZAtnFjd70Llc97ICbjNMd1phj
TSPeULWdMPt62/+vcwQOIuydsYw15Gxy0CpO+Uq5hKS3f5ikkbgQepMGxYdxRoRE
PIL8WkhhtDEc53wj5KJQRiDREq74E+NUgHevWD8XsNroc4ucfkGQAL+d1qsX5cR4
spxNo+YbGt5T7D072V/tNskUgpC7bC8AII1MJ4IJS16tazUOQnrB6dQeUUuImf8A
Mm5aQfBd1OSu56WNQ56MEFdcv75sJtvmblvJUAzHvLcyK8BU6ID+GvB5Oj3fOmY
CSesKcbwKpP42s2OqRx/LiwoD3lRdQk5HKbYWkTTgDjmTSPKdvCLnbcO20DMTPVG
xxhhwRcvYPUCOdI23SQF4y3UPkuM0ZwxE1GM/2mq6/Tlo7/Cwj0r7d8kDNrTZg==
=WQV9
-----END PGP PUBLIC KEY BLOCK-----
```

Add GPG key

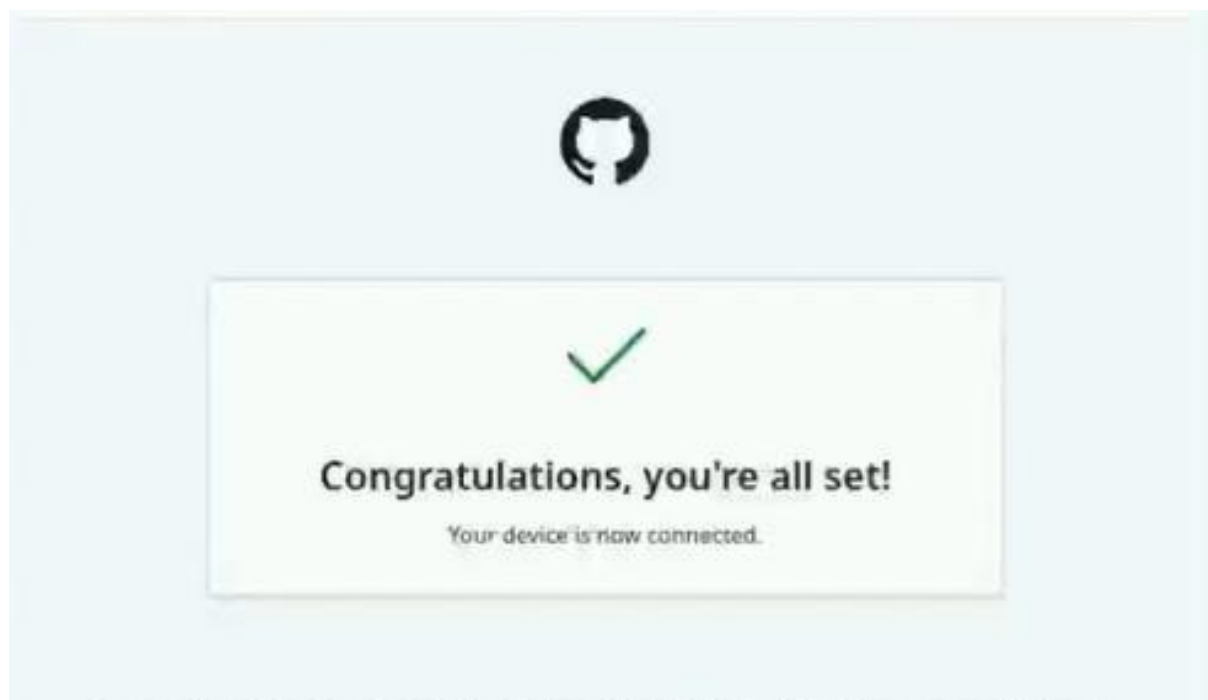
Настроим автоматические подписи коммитов git, используя введённый email, укажем Git применим его при подписи коммитов

```
[msyagodin@yagodin ~]$ git config --global user.signingkey maxagodin@gmail.com
[msyagodin@yagodin ~]$ git config --global commit.gpgsign true
[msyagodin@yagodin ~]$ git config --global gpg.program $(which gpg2)
```

Теперь настроим Gh, для начала нам необходимо авторизоваться

```
[msyagodin@yagodin ~]$ gh auth login
? Where do you use GitHub? (Use arrows to move, type to filter)
> GitHub.com
Other
```

```
? What is your preferred protocol for Git operations on this host? [Use arrows to move, type to filter]
  HTTPS
> SSH
? Upload your SSH public key to your GitHub account? [Use arrows to move, type to filter]
  /home/msyagodin/.ssh/id_ed25519.pub
> /home/msyagodin/.ssh/id_rsa.pub
Skip
```



Далее мы создадим репозиторий на основе шаблона который нам дан

```
[msyagodin@yagodin ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
[msyagodin@yagodin ~]$ cd ~/work/study/2022-2023/"Операционные системы"
[msyagodin@yagodin Операционные системы]$ gh repo create study_2024-2025_os-intro --template yamadharna/course-directory-student-template --public
Created repository MaxYagod/study_2024-2025_os-intro on github.com
https://github.com/MaxYagod/study_2024-2025_os-intro
[msyagodin@yagodin Операционные системы]$ git clone --recursive git@github.com:<owner>/study_2024-2025_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[msyagodin@yagodin Операционные системы]$ git clone --recursive git@github.com:MaxYagod/study_2024-2025_os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvvV6TuJJhbpZ1sF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```



Переходи в каталог курса удаляем лишние файлы и создаём необходимые каталоги

```
[msyagodin@yagodin Операционные системы]$ cd ~/work/study/2024-2025/"Операционные системы"/os-intro
[msyagodin@yagodin os-intro]$ rm package.json
[msyagodin@yagodin os-intro]$ echo os-intro > COURSE
[msyagodin@yagodin os-intro]$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule      Update submules

[msyagodin@yagodin os-intro]$ make prepare
[msyagodin@yagodin os-intro]$
```

И теперь мы отправляем файлы на сервер

```
[msyagodin@yagodin os-intro]$ git add .
[msyagodin@yagodin os-intro]$ git commit -am 'feat(main): make course structure'
[msyagodin@yagodin os-intro]$ git push
```

## **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

2. 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)

3. 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”.

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

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

5. 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.

6. 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.

7. 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 - запустить текущую ветку в удаленную ветку.

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