

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: Виноградова Мария Андреевна

Группа: НПИбд-01-24

МОСКВА

2025г.

ЦЕЛЬ РАБОТЫ:

- Изучить идеологию и применение средств контроля версий.
 - Освоить умения по работе с git.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ:

1. Установка git и gh

Устанавливаю git

```
[mavinogradova@vbox ~]$ sudo -i
[sudo] пароль для mavinogradova:
[root@vbox ~]# dnf install git
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.

Нечего делать.
[root@vbox ~]#
```

Рис.1.1 установка git

Установка gh

```
[root@vbox ~]# dnf install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "gh-2.65.0-1.fc41.x86_64" уже установлен.

Нечего делать.
[root@vbox ~]#
```

Рис.1.2 установка gh

2. Базовая настройка git

Зададим имя и email владельца репозитория

```
[root@vbox ~]# git config --global user.name "MASHA VINOGRADOVA"
[root@vbox ~]# git config --global user.email "1132240691@pfur.ru"
[root@vbox ~]#
```

Рис.2.1 задаю имя и email владельца

Настроим utf-8 в выводе сообщений git:

```
[root@vbox ~]# git config --global core.quotepath false
[root@vbox ~]#
```

Рис.2.2 настраиваю utf-8 в выводе сообщений

Настраиваю верификацию и подписание коммитов git. И задаю имя начальной ветки (буду называть её master)

```
[root@vbox ~]# git config --global init.defaultBranch master
[root@vbox ~]# git config --global core.autocrlf input
[root@vbox ~]# git config --global core.safecrlf warn
[root@vbox ~]#
```

Рис.2.3 настраиваю верификацию и задаю имя ветки

3. Создание SSH ключей

По алгоритму rsa создаю ключ размером 4096 бит:

```
ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /root/.ssh/id_rsa
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase for "/root/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:EtUMENUHkwseqSKbxoMCgQ2IjbpguMMgGDkz1PVxbJM root@vbox
The key's randomart image is:
+---[RSA 4096]-----+
|+.=. .o=B0o.      |
|@ .. .+E+        |
|=B   ...o.       |
|X..  .=          |
|*+   .+S.        |
|=.. . . .        |
|. + + .          |
|o *              |
|.. .            |
+---[SHA256]-----+
[root@vbox ~]#
```

Рис.3.1 создаю ключ

По алгоритму ed25519 создаю ключ

```
[root@vbox ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519): /root/.ssh/id_ed25519
/root/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase for "/root/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:CIvtbd20x0iR016WrUnC8FHBi9MJy5dg6WoHalvzUuE root@vbox
The key's randomart image is:
+--[ED25519 256]--+
|      +=..      |
|      o+o=o     |
|      ..oo+.    |
|      o.oo+     |
|      ..++o.5   |
|      o.==E* +   |
|      . +=X 0 o  |
|      . .oo0 =   |
|      o+.+.     |
+-----[SHA256]-----+
[root@vbox ~]#
```

Рис.3.2 создаю ключ

4. Создание ключей *pgp*

Генерирую ключ

```
[root@vbox ~]# gpg --full-generate-key
gpg (GnuPG) 2.4.5; 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.

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: masha
Адрес электронной почты: 1132240691@pfur.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "masha <1132240691@pfur.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? O
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.

[root@vbox ~]#
```

Рис.4.1 генерация ключа

Добавляю SSH ключ на github

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys



	Title
	SHA256:PHffxJKcJEK9VlrGEMiOCZmnzp1mOK2H+3CzTu7CpR8
	Added on Sep 28, 2024
	Last used within the last 5 months — Read/write
	

Рис.4.2 добавляю ключ на github

5. Настройка github

Учетная запись была создана мною ранее для выполнения предыдущих лабораторных работ

6. Добавление PGP ключа в GitHub

Вывожу список ключей и копирую отпечаток приватного ключа

```
[root@vbox .gnupg]# gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]
-----
sec   rsa4096/03D66D19B6768A30 2025-03-04 [SC]
      260AA860A92D20A83E12B64203D66D19B6768A30
uid           [ абсолютно ] masha <1132240691@pfur.ru>
ssb   rsa4096/B1E0571297A9ABEF 2025-03-04 [E]

[root@vbox .gnupg]#
```

Рис.6.1 выводжу список ключей

Копирую свой сгенерированный PGP ключ в буфер обмена

```
[root@vbox .gnupg]# gpg --armor --export 260AA860A92D20A83E12B64203D66D19B6768A30 x
clip -sel clip
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGfHiXgBEADuts2vmI07I4sx+Lj5v54dmA5xF9nRIKJQhNncUS0e99zTd78H
RIyEJgtFtt6n6duWjyohY5bq+41u7Mkej7npr+XexIeA3L9PSu0sUsmogjZ/sx3
VuhczoESTNX+2x1wu0EqbYX1F2w/IsxbZbmkgj2VMZdw0udC3Fetn2/illIenxq
Q0smkyCsXeHx0qiAeWKKPK110wyV5UamwhDTLu1uwvTxhJk8R3HTz4rN0K1W2aq7
Q+ph9YT1kH3okj7jgcyUmd+RNvhkZUAihhTlwb6dWIXIKZod1Ee8rDdVx/obk3Fu
UI94vwYs1kmB8yLL2Frohm9zW1Z0nBFfGt6ukKNjIPytagpsJX0V1xkzeepquIIZ
OqpRfMTFwk4QryFFZZzNwFNK8iM1Y9HYVxH2aAYCoUPKnig64AsuM1g2B3eZFek3
aI3AHX28kjan1+JL9t71EWDROXOTdgDv0DpXZIJQPIrX4qUsfXBw8P4oIA+R94qh
ypQ9kTJwG5WfP6qDNUls/FV+uf5aQL9TyVQUxmH5mFLcgxKxEf16ay7MDe3htAQw
10tVODpE2EmuEq3rR0G1ynETu7CxT+b6xtA5cYjBSYdC1XQIWLadJ4j1kJFdYY4x
H+eXTqwt4sDkN3ef0eSRV1wK7dJN+7+ynQX6S0mNueHJH+zw++7/RdKohwARAQAB
tBptYXNoYSA8MTEzOTY1MjY2OTQ0OTQ0OTQ0OTQ0OTQ0OTQ0OTQ0OTQ0OTQ0OTQ0
PhK2QgPwBmRm2doowBQJnx414AhsDBQsJCAcCAiICBhUKCQGLAgQWAgMBAh4HAheA
AAoJEAPlwBmRm2doowts0P93zxmVscIfMNSyS5WOMk46bCqWL3vf1XTjzhcQupZHJR
dx13+sxAy7iQVHHS+WP9X1H6snmA3zh0gsX9D3xtRY1ZtJ1+7BwqHvE7Bx5vChDJ
ztj0cInFRANEHuoXCZSNQ1ZYkG1MrWDZUU2W+9XbIyRaCNrc5sGVj7YLTWzGRaLF
Nb3XUC6GN6suzgYpVgR+CkKQJ++20EC7WLlhrf4W80xzwkZ7W9mTYj8Rq50EaVJ1
DCNrVQOGW61GJu8yvNZ+80Ep1+y7MGUJ+iCoEuNpbU4D/PbiDS0rp8Na18Rj+/jE
j2Io0eDNIAbuKgl2Am7atpjPwzhYRK88JxZm+ze1CM7fFZYs+xZlgkn60KgVIAcN
8d4pHEOBGhgtLn1xADgE3IPVZPvtf2buRYGyqX7/BY4TsHvLbszuJtUfkmb7NupE
lsOyUeMx/sCsRHYEij8pFQUEdQ50JR+7+oD0IW2XivK66KqDDKchylLyp0VMj1sk
H9WUzxfTic05E1Nx73n3G9b7W7oAamI4Hn1L0TEdLUN8wicmGYsYm1Mbk409V8h3
```

Рис.6.2 копирую ключ

Перехожу в настройки GitHub, нажимаю на кнопку New GPG key и вставляю полученный ключ в поле ввода.

GPG keys

[New GPG key](#)

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.



Рис.6.3 добавляю ключ на github

7. Настройка автоматических подписей коммитов git

Используя введённый email, указываю Git применять его при подписи коммитов

```
[root@vbox .gnupg]# git config --global user.signingkey 260AA860A92D20A83E12B64203D66D19B6768A30
[root@vbox .gnupg]# git config --global commit.gpgsign true
[root@vbox .gnupg]# git config --global gpg.program $(which gpg2)
[root@vbox .gnupg]#
```

Рис.7.1 настраиваю подписи коммитов

8. Настройка gh

Для начала авторизуюсь

```
[root@vbox .gnupg]# gh auth login
? Where do you use GitHub? Other
? Hostname: mavinogradova
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

error connecting to mavinogradova
check your internet connection or https://githubstatus.com
[root@vbox .gnupg]# gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 8E40-EBCC
Press Enter to open https://github.com/login/device in your browser...
Authorization required, but no authorization protocol specified

Error: cannot open display: :0
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
! Authentication credentials saved in plain text
✓ Logged in as mavinogradova
[root@vbox .gnupg]#
```

Рис.8.1 авторизуюсь и отвечаю на вопросы утилиты

После этого авторизуюсь через браузер (ввожу код в предложенное окно)

9. Создание репозитория курса на основе шаблона

Создаю шаблон рабочего пространства

```
[root@vbox ~]# mkdir -p ~/work/study/2024-2025/"Операционные системы"
[root@vbox ~]# cd ~/work/study/2024-2025/"Операционные системы"
[root@vbox Операционные системы]# gh repo create study_2024-2025_os-intro --template=yamadharma/course-directory-student-template --public
GraphQL: Could not clone: Name already exists on this account (cloneTemplateRepository)
[root@vbox Операционные системы]# git clone --recursive git@github.com: mavinogradova/study_2024-2025_os-intro.git os-intro
fatal: Слишком много аргументов.

использование: git clone [<опции>] [--] <репозиторий> [<каталог>]

-v, --[no-]verbose      быть многословнее
-q, --[no-]quiet        тихий режим
--[no-]progress         принудительно выводить прогресс
--[no-]reject-shallow   don't clone shallow repository
-n, --no-checkout       не переключать рабочую копию на HEAD
--checkout              opposite of --no-checkout
--[no-]bare             создать голый репозиторий
--[no-]mirror           create a mirror repository (implies --bare)
-l, --[no-]local        для клонирования из локального репозитория
--no-hardlinks          не использовать жесткие ссылки, всегда копировать файлы
--hardlinks             opposite of --no-hardlinks
-s, --[no-]shared       настроить как общедоступный репозиторий
--[no-]recurse-submodules=<спецификатор-пути>
```

Рис.9.1 создаю рабочее пространство

10. Настройка каталога курса

Перехожу в каталог курса, удаляю лишние файлы и создаю необходимые каталоги

```
cd ~/work/study/2024-2025/"Операционные системы"/os-intro
[root@vbox os-intro]# rm package.json
rm: удалить обычный файл 'package.json'? y
[root@vbox os-intro]# echo os-intro > COURSE
make
Usage:
  make <target>

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

[root@vbox os-intro]#
```

Рис.10.1 удаляю файлы и создаю каталоги внутри каталога курса

Отправляю файлы на сервер

```
[root@vbox os-intro]# git add .
[root@vbox os-intro]# git commit -am 'feat(main): make course structure'
[master 5a657a9] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
[root@vbox os-intro]# git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 953 байта | 953.00 КиБ/с, готово.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:mavinogradova/study_2024-2025_os-intro.git
   8c5d05c..5a657a9  master -> master
[root@vbox os-intro]#
```

Рис.10.2 отправляю файлы на сервер

Контрольные вопросы

1. Системы контроля версий (VCS) — это программные инструменты, предназначенные для управления изменениями в исходном коде или других файлах.

Они позволяют:

Сохранять историю изменений.

Отслеживать, кто и когда внёс изменения.

Возвращаться к предыдущим версиям файлов.

Совместно работать над проектами без конфликтов.

Создавать ветки для параллельной разработки.

2. Хранилище (репозиторий): Место, где хранится вся история изменений проекта, включая файлы, коммиты и метаданные.

Commit (фиксация): Отдельная запись в истории изменений, которая содержит изменения файлов, комментариев и автора.

История: Последовательность коммитов, которая показывает, как проект развивался с течением времени.

Рабочая копия: Текущая версия файлов проекта, с которой работает пользователь. Она может быть изменена, а затем зафиксирована в репозитории.

Отношения:

Рабочая копия берётся из хранилища.

Изменения в рабочей копии фиксируются в виде коммитов.

Коммиты формируют историю проекта.

3. Централизованные VCS:

Все изменения хранятся на центральном сервере.

Пользователи работают с локальными копиями, но для фиксации изменений требуется подключение к серверу.

Пример: SVN (Subversion).

Децентрализованные VCS:

Каждый пользователь имеет полную копию репозитория, включая всю историю изменений.

Работа возможна без постоянного подключения к серверу.

Пример: Git, Mercurial.

4. Инициализация репозитория; Добавление файлов в отслеживание; Фиксация изменений; Просмотр истории изменений; Возврат к предыдущей версии

5. Опишите порядок работы с общим хранилищем VCS.

Клонирование удалённого репозитория; Создание новой ветки для работы; Внесение изменений и их фиксация; Отправка изменений на удалённый репозиторий; Получение изменений от других разработчиков; Слияние веток (например, после code review)

6. Управление версиями файлов.

Совместная работа над проектами.

Создание и управление ветками.

Отслеживание изменений и их авторов.

Резервное копирование и восстановление проекта.

7. git init: Инициализация нового репозитория.

git clone <URL>: Клонирование удалённого репозитория.

git add <файл>: Добавление файлов в индекс для последующего коммита.

git commit -m "сообщение": Фиксация изменений.

git status: Просмотр состояния рабочей копии.

git log: Просмотр истории коммитов.

git push: Отправка изменений на удалённый репозиторий.

git pull: Получение изменений из удалённого репозитория.

git branch: Управление ветками.

git merge: Слияние веток.

8. Локальный репозиторий:

Создание репозитория: git init

Добавление файлов: git add .

Удалённый репозиторий:

Клонирование: `git clone`

Отправка изменений: `git push origin main`

Получение изменений: `git pull origin main`

9. Ветви (branches) — это отдельные линии разработки в репозитории. Они нужны для:

Параллельной работы над разными функциями или исправлениями.

Изоляции экспериментальных изменений.

Упрощения процесса code review и слияния изменений.

10. Для игнорирования файлов используется файл `.gitignore`. В него добавляются шаблоны файлов или каталогов, которые не должны отслеживаться Git. Это полезно для:

Исключения временных файлов (например, кэша или логов).

Игнорирования конфиденциальных данных (например, паролей или ключей).

ВЫВОД

В ходе выполнения лабораторной работы были изучены основные принципы идеологии и применения систем контроля версий (VCS), в частности, инструмента **Git**. Освоены ключевые навыки работы с Git, включая создание и клонирование репозитория, фиксацию изменений, работу с ветками, а также взаимодействие с удалёнными репозиториями. Были выполнены практические задания по управлению версиями файлов, что позволило понять важность и удобство использования VCS для совместной разработки и контроля изменений в проектах.